

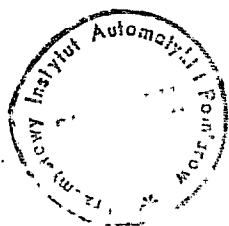
4

**INSTYTUT TECHNOLOGII MASZYN I AUTOMATYZACJI
POLITECHNIKI WROCŁAWSKIEJ**

**SIECI KOMUNIKACYJNE W ZASTOSOWANIACH
PRZEMYSŁOWYCH - ZAGADNIENIA SYSTEMOWE**

DOKUMENT WZORCOWY

**KRZYSZTOF SKURA (red.)
ZBIGNIEW SMALEC (red.)
MARIUSZ MRZYGLÓD
JACEK REINER
PIOTR TRZCIŃSKI**



nr archiw. 7601

WROCŁAW 1997

1

Niniejszą książkę opracowano przy współfinansowaniu przez Komitet Badań Naukowych w ramach Projektu Badawczego Zamawianego PBZ - 31- 05 pt. „Sieciowe systemy komunikacyjne integrujące automatyzację wytwarzania”, realizowanego przez PIAP Warszawa oraz Instytut Technologii Maszyn i Automatyzacji Politechniki Wrocławskiej

Kierownik zespołu realizacyjnego: Prof. zw. dr hab. inż. Jan KOCH

Systemy komunikacyjne oraz elektroniczne środki przekazu informacji odgrywają bardzo ważną rolę w wielu dziedzinach działalności człowieka, w tym także w funkcjonowaniu przedsiębiorstw produkcyjnych. W niniejszej książce omówione zostały głównie techniczne aspekty sieci komunikacyjnych.

Przedstawiono Model Odniesienia dla Współdziałania Systemów Otwartych ISO/OSI oraz przykłady otwartych systemów komunikacyjnych. Scharakteryzowane zostały elektryczne łącza komunikacyjne, podstawowe metody transmisji danych oraz protokoły warstwy łącza danych. Szczególną uwagę poświęcono miejscowym (FAN), lokalnym (LAN) i rozległym (WAN) sieciom komputerowym. Rozpatrzono również standardy otwartych systemów komunikacyjnych oraz protokoły warstwy transportowej i aplikacyjnej. Omówiono także architektury sieci komunikacyjnych stosowanych w zautomatyzowanym wytwarzaniu i automatyce przemysłowej oraz zagadnienia integracji heterogenicznych sieci komputerowych. Zamieszczono przykładowe rozwiązania sieci komunikacyjnych CIM.

Wrocław 1997

SPIS TREŚCI

I. Systemy komunikacyjne

1. Wprowadzenie do sieci komunikacyjnych

- 1.1 Ogólny podział sieci
- 1.2 Model odniesienia ISO/OSI
 - 1.2.1 Wstęp
 - 1.2.2 Warstwa fizyczna (transmisji bitów)
 - 1.2.3 Warstwa łącza danych
 - 1.2.3 Warstwa sieci
 - 1.2.4 Warstwa transportowa
 - 1.2.5 Warstwa sesji
 - 1.2.6 Warstwa prezentacji
 - 1.2.7 Warstwa aplikacji
- 1.3 Usługi do komunikacji pomiędzy warstwami i przebieg komunikacji
- 1.4 Przebieg komunikacji w modelu OSI
- 1.5 Przykłady otwartych systemów komunikacyjnych

2. Elektryczne łącza komunikacyjne

- 2.1 Wstęp
- 2.2 Media transmisyjne
 - 2.2.1 Otwarta linia dwuprzewodowa
 - 2.2.2 Skrętka
 - 2.2.3 Kabel koncentryczny
 - 2.2.4 Światłowód
- 2.3 Sygnały transmisji danych
 - 2.3.1 Standard RS-232C/V.24
 - 2.3.2 Pętla prądowa 20 mA
 - 2.3.3 Standard RS-423A
 - 2.3.4 Standard RS-422A/V.11
 - 2.3.5 RS-485
 - 2.3.5 Transmisja kablem koncentrycznym
 - 2.3.7 Transmisja w paśmie podstawowym
 - 2.3.8 Transmisja szerokopasmowa
 - 2.3.9 Światłowód
 - 2.3.10 IrDA

3. Podstawowe metody transmisji danych

- 3.1 Transmisja równoległa
- 3.2 Szeregowa transmisja synchroniczna
- 3.3 Protokół HDLC
- 3.4 Szeregowa transmisja asynchroniczna - UART
- 3.5 Ochrona danych
 - 3.5.1 Rodzaje błędów
 - 3.5.2 Stopa błędów bitów
 - 3.5.3 Powtórzenie transmisji
 - 3.5.4 Resztkowa stopa błędów
 - 3.5.5 Odległość Hamming'a
 - 3.5.6 Efektywność transmisji telegramów
 - 3.5.7 Wybrane strategie rozpoznawania błędów
 - 3.5.7.1 Bit parzystości
 - 3.5.7.2 Zabezpieczenie blokowe
 - 3.5.7.3 CRC
 - 3.5.8 Klasy integralności danych
 - 3.5.9 Formaty telegramów
 - 3.5.9.1 Telegram z bitem parzystości
 - 3.5.9.2 Telegram z CRC
- 3.6 Binarne zobrazowanie informacji
 - 3.7.1 NRZ, RZ
 - 3.7.2 Kodowanie bipolarne i kodowanie HDBn

- 3.7.3 NRZI
- 3.7.4 Kodowanie Manchester II
- 3.7.5 FSK, ASK, PSK

3.7 Interfejs IEC

4. Protokoły warstwy łącza danych

4.1 Topologia sieci

- 4.1.1 Wstęp
- 4.1.2 Połączenia dwu - punktowe
- 4.1.3 Połączenia dwu-punktowe z multiplekserem
- 4.1.4 Struktura magistralowa (*bus*)
- 4.1.5 Struktura drzewiasta
- 4.1.6 Struktura pierścieniowa
- 4.1.7 Struktura gwiazdy
- 4.1.8 Topologia sieci lokalnych LAN i WAN

4.2 Metody dostępu do medium transmisyjnego

- 4.2.1 Wstęp
- 4.2.2 Metoda dostępu CSMA/CD
- 4.2.3 Metoda dostępu CSMA/CA
- 4.2.4 Metoda master/slave
- 4.2.5 Zasada *token'a* (żetonu)
- 4.2.6 Metoda dostępu *Token-Passing*

4.3 Formaty ramek transmisyjnych

- 4.3.1 Protokół *Kermit*
- 4.3.2 LLC
- 4.3.3 Ramki PROFIBUS'a
- 4.3.4 Sposób przesyłania informacji w systemie *InterBus-S*

II. Sieci typu *Fieldbus* oraz sieci komputerowe

5. Sieci miejscowe typu Fieldbus

5.1 Standard PROFIBUS

- 5.1.1 Wstęp
- 5.1.2 Rodzina sieci PROFIBUS
- 5.1.3 Charakterystyka ogólna
 - 5.1.3.1 Architektura protokołu
 - 5.1.3.2 Technologia transmisji
 - 5.1.3.3 RS485 - PROFIBUS-DP/FMS
 - 5.1.3.4 IEC 1158-2 PROFIBUS-PA
 - 5.1.3.5 Światłowód
 - 5.1.3.6 Dostęp do magistrali
- 5.1.4 PROFIBUS-DP
 - 5.1.4.1 Podstawowe funkcje PROFIBUS'a-DP
 - 5.1.4.1.1 Charakterystyka ogólna
 - 5.1.4.1.2 Typy urządzeń oraz konfiguracja systemu
 - 5.1.4.1.3 Zachowanie systemu
 - 5.1.4.2 Funkcje rozszerzonego PROFIBUS-DP
 - 5.1.4.3 Pliki z danymi o urządzeniu (GSD)
 - 5.1.4.4 Profile PROFIBUS-DP
- 5.1.5 PROFIBUS-PA
 - 5.1.5.1 Protokół transmisyjny PROFIBUS-PA
 - 5.1.5.2 Profile urządzeń PROFIBUS-PA

5.1.6 PROFIBUS-FMS

5.2 Sieć LonWorks

5.3 System InterBus-S

- 5.3.1 Wprowadzenie
- 5.3.2 Topologia systemu InterBus-S
- 5.3.3 Centralny sterownik systemu IBS
- 5.3.4 Magistrala odległa (*Remote Bus*)
- 5.3.5 Segment magistrali odległej
- 5.3.6 Instalacyjna magistrala odległa (*Installation Remote Bus*)
- 5.3.7 Magistrala lokalna (*Local Bus*)

- 5.3.8 Magistrala lokalna typu ST (moduły typu *Compact*)
- 5.3.9 Magistrala lokalna typu SL (moduły typu *Slim Line*)
- 5.3.10 Okablowanie sieci InterBus-S
 - 5.3.10.1 Kabel miedziany - skrętka ekranowana
 - 5.3.10.2 Kable światłowodowe
- 5.3.11 InterBus-Loop - technika łączenia dyskretnych oraz analogowych urządzeń peryferyjnych
- 5.3.12 Podstawowa specyfikacja systemu InterBus-S
- 5.3.13 Format protokołu komunikacyjnego
- 5.3.14 Tryby pracy systemu IBS
- 5.3.15 Sterowanie siecią InterBus-S

5.4 Sieć CAN

- 5.4.1 Wstęp
- 5.4.2 Warstwa fizyczna
- 5.4.3 CAN High-Speed (ISO-DIS 11898)
- 5.4.4 *CAN Low-Speed* (ISO-DIS 11519)
- 5.4.5 Protokół komunikacyjny
- 5.4.6 Typy ramek transmisyjnych
 - 5.4.6.1 Ramka danych (ang. *Data Frame*)
 - 5.4.6.2 Ramka zdalna (ang. *Remote Frame*)
 - 5.4.6.3 Ramka błędów (ang. *Error Frame*)
 - 5.4.6.4 *Overload Frame*
- 5.4.7 Organizacja dostępu do magistrali
- 5.4.8 Warstwa zastosowań
 - 5.4.8.1 CAL

5.5 Standard ASI

- 5.5.1 Wprowadzenie
- 5.5.2 Modułowa koncepcja systemu
- 5.5.3 Otwartość systemu
- 5.5.4 System transmisji ASI
 - 5.5.4.1 Okablowanie ASI
 - 5.5.4.2 Topologia systemu ASI
 - 5.5.4.3 Metoda modulacji
 - 5.5.4.4 Zasilanie energii i filtr separujący
 - 5.5.4.5 Metody dostępu do magistrali
 - 5.5.4.6 Budowa telegramów
 - 5.5.4.7 Rozkazy
- 5.5.5 Wymiana danych (*data exchange*)
 - 5.5.5.1 Zapis parametrów (*write parameters*)
 - 5.5.5.2 Przydzielenie adresu (*assign address*)
 - 5.5.5.3 Reset (*reset ASI slave*)
 - 5.5.5.4 Kasowanie adresu operacyjnego (*delete operating address*)
 - 5.5.5.5 Odczytanie kodu identyfikacyjnego (*read ID code*)
 - 5.5.5.6 Odczytanie konfiguracji we/wy (*read I/O configuration*)
 - 5.5.5.7 Odczytanie statusu (*read status*)
 - 5.5.5.8 Odczytanie i kasowanie statusu (*read and reset status*)
- 5.5.6 Działanie systemu w przypadku zakłóceń i błędów
 - 5.5.6.1 ASI - *slave*
 - 5.5.6.1.1 Opis schematu blokowego
 - 5.5.6.1.2 Bezpieczeństwo danych i zachowanie systemu w przypadku wystąpienia błędów
 - 5.5.6.2 ASI - *master*

6. Standardy sieci lokalnych LAN

- 6.1 Wstęp
- 6.2 Sieć typu Ethernet (IEEE 802.3)
- 6.3 Sieć typu *Token Bus* (IEEE 802.4)
- 6.4 Sieć typu *Token Ring* (IEEE 802.5)

7. Szybkie sieci lokalne - FASTLAN

- 7.1 Wstęp

- 7.2 Szybki Ethernet 100Base-T
- 7.3 Standard sieci 100VG-AnyLAN
- 7.4 System FDDI
- 7.5 Standard IEEE 802.6 (DQDB)

8. Rozległa sieć komputerowa - WAN

- 8.1 Wstęp
- 8.2 Sieć cyfrowa z integracją usług - ISDN
- 8.3 Sieć pakietowa X.25
- 8.4 Sieć pakietowa FR
- 8.5 Sieć ATM

III. Systemy otwarte

9. Standardy systemów otwartych

10. Protokoły warstwy transportowej

- 10.1 Protokół TCP/IP
- 10.2 Stos protokołowy TCP/IP
- 10.3 Warstwa sieciowa
- 10.4 Warstwa transportowa
- 10.5 Protokoły wspomagające komunikację
- 10.6 Aplikacje
- 10.7 Ruch w czasie rzeczywistym
- 10.8 Protokoły warstwy transportowej systemu MAP 3.0

11. Protokoły warstwy aplikacyjnej systemów typu *Fieldbus* - FMS i PMS

- 11.1 PROFIBUS-FMS
- 11.2 Warstwa aplikacyjna PROFIBUS-FMS
- 11.3 Model komunikacyjny PROFIBUS-FMS
- 11.4 Elementy komunikacyjne warstwy siódmej
- 11.5 Lista relacji komunikacyjnych
 - 11.5.1 Adresy i porty komunikacyjne
 - 11.5.2. Typy połączeń
 - 11.5.3 Atrybuty połączeń
 - 11.5.4. Kodowanie wspieranych usług
- 11.6. Słownik obiektów
 - 11.6.1. Typy zmiennych
 - 11.6.2 Mechanizm dostępu do obiektów
- 11.7 Usługi warstwy siódmej
- 11.8 Protokół PMS systemu InterBus-S
 - 11.8.1 Interfejs warstwy aplikacyjnej (ang. *Application Layer Interface ALI*)
 - 11.8.2. Model maszyny wirtualnej
 - 11.8.3 *Peripherals Message Specification* (PMS)

12. Protokoły warstwy aplikacyjnej (FTP, Telnet)

- 12.1 FTP
- 12.2 Protokół TELNET

13. Protokoły warstwy aplikacyjnej wg specyfikacji MAP/TOP

- 13.1 Specyfikacja komunikatów przemysłowych MMS
- 13.2 Normy stowarzyszone MMS - *Companion Standards* do MMS
- 13.3 Protokół transferu, dostępu i zarządzania zbiorami - FTAM
 - 13.3.1 Wprowadzenie
 - 13.3.2 Wirtualny system plików (VFS - *Virtual Filestore*)
 - 13.3.3 Reżimy (ang. *Regimes*)
 - 13.5 Jednostki aplikacji AE (ang. *Application Entities*)
 - 13.6 Wirtualna struktura pliku FTAM
 - 13.7 Typy dokumentów w systemie FTAM

IV. Architektury sieci w CIM

14. Architektury sieci w zautomatyzowanym wytwarzaniu i automatyce przemysłowej

14.1 Wstęp

14.2 Architektura sieci komunikacyjnej w zautomatyzowanym wytwarzaniu

15. Integracja heterogenicznych systemów komunikacyjnych

16. Przykłady pilotowych instalacji CIM

I. Systemy komunikacyjne

1. Wprowadzenie do sieci komunikacyjnych

1.1 Ogólny podział sieci

Podstawowym celem budowy sieci komunikacyjnych jest tworzenie połączeń wielu elementów sprzętowych i programowych w jeden funkcjonujący system. Jednakże realizacja tego celu wiąże się z różnymi problemami wynikającymi z istnienia wielu standardów i rozwiązań firmowych poszczególnych składników sieci.

Sieć komputerowa jest systemem komunikacyjnym przeznaczonym do przesyłania danych pomiędzy dwoma lub wieloma komputerami i urządzeniami peryferyjnymi. Generalnie wśród wielkiej różnorodności sieci teleinformatycznych przyjął się podział na:

- sieć miejscową FAN (ang. *Field Area Network*) - jest to sieć komputerowa łącząca sterowniki maszyn i urządzeń oraz komputery w obszarze procesu,
- sieć lokalną LAN (ang. *Local Area Network*) - jest to sieć komputerowa łącząca grupę użytkowników pracujących na stosunkowo niewielkim obszarze, np. grupa robocza, oddział firmy lub budynek,
- sieć terytorialną (ang. *Campus Network*) - jest to sieć obejmująca swoim zasięgiem kilka budynków znajdujących się na terenie uczelni lub przedsiębiorstwa,
- miejską sieć komputerową MAN (ang. *Metropolitan Area Network*) - jest to sieć obejmująca swoim zasięgiem miasto, przy czym MAN jest siecią szkieletową (ang. *backbone network*) zarządzaną przez organ lokalny lub stanowy (w USA),
- sieć rozległą WAN (ang. *Wide Area Network*) - tego rodzaju sieci obejmują swoim zasięgiem miasta, kraje oraz kontynenty i wykorzystują istniejące rozwiązania telekomunikacyjne, w tym także mikrofalowe i satelitarne,
- sieć korporacyjną (ang. *Enterprise Network*) - łączy ona systemy komputerowe wewnątrz organizacji jaką może być przedsiębiorstwo produkcyjne. Sieć ta łączy ze sobą wiele systemów operacyjnych, protokołów komunikacyjnych i może być siecią typu LAN, MAN lub WAN.

Burzliwy rozwój sieci komputerowych datuje się od lat 70-tych. Wówczas dziesiątki uniwersytetów, ośrodków badawczych i firm brało udział w projekcie DARPA agencji Departamentu Obrony USA oraz w szeregu innych programów. Celem prac realizowanych w tych projektach było opracowanie rozwiązań umożliwiających łączenie komputerów w sieci i wymianę informacji pomiędzy nimi. W efekcie tych prac powstały takie sieci komputerowe o zasięgu krajowym jak np.:

- ARPA (USA, 1969),
- CYCLADES (Francja, 1974)

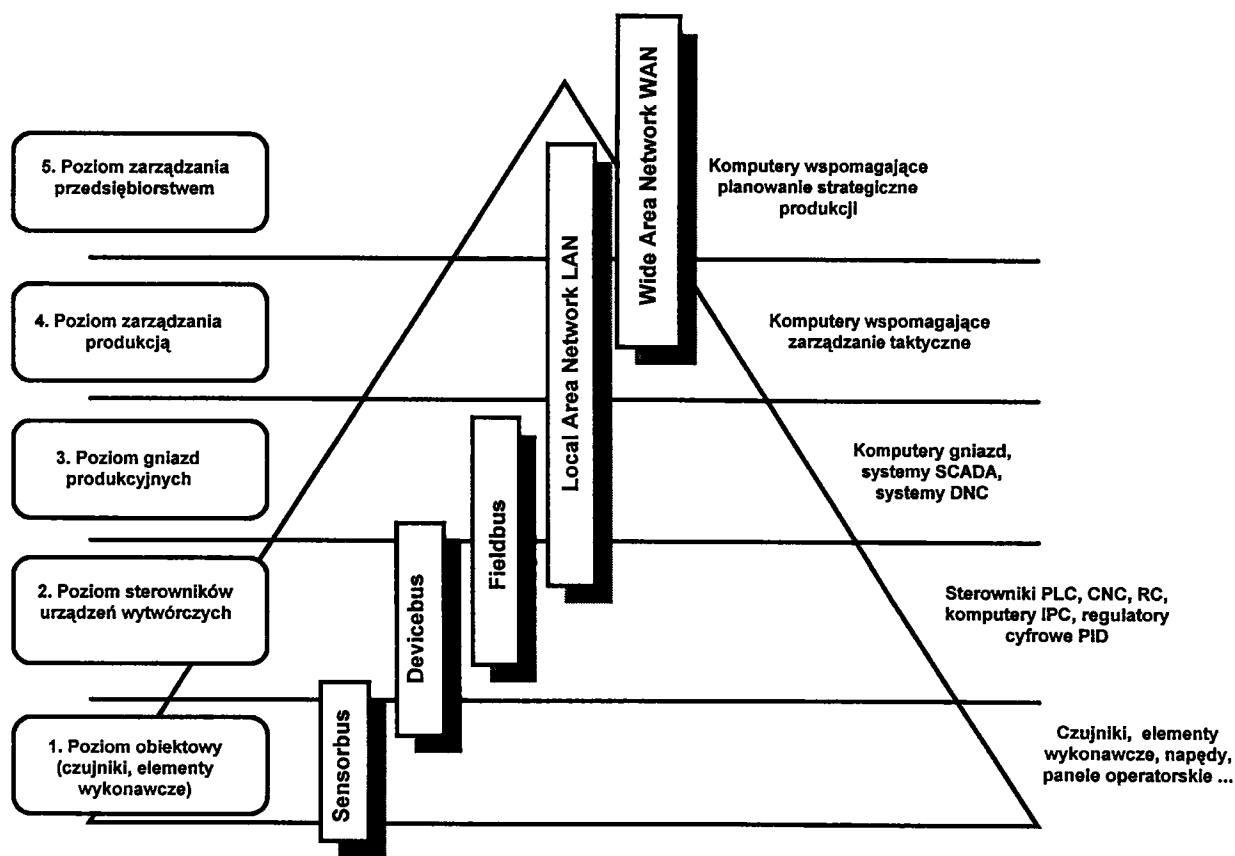
oraz modele sieciowe opracowane przez dominujące na rynku firmy komputerowe, jak np.:

- SNA (*Systems Network Architecture*) firmy IBM,

- DECnet (lub DNA - *Digital Network Architecture*) firmy DEC,
- XNS (*Xerox Network Standard*) firmy Xerox,
- DSA (*Distributed Systems Architecture*) firmy Honeywell,
- DCA (*Distributed Communications Architecture*) firmy UNIVAC.

Nie są to jednak systemy wzajemnie kompatybilne i wyróżniają się one specyficznymi metodami przesyłania danych, formatami informacji i sposobem dekompozycji systemu sieciowego. Wymiana informacji pomiędzy komunikującymi się ze sobą systemami komputerowymi wymaga precyzyjnego zdefiniowania procedur wymiany, formatowania i kodowania informacji.

Istotną rolę systemy komunikacyjne odgrywają w automatyzacji procesów produkcyjnych. I tak np. obszary zastosowania systemów komunikacyjnych w nowoczesnym przedsiębiorstwie przemysłowym obejmują wszystkie poziomy od zarządzania przedsiębiorstwem, poprzez poziom zarządzania produkcją, gniazd produkcyjnych, sterowników maszyn i urządzeń wytwórczych aż do poziomu czujników i elementów wykonawczych (rys.1.1).



Rys.1.1 Obszary zastosowań systemów komunikacyjnych w wytwarzaniu

Hierarchiczny model struktury przedsiębiorstwa produkcyjnego z punktu widzenia systemów komunikacyjnych, zawiera następujące istotne elementy:

1. Komunikację na poziomie zarządzania przedsiębiorstwem oraz przygotowania produkcji (poziom 4, 5 na rys.1.1), w której dominują usługi transferu dużych ilości danych o złożonych

strukturach, przy czym przepustowość systemu komunikacji jest tutaj ważniejsza niż praca w czasie rzeczywistym. Przykładem tego rodzaju komunikacji jest transmisja danych z systemu CAD do obszaru planowania procesu wytwarzania CAP, lub też przesyłanie zlecenia wytwórczego z komputera zarządzania do komputera gniazda produkcyjnego. W obszarze tym wykorzystuje się lokalne LAN (ang. *Local Area Network*) i rozległe sieci komputerowe WAN (ang. *Wide Area Network*).

2. Komunikację na poziomie gniazd produkcyjnych (poziom 3), w której wykorzystuje się rozwiązania dla sieci lokalnych LAN oraz powstałe w latach '80-tych liczne rozwiązania sieci miejscowych FAN, określanych także jako *Fieldbus*. Systemy te mają za zadanie transmisję danych w obrębie gniazda wytwórczego, np. z nadrzędnego systemu sterowania i monitorowania procesu do sterownika swobodnie programowalnego PLC (ang. *Programmable Logic Controller*).
3. Komunikację na poziomie sterowników (poziom 1, 2) urządzeń wytwórczych oraz czujników i elementów wykonawczych, w której wykorzystuje się cykliczną transmisją małej ilości danych obiektowych (dane pomiarowe, nastawy regulatorów itp.) w czasie rzeczywistym. Stosuje się tu magistrale przemysłowe sterowników (ang. *Devicebus*) oraz czujników (ang. *Sensorbus*).

Sieć komputerowa na górnych poziomach struktury zakładu spełnia funkcję szkieletu (ang. *backbone*), co oznacza, że musi ona łączyć różne działy przedsiębiorstwa i umożliwiać dostęp do sieci rozległych WAN po to, aby można było wymieniać dane także z filiami przedsiębiorstwa lub klientami i dostawcami.

1.2 Model odniesienia ISO/OSI

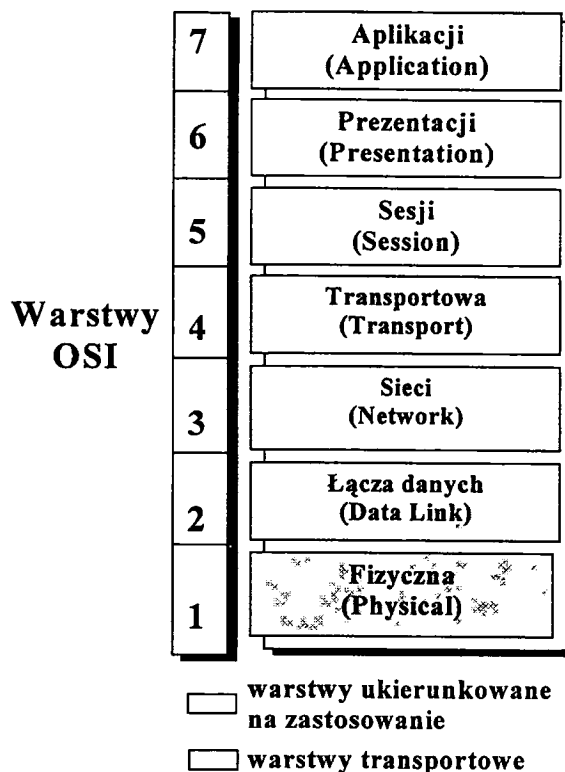
1.2.1 Wstęp

Komunikacja pomiędzy komputerami w celu wymiany danych wymaga uzgodnienia sposobu jej przebiegu. Jeżeli jednak rozpatrzy się różne możliwości komunikacji to okaże się, że ich przebiegi są często podobne. Wzrost znaczenia komunikacji w gospodarce światowej spowodował, że w roku 1978 międzynarodowa organizacja standaryzacyjna ISO (*International Standards Organisation*) zaproponowała Model Odniesienia Dla Współdziałania Systemów Otwartych (OSI RM - ang. *Open Systems Interconnections Reference Model*). Jest to model teoretyczny o dużej ogólności, oparty na warstwowej strukturze systemu komputerowego traktowanego jako system otwarty, tzn. jest on zdolny do wymiany informacji z innymi systemami o takiej samej strukturze. Podstawą dla Modelu Odniesienia była architektura SNA firmy IBM z 1974 roku. Dalsze prace doprowadziły w 1983 r. do powstania normy ISO 7498 (później przyjętej również przez *Comité Consultatif International Télégraphique et Téléphonique* CCITT jako X.200) modelu odniesienia (referencyjnego) dla komunikacji komputerowej pt. „*Basic Reference Model for Open Systems Interconnection (OSI)*”. Model ten opisuje komunikację w systemach otwartych, tzn. takich systemach, które są otwarte dla komunikacji.

W modelu tym określone zostały również zasady komunikowania się warstw pomiędzy sobą, formaty informacji oraz repertuar usług każdej z warstw. Funkcjonowanie poszczególnych warstw opisują protokoły.

Model odniesienia OSI dzieli abstrakcyjnie komunikację na siedem poziomów (warstw) z określoną funkcjonalnością i dlatego też model ten jest określany również jako model warstwowy OSI (rys.1.2). Każdej z warstw jest przydzielona realizacja określonych, wyraźnie zdefiniowanych zadań cząstkowych komunikacji.

Dlatego też w każdym z uczestniczących w komunikacji partnerów jest zawartych siedem warstw. Warstwy te komunikują się ze sobą za pomocą dokładnie zdefiniowanych interfejsów, co ułatwia wymianę poszczególnych warstw bez ingerencji w funkcjonalność całego systemu. W tych interfejsach każda warstwa przygotowuje usługi, które mogą być wykorzystywane przez warstwy sąsiednie. Warstwowy model OSI opisuje proces komunikujących się partnerów na poziomie abstrakcyjnym. Nie są w nim jednak zawarte żadne dane dotyczące tego, jak ostatecznie poszczególne warstwy powinny być zaimplementowane. Tym samym zapewniona jest baza dla systemu otwartego, który przez zdefiniowanie zawartości poszczególnych warstw i określenie interfejsów może być wykorzystany również wtedy, gdy cały system komunikacyjny jest zestawiony ze składników pochodzących od wielu producentów. Podział na warstwy służy abstrakcyjności zachodzących procesów komunikacyjnych.



Rys.1.2 Warstwy modelu odniesienia OSI

Opis warstw modelu odniesienia OSI rozpoczyna się od warstwy 1 (najniższej) i kończy na warstwie 7 (najwyższej). W obrębie tego modelu przewidziany jest podział na dwie zasadnicze części. I tak część pierwsza obejmuje warstwy 1 do 4, które są odpowiedzialne za transmisję danych pomiędzy urządzeniami końcowymi. Warstwy te zawierają funkcje wzgl. oferują usługi, które są wykorzystywane do niezawodnego transportu informacji pomiędzy procesami użytkownika. Dlatego też są one łącznie określane jako warstwy transportujące. Natomiast warstwy od 5 do 7 podczas transmisji danych koordynują współdziałanie z programami użytkownika i systemami operacyjnymi używanych komputerów. Ze względu na to, że dają one do dyspozycji procesom użytkownika usługi i funkcje, to są one określane jako warstwy zorientowane na użytkownika.

1.2.2 Warstwa fizyczna (transmisji bitów)

Warstwa 1 jest warstwą fizyczną (ang. *Physical Layer*). Określa ona w jaki sposób fizycznie ma się odbywać transmisja danych, tzn. elektryczne i mechaniczne właściwości transmisji. W warstwie fizycznej są zatem określone takie szczegóły dotyczące transmisji informacji pomiędzy systemami otwartymi jak np. przebiegi modulacji i kodowania lub elektryczne właściwości fizycznego medium transmisyjnego. Dzięki funkcjom warstwy fizycznej w systemie komunikacyjnym możliwa jest transmisja strumienia bitów. Obejmuje to rodzaj kodowania (*Immediate Return to Zero, No Return to Zero Invert, No Return to Zero, Manchester, FSK* itp.), poziom napięcia dla transmisji, czas

trwania poszczególnych bitów, wybór kabla transmisyjnego i końcowego sprzęgu systemowego (złącza) oraz przyporządkowanie wtyków (obsada pinów) dla transmisji strumienia bitów.

Warstwa fizyczna, tak jak i każda inna warstwa, jest w systemie komunikacyjnym wymienna bez oddziaływania na inne warstwy. Komunikacja w całym systemie jest niezależna od odrębności poszczególnych warstw. Warstwa 1 może np. zajmować się odcinkiem światłowodowym podobnie jak i elektrycznym medium transmisyjnym wg norm RS232, RS422 lub RS485 albo też dowolnym innym. Warstwa 1 nie jest zatem sama medium fizycznym, lecz tego rodzaju częścią definicji transmisji, która określa odcinek fizyczny.

1.2.3 Warstwa łączy danych

Warstwa 2 jest warstwą łączy danych na poziomie kabla (*Data Link Layer*). Jej zadaniem jest niezawodny transport danych od jednej stacji do drugiej. Służy ona zatem do ochrony danych podczas fizycznej transmisji. Dane są tak pakowane, żeby można było rozpoznać błędy transmisji uczestniczących w niej stacji. W tym celu transmitowane dane są w taki sposób dzielone na ramki (*data frames*), że w każdej ramce jest zawarta maksymalna ilość bajtów. Zwykle wielkości ramek zawierają się w zakresie setek bajtów. Ramki te, oprócz „surowych” danych, zawierają dodatkowe informacje dla transmisji, które do istniejących danych ze swojej strony wstawia warstwa łączy danych. Te dodatkowe informacje zawierają np. sumę kontrolną oraz informacje początkowe i końcowe dla ramek. Ponadto dodatkowe informacje mogą służyć do potwierdzania telegramów, które zostały już przesłane do partnera komunikacyjnego. Za pomocą stosowanych tutaj mechanizmów powinno się np. stwierdzić, czy ramki zostały bezbłędnie przesłane lub też czy na drodze transmisji nastąpiła utrata ramek. Jeżeli już utracone ramki zostały powtórnie przesłane, to warstwa łączy danych troszczy się o to, aby przy odbiorze nie zostały one zdublowane. Warstwa łączy danych nie ma żadnej wiedzy o zawartości informacji.

Warstwa łączy danych daje następnemu wyższemu poziomowi (3) do dyspozycji kanał logiczny, który funkcjonuje bez błędów transmisji. Ponadto wyrównuje ona na poziomie 2 różne szybkości transmisji danych (odczyt/zapis) dla nadajnika i odbiornika i tym samym kontroluje przepływ danych pomiędzy uczestniczącymi stacjami oraz pozwala na uniknięcie „nadgania” pewnej stacji w tym przypadku, gdy jedna stacja szybciej nadaje aniżeli odbierające stacje mogą dalej przetwarzać dane.

Warstwa łączy danych jest opisana w dwóch częściach normy IEEE 802, przy czym *Logical Link Control* (LLC) stanowi usługi dla komunikacji z poziomem 3, a *Medium Access Control* (MAC) jest wymagana do włączenia warstwy 1.

1.2.3 Warstwa sieci

Podczas gdy w przypadku warstwy 2 była rozpatrywana komunikacja pomiędzy dwoma stacjami, to w trzeciej warstwie sieci (*Network Layer*) cała sieć jest traktowana jako logiczna jednostka, która jest w całości przetwarzana. Zadaniem warstwy sieci są:

- transport danych ze źródła do miejsca docelowego, ewentualnie przez stacje pośrednie,
- przygotowanie interfejsów pomiędzy systemami końcowymi,
- *Routing*, tzn. określenie dróg danych w sieci przez sterowanie drogami, co może się odbywać statycznie lub dynamicznie oraz
- pakowanie i rozpakowywanie pakietów, które mogą być przetwarzane przez warstwę 2.

Warstwa sieci troszczy się o to, aby uniknąć zatorów w podrzędnych sieciach, tzn. musi być przez nią kontrolowana ilość znajdujących się w sieci pakietów danych.

Zasadniczo rozróżnia się usługi połączeniowe i bezpołączeniowe. Jeżeli usługa jest ukierunkowana na połączenie, to daje ona użytkownikowi do dyspozycji wirtualny kanał (*Virtual Circuit Service*). Związany z nim przebieg komunikacji składa się z:

- budowy połączenia,
- wymiany danych,
- likwidowania połączenia.

Takie formy komunikacji są porównywalne z rozmową telefoniczną, w której budowa połączenia realizowana jest po wyborze numeru abonenta, wymiana danych odbywa się za pomocą rozmowy, a odwieszenie słuchawki powoduje rozłączenie połączenia.

Usługi bezpołączeniowe (*Datagram Service*) nie stanowią żadnego połączenia pomiędzy partnerami komunikacyjnymi. Pakiety danych przewidziane do transmisji są dostarczane do sieci z pełnymi adresami docelowymi i tam są dalej transportowane. Zatem są one podobne do przesyłania listów, w którym pakiety danych (listy) z umieszczonymi pełnymi adresami odbiorcy są przekazywane do przewidzianych w tym celu w sieci punktów (skrzynki pocztowe) i bez oddziaływania na drogę transportu są dostarczane użytkownikowi przez serwis sieci pod adres docelowy.

1.2.4 Warstwa transportowa

Poziomem 4 w modelu odniesienia OSI jest warstwa transportowa (*Transport Layer*). Opisuje ona komunikację pomiędzy procesami takimi jak np. programy w komputerze *host A* i komputerze *host B*, które wymieniają ze sobą dane. Warstwa transportowa ma następujące zadania cząstkowe:

- nadawanie nazwy komputerowi *host*,
- adresowanie uczestników,
- budowę i kasowanie połączenia (odnośnie transportu),
- obsługę błędów i kontrolę przepływów,
- multiplekowanie różnych strumieni danych dla jednego kanału,

- synchronizację *host*'ów,
- ponowne utworzenie połączenia w przypadku błędu w podrzędnej sieci,
- *Internetworking*.

Warstwa transportowa rozdziela dane wyższych poziomów na jednostki przeznaczone do transportu. W sieciach zorientowanych na połączenia warstwa ta buduje połączenie z partnerem komunikacyjnym. Zależnie od żądanej własności dla każdego połączenia transportowego zestawiane jest jedno własne połączenie sieciowe, wiele połączeń sieciowych (przy dużej ilości danych) lub dla wielu połączeń transportowych jedno jedyne połączenie sieciowe (połączenie zbiorcze). Połączenia zbiorcze są w praktyce wykorzystywane najczęściej ze względu na koszty. Istnienie takiego połączenia zbiorczego dla wyższych warstw jest transparentne. Dla wyższych poziomów pozostają tzw. *Service Access Points* z nazwą i adresem (SAP). Zależnie od tego jakie usługi warstwy 4 zostaną wzięte pod uwagę, istnieją różnorodne klasy usług (*Service-Klassen*), które każdorazowo zawierają wymienione wyżej zadania.

Dla zadania *Internetworking* w komputerze-gateway'u (*host* A w sieci 1 komunikuje się z *host*'em B w sieci 2 przez ten komputer-gateway) zadaniem warstwy transportowej komputera-gateway'a jest przekształcanie różnorodnych protokołów.

Przy budowie połączenia określany jest rodzaj transportu. Istnieje możliwość połączenia punkt-punkt (*peer to peer*), w którym dane są transmitowane w kolejności swojego występowania, podobnie jak przesyłanie pakietów, w którym dane są przekazywane do sieci a kolejność ich występowania u partnera nie jest określona. Rodzaje transmisji *Broadcast* wzgl. *Multicast* służą do tego, aby równocześnie tymi samymi informacjami zasilać wszystkie wzgl. określoną ilość przyłączonych stacji. Poziomy 1-4 tworzą wspólnie system transportowy w modelu odniesienia OSI.

1.2.5 Warstwa sesji

Poziom 5 w modelu odniesienia OSI jest określany jako warstwa sesji (*Session Layer*). Pod pojęciem sesji rozumie się wykorzystanie systemu transportowego, tzn. bezbłędnego kanału logicznego, który daje do dyspozycji warstwa transportowa. Do tego celu są przygotowane usługi do budowy i zrywania sesji, dzięki czemu do systemu transportowego może mieć dostęp jeden lub wiele procesów. Warstwa sesji jest normalnie połączona z systemem operacyjnym komputera. Synchronizuje ona, jeżeli jest to konieczne, komunikujące się procesy, aby umożliwić bezbłędny przepływ danych. Zależnie od tego jakie działania powinny być wykonywane w wyższych warstwach, można zaimplementować różnorodny zakres funkcyjny warstwy sesji. W modelu OSI występują następujące grupy funkcji:

- BCS *Basic Combined Subset* dla sterowania połączeniami i transmisją danych,
- BAS *Basic Activity Subset* dla zarządzania działaniami oraz
- BSS *Basic Synchronised Subset* dla synchronizacji.

Warstwa sesji może tak samo zarządzać symetrycznymi konstelacjami partnerów jak i połączeniami niesymetrycznymi (architektura *Client-Server*). Wywołanie procedury na pewnym zewnętrznym komputerze (*Remote Procedure Call*) jest sterowane przez warstwę sesji.

1.2.6 Warstwa prezentacji

Warstwa prezentacji (*Presentation Layer*), warstwa 6, daje do dyspozycji usługi do prezentacji transmitowanych danych. Obejmuje to funkcje:

- dla używanego zestawu znaków,
- dla kodowania transmitowanych danych oraz
- prezentacji danych na ekranie monitora lub drukarce.

Procesy podczas komunikacji wymieniają ze sobą dane, które podlegają określonej syntaktyce (składni) i mają określone znaczenie (semantykę). Podczas tej wymiany danych musi być uzgodnione, jak powinno wyglądać zobrazowanie informacji podczas transmisji danych i jaki rodzaj zobrazowania wykorzystują oba komunikujące się procesy. Transmitowane dane, np. dla terminali EBCDIC lub ASCII, mogą być dostarczane w różnych kodach albo w różnorodnych formatach zbiorów. Dlatego też zadania warstwy prezentacji zawierają również kodowanie i dekodowanie danych (*Data Encryption*) oraz ochronę niezawodności danych (*Data Security & Privacy*). Również kompresja danych w celu zmniejszenia ilości danych i tym samym oszczędności czasu i kosztów transmisji jest realizowana przez warstwę prezentacji.

1.2.7 Warstwa aplikacji

Najwyższą warstwą modelu odniesienia OSI jest warstwa 7, warstwa aplikacji (*Application Layer*). Obejmuje ona funkcje, za pomocą których użytkownik może mieć dostęp do systemu komunikacyjnego. Użytkownikiem jest tutaj zwykle nie człowiek, ale taki program komputerowy jak np. FTAM (*File Transfer, Access and Management*), tzn. program do transmisji zbiorów i dostępu do zbiorów poza granicami komputera.

Warstwa aplikacji ma zapewniać transparenację miejscową, np. w rozproszonych bankach danych, gdzie logicznie powiązane ze sobą dane fizycznie są złożone w różnych komputerach, które mogą być umieszczone w różnych geograficznie miejscach. Przy odpytywaniu przez sieć nie musi się zaznaczać fizycznych własności banku danych. Warstwa aplikacji (zastosowań) jest tą częścią systemu komunikacyjnego, która komunikuje się bezpośrednio z procesem użytkownika przez określony interfejs. Zadanie komunikacyjne w tzw. instancji użytkownika jest każdorazowo wykonywane przez jeden lub wiele elementów usług.

1.3 Usługi do komunikacji pomiędzy warstwami i przebieg komunikacji

Każda instancja warstwy OSI oferuje położonym nad nią warstwom swoje usługi. Podczas wymiany danych pomiędzy warstwą N i warstwą N+1, warstwa N dostarcza wymaganych usług,

tn. jest ona *service provider*. Warstwa $N+1$ wykorzystuje te usługi i tym samym jest *service user*. Usługi są dostępne w zaznaczonych punktach dostępu, tzw. *service access points* (SAP). Każdy SAP ma jeden jednoznaczny adres.

Rozróżnia się usługi niezależne od połączenia i usługi ukierunkowane na połączenie (p. wyżej). Dla realizacji tych zadań usługowych są do dyspozycji tzw. prymitywy usług. Są to:

- żądanie (*request*),
- wskazanie (*indication*),
- odpowiedź (*response*) oraz
- potwierdzenie (*confirmation*).

Dla usług z potwierdzeniami są przewidziane wszystkie cztery prymitywy usług, natomiast dla usług bez potwierdzeń tylko żądanie i indykacja.

Typową usługą z potwierdzeniem jest budowa połączenia (*connect*). W notacji OSI podawana jest każdorazowo usługa i za pomocą kropki oddzielony jest każdorazowy prymityw. Przy budowie połączenia pomiędzy dwoma stacjami przez tego partnera, który chciałby otworzyć połączenie, jest najpierw wysyłany *connect.request*. U partnera odbierającego wywołuje on *connect.indication*, tzn. pokazanie, że połączenie powinno być zbudowane. Zapytywany partner przez *connect.response* odpowiada czy chce on podjąć połączenie. Ta odpowiedź u inicjującego połączenie wywołuje potwierdzenie *connect.confirmation*. Dalszymi usługami są transmisja danych (*data*) i zrywanie połączenia (*disconnect*). Mogą być one zresztą wykorzystywane jako usługi z potwierdzeniami lub bez potwierdzeń.

Pomiędzy procesem użytkownika dwóch wymieniających informacje otwartych systemów istnieje ogólnie jedno logiczne połączenie (relacja ukierunkowana na połączenie). To połączenie musi być zbudowane przed rozpoczęciem wymiany informacji, a po jej zakończeniu usunięte. Budowa i kasowanie połączeń komunikacyjnych oraz transmisja danych pomiędzy instancjami użytkownika są regulowane przez protokoły aplikacji. Protokoły te do transmisji informacji pomiędzy dwoma instancjami aplikacji wykorzystują jednostki danych protokołów. Jednostki danych protokołów są telegramami, których znaczenie (semantyka) oraz ich struktura (syntaktyka) jest jednoznacznie określona w każdorazowej normie OSI. Te definicje oraz określenie kiedy i jak są wymieniane jednostki danych protokołów i w jakiej zależności są one względem elementarnych usług jest opisane w definicji protokołu. Zatem protokół jest zbiorem reguł określających wymianę informacji między warstwami (i w ogóle między dowolnymi urządzeniami, jednostkami itp.).

W taki sam sposób, w jakim w procesie użytkownika wykorzystywane są usługi systemu komunikacyjnego, instancja aplikacji wykorzystuje usługi podległej jej hierarchicznie instancji prezentacji. Elementy usług instancji prezentacji wykorzystują ze swojej strony elementy usług instancji sesji i tak dalej. Zatem ogólnie można stwierdzić, że element usług warstwy (N) stawia do dyspozycji swoje usługi elementowi (-om) usług warstwy ($N+1$). On sam wykorzystuje natomiast usługi jednego lub wielu elementów usług warstwy ($N-1$). Wykorzystanie usług jest opisane za

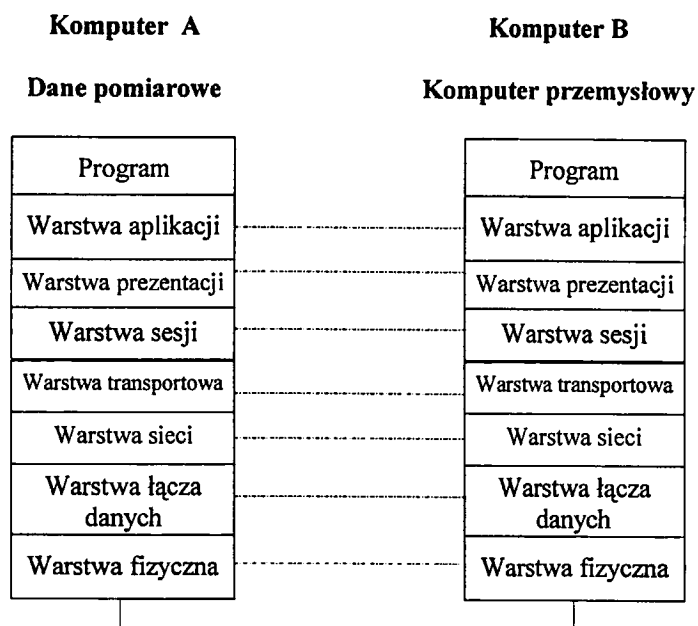
pomocą prymitywów usług. Jednakowe instancje różnych systemów komunikacyjnych komunikują się ze sobą za pomocą jednostek danych protokołów wg reguł, które są określone jako protokoły. Protokoły są zbiorami reguł semantycznych i syntaktycznych, które określają przebiegi elementów usług podczas wymiany informacji za pomocą elementu usług partnera w innym systemie otwartym.

Warstwa łącza danych daje do dyspozycji mechanizmy za pomocą których, z jednej strony w oparciu o niepewną transmisję przez warstwę fizyczną zapewniona jest niezawodna transmisja danych w blokach bitów. Z drugiej strony warstwa łącza danych umożliwia sterowanie przepływem danych, tzn. powoduje dopasowanie „wydajności” nadajnika do odbiornika. Warstwa łącza danych dla sieci lokalnych definiuje natomiast przebiegi sterowania dostępem do medium. Jeżeli za pomocą warstw fizycznej i łącza danych łącznie opisana jest niezawodna komunikacja pomiędzy otwartymi systemami w tym samym segmencie kabla fizycznego medium transmisyjnego, to warstwa łącza danych dzięki swoim funkcjom pozwala na wymianę informacji pomiędzy otwartymi systemami w różnych segmentach za pomocą systemów sprzęgających. Nadbudowana na funkcje warstwy łącza danych warstwa transportowa oferuje funkcje, które umożliwiają utrzymywanie połączeń pomiędzy dowolnymi programami użytkownika w systemach otwartych.

Usługi, które są niezbędne do otwarcia, ukierunkowanego wykonania i zakończenia relacji komunikacyjnej (zwanej sesją) są przyporządkowane warstwie sterującej komunikacją. W warstwie prezentacji są oferowane usługi instancji aplikacji, które pozwalają na określenie rodzaju kodowania w którym jest zobrazowana wymieniana informacja oraz wyszukania wzgl. przeprowadzenia konwersji. Warstwa aplikacji sama służy procesowi użytkownika jako okno do systemu komunikacyjnego, tzn. stawia ona do dyspozycji procesowi użytkownika specyficzne dla zastosowania usługi komunikacyjne.

1.4 Przebieg komunikacji w modelu OSI

Dwa komputery A i B komunikują się w modelu odniesienia OSI w taki sposób, że każdorazowo komunikują się ze sobą te same warstwy w obrębie partnerów komunikacyjnych. Warstwa fizyczna stacji A jest bezpośrednio połączona z warstwą fizyczną stacji B. Warstwy sieciowe obu stacji wymieniają ze sobą dane, przy czym niżej położone warstwy są transparentne dla wymiany danych z warstwy 3 do warstwy 3. Przewidziane są zawsze wszystkie warstwy modelu odniesienia, ale warstwy mogą być również puste. Wówczas podczas komunikacji np. istnieją tylko kody programu dla warstw 1, 2 i 7, natomiast warstwy od 3 do 6 są puste (np. definicja Profibus’a FMS). Warstwy wymieniają ze sobą dane przez określone usługi z potwierdzeniami lub bez potwierdzeń, jak to opisano we wcześniejszym rozdziale. Przebieg komunikacji pokazano na przykładzie (rys.1.3).



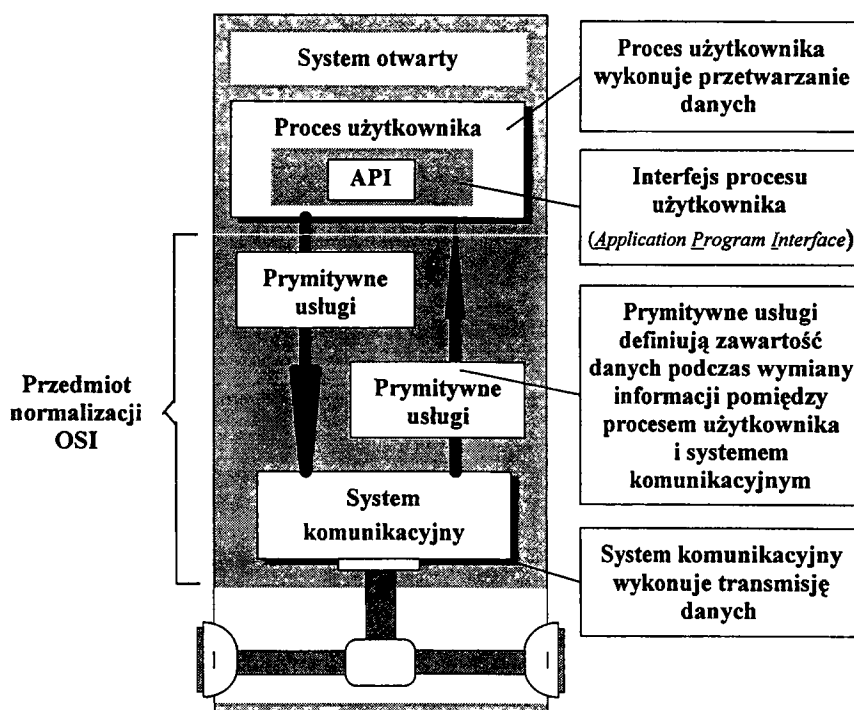
Rys.1.3 Przykład przebiegu komunikacji w modelu OSI

Użytkownik stacji A, w rozpatrywanym przykładzie jest to program do przetwarzania danych pomiarowych, „chce” zapytać o nową daną pomiarową stację zewnętrzną B. Połączenie pomiędzy stacjami A i B zostało już zbudowane. Program użytkownika poleca jako pierwszej warstwie użytkownika (aplikacji) zażądać daną pomiarową od stacji B. Warstwa użytkownika przetwarza tą dyrektywę i przekazuje dalej jako dane do niżej położonych warstw. Warstwa prezentacji przetwarza dane do postaci uzgodnionej dla transmisji i dostarcza je do warstwy sesji, która wstawia informację o tym, z jakiej sesji pochodzi żądanie. Dane te są następnie podawane do kanału transportowego i w warstwie sieci są kierowane na właściwą drogę (*routing*). Warstwa zabezpieczeń wstawia do danych informację zabezpieczającą i przekazuje dalej powiększony na długości telegram do warstwy fizycznej dla transmisji bitów. Przez kabel danych telegram transmisji dociera do stacji B, gdzie jest on odbierany z warstwy fizycznej. Stąd jest on kierowany dalej do warstwy zabezpieczeń. Sprawdza ona poprawność transmisji bitów, usuwa informacje zabezpieczające i przekazuje pozostałe dane do następnej wyższej warstwy. Warstwa użytkownika rozpoznaje z danych telegramu żądanie odczytu wielkości zmierzonej i kieruje dalej do programu dla przetwarzania. Odczytana dana pomiarowa pokonuje w sposób analogiczny do wyżej pisanego drogę powrotną przez wszystkie warstwy, aż warstwa użytkownika w stacji A może „oddać” odczytaną daną pomiarową programowi.

1.5 Przykłady otwartych systemów komunikacyjnych

Chociaż w rzeczywistości nie istnieją implementacje dokładnie i wiernie realizujące architekturę OSI/ISO, to jednak 7-mio warstwowy model systemu otwartego został powszechnie zaakceptowany i różne rozwiązania firmowe do niego się odwołują.

Pod pojęciem systemu jest tutaj rozumiana autonomiczna jednostka. Może się ona składać z jednego lub kilku komputerów, przynależnego do nich oprogramowania, urządzeń peryferyjnych, terminali, operatorów itp. oraz musi być w stanie przeprowadzać przetwarzanie danych i wykonywać transmisję danych. Tym samym systemem może być np. komputer wytwarzania lub obrabiarka sterowana numerycznie. Ta definicja jest w przypadku OSI ograniczona do systemów otwartych, pod którymi rozumie się takie systemy, w których wymiana informacji odbywa się wg norm komunikacyjnych, które zostały opracowane na bazie modelu odniesienia OSI.



Rys.1.4 Struktura systemu otwartego (wg Weck'a)

Otwarty system dzieli się na dwie zasadnicze części (rys.1.4). Te obie główne części są nazywane odpowiednio procesem użytkownika (aplikacji) i systemem komunikacyjnym. Proces użytkownika jest tą częścią systemu otwartego, która prowadzi do określonego zastosowania przetwarzania danych. Właściwe przetwarzanie danych jest natomiast oddzielone od transmisji danych. Wykonywanie transmisji należy do zadań systemu komunikacyjnego. System komunikacyjny umożliwia to, że za pomocą medium transmisji danych można wymieniać informacje z partnerem komunikacyjnym innego otwartego systemu. Zasady wg których odbywa się wymiana informacji są określane jako protokół. Przedmiotem normalizacji OSI są wyłącznie system komunikacyjny i jego interfejsy do procesu użytkownika. Ważne w tej „filozofii” jest to, że system komunikacyjny może być wymieniony na inny bez konieczności zmiany procesu użytkownika.

Model odniesienia OSI wychodzi zatem z założenia, że w tzw. środowisku OSI są połączone ze sobą dwa lub więcej otwartych systemów do spełniania wspólnego zadania przetwarzania danych za pomocą fizycznego medium transmisji danych dla wymiany informacji.

Na rys.1.5 zamieszczono zestawienie podstawowych cech otwartej komunikacji.

<i>Interconnectivity</i>	• możliwość połączenia, urządzenia mogą wymieniać dane, • zabezpieczenie dzięki wygodnym testom, tzn. implementacja protokołu odpowiada przyjętej specyfikacji
<i>Interoperability</i>	• urządzenia mogą współpracować na poziomie użytkownika, • testowanie urządzenia w instalacji odniesienia odnośnie bezbłędnej komunikacji z urządzeniami innych producentów
<i>Interchangeability</i>	• możliwość zamiany przez urządzenia różnych producentów; wymagany jest jednakowy zakres funkcji w urządzeniach

Rys.1.5 Podstawowe cechy otwartej komunikacji

Bezpośrednie zastosowanie siedmiowarstwowego modelu odniesienia ISO/OSI (ang. *International Standard Organisation/Open System Interconnection*) dla sieci lokalnych okazało się niecelowe, dlatego też w ramach IEEE Project 802 przyjęto Model Odniesienia dla Sieci Lokalnej (ang. *Local Network Reference Model*) (rys.1.6). Model LAN wg IEEE 802 obejmuje dwie najniższe warstwy Modelu Odniesienia ISO (z pewnymi funkcjami warstwy sieciowej) [3].

Warstwę łącza danych (*Data Link Layer*) podzielono na dwie podwarstwy:

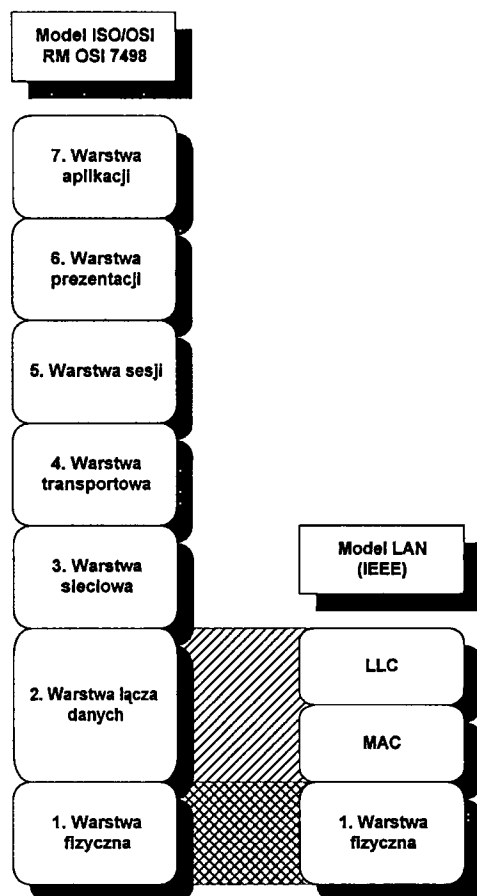
- **MAC** (*Medium Access Control*) - sterowania dostępem do medium oraz
- **LLC** (*Logical Link Control*) - sterowania łączem logicznym.

Podwarstwa MAC definiuje rozwiązania sprzętowe i programowe dla konkretnego typu sieci, natomiast podwarstwa LLC tworzy pomost do wyższych warstw wg modelu OSI. Aktualna struktura komitetu IEEE Project 802 jest przedstawiona na rys.1.7. Poniżej przedstawione są podkomitety opracowujące sieci lokalne, a także sieci miejskie MAN (ang. *Metropolitan Area Network*):

- 802.1 warunki współpracy sieci (ang. *Internetworking*) - określa relacje pomiędzy standardami 802, a modelem ISO/OSI. Komitet 802.1 definiuje na użytek wszystkich standardów 802 adres stacji w sieci lokalnej (adres 48 bitowy). Wszyscy producenci kart sieciowych są zobowiązani do stosowania unikatowych adresów kart sieciowych,
- 802.2 sterowanie łączem logicznym - określa protokół sterowania łączem logicznym LLC (ang. *Logical Link Control*), który zapewnia niezawodny przekaz danych przez łącze komunikacyjne. 802.2 w odniesieniu do modelu ISO/OSI znajduje się w warstwie łącza danych i jest podzielona na dwie podwarstwy: sterowania dostępem do medium MAC (ang. *Media Access Control*) oraz sterowania łączem logicznym (LLC). LLC posługuje się adresami punktów dostępu do usługi

SAP (ang. *Service Access Points*), natomiast podwarstwa MAC - adresami fizycznymi urządzeń sieciowych. LLC jest odpowiedzialna za usługi połączeniowe (ang. *Connection-oriented service*), usługi połączeniowe z potwierdzeniami (ang. *Acknowledged connection-oriented services*) i usługi połączeniowe bez potwierdzenia (ang. *Unacknowledged connection-oriented services*).

- 802.3 (ISO 8802.3) sieci typu CSMA/CD (ang. *Carrier Sense Multiple Access/Collision Detection*). Standard ten określa szybkość transmisji danych 10Mbit/s i przewiduje użycie przewodu koncentrycznego, skrętki i światłowodu. Nowsze rozwiązania pozwalają na szybkość transmisji 100Mbit/s. Standard ten jest szczegółowo opisany w dalszej części książki.
- 802.4 szerokopasmowa sieć typu *Token Bus* - opracowana do wykorzystania w przemyśle wytwórczym. Standard ten wykorzystuje technikę przekazywania znacznika (ang. *Token passing*). *Token* przesyłany jest od stacji do stacji, a stacja mająca *token* jest uprawniona do transmisji danych. Przekazywanie znacznika tworzy pierścień logiczny, który nie musi pokrywać się z pierścieniem fizycznym wyznaczonym przez kolejność stacji.
- 802.5 sieć typu *Token Ring* - spopularyzowana przez firmę IBM. Standard ten definiuje sieć o topologii gwiazdy, ale tworzącą pierścień logiczny. Wykorzystuje się tu technikę przekazywania znacznika (ang. *Token passing*). Poszczególne węzły włączone są do jednostki centralnej (koncentratora). Na protokole 802.5 bazuje również sieć światłowodowa FDDI, ale jej standard został opracowany przez komitet ASC X3T9. Sieć FDDI jest kompatybilna ze standardami 802 poprzez warstwę sterowania łączem określoną w 802.2.
- 802.6 sieci miejskie MAN - standard ten określa szybki protokół obsługujący stacje włączone do podwójnej magistrali światłowodowej. Zaletą tego standardu jest możliwość utrzymania połączenia nawet w przypadku uszkodzenia magistrali. W rozwiązaniu tym wykorzystywana jest metoda dostępu do medium określana jako DQDB (ang. *Distributed Queue Dual Bus*). Standard ten umożliwia także przesyłanie danych, głosu i obrazu na obszarze o zasięgu do 50 km z szybkością do 155Mbit/s. Przekaz danych następuje w formie komórek o stałej długości 53 bajtów, a zatem sieć ta jest kompatybilna z protokołem B-ISDN oraz ATM.



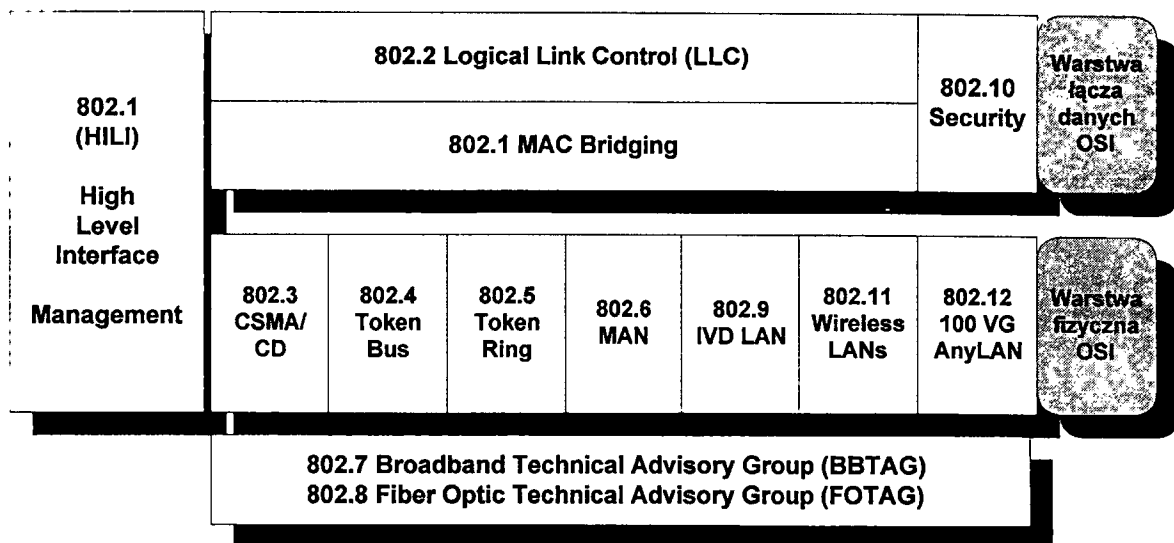
Rys.1.6 Model Odniesienia ISO
i model odniesienia LAN

- 802.7 grupa doradztwa technicznego do spraw szerokiego pasma w sieciach LAN (ang. *broadband LAN*).
- 802.8 grupa doradztwa technicznego do spraw światłowodów w sieciach LAN oraz MAN.
- 802.9 zintegrowana transmisja głosu i danych w sieciach LAN i MAN. Specyfikacja węzłów takich sieci obejmuje komputery, telefony oraz kodery/dekodery video. Generalnie grupa robocza 802.9 zajmuje się zintegrowanymi sieciami cyfrowymi ISDN (ang. *Integrated Services Digital Networks*).
- 802.10 grupa doradcza do spraw bezpieczeństwa, ochrony danych i zasobów sieci.
- 802.11 sieci lokalne bezprzewodowe. Komitet ten zajmuje się problemami transmisji radiowej wąsko- i szerokopasmowej, transmisją w podczerwieni oraz za pomocą linii energetycznych. Standard ten przewiduje stosowanie dwóch rodzajów sieci bezprzewodowych: rozproszonych i z punktem koordynacyjnym.

- 802.12 opis funkcji warstwy MAC oraz warstwy fizycznej dla sieci 100VG AnyLAN. W standardzie tym przewidziano stosowanie techniki dostępu do medium transmisyjnego określanej jako priorytet na żądanie (ang. *Demand Priority*).

Jak wynika z rysunku 1.7, zdefiniowany został tylko jeden wariant warstwy LLC (802.2), który jest wspólny dla wszystkich typów sieci lokalnych. Typ sieci lokalnej jest określony przez warstwę MAC, przy czym danemu protokołowi MAC odpowiada wiele możliwych sposobów realizacji warstwy fizycznej. Określony protokół MAC oraz dopuszczone standardem rodzaje warstwy fizycznej są opisane we wspólnym dokumencie standaryzacyjnym.

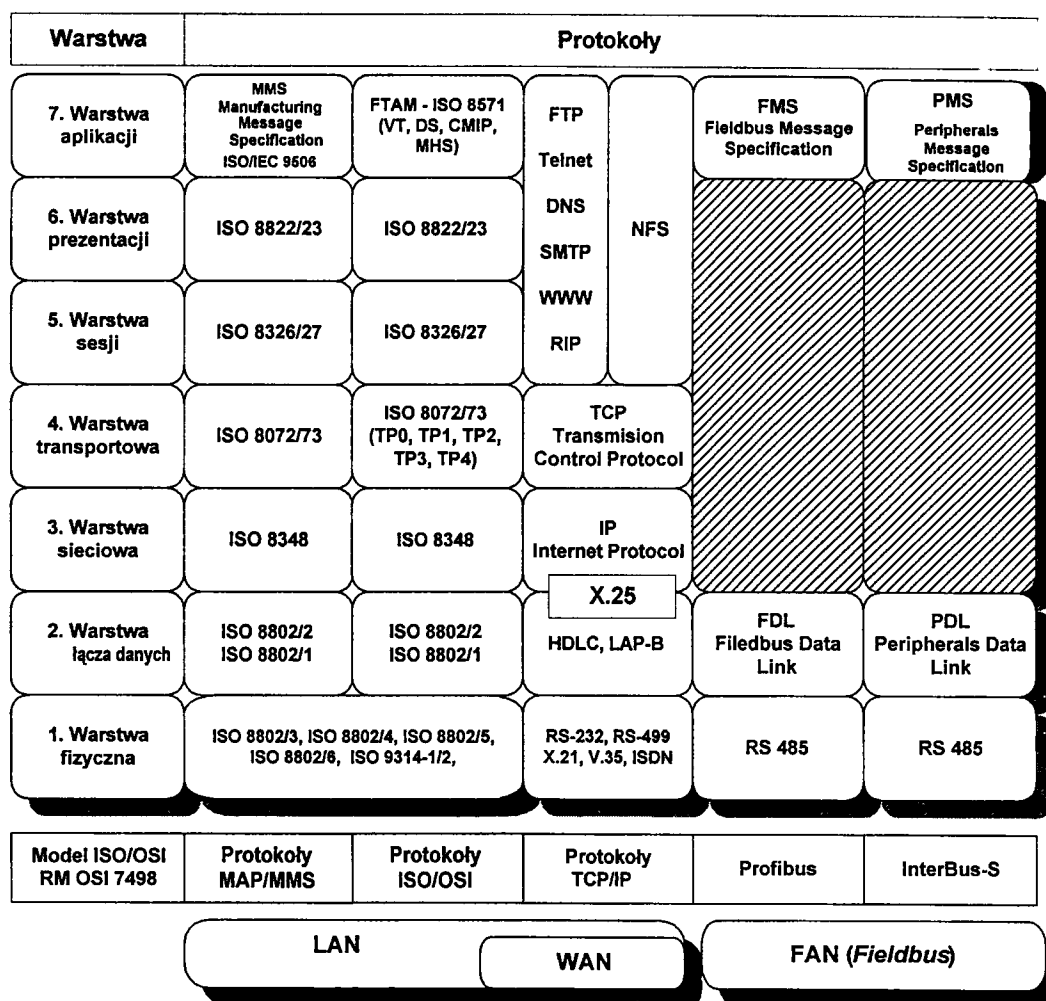
Na rysunku 1.8 zamieszczone zostały stosowane w przemyśle protokoły komunikacyjne dla sieci lokalnych LAN, WAN oraz miejscowych *Fieldbus* w odniesieniu do modelu referencyjnego ISO/OSI (*International Standard Organisation/Open System Interconnection*). Model OSI jest standardem opracowanym przez ISO i definiuje on obszary funkcjonalne, określane jako warstwy, wyznaczające ogólne ramy dla tworzonych przez różne firmy i organizacje standardowych protokołów komunikacyjnych [6].



Rys.1.7 Struktura komitetu IEEE 802

Model ten został zaprojektowany w celu wspomoczenia producentów oprogramowania sieciowego w tworzeniu aplikacji kompatybilnych z aplikacjami innych firm oraz promowania otwartych, współdziałających systemów sieciowych. Model ISO/OSI składa się z siedmiu następujących warstw:

- **Warstwa fizyczna** - definiuje mechaniczne (budowa) oraz elektryczne (prądy, napięcia) cechy interfejsu komunikacyjnego, niezbędne do uaktywniania połączenia między urządzeniami, umożliwiając transmisję bitów przez element fizyczny (okablowanie, łącze radiowe i inne). Przykładami interfejsów tej warstwy są: RS232C, RS499, 10/100BaseT.
- **Warstwa liniowa (łącza danych)** - definiuje reguły przesyłania i otrzymywania informacji poprzez łącze fizyczne między dwoma systemami, zawiera funkcje i procedury kontroli elementów fizycznych łącza, detekcji błędów (powstających w warstwie pierwszej) oraz ich korekty, ustala kolejność dostępu do łącza itp. Przykładami protokołów tej warstwy są: ATM, HDLC, LAP-B, IEEE 802/2/3/4/5.
- **Warstwa sieci** - definiuje protokoły służące zakładaniu i utrzymywaniu ścieżek komunikacyjnych pomiędzy systemami otwartymi w sieci. Zapewnia ona sterowanie strumieniami danych, *routing* oraz adresowanie w sieci. Przykładami protokołów tej warstwy są: IP (TCP/IP), IPX (Novell'a), X.25.
- **Warstwa transportowa** - zapewnia wysoki poziom kontroli przesyłania informacji między systemami (obsługa błędów, wyznaczanie priorytetów, ochrona danych), Warstwa ta zapewnia dokładne adresowanie i bezbłędne przesyłanie lub transfer danych pomiędzy dwoma końcowymi systemami. Przykładami protokołów tej warstwy są: TCP i UDP (rodzina TCP/IP), NetBIOS i NetBEUI (Microsoft), SPX (Novell),



Rys.1.8 Protokoły komunikacyjne MAP/MMS, FTAM, TCP/IP oraz protokoły sieci typu *Fieldbus* w odniesieniu do modelu referencyjnego ISO/OSI

- **Warstwa sesji** - koordynuje wymianę informacji pomiędzy systemami poprzez wykorzystanie technik konwersacji, zarządza m.in. procedurami przerwania i wznowienia dialogu.
- **Warstwa prezentacji** - definiuje składnię danych (strukturę, formatowanie do druku lub wyświetlania na ekranie) będących obiektem komunikacji. Spełnia ona rolę tłumacza między systemami heterogenicznymi (dekodowania, translacji). Zalecany standard w tej warstwie jest ASN.1 (ang. *Abstract Syntax Notation 1*).
- **Warstwa aplikacji** - zapewnia dostęp do usług sieciowych systemu, definiuje funkcje komunikacji pomiędzy systemami umożliwiając aplikacjom heterogenicznym współpracę opartą na wspólnych modułach AE (ang. *Application Entity*) i ASE (ang. *Application Service Element*). Przykładami protokołów tej warstwy są: ACSE, MHS, FTAM, VT, MMS.

Z punktu widzenia projektanta sieciowych systemów komunikacyjnych oraz użytkownika urządzeń transmisji danych, szczególne znaczenie mają elementy usługowe warstwy fizycznej i liniowej (warstwy 1 i 2), jak również warstwa aplikacji (warstwa 7). Znaczenie to należy rozumieć

w taki sposób, że normy dotyczące warstw 1 i 2 określają typ medium transmisyjnego i rodzaj dostępu do tego medium. Elementy usług warstwy aplikacji są natomiast dlatego ważne, ponieważ oddziałują one w sposób bezpośredni na strukturę i funkcjonalność programów użytkownika.

Literatura

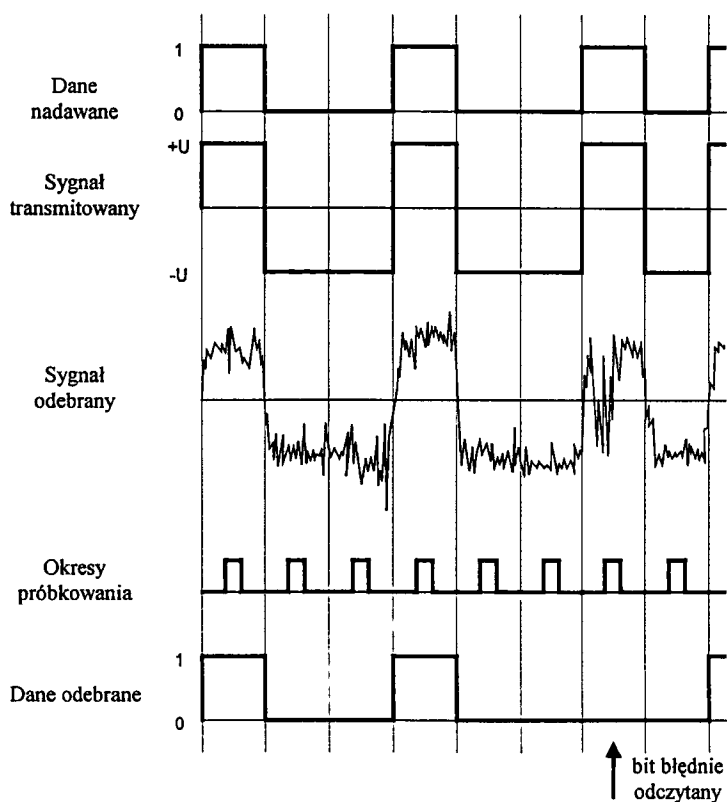
- [1] BAGINSKI A., MÜLLER M., *InterBus-S. Grundlagen und Praxis*, Hüthig Buch Verlag GmbH, Heidelberg, 1994
- [2] CHLEBUS E. i inni, *PBZ-3105, Sieciowe systemy komunikacyjne integrujące automatyzację wytwarzania. Etap I*, Raporty Inst. Technologii Maszyn i Automat. PWr serii: Sprawozdania nr 53/95, Wrocław 1995
- [3] HALSALL F., *Data Communications, Computer Networks and Open Systems*, Addison - Wesley, 1992
- [4] ISO/IEC 9506-2 MMS Part 2, Protocol specification
- [5] ISO/IEC 9506-1 MMS Part 1, Service definition
- [6] JAIN B. N., AGRAWALA A. K., *Open Systems Interconnection. its Architecture and Protocols*, Elsevier, 1990
- [7] Phoenix Contact, *InterBus the Sensor/Actuator Bus*, 1994
- [8] SKURA K., SMALEC Z., REINER J., TRZCIŃSKI P., MRZYGŁÓD M., *PBZ-3105, Sieciowe systemy komunikacyjne integrujące automatyzację wytwarzania. Etap IV*, Raporty Inst. Technologii Maszyn i Automat. PWr serii: Sprawozdania nr 1/97, Wrocław 1996
- [9] SCHNELL G., *Bussysteme in der Automatisierungstechnik*, Vieweg, Braunschweig/Wiesbaden, 1994
- [10] WECK M., *Werkzeugmaschinen. Fertigungssysteme*, VDI, 1995

2. Elektryczne łącza komunikacyjne

2.1 Wstęp

Aby przesłać dane binarne przez linię transmisyjną, każda cyfra binarna informacji musi zostać przetworzona na sygnał elektryczny. Przykładowo, binarne „1” może być transmitowane przez zastosowanie sygnału napięciowego o wartości $+U$ Voltów, a nadanie binarnego „0” i znaku końca transmisji sygnałem o wartości $-U$ Voltów. Przy odbiorze sygnałów, urządzenie odbiorcze interpretuje $+U$ Voltów jako binarne „1” i $-U$ Voltów jako binarne „0”. W praktyce w medium transmisyjnym sygnał jest tłumiony i zniekształcany (rys.2.1), wskutek czego czasami jest niemożliwe odróżnienie sygnału binarnego „1” i „0”. Wielkość tłumienia i zniekształceń jest silnie zależna od:

- typu medium transmisyjnego,
- szybkości transmisji bitów oraz
- odległości pomiędzy dwoma komunikującymi się urządzeniami.



Rys.2.1 Skutek stosowania wadliwego medium transmisyjnego

Dla różnych elektrycznych interfejsów komunikacyjnych istnieją międzynarodowe standardy definiujące wielkość tłumienia i zniekształcenia dla różnych mediów transmisyjnych. Te standardy, oprócz zdefiniowania poziomów sygnałów elektrycznych, zawierają dodatkowe sygnały sterujące i protokoły używane przez interfejsy fizyczne. Organy tworzące standardy dla sieci komunikacyjnych

to w Europie CCITT (fran. *International Telegraph and Telephone Committ  *), a w Stanach Zjednoczonych EIA (ang. *Electrical Industries Association*). Mimo tego,  e standardy te definiuj  dwie r  ne organizacje, kt re pos uguj  si  inn  terminologią, to podstawowe sygnały i ich znaczenie s  takie same.

2.2 Media transmisyjne

Przesyłanie elektrycznych sygnał w wymaga stosowania medium transmisyjnego, kt re zwykle ma form  linii transmisyjnej. Cz sto jest to para przewodnik w lub kabel. Alternatywnie stosuje si  promie   wiatlny w  wiat owodzie lub fale elektromagnetyczne w otwartej przestrzeni. Typ medium transmisyjnego determinuje mi dzy innymi maksymaln  liczb  bit w transmitowan  w czasie jednej sekundy oraz maksymaln  odległość mi dzy urz dzeniem nadaj cym a odbieraj cym.

2.2.1 Otwarta linia dwuprzewodowa

Otwarta linia dwuprzewodowa jest najprostszym medium transmisyjnym, sk adaj cym si  z dw ch odizolowanych od siebie przewod w, kt rych ko ce nie s  zako czone terminatorami. Ten typ linii jest odpowiedni do po łącze  dwu punktowych o d ugościach nie przekraczaj cych 50m i małych szybkościach transmisji (do 19.2 kbit/s). Sygna , typowy poziom napi ciowy lub pr dowy wzgl dem masy, do  czony jest do jednego z przewod w, a masa do  czona jest do drugiego. Cz sto stosuje si  po łączenia za pomoc  wielu takich linii; realizuje si  to za pomoc  kabla wielo yłowego lub ta my. Otwarta linia dwuprzewodowa jest podatna na zewn trzne zak łocenia magnetyczne, przes uchy i spr ęgi pojemno ciowe.

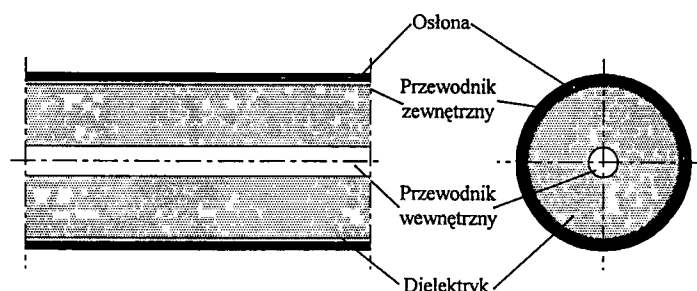
2.2.2 Skr tka

Skr tka ma post c pary identycznych izolowanych przewodnik w, splecionych i umieszczonych we wsp lnej os lonie izolacyjnej, a cz sto w pojedynczej os lonie prowadzi si  wi ksz  liczb  par. Typowo jeden skr t przypada na 15 cm d ugości kabla. Jako przewodnika u ywa si  litego drutu miedzianego lub wielo  knowej linki miedzianej. Przez skr cenie pary identycznych przewod w sygnały napi ciowe wygenerowane w s siednich przewodach wskutek dzia ania zewn trznego pola magnetycznego znos  si  wzajemnie, dzi ki czemu uzyskuje si  znaczny stopie  odporności na zak łocenia. Skr tka jest u ywana wy acznie do transmisji w pa mie podstawowym przy szybkościach sygnalizacji dochodz cych do kilku milion w zmian sygna u na sekund . Klasyczna skr tka jest ma o odporna na zak łocenia zewn trzne i emituje zak łocenia na zewn trz do otoczenia. Jest ona podatna na nielegalne przechwycenie informacji przez analiz  emitowanego pola magnetycznego. Podstawow  zalet  skr tki jest natomiast jej najni sza - w por wnaniu z innymi rodzajami kabli - cena i  atwo c instalacji, w zwi zku z do c du   elastyczno ci  i małym ci żarem jednostkowym. Coraz cz ściej stosuje si  skr tk  w pojedynczym

lub w podwójnym ekranie, która zachowując zalety skrętki klasycznej, równocześnie jest pozbawiona jej wad, ma znacznie szersze pasmo przenoszenia, mniejszy stopień tłumienia, wykazuje znaczne ograniczenie emisji pola magnetycznego oraz małą wrażliwość na zakłócenia zewnętrzne. Skrętka ekranowana umożliwia transmisję rzędu kilkudziesięciu milionów bitów na sekundę.

2.2.3 Kabel koncentryczny

Kabel koncentryczny jest zbudowany z dwóch przewodów, umieszczonych jeden wewnątrz drugiego, przy czym przewód wewnętrzny ma kształt walca wykonanego z litego drutu lub wielowłóknowej linki, a zewnętrzny - odseparowany warstwą izolacji - stanowi cienki cylinder z litej blachy lub ażurowej siatki. Całość jest umieszczona w izolacyjnej osłonie, a czasem stosuje się jeszcze dodatkowo współosiowe ekrany (rys.2.2). Zewnętrzny przewód, zwykle uziemiony, pełni równocześnie funkcję przewodu zerowego oraz ekranu dla przewodu wewnętrznego, zapewniając znacznie mniejszą niż w przypadku skrętki wrażliwość na pola magnetyczne i małą ich emisję.

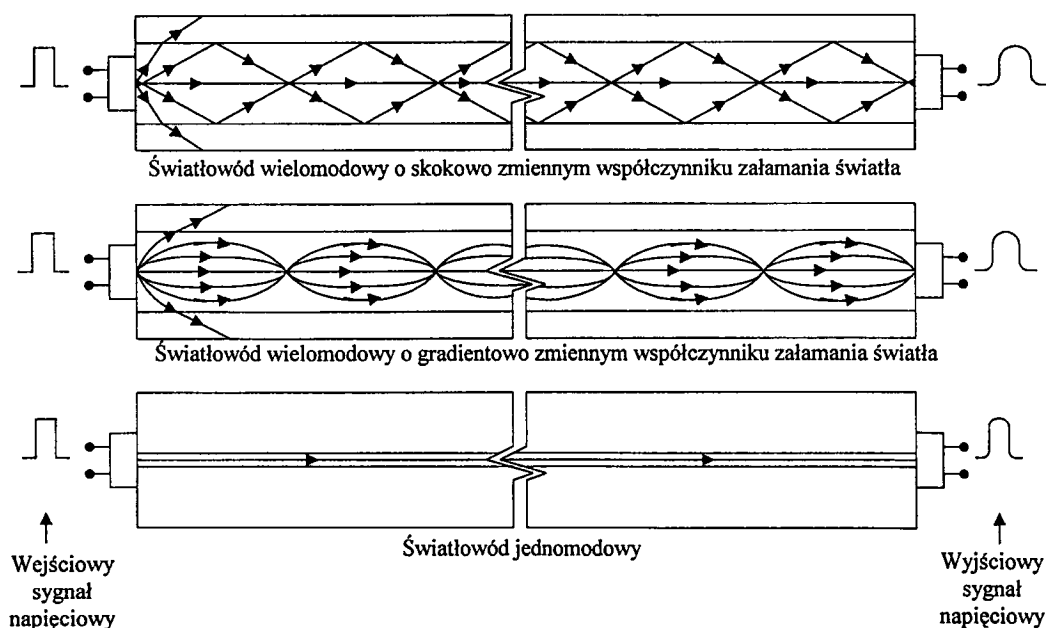


Rys.2.2 Budowa przewodu koncentrycznego

Kable koncentryczne, w porównaniu ze skrętką, mają także znacznie mniejszy wzrost stopnia tłumienia w funkcji częstotliwości. Przy przesyłaniu sygnałów o wysokiej częstotliwości występuje tzw. efekt powierzchniowy, który polega na tym, że sygnał rozprzestrzenia się nie w całym przekroju przewodnika lecz wyłącznie w cienkiej warstwie zewnętrznej. Dlatego też zwiększenie średnicy przewodu wpływa bardzo silnie na zmniejszenie stopnia tłumienia sygnałów o wysokich częstotliwościach. Pasmo przenoszenia kabli koncentrycznych sięga kilkuset megaherców, a przy transmisji szerokopasmowej wykorzystuje się zwykle zakres do 600 MHz. Na ogół stosuje się kable o impedancji równej 50, 75 i 93 Ω . Kable o większych średnicach odznaczają się znaczną sztywnością, wyrażającą się dużymi wartościami minimalnych promieni skrętu i ciężarem, co utrudnia ich układanie. Kable koncentryczne umożliwiają tworzenie konfiguracji wielopunktowych liniowych, przy czym dla kabli o litym przewodzie wewnętrznym i większych średnicach są dostępne przyłącza nie wymagające przecinania kabla - tzw. przyłącza penetrujące, które umożliwiają łatwe dołączanie nowych stacji bez konieczności przerywania pracy sieci.

2.2.4 Światłowód

Transmisja sygnału świetlnego przez światłowód opiera się na zjawisku całkowitego wewnętrznego odbicia światła na granicy dwóch ośrodków o różnych współczynnikach załamania światła. Promienie świetlne padające pod małym kątem do osi światłowodu, ulegają całkowitemu wewnętrznemu odbiciu i rozchodzą się w jego wnętrzu. Największy kąt, przy którym to zjawisko zachodzi, nosi nazwę rozwartości optycznej i stanowi, obok wymiarów geometrycznych D_r i D_p (D_r – średnica rdzenia, D_p – pokrycia), podstawową cechę światłowodu. Jeśli średnica D_r jest rzędu 8 - 9 μm , to dostać się do niego może wyłącznie światło lasera, którego promienie są równoległe. Takie światłowody charakteryzują się bardzo małym tłumieniem i określa się je jako jednomodowe (rys.2.3). Jednakże ze względu na małą średnicę D_r i małą rozwartość optyczną rzędu 0.1° , wymagają one źródeł światła o dużej mocy generowanej z jednostki powierzchni. Przy większych średnicach D_r mogą być stosowane promienie nierównoległe, a światłowód taki nosi nazwę wielomodowego. Jeśli światłowód jest wykonany z materiału o skokowo zmiennym współczynniku załamania światła, to promienie świetlne padające pod różnym kątem muszą przebyć różną drogę, przez co dochodzą do wylotu światłowodu po różnym czasie, a sygnał przez nie przenoszony staje się rozmyty. Zniwelowanie tego zawisaka uzyskuje się przez stosowanie światłowodu o zmiennym wzdłuż średnicy współczynniku załamania światła, tzn. że wraz z oddalaniem się od osi prędkość światła w warstwach światłowodu zwiększa się. W światłowodzie tego rodzaju, promienie padające pod różnym kątem wnikają przed odbiciem na różną głębokość, dzięki temu czas przejścia wyrównuje się. Nadajnikami sygnału świetlnego mogą być diody świecące LED lub diody laserowe ILD, a jako odbiorniki pracują najczęściej fotodiody lub fototranzystory. Łącze światłowodowe jest całkowicie odporne na zakłócenia elektromagnetyczne i także nie emituje pola elektromagnetycznego.



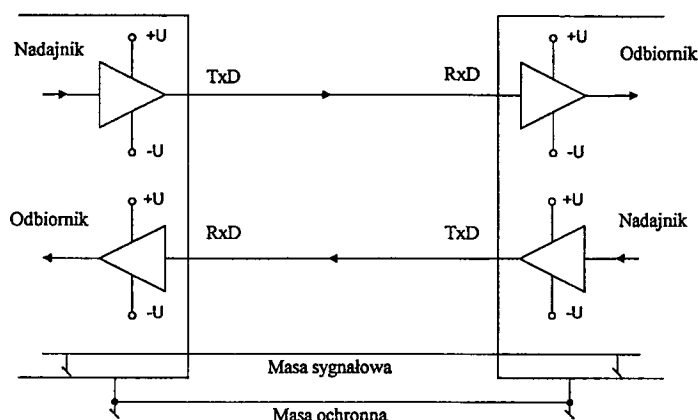
Rys.2.3 Zasada transmisji światłowodowej

Jest ono również niewrażliwe na różnicę potencjałów w punktach końcowych i nie iskrzy w przypadku mechanicznego uszkodzenia. Instalacja światłowodów nie stwarza dużych problemów, łącze jest lekkie, może być prowadzone obok kabli energetycznych i w pobliżu źródeł pola elektromagnetycznego. Należy jednak unikać załamania światłowodu, ale dopuszczalny promień zagięcia jest na tyle mały, że nie powoduje to trudności. Obecnie dla sieci typowe jest używanie szklanych światłowodów wielomodowych o zmiennym współczynniku załamania o wymiarach D_r/D_p 62,5/125 μm oraz światłowodów z tworzywa sztucznego o wymiarach 980/1000 μm . Światłowody z tworzywa sztucznego mimo tego, że zarówno one same jak i ich oprzyrządowanie jest dużo tańsze niż światłowodów szklanych, stosowane są rzadko, gdyż cechują się większym współczynnikiem tłumienia i matowieniem wraz z upływem czasu.

2.3 Sygnały transmisji danych

2.3.1 Standard RS-232C/V.24

RS-232C (zdefiniowany przez EIA) i V.24 (zdefiniowany przez CCITT) przedstawiają wymagania, jakie powinien spełniać moduł sprzęgający służący do asynchronicznej, szeregowej transmisji danych. Znaki przesyłane są bit po bicie za pomocą pojedynczej linii danych. W interfejsie tym przewiduje się pracę w układzie asymetrycznym (rys.2.4) na łączu dwupunktowym. W przypadku zastosowania kilku równoległych torów transmisji danych wykorzystują one wspólny przewód "zerowy". W zaleceniach nie przewiduje się stosowania poziomów TTL, lecz przeważnie +12 V dla stanu logicznego „0” i -12 V dla stanu logicznego „1”.

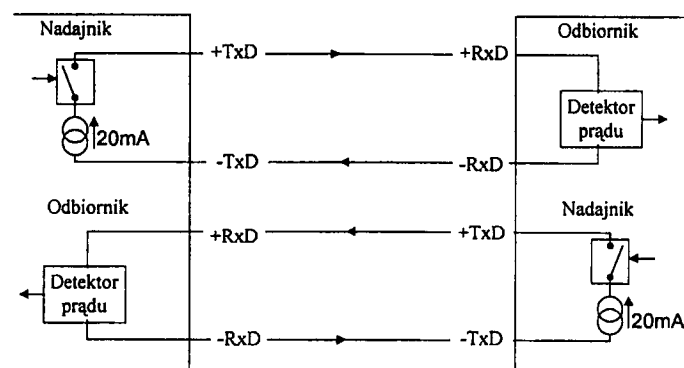


Rys.2.4 Asymetryczny interfejs RS-232C

Maksymalny dopuszczalny zakres napięć wynosi -15...+15 V. Zakres -3...+3 V jest wzbroniony, co oznacza, że na żadnym z wejść interfejsu nie może pojawić się napięcie o wartości bezwzględnej mniejszej od 3 V. Dzięki tak dużym skokom napięcia transmisja danych jest odporna na zakłócenia nawet przy wykorzystaniu nie ekranowanych przewodów. Długość łącza jest ograniczona narzuconą w normie dopuszczalną pojemnością kabla, która nie powinna przekraczać 2500 pF. Dlatego też długość łącza jest ograniczona do 15m, a szybkość transmisji do 19200 bit/s.

2.3.2 Pętla prądowa 20 mA

Przy bardzo długich liniach przesyłowych moduł V.24 nie pracuje poprawnie, gdyż napięcia pasożytnicze powstające w linii na skutek sprzężeń indukcyjnych mogą prowadzić do powstawania błędów transmisji. W takich przypadkach stosuje się tzw. moduł TTY (moduł dalekopisowy), który jest bardziej odporny na zakłócenia, co pozwala na transmisję na odległości dochodzące do 1 km (rys.2.5).

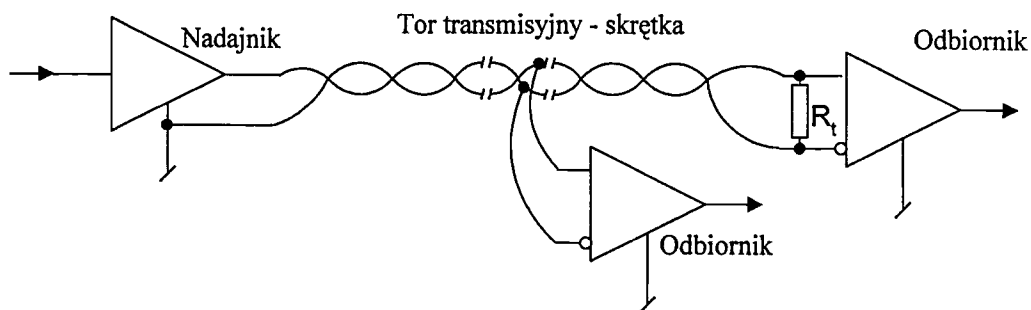


Rys.2.5 Standard TTY – pętla 20mA

Ten moduł sprzęgający jest nazywany pętlą 20 mA, gdyż logiczna jedynka powoduje w nim przepływ prądu o natężeniu ok. 20 mA, natomiast logiczne zero - brak przepływu prądu. W module pętli prądowej nie definiuje się żadnego potencjału odniesienia, ani wspólnej masy lub uziemienia, a stanem spoczynku jest logiczne „1”, odpowiadające przepływowi prądu o natężeniu 20 mA. Ponieważ między dwoma urządzeniami połączonymi ze sobą za pomocą pętli prądowej nie istnieje żaden potencjał odniesienia, to najczęściej stosuje się tu transoptory w celu zupełnego odizolowania galwanicznego nadajnika i odbiornika.

2.3.3 Standard RS-423A

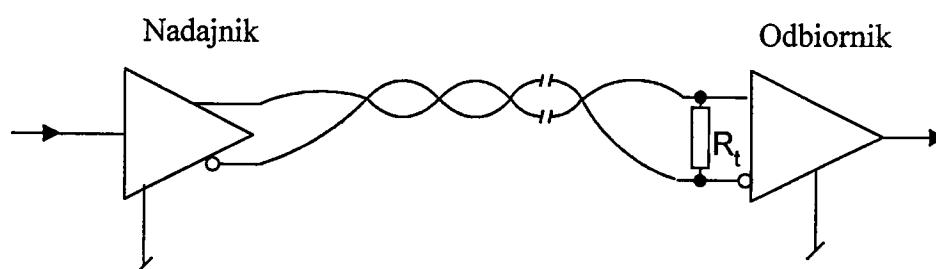
Standard RS-423A określa elektryczną charakterystykę napięciowego obwodu transmisyjnego, który jest złożony z niesymetrycznego nadajnika i symetrycznego odbiornika. Typowa szybkość transmisji dla tego standardu to 100 kbit/s dla odległości nie przekraczającej 30 m, przy czym szybkość ta maleje do 3 kbit/s przy 1200 m. Schemat połączenia nadajnik-odbiornik pokazano na rys.2.6. Zastosowanie różnicowego odbiornika pozwala na znaczne zmniejszenie wpływu napięcia wspólnego powstałego na skutek różnicy potencjałów masy nadajnika i odbiornika, jak również przesłuchów między liniami. Maksymalna bezwzględna wartość napięcia wspólnego może dochodzić do 6 V. Nadajnik generuje dwa poziomy sygnały napięciowego: wysoki od 3,6 V do 6 V i niski od -6 V do -3,6 V. Dla zmniejszenia przesłuchu wprowadzono ograniczenie szybkości zmian napięcia do 15 V/ μ s. Do pojedynczego nadajnika można podłączyć do 10 odbiorników, których impedancja wejściowa nie może być mniejsza niż 4 k Ω . Całkowite obciążenie wnoszone przez odbiorniki i inne obwody musi być większe niż 400 Ω .



Rys.2.6 Połączenie nadajnik - odbiornik w interfejsie RS-423A

2.3.4 Standard RS-422A/V.11

Transmisja sygnałów cyfrowych z większymi szybkościami na większe odległości pozwala realizować zalecenie RS-422/V.11. Standardy te definiują symetryczny, zrównoważony system transmisji danych złożony z różnicowego nadajnika, dwuprzewodowego zrównoważonego toru przesyłowego oraz odbiornika z różnicowym obwodem wejściowym (rys.2.7). Prędkość transmisji wynosi od 100 kbit/s przy odległościach 1200 m, do 10 Mbit/s dla odległości nie przekraczających 30 m. Standard RS-422A nie wprowadza ograniczeń na minimalną lub maksymalną częstotliwość, a jedynie na zależność pomiędzy czasem trwania bitu a szybkością zmian sygnału.



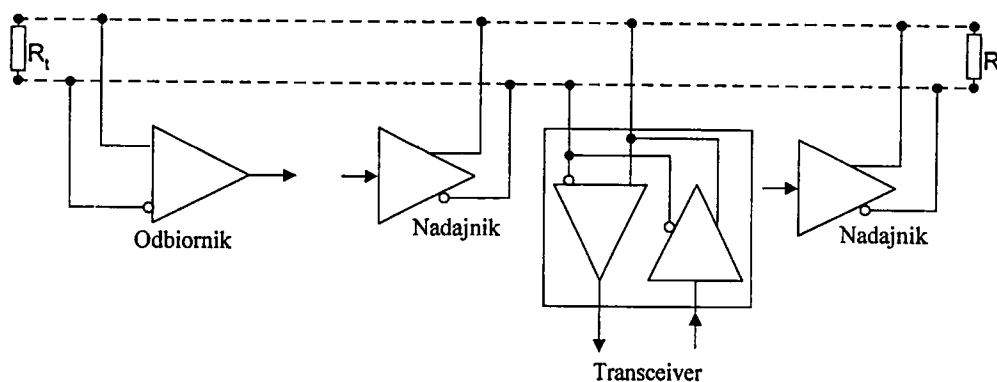
Rys.2.7 Połączenie nadajnik – odbiornik w interfejsie RS-422A

Odbiornik interfejsu RS-422A jest w zasadzie taki sam jak interfejsu RS-423A. Wyjściowe napięcie różnicowe nadajnika generowane jest na poziomie od 2 V do 6 V, a jego rezystancja wewnętrzna jest nie większa od 100 Ω . Ograniczenie widma sygnału uzyskuje się przez określenie czasu narastania (opadania) napięcia wyjściowego, który nie może być mniejszy niż 10% czasu trwania bitu lub 20 ns, w zależności od tego, która z tych wartości jest większa. Połączenia między urządzeniami realizuje się za pomocą skrętki. Odbiornik odznacza się dużą impedancją (nie mniejszą niż 4 k Ω) i strefą nieczułości $\pm 0,2$ V. Do jednego nadajnika można podłączyć do 10 odbiorników, przy czym w celu uniknięcia odbić sygnału tor przesyłowy zamyka się rezystorem dopasowującym o wartości równej impedancji falowej, na wejściu odbiornika najbardziej oddalonego od nadajnika.

2.3.5 RS-485

Standard RS-485 jest rozwinięciem RS-422A (rys.2.8). Jest to również interfejs symetryczny i zrównoważony, przy czym dopuszcza się w nim nie tylko pracę wielu odbiorników, ale również wiele nadajników podłączonych do jednej linii. Nadajniki muszą być trójstanowe, a ponieważ w jednej chwili może nadawać tylko jeden z nich, to pozostałe muszą być w stanie wysokiej impedancji.

Dopasowanie toru transmisyjnego stanowią rezystory umieszczone na końcach linii przesyłowych. Nadajnik powinien zapewnić różnicowe napięcie wyjściowe z przedziału 1,5 V do 5 V, przy obecności napięcia wspólnego w zakresie -7 V do 12 V.



Rys.2.8 Wielopunktowe połączenie interfejsu RS-485

Jeden nadajnik może sterować do 32 odbiorników. Rezystancja wejściowa odbiornika jest niezależna od tego czy w danej chwili urządzenie jest zasilane czy też nie i wynosi 120 k Ω . Minimalna obciążalność nadajnika wynosi 60 Ω .

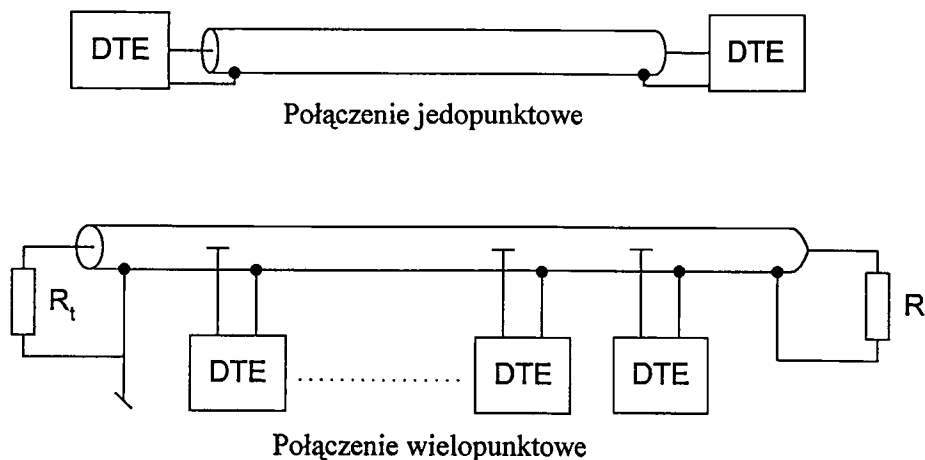
2.3.6 Transmisja kablem koncentrycznym

W przeciwieństwie do łączy realizowanych za pomocą skrętki, kable koncentryczne mają szersze pasmo przenoszenia dochodzące do 600 MHz. Ten potencjał może być wykorzystany na dwa sposoby:

- transmisja w paśmie podstawowym, w której całe dostępne pasmo jest wykorzystane do transmisji pojedynczego sygnału przy dużych szybkościach (10 Mbit/s lub wyższa),
- transmisja szerokopasmowa, w której dostępne pasmo jest dzielone na kanały o małym paśmie przenoszenia, dzięki czemu uzyskuje się możliwość transmisji wielu sygnałów w jednym kablu.

2.3.7 Transmisja w paśmie podstawowym

W transmisji w paśmie podstawowym, normalnie stosuje się kabel zasilany z jednego źródła zasilania. Typowe połączenia są pokazane na rys.2.9.

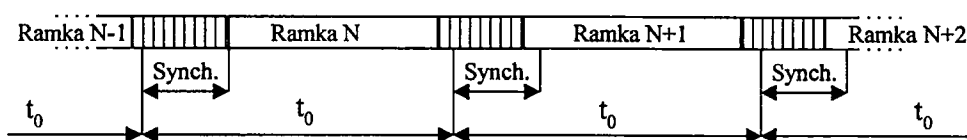


Rys.2.9 Typy połączeń stosowanych przy transmisji w paśmie podstawowym

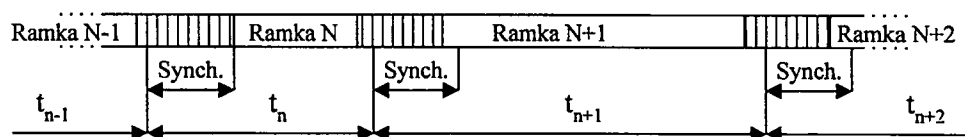
W wielu aplikacjach, transmisja kablem koncentrycznym stosowana jest do komunikacji dwupunktowej (*point-to-point*), a w przypadkach konfiguracji wielopunktowej stosuje się podział czasu transmisji TDM (ang. *Time Division Multiplexing*). Są stosowane dwie metody TDM:

- **Synchroniczna:** każdy użytkownik uzyskuje dostęp do kanału w ściśle zdefiniowanym interwale czasowym, a każda z transmitowanych ramek ma ustaloną długość. Aby rozdzielić poszczególne ramki, na początku każdej z nich nadawany jest specjalny wzór bitów zwany wzorcem synchronizacji. Każdy dołączony system potrafi wyznaczyć początek ramki oraz pozycje ramki (numer ramki) w kompletnym cyklu ramek.
- **Asynchroniczna:** użytkownicy mają losowy dostęp do kanału, a w jednej chwili tylko jeden użytkownik ma dostęp do medium i zajmuje je na czas trwania własnej transmisji.

Na rys.2.10 przedstawione są dwie formy transmisji typu TDM.



Transmisja synchroniczna w paśmie podstawowym

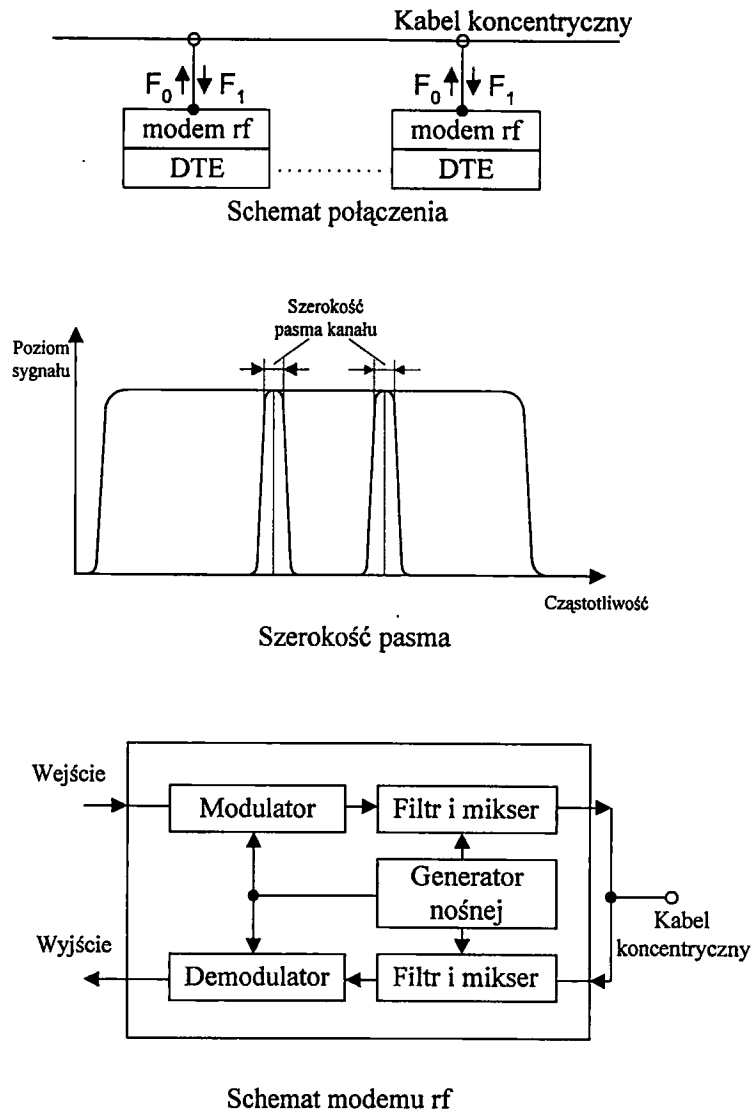


Transmisja asynchroniczna w paśmie podstawowym

Rys.2.10 Metody TDM stosowane w transmisji w paśmie podstawowym

2.3.8 Transmisja szerokopasmowa

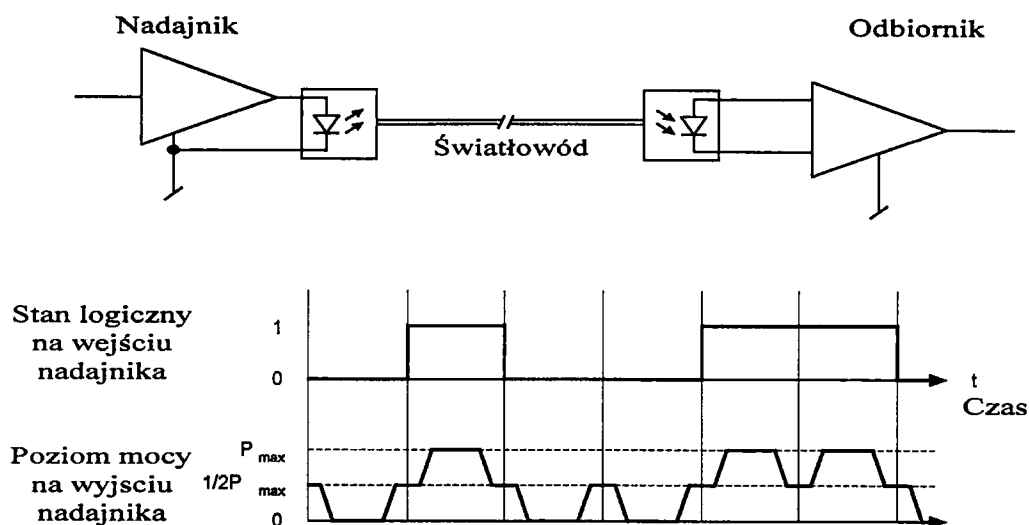
Używanie transmisji szerokopasmowej umożliwia zwielokrotnienie (niezależnych i współbieżnych) kanałów transmisyjnych, przy zastosowaniu techniki zwanej multipleksowaniem z podziałem częstotliwości FDM (ang. *Frequency-Division Multiplexing*). FDM wymaga stosowania urządzeń zwanych „modemem rf” dla każdego z urządzeń dołączonych do sieci, w których sygnał przeznaczony do nadania jest zmodulowany częstotliwością nośną, a sygnał odebrany poddaje się procesowi demodulacji. Modulacji może podlegać dowolny parametr sygnału: amplituda, częstotliwość lub faza. Komunikacja na każdym z udostępnionych kanałów jest analogiczna do komunikacji w paśmie podstawowym. Szerokość wymaganego pasma dla każdego z kanałów jest zależna od pożądanej szybkości transmisji oraz metody modulacji. Dla szybkości 9600 bit/s wymagany jest kanał o szerokości 20 kHz i około 18 MHz dla szybkości 10 Mbit/s. Podstawę pracy transmisji szerokopasmowej i schemat blokowy „modemu rf” przedstawia rys.2.11.



Rys.2.11 Reguły transmisji szerokopasmowej

2.3.9 Transmisja światłowodowa

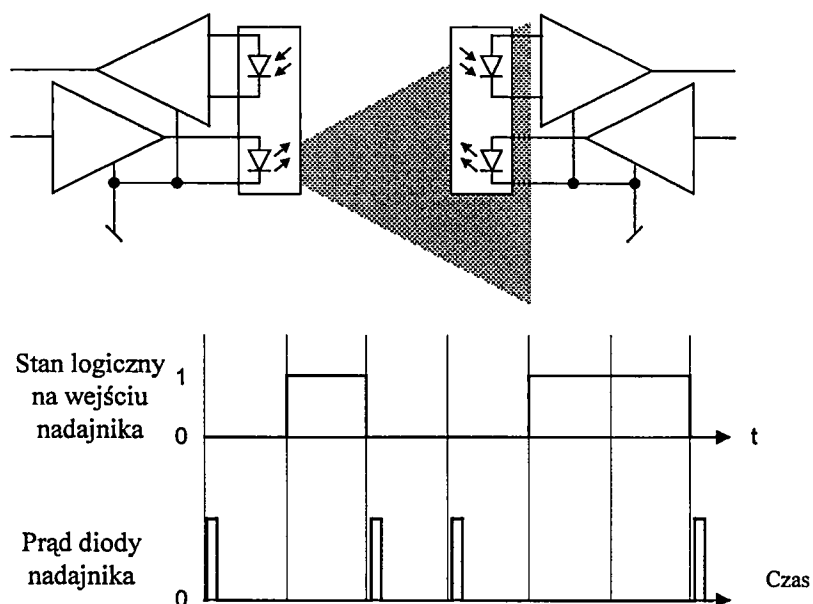
Istnieje wiele różnych metod kodowania sygnału optycznego. Jedną z nich - kodowanie bipolarne - przedstawia rys.2.12. Wykorzystuje ona trzy poziomy mocy wyjściowej promienia świetlnego: zero, połowę mocy maksymalnej i maksymalną moc. Nadajnik przetwarza wewnętrzne binarne poziomy napięcie na trójpoziomowy sygnał świetlny, który jest wprowadzany do światłowodu za pomocą specjalnego złącza i szybkiej diody LED lub ILD. W odbiorniku umieszczonym na końcu światłowodu, sygnał wyjściowy z szybkiej fotodiody, proporcjonalny do mocy odbieranego promienia świetlnego, przetwarzany jest na wewnętrzny binarny poziom napięcie. Kodowanie według tej metody jest odpowiednie do szybkości transmisji dochodzącej do 50 Mbit/s.



Rys.2.12 Kodowanie bipolarne przy transmisji światłowodowej

2.3.10 IrDA

IrDA (ang. *Infrared Data Association*) jest to standard komunikacji bezprzewodowej do szybkiej wymiany danych, wykorzystujący promieniowanie podczerwone (rys.2.13). System ten realizuje transmisję półdupleksową, tzn. w tym samym czasie może się odbywać przesyłanie danych tylko w jednym kierunku.



Rys.2.13 Transmisja według standardu IrDA

Długość fali elektromagnetycznej, emitowanej przez szybką diodę LED wchodzącą w skład nadajnika, mieści się w przedziale 850-900 nm. Maksymalna szybkość transmisji zdefiniowana dla tego standardu wynosi 115.2 kbit/s (istnieją także rozwiązania z szybkością transmisji dochodzącą

do 4 Mbit/s). Dane są reprezentowane jako ciąg impulsów o szerokości zawartej między 1,6 ms a 3/16 czasu trwania bitu, przy czym nadawane są tylko stany niskie - logiczne „0”. Brak sygnału w fotodiodzie odbiornika jest interpretowany jako „1” logiczna. Taki sposób przesyłania danych zapewnia minimalizację mocy dostarczanej do nadajnika, gdyż w transmisji asynchronicznej częściej występuje poziom wysoki. Rozwiązanie takie poszerza spektrum zastosowań o urządzenia bateryjne, np. telefony bezprzewodowe, palmtopy, laptopy, notatniki elektroniczne. Standardowa odległość między nadajnikiem a odbiornikiem wynosi 1 m, lecz przez zastosowanie większej liczby diod nadawczych może być w łatwy sposób powiększona do kilku metrów. W celu zawężenia odbioru do pojedynczego urządzenia płaski kąt promieniowania w płaszczyźnie poziomej ogranicza się do $\pm 30^\circ$.

3. Podstawowe metody transmisji danych

3.1 Transmisja równoległa

W przypadku interfejsów transmisji równoległej, obok linii danych, występują zawsze linie sterujące. Sygnały te określane jako „*handshake*” synchronizują pracę odbiornika, przekazując informacje o gotowości (ważności) danych i ich odczytaniu. Dla transmisji równoległej pojęcie synchronizacji bitowej jest równoznaczne z synchronizacją słowa (bajtową), gdyż wszystkie bity słowa dostępne są dla odbiornika równocześnie. O transmisji tej mówi się również, że jest równoległa-bitowo lub szeregowo-bajtowo.

Sygnały synchronizujące są najczęściej opisywane jako *strobe*, *clock*, *qualifier*. Dla odebrania danych przewidziany jest określony czas, w którym dane (sygnały) pozostają stabilne na magistrali. Początek tego okresu sygnalizowany jest zwykle zboczem sygnału synchronizującego, natomiast zakończenie wystawieniem sygnału kwitującego *Ready* lub potwierdzającego *Busy*, *Acknowledge*.

3.2 Szeregowo transmisja synchroniczna

W transmisji szeregowej synchronicznej z taktiem zegarowym przesyłanym przez dodatkową linię, zależności synchronizacyjne są podobne do występujących podczas transmisji równoległej. Dla transmisji synchronicznej wykorzystuje się zarówno falę synchronizującą deterministyczną jak i falę o zmiennych odstępach czasowych. Jednakowe odstępy czasowe między impulsami taktującymi pozwalają po stronie odbiorczej na odfiltrowanie pojawiających się zakłóceń. Nieregularnie występujące takty synchronizujące określają ważność bitów danych. Najczęściej zbocze impulsu taktującego jest wykorzystywane do zatrzaśnięcia impulsu danych w rejestrze.

Synchronizacja bajtowa realizowana jest przez dodatkową linię impulsów synchronizujących *Byte-Qualifier*, albo też musi być rozpoznawana na podstawie zawartości strumienia bitów danych. W tym celu w nieprzekraczalnych odcinkach czasowych przesyłane są określone kombinacje bitowe. Ponieważ nie przenoszą one informacji o danych, to muszą zostać odfiltrowane. Takie znaki synchronizacyjne służą do oznaczania np. początku nowego bajta.

Protokół BCS (ang. *Binary Synchronous Protocol*) został opracowany przez firmę IBM. Do synchronizacji słów 7-bitowych stosowany jest znak SYN (Hex 16), który jest zwykle transmitowany we wszystkich przerwach transmisji danych, a przynajmniej na zakończenie bloku danych. Ponieważ protokół ten przewidziany jest jedynie do transmisji liter i cyfr kodowanych za pomocą 7-miu bitów, to znak SYN nie może wystąpić podczas transmisji bloku danych. Podczas transmisji dowolnych, binarnych bloków danych, znaki synchronizujące muszą być jednak szczególnie chronione.

3.3 Protokół HDLC

Warstwa 2 OSI służy do transmisji i ochrony danych. Specjalne protokoły dla tej warstwy OSI oferuje zestaw protokołów do szeregowego przesyłania bitów danych, który został ujęty w normie

ISO 1987. Są to protokoły transmisji danych znane pod nazwą HDLC (ang. *High Level Datalink Control Procedures*), które znalazły szerokie zastosowanie.

Transmisja danych HDLC przebiega synchronicznie lub w trybie start/stop i obejmuje wielokrotne zabezpieczenie. Pakiet danych zawiera każdorazowo adres, informację kontrolną i sterującą, przesyłane dane i informacje sprawdzające.

Jeżeli są transmitowane bajty i kolejność transmisji poszczególnych bitów w obrębie jednego bajta w poniższym przykładzie jest od najmniej znaczącego do przodu, to dla przesyłanego ciągu znaków ASCII „987” uzyskuje się odpowiedni ciąg bitów:

„10011100 10001100 01111100”.

Jeżeli ten ciąg bitów jest określony, to wtedy na końcu dane są tak zmienione, że maksymalnie na logiczną jedynekę może być po kolei ustawionych pięć bitów, tzn. że maksymalnie po sobie może wystąpić pięć jedynek. Dlatego też po każdym ciągu pięciu jedynek jest wstawiane jedno zero, które odbiornik później odfiltrowuje. Ten sposób postępowania określa się jako *Stuffing*. Dla powyższego przykładu ma on następującą postać:

„10011100 10001100 011111 0 00”.

Na początku i po zakończeniu każdej ramki HDLC są przesyłane znaczniki początku i końca. Składają się one każdorazowo z ciągu sześciu jedynek ujętych zerami (heksadecymalnie „7E”, binarnie „01111110”). Te tzw. *Flag-Byte*, zależnie od uzgodnienia i rodzaju transmisji, można przysyłać raz lub wielokrotnie przed i po pakiecie danych. W niektórych transmisjach danych wówczas, gdy nie ma żadnej komunikacji, to przez kabel są przesyłane permanentnie *Flag-Byte*. Natomiast w innych rodzajach transmisji przed każdym nowym telegramem jest przesyłana wcześniej ustalona ilość *Flag-Byte*, tak że odbiornik może się na nich zsynchronizować.

Ramka HDLC składa się konsekwentnie z wiodącego *Flag-Byte* (01111110), adresu stacji danych (8 bitów), informacji kontrolnych (8 bitów), bitów informacyjnych (nie określona ilość bitów), sumy kontrolnej (*Frame Check Sequence* FCS, 16 lub 32 bity) i na zakończenie znowu *Flag-Byte* (01111110).

Flaga	Adres	Kontrola	Informacja	FCS	FLAGA
01111110	8 bitów	8 bitów	n bitów	16 lub 32 bity	01111110

Bity informacyjne nie są zawarte we wszystkich telegramach, natomiast wszystkie pozostałe części telegramu są obligatoryjne. We wszystkich wymienionych częściach składowych ramek HDLC, oprócz bitów informacyjnych, poszczególne bity są przekazywane w kolejności od najniższej wartości do najwyższej wartości bajta. Dla bitów informacyjnych nie jest natomiast określona żadna kolejność. Ilość bitów informacyjnych nie jest zadawana, ale musi być podzielna przez 8, tzn. może być ona arbitralnie wybierana przez użytkownika.

Struktura pola kontrolnego jest określona w normie HDLC. Rozróżnia się trzy rodzaje telegramów z przynależnymi komendami i odpowiedziami. Istnieją następujące telegramsy:

- do transferu informacji (*format I*),

- do nadzorowania komunikacji danych (*format S*) oraz
- takie, które mają nie numerowany format (*format U*).

Struktura pola kontrolnego dla tych trzech różnych formatów jest pokazana w poniższej tabeli.

	1	2	3	4	5	6	7	8
Format I	0		N(S)		P/F		N(R)	
Format S	1	0	S	S	P/F		N(R)	
Format U	1	1	M	M	P/F	M	M	M

Zawarte w tablicy skróty oznaczają:

N(S) - licznik ciągu wysyłanego

P/F - bit *Pool* i końcowy

N(R) - licznik ciągu odbieranego

S, M - bity funkcyjne

Telegramy HDLC w formatach I lub S zawierają liczniki, które służą do zabezpieczenia danych. Licznik ciągu wysyłanego N(S) jest inkrementowany z każdym wysłanym telegramem (modulo 8) i podaje, ile telegramów wysłała już stacja nadająca. Natomiast licznik ciągu odbieranego N(R) podaje na jaki telegram odbiornik właśnie czeka (modulo 8) i potwierdza on wejście wszystkich telegramów z niższym numerem odbioru.

Numery ciągu wysyłanego i odbieranego są określane na podstawie zmiennych statusu nadawania i odbioru, które są zarządzane u partnerów komunikacyjnych. W normie HDLC są zdefiniowane trzy klasy transmisji. Są to:

- *Normal Response Mode* (NRM),
- *Asynchronous Response Mode* (ARM) oraz
- *Asynchronous Balanced Mode* (ABM).

Normal Response Mode (NRM) opisuje pracę *master-slave*, w której stacja pierwotna (*master*) komunikuje się ze stacją wtórną (*slave*). Stacja wtórna nie może nadawać spontanicznie, lecz tylko na żądanie stacji pierwotnej.

W *Asynchronous Response Mode* (ARM) stacja wtórna może wysyłać telegramy spontanicznie, bez wcześniejszego oczekiwania na zezwolenie stacji pierwotnej. Transmitowane są przede wszystkim pola informacyjne (ramki I) i informacje o zmianach statusu.

W *Asynchronous Balanced Mode* (ABM) obie stacje są równouprawnione, gdyż obie zawierają każdorazowo wszystkie funkcje stacji pierwotnej i wtórnej i dlatego też są one określane jako stacje kombinowane.

Protokół HDLC obejmuje rozkazy dla budowy połączenia, przebiegu transmisji oraz zrywania połączenia. Na przykładzie *Normal Response Mode* są pokazane komendy i odpowiedzi repertuaru funkcji bazowych.

Komendami bazowymi są:

I Wysłanie ramki informacji,

RR Informacja, że kanał odbiorczy jest gotowy (*Receiver Ready*),

RNR Informacja, że kanał odbiorczy nie jest gotowy (*Receiver Not Ready*),

SNRM Żądanie budowy połączenia (*Set Normal Response Mode*),
 DISC Żądanie zerwania połączenia (*Disconnect*).

Możliwymi odpowiedziami stacji wtórnej na te komendy są:

I Wysłanie ramki z informacjami,
 RR Informacja, że odbiornik jest gotowy (*Receiver Ready*),
 RNR Informacja, że kanał odbiorczy nie jest gotowy (*Receiver Not Ready*),
 DM Połączenie zerwane (*Disconnect Mode*),
 UA Odpowiedź przy skutecznej budowie połączenia (*Unnumbered Acknowledgement*),
 FRMR Odrzucenie niezrozumiałej ramki HDLC (*Frame Error*).

Typowe szybkości transmisji w zastosowaniach technicznych wynoszą od 9.6 kbit/s do 2 Mbit/s. Protokół HDLC obejmuje cały szereg zabezpieczeń dla transmisji. Najpierw poprzednio wspomniany bit-*Stuffing* zapewnia fizyczne zabezpieczenie transmisji, gdyż błędne ramki są rozpoznawane już na wejściu i mogą być odrzucone. Liczniki ciągów nadawania i odbioru umożliwiają rozpoznawanie utraty danych lub wystąpienie błędu na odcinku transmisyjnym. Ponadto linie transmisyjne HDLC mają nadzorowanie czasowe. Jeżeli telegram nie zostanie potwierdzony w obrębie zadanego czasu, który zależy od szybkości transmisji, to telegram traktuje się jako stracony i przeprowadzana jest wcześniej zaprogramowana obsługa błędów.

SDLC jest protokołem firmy IBM, który opiera się ściśle na HDLC w synchronicznym trybie pracy.

3.4 Szeregowa transmisja asynchroniczna - UART

Jednym z najczęstszych modułów transmisji w obszarze sprzętu mikrokomputerowego jest *Universal Asynchronous Receiver and Transmitter* (UART). Służy on do szeregowej transmisji danych przez linię transmisyjną i przekształcania słów danych na/lub z równoległej magistrali komputera. Jak to już wyjaśnia sama jego nazwa chodzi tutaj o asynchroniczną transmisję danych, tzn. taką transmisję w której nie ma żadnego sygnału synchronizacyjnego dla nadajnika i odbiornika. Transmisja ta jest ukierunkowana na znaki i dlatego też synchronizacja musi być wykonywana od nowa przy każdym znaku.

Struktura znaku UART jest opisana w normie DIN 66022/66203. Składa się on z jedenastu bitów i rozpoczyna się jednym bitem startu, który jest logicznym zerem. Kończy się natomiast bitem stopu, który zawiera zawsze logiczną jedynkę.

Bit startu	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7	Parzystość	Bit stopu
	LSB							MSB		

Po bicie startu następuje przekazywana informacja jako bity danych. Po bitach danych i przed bitem stopu jest wysyłany jeden tzw. bit parzystości, który służy do zabezpieczenia danych i podaje, czy ilość logicznych jedynek w obrębie bitów danych jest parzysta (parzystość - logiczne zero) lub

44

nieparzysta (nieparzystość - logiczna jedynka). Liczbę bitów danych można nastawić na siedem lub osiem. Najmniej znaczący bit (LSB) jest przesyłany zawsze bezpośrednio po bicie startu, natomiast bit najbardziej znaczący (MSB) jako ostatni bit danych.

Możliwymi rodzajami transmisji są simplex, półduplex i pełny duplex. W trybie transmisji simplex dane mogą być transmitowane tylko w jednym kierunku. Dla odcinków, które pracują w trybie półduplex dane mogą być transmitowane w obu kierunkach, jednak nie w obu kierunkach równocześnie. Jeżeli przesyłanie danych odbywa się z pełnym duplex'em, to w każdej chwili możliwa jest asynchroniczna transmisja danych w obu kierunkach. Nadajnik umieszcza swoje znaki w pewnym wcześniej uzgodnionym przez obie stacje rastrze czasowym (siatce) do odcinka transmisyjnego. W odbiorniku przychodzące znaki są oceniane na podstawie nastawionej szybkości transmisji i struktury przysyłanych ramek.

Wszystkie parametry transmisji danych muszą być nastawione po obu końcach odcinka transmisyjnego. Szybkość transmisji jest nastawiana w sposób stopniowy. Zwykle są to: 150 bit/s, 300 bit/s, 600 bit/s, 1200 bit/s, 2400 bit/s, 4800 bit/s, 9600 bit/s, 19200 bit/s, 38400 bit/s i 64 kbit/s.

Maksymalna długość odcinka transmisyjnego zależy przede wszystkim od rodzaju sprzętu i szybkości transmisji.

3.5 Ochrona danych

Każda transmisja danych podlega zakłóceniom wskutek oddziaływań elektromagnetycznych, szumów, różnic potencjałów, starzenia się elementów itp. Zakłócenie oznacza odwrócenie bitów. Przeciwdziałanie zakłóceniom może być realizowane w dwojaki sposób:

1. Przez stosowanie przedsięwzięć technicznych, tzn. unikanie prawdopodobieństwa oddziaływania zakłóceń, np. przez ekranowany kabel, kabel światłowodowy, transmisję niezależną od potencjału.
2. Nadzorowanie transmisji informacji ze względu na błędy, a w przypadku wystąpienia błędu stosowanie różnego rodzaju środków zaradczych.

W dalszej części tego rozdziału, w oparciu o informacje zawarte w normie DIN 19 244 cz.10, będzie mowa o drugim sposobie przeciwdziałania zakłóceniom transmisji. Ta pochodząca z marca 1988 r. norma niemiecka jest identyczna z projektem normy międzynarodowej opracowanym przez *IEC Technical Committee 57*, który w najbliższym czasie powinien stać się standardem międzynarodowym.

3.5.1 Rodzaje błędów

Będą tutaj rozpatrywane kody ściśle transparentne (tzn. kody zorientowane na bity). Są to takie kody, w których dozwolona i sensowna jest każda kombinacja bitów. Dlatego też z samego ciągu bitów nie można wnioskować o ewentualnych błędach transmisji. Można przy tym wyróżnić trzy rodzaje błędów:

- błędy rozpoznawalne i możliwe do skorygowania,
- błędy rozpoznawalne i niemożliwe do skorygowania oraz
- błędy nie rozpoznawalne.

Jeżeli błąd został rozpoznany to można wtedy podjąć albo automatyczną korektę błędu (jednak taki przypadek występuje stosunkowo rzadko, np. w komunikacji satelitarnej), albo też powtórzyć transmisję sądząc, że błąd nie wystąpi już w tym samym miejscu co poprzednio.

3.5.2 Stopa błędów bitów

Stopa błędów bitów jest miarą podatności kanału transmisyjnego na zakłócenia:

$$p = \frac{\text{Ilość błędnych bitów}}{\text{Całkowita ilość przesłanych bitów}} \quad (3.1)$$

Najbardziej niekorzystną wartością, którą może przyjąć p jest $p=0.5$. Wówczas co drugi bit byłby wtedy zakłócony, a zatem informacja taka byłaby bezwartościowa (jeżeli byłoby $p=1$, to wystarczyłoby bity tylko odwrócić i miałyby się bezbłędną informację).

W technice wartością uzyskiwaną przy normalnych nakładach na transmisję danych jest $p=10^{-4}$. Jest to taka jakość transmisji, którą można w praktyce zaakceptować.

Zamiast stopy błędów można również podawać prawdopodobieństwo q odebrania bezbłędnego bitu:

$$q = 1 - p \quad (3.2)$$

3.5.3 Powtórzenie transmisji

Normalną reakcją na rozpoznaną błędną transmisję informacji, stosowaną powszechnie w takiej dziedzinie jaką jest transmisja danych, jest żądanie powtórzenia transmisji. Sposób ten jest znany pod określeniem ARQ (ang. *Error Detection with Automatic Request Repeat*). Wprawdzie również przy powtórnej transmisji pozostaje ta sama stopa błędów bitów, to jednak prawdopodobieństwo p^* wystąpienia zakłócenia w tym samym miejscu silnie maleje:

$$p^* = p^a, \quad (3.3)$$

p - stopa błędów bitów,

a - ilość transmisji.

3.5.4 Reszkowa stopa błędów

Reszkowa stopa błędów R odnosi się do nierozpoznanych błędów, które pozostają jeszcze po zastosowaniu strategii rozpoznawania błędów:

$$R = \frac{\text{Ilość nierozpoznanych kombinacji błędnych bitów}}{\text{Całkowita ilość } n \text{ przesłanych kombinacji bitów}} \quad (3.4)$$

Reszkowa stopa błędów R jest miarą integralności danych, tzn. miarą nienaruszalności danych. Z wartości R daje się obliczyć średni czas T pomiędzy dwoma nierozpoznanymi błędami:

$$T = \frac{n}{v R} \quad (3.5)$$

n - długość telegramu w bitach,

v - szybkość transmisji w bit/s.

Przykład z DIN 19244: telegramy z $n=100$ bitów są przesyłane bez przerw z szybkością transmisji 1200 bit/s. Kanał danych ma stopę błędów bitów $p=10^{-4}$. Uzyskiwane wg (3.5) średnie czasy T pomiędzy dwoma błędami wynoszą:

R wg (3.4)	T wg (3.5)	Typowe zastosowanie
10^{-6}	1 dzień	• systemy cyklicznie wymieniające dane
10^{-10}	26 lat	• transmisja sterowana wynikami
10^{-14}	260 000 lat	• sterowanie zdalne

3.5.5 Odległość Hamming'a

Odległość Hamming'a d jest miarą podatności kodu na zakłócenie (nazwa pochodzi od nazwiska Amerykanina R.W. Hamming'a).

Jeżeli e jest liczbą pewnie rozpoznanych błędów, to mamy:

$$d = e + 1 \quad (3.6)$$

Przykład: Jeżeli został rozpoznany pewnie jeden jedyny błąd (np. przy zastosowaniu bitu parzystości), a zatem $e=1$, to wg (3.6) odległość Hamming'a wynosi $d = 2$

Jest to najmniejsze wymaganie odnośnie do niezawodności podczas transmisji danych. W systemach *Fieldbus*'owych zwykle $d = 4$, a bardzo wysokie wymagania bezpieczeństwa spełniają warunki $d = 6$.

Odległość Hamming'a d można również połączyć z stopą błędów bitów p wg (3.1) i resztkową stopą błędów R wg (3.4) w następującej zależności:

$$d = \frac{\lg R(p_1) - \lg R(p_2)}{\lg p_1 - \lg p_2}, p_1 > p_2 \quad (3.7)$$

Przykład: do stopy błędów bitów $p_1=10^{-3}$ należy $R(p_1) = 10^{-4}$, a do stopy błędów bitów $p_2=10^{-4}$ należy $R(p_2) = 10^{-6}$.

Opisany przez tą kombinację kanał transmisyjny ma wtedy wg (3.7) odległość Hamming'a d :

$$d = \frac{\lg 10^{-4} - \lg 10^{-6}}{\lg 10^{-3} - \lg 10^{-4}}$$

$$d = 2.$$

3.5.6 Efektywność transmisji telegramów

Efektywność transmisji telegramów jest określana przez:

$$E = \frac{\text{Ilość bezbłędnych bitów}}{\text{Całkowita ilość przesłanych bitów}} \quad (3.8)$$

Efektywność tą można obliczyć w następujący sposób:

$$E = \frac{k q^n}{n} \quad (3.9)$$

k - ilość bitów informacji zawartych w telegramie,

q - prawdopodobieństwo odbioru bezbłędного bitu,

n - całkowita liczba wszystkich bitów dla telegramu, łącznie z bitami synchronizacyjnymi i bitami sprawdzania błędów.

Przykład 1: $k = 8$ bit,

$n = 11$ bit (1 bit startu + 8 bitów danych + 1 bit parzystości + 1 bit stopu),

$q = 1 - 10^{-3}$.

Wg (3.9) uzyskuje się:

$E = 0.72$.

Zatem efektywność transmisji telegramów wynosi 72%.

Przykład 2: $k = 8$ bit,

$n = 24$ bit (8 bitów startu + 8 bitów danych + 8 bitów CRC),

$$q = 1 - 10^{-3}.$$

Wg (3.9) uzyskuje się:

$$E = 0.32.$$

Zatem efektywność transmisji telegramów wynosi 32%.

W obu przypadkach przyjęto jednakową stopę błędów bitów p dla kanału transmisyjnego (mianowicie 10^{-3}), tak samo jak i ilość przesyłanych bitów informacji ($k=8$). W przykładzie 1-szym przewidziano 3 bity dla uniknięcia resztkowej stopy błędów R , natomiast w 2-gim przykładzie 16 bitów. Ogólnie można stwierdzić, że efektywność transmisji E i resztkowa stopa błędów R są w odwrotnej relacji względem siebie. Im bardziej niezawodna powinna być transmisja, tym gorsza jest efektywność transmisji.

3.5.7 Wybrane strategie rozpoznawania błędów

Koncepcja każdej ze strategii rozpoznawania błędów jest zawsze taka sama i wymaga:

- opracowania strategii, która rozpozna błędy,
- skorygowania rozpoznanych błędów (automatycznie lub przez ARQ),
- dokonanie kwantyfikacji nie rozpoznanych błędów resztkowych przez obliczenie prawdopodobieństwa.

Poniżej zostaną opisane niektóre strategie rozpoznawania błędów.

3.5.7.1 Bit parzystości

Jeżeli przesyłany jest znak ASCII o 7-miu bitach, np. 1 0 1 0 1 1 1, to tworzy się sumę kontrolną i stwierdza, że jest ona nieparzysta. Jeżeli nadajnik i odbiornik uzgodniły ze sobą parzystość sumy kontrolnej, to wstawiany jest bit parzystości $P=1$ i jest on również transmitowany:

1 0 1 0 1 1 1 $P=1$, parzystość.

Jeżeli wystąpi błąd (tzn. jeden z bitów wskutek wystąpienia zakłócenia zostanie odwrócony), to nie ma już parzystości i dzięki temu błąd jest rozpoznawany:

1 0 0 0 1 1 1 $P=1$, nieparzystość, a zatem wystąpił błąd transmisji.

Jednak jeżeli wystąpią dwa błędy, to nie zostaną rozpoznane, ponieważ się one wzajemnie kompensują, podobnie jak przy 4, 6 lub 8 błędach.

Natomiast wystąpienie 3, 5, 7 błędów zostanie rozpoznanych, ale nie będzie można ich odróżnić od wystąpienia jednego błędu. Pewne jest zatem rozpoznanie $e=1$ błędu i dlatego też odległość Hamming'a wg (3.6) w tej metodzie wynosi $d=2$.

3.5.7.2 Zabezpieczenie blokowe

W przypadku blokowego zabezpieczenia transmisji danych przesyłanych jest kolejno 7 znaków ASCII wraz z bitem parzystości. Następnie występuje 8-my oktet P , który zawiera informację o parzystości w kolumnach.

	1.	2.	3.	4.	5.	6.	7.	P.
1.	0	1	0	1	0	1	1	0
2.	1	1	1	0	0	0	1	0
3.	1	0	0	1	0	0	1	1
4.	0	0	1	1	0	0	1	1
5.	1	1	0	0	1	1	1	1
6.	0	0	1	1	0	0	1	1
7.	1	1	0	0	0	1	1	0
P	0	0	1	0	1	1	1	0

Jeżeli 1 bit zostanie zniszczony, tzn. ulegnie inwersji, to występuje naruszenie parzystości, np. w wierszu 4 i kolumnie 5. Błąd jest jednak rozpoznawany i można go skorygować.

	1.	2.	3.	4.	5.	6.	7.	P
1.	0	1	0	1	0	1	1	0
2.	1	1	1	0	0	0	1	0
3.	1	0	0	1	0	0	1	1
4.	0	0	1	1	1	0	1	1
5.	1	1	0	0	1	1	1	1
6.	0	0	1	1	0	0	1	1
7.	1	1	0	0	0	1	1	0
P	0	0	1	0	1	1	1	0

Jeżeli zostaną zniszczone 2 bity, to występuje uszkodzenie parzystości, np. w kolumnie 5 i w wierszach 4 i 5. Kolumna 5 nie zgłasza żadnego meldunku parzystości. Są zatem rozpoznawalne 2 błędy, ale żaden nie jest lokalizowany i tym samym również korygowany.

	1.	2.	3.	4.	5.	6.	7.	P
1.	0	1	0	1	0	1	1	0
2.	1	1	1	0	0	0	1	0
3.	1	0	0	1	0	0	1	1
4.	0	0	1	1	1	0	1	1
5.	1	1	0	0	0	1	1	1
6.	0	0	1	1	0	0	1	1
7.	1	1	0	0	0	1	1	0
P	0	0	1	0	1	1	1	0

Teraz rozpatrzmy przypadek wystąpienia trzech błędów w niekorzystnej kombinacji:

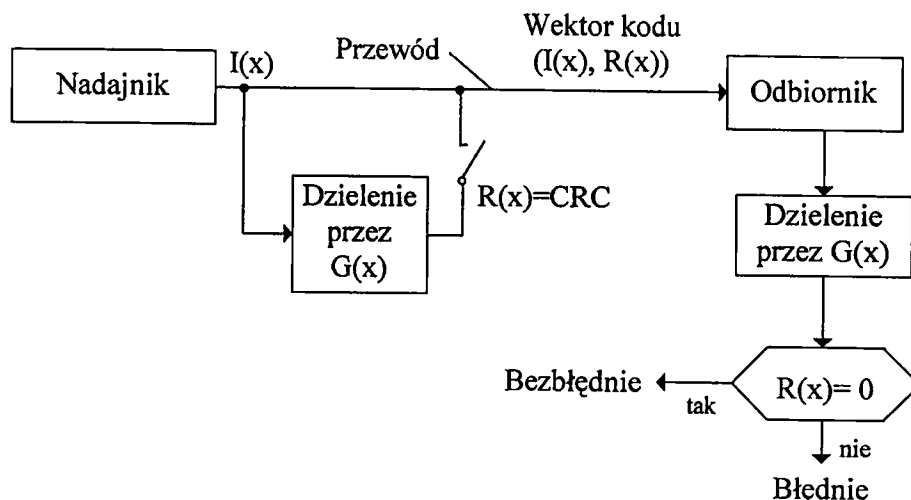
	1.	2.	3.	4.	5.	6.	7.	P
1.	0	1	0	1	0	1	1	0
2.	1	1	1	0	0	0	1	0
3.	1	0	0	1	0	0	1	1
4.	0	0	1	1	1	1	1	1
5.	1	1	0	0	0	1	1	1
6.	0	0	1	1	0	0	1	1
7.	1	1	0	0	0	1	1	0
P	0	0	1	0	1	1	1	0

Wprawdzie sygnalizacja błędów tutaj występuje, ale jest ona w złym miejscu. Nie jest zatem możliwa korekcja błędów, ponieważ byłaby ona przeprowadzona w złym miejscu. Wystąpienie

czterech błędów nie zostanie rozpoznane, tak samo jak 5, 6 itd. Zatem na pewno są rozpoznawane $e=3$ błędy i dlatego też wg (3.6) odstęp Hamming'a wynosi $d=4$.

3.5.7.3 CRC

Z samego określenia CRC (*Cyclic Redundancy Check*) nie wynika, jak działa ta często wymagająca znacznych nakładów metoda korygowania błędów. Traktuje się w niej informację niezależnie od jej długości, jako liczbę, która jest dzielona w nadajniku przez inną stałą liczbę, tzw. generator wielomianu $G(x)$ (rys.3.1).



Rys.3.1 Rozpoznawanie błędów przez zastosowanie CRC

Iloraz Q odrzuca się, a wypadkową resztę R doczepia się do informacji I i tak powstały wektor kodu IR wysyła się. Odbiornik dzieli wektor kodu przez ten sam wygenerowany wielomian G i przy bezbłędnej transmisji otrzymuje się resztę $R=0$.

Dzielenie podlega prostym regułom: $1 + 1 = 0$

$$0 - 1 = 1$$

$$1 - 1 = 0$$

$$0 + 1 = 1$$

Te reguły są realizowane sprzętowo przez działanie EXOR. Na rys.3.2 pokazano przykład w układzie dziesiętnym i binarnym. Generatory wielomianu G są opisywane w różny sposób:

Jako zależność: $G = x^7 + x^6 + x^5 + x^2 + x^1$

Jako liczba binarna: $G = 11\ 100\ 101$

Jako liczba ósemkowa: $G = 345$

Długość wielomianu, oprócz innych właściwości, jest miarodajna dla uzyskania odległości Hamming'a. Zaproponowane w DIN 19 244 wielomiany mają następującą postać:

Postać wielomianu	HD d	Format	i = 2	i = 3	i = 4
345	4	FT2			
235 545	6	FT3	26	94	304

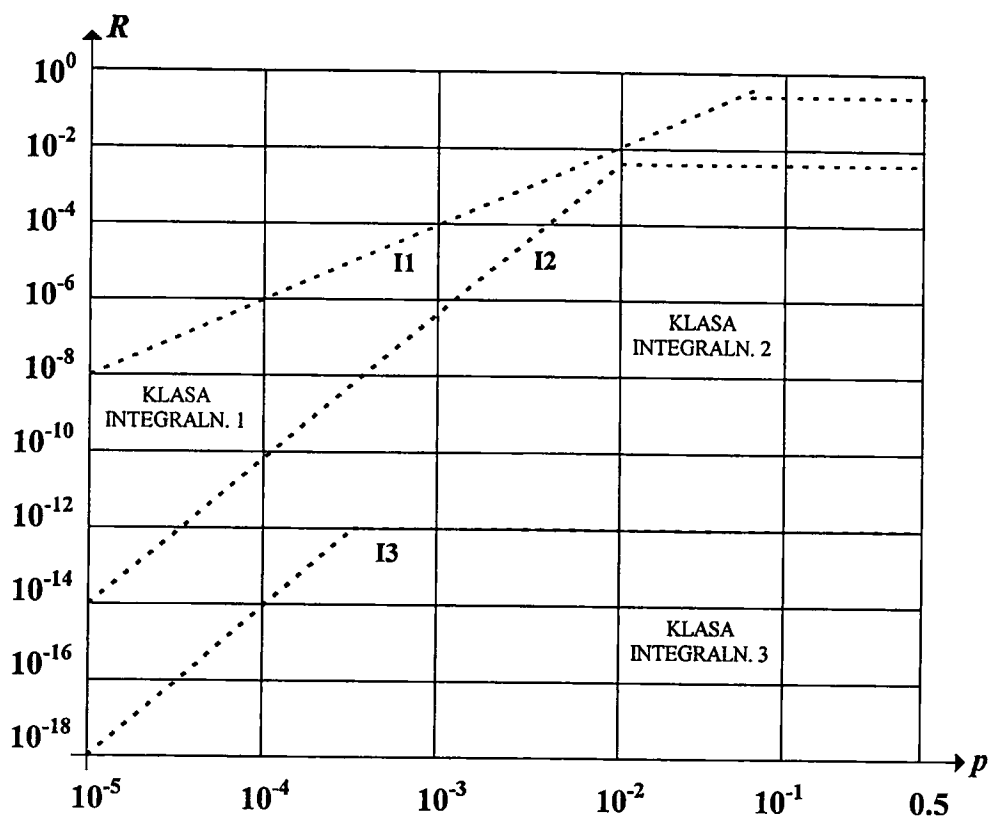
CRC (Cyclic Redundancy Check)	
1. Niech informacja będzie I:	
Przykład dziesiętnie (d): I=14	binarnie (b): I=110101
2. Wielomian kontrolny niech będzie G:	
$G = \dots a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 x^0$	
d: $\dots 0 x + 3 x^0 = 3$	b: $\dots 1 x^3 + 0 x^2 + 1 x + 1 x^0 = 1011$
3. Wektor pomocniczy jest wtedy $H = 100 \dots 0$:	
d: H = 10	b: H = 10000
4. Informacja jest mnożona przez H:	
$B = I H$	
d: B = 140	b: B = 1101010000
5. Wynik B jest dzielony przez G:	
$\frac{B}{G} = Q + \frac{R}{G}$	
d: $\frac{B}{G} = \frac{140}{3} = 46 + \frac{2}{3}$	b: $\frac{B}{G} = \frac{1101010000}{1011} = 1111011 + \frac{101}{1011}$
6. Reszta R jest dodawana i przesyłana jest całość:	
d: B + R = 142	b: B + R = 1101010101
7. Odbiornik tworzy:	
$(B - R) : G$	
d: $(B - R) : G = 138 : 3$ = 46, R = 0	b: $(B - R) : G$ = $(B + R) : G = 1101010101 : 1011$ = 1111011, R = 0
8. R = 0 oznacza bezbłędną transmisję	

Rys.3.2 Zasada korygowania błędów transmisji metodą CRC ($1 + 1 = 0$, $0 - 1 = 1$)

Na rysunku rys.3.1 dla zoptymalizowanego generatora wielomianu $G = 235\ 545$ pokazano, że nie są rozpoznawane wszystkie błędy wzorcowych bitów (dla przypadku 6 losowo błędnych bitów ilość nie rozpoznanych błędów dla $i = 2, 3$ i 4 przesłanych łańcuchów danych można wziąć np. z dodatku do normy DIN 19 244). Również odległość Hamming'a $d=6$ jest gwarantowana tylko do 16 kolejno następujących po sobie łańcuchów danych.

3.5.8 Klasy integralności danych

Na rysunku 3.2 pokazano 3 klasy integralności danych I1, I2, I3 określone w normie DIN 19 244. Na osi rzędnych naniesiona jest stopa błędów bitów p kanału transmisyjnego, natomiast na osi odciętych resztkowa stopa błędów R uzyskiwana dla określonego kodu. Naniesione trzy krzywe I1, I2 i I3 reprezentują klasy integralności danych, które ta norma zaleca dla różnych wymagań bezpieczeństwa transmisji.



Rys.3.2 Definicja klas integralności danych wg DIN 19 244

Z przebiegu krzywych na rys.3.3 wynika, że klasa integralności I1 zapewnia najniższą miarę pewności transmisji, tzn. najniższą integralność, natomiast I3 najwyższą. Nachylenia zboczy odpowiadają zależności (3.7), tzn. że nachylenie zbocza odpowiada odległości Hamming'a dla określonej metody kodowania.

3.5.9 Formaty telegramów

Klasa integralności danych tak długo pozostaje teoretycznym terminem, jak długo nie stoi za nią żaden format telegramu ze swoją specyficzną metodą zabezpieczenia danych. Poniżej zostanie rozpatrzonych kilka znanych formatów telegramów.

3.5.9.1 Telegram z bitem parzystości

Szeroko rozpowszechniony jest format telegramu w następującej postaci:

Bit startu, 7 bitów znaków ASCII, bit parzystości, bit stopu

i jest on określany jako format UART (ang. *Universal Asynchronous Receiver / Transmitter*).

Format ten może mieć również postać:

Bit startu 0, 8 bitów informacji, prosty bit parzystości, bit stopu 1.

Jest on sprzętowo kompatybilny z wszystkimi komputerami PC i ich urządzeniami peryferyjnymi.

Oznaczenie wg normy DIN 19 244: *Kod-(11,8)*

Klasa formatu wg DIN 19 244: *FT 1.1*

Odległość Hamming'a: $d = 2$

Ilość pozostawionych nierozpoznanych wzorców błędów (= błędy resztkowe) wynosi:

$$A_e = \binom{m}{n}, \quad (3.10)$$

przy czym: $e = 2, 4, 6, 8$, ilość równocześnie występujących zakłóconych bitów,
 $m = 9$, ilość bitów danych, łącznie z bitem parzystości.

Z zależności (3.10) dla kodu (11,8) wynika:

$$A_2 = 36,$$

$$A_4 = 126,$$

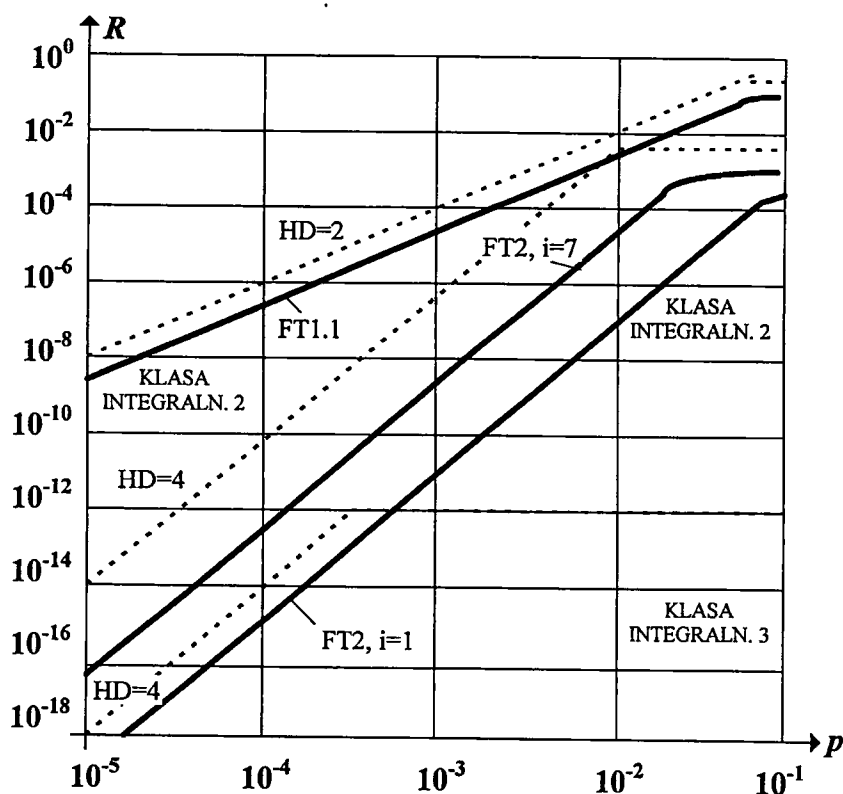
$$A_6 = 84,$$

$$A_8 = 9.$$

Oznacza to np., że istnieje 126 kombinacji bitów, gdzie występuje $e = 4$ zniszczone bity równocześnie, które się wzajemnie znoszą, a zatem parzystość nie ulega zmianie.

Z liczby A błędu resztkowego i stopy błędów bitów p wzgl. $q = 1 - p$ daje się obliczyć resztkową stopę błędów R :

$$R_{(p)} = \sum_e A_e p^e q^{m-e} \quad (3.11)$$



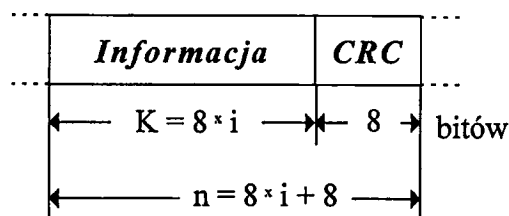
Rys.3.3 Resztkowa stopa błędów R dla różnych klas formatów danych

Na wykresie (rys.3.3) zależność (3.11) jest naniesiona jako najwyżej położona krzywa [1]. Można zauważyć, że telegram z bitem parzystości spełnia klasę integralności danych I1, a efektywność transmisji E wynosi wówczas:

$$E = 72\%.$$

3.5.9.2 Telegram z CRC

Ten format telegramu jest stosowany szczególnie wtedy, gdy powinno być przekazywanych kolejno po sobie wiele oktetów danych. Ma on postać pokazaną na rys.3.4.



Rys.3.4 Telegram o formacie klasy FT2 z 8-mioma bitami kontrolnymi

Oznaczenie wg DIN 19 244: *kod* ($8i+8$, $8i$); $i = 1 \dots 15$

Klasa formatu wg DIN 19 244: *FT2*

Odległość Hamming'a: $d = 4$

Generator wielomianu: $x^7 + x^6 + x^5 + x^2 + 1$

Znaki kontrolne są uzupełniane zwykłym bitem parzystości i odwracane (inwersja).

Dla ilości A pozostałych nierozpoznanych wzorców błędów zgodnie z (1.10) [1] mamy:

$$A_e \cong \frac{1}{128} \binom{m}{n}, \quad (3.12)$$

przy czym: $e = 4, 6, 8, \dots n$

n - całkowita ilość przesłanych bitów.

Przykład: należy przesłać tylko 1 oktet danych, a zatem $i = 1$, $n = 16$. Z (3.12) dla ilości pozostałych nierozpoznanych błędów $e = 4$ wynikają równocześnie zakłócone bity:

$$A_{16,4} = 14,2.$$

A zatem jest to prawie 10-krotnie mniej nierozpoznanych błędów niż przy kontroli parzystości, co świadczy o skuteczności działania CRC.

Z liczby A błędów reszkowych i stopy błędnych bitów p wzgl. $q = 1 - p$ daje się z (3.11) obliczyć reszkowa stopa błędów $R(p)$. Funkcja $R(p)$ jest naniesiona na rys.3.2 dla dwóch przypadków:

$i = 1$ (tylko jeden oktet danych) oraz

$i = 7$ (7 oktetów danych) [1].

Można zauważyć, że telegram z 8-mio pozycyjnym wielomianem generatora należy do klasy integralności I2.

Efektywność transmisji E dla oktetu danych $i = 1$ wynosi:

$$E_1 = 32\%.$$

Dla 7 oktetów danych ($i = 70$) uzyskuje się (3.9):

$$E_7 = 81\%.$$

Widać zatem, że transmisja danych w postaci kodu (64, 56), a zatem dla 7 oktetów danych jest znacznie bardziej efektywna niż dla 1 oktetu.

Literatura

[1] Norma DIN 19 244, cz. 10 (IEC 57 Sec)

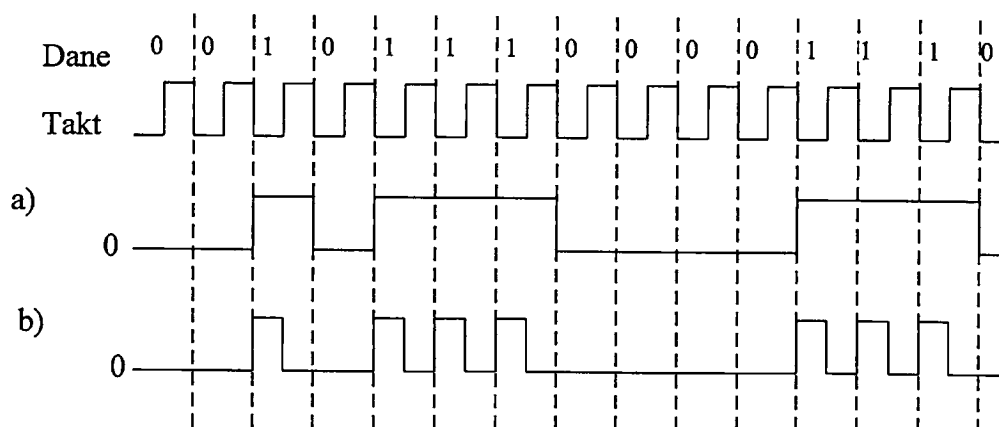
3.6 Binarne zobrazowanie informacji

Niezależnie od formatu telegramu muszą być kodowane wszystkie stany cyfrowe. Informacja cyfrowa może być zawarta w amplitudzie, zboczu, fazie i częstotliwości. Jeżeli została wybrana synchroniczna transmisja danych, jak to ma miejsce dla formatów telegramów HDLC i SDLC, to sensowne jest zastosowanie takiego kodowania, które zawiera informacje o takcie. Tym samym odpada konieczność stosowania dodatkowego przewodu taktującego. Jeżeli kodowanie nie zawiera składowych stałych, to możliwa jest transmisja danych przez przewody zasilające w energię, o ile jest ono realizowane prądem stałym. Odnośnie do tych kryteriów należy rozpatrzyć omówione niżej możliwości binarnego zobrazowania informacji.

3.6.1 NRZ, RZ

Najczęściej wykorzystywanym sposobem modulacji jest metoda NRZ (*Non Return to Zero*), która jest pokazana na rys.3.5 a. Sygnały logicznego „0” i „1” są zobrazowane przez różne amplitudy, przy czym podczas taktowania nie występuje żadna zmiana potencjału. Są przy tym następujące możliwości zobrazowania stanów logicznych:

- masa i potencjał dodatni,
- potencjał ujemny i masa,
- wartość bezwzględna potencjału ujemnego i dodatniego.



Rys.3.5 Kodowanie bez taktu zawierające składową stałą: a) NRZ, b) RZ

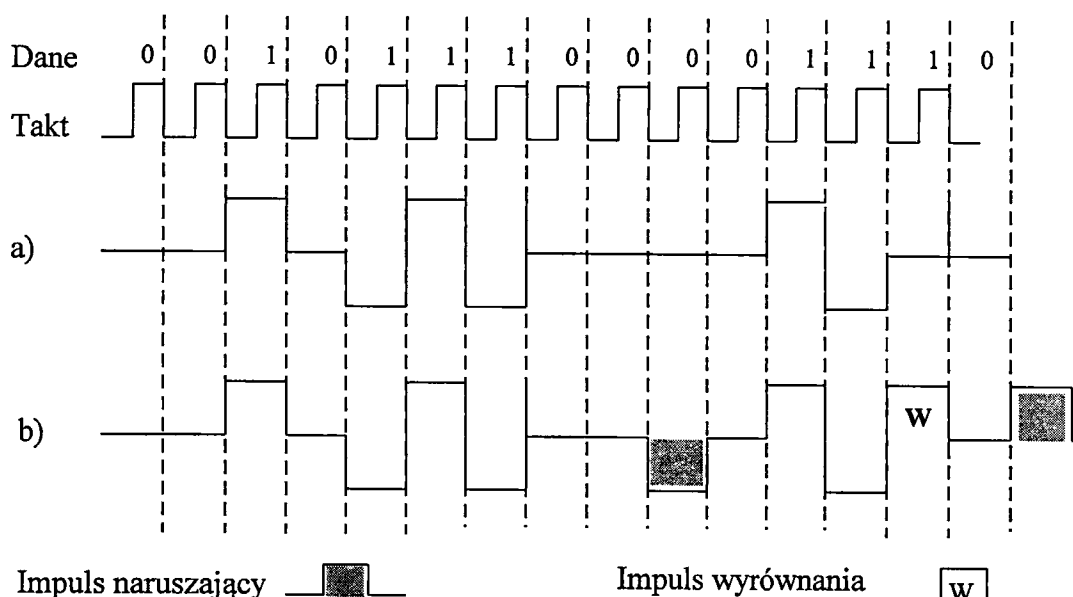
Dla metody NRZ ogólnie słuszne jest to, że nie są w niej zawarte informacje taktujące oraz nie jest ona niezależna od składowej stałej sygnału, również wtedy gdy pracuje się z potencjałami ujemnymi i dodatnimi.

Metoda RZ (*Return to Zero*) (rys.3.5 b) od metody NRZ różni się tylko tym, że poziom sygnału przy przesyłaniu „1” podczas taktu powraca do stanu początkowego. Również tutaj informacje taktujące nie są zawarte w przesyłanym sygnale, ponieważ przy długim ciągu „0” nie występują żadne impulsy.

3.6.2 Kodowanie bipolarne i kodowanie HDBn

Aby ukształtować zobrazowanie informacji w formacie NRZ w sposób niezależny od składowej stałej, można logiczną „1” przedstawić z potencjałami o zmiennej polaryzacji. W ten sposób uzyskuje się kodowanie pokazane na rys.3.6 a. Dzięki alternatywnym polaryzacjom to kodowanie jest niezależne od składowych stałych, a pomimo tego informacje o takcie nie są zawarte w sygnale. Wstawienie informacji o takcie można uzyskać przez to, że po n kolejnych logicznych „0” jest wysyłany impuls naruszający. Tym samym uzyskuje się format HDBn (ang. *High Density Bipolar Format*). Na rys.3.6 b jest pokazana metoda kodowania HDB₂.

Impuls naruszający charakteryzuje się tym, że ma on jednakową polaryzację z ostatnio przesłaną „1”. Tym samym zapewnione jest to, że najpóźniej po n taktach wysyłany jest jeden impuls i w ten sposób można odtworzyć informację taktującą. Może się jednak zdarzyć tak, że wszystkie impulsy naruszające mają jednakową polaryzację. Tym samym kodowanie nie jest już niezależne od składowej stałej.



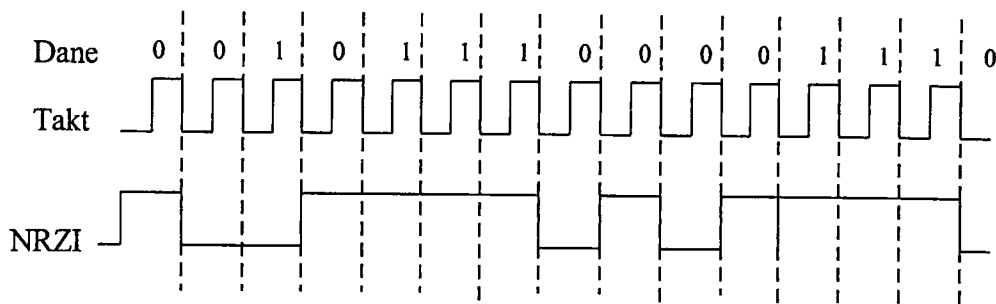
Rys.3.6 Kodowanie bez składowej stałej: a) bipolarne bez taktu, b) kodowanie HDB₂ z taktom

Wstawiając tzw. impulsy wyrównujące można ponownie uniezależnić się od składowej stałej. Jeżeli po ostatnim impulsie naruszającym jest przesłana parzysta ilość „1” i konieczny nowy impuls naruszający, to impuls wyrównujący jest przesyłany z odwrotną polaryzacją do ostatniej „1” i impuls naruszający z jednakową polaryzacją jak impuls wyrównujący.

3.6.3 NRZI

W metodzie NRZI (*Non Return to Zero Insert*) logiczne „0” jest przedstawione w taki sposób, że na początku taktu polaryzacja ulega zmianie. Dla logicznej „1” ta zmiana polaryzacji nie występuje (rys.3.7). Ten sposób kodowania nie jest niezależny od składowej stałej. Jeżeli, tak jak np. w Bitbus’ie, po zdefiniowanej ilości przesłanych „1” nadajnik wstawia automatycznie jedno „0”,

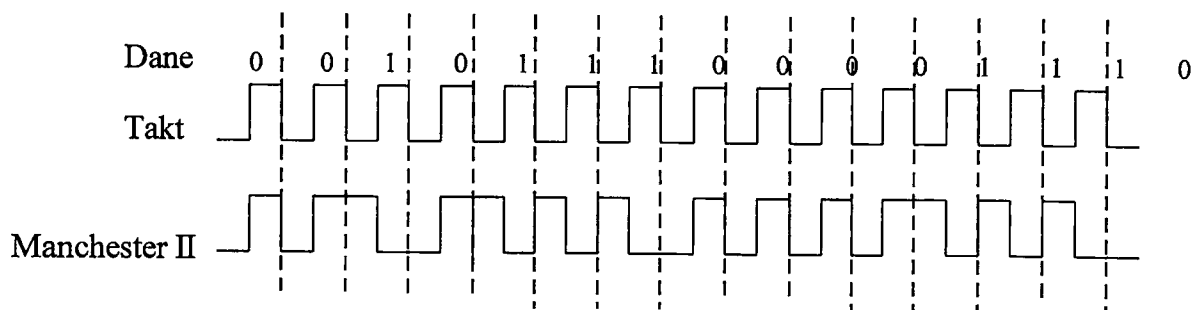
to wówczas zawarta jest informacja o takcie. Odbiornik musi to dodatkowe „0” usunąć, aby uzyskać właściwą informację.



Rys.3.7 Kodowanie metodą NRZI ze składową stałą

3.6.4 Kodowanie Manchester II

W kodowaniu Manchester II informacja jest zawarta w długości fazy sygnału, tzn. jeżeli w środku taktu występuje dodatnie zbocze to chodzi o „0”, a jeżeli zbocze jest ujemne, to chodzi o „1” (rys.3.8). Jeżeli stosowany jest sygnał bipolarny, to jest on niezależny od składowych stałych. Wskutek tego, że w środku taktu musi zawsze wystąpić zbocze, to informacja o takcie jest już zawarta w sygnale. Problemатyczne jest jednak to, że maksymalnie występująca częstotliwość jest jednakowa częstotliwości taktowania wzgl. dwukrotnie taka jak maksymalnie występująca częstotliwość w metodzie NRZ. Wskutek tego odbicia zakłócające są stosunkowo duże.



Rys.3.8 Kodowanie Manchester II

3.6.5 FSK, ASK, PSK

Rozpatrywane dotychczas metody kodowania wykorzystują sygnały prostokątne. W omawianych niżej metodach stosowane są natomiast sygnały sinusoidalne. Ma to tę zaletę, że sygnały nie mają „nadfal” i tym samym są ograniczone w swojej szerokości pasma.

W metodzie FSK (ang. *Frequency Shift Keying*) „0” i „1” są zobrazowane przez różnorodne częstotliwości. Ten rodzaj transmisji jest nazywany *Carrierband*.

Metoda ASK (ang. *Amplitude Shift Keying*) różni się od metody FSK tym, że jeden z obu stanów logicznych jest przedstawiony przez częstotliwość $f=0$ Hz. Oznacza to, że jeden stan logiczny jest zobrazowany przez istnienie amplitudy, natomiast drugi przez brak amplitudy.

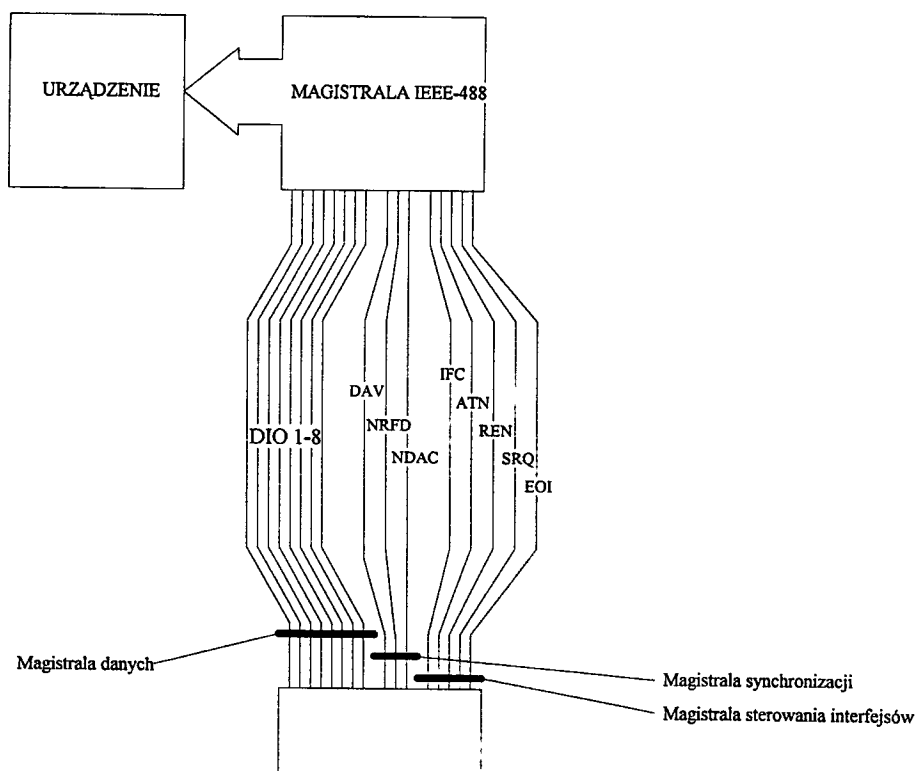
W metodzie PSK (ang. *Phase Shift Keying*) „1” jest przedstawiana przez skok fazy na początku taktu, a „0” przez brak skoku amplitudy. W metodzie PSK (ang. *Phase Shift Keying*) „1” jest przedstawiana przez skok fazy na początku taktu, a „0” przez brak skoku fazy.

Dzięki sinusoidalnemu kształtowi sygnału wszystkie te możliwości są niezależne od składowej stałej. Informacje o takcie w metodach FSK i PSK są zawarte w sygnale, ponieważ musi być podana zależność pomiędzy częstotliwością transmisji i taktem. W metodzie ASK informacja o takcie jest tracona wówczas, jeżeli powstają długie ciągi, w których nie jest przesyłany żaden sygnał.

3.7 Interfejs IEC

Na początku lat '60 firma Hewlett-Packard opracowała standard komunikacji, głównie dla wsparcia komputerowej techniki pomiarowej. System ten określony jako GPIB (ang. *General Purpose Interface Bus*), został zaakceptowany przez gremia normalizacyjne jako standard IEC 625. Standard IEC 625 różni się jedynie typem gniazda połączeniowego od aktualnego wtedy IEEE-488.

W systemie pracującym w standardzie IEEE-488 urządzenia sterujące, rejestrujące oraz przyrządy pomiarowe są przyłączone równolegle do magistrali, której zadaniem jest przesyłanie rozkazów i instrukcji organizujących pracę systemu, a także informacji o stanie urządzeń oraz wyników pomiarowych. Taka konfiguracja systemu interfejsu umożliwia bezpośrednią współpracę między dowolną parą urządzeń dołączonych do magistrali lub przesyłanie informacji z dowolnego urządzenia do kilku innych jednocześnie.



Rys.3.9 Struktura magistrali IEEE-488

Magistrala IEEE-488 składa się z 16 linii, które można podzielić na 3 grupy funkcjonalne (rys.3.9):

- *magistrala danych* – zawiera 8 linii (*Data Input-Output lines, DIO lines*), służy do szeregowego przesyłania bajtów (słów 8-bitowych). Transmisja może zachodzić w obu kierunkach i odbywa się w sposób asynchroniczny, przy czym każdy następny bajt pojawia się na liniach DIO dopiero

wtedy, gdy zostanie potwierdzony odbiór poprzedniego za pomocą sygnałów na liniach synchronizacji transmisji

- *magistrala synchronizacji (handshake's)* – jest złożona z trzech linii DAV, NRFD oraz NDAC. Linie te nazywane są liniami kontroli transmisji, które uniemożliwiają niezakłócone asynchroniczne przesyłanie informacji na liniach DIO ze zwrotnym potwierdzeniem odbioru. Wymóg zwrotnego potwierdzenia odbioru powoduje automatyczne dostosowanie szybkości transmisji do najwolniejszego z odbiorców.
 - DAV – komunikaty informujące odbiorców, czy informacje na liniach DIO są aktualne,
 - NRFD – komunikaty informujące nadawcę, czy odbiorcy są gotowi do odebrania informacji znajdujących się na liniach DIO,
 - NDAC – komunikaty informujące nadawcę potwierdzające przyjęcie informacji,
- *magistrala sterowania* – składająca się z pozostałych pięciu linii. Są to linie ATN, IFC, REN, SRQ, EOI nazywane liniami sterowania interfejsów:
 - ATN – uwaga; umożliwia interfejsom przyrządów rozróżnienie, czy informacja znajdująca się w danej chwili na liniach DIO jest instrukcją, adresem, czy rozkazem (ATN=1 linia w stanie przesyłania instrukcji), czy też przesyłane są dane dotyczące wyników pomiarów lub stanu przyrządu (ATN=0),
 - IFC – zerowanie interfejsu; służy do przesyłania zerującego, tzn. takiego, który wprowadza interfejsy wszystkich urządzeń dołączonych do magistrali w określony stan początkowy,
 - REN – umożliwienie zdalnego sterowania; służy do przełączania przyrządów na zdalne sterowanie, dokonywane z magistrali. Tym samym, w razie potrzeby, może być wyłączone ręczne sterowanie układem pomiarowym.
 - SRQ – żądanie obsługi; sygnalizuje kontrolerowi, że jedno lub więcej urządzeń przyłączonych do magistrali żąda obsługi, np. po to, żeby przesłać wynik pomiaru. Urządzenie żądające obsługi ustawia linię SRQ w stan SRQ=1. Sygnał taki może być wykorzystywany do przerywania pracy kontrolera w celu odebrania wyniku.
 - EOI – koniec / identyfikacja; pełni w systemie pomiarowym dwie funkcje w zależności od stanu linii ATN. Jeśli ATN=1, to przejście linii EOI w stan EOI=1 oznacza, że kontroler wykonuje tak zwaną kontrolę równoległą, natomiast, gdy ATN=0, to linia EOI może być użyta przez dowolnego nadawcę informacji do zasygnalizowania bajtu kończącego transmisję informacji.

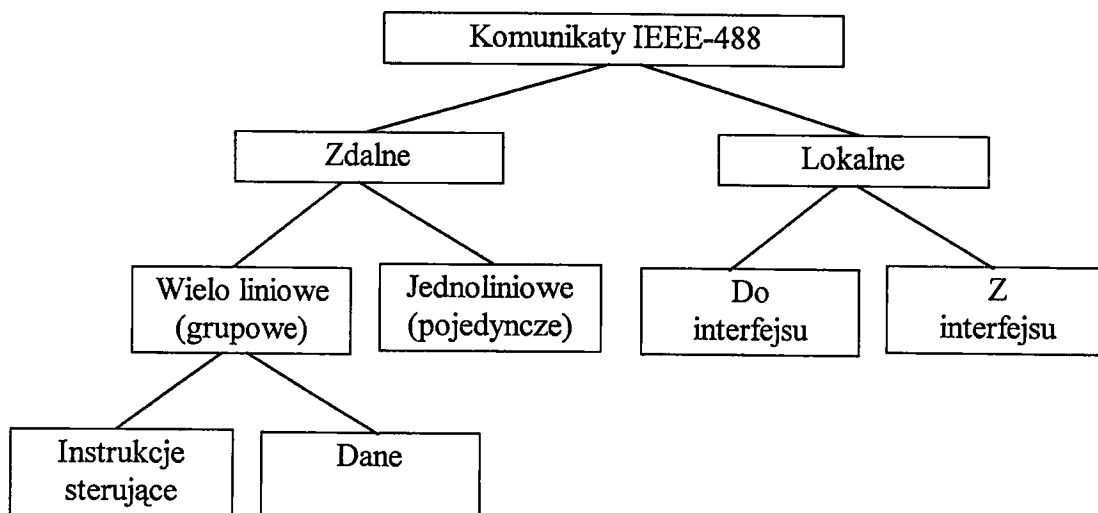
W standardzie IEEE-488 nie są narzucane żadne warunki na urządzenie właściwe i realizowane przez nie funkcje, natomiast ściśle określone normą są funkcje interfejsowe, których jest 10. Są to:

- inicjator transmisji,
- akceptor transmisji – funkcje te odpowiedzialne są za właściwy przebieg cyklu synchronizacji,

- nadajnik – układ realizujący tę funkcję umożliwia wysyłanie informacji po liniach DIO,
- odbiornik – układ realizujący tę funkcję umożliwia odbiór informacji,
- żądanie obsługi – przesyłanie do kontrolera informacji o tym, że istnieje urządzenie, które żąda obsługi programowej,
- zerowanie urządzenia – realizacja tej funkcji umożliwia wprowadzenie urządzenia w stan początkowy,
- wyzwolenie urządzenia – układ realizujący tę funkcję służy do inicjacji pomiaru przez urządzenie sterujące systemem,
- zdalne / lokalne – przełączenia przyrządów pomiarowych na sterowanie ręczne, bądź zdalne,
- kontrola równoległa – umożliwienie wysłania przez interfejs informacji o stanie funkcji urządzenia przyrządu,
- kontroler – sterowanie systemem pomiarowego za pośrednictwem magistrali IEEE-488,

System IEEE-488 umożliwia stosowanie funkcji z rozszerzonym adresem, przy czym można go również różnie konfigurować w zależności od wymaganego poziomu kontroli (system może pracować bez kontrolera, z jednym kontrolerem lub z wieloma).

Wszystkie rodzaje informacji systemu nazywają się komunikatami (rys.3.10).



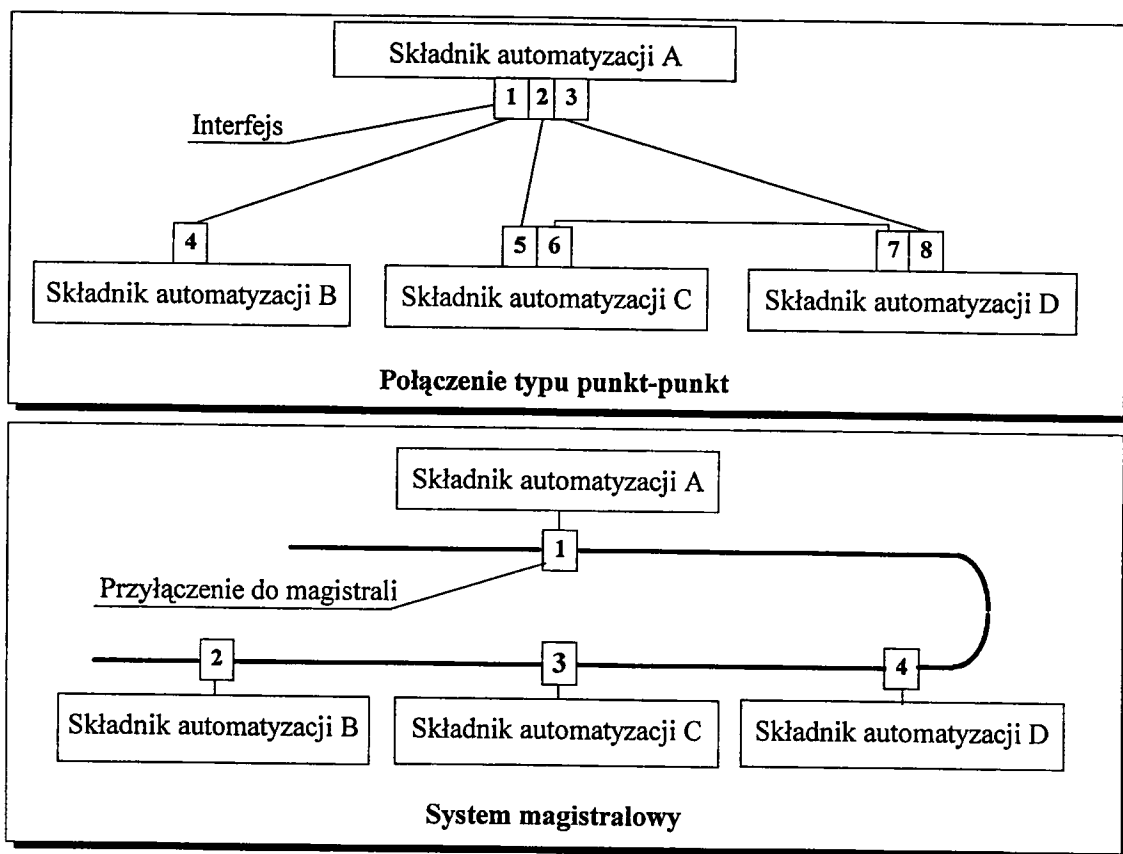
Rys.3.10 Podstawowa klasyfikacja komunikatów standardu IEEE-488

4. Protokoły warstwy łącza danych

4.1 Topologia sieci

4.1.1 Wstęp

Stacje robocze, serwery, sterowniki i inne urządzenia występujące w sieci komputerowej muszą być ze sobą połączone za pomocą nośnika informacji - okablowania (rys.4.1). Fizyczny układ sieci komputerowej jest nazywany jej topologią. Pod pojęciem topologii sieci rozumie się jednak, po pierwsze geometryczne przyporządkowanie uczestników w sieci, a po drugie logiczne przyporządkowanie tych uczestników niezależnie od geometrii.



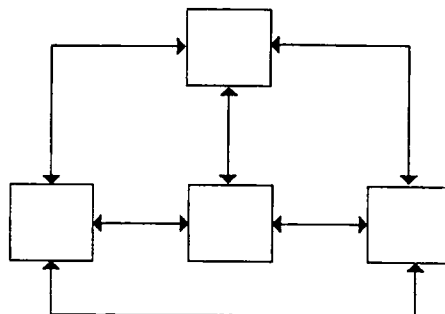
Rys.4.1 Podstawowe rodzaje połączeń sieci na poziomie fizycznym

4.1.2 Połączenia dwu - punktowe

Najprostsza możliwość wymiany danych polega na tym, że dwóch partnerów komunikacyjnych jest ze sobą połączonych za pomocą przewodów, jak np. w instalacji do rozmów, modemie lub połączeniu komputera PC i drukarki. Sterowanie procesem komunikacyjnym jest wówczas realizowane za pomocą przewodów sterujących, potwierdzeń i taktu (tryb pracy „handshake”).

Jeżeli za pomocą połączeń dwu-punktowych (*peer to peer*) jest powiązanych ze sobą wielu uczestników, to powstaje sieć „oczkowa” (rys.4.2). W tej topologii pomiędzy dwoma komunikującymi się partnerami powstaje połączenie dwu-punktowe. Dla n uczestników

wymaganych jest $(n-1)$ interfejsów dla uczestnika oraz $(n/2)$ przewodów połączeń. Wynika z tego, że koszty takiej topologii sieci są bardzo wysokie. W przypadku wystąpienia błędu transmisji odpadałby jednak tylko jeden uczestnik, albo też tylko jeden kanał komunikacyjny, a diagnostyka takiej sieci byłaby bardzo prosta.



Rys.4.2 Komunikacja w sieci „oczkowej”

W przypadku połączenia typu punkt-punkt podstawowymi cechami charakterystycznymi sieci są:

- wymiana danych jest możliwa tylko pomiędzy dwoma bezpośrednio połączonymi urządzeniami,
- dla każdego połączenia konieczne są dwa interfejsy,
- nakłady na okablowanie są bardzo duże,
- występuje ciągłe połączenie pomiędzy dwoma partnerami.

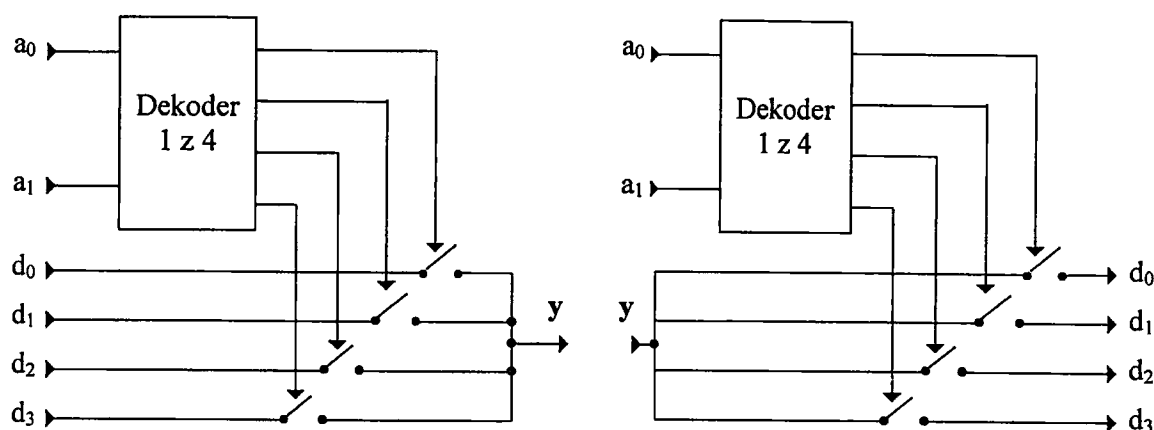
Natomiast dla sieci z systemem magistralowym charakterystyczne jest to, że:

- wymiana danych jest możliwa pomiędzy dowolnymi urządzeniami,
- dla każdego urządzenia automatyzacji wymagane jest tylko jedno przyłączenie do magistrali,
- nakłady na okablowanie są małe,
- nie występuje ciągłe połączenie pomiędzy dwoma partnerami.

4.1.3 Połączenia dwu-punktowe z multiplekserem

Jeżeli połączenie dwu-punktowe ma być wykorzystywane przez więcej niż dwóch uczestników, to musi być zastosowane takie rozwiązanie sieci, które pozwoli uniknąć wzajemnego oddziaływania sygnałów i tym samym ich zniszczenia. Możliwości realizacji takiej sieci komunikacyjnej polegają na zastosowaniu multipleksowania w czasie lub multipleksowania częstotliwości. Jeżeli zastosowana jest metoda multipleksowania w czasie, to mówi się o transmisji w paśmie podstawowym, ponieważ jest tutaj do dyspozycji niezmodulowany sygnał w paśmie częstotliwości od 0 Hz aż do częstotliwości granicznej medium transmisyjnego. Natomiast w przypadku stosowania metody multipleksowania częstotliwościowego zmodulowany sygnał jest transmitowany przy zdefiniowanej szerokości pasma. Zasada multipleksowania w czasie jest pokazana na rys.4.3. W multiplekserze (MUX) przy pomocy sygnałów sterujących a_0 i a_1 do przewodów transmisyjnych y przyłączane są kolejno dane $d_0 \dots d_3$. W ten sposób każdemu z czterech uczestników przydzielane

jest okno czasowe, w którym może on podjąć swoją transmisję. Demultiplekser (DEMUX) przy pomocy generowanych samoczynnie sygnałów sterujących a_0 i a_1 troszczy się o to, żeby odbierane szeregowo dane przyłączyć znowu do przewodów danych $d_0 \dots d_3$. Aby móc to w sposób bezproblemowy zrealizować oba dekodery 1 z 4 muszą pracować synchronicznie. Jest to uzyskiwane w ten sposób, że na początku każdego okresu transmisji do przewodów danych wysyłany jest sygnał synchronizacyjny. W wybranym wyżej zobrazowaniu przyporządkowanie jest odpowiednie tylko dla tzw. trybu *simplex*. Pod tym pojęciem rozumie się transmisję informacji wyłącznie w jednym kierunku.



Rys.4.3 Zasada działania multipleksera i demultipleksera

Ponieważ multiplekser i demultiplekser nie różnią się zasadą działania, dlatego też wymagane jest tylko jedno urządzenie MUX/DEMUX, w którym musi być tylko rozróżnienie pomiędzy przewodem „odbierającym” i „wysyłającym”. Dzięki temu możliwy jest tzw. tryb *półduplex*, w którym można wymieniać ze sobą informacje w obu kierunkach.

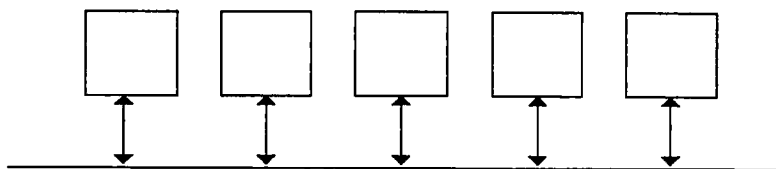
W metodzie multipleksowania częstotliwościowego kanał transmisyjny jest dzielony na niezależne od siebie pasma częstotliwości o zdefiniowanej szerokości. Tym samym powstaje możliwość równoczesnej transmisji wielu sygnałów w obu kierunkach. Ten sposób postępowania nadaje się do transmisji w trybie pełnego *duplex*’u.

W praktyce są stosowane modulacje: amplitudy, częstotliwości i fazy. Korzyścią wynikającą z ich zastosowania jest optymalne wykorzystanie medium transmisyjnego. Ponieważ wymagane do modulacji moduły są względnie drogie, dlatego też taka transmisja szerokopasmowa znajduje swoje zastosowanie głównie w sieciach rozległych WAN (*Wide Area Networks*).

4.1.4 Struktura magistralowa (*bus*)

W strukturze magistralowej, zwanej również strukturą liniową, wszyscy uczestnicy komunikują się przez wspólne przewody (rys.4.4). Przyłączenie uczestników do kabla magistrali odbywa się za pomocą krótkich przewodów (*Dropkabel*). Dzięki temu nakłady na okablowanie, w porównaniu z siecią oczkową, znacznie się redukują. Każdy uczestnik wymaga tutaj tylko jeszcze jednego

interfejsu, aby móc się komunikować z dowolnym, przyłączonym do magistrali uczestnikiem. Powstaje przy tym jednak taki problem, że zawsze tylko jeden uczestnik może wysyłać do jednego określonego punktu. Wskutek tego wymagane są reguły, które określają prawa dostępu do magistrali, tzw. metody dostępu do magistrali.



Rys.4.4 Liniowa struktura sieci

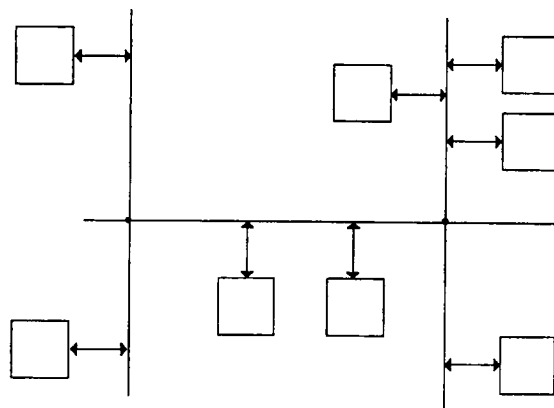
Podczas wykorzystywania struktury magistralowej występują następujące problemy:

1. Ponieważ wymagana jest dowolna komunikacja danych, to wszyscy uczestnicy muszą się „przysłuchiwać” każdej wysłanej transmisji. Wskutek tego przy rosnącej ilości uczestników nadajnik jest coraz mocniej obciążony.
2. Odcinki transmisyjne dla systemów komunikacyjnych rozciągają się często w zakresie kilkuset metrów. Tym samym długości przewodów nie są już pomijalnie małe w porównaniu z transmitowaną długością fal. Dlatego też przewody magistrali na obu końcach muszą mieć zamkniętą swoją oporność falową, aby uniknąć odbić w przewodach, które mogą znacznie wpływać na jakość sygnału. Ta oporność zamykająca obciąża nadajnik.

4.1.5 Struktura drzewiasta

W strukturze drzewiastej chodzi o dalsze rozwinięcie struktury liniowej. Ta topologia obejmuje większe powierzchnie aniżeli struktura magistralowa (rys.4.5). Natomiast wymagania dotyczące maksymalnej długości przewodów, maksymalnej ilości uczestników i maksymalnej szybkości transmisji są takie same jak w strukturze magistralowej. Wartości te można zwiększyć za pomocą tzw. *repeater*'s. W urządzeniach tych chodzi o elementy wzmacniające. W strukturze drzewiastej są one używane do tworzenia nowych gałęzi. Wskutek większych długości przewodów konieczne jest m.in. oddzielenie (separacja) galwaniczne uczestników od siebie. Jest to realizowane zwykle na wejściu każdego uczestnika, przy czym *repeater* jest także uczestnikiem. Dzięki separacji galwanicznej pokonuje się problemy, które powstają wskutek różnicy potencjałów wzdłuż przewodów i wynikających stąd prądów wyrównawczych.

Jeżeli stosuje się nadajniki, które mają różnicowe wyjścia napięciowe i odbiorniki z napięciowym wejściem różnicowym, to można w znacznym stopniu stłumić zakłócenia powstające wskutek sprzężeń elektromagnetycznych.

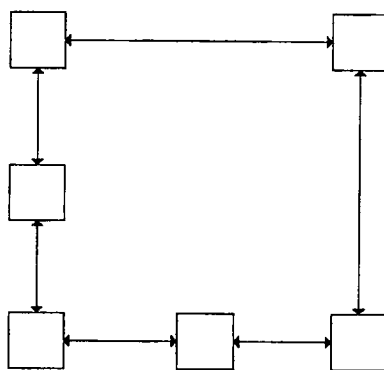


Rys.4.5 Struktura drzewiasta sieci komunikacyjnej

Przyjmując założenie, że w kablu chodzi o skręconą parę przewodów, to można przyjąć, że sprzężenia elektromagnetyczne oddziałują w jednakowym stopniu na oba przewody. Tym samym nie ma już oddziaływania na wejście wzgl. wyjście *transceiver*'a (słowo utworzone z *Transmitter* = nadajnik i *Receiver* = odbiornik). Dalszą poprawę jakości transmisji sygnałów uzyskuje się dzięki ekranowaniu skrętki.

4.1.6 Struktura pierścieniowa

Za pomocą wielu połączeń dwóch punktów jest w tej strukturze zbudowany fizyczny pierścień (rys.4.6). Transmitowana informacja jest przy tym przekazywana dalej od uczestnika do uczestnika. Również tutaj za pomocą odpowiedniej metody dostępu do magistrali musi być zapewnione to, że w określonej chwili czasu informacja jest wysyłana tylko przez jednego uczestnika.



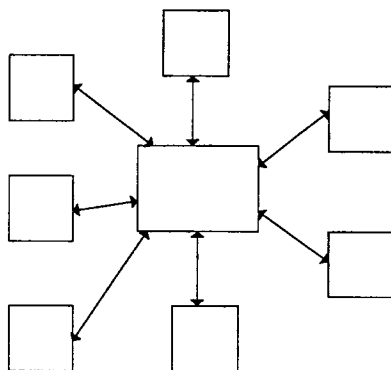
Rys.4.6 Struktura pierścieniowa sieci

Wskutek tego, że struktura pierścieniowa jest zbudowana z połączeń dwu-punktowych i każdy z uczestników może działać jako *repeater*, to mogą być tutaj pokonywane względnie duże odległości. Pomiędzy dwoma uczestnikami przy zastosowaniu światłowodów są one w zakresie kilometrów, przy dużej szybkości transmisji. I tak np. system magistralowy SINEC H3 pozwala na uzyskanie obwodu pierścienia 100 km przy szybkości transmisji 100 Mbit/s. Problematiczna jest ta topologia w przypadku awarii jednego uczestnika wzgl. przerwaniu kabla lub zwarcia. Bez

odpowiednich środków zaradczych oznaczałoby to awarię całej sieci. Jeżeli pierścień jest wykonany redundantnie (nadmiarowo) w taki sposób, że można transmitować informacje w obu kierunkach, to można obejść uszkodzone miejsca. Dzięki odpowiednim mechanizmom poszukiwania można je zlokalizować i za pomocą mostków zwierających wyłączyć z pierścienia.

4.1.7 Struktura gwiazdy

W tym przypadku centralna stacja za pomocą połączeń dwóch punktów jest łączona z każdym z uczestników (rys.4.7). Istnieją dwie możliwości realizacji stacji centralnej. Po pierwsze tzw. sprzęg gwiazdzisty (*hub* - koncentrator), którego zadaniem jest wyłącznie przekazywanie sygnałów od nadajnika do właściwego odbiornika. Hub może być przy tym zarówno bierny (pasywny) jak i czynny (aktywny), tzn. odbierane sygnały przed dalszym przekazywaniem są jeszcze przygotowywane.



Rys.4.7 Struktura gwiazdzista sieci komunikacyjnej

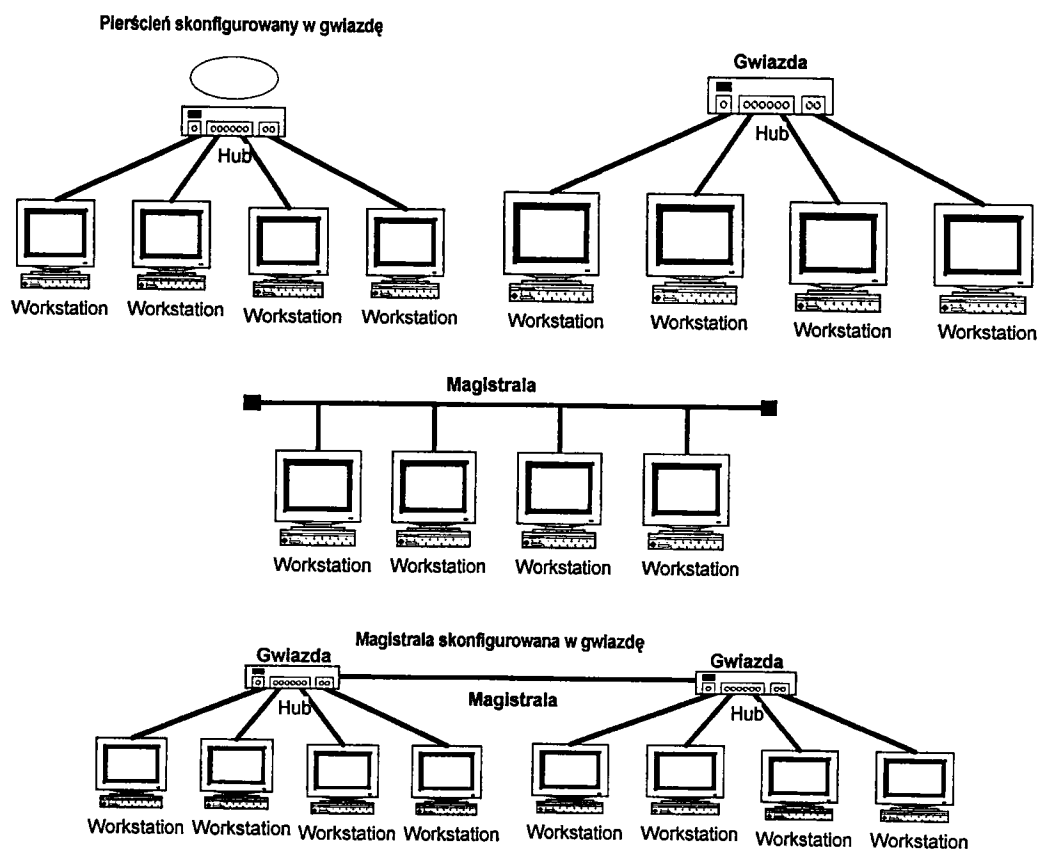
Drugą możliwością jest implementacja inteligencji w centralnej stacji. Tym samym stacja ta może przejąć sterowanie całym procesem komunikacyjnym, który w pierwszym przypadku musi być podejmowany przez jednego lub wszystkich pozostałych uczestników. Zasadniczo komunikacja pomiędzy dwoma uczestnikami ma miejsce przez stację centralną. Tym samym jest ona wąskim gardłem, gdyż awaria tej stacji powoduje niesprawność całej sieci.

4.1.8 Topologia sieci lokalnych LAN i WAN

Topologia sieci określa fizyczne rozmieszczenie urządzeń sieci oraz technikę ich włączania do medium transmisyjnego. Generalnie rozróżnia się dwie kategorie układów topologicznych, w zależności od tego, czy jest siecią lokalną LAN, czy też siecią złożoną, np. rozległą - WAN [2]. Wyróżnia się sieci o następujących topologiach (rys.4.8):

- Magistrala (ang. *bus*) - do magistrali, tzn. pojedynczego kabla głównego (ang. *trunk cable*), stanowiącego wspólną magistralę, podłączone są wszystkie stacje robocze. Nadawane sygnały docierają do wszystkich stacji w sieci, ale są odbierane wyłącznie przez jedną stację, do której są adresowane. Powszechnie znanym standardem wykorzystującym topologię magistrali jest Ethernet (IEEE 802.3).

- Gwiazda (ang. *star*) - stacje robocze są podłączone do wspólnego koncentratora (ang. *hub*), a nadawane sygnały są przekazywane od stacji do stacji lub są odbierane przez wszystkie stacje pracujące w sieci.
- Pierścień skonfigurowany w gwiazdę (ang. *star-configured ring*). Fizyczną topologią w tym rozwiązaniu jest gwiazda, w której centralną rolę pełni koncentrator, jednak sygnały przekazywane są od stacji do stacji tworząc logiczny pierścień. Podstawowym standardem tej topologii sieci jest *IEEE 802.5 Token Ring*.
- Konfiguracja gwiazda / magistrala (ang. *star/bus configuration*). Jest to sieć, w której grupy stacji roboczych, połączonych w gwiazdę, podłączone są do długich odcinków kabli głównych, stanowiących magistralę. Przykładem sieci, która wykorzystuje tą topologię jest *IEEE 802.4 Token Bus*.



Rys.4.8 Przykłady topologii sieci lokalnych LAN

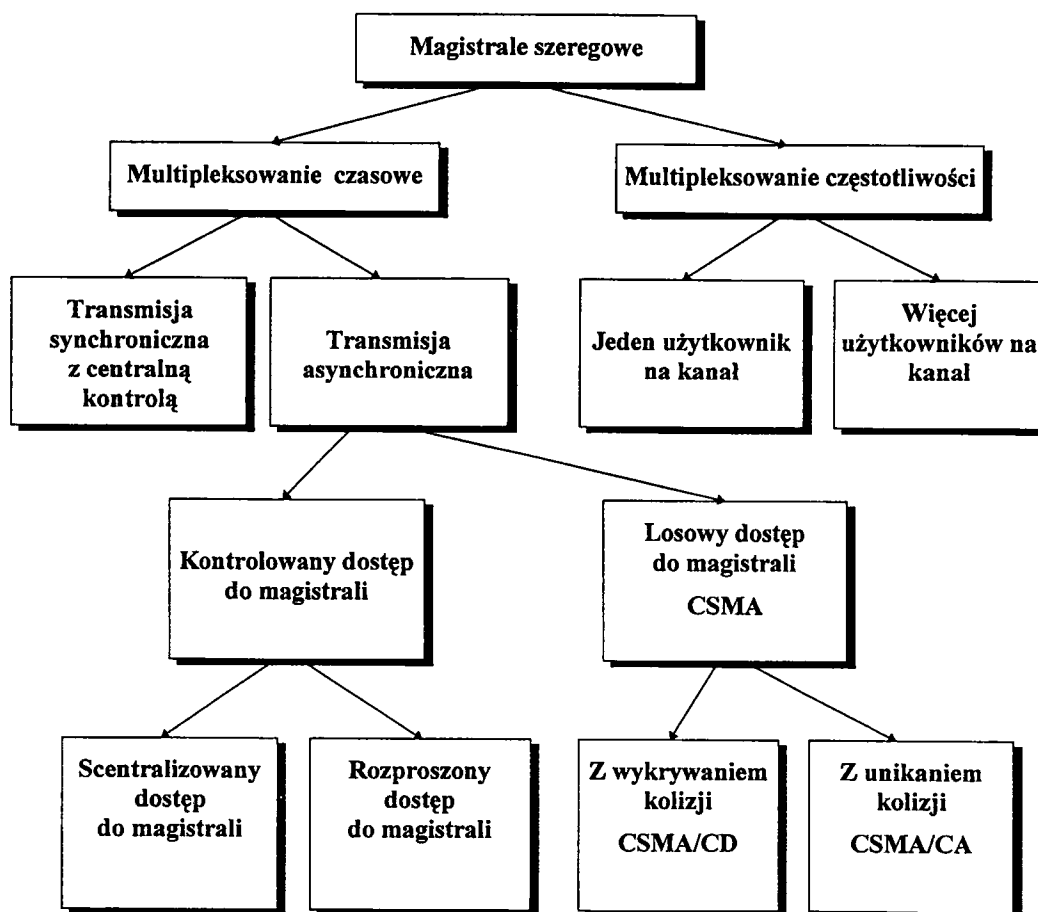
Do najważniejszych standardów sieci informatycznych LAN, które powstały i były wylansowane przez duże koncerny komputerowe, należy zaliczyć :

- Ethernet IEEE 802.3 (Intel, Xerox, Dec),
- Token Ring IEEE 802.5 (IBM),
- Token Bus IEEE 802.4 (Datapoint).

4.2 Metody dostępu do medium transmisyjnego

4.2.1 Wstęp

Prawie wszystkie omówione topologie sieci komunikacyjnych wymagają, żeby w określonej chwili czasu do wspólnego medium transmisyjnego miała dostęp tylko jedna stacja. Poniżej zostaną omówione różne możliwości regulowania dostępu do magistrali. Rozróżnia się przy tym kontrolowane (deterministyczne) i przypadkowe (losowe) metody dostępu do magistrali (rys.4.9).



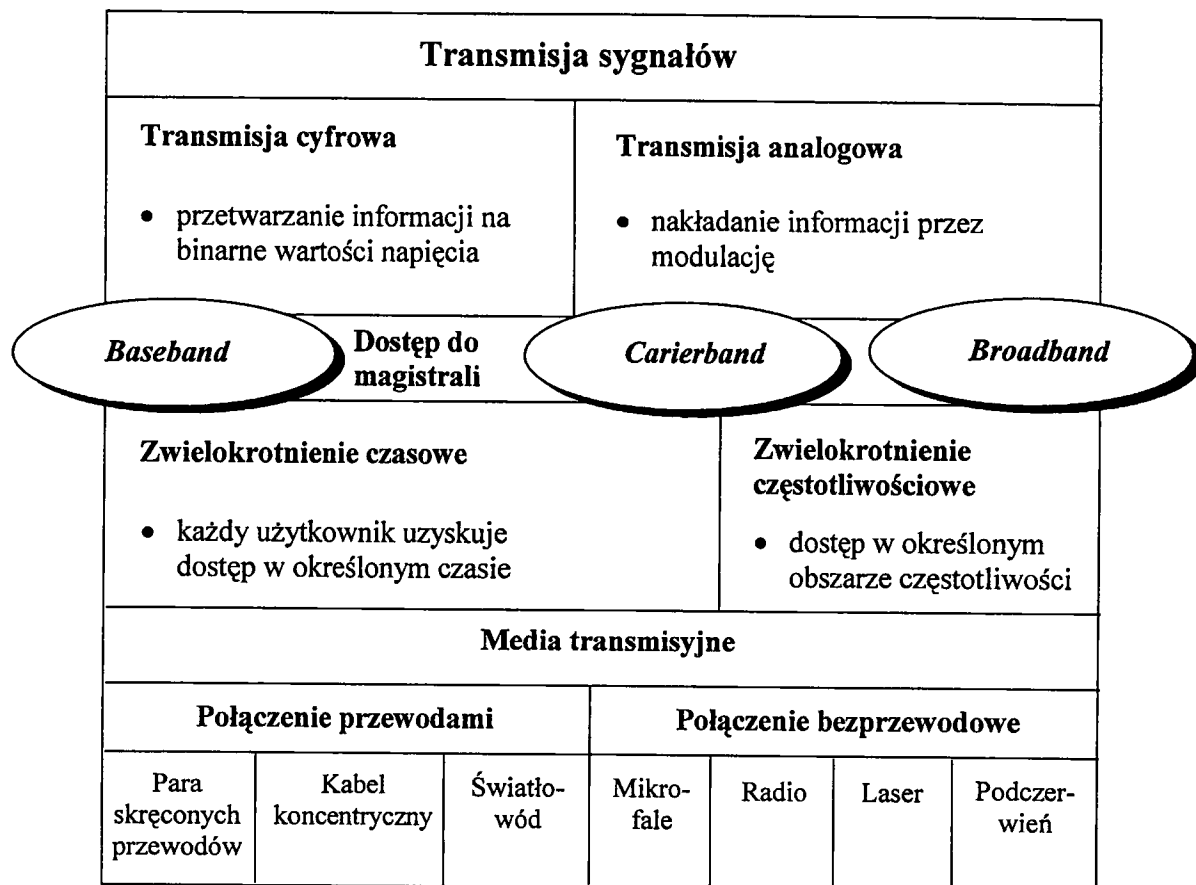
Rys.4.9 Zestawienie metod dostępu do magistrali

W deterministycznych metodach dostępu do magistrali nadajnik przed rozpoczęciem nadawania jest jednoznacznie określony. Tym samym dla każdego nadajnika konieczny jest przydział magistrali. Może być on przeprowadzany centralnie przez stację nadrzędną (*metoda master/slave*) lub w sposób zdecentralizowany przez wiele jednostek sterujących (*Token-Bus, Token-Ring*). Jeżeli okres czasu lub długość danych dla cyklu komunikacyjnego jest ograniczona, to maksymalny czas trwania okresu w którym dane są transmitowane jest obliczalny (czas cyklu). Takie systemy nazywa się systemami czasu rzeczywistego.

W losowych metodach dostępu uprawnieni do nadawania użytkownicy mają dostęp do medium transmisyjnego tylko w przypadku potrzeby. Musi być przy tym zapewnione to, że medium nie jest zajęte przez innego uczestnika. Jeżeli występuje taki przypadek, to nadawanie musi być przesunięte

na późniejszą chwilę. Tym samym określenie maksymalnego okresu czasu, w którym informacja jest przesyłana, nie jest już możliwe. Dlatego też losowe metody dostępu do magistrali nie są w zasadzie metodami czasu rzeczywistego.

Na rys.4.10 zamieszczono techniki transmisji wg otwartego modelu warstwowego ISO/OSI.



Rys.4.10 Techniki transmisji wg otwartego modelu warstwowego ISO/OSI

W przeciwieństwie do połączeń typu punkt-punkt, w magistrali szeregowej do wspólnego systemu przewodów jest fizycznie lub logicznie przyłączonych wiele stacji sieciowych. Aby podczas transmisji danych nie występowały kolizje pomiędzy różnymi partnerami komunikacyjnymi, to możliwości transmisyjne systemu przewodów muszą być dzielone pomiędzy poszczególne stacje sieciowe. Możliwe jest tutaj w zasadzie wykorzystanie dwóch podstawowych metod transmisji, a mianowicie: multiplexowania (zwielokrotnienia) częstotliwości oraz multiplexowania czasowego. W systemach magistrali szeregowej multipleksowanie częstotliwości nie jest stosowane, ponieważ jest ono technicznie związane z dużymi nakładami. W multipleksowaniu czasowym zawsze tylko jedna stacja danych może w danej chwili czasu wysyłać dane. Aby pomimo tego móc zapewnić wrażenie równoczesnej koegzystencji różnych relacji komunikacyjnych, poszczególne stacje danych muszą dzielić pomiędzy siebie czas wysyłania danych. Do tego celu niezbędne są metody dostępu, które sterują dostępem poszczególnych stacji sieciowych do systemu przewodów (medium transmisyjnego).

Metody dostępu do medium transmisyjnego określają sposoby wprowadzania i odbierania danych z medium przez poszczególne stacje sieci komunikacyjnej. Ogólnie metody dostępu do medium można podzielić na losowe i deterministyczne. Wśród losowych metod dostępu CSMA wyróżnia się metodę z wykrywaniem kolizji CSMA/CD (ang. *Carrier Sense Multiple Access with Collision Detection*) i metodę z unikaniem kolizji CSMA/CA (ang. *Carrier Sense Multiple Access with Collision Avoid*).

4.2.2 Metoda dostępu CSMA/CD

Metoda dostępu do medium z wykrywaniem kolizji CSMA/CD polega na tym, że stacje „przysłuchują” się ruchowi w sieci komunikacyjnej i transmitują dane tylko wówczas, kiedy wykryją „ciszę”. Występuje jednak rywalizacja poszczególnych stacji w dostępie do medium, co w praktyce prowadzi do występowania kolizji wówczas jeżeli dwie lub więcej stacji równocześnie wysyła dane. Dlatego też podczas nadawania stacja „nasłuchuje”, czy w medium nie występują nieoczekiwane zmiany sygnałów świadczące o wystąpieniu kolizji. Wówczas stacja, która pierwsza wykryła kolizję przestaje nadawać i wysyła charakterystyczny sygnał (ang. *Jamming signal*) informujący pozostałe stacje o wystąpieniu kolizji. Stacje, które nadawały podczas wystąpienia kolizji na pewien okres przerywają transmisję, a następnie w losowych odstępach czasu ponawiają wysłanie danych. Jeżeli ponownie wystąpi kolizja, to czas przerwy w nadawaniu dla stacji jest podwajany. Ponieważ stacja ma dostęp do medium transmisyjnego tylko wtedy, gdy chce wysłać dane, to nie można z góry określić, która ze stacji będzie przysyłać dane. Tym samym jest to losowa metoda dostępu. Ponieważ nie można również zapewnić, że magistrala będzie wolna wówczas gdy wystąpi życzenie nadawania, to nie można również zagwarantować maksymalnego czasu, w którym dane zostaną przesłane. Tym samym metoda ta nie spełnia warunków czasu rzeczywistego.

Występują dwa warianty regulowania ponownego dostępu do magistrali po nieudanej próbie:

W *wariancie 1* uprawniona do nadawania stacja wycofuje się na losowo wybrany czas i próbuje ponownie uzyskać dostęp do magistrali. Jeżeli magistrala jest jeszcze ciągle zajęta lub została znowu zajęta, to stacja ponownie wycofuje się na losowo wybrany czas, który jest jednak dłuższy niż wcześniejszy okres. Występuje przy tym taki problem, że m.in. uprawniony do nadawania użytkownik musi bardzo długo czekać, zanim znajdzie wolne medium transmisyjne. Problem ten jest tym większy, im silniej jest obciążona magistrala. Wskutek losowo wybranych czasów oczekiwania może dochodzić do okresów czasu, w których medium transmisyjne jest niewykorzystywane. Ten wariant jest stosowany np. w komunikacji biurowej przez sieć Ethernet.

W *wariancie 2*, który jest opisany w normie IEE 802.3, uprawniony do nadawania użytkownik ciągle nasłuchuje medium transmisyjne i nadaje natychmiast, gdy bieżąca komunikacja została zakończona. Wskutek tego nie występują żadne czasy oczekiwania. Może się jednak przy tym zdarzyć, że dwie stacje zaczną równocześnie nadawać, ponieważ one obie podczas wcześniejszej transmisji próbowały uzyskać dostęp do magistrali. Następstwem tego jest kolizja wysyłanych

informacji i ich wzajemne niszczenie. Bez dodatkowych przedsięwzięć byłoby to rozpoznawane dopiero wówczas, gdy odbiornik sprawdzi poprawność przesłanych danych. Tym samym magistrala byłaby przez cały czas zajęta nakładającymi się transmisjami i nie mogłaby być szerzej wykorzystana, a efektywność transmisji znacznie by zmalała. Problem ten występuje również w wariancie 1, jeżeli dwie stacje quasi równocześnie odsłuchują magistralę, określają ją jako wolną i rozpoczynają nadawanie. Również tutaj występuje taka sytuacja, że ze wzrostem obciążenia magistrali wzrasta prawdopodobieństwo kolizji.

Istotnymi parametrami sieci z dostępem CSMA/CD są prędkość transmisji i rozmiar pakietu danych (liczby bitów przesyłanych w jednym akcie transmisji). W celu umożliwienia wykrycia kolizji czas nadania pakietu danych powinien być większy od dwukrotnego czasu propagacji sygnału w sieci. Dlatego też w sieciach z dostępem CSMA/CD (np. Ethernet) konieczne jest zachowanie kompromisu pomiędzy zasięgiem sieci, szybkością transmisji oraz wielkością przesyłanych pakietów informacji.

4.2.3 Metoda dostępu CSMA/CA

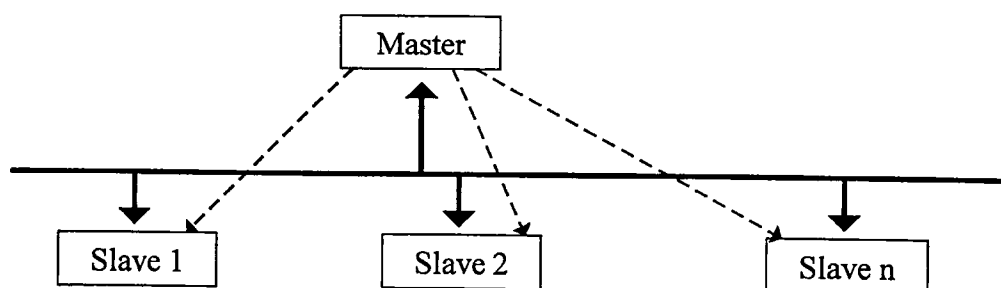
Metoda dostępu do medium CSMA/CA polega na unikaniu występowania kolizji w sieci. W tej metodzie uprawniony do nadawania uczestnik nasłuchuje medium transmisyjne tak jak w CSMA/CD i rozpoczyna transmisję wtedy, gdy medium jest wolne. Jeżeli jest ono zajęte, to odczekuje się bieżącą transmisję i bezpośrednio po jej zakończeniu rozpoczyna się nadawanie, przy czym nadawanie jest ciągle nadzorowane. Jeżeli dwóch uczestników powinno równocześnie rozpocząć nadawanie, to zadawane są priorytety, tak że uczestnik z mniejszym priorytetem wycofuje się, tzn. przerywa swoją transmisję, ponieważ nie odebrał on swojej własnej przesyłki. Tym samym unika się kolizji (*Collision Avoidance*, CA). W metodzie tej stacja przed przystąpieniem do transmisji sprawdza stan sieci i jeżeli nie wykryje ona transmisji pochodzącej od innych stacji, to wysyła krótki sygnał oznaczający zgłoszenie nadawania. Następnie, po odczekaniu określonego okresu czasu dla zapewnienia dotarcia tego sygnału do wszystkich stacji, rozpoczyna ona nadawanie. Każda stacja, która nawet w trakcie przesyłania danych otrzymała sygnał zgłoszenia nadawania innej stacji, zaprzestaje transmisji i odczekuje losowo dobierany okres czasu. Jeżeli wystąpi kolizja w sieci, to stacja wykrywająca kolizję zaprzestaje nadawania na krótki czas o losowej długości, a następnie ponawia próbę zarezerwowania sieci dla swojej transmisji, przy czym o prawo dostępu do medium ubiegają się tylko te stacje, których sygnały zgłoszenia nadawania brały udział w kolizji. Stacja, która wygrała rywalizację o dostęp do sieci wysyła pakiet danych, a wszystkie pozostałe stacje w tym czasie nasłuchują nadejścia sygnału oznaczającego koniec tego pakietu. Metoda CSMA/CA zapewnia efektywne przesyłanie małych pakietów danych i jest ona stosowana w sieciach szerokopasmowych i innych wolnych sieciach LAN.

Telegram rozpoczyna się zawsze znacznikiem początku transmisji. Ta informacja jest jednakowa dla wszystkich uprawnionych do nadawania stacji, tak że dwie znajdujące się

równocześnie w kablu transmisyjnym przesyłki nie oddziałują na siebie wzajemnie. Może występować własny adres nadajnika, tzw. *Identifier*. Definiując jeden stan logiczny jako dominujący, np. „0”, a inny jako *recessive*, to dominująca jest stacja z niższym adresem. Stacja z wyższym adresem przerywa nadawanie i próbuje wysłać swoje dane dopiero po wykonaniu bieżącej transmisji. Jeżeli czas trwania dla transmisji jest ograniczony, to stacja z niskim adresem ma warunki pracy w czasie rzeczywistym. Jeżeli jednak ta stacja ze względu np. na błędną transmisję stale by nadawała, to cała sieć by nie funkcjonowała. Dlatego też definiuje się okres czasu, w obrębie którego ta sama stacja po wykonaniu transmisji danych nie może już wysłać. Tym samym zawsze pozostaje przedział czasu, w którym inne stacje mogą przeprowadzić swoje transmisje danych. Wskutek tego również inne stacje uzyskują warunki pracy w czasie rzeczywistym. Ilość stacji pracujących w czasie rzeczywistym w tym systemie jest zależna od długości „zakazanego” przedziału czasu.

4.2.4 Metoda *master/slave*

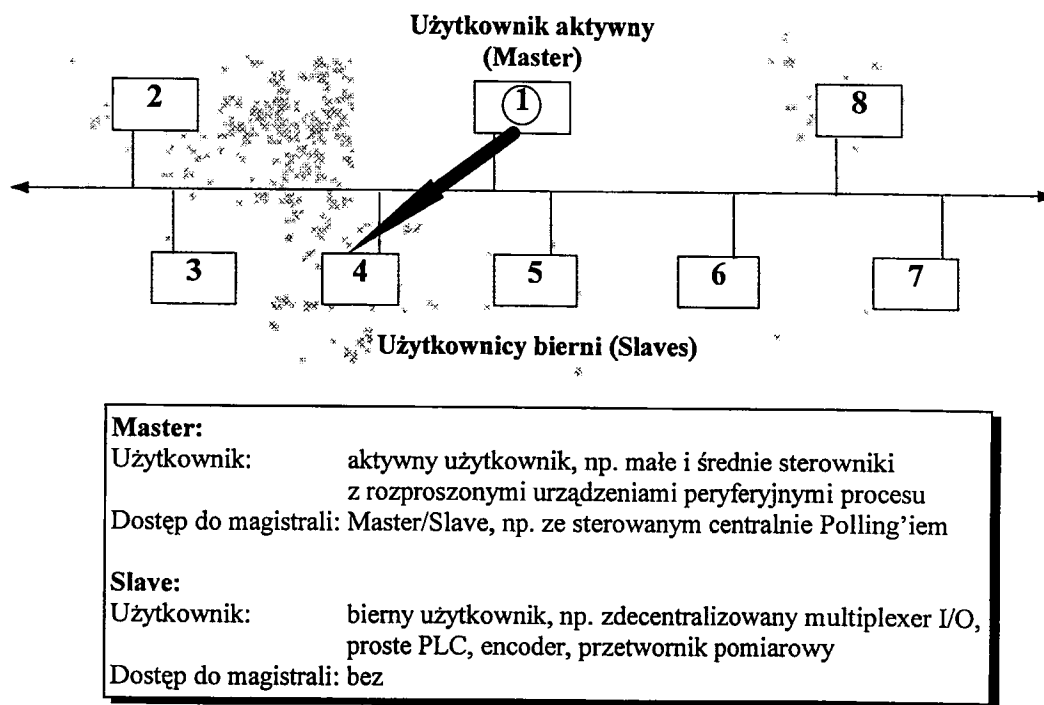
W metodzie *master/slave* jednostka sterująca magistralą, tzw. *master*, tworzy połączenie z biernym uczestnikiem, *slave*'m. Ten odpowiada bezpośrednio na żądanie danych *master*'a (*immediate response*) (rys.4.11). *Master* zwykle tworzy cyklicznie połączenia z każdym *slave*'m. (*polling*). W ten sposób w *master*'ze jest zawsze złożone aktualne odwzorowanie sterowanego procesu. Priorytety mogą być zadawane w ten sposób, że niektóre ze *slave*'ów mogą być wielokrotnie odpytywane podczas jednego cyklu. Ta metoda ma tą zaletę, że przyłączenie *slave*'ów do magistrali jest bardzo proste i tym samym tanie, ponieważ cała wymagana inteligencja jest zaimplementowana w *master*'ze. Problematiczne mogą być czasy, które są wymagane wtedy, gdy pomiędzy dwoma *slave*'ami muszą być wymieniane dane. W tym przypadku *master* wysyła żądanie danych do *slave*'a dostarczającego sygnału, na co on bezpośrednio odpowiada. Te informacje muszą być przetworzone w *master*'ze i przesłane do odbierającego *slave*'a. Oznacza to, że przypadku ekstremalnym wymagany jest czas cyklu zarówno dla żądania danych jak i wysłania danych. Ponadto musi być uwzględniony czas przetwarzania w *master*'ze. Tym samym czas trwania tej transmisji danych jest wielokrotnością czasu cyklu.



Rys.4.11 Metoda dostępu *master/slave*

Realizacja złożonych metod sterowania dostępem do medium dla sieci lokalnych jest związana ze znacznymi nakładami na sprzęt i oprogramowanie. Dlatego też dla łączonych za pomocą sieci miejscowych prostych urządzeń końcowych wzgl. członów wykonawczych i czujników są wymagane tanie i proste metody sterowania dostępem do medium. W stosowanych obecnie sieciach miejscowych do sterowania dostępem do magistrali jest często stosowana metoda przepytывania (ang. *Polling*). W przeciwieństwie do trzech metod sterowania dostępem do medium wg ISO 8802, w których sterowanie dostępem systemu komunikacyjnego do medium transmisyjnego odbywa się w sposób zdecentralizowany, to w *Polling*'u zadanie to spełnia centralna stacja - *master* (rys.4.12). Ten *master* steruje i kontroluje wszystkie operacje komunikacyjne stacji na magistrali. *Master* żąda od podporządkowanych mu hierarchicznie stacji - *slave*'s przesłania lub odbioru danych. Ponieważ tylko *master* jest odpowiedzialny za sterowanie dostępem, dlatego też ten sposób daje się zrealizować prościej i tym samym taniej niż w metodach wg ISO 8802.

Istotną wadą metody *master/slave* jest to, że w przypadku awarii *master*'a niesprawny jest cały system sieciowy. Również tutaj istnieje jednak możliwość nakazania nadzorowania *master*'a przez *slave*'a. Ten *slave* musi móc przejąć wszystkie zadania *master*'a. Należą do nich, oprócz sterowania komunikacją, nadzorowanie awarii *slave*'ów, nadzorowanie jakości transmisji i obsługa błędów.



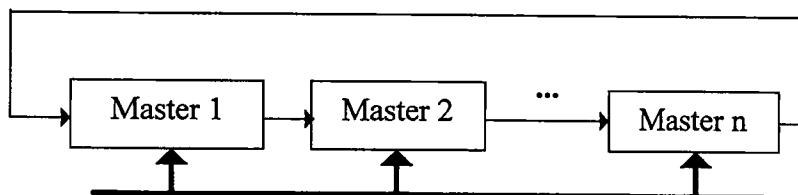
Rys.4.12 Zasada dostępu do magistrali w metodzie przepytывania (*Polling*'u)

Przez nadzorowanie awarii *slave*'ów uzyskuje się możliwość ustawienia uszkodzonego *slave*'a w stan nieaktywny i usunięcia z listy *Poll*. Równocześnie musi być zapewniona możliwość tego, że ten *slave* zostanie ponownie umieszczony na liście *Poll*, gdy będzie on sprawny.

4.2.5 Zasada *token'a* (żetonu)

Deterministyczne metody dostępu do medium transmisyjnego opierają się na wykorzystaniu znacznika (ang. *token*) tzn. małej, łatwo rozpoznawalnej ramki ciągle krążącej w sieci. Metoda ta jest stosowana zarówno w sieciach o topologii pierścienia (*Token Ring*) jak i magistrali (ang. *Token Bus*). W systemach, które pracują wg zasady *token'a*, wszyscy użytkownicy są w stanie przejąć sterowanie komunikacją. Uprawnienie posiadania dostępu do magistrali i tym samym przejęcie sterowania komunikacją jest udzielane za pomocą specjalnego znaku lub specjalnej informacji (*token'a*) dokładnie jednej stacji. Jeżeli zakończyła ona transmisję swoich danych, to *token* jest przekazywany dalej do następnej stacji. Ponieważ okres posiadania *token'a* jest limitowany, to te metody dostępu umożliwiają pracę w czasie rzeczywistym.

Na rys.4.13 pokazano sieć **Token-Bus**, która jest wyspecyfikowana w normie IEEE 802.4. Wszyscy użytkownicy są tutaj przyłączeni do wspólnego kabla magistrali (topologia liniowa). *Token* stanowi specjalną krótką informację i przez przewody danych jest przekazywany od jednej stacji do drugiej. Jedną z możliwości realizacji jest przekazywanie *token'a* dalej do stacji z następnym niższym adresem. Stacja z najniższym adresem przekazuje *token* do stacji z wyższym adresem. W ten sposób powstaje logiczny pierścień. Priorytety można realizować przez różnorodne maksymalne ilości danych lub przez wielokrotny przydział *token'a* w obrębie jednego cyklu.



Rys.4.13 Sieć *Token Bus*

Użytkownik w pierścieniu logicznym musi być w stanie realizować następujące funkcje nadzorujące:

- nadzorowanie *token'a* - jeżeli w wyniku błędu nie ma żadnego *token'a* lub też jest wiele *token'ów*, to „stare” *token'y* muszą być skasowane i utworzony jeden „nowy” *token*,
- inicjalizacja po włączeniu - np. zdefiniowany uczestnik musi utworzyć *token*,
- usunięcie stacji z pierścienia logicznego: musi być możliwość usunięcia uszkodzonych stacji z logicznego pierścienia,
- włączenie wstawianych stacji do pierścienia logicznego - musi być możliwość ponownego przyjęcia do pierścienia logicznego wcześniej usuniętych stacji, wzgl. rozszerzenia pierścienia logicznego.

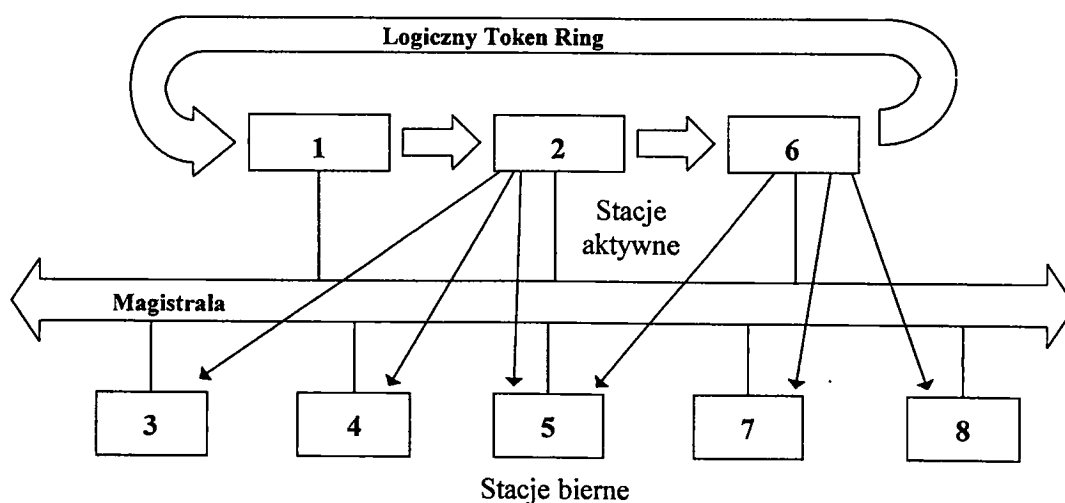
Natomiast w sieciach **Token Ring** każdy sygnał jest odbierany i dalej przekazywany przez każdą stacją. W tym przypadku, gdy żadna ze stacji nie wysyła danych, to w sieci krąży tylko znacznik (ang. *token*). Uprawnienia do nadawania ma tylko ta stacja, która posiada znacznik. Tak

więc stacja, która ma do wysłania informację, musi czekać na otrzymanie znacznika. Metoda dostępu do medium z przesyłaniem znacznika jest wprawdzie mniej efektywna niż metody CSMA, ale zapewnia ona stacji dostęp do sieci w określonym czasie. Poszczególnym stacjom w sieci *Token Ring* mogą być przypisane różne priorytety transmisji, a sieć ta jest objęta międzynarodową normą IEEE 802.5. Jeżeli stacja zakończyła swoją transmisję danych, to przekazuje ona tzw. wolny *token* do następnej stacji w pierścieniu logicznym. Ta stacja przetwarza wolny *token* w *token* zajęty (zmiana 1-go bitu) wówczas, jeżeli ma dane do transmisji. Jeżeli nie ma takiego przypadku, to wolny *token* jest bezpośrednio przekazywany dalej do następnej stacji. Dane do transmisji są wstawiane za zajęętym *token*'em i są przekazywane do następnej stacji. Jeżeli dane nie są przeznaczone dla tej następnej stacji, to są one bezpośrednio przekazywane dalej. Jeżeli dane osiągnęły odbiorcę, to są one kopiowane do specjalnego obszaru pamięci. Odbiorca przesyła odebrane dane do następnej stacji, aż dotrą one z powrotem do nadawcy. Nadajnik zmienia zajęty *token* na *token* wolny wówczas, jeżeli odebrał on bezbłędnie swoją przesyłkę i przekazuje go (wolny *token*) do następnej stacji.

Problematyczny jest ten system wówczas, jeżeli jedna ze stacji jest uszkodzona lub też dojdzie do przerwania wzgl. zwarcia kabla. Jedną z możliwości rozwiązania tego problemu jest zastosowanie tzw. struktury podwójnego pierścienia. Pierścień jest przy tym zrealizowany w sposób redundantny, przy czym każdy uczestnik ma teraz możliwość komunikowania się zarówno ze swoim następcą jak i poprzednikiem. Te możliwości komunikacyjne mogą być wtedy cyklicznie sprawdzane. W ten sposób można zlokalizować uszkodzone miejsca i wyłączyć je z pierścienia.

4.2.6 Metoda dostępu *Token-Passing*

W metodzie *Token-Passing*, zwanej również hybrydową metodą dostępu, chodzi o kombinację *Token-Bus* z metodą *Master/Slave* (rys.4.14).



Rys.4.14 Zasada działania metody *Token Passing*

Występują tutaj zarówno stacje aktywne, które mogą przejąć sterowanie komunikacją magistrali, jak i stacje bierne, które nie mają dostępu do magistrali. *Token* jest kolejno przekazywany pomiędzy aktywnymi stacjami w pierścieniu logicznym.

Każdorazowy posiadacz *token*'a może się komunikować z bierną stacją metodą *master/slave*, a z aktywnym użytkownikiem wg zasady *token*'a. Oznacza to, że posiadacz *token*'a musi rozróżniać, czy komunikuje się on ze stacją aktywną czy bierną. W PROFIBUS'ie, którego model komunikacyjny opiera się na modelu warstwowym OSI, odbywa się to w warstwie 2 i tym samym nie jest widoczne dla użytkownika. Ta metoda ma tą zaletę, że jest ona elastyczna, co dotyczy modułów przyłączanych. Jednak jako wada uwidaczniają się zwiększone nakłady na projektowanie.

Literatura

- [1] Bender K.: *Automatisierungstechnik im Maschinenbau I*, Vorlesungsskriptum, TU München, 1995
- [2] Weck M., *Werkzeugmaschinen. Fertigungssysteme*, VDI, 1995
- [3] Schnell G., *Bussysteme in der Automatisierungstechnik*, Vieweg, Braunschweig/Wiesbaden, 1994

4.3 Formaty ramek transmisyjnych

4.3.1 Protokół *Kermit*

Opracowany na Uniwersytecie Columbia (Nowy Jork - USA) protokół *Kermit* miał początkowo służyć do obsługi połączeń pomiędzy mikrokomputerami i komputerami klasy *mainframe*. Dlatego też *Kermit* jest szczególnie przydatny dla użytkowników komputerów *mainframe*, stosowanych do przechowywania danych. Dzięki temu protokołowi użytkownicy mogą np. wysyłać napisane przez siebie programy do komputera *mainframe* i w ten sposób udostępniać je innym użytkownikom. *Kermit* został rozpowszechniony głównie w środowiskach akademickich, a jego wersje opracowano dla wszystkich głównych standardów komputerów. Oprócz wersji zawierających sam protokół *Kermit* dostępne są również wersje zawierające wszystkie usługi i funkcje komunikacyjne dla określonych komputerów. Tutaj pod pojęciem *Kermit* rozumie się jedynie protokół przesyłania plików.

Kermit nie stawia zbyt wysokich wymagań sprzętowych komputerom i może być wykorzystywany w systemach z ograniczeniami długości słów (do siedmiu bitów), nawet wówczas gdy muszą być przesyłane znaki ośmiobitowe, a ponadto nie jest wymagane dodatkowe potwierdzanie sygnałów (oprócz stosowanego przez sam *Kermit*'a). Kodowanie znaków odbywa się w ten sposób, że wszystkie przesyłane znaki są zamieniane do formatu wydruku (znaki ASCII od 32 do 126). Dzięki temu wszystkie pozostałe znaki mogą być przesyłane bez specjalnego ich traktowania w komputerze odbierającym. Znaki sterujące są z zakresu 0 do 31 oraz 127. Przed wysłaniem takich znaków są one tłumaczone na znaki z zakresu 32-126 z „XOR’owaniem” z liczbą 64 oraz poprzedzane zwykle znakiem #. I tak np. pierwszy znak w tabeli ASCII *Ctrl-A*, staje się *A*, czyli ASCII 65 i jest przesyłany jako *#A*.

Znaki nie składające się na przesyłaną wiadomość, ale zawierające numeryczną informację wewnętrzną dla protokołu *Kermit* taką jak np. długość pakietu, są kodowane przez dodanie liczby 32, co jest określane jako funkcja *tochar()*, natomiast funkcją dekodującą jest *unchar()*. Największą liczbą, którą może zakodować funkcja *tochar()* jest 94 ($94 + 32 = 126$, czyli liczba zawierająca się jeszcze w siedmiu bitach). Ogranicza to liczbę różnych parametrów protokołu *Kermit*. Znaki reprezentujące same siebie podczas przesyłania nie są kodowane (np. informacja o znaku poprzedzającym znaki sterujące, innym niż #). Wówczas gdy przesyłane są sekwencje powtarzających się znaków, to protokół *Kermit* może je poddać kompresji (podawany jest znak oznaczający powtarzanie tzn. *tylda* (~), liczba powtórzeń oraz znak, który ma być powtarzany).

Podstawową jednostką wielkości wiadomości protokołu *Kermit* jest pakiet, który może być zbudowany z danych lub innego rodzaju informacji (potwierdzenie lub komunikaty błędów). Każdy rodzaj komunikatu jest oznaczony literą, która jest przechowywana w polu typu pakietu:

- D Pakiet danych
- Y Potwierdzenie (ACK)

N	Potwierdzenie negatywne (NAK)
S	Wysłanie inicjacji (zamiana parametrów)
B	Przesłanie sygnału <i>Break</i>
F	Nagłówek pliku
Z	Koniec pliku
E	Błąd
T	Zarezerwowane dla użytku wewnętrznego
I	Inicjacja (zamiana parametrów)
A	Atrybuty pliku
R	Odbiór inicjacji
C	Polecenie <i>host'a</i> , zawierające polecenia serwera dla komputera <i>mainframe</i>
K	Polecenie protokołu <i>Kermit</i>
G	Ogólne polecenie <i>Kermit'a</i>

Ogólny format pakietu pokazano w tab.4.1. Długość poszczególnych pakietów różni się między sobą, a pakiety o różnej długości mogą być przesyłane podczas jednej operacji, przy czym wg standardu protokołu długość pakietu nie może przekraczać 96 znaków. W drugim polu każdego pakietu, tzn. polu LEN, zawarta jest liczba znaków po nim występujących aż do pola CHECK. Tak więc dla największego dozwolonego pakietu o długości 96 bajtów w polu LEN umieszczana jest wartość 94. Każdy pakiet ma pole liczby sekwencyjnej, która jest liczona od 0 dla pakietu inicjującego, a liczba jest modulo 64 (po liczbie 63 następuje 0).

Operacja przesłania danych rozpoczyna się od wysłania przez urządzenie nadające pakietu inicjującego, który zawiera różne parametry. Pakiet ten jest potwierdzany przez urządzenie odbierające, które w pakiecie ACK przekazuje również informację o tym, jakie parametry i opcje mogą być przez nie obsługiwane. Jeżeli wymiana informacji nie jest zakończona przesłaniem pliku, to pakiety inicjujące są nazywane pakietami typu I o nazwie *Init-Info*.

Tab.4.1 Format pakietu *Kermit*

OFFSET	ZAWARTOŚĆ	ZNACZENIE
0	MARK	Zwykle ^A, oznaczający początek pakietu
1	LEN	Liczba znaków następujących po tym polu
2	SEQ	Liczba sekwencyjna, modulo 64, rozpoczyna się od 0
3	TYPE	Typ pakietu
4 ...	DATA	Zawartość pakietu, dane
[end]	CHECK	Suma kontrolna

Pakiet, który jest wysłany przez urządzenie nadające i jest powtarzany do momentu potwierdzenia, zawiera sześć najważniejszych znaków umieszczonych w polach. Podobną informację zawiera pakiet pochodzący od urządzenia odbierającego. Pakiet inicjujący zawiera pola:

MAXL	Maksymalna długość pakietu, który może być odebrany.
TIME	Liczba sekund, jaką powinno oczekiwać urządzenie przed zgłoszeniem błędu upływu czasu.
NPAD	Liczba znaków, poprzedzających każdy nadchodzący pakiet.
PADC	Znak sterujący poprzedzający pakiet.
EOL	Znak potrzebny do przerwania przesyłania pakietu.
QTCL	Znak używany do poprzedzania znaków sterujących (zwykle #).

Ponadto pakiet inicjujący może zawierać następujące dodatkowe informacje:

QBIN	Znak używany do poprzedzania znaków ośmiobitowych, kiedy bit parzystości nie może być użyty jako bit danych.
CHKT	Informacja o typie kontroli błędów: 1 - jednobajtowa suma kontrolna, 2 - dwubajtowa suma kontrolna, 3 - trzybajtowa kontrola CRC-CCITT.
REPT	Znak używany do poprzedzania znaku powtarzającego się: spacja (32) oznacza wyłącznie zliczanie powtarzających się znaków, a <i>tylda</i> (~) jest zwykle stosowanym znakiem powtarzania.
CAPAS	Maska bitowa informująca o różnych możliwościach <i>Kermit</i> 'a. Bit maski jest ustawiony wówczas, gdy dana możliwość jest dostępna.

Znak QBIN musi się różnić od znaku QTCL, przy czym odpowiedź typu Y w tym polu oznacza zgodę na cytowanie znaków ośmiobitowych, N zakazuje cytowania, natomiast *ampersand* (&) lub każdy inny znak z zakresu 33-62 albo 96-126 oznacza cytowanie właśnie tym znakiem. Znak(i) CAPAS zawierają informację o oczekiwanych możliwościach (gdy jest przysyłany przez urządzenie nadające) lub o możliwościach urządzenia odbierającego (jeśli znak ten jest przez nie wysyłany). Zwykle jest tylko jeden bajt CAPAS, ale także, przez wygaszenie najmniej znaczącego bitu pierwszego bajta, istnieje możliwość dołączenia bajta drugiego.

Po wymianie pakietów inicjujących dane wysyłane są w postaci kolejnych pakietów. Urządzenie odbierające musi na każdy z takich pakietów odpowiedzieć pakietem potwierdzającym, przy czym urządzenie nadające, przed wysłaniem kolejnego pakietu, musi oczekiwać na potwierdzenie pakietu. Gdy komputer odbierający zgłosi błąd wysyłając NAK, to nadajnik musi powtórnie przesłać pakiet. Pakiet NAK jest traktowany jako potwierdzenie nadejścia poprzedniego pakietu danych. Jeżeli pakiet danych dotrze do urządzenia odbierającego więcej niż jeden raz rzędu, to wysyła ono pakiet potwierdzający i ignoruje duplikat. Podczas jednej operacji może być przesłanych wiele plików, a razem z każdym plikiem wysyłany jest pakiet jego nagłówek, jeden lub więcej pakietów danych oraz pakiet końca pliku. Kiedy nie ma więcej plików do przesłania, to urządzenie nadawcze wysyła pakiet oznaczający koniec transmisji. Sekwencję wymiany pakietów pomiędzy urządzeniami pokazano w tab.4.2

Przesłanie pakietu EOF ze znakiem D w polu danych oznacza „*zaniechaj odbierania pliku*”, przy czym pakiet EOF powinien potwierdzony. Przesłanie pakietu ACK ze znakiem Z w polu danych oznacza „*przerwij nadawanie wszystkich plików*”. Każdy ze znaków przerywających transmisję może być dodany do zawartości pakietu (normalnie jest to znak powrotu karetki). Znaki takie nie są traktowane jako część pakietu przy obliczaniu sumy kontrolnej czy długości pakietu. Inne znaki (np. potwierdzające) mogą być również przesyłane między pakietami. Ponieważ istnieją trzy metody obliczania liczby kontrolnej, to podczas inicjacji komputery muszą „uzgodnić” metodę kontroli błędów. Jeżeli po odebraniu pakietu obliczona liczba kontrolna się nie zgadza z zawartością pola CHECK, to wysyłany jest pakiet NAK. Pakiety ACK i NAK mają również pola liczby kontrolnej.

Tab.4.2 Sekwencja wymiany pakietów pomiędzy urządzeniami

NAZWA ZDARZENIA	NADAJNIK DO ODBIORNIKA	ODBIORNIK DO NADAJNIKA
SEND INIT (wysłanie inicjacji)	Pakiet parametrów	
ACK (potwierdzenie)		Pakiet parametrów
HEADER (nagłówek pliku)	Pakiet nagłówka	
ACK		Pakiet potwierdzający
DATA (pakiet danych)	Pakiet danych	
ACK		Pakiet potwierdzający
[powrót do zdarzenia DATA, jeżeli istnieją dalsze dane do przesłania]		
EOT (koniec pliku)	Pakiet EOT	
ACK		Pakiet potwierdzający
[powrót do zdarzenia DATA, jeżeli istnieją dalsze dane do przesłania]		
EOT (koniec operacji)	Pakiet EOT	
ACK		Pakiet potwierdzający

Najprostszą do obliczenia liczbą kontrolną jest jednobajtowa suma kontrolna, która jest obliczana w następujący sposób: wartości ASCII wszystkich znaków bloku są do siebie dodawane, przy czym znaki ośmiobitowe dodaje się jako liczby ośmiobitowe, następnie bity 6 i 7 są dodawane do liczby utworzonej przez bity od 0 do 5, a uzyskany wynik jest zamieniany na znak zwykły (dający się wydrukować). Np. jeżeli s jest arytmetyczną sumą wszystkich znaków ASCII, to:

$$check = tochar(32 + ((s + ((s \text{ AND } 192)/64 \text{ AND } 63)))$$

W przypadku sumy dwubajtowej suma arytmetyczna jest obliczana jako liczba 16-bitowa, a 12 mniej znaczących bitów jest przesyłanych jako dwa znaki zamienione za pomocą funkcji *tochar()* (bity od 6 do 11 i dołączone bity od 0 do 5).

Dwubajtowa liczba kontrolna CRC jest natomiast obliczana i przesyłana jako trzy bajty tworzone po zastosowaniu funkcji *tochar()* kolejno na bitach od 12 do 15, od 6 do 11 i od 0 do 5.

W niektórych wersjach protokołu *Kermit* są jeszcze dostępne możliwości przesyłania długich pakietów oraz okienkowanie, które powodują, że protokół ten jest bardziej użyteczny.

4.3.2 LLC

Norma ANSI/IEEE 802.3 specyfikuje część warstwy łącza danych (warstwa 2. modelu ISO/OSI). Opisywana podwarstwa jest oznaczona jako MAC (ang. *Medium Access Control*). Udostępnia ona podwarstwie wyższej LLC (ang. *Logical Link Control*), opisanej w normie ANSI/IEEE 802.2, funkcje służące do nadawania i odbierania paczek danych. Podczas nadawania podwarstwa MAC konstruuje ramkę zawierającą adresy nadawcy i odbiorcy oraz dodatkowe dane do kontroli poprawności przesyłanej informacji, a następnie przekazuje ramkę do niższej warstwy w celu zamiany ciągu bitów na sygnały przenoszone przez medium transmisyjne. Podczas odbioru

podwarstwa ta zamienia ciąg bitów otrzymany od warstwy fizycznej na strukturę ramki, określa adresy nadawcy i odbiorcy oraz oblicza sumę kontrolną w celu detekcji błędu. Jeżeli adres odbiorcy jest tożsamy z adresem stacji, a transmisja przebiegła bez kolizji i błędów, to fakt skompletowania ramki jest sygnalizowany wartością statusu dla wyższych warstw. Poszczególne, przesyłane bity są kodowane kodem *Manchester*. Bity są wysyłane poczynając od mniej znaczących w bajcie. Struktura przesyłanych pakietów jest ściśle ustalona i ma postać zamieszczoną w tab.4.xxxx.

Tab.4.3 Struktura pakietów

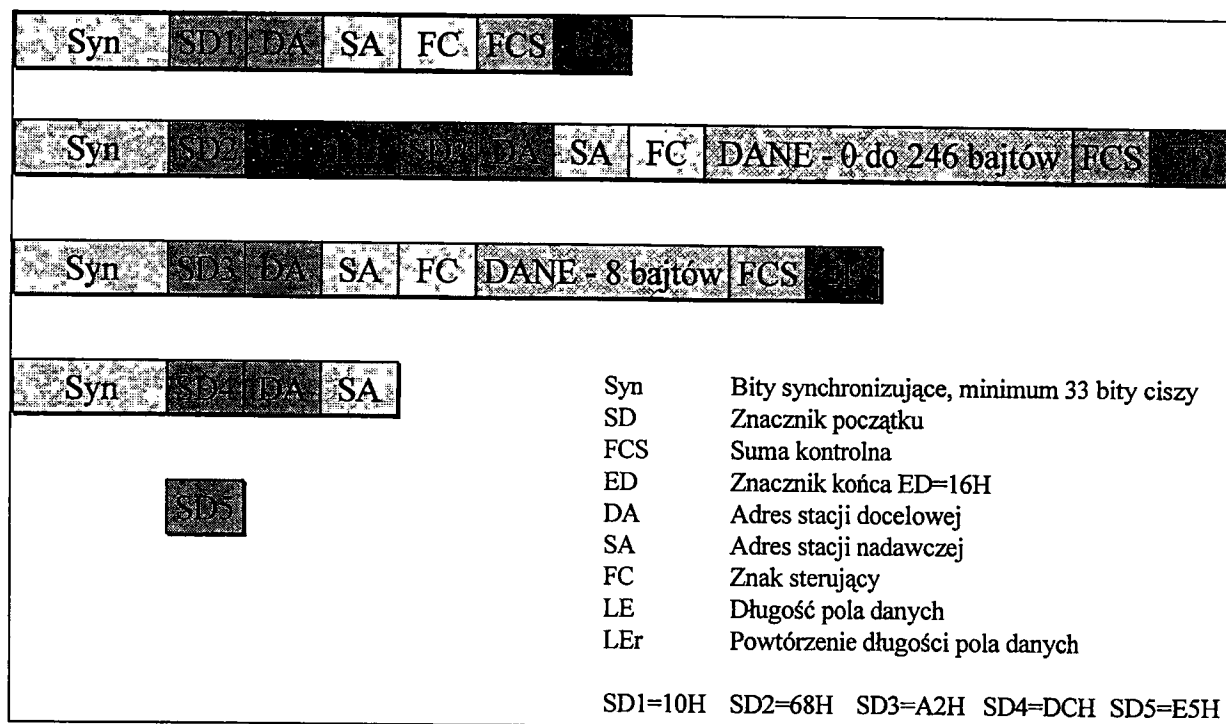
7B	<i>ciąg wstępny</i>
1B	<i>SFD</i>
2 lub 6B	<i>adres docelowy</i>
2 lub 6B	<i>adres źródłowy</i>
2B	<i>długość</i>
	<i>dane</i>
	<i>PAD</i>
4B	<i>sekwencja sumy kontrolnej</i>

Ciąg wstępny (preambuła) to 7 bajtów o wartości 55h. Po nich następuje znak SFD o wartości D5h. Adresy docelowy i źródłowy, w zależności od trybu adresowania, mogą mieć 16 lub 48 bitów. Nie jest wymagane aby każdy element sieciowy miał możliwość pracy w obu trybach adresowania. Wybór trybu adresowania zależy od implementacji i musi być identyczny w całej sieci. Dla adresów 48 bitowych dwa pierwsze bity służą do określenia tego, czy adres jest lokalny czy globalny oraz czy adresowanie obejmuje indywidualną stację czy ich grupę (*broadcast*). Pole *długość* określa natomiast długość całego pakietu. Po określeniu długości następują dane przesyłane przez warstwę kanałową. Za danymi może następować informacja dodatkowa, która jest uzupełnieniem danych wówczas, jeżeli transmitowany pakiet jest zbyt krótki (zazwyczaj jego minimalna długość to 64 bajty). Cztery ostatnie bajty to sekwencja kontrolna transmitowana w kolejności od najstarszego bitu do najmłodszego.

Za niepoprawny pakiet uważa się taki pakiet, w którym pole *długość* ma inną wartość niż długość odebranych danych, liczba bitów składających się na oktety nie jest podzielna przez 8 oraz gdy obliczona na podstawie pakietu sekwencja kontrolna nie odpowiada sekwencji otrzymanej w pakiecie. Niepoprawne pakiety nie są udostępniane wyższym warstwom. Całkowita długość pakietu w standardzie IEEE 8802.3 nie może przekroczyć 1500 bajtów. Ponieważ wcześniejsze implementacje protokołu Ethernet różnie określały wielkość pakietu, dlatego też można spotkać wiele takich kart sieciowych, które mogą przysyłać ramki większe (do 2000 bajtów), lecz ich dodatkowe możliwości nie są wykorzystywane w klasycznych sieciach standardu 802.3.

4.3.3 Ramki PROFIBUS'a

Ramki przesyłane w sieci PROFIBUS składają się z ciągów 11 bitowych znaków, które zawierają 8 bitów danych, 1 bit startu, 1 bit parzystości i 1 bit stopu (rys.4.xxx). Kolejne znaki ramki nadawane są jeden po drugim bez przerw między znakami. Znacznik początku (SD) identyfikuje typ ramki. Ramka SD1 przynosi polecenie zakodowane w polu FC i nie zawiera danych. Ramka SD2, oprócz polecenia zawiera, pole danych o zmiennej długości, a liczba znaków danych jest zapisana dwukrotnie w polach LE i LEr. Długość pola danych nie może przekroczyć 246. Ramka SD3 zawiera 8 bajtowe pole danych i bajt rozkazu. Komunikaty SD1, SD2 i SD3 mogą przynosić zarówno dane lub rozkazy węzła inicjującego, jak i odpowiedź węzła zapytanego.



Rys.4.15 Format ramki sieci PROFIBUS

Transmisja danych w Profibus'ie opiera się na UART. Nie są jednak przesyłane poszczególne znaki, lecz telegramy znaków składające się z łańcuchów znaków. Telegramy w transmisjach Profibus'a mogą być czystymi telegramami informacyjnymi bez danych lub telegramami z danymi. Telegramy informacyjne bez danych mają stałą długość 6 bajtów:

Znak startu	Adres docelowy	Adres źródła	Bajt kontrolny	FCS	Znak końcowy
-------------	----------------	--------------	----------------	-----	--------------

Zaczynają się one znakiem startu, który zawiera kod rozkazu formatu telegramu. Następnie występują adresy docelowy oraz źródła dla identyfikacji stacji nadawczej i odbiorczej. Przesyłana na zakończenie suma kontrolna FCS (*Frame Check Sequence*) służy do zabezpieczenia danych. Na końcu telegramu jest znak końcowy. Telegramy z danymi mogą być ze stałą długością danych i tym

samym również stałą długością całkowitą oraz ze zmienną długością danych. Telegramy z danymi ze stałą długością mają następującą strukturę:

Znak startu	Adres docelowy	Adres źródła	Bajt kontrolny	8 bajtów danych	FCS	Znak końcowy
-------------	----------------	--------------	----------------	-----------------	-----	--------------

Ponieważ długość danych jest stała, to nie jest wymagane żadne dodatkowe podawanie długości.

W telegramach, których długość nie jest z góry określona, musi być ona transmitowana w telegramie. Maksymalna długość telegramu Profibus'a wynosi 255 bajtów, tzn. maksymalna ilość bajtów informacji jest $n=246$:

1. znak startu	Długość	(Długość)	2. znak startu	Adres docelowy	Adres źródła
----------------	---------	-----------	----------------	----------------	--------------

...

...	Bajt kontrolny	n danych	FCS	Znak końcowy
-----	----------------	----------	-----	--------------

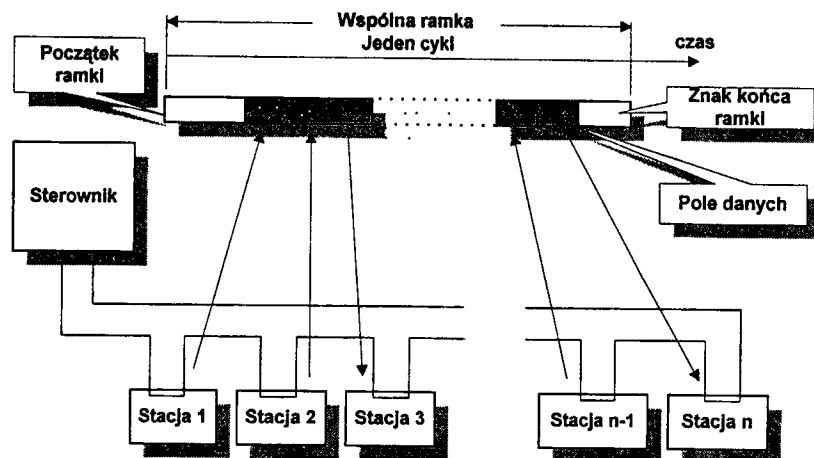
Dwukrotne przesłanie znaku startu w tego rodzaju telegramach służy do zabezpieczenia transmisji. Również podawanie długości jest powtarzane ze względu na pewność danych.

4.3.4 Sposób przesłania informacji w systemie InterBus-S

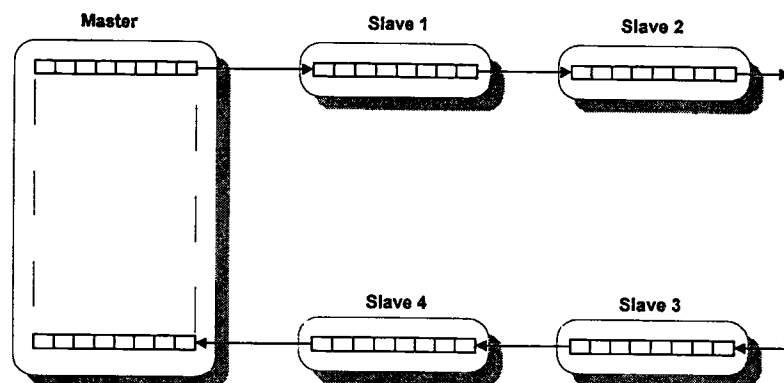
Podstawową metodą transmisji danych w systemie *InterBus-S* jest tzw. metoda całkowitej, wspólnej ramki (ang. *total frame*) (rys.4.16). Taka budowa protokołu zapewnia możliwość efektywnego i synchronicznego przesłania danych. Przez zastosowanie wspólnej ramki w protokole *InterBus'a-S* oraz fizycznej struktury pierścienia (ang. *ring*) w tym systemie (rys.4.17) uzyskano możliwość uniknięcia adresowania urządzeń włączonych w ten system. Dzięki temu podłączenie elementu do sieci odbywa się szybko i łatwo, bo bez ustawiania włączników DPI oraz przestawiania *jumper'ów*.

Każdy element systemu *InterBus-S* ma wewnętrzny rejestr przesuwany, przez który może transportować dane. Wielkość tego rejestru przesuwanego zależy od ilości danych we/wy elementu dołączonego do systemu. W każdym elemencie znajdują się także: rejestr wejściowy, wyjściowy oraz rejestr CRC (ang. *Cyclic Redundancy Check*) (rys.4.18). *Master* musi "wiedzieć" jakie elementy są włączone do systemu, dlatego też za pomocą rejestru identyfikacyjnego "dowiaduje" się on podczas cyklu identyfikacyjnego o tym jakie elementy są dołączone do systemu.

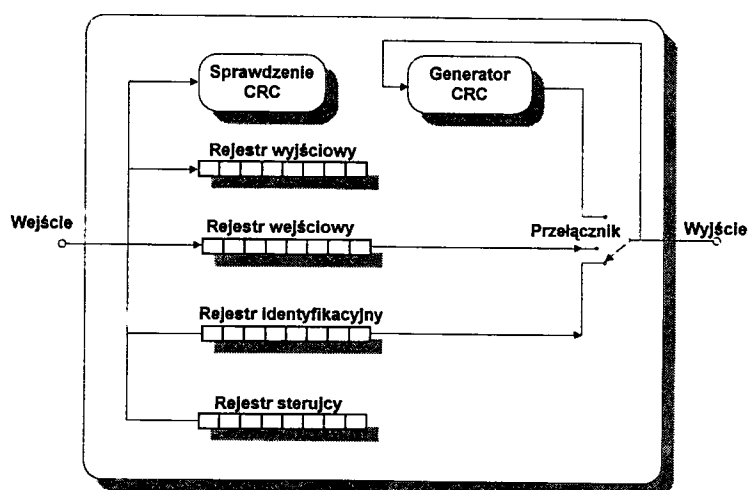
Podczas cyklu identyfikacyjnego, równoległe do rejestru identyfikacyjnego, *InterBus-S* używa rejestru sterującego, dzięki któremu *master* może sterować pojedynczym elementem. Rejestr może się znajdować w różnej fazie wykonywania działań protokołu i dlatego też jest on sterowany przez *master'a* przełącznikiem (rys.4.18).



Rys.4.16 Metoda wspólnej ramki



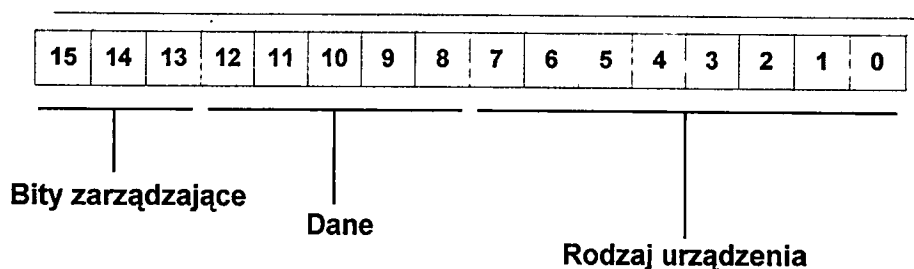
Rys.4.17 Struktura systemu *InterBus-S*



Rys.4.18 Budowa elementu systemu *InterBus-S*

Podczas startowania systemu *InterBus-S* master rozpoczyna pracę sieci cyklem identyfikacyjnym, dzięki któremu następuje rozpoznanie ile i jakie elementy są dołączone do systemu. Każdy *slave* ma 16-to bitowy rejestr identyfikacyjny w którym znajduje się kod

identyfikacyjny (*ID-kod*) (rys.4.19). Pierwsze 8 bitów (0-7) kodują rodzaj urządzenia, następne 5 bitów (8-12) kodują dane, natomiast ostatnie 3 bity (13-15) są bitami zarządzającymi.



Rys.4.19 Budowa kodu identyfikacyjnego

Za pomocą bitów zarządzających można przysyłać dynamicznie meldunki o błędach. Każdy element węzła sieci *InterBus-S* może zgłaszać *master*'owi zapotrzebowanie na rekonfigurację systemu (odbywa się to przy jakiegokolwiek ingerencji w system). Poprzez bity zarządzające mogą być sygnalizowane błędy CRC (ang. *Cyclic Redundancy Check*) oraz błędy modułów. Dla *master*'a ważne są informacje o wielkości danych zajmowanych w rejestrze elementu. Jeżeli np. moduł we/wy ma 16 bitów wejściowych i 32 bity wyjściowe, to zajmuje on 2 słowa magistrali. Za pomocą bitów danych (8 -12) kodowane są informacje o wielkości zajmowanych danych w rejestrze.

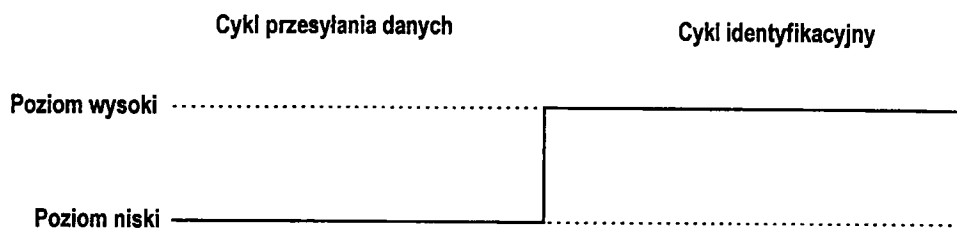
Za pomocą bitów rodzaju urządzenia (rys.4.19) *master* otrzymuje informacje o klasie i grupie urządzenia, np. czy element otrzymuje lub wysyła sygnały cyfrowe albo analogowe, czy jest to urządzenie należące do grupy elementów magistrali odległej czy lokalnej itp.

Dane informacyjne z kodu identyfikacyjnego są dla *master*'a informacjami wejściowymi z pojedynczych urządzeń. Obok tych danych każdy element w cyklu identyfikacyjnym zawiera także dane wyjściowe, które są danymi sterującymi lub danymi kontrolnymi. Za pomocą tych danych *master* może wpływać na system. Podczas cyklu identyfikacyjnego, równolegle do informacji identyfikacyjnych, przesyłane są dane sterujące. Dane sterujące są przechowywane w 16-to bitowym rejestrze sterującym (ang. *Control-Register*). Za pomocą bitu 8 i 9 *master* ma możliwość „resetowania” segmentów magistrali lokalnej i odległej. Bity od 0 - 7 oraz od 12 - 14 są natomiast zarezerwowane, a dzięki bitom 10 -11 *master* może włączyć lub wyłączyć segmenty magistrali lokalnej lub odległej.

Sygnały protokołu *InterBus*'owego są przesyłane przez dwie linie. W protokole *InterBus*'owym rozróżniamy następujące sygnały:

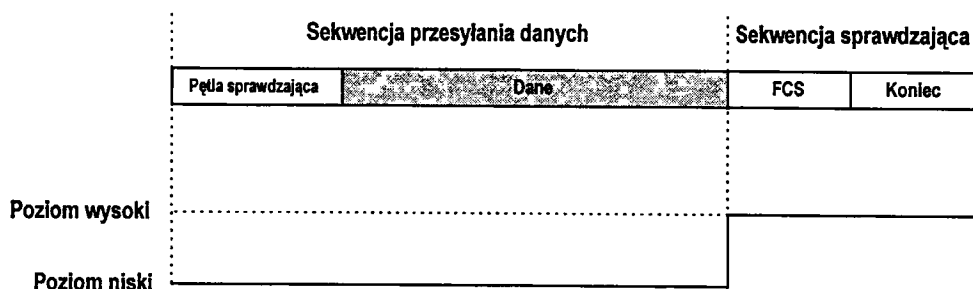
- taktowanie magistrali,
- dane,
- sygnały selekcyjne,
- sygnały kontrolne oraz
- sygnały restartujące.

Za pomocą sygnału selekcyjnego *master* może sterować wyborem cyklu przesyłania danych lub cyklem identyfikacyjnym. W cyklu przesyłania danych sygnał selekcyjny ma poziom niski w przeciwieństwie do cyklu identyfikacyjnego, gdzie ma on poziom wysoki (rys.4.20).



Rys.4.20 Status sygnału selekcyjnego

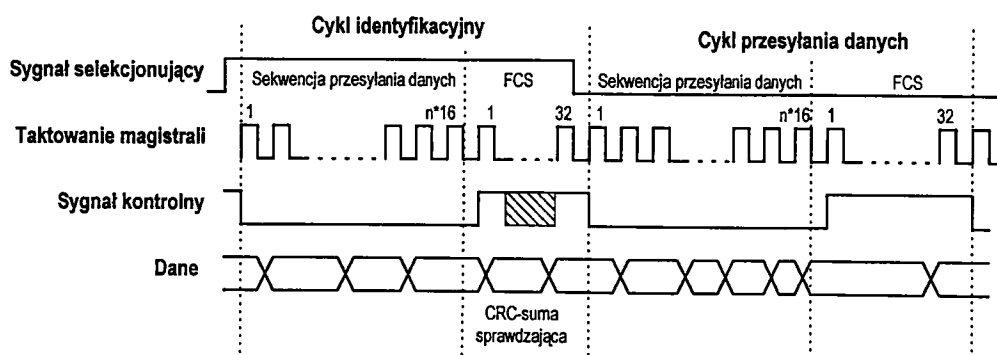
Poziom sygnału selekcyjnego może być zmieniony przez *master'a* systemu *InterBus-S* tylko pomiędzy dwoma cyklami. Za pomocą sygnału kontrolnego *master* rozróżnia fazę przesyłania danych oraz fazę CRC. Sygnał kontrolny ma poziom niski przy sekwencji przesyłania danych, a wysoki przy sekwencji CRC (rys.4.21).



Rys.4.21 Zestawienie sygnału kontrolnego z ramką sumacyjną protokołu *InterBus-S*

Za pomocą sygnałów kontrolnych elementy systemu *InterBus-S* mają możliwość zasygnalizowania błędów CRC.

W protokole *InterBus-S* rozróżniamy cykl przesyłania danych oraz cykl identyfikacyjny. Wewnątrz obu cykli istnieją blok danych oraz blok kontrolny FCS (ang. *Frame Check Sequence*) (rys.4.22).



Rys.4.22 Przebieg sygnałów w protokole *InterBus-S*

W bloku danych znajdują się informacje z *master*'a do elementów systemu *InterBus-S* oraz z poszczególnych elementów do *master*'a. W bloku kontrolnym przechowywana jest natomiast informacja o miejscu wystąpienia błędu podczas przesyłania danych. Na początku cyklu *master* wybiera za pomocą sygnału SL pomiędzy cyklem identyfikacyjnym a cyklem przesyłania danych. Pojedynczy element systemu przełączany jest albo do rejestru identyfikacyjnego albo do rejestru wejściowego i wyjściowego w zależności od stanu sygnału SL.

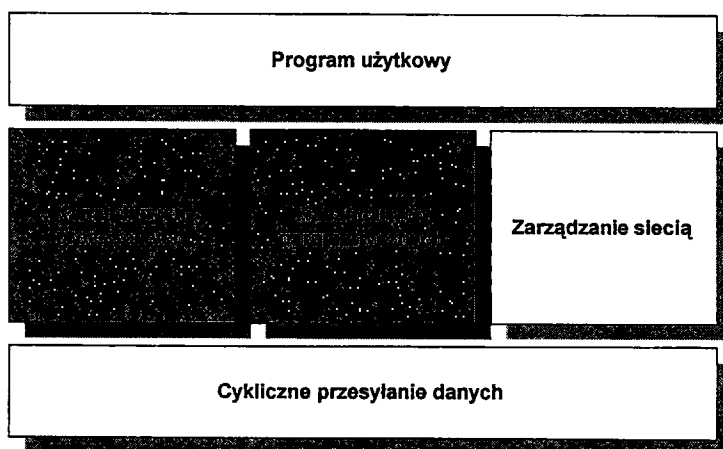
Na podstawie informacji z cyklu identyfikacyjnego *master* uzyskuje informacje o ilości danych (ogółem) znajdujących się w systemie *InterBus-S*. Do kontroli ilości tych danych a także wielkości rejestrów przesuwanych, *master* używa wysyłanego na początku cyklu tzw. *Loopbackwort*. Jest to 16-to bitowe pole ramki po którym *master* wysyła informacje wyjściowe do dołączonych elementów systemu.

Metoda wspólnej ramki (ang. *total frame*) jest bardzo efektywna przy przesyłaniu danych procesowych. Aby jednak zapewnić możliwość transmisji danych parametrycznych, pozostawiono odpowiednie miejsca (luki) w ramce, tzn. od 2 do 16 bajtów dla urządzeń, które przesyłają tego typu dane.

Aby transmisja w *InterBus*'ie-S była efektywniejsza, ma on dwa niezależne kanały transmisyjne (rys.4.23):

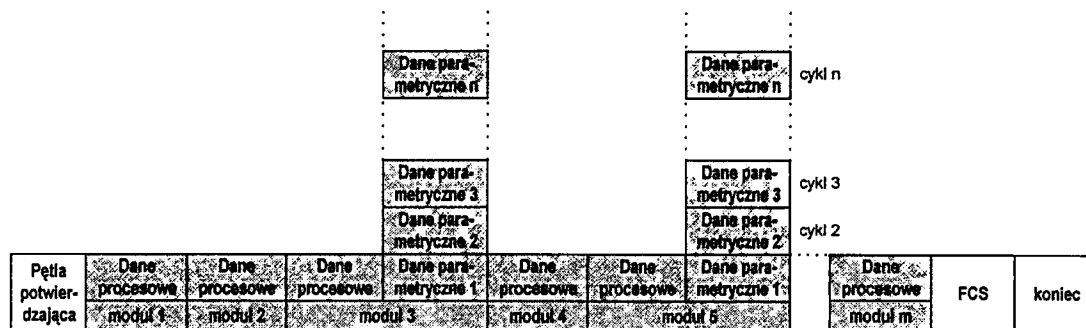
- kanał danych procesowych oraz
- kanał danych parametrycznych.

Ta dwukanałowa struktura może być zrealizowana we wszystkich elementach systemu *InterBus-S*.



Rys.4.23 Kanał danych procesowych i danych parametrycznych w *InterBus-S*

Nie jest jednak konieczne wykorzystywanie obu kanałów przez każdy z elementów. Wydajność magistrali może być bowiem dopasowana do funkcjonalności każdego urządzenia. Przy konfiguracji systemu należy natomiast zdefiniować rodzaj przesyłanych danych. I tak na rys.4.24 przedstawiono przykład konfiguracji systemu *InterBus-S*.

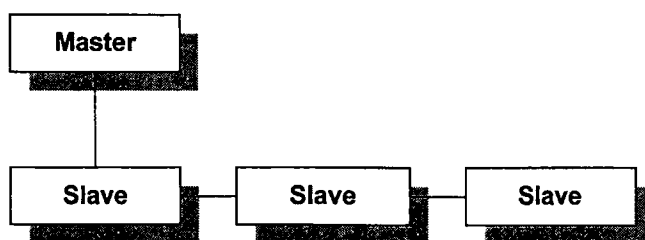


Rys.4.24 Zasada przesyłania danych przez protokół *InterBus-S*

Moduły 1, 2, 4 przesyłają tylko dane procesowe, natomiast moduły 3 i 5, oprócz danych procesowych, zawierają także dane parametryczne. Przy danych parametrycznych o długości większej od wielkości pola w które można je wpisać, cały blok danych dzielony jest na mniejsze segmenty, a następnie informacja przesłana jest do odbiorcy część po części w protokole cyklicznym. Tam zostaje ona złożona ponownie w jedną całość. Przy każdym uaktualnieniu danych procesowych zostaje także przesłany jeden segment danych parametrycznych. Taki sposób przesyłania danych nie wpływa na cykliczność przesyłania danych procesowych. Złożenie informacji podzielonej na bloki jest realizowane za pomocą dodatkowego protokołu *software*'owego warstwy łącza danych zwanego Peryferyjnym Protokołem Komunikacyjnym (ang. *Peripherals Communication Protocol - PCP*), który nie jest widoczny dla użytkownika.

InterBus-S wykorzystuje do przesyłania danych metodę *master-slave*. Całkowita sieć *InterBus*'owa może składać się z jednego *master*'a i maksymalnie 512 *slave*'ów. *Master* wysyła do wszystkich elementów sieci dane i odbiera jednocześnie informacje zwrotne. Elementy sieci, które są związane z *master*'em poprzez magistralę, nazywane są *slave*'mi (rys.4.25).

Rys.4.25 Struktura protokołu PCP (*Peripherals Communication Protocol*)



Rys.4.25 Struktura *master - slave* w *InterBus*'ie-S

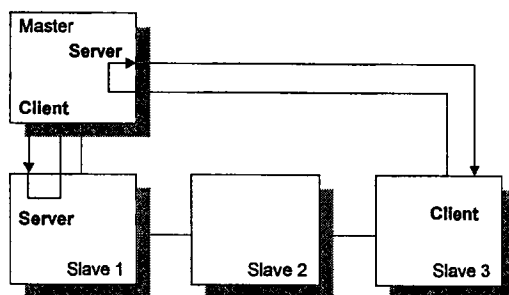
Zasada działania *master-slave* opisuje sposób dostępu do magistrali w którym *master* wysyła zapytania do *slave*'ów, a one odpowiadają. *Master* kontroluje przesyłanie wszystkich danych. *InterBus-S* używa takiego samego mechanizmu dostępu do magistrali. *InterBus-S* ma przy tym strukturę pierścienia (*ring*), w którym wszystkie dane przesyłane są przez rejestr przesuwany. Dane parametryczne logicznie przesyłane są punkt - punkt (ang. *point to point*) pomiędzy dwoma

elementami. Jest to model przesyłania danych typu *client-server*. *Client* wysyła zapotrzebowanie do innego elementu w sieci, a ten uruchamia proces *server*'a i kieruje potrzebną usługę do *client*'a.

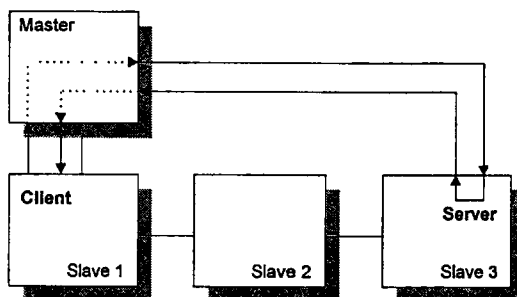
W *InterBus*'ie-S element może przysyłać informację poprzez kanał danych parametrycznych i mogą być przesyłane nie tylko funkcje *client*'a, ale także funkcje *server*'a. W *InterBus*'ie-S wyróżnia się komunikację z *czystym mechanizmem client-server* oraz komunikację z *elementem pośredniczącym*. Przy komunikacji z *czystym mechanizmem client-server* występuje struktura w której *master* realizuje zarówno funkcje *client*'a jak i funkcje *server*'a. *InterBus*-S oferuje także możliwość przesyłania danych parametrycznych także pomiędzy dwoma *slave*'ami.

Na rysunku 4.26 pokazano sposób przesyłania danych z *czystym mechanizmem client-server*. *Slave 3* uruchamia proces *client*'a i wysyła zapotrzebowanie na usługę do *master*'a. Ten uruchamia proces *client*'a do elementu *slave 1*. *Slave 1* jako *server* udziela usługi *master*'owi. Natomiast *master* jako *server* odpowiada *slave*'owi 3. Jak z tego widać, są tu uruchomione dwa mechanizmy typu *master-slave* i jest to struktura przejrzysta. W przesyłaniu danych z *elementem pośredniczącym* mamy także do czynienia ze strukturą *client-server*, ale nie jest to do końca czysty mechanizm.

Pomiędzy procesem *client*'a, który działa na *slave*'ie 1 (rys.4.27), a *server*'em występuje *master*. Logicznie połączenie pomiędzy dwoma *slave*'mi (*client*'em i *server*'em) jest realizowane wprawdzie bezpośrednio, to jednak fizycznie odbywa się ono przez *master*'a.



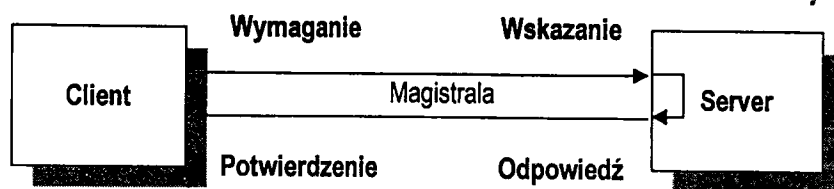
Rys.4.26 Przesyłanie danych z *czystym mechanizmem client-server*



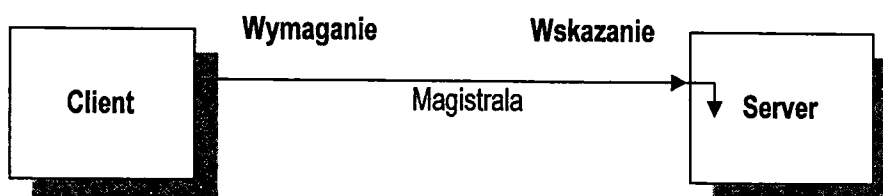
Rys.4.27 Przesyłanie danych z *elementem pośredniczącym*

InterBus-S przesyła dane parametryczne przy użyciu usług PCP z potwierdzeniami. Odbywa się to w sposób następujący (rys.4.28):

wymaganie *client*'a dochodzi do *server*'a w postaci wskazania, po czym *server* oddaje wymaganą wartość do *client*'a, który następnie potwierdza otrzymaną odpowiedź.



Rys.4.28 Przesyłanie danych z potwierdzeniem



Rys.4.29 Przesyłanie danych bez potwierdzenia

W *InterBus*'ie-S używane są także usługi bez potwierdzenia (rys.4.29). Są to usługi bez zwrotnej informacji o otrzymaniu odpowiedzi. Przykładem takiej usługi może być zerwanie połączenia (ang. *abort*).

II. Sieci typu *Fieldbus* oraz sieci komputerowe

5. Sieci miejscowe typu *Fieldbus*

5.1 Standard PROFIBUS

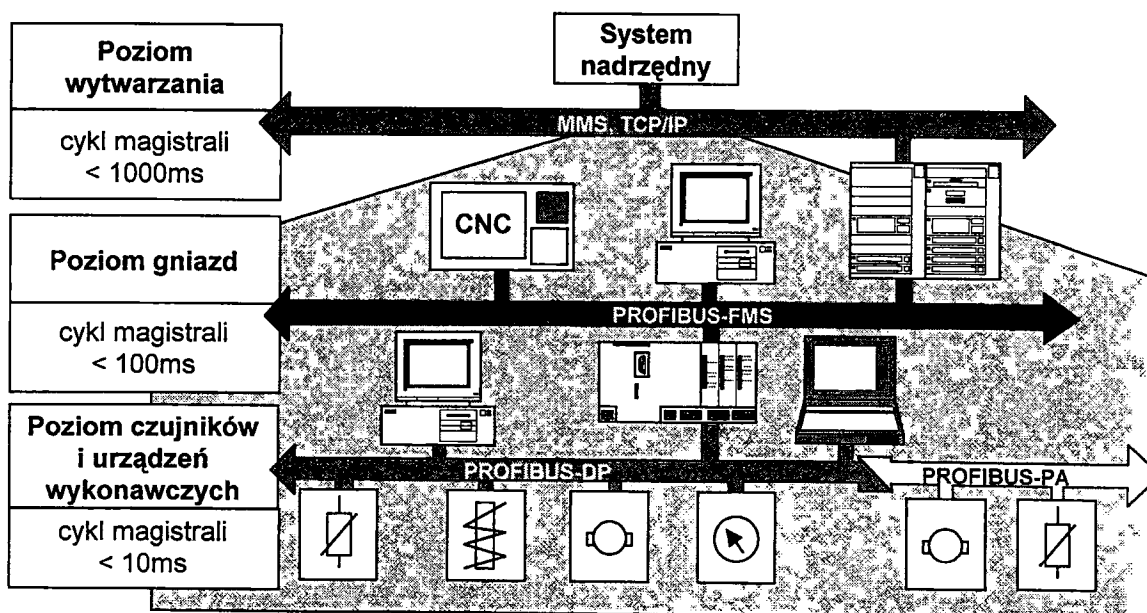
5.1.1 Wstęp

Sieć PROFIBUS zrodziła się z inicjatywy trzech firm: BOSCH, Klöckner-Moeller i SIEMENS, które w 1987 r. postanowiły uruchomić wspólny projekt badawczy nad szybką, otwartą siecią typu *Fieldbus*. Projekt ten był dotowany przez niemieckie Ministerstwo Badań i Technologii. Wzięło w nim udział pięć instytutów i 13 firm przemysłowych. W rezultacie tych badań w 1991 r. ustanowiono złożoną z dwóch części normę DIN 19245. PROFIBUS, obok FIP'a i P-Net'a, został następnie przyjęty jako standard europejski i znalazł się w uchwalonej 1996 r. normie EN 50170, co zapewnia jego otwartość, stabilność i niezależność od producenta.

Przez uproszczenie ogólnej architektury i zastosowanie tanich technik transmisyjnych (RS485 i skrętka jednoparowa) obniżono koszty sieci PROFIBUS w stosunku do sieci standardów IEEE 802 i MAP. PROFIBUS ma zastosowanie zarówno tam gdzie potrzebna jest wysoka szybkość transmisji a przepływ danych jest czasowo krytyczny, jak i w poważnych kompleksowych zadaniach komunikacyjnych.

5.1.2 Rodzina sieci PROFIBUS

Rodzina sieci PROFIBUS składa się z trzech zgodnych wersji ukierunkowanych na różne zastosowania (rys.5.1.1).



Rys.5.1.1 Rodzina sieci PROFIBUS na tle poziomów w przedsiębiorstwie

PROFIBUS – DP. Ta wersja PROFIBUS’a jest zoptymalizowana pod względem szybkości transmisji i jest dedykowana do komunikacji pomiędzy kontrolnymi systemami automatyki a rozproszonymi układami wejścia – wyjścia. Eliminuje ona stosowanie sygnałów obiektowych 24 V lub $0 \div 20$ mA.

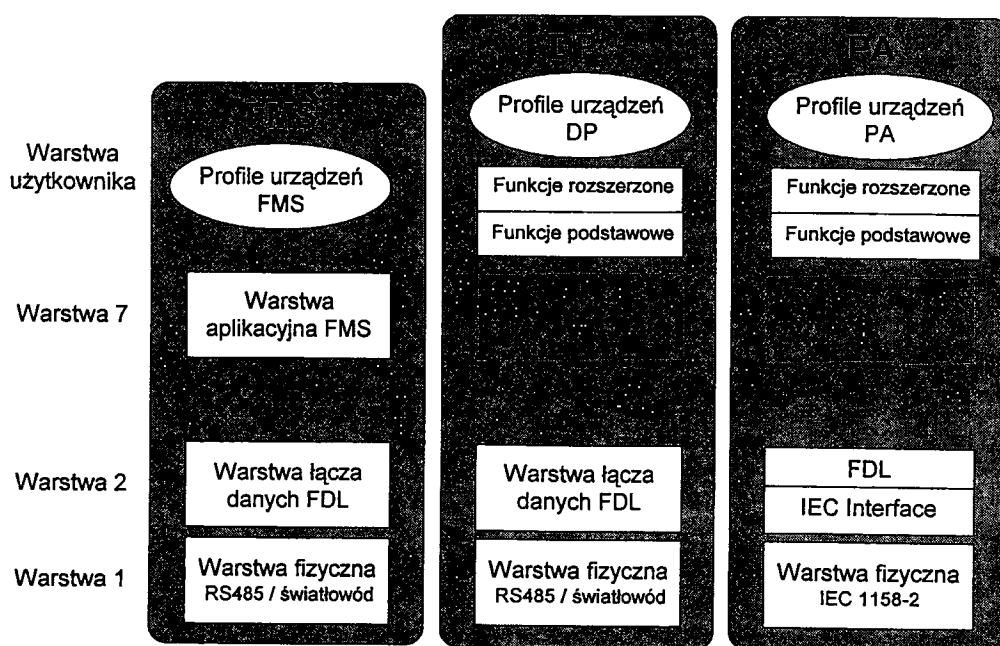
PROFIBUS – PA. Wersja ta jest przeznaczona do zastosowania w automatyzacji procesów produkcyjnych i pozwala ona na podłączenie czujników i urządzeń wykonawczych za pomocą wspólnej magistrali w obszarach zagrożonych wybuchem. Dzięki zastosowaniu techniki transmisyjnej standaryzowanej w normie IEC 1158-2 za pomocą dwuprzewodowego okablowania możliwe jest przesyłanie zarówno danych komunikacyjnych jak i zasilania.

PROFIBUS – FMS. Jest to uniwersalna wersja sieci dla poziomu gniazd, realizuje ona cykliczne i acykliczne zadania przekazu danych między takimi inteligentnymi urządzeniami obiektowymi jak sterowniki PLC, komputery przemysłowe itp. Bogata funkcjonalność usług FMS pozwala na budowanie złożonych i elastycznych aplikacji.

5.1.3 Charakterystyka ogólna

5.1.3.1 Architektura protokołu

Sieć PROFIBUS jest oparta na sprawdzonych standardach międzynarodowych. Architektura protokołu opiera się na warstwowym modelu odniesienia OSI (*Open System Interconnection*), zgodnym z międzynarodowym standardem ISO 7498. W modelu tym wszystkie warstwy transmisyjne mają precyzyjnie zdefiniowane zadania oraz interfejsy wejściowe i wyjściowe. Warstwa pierwsza (fizyczna) definiuje fizyczne charakterystyki transmisji, warstwa druga (łącza danych) określa protokół dostępu do magistrali, natomiast warstwa siódma (aplikacyjna) definiuje funkcje aplikacyjne. Architekturę protokołu PROFIBUS przedstawia rys.5.1.2.



Rys.5.1.2 Architektura protokołu PROFIBUS

PROFIBUS DP wykorzystuje warstwę fizyczną (1), łącza danych (2) oraz interfejs użytkownika, natomiast warstwy od 3 do 7 nie są zdefiniowane. *Direct Data Link Mapper* (DDLM) dostarcza interfejsowi użytkownika łatwego dostępu do warstwy drugiej. Z powodu różnorodności urządzeń PROFIBUS'a DP, zarówno funkcje aplikacyjne dostępne dla użytkownika, jak i systemu oraz zachowanie urządzenia są zdefiniowane w interfejsie użytkownika. Dla PROFIBUS'a DP stosuje się transmisję wg standardu RS 485 lub światłowodową.

W PROFIBUS'ie FMS także są zdefiniowane warstwy 1, 2 oraz 7. Warstwa aplikacyjna składa się z dwóch podwarstw: FMS (*Fieldbus Message Specification*) i LLI (*Low Layer Interface*). FMS obejmuje protokół aplikacyjny, dostarczający szerokiego wyboru usług komunikacyjnych. LLI uruchamia natomiast różnorodne powiązania komunikacyjne oraz dostarcza warstwie FMS niezależnego od urządzenia dostępu do warstwy 2. Warstwa 2 FDL (ang. *Fieldbus Data Link*) steruje dostępem do magistrali i zabezpiecza przesyłane informacje. Dla PROFIBUS'a FMS także stosuje się transmisję według standardu RS 485 lub światłowodową.

PROFIBUS PA - używa rozszerzonego protokołu PROFIBUS DP. Wykorzystywane są także profile PA, które definiują zachowanie urządzenia obiektowego. Technologia transmisji jest zgodna z IEC 1158-2, co pozwala na pracę w strefach chronionych i zasilanie urządzeń przez magistralę. PROFIBUS PA można również zintegrować z PROFIBUS'em DP, używając do tego celu interfejsu międzysegmentowego.

Ponieważ PROFIBUS DP i PROFIBUS FMS wykorzystują tę samą technikę transmisyjną i jednakowy protokół dostępu do medium, dlatego też obie wersje mogą pracować jednocześnie z tym samym medium transmisyjnym.

5.1.3.2 Technologia transmisji

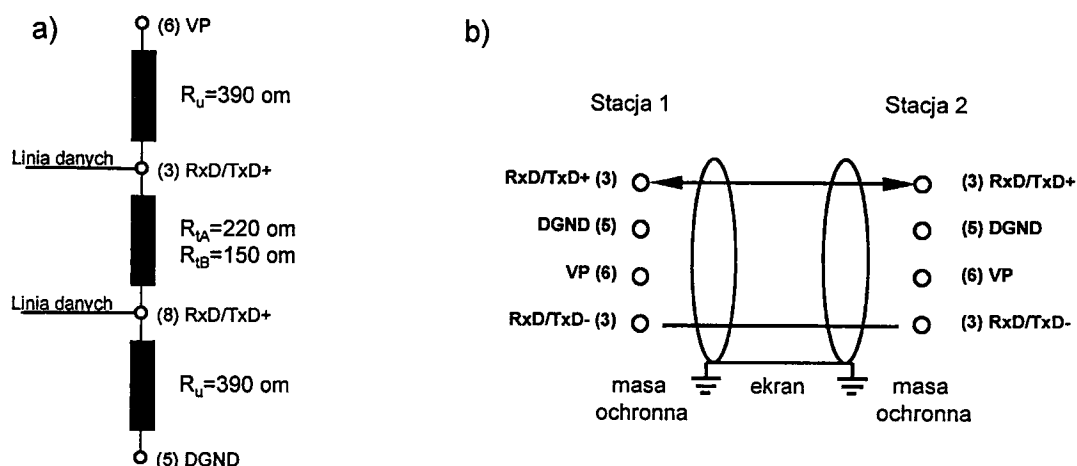
Obszar zastosowań systemów komunikacyjnych z sieciami miejscowymi w dużym stopniu jest zdeterminowany przez używaną technologię transmisji. Do podstawowych kryteriów wyboru rozwiązania należą: ochrona przesyłanych informacji, szybkość transmisji, odległość na którą można przesyłać informację i iskrobezpieczeństwo. PROFIBUS stosuje trzy podstawowe techniki transmisyjne:

- RS 485 dla PROFIBUS DP/FMS,
- IEC 1158-2 dla PROFIBUS PA,
- Światłowód dla PROFIBUS DP/FMS.

5.1.3.3 RS485 - PROFIBUS-DP/FMS

RS 485 jest podstawową technologią transmisyjną dla PROFIBUS'a, często określaną jako H2 (rys.5.1.3). Stosowanie tej technologii transmisji jest wskazane wszędzie tam, gdzie potrzebna jest duża prędkość transmisji, nie występuje wysoki poziom zakłóceń elektromagnetycznych, a obiekt nie jest zagrożony wybuchem. Wyróżnia się 6 podstawowych szybkości transmisji: 9.6 kbit/s, 19.2 kbit/s, 93.75 kbit/s, 187.5 kbit/s, 500 kbit/s oraz 12 Mbit/s. Dane kodowane są metodą NRZ

sygnałami kompatybilnymi ze standardem EIA RS-485, a każda ze stacji jest izolowana galwanicznie od linii przesyłowych.



Rys.5.1.3 Medium transmisyjne: a) terminacja, b) wyprowadzenia wtyczki D-SUB 9

Jako medium transmisyjne stosowana jest miedziana skrętka jednoparowa w ekranie (rys.5.1.3). Wyróżniane są dwa typy okablowania A i B, których parametry przedstawione są w tab.5.1.1.

Tab.5.1.1 Parametry okablowania sieci PROFIBUS

Parametry kabla	Typ A	Typ B
Impedancja	135 do 165 Ω ($f=3$ do 20 MHz)	100 do 130 ($f=100$ kHz)
Pojemność	< 30 pF/m	< 60 pF/m
Rezystancja pętli	110 Ω /km	-
Przekrój przewodnika	$\geq 0,64$ mm ² (AWG 22)	$\geq 0,22$ mm ² (AWG 24)

Maksymalna odległość na jaką można przesyłać informacje zależy od stosowanego typu okablowania i szybkości transmisji. Zależności te przedstawia tabela 5.1.2.

Tab.5.1.2 Zależność odległości od szybkości transmisji

Szybkość transmisji [kbit/s]	9.6	19.2	93.75	187.5	500	1500	12000
Długość segmentu z okablowaniem typu A [m]	1200	1200	1200	1000	400	200	100
Długość segmentu z okablowaniem typu B [m]	1200	1200	1200	600	200	70	-

Podstawową strukturą sieci jest liniowy segment kabla, zakończony na obu końcach aktywnymi terminatorami. Schemat terminatora i wartości rezystorów dla okablowania typu A i B przedstawiono na rys.5.1.3 a. Maksymalna liczba węzłów sieci dołączonych do segmentu jest

ograniczona do 32. Do dołączenia stacji do sieci używane jest łącze D-SUB 9. Przyporządkowanie wyprowadzeń we wtyczce przedstawiono na rys.5.1.3 b.

Doprowadzenie od kabla głównego może mieć długość 0,3 m, jednak nie zaleca się jego stosowania przy prędkości transmisji większej niż 500 kbit/s.

5.1.3.4 IEC 1158-2 – PROFIBUS-PA

IEC 1158-2 jest techniką transmisyjną stosowaną w PROFIBUS'ie PA, która jest przeznaczona głównie dla przemysłu chemicznego i petrochemicznego. Urządzenia stosujące tę technikę mogą pracować w obszarach zagrożonych wybuchem. Wersja ta zapewnia zasilanie stacji przez magistralę, a dzięki zarówno ograniczeniu energii zasilania doprowadzonego do stacji, jak i energii sygnału spełnia wymagania iskrobezpieczeństwa. Dane transmitowane są synchronicznie z szybkością 31.25 kbit/s, a kodowane metodą *Manchester* w trybie prądowym.

Każda ze stacji pobiera z zasilacza sieci podstawowy prąd o wartości minimalnej 10 mA, który służy do jej zasilania. W czasie nadawania urządzenie moduluje swój prąd podstawowy o wartości ± 9 mA.

Transmisja w sieci jest realizowana według następujących zasad:

- każdy segment ma tylko jedno źródło zasilania – zasilacz sieciowy,
- moc nie jest dostarczana do sieci w tym czasie, gdy stacja nadaje,
- każde urządzenie obiektowe pobiera stały, podstawowy prąd w stanie ustalonym,
- każde urządzenie działa jako obciążenie bierne,
- topologia magistrali może być liniowa, gwiazdista lub drzewiasta,
- oba końce głównej magistrali muszą być zakończone biernymi terminatorami, które składają się z szeregowo połączonego rezystora o rezystancji 100 Ω i kondensatora o pojemności 1 μ F.

Podłączenie stacji do magistrali nie uwzględnia biegunowości dołączeń, gdyż urządzenia te zawierają układ automatycznej zmiany polaryzacji. Do transmisji zalecany jest kabel o parametrach przedstawionych w tab.5.1.3.

Tab.5.1.3 Parametry kabla sieci PROFIBUS-PA

Rodzaj kabla	Skłętka jednoparowa, ekranowana
Przekrój przewodnika (nominalny)	0,8 mm ² (AWG 18)
Rezystancja pętli	44 Ω /km
Impedancja (f=31.25 kHz)	100 $\Omega \pm 20\%$
Tłumienie (f=39 kHz)	3 dB/km
Pojemność asymetryczna	2 nF/km

Liczba stacji dołączonych do segmentu jest ograniczona do 32. Wartość ta może ulec zmniejszeniu przez wymaganą klasę ochrony i ograniczenie mocy zasilacza sieciowego (tab.5.1.3).

Tab.5.1.4 Liczba stacji dołączonych do segmentu

Typ	Obszar zastosowania	Napięcie zasilania	Maksymalny całkowity prąd	Maksymalna moc	Typowa* liczba stacji
I	EEx ia/ib IIC	13,5 V	110 mA	1,8 W	8
II	EEx ib II C	13,5 V	110 mA	1,8 W	8
III	EEx ib II C	13,5 V	250 mA	4,2 W	22
IV	Bez wew. ochrony	24 V	500 mA	12 W	32

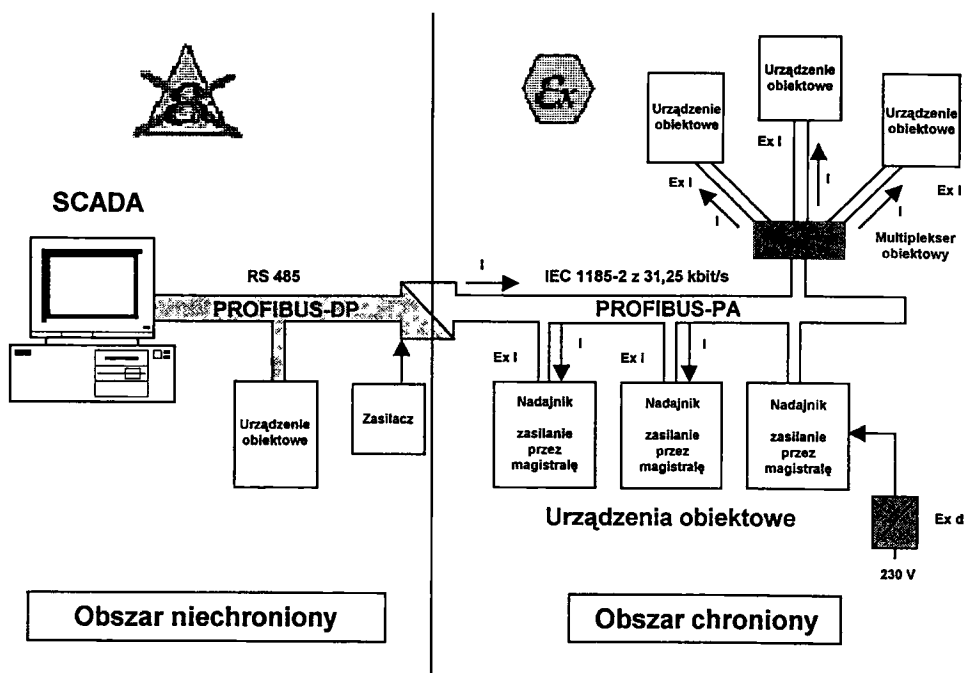
*Liczba stacji przy założeniu pobierania przez każdą ze stacji prądu o wartości 10 mA. Jeśli któraś ze stacji pobiera większy prąd, to liczba stacji ulega zmniejszeniu.

Maksymalna długość segmentu magistrali zależy od stosowanego kabla i typu zasilacza. Zależność tę przedstawia tab.5.1.5.

Tab.5.1.5 Maksymalna długość segmentu

Zasilacz		Typ I	Typ II	Typ III	Typ IV	Typ V	Typ VI
Napięcie	V	13,5	13,5	13,5	24	24	24
Prąd	mA	≤ 110	≤ 110	≤ 250	≤ 110	≤ 250	≤ 500
Długość linii dla A=0,8 mm ²	m	≤ 900	≤ 900	≤ 400	≤ 1900	≤ 1300	≤ 650
Długość linii dla A=1,5 mm ²	m	≤ 1000	≤ 1500	≤ 500	≤ 1900	≤ 1900	≤ 1900

Segmenty można łączyć za pomocą maksymalnie 4 *repeater*’ów. Dopuszczalne jest zasilanie urządzeń dołączonych do sieci w obszarze chronionym przez lokalne źródło zasilania, pod warunkiem odizolowania tego urządzenia od magistrali zgodnie z normą EN 50 020.



Rys.5.1.4 Typowa konfiguracja systemu w automatyzacji procesu

Ponieważ w PROFIBUS'ie PA najczęściej urządzenia aktywne (sterownik, system monitorujący, konsola operatorska) znajdują się w pokoju kontrolnym, dlatego też stosuje transmisję wg technologii RS 485.

Połączenie z urządzeniami pracującymi w strefie chronionej ze standardem IEC 1158-2 odbywa się za pomocą interfejsu międzysegmentowego (rys.5.1.4).

5.1.3.5 Światłowód

Użycie światłowodu umożliwia stosowanie PROFIBUS'a w środowisku o bardzo dużych zakłóceniach elektromagnetycznych oraz osiąganie dużych odległości przesyłu przy wysokich prędkościach transmisji. Dostępne są dwa typy światłowodów:

- światłowód z tworzywa sztucznego – może on być stosowany do odległości nie przekraczających 50 m,
- światłowód szklany – może być stosowany do odległości nie przekraczających 1 km.

Dostępnych jest wiele specjalizowanych urządzeń, w których zintegrowana jest konwersja sygnałów standardu RS 485 do standardu łączy światłowodowych i odwrotnie. Dzięki temu można w prosty sposób „przechodzić” w instalacjach pomiędzy RS 485 a światłowodem.

5.1.3.6 Dostęp do magistrali

Wszystkie trzy wersje PROFIBUS'a używają jednego protokołu dostępu do magistrali. Protokół ten zaimplementowany jest w drugiej warstwie modelu referencyjnego ISO/OSI. W PROFIBUS'ie warstwa druga jest oznaczana skrótem FDL (ang. *Fieldbus Data Link*). Wypełnia ona zadania protokołu komunikacyjnego wraz z tworzeniem i obsługą przesyłek. Jej podwarstwa - MAC (ang. *Medium Access Control*) określa, kiedy stacja może nadawać i zapewnia, że tylko jedna stacja nadaje w danej chwili. Protokół PROFIBUS łączy dwa zasadnicze wymagania podwarstwy MAC:

- w przypadku komunikacji między złożonymi urządzeniami automatyki, o równych prawach dostępu do medium, należy zapewnić każdej stacji warunki do wykonywania jej zadań komunikacyjnych w ściśle określonym przedziale czasu,
- w przypadku komunikacji między złożonym urządzeniem centralnym (*master*) a prostym urządzeniem oddalonym (*slave*) wymiana danych powinna być cykliczna, następować jak najszybciej i przy użyciu jak najprostszych środków.

Protokół dostępu do mediów zastosowany w sieci PROFIBUS ma charakter hybrydowy, gdyż zawiera mechanizm przekazywania uprawnień dostępu między stacjami *master* i mechanizm obiegu cyklicznego (odpytywania) *master-slave* do komunikacji z urządzeniami typu *slave*. Metoda przekazywania uprawnień polega na przydzielaniu dostępu do magistrali, na ściśle określony czas, za pomocą przekazywania między stacjami *master* specjalnej przesyłki zwanej *token*'em. Posiadanie *token*'a uprawnia do inicjowania przekazu danych. *Token* jest przekazywany między wszystkimi stacjami *master*, co tworzy logiczny łańcuch uprawnień. Metoda

cyklicznego/acyklicznego odpytywania *master-slave* pozwala urządzeniu *master*, które posiada prawa do transmisji danych na komunikację z urządzeniami typu *slave*.

Hybrydowa zasada dostępu do mediów pozwala na następujące konfiguracje systemu:

- czysty *master – slave*,
- czysty *master – master* oraz
- łączenie obu możliwości.

Warstwa łącza danych udostępnia użytkownikom 4 rodzaje usług: nadanie danych z potwierdzeniem, nadanie danych bez potwierdzenia, nadanie danych i żądanie danych zwrotnych oraz cykliczne nadawanie i żądanie danych jako odpowiedzi.

SDA (*Send Data with Acknowledge*) – **nadanie danych z potwierdzeniem**. Usługa ta jest oferowana tylko w wersji PROFIBUS FMS. Wykonanie tej usługi umożliwia wysłanie komunikatu akcji, który zawiera dane do stacji docelowej oraz odebranie pozytywnego lub negatywnego potwierdzenia odbioru. Warstwa 2 stacji docelowej przesyła pozytywne potwierdzenie po bezbłędnym odebraniu komunikatu i przekazaniu danych do wyższej warstwy. Brak potwierdzenia, potwierdzenie negatywne lub błąd w odbiorze odpowiedzi powoduje retransmisję komunikatu przez stację macierzystą.

SDN (*Send Data with No acknowledge*) – **nadanie danych bez potwierdzenia**. Usługa ta jest oferowana we wszystkich wersjach PROFIBUS'a. Wykonanie tej usługi umożliwia wysłanie komunikatu akcji zawierającego dane do stacji docelowej, grupy użytkowników (*multicast*) lub wszystkich użytkowników sieci (*broadcast*). Dla transmisji bez potwierdzeń nie jest gwarantowana poprawność przekazu komunikatu, brak potwierdzeń odbioru i nie jest wykonywana retransmisja.

SRD (*Send and Request Data with reply*) – usługa ta jest oferowana we wszystkich wersjach PROFIBUS'a. Wykonanie tej usługi umożliwia wysłanie komunikatu akcji zawierającego dane i w odpowiedzi odebranie od stacji docelowej komunikatu zawierającego wcześniej przygotowane informacje. Warstwa 2 stacji docelowej przesyła odpowiedź po bezbłędnym odebraniu komunikatu i przekazaniu danych do warstwy wyższej. Brak potwierdzenia, potwierdzenie negatywne lub błąd w odbiorze odpowiedzi, powoduje retransmisję komunikatu przez stację macierzystą.

CSRD (*Cyclic Send and Request Data with reply*) – usługa ta jest oferowana tylko w wersji PROFIBUS FMS. Wykonanie tej usługi umożliwia cykliczne odpytywanie jednej lub wielu stacji docelowych. Proces ten polega na wysłaniu komunikatu akcji i odebraniu w odpowiedzi od stacji docelowej komunikatu zawierającego dane. Na poziomie sieci wymiana komunikatów z każdą stacją realizowana jest jako odrębna transakcja nie priorytetowa. Sposób wymiany komunikatów akcji jest taki sam jak w przypadku usługi SRD.

Przed wywołaniem usługi użytkownik inicjujący musi zdefiniować dane przeznaczone do wysłania do stacji docelowych oraz określić listę kolejności odpytywania, która zawiera adresy i porty odpytywanych stacji. Jedna stacja może znajdować się w kilku miejscach na liście. Proces odpytywania jest kontynuowany do czasu jego wyłączenia.

5.1.4 PROFIBUS-DP

PROFIBUS-DP jest dedykowany dla bardzo szybkiej wymiany danych w aplikacjach na poziomie urządzeń, gdzie najczęściej sterownik lub inna jednostka centralna komunikuje się ze swoimi rozproszonymi wejściami i wyjściami. Większość komunikacji na tym poziomie jest realizowana za pomocą cyklicznej wymiany danych. Wszystkie mechanizmy wymagane przy tego typu komunikacji są wyspecyfikowane w podstawowej normie EN 50 170 dotyczącej PROFIBUS'a-DP.

W odróżnieniu od połączeń cyklicznych, komunikacja acykliczna jest wymagana w urządzeniach inteligentnych do obsługi alarmów lub diagnostyki. PROFIBUS-DP z nowymi funkcjami (*Extension*) również wspiera tego typu komunikację.

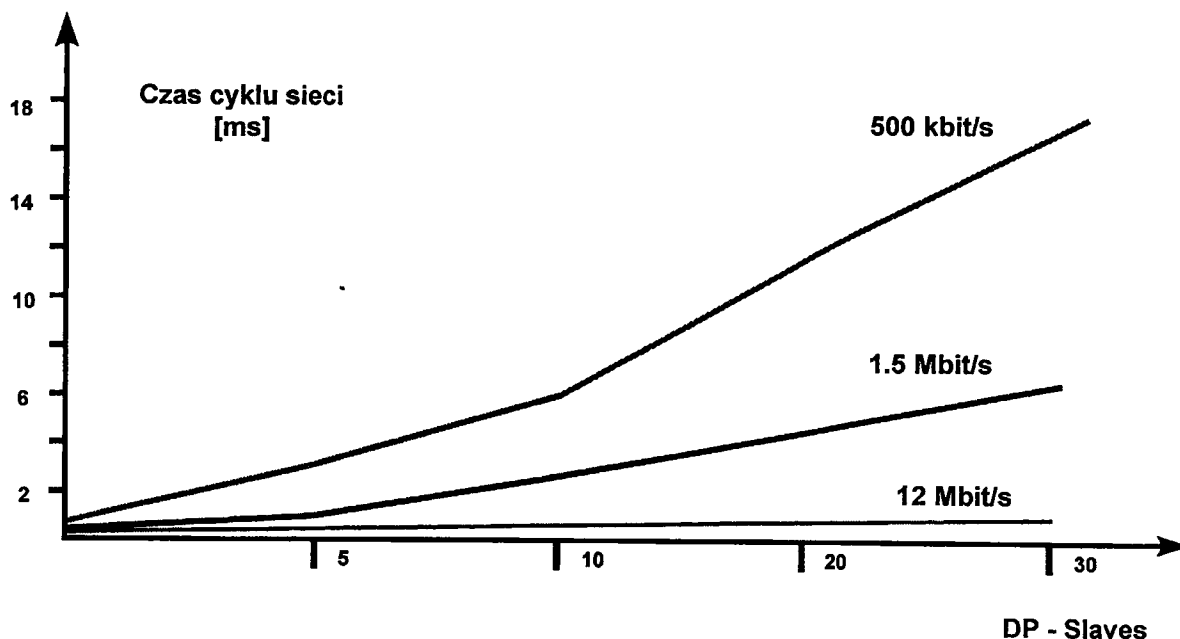
5.1.4.1 Podstawowe funkcje PROFIBUS'a-DP

5.1.4.1.1 Charakterystyka ogólna

Duża przepustowość sieci nie jest jedynym kryterium wyboru stosowania danej magistrali. Bardzo ważnymi kryteriami są również:

- łatwa instalacja oraz obsługa,
- możliwość diagnostyki oraz
- niezawodność.

Czas cyklu magistrali w sieci PROFIBUS-DP zależy głównie od stosowanej prędkości transmisji. Prędkość ta może być zadawana od 9.6 kbit/s do 12 Mbit/s zgodnie z szeregiem. Na rysunku 5.1.5 pokazano czasy cyklu magistrali PROFIBUS-DP w konfiguracji *mono-master*.



Rys.5.1.5 Czasy cyklu sieci PROFIBUS-DP *Mono-Master*

I tak np. dla systemu składającego się z 30 *DP-Slave*’ów, mających po 2 byte informacji wejściowej (480 wejść binarnych) i 2 byte informacji wyjściowej (480 wyjść binarnych), jest możliwe uzyskanie czasu cyklu sieci na poziomie 1 ms. Pozwala to na zastosowanie sieci PROFIBUS-DP w aplikacjach o dużych wymaganiach czasowych.

Funkcje diagnostyczne. Rozszerzone funkcje diagnostyczne PROFIBUS’a-DP pozwalają na szybką lokalizację awarii. Informacje diagnostyczne podzielone są na trzy poziomy:

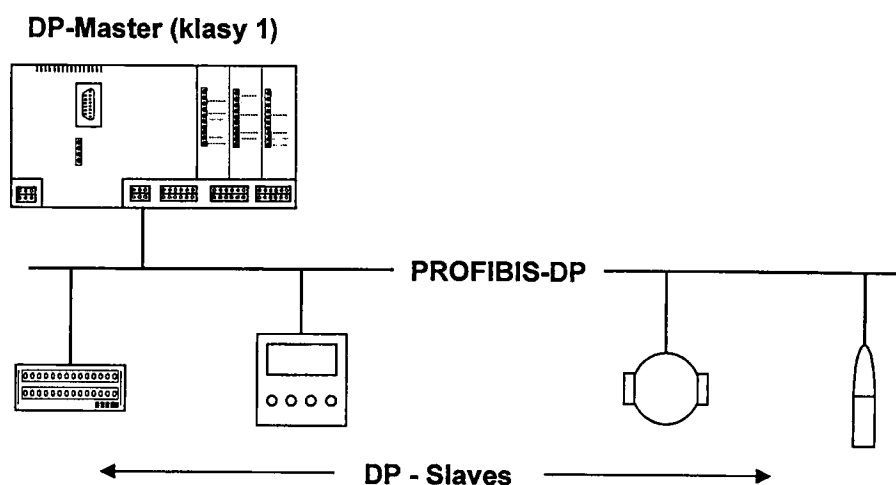
- diagnostyka na poziomie stacji,
- diagnostyka na poziomie modułu oraz
- diagnostyka na poziomie kanału.

5.1.4.1.2 Typy urządzeń oraz konfiguracja systemu

W sieci PROFIBUS - DP wyróżniamy trzy rodzaje urządzeń: *DP-Master Klasy 1*, *DP-Master Klasy 2* oraz *DP-Slave*:

- **DPM1** - to węzeł nadrzędny, pełniący rolę sterownika centralnego, który w ściśle określonym cyklu komunikacyjnym wymienia informacje ze swoimi zdecentralizowanymi modułami peryferyjnymi *Slave-DP*. Typowymi przedstawicielami DPM1 są sterowniki swobodnie programowalne PLC, komputery PC lub VME.
- **DPM2** - to programatory, konfiguratorzy, przyrządy diagnostyczne oraz urządzenia operatorskie. Są one wykorzystywane podczas uruchamiania instalacji oraz zadawania i kontroli pracy całego systemu.
- **DP-Slave** - to pasywne węzły będące modułami peryferyjnych wejść i wyjść cyfrowych, wejść i wyjść analogowych, napędów, zaworów, paneli operatorskich itp.

PROFIBUS-DP system mono-master. Podstawową architekturą pracy dla sieci PROFIBUS-DP jest konfiguracja *mono-master*. Oznacza to, że jedna stacja nadrzędna, np. sterownik swobodnie programowalny PLC albo komputer PC/VME, jest połączona ze zdecentralizowanymi modułami wejść i wyjść, czujnikami lub elementami wykonawczymi.

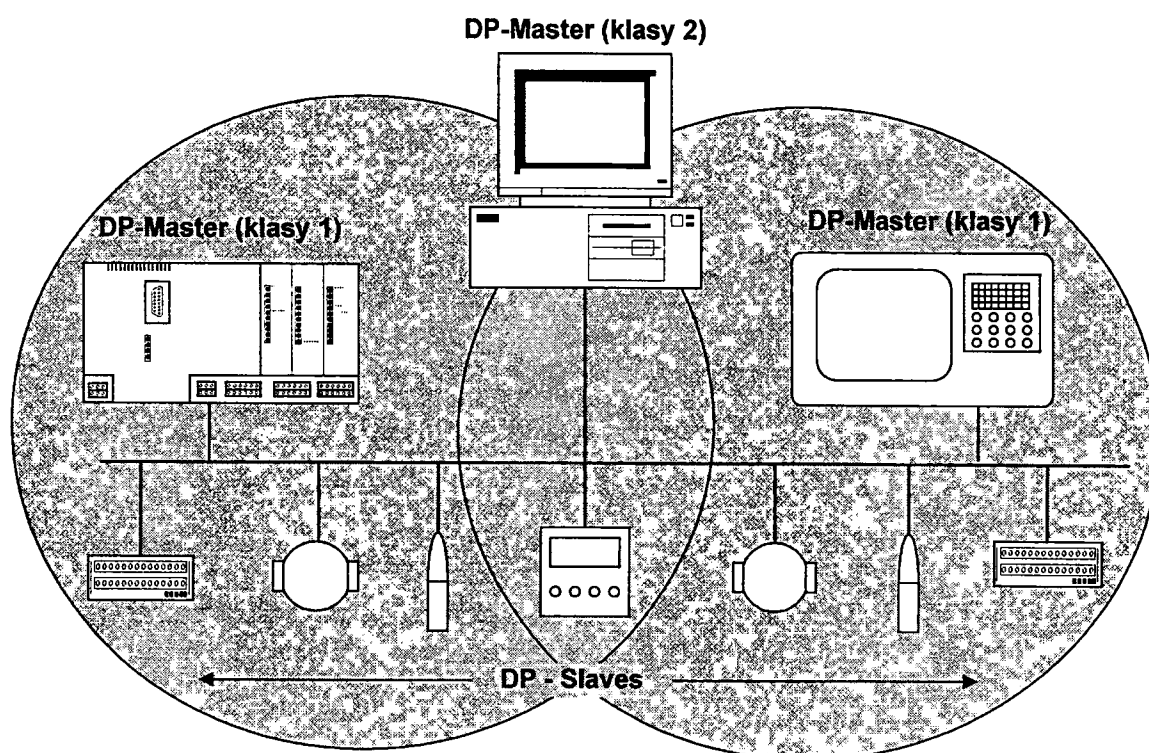


Rys.5.1.6 System *mono-master*

PROFIBUS-DP zastępuje wiązki równoległych przewodów między sterownikiem a urządzeniami peryferyjnymi na obiekcie, co obniża koszty komunikacji (wraz z inżynierią) nawet o 40%. System *mono-master* składa się z 1 do 125 urządzeń *Slave*, 1-go Master'a klasy 1 - jest nim np. sterownik PLC oraz opcjonalnie jednego Master'a Klasy 2 - urządzenia do konfiguracji i diagnozowania (rys.5.1.6).

Systemy *mono-master* zapewniają najkrótsze cykle systemu komunikacyjnego, gdyż nie zachodzi w nich konieczność przekazywania uprawnień między urządzeniami *master* (tzw. *token'a*) oraz wyszukiwania nowych stacji. Systemy takie są w stanie przesłać 1 kByte danych wejściowych i wyjściowych w czasie krótszym niż 2 ms.

PROFIBUS-DP system multimaster. PROFIBUS-DP może być stosowany również w systemach z wieloma stacjami nadrzędnymi - *multi-master*.



Rys.5.1.7 System *multi-master*

Na rysunku 5.1.7 zostały przedstawione dwie nadrzędne stacje *Master DP-klasy 1*: sterownik PLC i CNC wraz z podległymi stacjami pasywnymi *Slave*. Zaletą PROFIBUS-DP jest możliwość dzielenia dostępu do danych poprzez odpytywanie wspólnych stacji *Slave*. *Master DP-klasy 2* (komputer PC) pozwala natomiast na odczyt danych diagnostycznych oraz parametryzowanie wszystkich węzłów sieci.

5.1.4.1.3 Zachowanie systemu

Stan systemu określany jest przez stan urządzenia *master'a klasy 1*, które może przyjmować następujące wartości:

- **STOP** - gdy nie zachodzi żaden transfer danych,
- **CLEAR** - gdy DPM1 odczytuje stany wejść urządzeń *Slave* i utrzymuje stany ich wyjść w stanie "bezpiecznym" (*fail-safe*),
- **OPERATE** - gdy zachodzi transmisja danych - master cyklicznie komunikuje się z urządzeniami *Slave* (odczyt i/lub zapis).

5.1.4.2 Funkcje rozszerzonego PROFIBUS-DP

Organizacja Użytkowników PROFIBUS (PNO) opracowała rozszerzenie dla standardu PROFIBUS-DP, które zostało wydane jako zalecenia techniczne PNO 2.081. Celem tego rozszerzenia było zaspokojenie wymagań coraz bardziej kompleksowych urządzeń automatyki, dla których czysta cykliczna wymiana danych stała się niewystarczająca. Dotyczy to urządzeń wymagających parametryzacji podczas normalnej pracy, takich jak falowniki, sterowniki napędów, regulatory i urządzenia pomiarowe.

Do tej pory rozszerzenie funkcjonalności było dokonywane samodzielnie przez użytkowników, którzy definiowali własne słowa sterujące w polu danych. Takie rozwiązanie pogarsza jednak efektywność protokołu i prowadzi do niekompatybilności.

Zatwierdzone przez PNO rozszerzenie PROFIBUS-DP zapewnia dodatkowo:

- acykliczny odczyt i zapis dowolnych bloków danych *Slave'a* przez DPM1,
- acykliczny odczyt i zapis dowolnych bloków danych *Slave'a* przez DPM2,
- kompatybilność z istniejącymi implementacjami PROFIBUS-DP.

5.1.4.3 Pliki z danymi o urządzeniu (GSD)

PROFIBUS definiuje nie tylko komunikację między urządzeniami, ale specyfikuje on również otwartą i niezależną od producenta konfigurację węzłów sieci. Narzędzia konfiguracyjne (oprogramowanie) używają elektronicznych opisów urządzeń. Opisy te utworzone są w postaci plików tekstowych ASCII w bazie danych urządzenia i są oznaczane jako *.GSD.

Producent modułów PROFIBUS-DP jest zobligowany do opisu ich parametrów w formacie pliku GSD. Format ten jest szczegółowo zdefiniowany przez standard PROFIBUS. Konfigurator sieci odczytuje dane zawarte w plikach i przetwarza je na odpowiednią konfigurację stacji nadrzędnej. Zaletą tego jest bardzo łatwa konfiguracja sieci i niezależność od producenta, co obniża koszty inżynierskie. Rozwiązanie takie określane jest w świecie komputerów PC i systemu Windows jako „*Plug and Play*”.

5.1.4.4 Profile PROFIBUS-DP

Protokół PROFIBUS-DP definiuje sposób przesyłania danych pomiędzy stacjami. Zadaniem profilu jest definiowanie jak PROFIBUS-DP ma być używany od strony aplikacyjnej, np.:

- **profil dla NC/RC (3.052)**, który opisuje sposób sterowania przez PROFIBUS'a-DP manipulatorów oraz robotów montażowych,
- **profil dla przetworników (3.062)**, który opisuje połączenie PROFIBUS'a-DP z przetwornikiem obrotowym, przetwornikiem kątowym oraz liniowym,
- **profil dla napędów (3.071)**, który opisuje sposób sparametryzowania urządzenia oraz zawiera informacje dla sterowania prędkością oraz pozycjonowania,
- **profil dla paneli operatorskich HMI** (ang. *Human Machine Interface*), który opisuje dołączenie urządzeń typu HMI do PROFIBUS'a-DP. Profil ten uwzględnia również funkcje PROFIBUS'a-DP rozszerzonego.

5.1.5 PROFIBUS-PA

PROFIBUS-PA jest rozwiązaniem dedykowanym do automatyzacji procesów przemysłowych. Dzięki zastosowaniu tego rozwiązania techniki cyfrowej możliwe jest pełne zastąpienie aktualnie wykorzystywanej technologii 4..20mA, uzyskując ponadto rozszerzenie funkcjonalności oraz zwiększenie niezawodności. Sterowanie, regulacja i monitorowanie mogą być zrealizowane poprzez zwykły kabel dwużyłowy. Poprzez sieć PROFIBUS-PA możliwe jest zadawanie i odczytywanie nastaw urządzenia oraz zmiana jego konfiguracji. PROFIBUS-PA może być używany w systemach iskrobezpiecznych i umożliwia on doprowadzenie zasilania poprzez magistralę.

5.1.5.1 Protokół transmisyjny PROFIBUS-PA

PROFIBUS-PA używa podstawowych funkcji PROFIBUS'a-DP do transmisji wartości zmierzonych oraz funkcji rozszerzonego PROFIBUS'a-DP do parametryzacji i innych operacji na urządzeniach przemysłowych. Do transmisji danych używane są dwie technologie zgodne z normą IEC 1158-2.

5.1.5.2 Profile urządzeń PROFIBUS-PA

Każde urządzenie PROFIBUS-PA musi mieć zaimplementowane usługi zgodne ze swoim profilem. Zapewniają one pełną współpracę urządzeń pochodzących od różnych producentów. Standard ten bazuje na zaakceptowanej na całym świecie technologii bloków funkcyjnych.

5.1.6 PROFIBUS-FMS

PROFIBUS-FMS jest przeznaczony do realizacji połączeń komunikacyjnych na poziomie gniazd. W tym obszarze komunikacja przebiega pomiędzy programowalnymi sterownikami PLC, komputerami PC oraz inteligentnymi urządzeniami automatyki. Obszar ten charakteryzuje się wyższym stopniem funkcjonalności, przy mniejszych wymaganiach czasowych.

5.2 Sieć LonWorks

Przemysłowa sieć komunikacyjna LonWorks (ang. *Local Operating Network*) została opracowana przez firmę ECHELON (USA), w której 25% udziałów ma Motorola. W sieci LonWorks wykorzystuje się układ scalony tzw. *Neuron Chip*, który zawiera system komunikacji i sterowania oraz pamięci RAM i EEPROM, a także kompletny stos protokołów ISO/OSI (7). *Neuron* zbudowany jest z 3 procesorów, a jego pamięć stała jest zdalnie programowalna. Do takiego modułu komunikacyjno-sterującego jako stałe oprogramowanie wprowadzono protokół transmisji LonTalk. Sieć LonWorks jest systemem otwartym, produkcją *Neuron*'ów zajmują się koncerny Motorola i Toshiba, a produkty tego systemu są oferowane przez kilkaset firm w świecie. LonWorks wykorzystuje wiele rodzajów mediów komunikacyjnych takich jak: para przewodów skręconych, kabel koncentryczny, światłowód, radio oraz linie energetyczne. Maksymalna prędkość transmisji wynosi 1.25 Mbit/s na odcinku o długości 500 m i 78 kbit/s na odcinku 1200 m.

Technologia LonWorks jest platformą dla implementacji komunikacji sieciowej, zawierającą elementy do projektowania, implementacji i eksploatacji sieci tzn.:

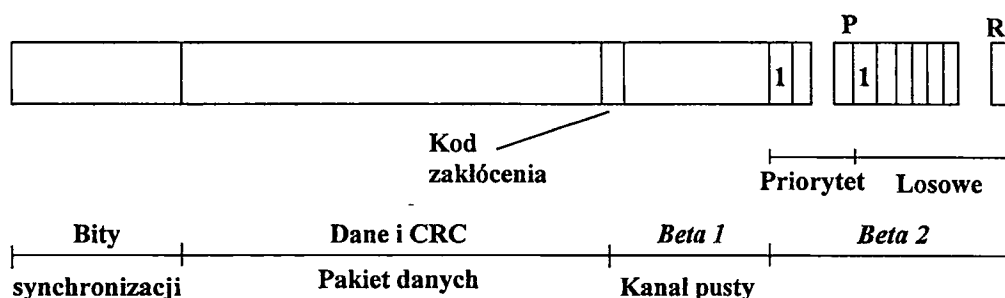
- *Neuron Chip* MC143150 lub MC143120,
- protokół LonTalk,
- nadajniki i odbiorniki oraz
- oprogramowanie narzędziowe LonBuilder, NodeBuilder i LonManager.

Protokół transmisji LonTalk jest zapisany w pamięci stałej układu *Neuron*. Podstawowe funkcje zarządzania siecią są wykonywane przez procesor sieciowy i procesor komunikacyjny w sposób niezauważalny dla użytkownika. Program użytkownika jest natomiast wykonywany przez trzeci z procesorów układu *Neuron*. Protokół ten zawiera oprogramowanie wszystkich siedmiu warstw modelu otwartego ISO/OSI, przy czym istnieje możliwość wyboru usług użytkownika w zależności od potrzeb wynikających z zastosowania sieci LonWorks. Dostęp do medium transmisyjnego jest realizowany metodą CSMA z opcjonalnym CD, przy czym poszczególne węzły instalacji sieciowej mogą być tak skonfigurowane, aby możliwa była transmisja informacji z priorytetem, tzn. przez rezerwację pasma tylko dla tych danych. Dzięki temu informacje o najwyższym priorytecie mają zapewniony określony czas odpowiedzi, a podczas transmisji priorytetowych pakietów nie występują kolizje. W warstwie aplikacyjnej protokół LonTalk jest ukierunkowany na przesyłanie danych. Dlatego też np. dane o wartości zmierzonej za pomocą czujnika są przesyłane między węzłami sieci w uprzednio zdefiniowanych ramach. Przetwarzanie takich pakietów danych odbywa się natomiast w różnych węzłach sieci za pomocą programów aplikacyjnych. W sieci LonWorks preferuje się przesyłanie zmiennych sieciowych po zmianie stanu, np. wejść/wyjść (I/O). Wskutek tego zmniejsza się przeciążenie sieci, gdyż unika się konieczności odpytywania poszczególnych węzłów o stany zmiennych. W protokole LonTalk udostępnione są następujące cztery podstawowe rodzaje transmisji:

- **z potwierdzeniem**, tzn. takie transmisje w których odbiorca pakietu przesyła potwierdzenie do nadawcy, a ten nadawca po odczekaniu zadanego czasu sprawdza otrzymanie wszystkich potwierdzeń i ewentualnie powtarza wysłanie pakietu,
- **żądanie-odpowieź**, na żądanie węzła sieci program aplikacyjny innego węzła, po przeanalizowaniu żądania, generuje odpowiedź, która może zawierać np. dane,
- **z automatycznym powtarzaniem - bez potwierdzeń**, która to transmisja jest stosowana w przypadku rozbudowanej instalacji sieciowej w celu uniknięcia blokowania sieci przez nadmierną ilość potwierdzeń,
- **bez potwierdzenia**, który jest stosowany w celu zwiększenia efektywności sieci w tych przypadkach, gdy nie jest wymagana 100% niezawodność transmisji.

W protokole LonTalk oferowane są również dwie usługi zapewniające skrócenie czasu odpowiedzi na transmisję pakietu, a mianowicie:

- **wykrywanie kolizji w dostępie do medium transmisyjnego**, co bez udziału programu aplikacyjnego pozwala na przerwanie dalszej transmisji po wykryciu zajętości kanału, a następnie po odczekaniu określonego czasu ponowne nadanie przesyłki oraz
- **zastosowanie priorytetów** w zakresie od 2 do 127, co pozwala na sterowanie czasem dostępu do medium dla poszczególnych węzłów sieci stosownie do wymagań wynikających z zastosowania sieci LonWorks.



Rys.5.2.1 Ramka transmisyjna sieci LonWorks

W sieci LonWorks przed rozpoczęciem transmisji przeprowadzana jest inicjalizacja danych. Polega ona na tym, że nadajnik wysyła bity synchronizacji (minimum 6 bitów o czasie trwania 125 μ s), które umożliwiają innym węzłom sieci zsynchronizowanie swoich zegarów (rys.5.2.1). Po wysłaniu ostatniego bitu CRC jeszcze przez czas odpowiadający dwóm okresom bitu wyjście stacji jest utrzymywane w stanie gotowości transmisji (stan wysoki sygnału). Przejście wyjścia w stan niski oznacza natomiast zakończenie transmisji. Czas potrzebny na sformatowanie (przygotowanie) i rozpoczęcie transmisji dla układu *Neuron Chip* zawiera się w przedziale 2.8 - 44.8 ms i zależy on od częstotliwości zegara (10 MHz - 625 MHz). Jeżeli wejście stacji do wykrywania wystąpienia kolizji jest aktywne i przyjmie poziom "0" na czas równy przynajmniej jednemu okresowi zegara (np. dla częstotliwości zegara 10 MHz okres ten wynosi 200 ns), to układ *Neuron Chip* sygnalizuje kolizję, co oznacza konieczność ponownego przesłania informacji. Pakiet o nazwie *Beta 1* jest

przebiegiem pustym i trwa zwykle 370 μ s dla szybkości transmisji 1.25 Mbit/s oraz 12-to bajtowych pakietów danych. Pakiet *Beta 2* jest natomiast wykorzystywany do ustalania priorytetu przesyłanych informacji. Jeżeli równocześnie dwa węzły próbują rozpocząć transmisję, co oznacza wystąpienie kolizji, to wówczas następuje przerwanie transmisji i w sposób losowy są ustalane czasy trwania odcinków *Beta 2*.

Każde urządzenie w sieci LonWorks, tzn. węzeł sieci, podobnie jak i w innych sieciach przemysłowych ma swój adres, który jest ustalany podczas konfigurowania sieci. Natomiast każda przesyłka zawiera adres źródła i adres stacji odbiorczej. Dla lepszego wykorzystania medium transmisyjnego adresowanie w sieci LonWorks jest hierarchiczne i składa się z numeru tzw. domeny, numeru podsieci (*subnet*) oraz numeru węzła. Podsieć (gałąź) stanowi logiczne połączenie od 2 do 127 węzłów. Ponieważ w domenie możliwe jest zdefiniowanie do 255 podsieci, dlatego też całą sieć LonWorks może zawierać do 32 285 urządzeń.

Typową topologią sieci LonWorks jest szyna (*bus*), w której poszczególne urządzenia są łączone krótkimi odcinkami kabla do jednej długiej magistrali zakończonej z obu stron terminatorami. Dzięki temu zmniejsza się długość kabla wymaganego do wykonania instalacji sieci.

Oprogramowanie narzędziowe (*LonBuilder*, *NodeBuilder* i *LonManager*) umożliwia przeprowadzanie: konfigurowania sieci, ładowanie, uruchamianie i zatrzymywania programów aplikacyjnych w wybranych węzłach sieci, badanie parametrów komunikacji itp.

Sieć komunikacyjna LonWorks spełnia wymagania otwartych rozwiązań systemów automatyki, jest stosunkowo prosta technologicznie, umożliwia wykorzystanie różnych topologii i mediów transmisyjnych. Ponadto na jej korzyść przemawiają niskie koszty urządzeń oraz dostępność i kompletność narzędzi do instalowania, konfigurowania i zarządzania. Dlatego też sieć LonWorks ma liczne zastosowanie w aplikacjach przemysłowych, automatyzacji budynków oraz transporcie.

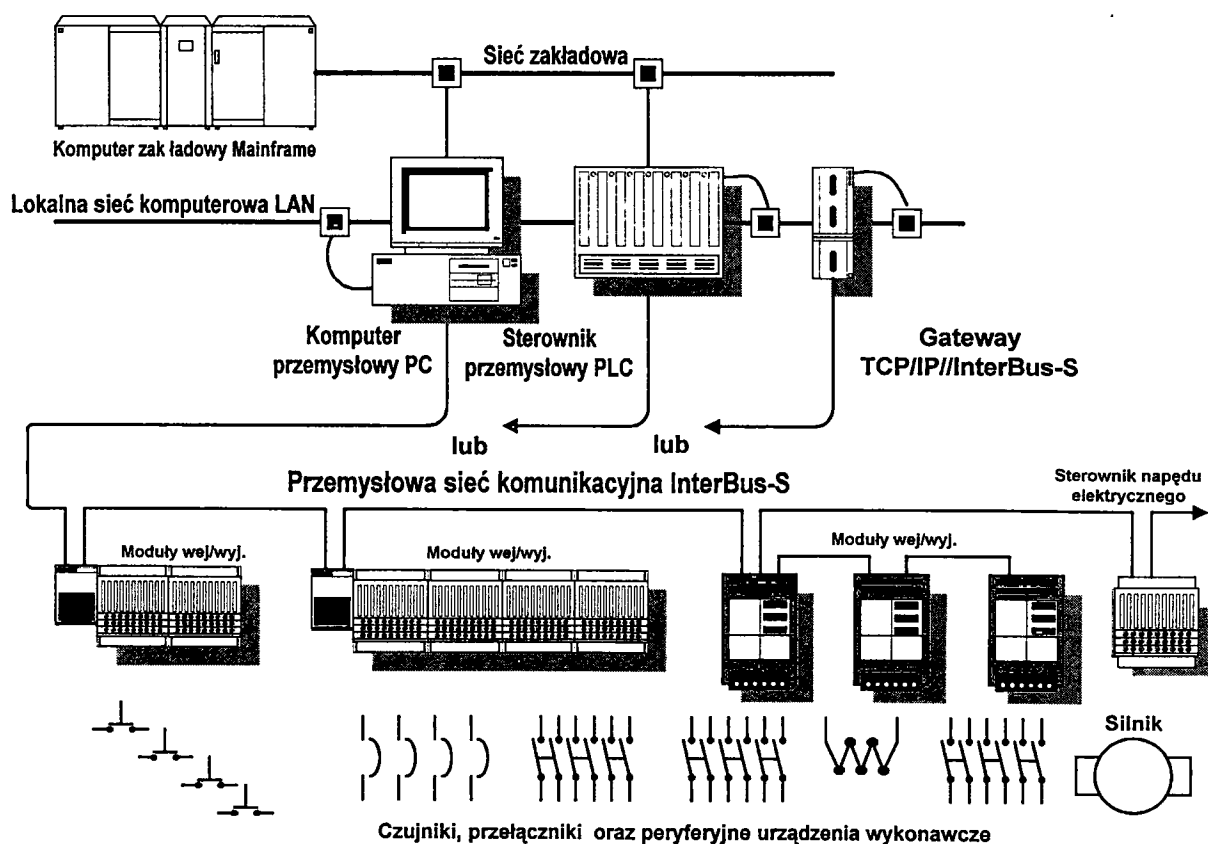
Literatura

- [1] Goszczyński T., *Sieć LonWorks*, Elektroinstalator 7-8/97 i 11/97
- [2] Werwka J., Drwal A., Żaba S., *Protokoły dostępu do medium transmisyjnego dla wybranych sieci przemysłowych*, Konf. AUTOMATION'97. Automatyzacja - Nowości i Perspektywy, PIAP, Warszawa 1997

5.3 System InterBus-S

5.3.1 Wprowadzenie

Z punktu widzenia automatyzacji procesów przemysłowych, tradycyjne techniki łączenia takich urządzeń automatyki jak sterowniki przemysłowe z czujnikami i elementami wykonawczymi, polegające na połączeniach typu *punkt-punkt* przy pomocy okablowania równoległego, są rozwiązaniami nieelastycznymi, trudnymi w utrzymaniu oraz bardzo kosztownymi. Chociaż idea wykorzystania szeregowych łączy komunikacyjnych do sprzęgania ze sobą poszczególnych urządzeń automatyki przemysłowej na najniższym poziomie automatyzacji procesów przemysłowych jest znana od wielu lat, to dopiero w latach 90'tych nabrała ona realnego kształtu w postaci wielu znormalizowanych rozwiązań. Głównym kryterium wyboru systemu komunikacyjnego dla obszaru sterowników oraz czujników-elementów wykonawczych jest jego otwartość, pozwalająca na współdziałanie wielu różnorodnych urządzeń pochodzących od wielu producentów w jednej sieci komunikacyjnej. Podstawowa korzyść płynąca z takiego rozwiązania to optymalne dopasowanie systemu sterowania automatycznego do obiektu sterowanego oraz specyficznych wymagań algorytmu sterowania.



Rys.5.3.1 Zastosowanie systemu komunikacyjnego w automatyzacji procesów przemysłowych

Centralną rolę w systemie InterBus-S (IBS) pełni, umieszczona w sterowniku PLC lub komputerze przemysłowym IPC, specjalizowana karta - *Host Controller Board* (HCB) (rys.5.3.1).

Niezależnie od zastosowanego mikroprocesorowego systemu sterowania, karta HCB jest odpowiedzialna za sterowanie przepływem danych procesowych i parametrycznych w systemie IBS. Istotną cechą systemu sterowania, złożonego np. ze sterownika PLC oraz współpracującego z nim systemu magistralowego IBS, jest krótki czas transferu danych obiektowych oraz łatwość konfiguracji i programowania.

Przemysłowy system komunikacyjny InterBus-S jest zbudowany z trzech podstawowych elementów: sterownika systemu HCB, urządzeń peryferyjnych wejść/wyjść (I/O - *Input/Output*) oraz okablowania łączącego poszczególne urządzenia peryferyjne ze sterownikiem HCB. Różnorodne wymagania procesów przemysłowych co do funkcji oraz środowiska pracy urządzeń automatyki, spowodowały powstanie szeregu specjalizowanych rozwiązań urządzeń IBS. Aktualnie na rynku są dostępne urządzenia automatyki IBS, które można podzielić na trzy grupy: urządzenia grupy topologicznej (karty sterowników HCB, moduły terminalowe BT), urządzenia grupy funkcyjnej (moduły I/O, liczniki czasu, zegary, kodery, falowniki) oraz urządzenia grupy środowiskowej w obudowach o stopniu ochrony IP 20, IP 65, IP 67.

Tab.5.3.1 Wybrane sterowniki centralne HCB systemu InterBus-S

System sterowania	Typ karty InterBus-S
Komputer PC	systemy komputerowe z magistralą ISA lub wykonane w technologii PC104
Komputer VMEbus	wszystkie systemy VMEbus 3U oraz 6U
Brama Ethernet	przyłącze do sieci Ethernet z protokołem TCP/IP
Allen-Bradley	systemy sterowania dla sterowników serii PLC 5/11, 5/20, 5/30/ 5/40, 5/60, 5/80
GE Fanuc	systemy sterowania serii 90-70
Siemens	systemy sterowania serii S5 95/100 U, S5 115..155U, S7 300, S7 400, Sinumerik 840C i 880
AEG-Schneider	TSX Quantum, TSX A250, A350, A500 . TSX A250 (<i>central rack</i>)
Bosch	CL 300, CL 400, CL 500, PC 400, PC 600
Klöckner-Moeller IPC	KM IPC 620
Honeywell	UDC 9000

Spektrum urządzeń peryferyjnych kompatybilnych z systemem InterBus-S obejmuje: cyfrowe i analogowe moduły I/O, specjalizowane moduły takie jak liczniki, moduły V.24 (RS232C), wyświetlacze, panele operatorskie oraz zawory, jak również urządzenia produkowane przez producentów stowarzyszonych w organizacjach użytkowników takich jak DRIVECOM i ENCOM. Poniżej przedstawiono podstawowe urządzenia peryferyjne automatyki dostępne dla systemu InterBus-S (tab.5.3.1):

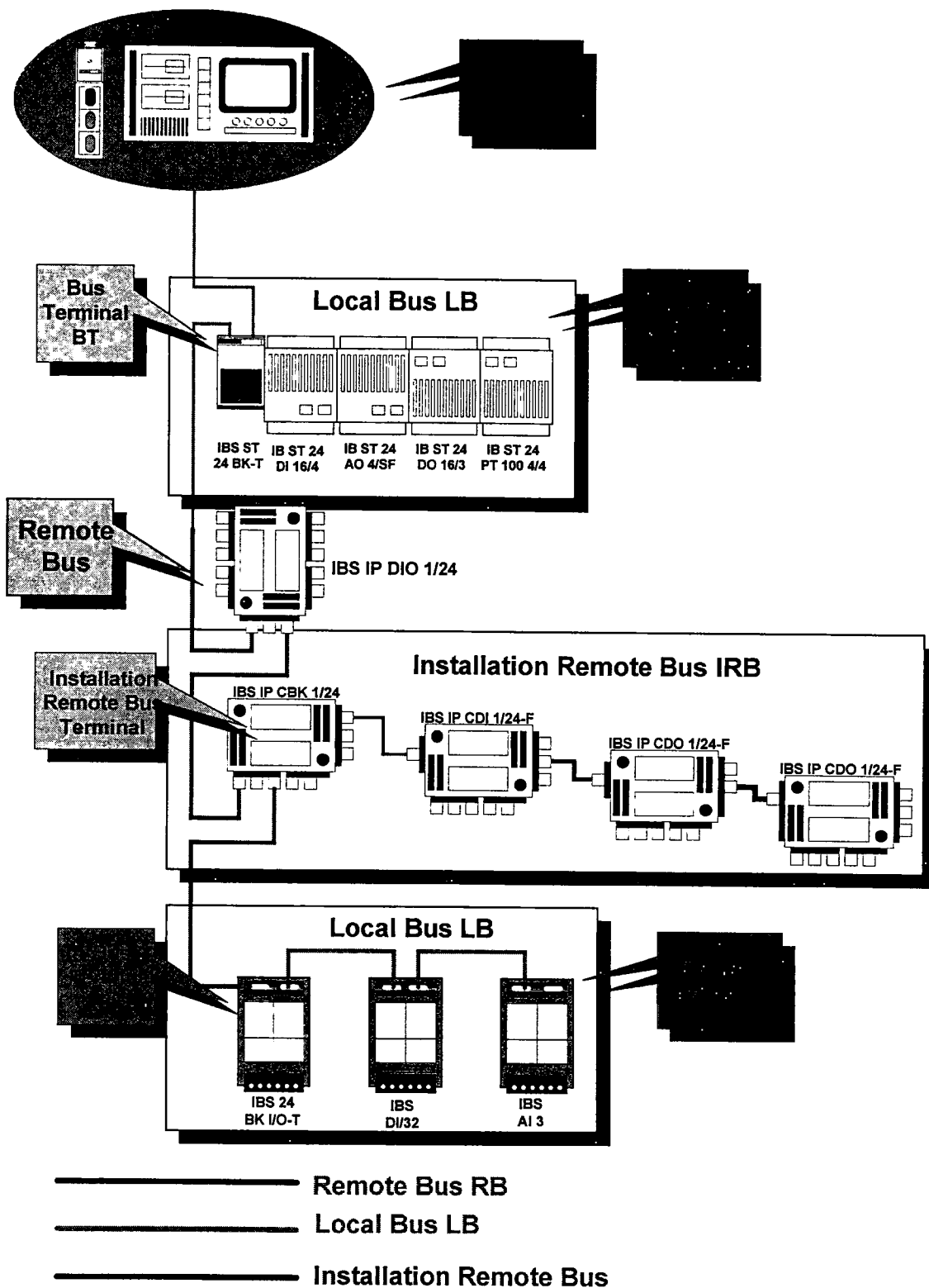
- karty sterowników centralnych HCB (*Host Controller Board*) dla systemów sterowania wysokiego poziomu (PLC, CNC, IPC, VME),
- przyłączeniowe moduły terminalowe BT (*Bus Terminal*),
- analogowe moduły I/O,
- cyfrowe moduły I/O,
- moduły komunikacyjne dla sterowników i komputerów (Siemens, Bosch, IBM PC),
- interfejsy operatora oraz wyświetlacze,
- moduły liczników,
- moduły V.24,
- moduły GATEWAY,
- falowniki, serwonapędy,
- systemy identyfikacji,
- kodery, enkodery,
- inne moduły takie jak: sterowniki-pozycjonery, sterowniki robotów, zawory pneumatyczne, zawory hydrauliczne itp.

5.3.2 Topologia systemu InterBus-S

Topologia systemu komunikacyjnego jest ściśle związana z wykorzystywanym w danym systemie interfejsem elektrycznym oraz protokołem komunikacyjnym warstwy liniowej (warstwa 2 wg modelu referencyjnego ISO/OSI). Stosowane obecnie sieciowe systemy komunikacyjne typu *Fieldbus* wykorzystują kilka rodzajów struktur topologicznych. W technice integracji elementów automatyki przemysłowej na poziomie czujników i urządzeń wykonawczych oraz sterowników używane są struktury typu magistralowego (ang. *bus*) oraz pierścienia (ang. *ring*). InterBus-S jest systemem o topologii pierścienia tzn., że wszystkie urządzenia automatyki są aktywnie włączone w zamkniętą pętlę komunikacyjną. Wyróżniającą cechą systemu InterBus-S, w porównaniu z innymi systemami o topologii pierścienia jest to, że zarówno dane wysyłane z wejściowych elementów peryferyjnych, jak i dane przychodzące do wyjściowych elementów są przesyłane w jednym kablu (3 parowa skrętka), który biegnie przez każde pojedyncze urządzenie systemu. Nadaje to systemowi fizyczny wygląd linii (ang. *bus*) lub drzewa (ang. *tree*).

Istotne parametry systemu InterBus-S takie jak: maksymalna liczba urządzeń dołączonych do sieci czy też maksymalna długość systemu, nie zależą od zastosowanego w nim protokołu komunikacyjnego a jedynie od cech centralnego sterownika systemu HCB. W przemysłowych systemach komunikacyjnych liczba urządzeń w sieci jest ograniczona przez pole adresu ramki. I tak np. przy 8-bit'owym polu możemy zaadresować tylko 256 różnych urządzeń. Protokół IBS ze swoją metodą transmisji danych, określaną jako metoda wspólnej ramki, nie wymaga adresów swoich urządzeń i dlatego też nie ogranicza ich liczby w sieci. Podstawowym ograniczeniem jest wydolność jednostki centralnej systemu IBS. Maksymalny dopuszczalny zasięg systemu jest

bezpośrednio zależny od szybkości transmisji danych i zmniejsza się wraz z jej wzrostem. Poprzez zastosowanie regeneratorów sygnałów (*repeater*), które odświeżają sygnały, można dołączać następne segmenty sieci (do 256). Maksymalna liczba *repeater*'ów jest także zależna od szybkości przesyłania i również maleje wraz z jej wzrostem. W systemie *InterBus-S* elementy magistrali odległej pracują tak jak *repeater*'y i dlatego też pomiędzy nimi może być zbudowane połączenie o długości do 400 m przy zastosowaniu kabla miedzianego (zakładając prędkość transmisji 500 kbit/s dla interfejsu RS485).



Rys.5.3.2 Budowa systemu InterBus-S

Na system InterBus-S składają się następujące elementy funkcjonalne (rys.5.3.2):

- centralny sterownik systemu HCB (ang. *Host Controller Board*) - zintegrowany z komputerem przemysłowym PC (np. IBS PC CB/COP/I-T) lub sterownikiem PLC (np. IBS S5 DSC/IT),

- magistrala odległa RB (ang. *Remote Bus*) wraz z modułami terminalowymi BT (ang. *Bus Terminal* - np. IBS ST 24 BK-T), FBT (ang. *Flat Bus Terminal* - np. IBS 24 BK-T), RT (ang. *Remote Terminal* - np. IBS RT 24 DIO 16/16-T) oraz IRBT (ang. *Installation Remote Bus Terminal* - np. IBS IP CBK),
- instalacyjna magistrala odległa wraz z modułami I/O typu CBK oraz
- magistrala lokalna LB (ang. *Local Bus*) z modułami funkcyjnymi wejść i wyjść obiektowych (ang. *I/O function module*).

Sieć InterBus-S jest systemem komunikacyjnym typu *mono-master*, tzn. że tylko jedno urządzenie w sieci jest zdolne do odpytywania i przesyłania danych do/z modułów funkcyjnych systemu. Rolę urządzenia nadrzędnego typu *master* w systemie InterBus-S pełni centralny sterownik HCB, natomiast moduły funkcyjne systemu pełnią rolę jednostek podrzędnych typu *slave*.

Na strukturę topologiczną sieci IBS mają przede wszystkim wpływ zastosowane urządzenia peryferyjne, a w szczególności moduły terminalowe typu BT (*Bus Terminal*). Natomiast typowe moduły terminalowe, np. IBS ST 24 BK-T, pełnią w systemie rolę wzmacniaczy sygnałów elektrycznych (ang. *Repeater*) standardu RS485. Jeżeli moduł BT jest oznaczony symbolem „T” to oznacza, że dany moduł pełni w systemie zadanie lokalnego rozgałęzienia typu „T”. Każdy taki moduł ma co najmniej trzy gniazda:

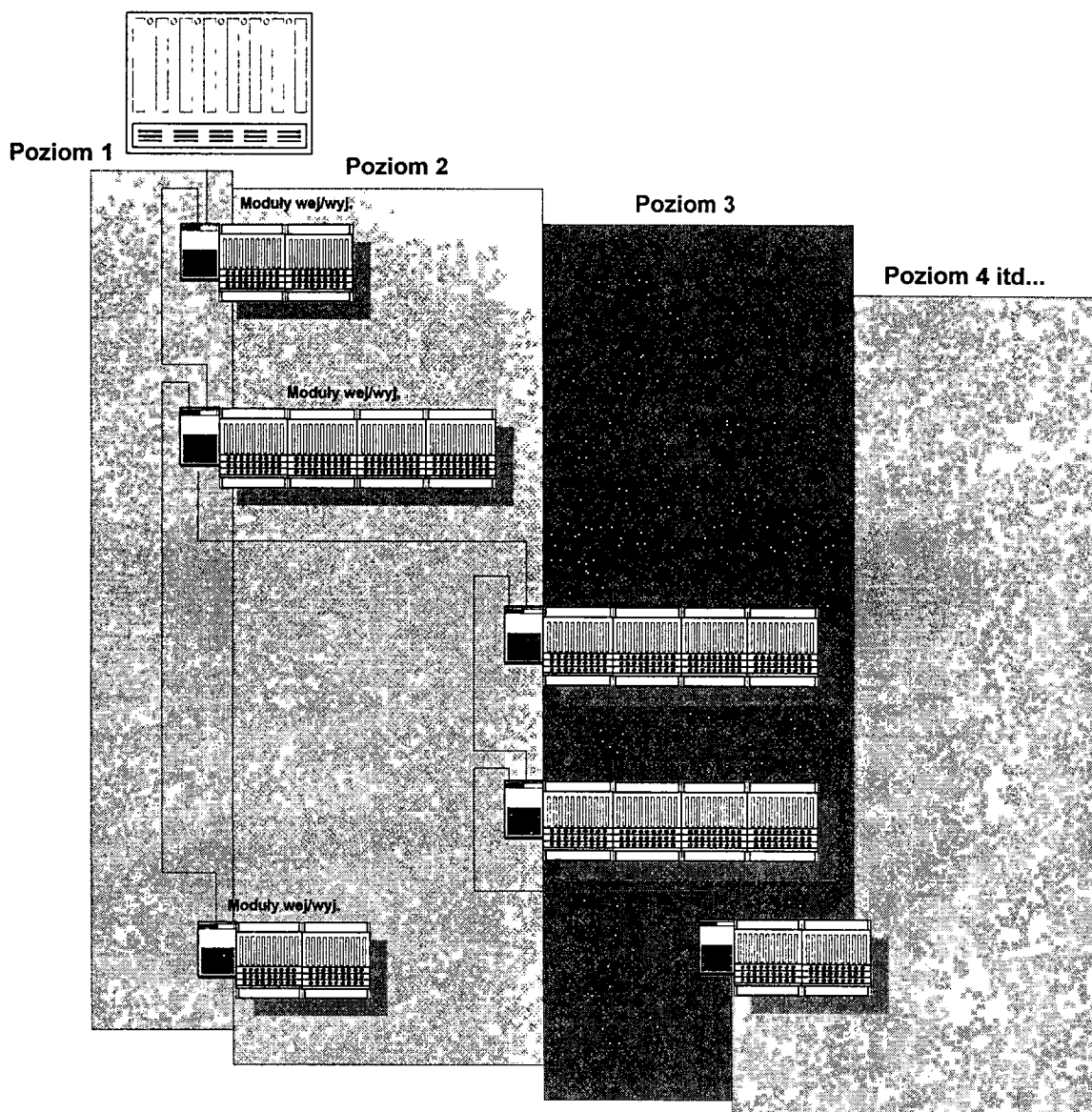
- *Remote IN* - wejścia magistrali odległej (RS485),
- *Remote OUT* - wyjście magistrali odległej (RS485) oraz w zależności od typu modułu:
- *Local Bus*, *Remote Bus*, *Installation Remote Bus* lub *InterBus-Loop*.

Moduł terminalowy typu „T” z trzema interfejsami typu *Remote Bus*, np. IBS ST 24 BK RB-T, jest podstawą do budowy sieci IBS o topologii drzewa.

Począwszy od wersji 4.X oprogramowania systemowego InterBus-S jest możliwe rozgałęzianie magistrali odległej RB systemu do postaci drzewa. Podstawową zaletą takiej architektury systemu jest możliwość elastycznego dopasowywania systemu IBS do potrzeb topologicznych danego urządzenia czy też rozproszonego systemu automatyki.

Główna linia magistrali odległej ma swój początek w centralnym sterowniku systemu HCB i jest nazywana magistralą poziomu 0. Kolejne rozgałęzienia magistrali odległej mają swój własny poziom zagnieżdżenia (poziom 1, 2 itd.). Poprzez budowanie kolejnych zagnieżdżeń jest możliwe rozbudowanie systemu aż do 16-tego poziomu (rys.5.3.3).

Magistrala lokalna w segmencie magistrali (n) jest zawsze dołączona do poziomu następnego (n+1). Oznacza to, że magistrala lokalna może być dodana maksymalnie do poziomu 15. Urządzeniem, które otwiera kolejne rozgałęzienia jest specjalizowany moduł terminalowy BT z trzema interfejsami typu RB (*Remote bus*). Ponadto moduł taki może zawierać dodatkowy interfejs magistrali lokalnej LB.



Rys.5.3.3 Poziomy zagłębienia systemu IBS generacji 4

Urządzenia instalacyjnej magistrali odległej mogą być rozpatrywane jak urządzenia magistrali lokalnej, tzn. urządzenia występujące za modułem terminalowym należą do kolejnego poziomu zagłębienia. To samo dotyczy urządzeń pętli IBS, które są rozpatrywane jak urządzenia magistrali lokalnej LB, tzn. należą do następnego poziomu.

Oznaczenie poszczególnych urządzeń w magistrali IBS jest podzielone na dwie części: część pierwsza wskazuje na segment magistrali (BS), w którym urządzenie jest zainstalowane; natomiast część druga oznaczenia wskazuje na pozycję urządzenia w segmencie magistrali. W systemie IBS zastosowano tą samą notację zarówno dla urządzeń magistrali odległej jak i lokalnej:

OZNACZENIE: *BUS_SEGMENT.POZYCJA*

Wszystkie urządzenia magistrali odległej mają zawsze oznaczenie 0 w drugiej części swojego identyfikatora, np.: 4.0. Numerowanie segmentów magistrali jest dowolne i może być wybierane

przez użytkownika w zakresie 1-254. Oznacza to, że kiedy kolejne urządzenia magistrali odległej są wstawiane do systemu, to identyfikacja nie musi być przeprowadzana ponownie.

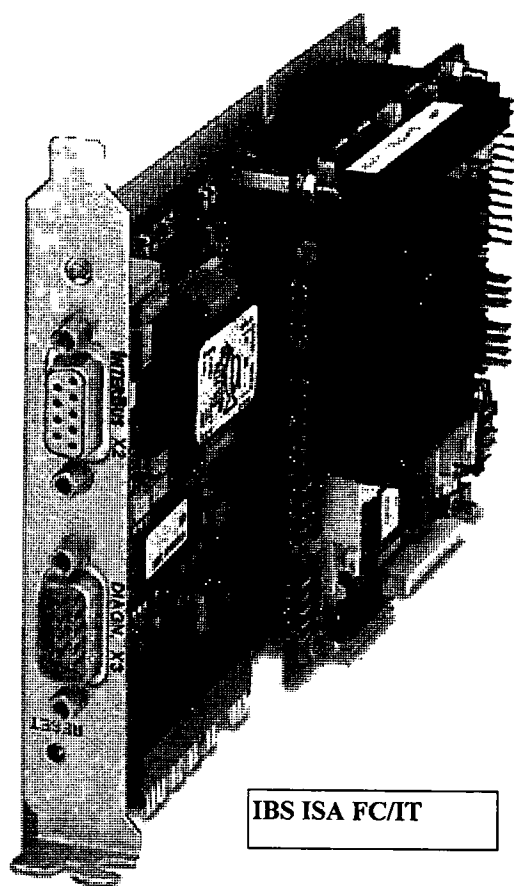
W przypadku urządzeń magistrali lokalnej, oznaczenie segmentu magistrali zajmuje pozycję 1, natomiast pozycja urządzenia w segmencie magistrali pozycję 2, np.: 6.2. Numer pozycji w magistrali lokalnej jest także dowolny (1-254).

5.3.3 Centralny sterownik systemu IBS

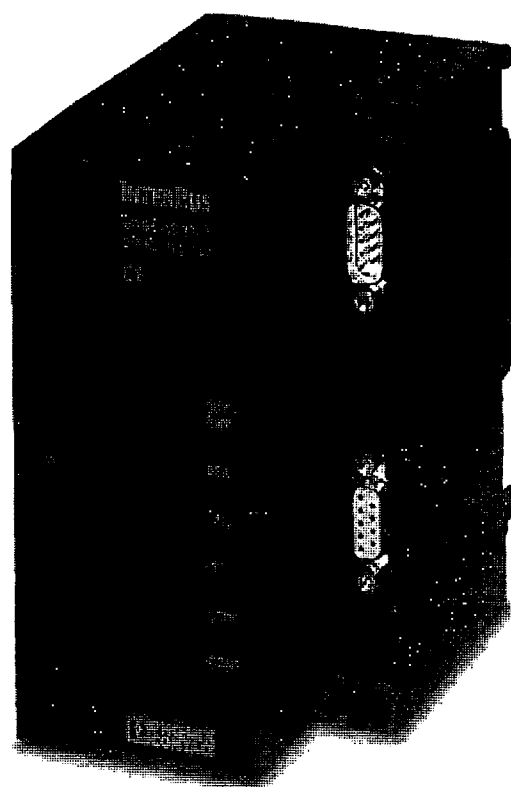
Podstawowym zadaniem centralnego sterownika systemu HCB jest połączenie szeregowej magistrali InterBus-S z takimi sterownikami przemysłowymi jak PLC i CNC oraz komputerem przemysłowym IPC lub VME. Wymiana danych między sterownikiem systemu InterBus-S a np. sterownikiem PLC odbywa się poprzez wewnętrzną, równoległą magistralę systemową taką jak VMEbus w sterownikach PLC-5/VME firmy Allen Bradley czy też SIMATIC® I/O w sterownikach SIMATIC® S5 firmy Siemens. Karta sterownika systemu HCB jest „widziana” w systemie komputerowym lub sterowniku przemysłowym jako karta peryferyjna wejść/wyjść (ang. *I/O board*). Funkcje centralnego sterownika systemu InterBus-S od strony magistrali szeregowej można podzielić na następujące części:

- start cyklu identyfikacyjnego oraz cyklu przesyłania danych systemu IBS,
- nadanie adresów elementom systemu,
- sterowanie cyklicznym protokołem IBS,
- transmisja danych procesowych do modułów obiektowych wejścia/wyjścia oraz do systemu sterowania,
- monitorowanie systemu InterBus-S,
- włączanie i wyłączanie pojedynczych segmentów magistrali,
- komunikacja z inteligentnymi elementami dołączonymi do magistrali poprzez kanał danych parametrycznych,
- detekcja błędów (określanie typu błędu oraz jego lokalizacja) oraz
- optyczne, poprzez wyświetlacz LCD oraz diody LED, wskazywanie takich informacji diagnostycznych o systemie jak: tryby pracy, wystąpienie błędu, status danych procesowych itp.

Realizacja tych zadań wymaga odpowiedniej mocy obliczeniowej sterownika systemu IBS (*master'a*) i dlatego też dzisiejsze jednostki nadrzędne HCB wyposażone są w dodatkowe koprocesory wspomagające pracę jednostki centralnej CPU systemu sterowania (IPC, PLC). Przejmują one całość funkcji i obliczeń związanych z przesyłaniem danych i działaniami na magistrali. Wiele sterowników centralnych IBS opiera się na module IBS MA firmy Phoenix Contact, która jest twórcą systemu *InterBus-S* (rys.5.3.4).



IBS ISA FC/IT



IBS S7 300 BC-T

Rys.5.3.4 Przykładowe sterowniki centralne systemu InterBus-S dla komputerów klasy PC (po lewej) oraz sterownika PLC Simatic S7-300 firmy Siemens

5.3.4 Magistrala odległa (*Remote Bus*)

Połączenie sterownika centralnego magistrali InterBus-S z oddalonymi podstacjami sterowania jest realizowane przy pomocy magistrali odległej RB wyposażonej w moduły terminalowe BT (*Bus Terminal*). Maksymalna długość kabla między centralnym sterownikiem HCB a pierwszym modułem terminalowym BT, a także między dwoma modułami terminalowymi BT wynosi 400 m. Natomiast maksymalna długość kabla między sterownikiem HCB a ostatnim modułem magistrali odległej nie powinna przekraczać 12.8 km, przy wykorzystaniu kabla miedzianego o ściśle zdefiniowanych parametrach elektrycznych. Ponieważ w systemie InterBus-S można zainstalować 256 modułów terminalowych BT, to teoretycznie można uzyskać połączenie przez kabel miedziany na odległość do 102 km ($= 256 \cdot 400 \text{ m}$). W praktyce testy wykazały, że dla niezawodnego działania systemu InterBus-S nie należy przekraczać długości 12.8 km. Uzyskanie większej odległości jest możliwe poprzez zastosowanie innego medium transmisyjnego, np. przy użyciu światłowodu jest możliwa transmisja danych na odległość 80 km. Tabela 5.3.2 zawiera podstawowe parametry magistrali odległej RB.

Tab.5.3.2 Dane techniczne magistrali odległej oraz instalacyjnej magistrali odległej

Max ilość elementów magistrali odległej	256
Interfejsy magistrali odległej	9-pin D-SUB,
Interfejsy instalacyjnej magistrali odległej	<i>IP 65 round connector (CCO)</i>
Max odległość pomiędzy dwoma elementami magistrali odległej	400 m (dla kabla miedzianego) 3.7 km (dla światłowodu)
Max całkowita długość systemu	12.8 km (kabel miedziany) ponad 80 km (światłowód)
Separacja galwaniczna	we wszystkich modułach
Napięcie zasilania	zewnętrzne 24 V DC lub poprzez kabel magistrali lub kabel hybrydowy (instalacyjna magistrala odległa)

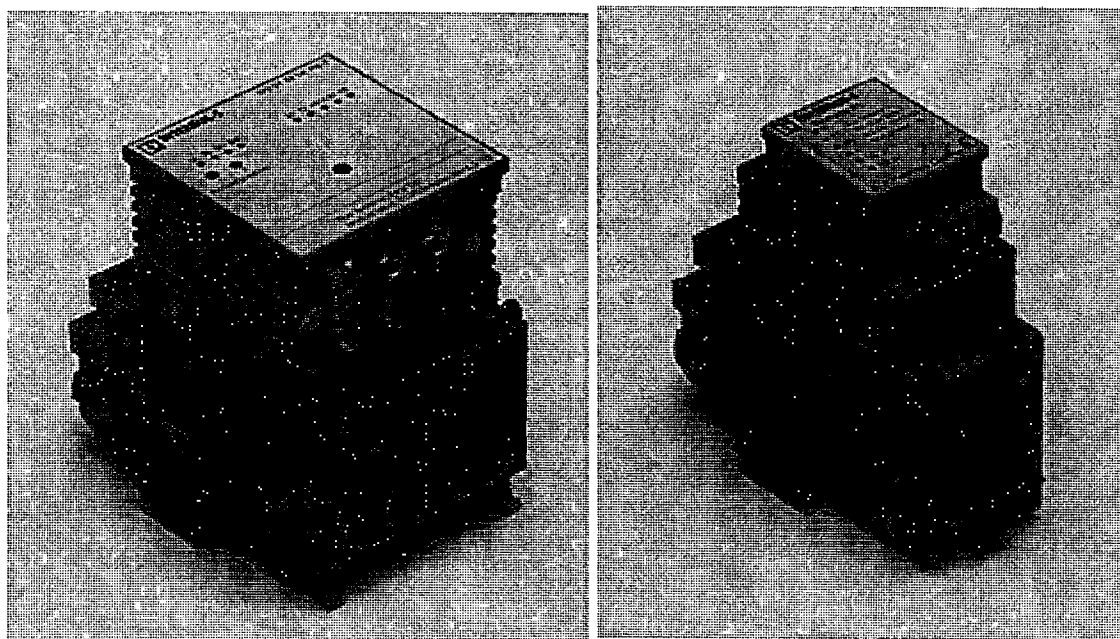
5.3.5 Segment magistrali odległej

Moduły terminalowe BT pełnią w systemie rolę wzmacniacza sygnału elektrycznego oraz stanowią interfejs do lokalnego rozgałęzienia sieci nazywanego magistralą lokalną LB. Odcinek magistrali pomiędzy dwoma modułami terminalowymi BT wraz z przyległymi lokalnymi rozgałęzieniami LB określany jest jako segment magistrali InterBus-S. W zależności od zastosowanego sterownika systemu HCB w sieci InterBus-S można zainstalować maksymalnie 256 segmentów magistrali. Transmisja danych w segmentach magistrali odległej odbywa się poprzez powszechnie używany interfejs szeregowy RS 485. Interfejs ten jest także wykorzystywany w innych systemach komunikacyjnych typu *Fieldbus* takich jak BITBus, DIN Meßbus, PROFIBUS, pod nazwą znanych specyfikacji np. EIA 485 (ang. *Electronics Industries Association*), ISO 8482 lub DIN 66259 part 4. Rysunek 5.3.5 przedstawia przykładowe moduły terminalowe magistrali odległej RB z interfejsem do magistrali lokalnej LB. Z lewej strony rysunku przedstawiono moduł BT z interfejsem do magistrali odległej (D-SUB 9-pin) oraz dwoma interfejsami do magistrali lokalnej (D-SUB 15-pin oraz COMBICON). Z prawej strony rysunku przedstawiono natomiast moduł terminalowy BT z interfejsem do magistrali odległej (D-SUB 9-pin) oraz interfejsem do magistrali lokalnej (COMBICON).

5.3.6 Instalacyjna magistrala odległa (*Installation Remote Bus*)

Instalacyjna magistrala odległa jest w swej strukturze równorzędna magistrali odległej RB, ale oferuje ona, obok transmisji danych procesowych z/do modułów funkcyjnych, dodatkowe możliwości dostarczania w jednym kablu napięcia zasilającego dla rozproszonych czujników. Pierwszym modułem instalacyjnej magistrali odległej jest moduł nazywany terminalowym, np. IBS IP CBK. Magistrala ta może składać się z szeregu modułów funkcjonalnych typu wejście/wyjście i jest ona przeznaczona do pracy w rozproszonych podstacjach sterowania z bezpośrednim połączeniem czujników oraz elementów wykonawczych. Długość kabla między modułem terminalowym CBK a pierwszym modułem wejścia/wyjścia, a także między modułem CBK a

ostatnim modulem wejścia/wyjścia, nie może przekraczać 50 m. Liczba modułów wejścia/wyjścia w instalacyjnej magistrali odległej jest ograniczona przez pobór prądu poszczególnych modułów oraz podłączonych do nich elementów wykonawczych i czujników. Maksymalny prąd obciążenia jaki może być dostarczony do modułów peryferyjnych instalacyjnej magistrali odległej wynosi 4,5 A lub 16 A. Prąd 4,5 A pozwala na przykład na przyłączenie 10 modułów wejść cyfrowych CDI o obciążalności 195 mA każdy, 10 modułów wyjść cyfrowych CDO o obciążalności 196 mA oraz 54 inicjatorów sygnałów takich jak czujniki o obciążalności 25 mA każdy. Wyjścia modułów muszą być zasilane niezależnie, ponieważ wymagają one zazwyczaj zasilania prądem od 500 mA do 2 A.



Rys.5.3.5 Moduł terminalowy BT typu IBS ST 24 BK LB-T (z lewej) oraz moduł terminalowy IBS ST 24 BKM-T (z prawej)

5.3.7 Magistrala lokalna (*Local Bus*)

Magistrala lokalna LB została zaprojektowana w celu umożliwienia elastycznego projektowania zdecentralizowanych podstacji sterowania wbudowanych w szafy sterownicze. Magistrala ta zastępuje skrzynki krosownicze, wykorzystywane w tradycyjnych metodach okablowania obiektów sterowanych, do których są kierowane wszystkie sygnały z czujników i urządzeń wykonawczych. Sygnały te są następnie połączone do modułów wejścia/wyjścia magistrali LB - modułów funkcyjnych I/O. Magistrala lokalna znajduje zastosowanie tam, gdzie w niewielkiej odległości od siebie występuje wiele czujników i elementów wykonawczych. Maksymalna długość magistrali LB może osiągnąć 10 m, a maksymalna odległość między dwoma modułami wejścia/wyjścia wynosi 1.5 m. Pierwszym modulem magistrali LB jest moduł terminalowy BT, przyłączony bezpośrednio do magistrali odległej RB. Moduł ten zawiera zintegrowany zasilacz napięcia stałego, który zasila układ logiczny magistrali w modułach terminalowych oraz, poprzez

kable magistrali lokalnej, układy logiczne magistrali w połączonych modułach funkcyjnych I/O. Strona urządzeń peryferyjnych, takich jak czujniki i urządzenia wykonawcze systemu InterBus-S przyłączonych do magistrali LB, jest zasilana ze źródła napięcia, które jest elektrycznie odizolowane od układu logicznego magistrali. W magistralach lokalnych przewidziano wykorzystanie poziomów napięć przyjętych w technice TTL (CMOS). Podobnie jak w magistrali odległej, transmisja danych w magistrali lokalnej odbywa się z prędkością 500 kbit/s w obu kierunkach (tzw. *full duplex*). Podstawowe dane techniczne magistrali lokalnej zawiera tab.5.3.3, natomiast dane techniczne systemu InterBus-S zostały przedstawione w tab.5.3.4.

Tab.5.3.3 Dane techniczne magistrali lokalnej LB

Max ilość elementów magistrali lokalnej	8 na segment w sumie 256
Interfejs	15-pin D-SUB, COMBICON
Max odległość pomiędzy dwoma elementami magistrali lokalnej typu SL	1,5 m
Max całkowita długość segmentu	10 m
Separacja galwaniczna	opcjonalnie
Napięcie zasilania	poprzez kabel magistrali (max 1A)

Tab.5.3.4 Podstawowe dane techniczne systemu InterBus-S

Liczba urządzeń I/O	max 512 (256 <i>Bus Terminal</i>)
Długość kabla segmentu magistrali odległej	max 400m
Całkowita długość magistrali IBS	12.8 km
Prędkość transmisji	500 kBit/s
Liczba punktów I/O	max 4096
Długość magistrali lokalnej	max 10 m
Długość instalacyjne magistrali odległej	max 50 m
Zabezpieczenie transmisji danych	CR-Check, Hd 4
Protokół komunikacyjny	DIN E 19258

5.3.8 Magistrala lokalna typu ST (moduły typu *Compact*)

Magistrala lokalna ST (*Smart Terminal*) składa się z modułu terminalowego (przyłączeniowego) (np. IBS ST 24 BK-T) oraz modułów peryferyjnych I/O (np. IB ST 24 DI/O). Moduły te są instalowane w szafach sterowniczych w bezpośrednim sąsiedztwie maszyn i urządzeń wytwórczych. Magistrala ST jest zasilana napięciem z odpowiadającego jej modułu terminalowego BT.

Urządzenia I/O typu ST są ze sobą połączone kablem magistrali lokalnej. Układy logiczne magistrali ST są zasilane z modułu terminalowego poprzez kabel magistrali lokalnej ST. W jednej magistrali lokalnej typu ST może być zainstalowanych 8 urządzeń I/O (modułów I/O). Maksymalny

prąd jaki mogą pobierać moduły I/O z modułu terminalowego wynosi 800 mA. Zależnie od typu urządzenia prąd pobierany przez moduły I/O wynosi od 20 do 250 mA.

Rodzina modułów terminalowych BT oraz modułów peryferyjnych I/O typu COMPACT jest bardzo liczna, przy czym należy wspomnieć, że wszystkie moduły peryferyjne I/O mają wejścia oraz wyjścia zgodne elektrycznie ze standardami przyjętymi w automatyce przemysłowej. Można w nich znaleźć moduły terminalowe BT z:

- interfejsem dla magistrali odległej typu RS485 lub światłowodowym,
- jednym lub dwoma interfejsami do magistrali lokalnej,
- z jednym lub dwoma interfejsami do magistrali odległej typu *Remote OUT*,
- możliwością przyłączenia 4 lub 8 modułów peryferyjnych I/O oraz
- moduły z wbudowanymi układami wejść/wyjść (I/O) peryferyjnych w standardzie 24V DC.

Generalnie moduły peryferyjne I/O (*Input/Output*) można podzielić na dwie części: moduły cyfrowe oraz moduły analogowe. Moduły cyfrowe występują w postaci 8, 16 lub 32 wejść lub wyjść dwustanowych 24V DC o zróżnicowanej obciążalności od 250 mA do 500 mA. Moduły analogowe z serii COMPACT mają 4 wejścia lub wyjścia analogowe o standardowym zakresie prądowym 0/4-20 mA lub napięciowym 0-10 V. Dostępne są także specjalizowane moduły analogowe do bezpośredniego przyłączenia czujników temperatury (PT 100, termopary).

5.3.9 Magistrala lokalna typu SL (moduły typu *Slim Line*)

Alternatywą dla magistrali lokalnej typu ST jest magistrala SL (*Slim Line*). Magistrala ta znajduje zastosowanie tam, gdzie występuje w niewielkiej odległości od siebie (10 m), wiele czujników i elementów wykonawczych. Maksymalna długość magistrali LB typu SL może osiągnąć 10 m, a maksymalna odległość między dwoma modułami wejścia/wyjścia (I/O) wynosi 1.5 m. Urządzeniami magistrali lokalnej typu SL, obok cyfrowych oraz analogowych modułów peryferyjnych I/O, mogą być np. moduły *slave* - panele operatorskie, urządzenia DRIVECOM oraz ENCOM.

W jednej magistrali lokalnej tego typu może być zainstalowanych 8 urządzeń I/O (modułów I/O). Maksymalny prąd, jaki mogą pobierać moduły I/O z modułu terminalowego wynosi 800 mA.

5.3.10 Okablowanie sieci InterBus-S

Dla sieci IBS są aktualnie dostępne następujące media transmisyjne:

- kabel miedziany (RS-485, transmisja na dystansie 400 m),
- światłowód (z tworzywa sztucznego 50-100 m, HCS 300 m, szklany 2500-3700 m),
- transmisja mikrofalowa oraz
- bariery foto-elektryczne.

5.3.10.1 Kabel miedziany - skrętka ekranowana

System okablowania sieci InterBus-S składa się z dwóch podstawowych typów kabli: kabel magistrali odległej RBC (ang. *Remote Bus Cable*) oraz kabel instalacyjnej magistrali odległej IRBC (ang. *Installation Remote Bus Cable*).

Kabel magistrali odległej jest zbudowany z trzech skręcanych par przewodów, oplecionych pojedynczym, wspólnym oplotem pełniącym rolę ekranu. Pary przewodów pełnią tutaj zadanie transmisji sygnałów elektrycznych, nie pełnią natomiast roli linii zasilających w energię elektryczną. Kabel instalacyjnej magistrali odległej ma budowę podobną do typowego kabla magistrali odległej za wyjątkiem tego, że zawiera trzy dodatkowe przewody do zasilania urządzeń peryferyjnych sieci InterBus-S a nie przekraczających w sumie prądu obciążenia 4.5 A (tzw. hybrydowa budowa kabla). Pełna specyfikacja kabli wykorzystywanych w sieci InterBus-S jest zdefiniowana w suplemencie do normy InterBus-S DIN E 19 258. W normie tej zawarte są także wytyczne dla celów certyfikacji kabli. Ponadto kable typu RBC oraz IRBC są dostępne w wersji giętkiego przewodu izolowanego do zastosowania w ruchomych instalacjach sieciowych, np. na częściach ruchomych manipulatorów.

5.3.10.2 Kable światłowodowe

Alternatywą dla kabli miedzianych w sieci InterBus-S są kable światłowodowe FO (ang. *Fibber Optic*). Stosowanie światłowodów jest zalecane w szczególności w środowisku silnych zakłóceń elektromagnetycznych oraz wówczas, gdy wymagana jest transmisja danych na dużych odległościach, np. pomiędzy budynkami. Istnieją dwie metody transmisji danych w sieci InterBus-S wykorzystującej kable światłowodowe: poprzez konwerter sygnałów standardu RS 485 na standard kabla światłowodowego oraz poprzez moduły terminalowe BT wyposażone w interfejs światłowodowy (rys.5.3.5). Użycie transmisji światłowodowej niesie za sobą wiele następujących korzyści: odporność na zakłócenia elektromagnetyczne, całkowita izolacja elektryczna pomiędzy nadajnikiem a odbiornikiem (nie istnieje problem różnicy potencjałów), brak przesłuchów sygnałów, wysoka prędkość transmisji oraz możliwość zastosowania w środowiskach niebezpiecznych i zagrożonych wybuchem.

Konwertery standardu RS485/FO są dostępne w wersji dla światłowodu szklanego (ang. *glass fibber*) oraz w wersji światłowodu z tworzywa sztucznego (ang. *polymer fibber*). Światłowody szklane są wykorzystywane do transmisji danych na duże odległości, np. konwerter PSM-EG-RS485/FO, który jest używany w sieci InterBus-S, umożliwia transmisję danych na odległość do 3700 m. Konwerter ten ma interfejs typu ST i jest dostosowany do transmisji danych poprzez gradientowy, dupleksowy światłowód szklany o rozmiarach włókna 62.5/125 μm . Kabel ten umożliwia transmisję sygnału optycznego w dwóch oknach 850 lub 1300 nm z tłumieniem 3.50 dB/km.

Światłowody z tworzyw sztucznych pozwalają na transmisję sygnałów na odległość nie przekraczającą 100 m. Wspomniany już konwerter PSM-EG-RS485/FO transmituje sygnały poprzez światłowód z tworzyw sztucznych na odległość 50 m. Światłowody z tworzyw sztucznych wykorzystywane w sieci InterBus-S charakteryzują się następującymi parametrami: rozmiary włókna - 980/1000 μm , długość fali - 660 nm, tłumienie sygnału < 220 dB/km, typ łącza F-SMA. W porównaniu ze światłowodem szklanym, światłowód z tworzyw sztucznych odznacza się dość krótkim dystansem na jaki można transmitować sygnały bez pośredniczącego wzmacniacza sygnałów, jednakże w wielu przypadkach jest to dystans wystarczający. Podstawowym powodem, dla którego często i chętnie się stosuje ten typ kabla, jest jego bardzo atrakcyjna cena oraz prosta obróbka i montaż (np. złącza F-SMA).

5.3.11 InterBus-Loop - technika łączenia dyskretnych oraz analogowych urządzeń peryferyjnych

System komunikacyjny InterBus-S pozwala na łączenie zarówno złożonych urządzeń automatyki przemysłowej, takich jak sterowniki PLC czy też komputery PC, jak i prostych urządzeń peryferyjnych - analogowych oraz dwustanowych czujników i elementów wykonawczych. W celu zwiększenia efektywności i przepustowości protokołu komunikacyjnego systemu InterBus-S oraz obniżenia kosztów instalacji, przy łączeniu do magistrali systemowej pojedynczych urządzeń peryferyjnych takich jak czujniki i elementy wykonawcze, zastosowano zmianę metody transmisji danych - tzw. pętlę InterBus (ang. *InterBus-Loop*).

Tab.5.3.5 Podstawowa specyfikacja systemu *InterBus-Loop*

<i>Medium transmisyjne</i>	kabel nie ekranowany, 2-żyłowy, 2x1,5 mm ²
<i>Maksymalna długość kabla pomiędzy dwoma urządzeniami</i>	10 m
<i>Maksymalna długość kabla w pętli</i>	100 m
<i>Maksymalna liczba przyłączonych urządzeń</i>	63
<i>Kodowanie sygnału</i>	kod Manchester
<i>Maksymalny prąd obciążenia</i>	1.5 A
<i>Napięcie zasilające</i>	24 V DC
<i>Adresowanie</i>	brak
<i>Moduł terminalowy magistrali</i>	konwersja metody transmisji, brak bramy
<i>Szybkość działania</i>	3.7 ms dla 1000 I/O
<i>Urządzenia automatyki</i>	cyfrowe, analogowe, binarne i „inteligentne”

Pętla InterBus łączy pojedyncze peryferyjne urządzenia automatyki poprzez prosty, dwu-żyłowy i nie ekranowany kabel zgodny ze specyfikacją IEC-1000-4-4. Kabel ten pozwala na transmisję danych do/z 63 prostych (analogowych i dwustanowych) urządzeń automatyki. Jednakże

poprzez ten sam kabel, oprócz transmisji danych, jest dostarczane napięcie zasilające 24V DC do wszystkich czujników i elementów wykonawczych o obciążalności do 40 mA każdy. Specjalnie zakodowane złącze kabla, tzw. IDC (ang. *Insulation Displacement Connector*) zabezpiecza użytkownika przed omyłkową zmianą biegunowości połączenia. Transmisja danych w pętli InterBus odbywa się poprzez zakodowany kodem *Manchester* sygnał prądowy, który jest modulowany na napięciu zasilającym 24 V DC. Zakodowane dane są modulowane jako sygnał prądowy o wartości $I = 0 \text{ mA} \pm 10 \text{ mA}$, przez co znacząco zwiększa się odporność systemu na zewnętrzne zakłócenia elektromagnetyczne, w porównaniu z konwencjonalnym sygnałem napięciowym.

Przyłączenie pętli prostych urządzeń peryferyjnych *InterBus-Loop* do magistrali odległej RB systemu InterBus-S odbywa się poprzez moduł terminalowy, np. IBS SL 24 BK-T. Moduł ten jest odpowiedzialny zarówno za konwersję metody transmisji jak i zasilanie urządzeń pętli napięciem 24V DC. Podstawowe parametry pętli InterBus zostały przedstawione w tab.5.3.5.

W pętli IBS może płynąć prąd o maksymalnym natężeniu 1.5 A. Wyjścia pętli, które wymagają zazwyczaj prądu o natężeniu około 2 A, muszą być zasilane niezależnie.

System *InterBus-Loop*, ze względu na dość dobre parametry techniczne oraz niezawodnościowe, ma zastosowania w aplikacjach przemysłowych przy łączeniu pojedynczych czujników i elementów wykonawczych bezpośrednio na obiekcie technologicznym. Połączenia pętli muszą być wykonane wg zasad ochrony IP 65. Jeśli kabel pętli jest doprowadzony do szafy sterowniczej systemu automatyki, to może on także łączyć takie elementy szafy jak styczniki, miniaturowe wyłączniki automatyczne, wskaźniki czy też lampki kontrolne wyposażone w układ scalony IBS Loop.

Podstawowym założeniem stawianym podczas konstruowania pętli InterBus była pełna kompatybilność tego systemu z magistralą główną InterBus-S. Oznacza to, że połączenie InterBus-Loop do głównego systemu InterBus-S nie wymaga specjalnej konwersji protokołów, nie ma potrzeby stosowania kosztownych bram (ang. *gateway*), a skanowanie danych procesowych jest całkowicie zsynchronizowane z systemem głównym.

Innym wymogiem współpracy pętli InterBus z systemem InterBus-S jest to, aby peryferyjne urządzenia wykonawcze oraz czujniki umożliwiały wymianę danych w dwóch niezależnych kanałach transmisyjnych. Pierwszy kanał transmisyjny, tzw. kanał danych procesowych, zapewnia synchroniczne oraz stałe w czasie przesyłanie danych do/z urządzeń peryferyjnych. Krótki, tzn. nie przekraczający 0,5 ms, czas skanowania danych procesowych pozwala na wykorzystanie systemu InterBus-Loop w zamkniętych pętlach sterowania oraz w maszynach i urządzeniach wytwórczych. Drugi kanał transmisyjny, tzw. kanał danych parametryczny, może być wykorzystany do inicjalizacji czujników lub wymiany złożonych danych parametrycznych, np. danych diagnostycznych z inteligentnych czujników temperatury.

System *InterBus-Loop* stanowi bardzo interesujące uzupełnienie standardowego systemu InterBus-S. Prosta oraz uniwersalna technika łączenia urządzeń peryferyjnych oraz w pełni

kompatybilny protokół komunikacyjny składają się na stosunkowo niski koszt aplikacji przemysłowej złożonej z szeregu pojedynczych czujników i elementów wykonawczych.

5.3.12 Podstawowa specyfikacja systemu InterBus-S

Uwaga: W przedstawionej niżej specyfikacji systemu InterBus-S długość magistrali odległej RB została określona przy założeniu, że wykorzystywany jest kabel miedziany. Użycie technologii światłowodowej pozwala na znaczne wydłużenie dystansu nawet do kilkudziesięciu kilometrów przy wykorzystaniu światłowodu odpowiedniej jakości.

SYSTEM

liczba punktów I/O	4096
liczba słów danych	256
prędkość transmisji	500 kbit/s

URZĄDZENIA

liczba urządzeń magistrali odległej	256
liczba urządzeń InterBus-S	512

ODLEGŁOŚCI POMIĘDZY

sterownik systemu HCB - 1 moduł BT	400 m
dwoma urządzeniami magistrali RB	400 m
sterownik systemu HCB - ostatni moduł BT	12.8 km

MAGISTRALA LOKALNA (ST)

liczba modułów	max 8
prąd zasilania modułu BT	800 mA
liczba modułów	max 4
prąd zasilania modułu BT	400 mA

MAGISTRALA LOKALNA (SL)

liczba modułów	max 8 (4)
prąd zasilania modułu BT	800 mA (400 mA)
odległości pomiędzy urządzeniami	max 1.5m
długość magistrali lokalnej	max 10 m

MAGISTRALA LOKALNA (SL)

prąd zasilania modułu BK	max.4.5 A (16 A)
odległości pomiędzy urządzeniami	max 50 m
długość magistrali lokalnej	max 50 m

INTERBUS-S LOOP

długość pętli	max 100 m
odległości pomiędzy urządzeniami	max 10 m

max prąd obciążenia

max 1.5 A

liczba urządzeń

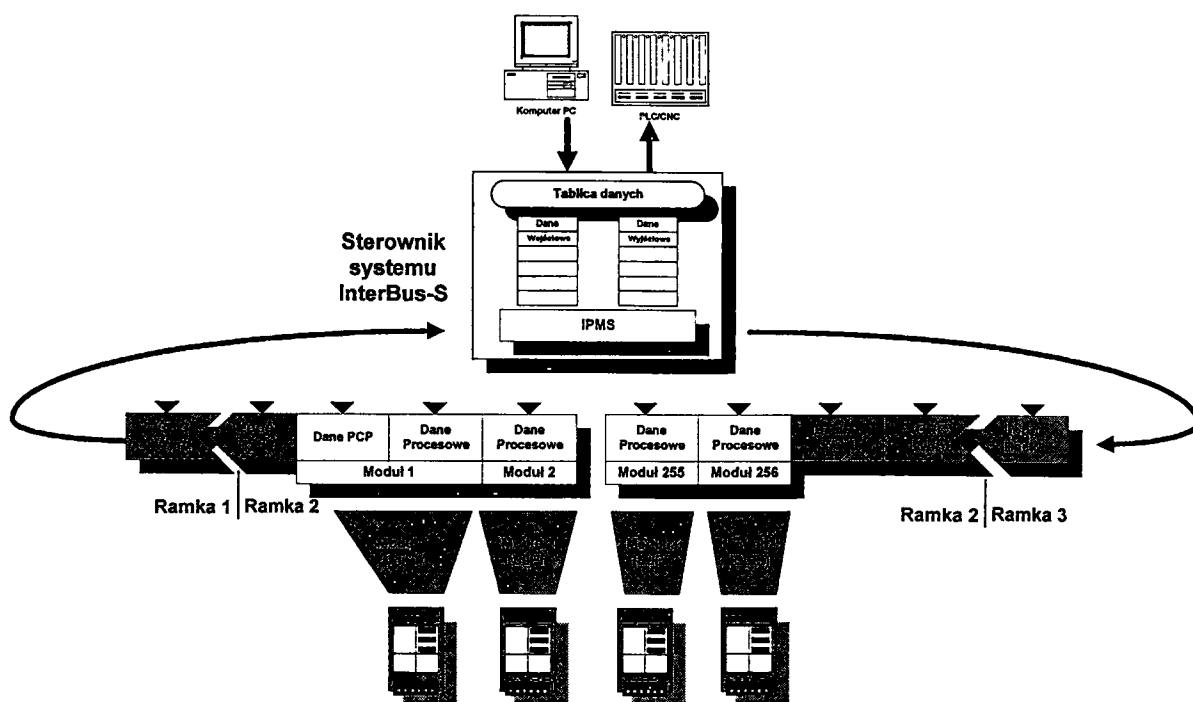
max 63

KATEGORIA EMC

EMC: IEC 801-4 kategoria 4

5.3.13 Format protokołu komunikacyjnego

Protokół komunikacyjny systemu InterBus-S (IBS) jest ściśle związany z jego strukturą sprzętową. Podstawowymi elementami systemu InterBus-S są dwa układy scalone: IPMS (*InterBus-S Protocol Master*) oraz SμPI (*Serial Microprocessor Interface*). Głównym zadaniem układu IPMS jest sterowanie i zarządzanie siecią i dlatego też jest on implementowany w centralnym sterowniku systemu, tzw. *Host Controller Board*. W praktyce sterownik ten stanowi kartę do komputera przemysłowego (ISA, VME) lub do sterowników przemysłowych PLC, a także CNC. Natomiast peryferyjne urządzenia automatyki, takie jak: czujniki, zadajniki, inteligentne sterowniki lokalne, są łączone do sieci IBS za pomocą układu scalonego SμPI. W typowym systemie IBS może pracować tylko jeden centralny sterownik systemu z układem IBMS oraz wiele modułów (256) urządzeń peryferyjnych z układem SμPI. Rysunek 5.3.6 pokazuje, przyjęty w systemie InterBus-S, sposób przesyłania danych i komunikatów pomiędzy sterownikiem systemu a urządzeniami peryferyjnymi. Poszczególne moduły I/O wraz z centralnym sterownikiem systemu stanowią pewnego rodzaju rejestr przesuwany, którego wielkość jest zależna od sumy danych procesowych poszczególnych modułów I/O.



Rys.5.3.6 Szeregowy przepływ danych w sieci InterBus-S

Dane procesowe w postaci słów 16-bitowych są przesuwane poprzez poszczególne rejestry modułów I/O w takt zegara systemowego. Liczba wymaganych taktów zegara w jednym cyklu przesyłania danych jest proporcjonalna do ilości słów danych procesowych. Dlatego też szybkość systemu, tj. zdolność do uaktualniania danych jest zależna od jego konfiguracji. Przejście wszystkich danych procesowych do/z poszczególnych modułów nazywamy cyklem skanowania. Dane procesowe zawierające informacje wyjściowe dla urządzeń peryferyjnych (elementów wykonawczych) są wpisywane do układów S μ PI, gdzie następnie są doprowadzane do właściwego wyjścia elementu wykonawczego. Natomiast dane procesowe, zawierające informacje wejściowe z czujników opuszczając układ S μ PI są doprowadzane do centralnego sterownika systemu IBMS. Ponieważ dane wejściowe i wyjściowe procesu są transmitowane równocześnie poprzez sieć (*full duplex*) w takt zegara systemowego, to dodatkową zaletą tego rozwiązania jest możliwość synchronizacji zdarzeń występujących na obiekcie automatyki.

5.3.14 Tryby pracy systemu IBS

Przesyłanie danych procesowych do/z poszczególnych modułów I/O w systemie InterBus-S jest poprzedzone cyklem identyfikacyjnym ID. Cykl ten jest inicjalizowany przez centralny sterownik sytemu, a jego podstawowym zadaniem jest dostarczenie informacji o konfiguracji sieci w szczególności zaś o typie i pozycji poszczególnych modułów I/O podłączonych do sieci. Każde urządzenie peryferyjne systemu IBS posiada 16-to bitowy rejestr identyfikacyjny, w którym znajdują się informacje o rodzaju urządzenia (cyfrowe, analogowe), wielkości danych wymaganych przez urządzenie (1, 2, 4 słowa) oraz o występujących błędach w module, np. CRC (*Cyclic Redundancy Check*). Dzięki zastosowaniu cyklu identyfikacji ID w systemie IBS nie wymaga się ręcznego ustawiania adresów poszczególnych modułów.

Działanie cyklu identyfikacyjnego ID można przedstawić w kilku następujących krokach:

1. Układ IPMS wysyła do wszystkich modułów I/O (S μ PI) telegram z informacją o przełączeniu systemu w tryb ID.
2. Układy S μ PI przełączają się w tryb ID.
3. Układ IPMS wstawia w kolejkę transmitowanych danych słowo końca pętli danych „*loop back*”. Układ S μ PI wstawia w kolejkę danych swój numer kodu ID.
4. Sterownik systemu odczytuje co 16 taktów zegara (16 bit) poszczególne numery kodu ID.
5. Sterownik systemu IPMS buduje, na podstawie przychodzących kodów ID, tablicę wszystkich modułów połączonych do sieci. Kiedy znaki sterujące końca pętli „*loop back*” powrócą do układu IPMS, to odczytywanie kodu ID jest zakończone.

Po zakończeniu cyklu identyfikacji ID, w układzie sterownika sytemu IPMS znajdują się wszystkie niezbędne informacje do uruchomienia cyklu przesyłania danych procesowych (cyklu skanowania).

Cykl przesyłania danych procesowych, podobnie jak cykl identyfikacyjny, przebiega w kilku następujących krokach:

1. Układ IPMS wysyła do wszystkich modułów I/O telegram z informacją, że wprowadza sieć w tryb przysyłania danych procesowych (cykl skanowania).
2. Wszystkie występujące w sieci układy S μ PI umieszczają wejściowe słowa danych pochodzące z czujników do rejestrów przesuwnych. Natomiast dane wyjściowe dla zadajników (elementów wykonawczych) są ustawiane w układzie sterownika IPMS.
3. Układ IPMS rozpoczyna przesuwanie danych taktowane zegarem systemowym ($n \cdot 16$), gdzie n oznacza ilość słów danych procesowych.
4. Przesuwanie danych jest kontynuowane do tego czasu, aż wszystkie dane procesowe dotrą do swoich układów - dane wyjściowe do S μ PI, natomiast dane wejściowe do IPMS.
5. Kiedy wszystkie dane dotrą do swoich układów, to rozpoczyna się faza nazywana fazą zatraskową (ang. *Latch Phase*). W fazie tej układ IPMS sprawdza sumę kontrolną CRC generowaną przez ostatni układ S μ PI w systemie w celu zbadania integralności przesyłanych danych.
6. Po pozytywnym wyniku testu CRC, układ IPMS zezwala na udostępnienie danych procesowych urządzeniom peryferyjnym (czujnikom/ elementom wykonawczym).
7. Krok 2 do 6 są powtarzane do czasu, aż sterownik systemu zatrzyma przesyłanie danych.

5.3.15 Sterowanie siecią InterBus-S

Zarówno podczas cyklu identyfikacyjnego ID jak i podczas cyklu przesyłania danych procesowych (skanowania), centralny sterownik systemu IPMS transmituje telegram z informacją sterującą (ang. *Telegram Message*) do wszystkich układów S μ PI pracujących w sieci. Telegram ten dostarcza bieżących informacji do układów S μ PI o stanie sieci oraz jest odpowiedzialny za synchronizację prędkości transmisji. Każdy telegram z informacją sterującą złożony jest z 5 bitów (*Flag, Control, Mode, Start, Stop*) i wysyłany jest po przesunięciu każdego bajtu danych (8 bit) w cyklu ID lub w cyklu skanowania danych. Aby telegram dotarł do wszystkich układów S μ PI równocześnie, układy te przy pomocy specjalnych przełączników wprowadzają wejściowy rejestr przesuwny w wielo-gałęziowy tryb pracy (ang. *multidrop*).

Jak już wcześniej wspomniano system InterBus-S wykorzystuje technikę transmisji danych określaną jako metoda przekazywania wspólnej, całkowitej ramki (ang. *One total frame*) co oznacza, że całość danych procesowych i identyfikacyjnych do/z modułów I/O jest przesyłana w jednym cyklu transmisyjnym. Cykl ID oraz cykl skanowania mają dwa różne typy ramek. Transmisja tych ramek rozpoczyna się od bloku znaków kontrolnych końca pętli, tzw. *Loop-back*. Wszystkie dane procesowe w cyklu skanowania lub dane identyfikacyjne w cyklu ID są transmitowane w bloku danych tuż za znakiem *Loop-back*. Ostatnimi danymi w ramce transmisyjnej są kolejno: blok sumy kontrolnej CRC oraz blok słowa sterującego. Blok CRC (ang. *Cyclic Redundancy Check*) jest odpowiedzialny za kontrolę błędów wynikłych podczas transmisji

danych, natomiast blok słowa sterującego (ang. *Control Word*) przechowuje informację o miejscu wystąpienia błędu.

5.4 Sieć CAN

5.4.1 Wstęp

Rozwój elektroniki samochodowej oraz wzrost jej zastosowań spowodował potrzebę integracji systemów elektronicznych w celu zapewnienia większej ich niezawodności oraz zmniejszenia ilości kabli niezbędnych do ich funkcjonowania. Potrzebom przemysłu samochodowego wyszła na przeciw firma *Bosch* z projektem systemu CAN (ang. *Controller Area Network*), integrującego układy sterowania dla nowych konstrukcji pojazdów samochodowych. Parametry, jakimi charakteryzuje się magistrala CAN, tj. niski koszt, odporność na zakłócenia elektromagnetyczne, duża szybkość transmisji, niska podatność na błędy, spowodowały duże zainteresowanie tym systemem również producentów automatyki przemysłowej i aparatury medycznej.

Magistrala CAN posiada następujące cechy:

- system z wieloma stacjami typu *master*,
- wykorzystuje asynchroniczną magistralę szeregową,
- szybkość transmisji danych do 1Mbit/s,
- długość magistrali do 40 m (przy szybkości 1Mbit/s),
- ramka transmisyjna może zawierać od 0 do 8 bajtów danych,
- wykrywanie i obsługa błędów transmisyjnych,
- łatwa rozbudowa sieci,
- arbitraż bitowy CSMA/CA.

Specyficzną cechą systemu CAN jest rozsyłanie danych do wszystkich urządzeń sieci tzn., że komunikaty wysyłane na magistralę są dostępne dla każdego urządzenia. Identyfikatory ramek transmisyjnych określają tylko typ przesyłanych danych oraz ich priorytet. Urządzenia systemu CAN mogą być połączone strukturą typu magistrala lub gwiazda (linie światłowodowe). Mogą też występować mieszane rozwiązania połączeń. Wg firmy *Bosch* w systemie czasu rzeczywistego eksploatowanym w samochodzie występuje ok. 3.95×10^{-3} nie wykrytych błędów podczas całej jego eksploatacji (4000 h, 1 Mbit/s, obciążenie magistrali 30%, długość ramki 90 bitów, a łączna ilość danych 4.8×10^{10}).

5.4.2 Warstwa fizyczna

Sygnały na magistrali sieci CAN kodowane są metodą NRZ (ang. *Non-Return-to-Zero*).

Reprezentacja bitowa sygnału jest następująca:

- bit recesywny - poziom logiczny '1',
- bit dominujący - poziom logiczny '0'.

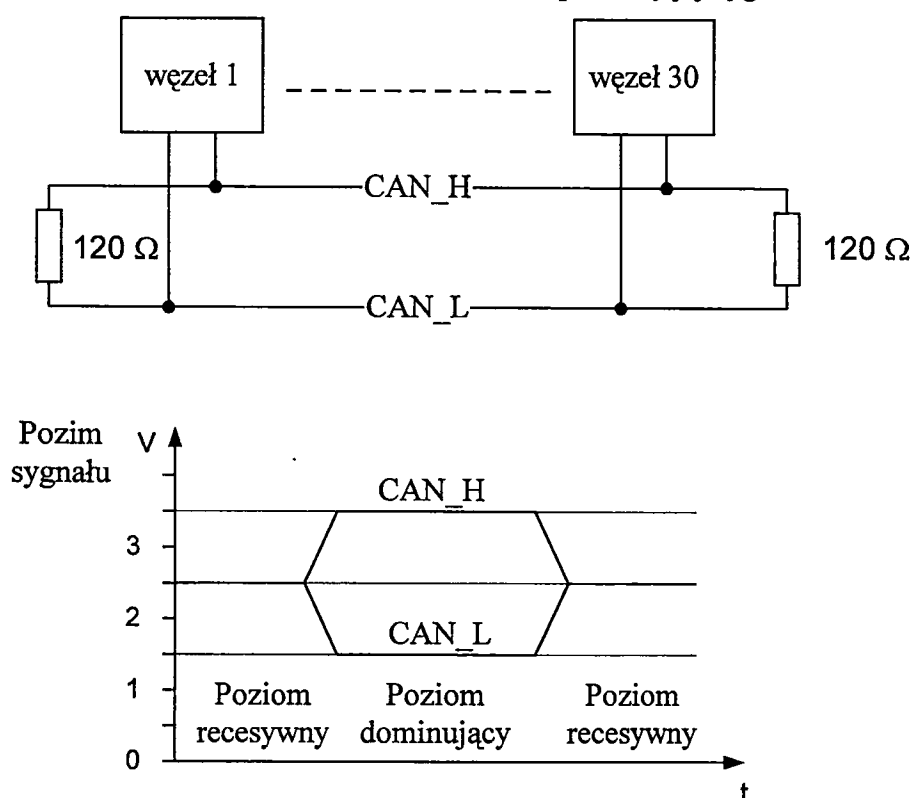
Długość magistrali zależy od szybkości transmisji (tab.5.4.1). Interfejs transmisyjny magistrali CAN określony jest normami ISO-DIS 11 898 i ISO-DIS 11 519. Spotykane są również modyfikacje interfejsu transmisyjnego przeznaczone dla specyficznych zastosowań, takich jak: ABUS, VAN (ang. *Vehicle Area Network*), J1850-DLC/-HBCC i inne.

Tab.5.4.1 Zależność długości magistrali od szybkości transmisji w sieci CAN

Szybkość transmisji [kbit/s]	Maksymalna długość magistrali [m]
1000	40
500	112
300	200
200	310
100	640
50	1300

5.4.3 CAN High-Speed (ISO-DIS 11898)

Na rysunku 5.4.1 zamieszczono strukturę sieci CAN i poziomy jej sygnałów.



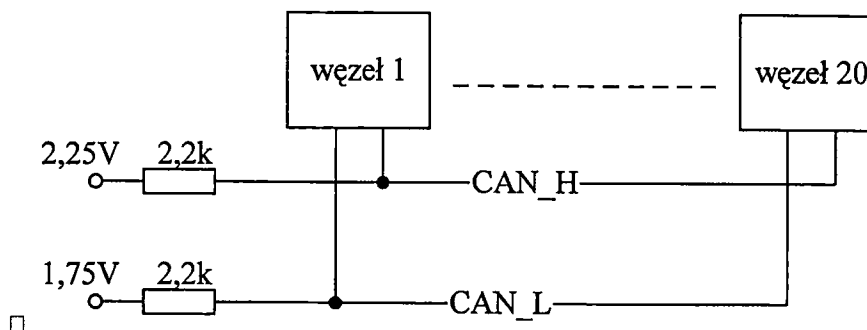
□ Rys.5..41 Struktura sieci i poziomy sygnałów wg normy ISO 11898

Interfejs transmisyjny *CAN High-speed* charakteryzuje się następującymi parametrami:

- szybkość transmisji 125 kbit/s - 1 Mbit/s,
- maks. długość magistrali 40 m przy 1 Mbit/s,
- 2 - 30 węzłów na sieć,
- impedancja terminatorów sieci 120 Ω,
- kabel dwużyłowy ekranowany,
- zasilanie 5V.

5.4.4 CAN Low-Speed (ISO-DIS 11519)

Specyfikacja *CAN Low-Speed* (ISO-DIS 11519) określa budowę i parametry sieci typu CAN, które pracują z szybkościami transmisji do 125 kbit/s. W sieci takiej może się znajdować do 20 węzłów (rys.5.4.2). W tab.5.4.2 zamieszczono poziomy sygnałów wg normy ISO 11519.



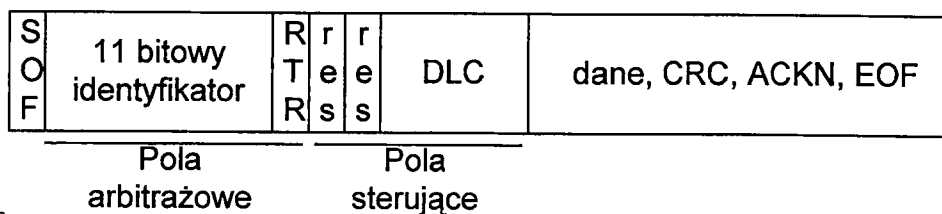
Rys.5.4.2 Struktura sieci CAN wg ISO 11519

Tab.5.4.2 Poziomy sygnałów wg normy ISO 11519

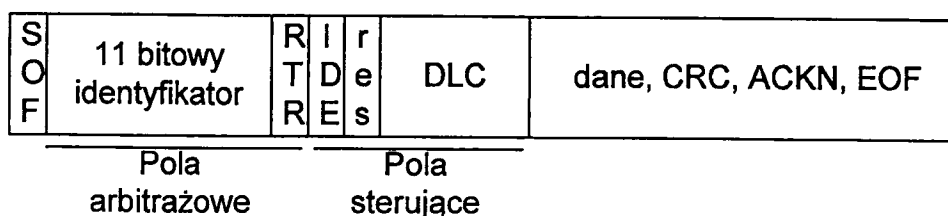
	Bit recesywny	Bit dominujący
V CAN_H	3.1 - 3.4V	0 - 1.15V
V CAN_L	1.6 - 1.9V	3.85 - 5.0V

5.4.5 Protokół komunikacyjny

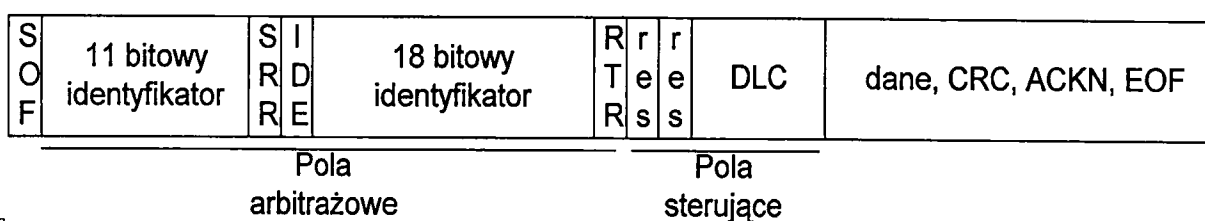
W systemie CAN występują dwa formaty ramek transmisyjnych, które różnią się przede wszystkim długością identyfikatora. Ramki transmisyjne formatu standardowego mają 11-bitowe pole identyfikatora, zaś w formacie rozszerzonym identyfikator ma długość 29 bitów. Rozszerzenie identyfikatora do 29 bitów miało na celu standaryzację identyfikatorów komunikatów zgodnie z normą SAE J1939. W sieci zawierającej kontrolery protokołu CAN zgodne z formatem standardowym (specyfikacje 1.0, 1.1, 1.2 i 2.0 A *Standard Frame*) mogą być przesyłane jedynie ramki transmisyjne formatu podstawowego (rys.5.4.3). W tym przypadku ramki formatu rozszerzonego (specyfikacja 2.0B *Extended Format*) (rys.5.4.5) będą niezrozumiałe. Nowsze kontrolery sieciowe umożliwiają rozróżnienie wersji protokołu w transmitowanych ramach.



Rys.5.4.3 Ramka transmisyjna wg CAN Specification Ver. 1.0, 1.1, 1.2, 2.0A



Rys.5.4.4 Ramka transmisyjna wg *CAN Specification Ver. 2.0B Standard Frame*



Rys.5.4.5 Ramka transmisyjna wg *CAN Specification Ver. 2.0B Extended Frame*

Skrótowe nazwy pól bitowych ramek transmisyjnych przedstawionych na rysunkach 5.4.3 - 5.4.5 wyjaśnione są poniżej jako:

SOF - *Start of Frame*,

IDE - *Identifier Extension*,

RTR - *Remote Transmission Request*,

SRR - *Substitute Remote Request bit*,

DLC - *Data Length Code* - 4 bity,

data - 0.8 bajtów danych,

CRC - *Cyclic Redundancy Check* - 16 bitów,

ACKN - *Acknowledge* - 2 bity,

EOF - *End of Frame* - 7 bitów,

res - zarezerwowane do zastosowań w późniejszych wersjach protokołu.

5.4.6 Typy ramek transmisyjnych

W zależności od rodzaju przesyłanych komunikatów rozróżnia się cztery następujące typy ramek transmisyjnych:

- *Data Frame* - transfer danych,
- *Remote Frame* - żądanie przesłania danych,
- *Error Frame* - informacja o wykrytych błędach,
- *Overload Frame* - rozszerzenie przerwy między sekwencją ramek transmisyjnych.

5.4.6.1 Ramka danych (ang. *Data Frame*)

Ramka transmisyjna danych (ang. *Data Frame*) składa się z następujących pól bitowych:

- bitu startu ramki - *Start-Of-Frame*,

- identyfikatora - *Arbitration Field (Identifier + RTR bit)*,
- pola sterującego - *Control Field (rezerwa 2 bity + Data Length Code)*,
- pola danych - *DATA*,
- pola kontrolnego - *CRC Field (CRC Sequence + CRC Delimiter)*,
- potwierdzenia - *Acknowledge Field (ACK Slot + ACK Delimiter)*,
- sekwencji kończącej ramkę - *End-Of-Frame*.

Start-Of-Frame jest początkiem ramek typu *Data* lub *Remote*. Składa on się z jednego bitu dominującego (stan niski magistrali). Bit ten zapewnia synchronizację urządzeń pracujących w sieci (ang. *hard synchronisation*).

Arbitration Field składa się z identyfikatora komunikatu i bitu *RTR*. Identyfikator komunikatu jest 11-bitowym (ramki standardowe) lub 29-bitowym (ramki rozszerzone) polem, które zawiera informację o komunikacie i priorytecie dostępu do magistrali. Identyfikator nie określa dokładnie adresu odbiorcy (w sieciach opartych na magistrali CAN nie rozróżnia się komunikacji typu punkt-punkt czy typu *broadcast*). Im niższy numer binarny komunikatu, tym wyższy jest jego priorytet.

Urządzenie pracujące jako odbiornik może zainicjować transmisję pożądaných danych przez wysłanie ramki zdalnej (ang. *Remote Frame*). Identyfikator tej ramki zawiera adres źródła danych, a bit *RTR* jest w stanie wysokim (recesywny poziom magistrali). Na podstawie tego bitu odróżnia się ramki typu *DATA* od ramek typu *REMOTE*, tj. ramki danych mają bit *RTR* w stanie niskim, zaś ramki zdalne w stanie wysokim.

W ramach transmisyjnych formatu rozszerzonego (CAN 2.0B *Extended Format*) bit *SRR* (ang. *Substitute Remote Request*) jest transmitowany z poziomem recesywnym, co daje wyższy priorytet standardowym ramkom transmisyjnym (CAN 2.0A i 2.0B *Standard Frame*) o tym samym identyfikatorze bazowym.

Pole *Control Field* składa się z 6 bitów. Dwa bity są zarezerwowane dla przyszłych rozwiązań protokołu CAN, zaś na pozostałych czterech bitach zawarta jest informacja o ilości bajtów *DLC* (ang. *Data Length Code*) przesyłanych w polu *DATA*.

Data Field jest to pole danych składające się z 0-8 bajtów (w zależności od pola *DLC*). Pierwszym bitem tego pola jest *MSB* pierwszego bajta.

Cyclic Redundancy Code Field (CRC) jest to pole *CRC* składa się z 15-to bitowej sekwencji *CRC* oraz jednego bitu *CRC delimiter*'a.

Przy obliczaniu *CRC* stosuje się wielomian generacyjny w postaci:

$$f(x) = (x^{14} + x^9 + x^8 + x^6 + x^5 + x^4 + x^2 + x + 1)(x+1) = 1100010110011001$$

Odległość Hamming'a w tym przypadku wynosi $d=6$.

Acknowledge Field jest to pole składające się z dwóch bitów, *Acknowledge Slot* i *Acknowledge Delimiter*, które są wysyłane z poziomem recesywnym. Odbiorcy informacji raportują poprawność odczytanej ramki przez ustawienie bitu *Acknowledge Slot* w stan dominujący. Nadawca ramki ciągle monitorując stan tego pola może stwierdzić, czy przynajmniej jeden odbiorca poprawnie odebrał ramkę transmisyjną.

Każda ramka transmisyjna typu *DATA* lub *REMOTE* jest zakończona sekwencją *End-Of-Frame* zawierającą siedem recesywnych bitów.

5.4.6.2 Ramka zdalna (ang. *Remote Frame*)

Urządzenie pracujące w sieci jako odbiornik może zainicjować transmisję danych przez wysłanie ramki transmisyjnej typu *REMOTE* do sieci z adresem źródła danych zawartym w identyfikatorze ramki oraz bitem *RTR* w stanie wysokim (recesywny poziom magistrali).

Ramka zdalna (*Remote Frame*) składa się z sześciu następujących pól bitowych:

- *Start-Of-Frame*,
- *Arbitration Field*,
- *Control Field*,
- *CRC-Field*,
- *Acknowledge Field*,
- *End-Of-Frame*.

Bit *RTR* jest ustawiony w stan wysoki. Wartość *DLC* (ang. *Data Length Code* w polu *Control*) jest ignorowana, ale powinna być zgodna z długością pola inicjowanych danych. Ramka zdalna jest podobna do ramki danych.

5.4.6.3 Ramka błędów (ang. *Error Frame*)

Podczas transmisji mogą wystąpić następujące błędy:

- bitowy,
- pola arbitrażu,
- sumy kontrolnej,
- formatu ramki,
- potwierdzenia.

Urządzenie (odbiornik), które wykryło błąd transmisji natychmiast wysyła na magistralę informację o wystąpieniu błędu i powoduje przerwanie transmitowanej ramki.

Ramka informująca o błędach (ang. *error frame*) składa się z dwóch pól bitowych:

- *error flag* oraz
- *error delimiter*.

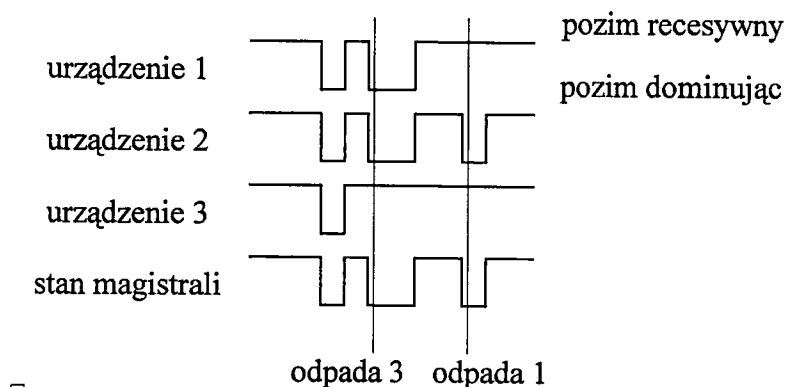
Pole *error flag* składa się z od sześciu do dwunastu kolejnych bitów dominujących (ilość bitów może się wahać w zależności od ilości urządzeń raportujących błąd w danej chwili). Po przesłaniu flagi błędu wysyłany jest *error delimiter* składający się z ośmiu bitów recesywnych. Jeśli błąd zostanie zasygnalizowany podczas wysyłania ramki danych lub ramki zdalnej, to następuje retransmisja błędnego komunikatu.

5.4.6.4 Overload Frame

Ramka *Overload Frame* jest wysyłana wtedy, kiedy urządzenie ma „problemy” z przetwarzaniem danych kolejno odbieranych z sieci i „chce” w ten sposób zwiększyć przerwę między ramkami transmisyjnymi, co umożliwi prawidłowe przetwarzanie danych.

5.4.7 Organizacja dostępu do magistrali

Początek transmisji następuje tylko podczas stanu biernego magistrali (*BUS-Idle*). Wszystkie urządzenia sieci są synchronizowane opadającym zboczem *Start-of-Frame* (ang. *hard synchronisation*). Konflikty związane z dostępem do magistrali są rozwiązywane podczas transmisji pola arbitrażu (ang. *Arbitration Field*) w procesie arbitrażu bitowego (ang. *bit-wise arbitration process*). Arbitraż polega na ciągłym monitorowaniu stanu magistrali podczas wysyłania ramki transmisyjnej. Urządzenie sieciowe, które wysłało bit recesywny, a odczytało z magistrali bit dominujący, automatycznie przechodzi w stan czytania. Identyfikator komunikatu definiuje priorytet dostępu do magistrali. Proces arbitrażu bitowego zapewnia bezpieczny dostęp do magistrali.



Rys.5.26 Arbitraż bitowy przy dostępie do magistrali

Przykładowy arbitraż dostępu do magistrali pokazano na rys.5.26, na którym trzy urządzenia próbują jednoczesnego dostępu do magistrali. W wyniku procesu arbitrażu na magistrali pozostaje urządzenie numer dwa. Pozostałe urządzenia automatycznie przechodzą w stan czytania.

Bity poszczególnych pól bitowych są kodowane metodą *bit-stuffing*'u, tj. jeśli występuje ciąg pięciu bitów o tej samej polaryzacji, to dodawany jest kolejny bit o przeciwnej polaryzacji. Bit ten jest pomijany w procesie czytania danych.

Powyższa metoda kodowania dotyczy pól:

- arbitrażu,
- sterowania,
- danych,
- CRC.

Odległość między ramkami transmisyjnymi wynosi min. 3 bity.

5.4.8 Warstwa zastosowań

Obecnie istnieje sześć następujących kompletnych specyfikacji warstwy zastosowań:

- **CAL** (*CAN Application Layer*) - jest to specyfikacja międzynarodowego stowarzyszenia użytkowników i producentów CAN (CiA - *CAN in Automation*), która definiuje standardowy interfejs aplikacyjny umożliwiający rozwój oprogramowania rozproszonych systemów czasu rzeczywistego.
- **CAN Kingdom** – nie jest kompatybilna z modelem referencyjnym ISO/OSI i jest stosowana w krajach skandynawskich w pojazdach drogowych i systemach przemysłowych.
- **CANopen Profile** – otwarty system stosowany w przemyśle, używa podzbioru usług zdefiniowanych w CAL, podobnych w zastosowaniu do *DeviceNet*.
- **DeviceNet** – amerykański otwarty system sieci przemysłowych, promowany przez firmę Allen Bradley, do integracji prostych i złożonych sieciowych urządzeń.
- **OSEK/VDX** – warstwa aplikacyjna stosowana w europejskiej motoryzacji, koordynowana przez Uniwersytet w Karlsruhe.
- **SDS** (*Smart Distributed System*) – protokół używany przez proste urządzenia wejścia – wyjścia, promowany przez firmę *Honeywell*.

5.4.8.1 CAL

Warstwa CAL specyfikuje komunikację na magistrali CAN. W ramach CAL rozwijane są cztery następujące specyfikacje:

CMS - *CAN Message Specification*,

DBT - *Distributor*,

NMT - *Network Management*,

LMT - *Layer Management*.

W specyfikacji CMS zdefiniowane są trzy różne typy obiektów komunikacyjnych: zmienne, domeny i zdarzenia. CMS jest oparta na modelu *client-server* i jest wzorowana na MMS (ang. *Manufacturing Message Specification*). Każdy z obiektów komunikacyjnych może transmitować maksymalnie 8 bajtów. Transmisja zmiennych i domen inicjowana jest przez klienta, a przesyłanie zdarzeń przez serwera.

Specyfikacja DBT jest odpowiedzialna za dystrybucję identyfikatorów w sieciach CAN. Pozwala ona na współdziałanie i współpracę modułów od różnych producentów, przy czym każdy z przyznawanych identyfikatorów musi być unikatowy i rozpoznawalny w całej sieci. DBT jest oparta na modelu *master – slave*, przy czym jedna stacja jest *master*'em, a inne węzły pracują jako *slave*. Każdy z węzłów może być skonfigurowany tylko raz (w czasie instalacji urządzenia) lub każdorazowo po włączeniu zasilania sieci.

Usługi zarządzania siecią NMT są podobne do usług DBT, a ich zadaniem jest konfigurowanie sieci, dodawanie / odejmowanie węzłów sieci oraz sterowanie w przypadkach występowania błędów.

Usługi LMT są podobne do usług zdefiniowanych w DBT i NMT. Korzystają one z jednej stacji *master* i ustawiają wszystkie parametry warstwy fizycznej (numer identyfikacyjny, prędkość transmisji itp.) w innych węzłach. Podobnie jak usługi DBT i NMT są one niewidoczne dla programów użytkownika. Typowo konfiguracja sieci jest dokonywana tylko w czasie włączenia zasilania sieci.

5.5 Standard ASI

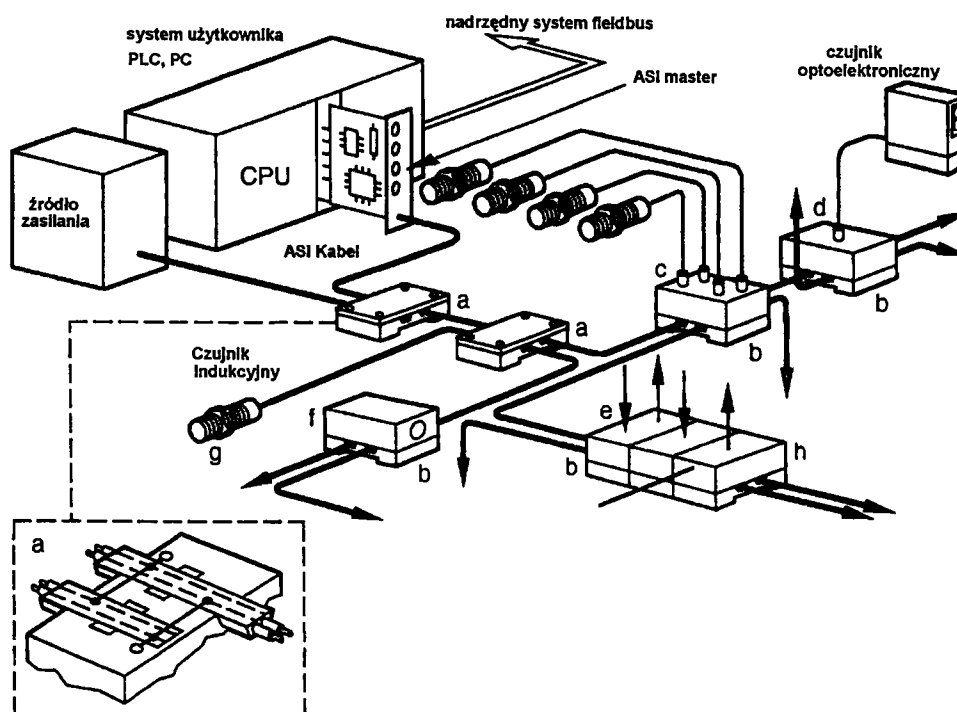
5.5.1 Wprowadzenie

System komunikacji ASI (ang. *Actuator Sensor Interface*) został zaprojektowany do zastosowań w najniższych warstwach modelu komunikacyjnego. Dzięki uwzględnieniu specyficznych warunków występujących w obszarze czujników i elementów wykonawczych, takich jak np. krótkie, ale często przesyłane telegramy, opracowane rozwiązanie jest bardzo efektywne. Do istotnych zalet systemu ASI należy zaliczyć możliwość spełnienia wymagań pracy w czasie rzeczywistym.

O szerokim zastosowaniu tego systemu zdecydowała jednak, obok parametrów technicznych i standaryzacji, także niska cena rozwiązania oraz możliwość wykorzystania ASI do konwencjonalnych czujników i elementów wykonawczych.

5.5.2 Modułowa koncepcja systemu

Na rysunku 5.5.1 przedstawiono przykład kompletnego rozwiązania systemu ASI. Elementem łączącym poszczególne elementy jest nieekranowany dwużyłowy kabel, który stanowi medium transmisyjne zarówno dla danych, jak i energii zasilającej. Sterownik mikroprocesorowy, za pośrednictwem systemu transmisji ASI, komunikuje się z czujnikami indukcyjnymi (g) oraz czujnikiem optycznym (d).



Rys.5.5.1 Modułowa koncepcja systemu ASI

W celu zrozumienia modułowej koncepcji budowy systemu ASI na wstępie należy zwrócić uwagę na jego moduły łączące (a, b na rys.5.5.1). W przypadku stosowania płaskiego kabla ASI moduł łączący uzyskuje połączenia elektryczne za pośrednictwem kontaktów szpilkowych, przebijając izolację kabla. Metoda montażu na zatrask, bez konieczności odizolowywania, jest bardzo szybka i pozwala na elastyczne przekonfigurowywanie systemu. W przypadku stosowania tradycyjnego kabla okrągłego należy natomiast wykorzystywać moduły łączące ze stykami dokręcanymi. Oba typy modułów są zamykane, dzięki normalizacji mechaniczno-elektrycznej (niem. *elektromechanische Schnittstelle ASI-EMS*), jednakowym typem modułu użytkownika. Ponieważ każdy z modułów łączących może zawierać dwa kable, dlatego też jest on wykorzystywany jako: skrzyżowanie połączeń, rozwidlenie lub element typu T. Moduły użytkownika (c, d, e, f) wykonywane z zachowaniem normalizacji ASI-EMS, oferują szeroką gamę usług (wejścia, wyjścia przekładnikowe i elektroniczne, moduły dwukierunkowe). Podstawowy podział modułów dokonywany jest na: zewnętrzne i zintegrowane ze *slave*'m, albo pasywne (g) i aktywne (c, d) - gdzie dołączany element nie musi mieć własnej części elektronicznej, co umożliwia dołączanie konwencjonalnych czujników i elementów wykonawczych.

5.5.3 Otwartość systemu

ASI jest systemem otwartym, dzięki czemu użytkownicy nie muszą obawiać się ograniczeń związanych z "przywiązaniem do producenta". Rozwiązanie takie pozwala na elastyczny wybór komponentów tego standardu od różnych producentów.

Najistotniejsze zalety otwartego systemu komunikacyjnego to:

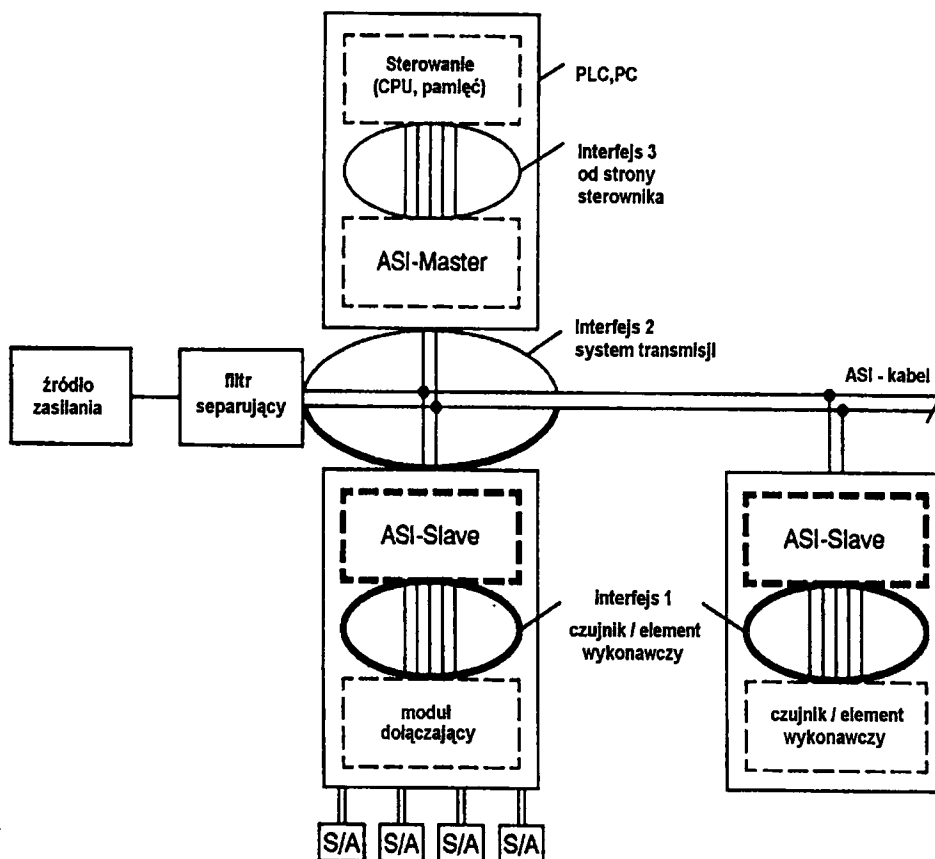
- dostępność specyfikacji standardu,
- dostępność układów scalonych do budowy interfejsu,
- możliwość dołączania czujników i elementów wykonawczych różnych producentów, poprzez znormalizowany interfejs,
- łatwość przebudowy i rozszerzania,
- możliwość jednoczesnego stosowania rozwiązań typowych i mieszanych,
- certyfikacja zapewniająca zgodność produktów różnych producentów oraz
- wspieranie standardu przez firmy dominujące.

5.5.4 System transmisji ASI

System ASI składa się (rys.5.5.2) z trzech modułów sprzętowych, które są wzajemnie połączone przez interfejsy i umożliwiają one dostęp do otoczenia.

Moduły te to:

- system transmisji,
- urządzenia typu *ASI - slave*,
- urządzenie typu *ASI - master*.



Rys.5.5.2 Moduły systemu ASI z zaznaczonymi interfejsami [1]

System transmisji stanowią dwużyłowe kable elektryczne, łączące w różnych topologiach urządzenia *master* i *slave*, jak również źródła zasilające oraz filtry. Przez dwużyłowe okablowanie systemu przekazywane są zarówno dane oraz energia zasilająca.

Urządzenia typu *slave*, poprzez interfejs 1 (rys.5.5.2) od strony sensor-element wykonawczy, realizują połączenia do czujnika i elementu wykonawczego. Są one tak zaprojektowane, że istnieje możliwość ich integracji (scalenia) z elementami końcowymi.

Urządzenie typu *master*, poprzez interfejs 3 od strony układu sterowania dołącza system nadrzędny, który może być programowalnym sterownikiem (PLC), komputerem (PC, VME) lub interfejsem do nadrzędnego systemu komunikacyjnego. Moduł ten steruje dostępem do sieci, kierując ruchem danych jak również przejmuje on funkcje zarządzające.

Standard ASI [3] definiuje interfejs 2, stanowiący magistralę transmisyjną oraz precyzuje wszelkie zdarzenia zachodzące na magistrali podczas przesyłania danych i energii. Standard ten definiuje jedynie koncepcyjnie interfejs 1, pozostawiając konkretną jego realizację producentowi. Dla najczęściej spotykanych urządzeń typu *slave* wprowadzono opisy formalne (*profile*) po to, aby zapewnić funkcjonalną zgodność produktów różnych producentów.

Z uwagi na różne struktury, jakimi zarządza urządzenie nadrzędne typu *master*, interfejs 3 został również określony jedynie w sposób koncepcyjny. Konkretna realizacja uzależniona jest od implementacji.

Pozostałe elementy systemu ASI znajdują zastosowanie jedynie w szczególnych przypadkach.

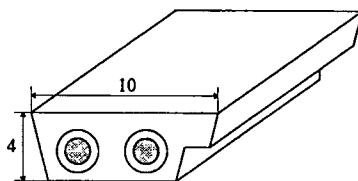
Repeater umożliwia zwiększenie maksymalnej długości sieci ze 100 m do 300 m.

Urządzenie serwisowe oraz **adresujące** są szczególnie pomocne podczas uruchamiania systemu (np.: programowanie adresów dla poszczególnych urządzeń) i usuwania nieprawidłowości występujących podczas działania systemu.

5.5.4.1 Okablowanie ASI

Jako medium transmisyjne dla systemu ASI zostały przyjęte dwa alternatywne rozwiązania:

- dla urządzeń wprowadzających duże obciążenie dwużyłowy nieekranowany elastyczny kabel okrągły - wg normy *CENELEC* lub *DIN VDE 0281* kabel typu: *H05VV-F2X1,5*,
- jako typowy dla systemu ASI - dwużyłowy, nieekranowany kabel płaski o podobnych właściwościach elektrycznych, ale z dodatkowymi zaletami instalacyjnymi takimi jak: łatwość montażu oraz brak możliwości zamiany polaryzacji ze względu na asymetryczny przekrój kabla (rys.5.5.3).



Rys.5.5.3 Płaski kabel ASI

Zaleca się stosowanie kabla o określonych barwach, zależnie od zastosowań [2]:

- **żółty:** typowy kabel stosowany dla transmisji danych oraz energii zasilającej,
- **czarny:** dodatkowe zasilanie 24V DC, dla modułów o dużej obciążalności,
- **czerwony:** dodatkowe zasilanie 230V AC.

Typowe media transmisyjne stosowane dla magistral przemysłowych (ang. *Fieldbus*), takie jak: skrętka, kabel współosiowy czy światłowód, nie znalazły w systemie ASI zastosowania ze względu na brak możliwości jednoczesnego przesyłania danych i energii, łamliwość, trudność montażu bądź wysoką cenę.

Przy wyborze medium transmisyjnego szczególnie istotną rolę odgrywają dwa parametry:

- rezystancja - dla przekazywania energii zasilającej,
- parametry częstotliwościowe - dla przekazywania danych.

Sieć połączeniowa systemu ASI powinna zapewnić przepływ prądu o natężeniu 2 A przy dodatkowych założeniach, że:

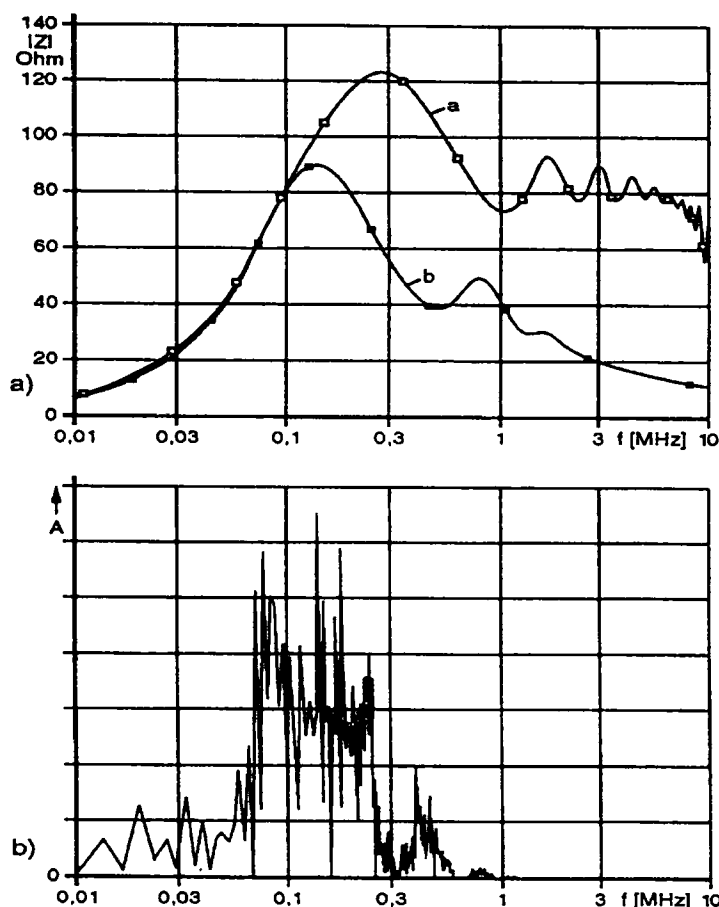
- całkowita długość połączeń nie przekracza 100 m,
- sieć łączy maksymalnie 31 urządzeń typu *slave* rozmieszczonych równomiernie na całej jej długości,
- poszczególne urządzenia wprowadzają jednakowe obciążenia 65 mA (co odpowiada całkowitemu obciążeniu 2 A).

Określony został również maksymalny spadek napięcia między źródłem zasilania systemu a ostatnim odbiornikiem (tab.5.5.1).

Tab.5.5.1 Zależność parametrów kabla ASI od jego przekroju

Przekrój przewodnika [mm ²]	0,75	1,5	2,5
Rezystancja [mΩ/m]	52,0	26,6	16,0
Maksymalny spadek napięcia [V]	5,4	2,7	1,6

Ponieważ urządzenia wykonawcze, które są dołączone do interfejsu 1 (rys.5.5.2), mają ściśle określone napięcie zasilania 24V (+10% / -15%), to spadek napięcia w sieci ASI nie może przekroczyć 3V. Wymagania takie zdecydowały o wyborze wartości przekroju kabla łączącego 1,5 mm², co pozwala na zapewnienie większości wymagań stawianych przez urządzenia wykonawcze.



Rys.5.5.4 Parametry transmisyjne kabla ASI [1]: a) przebieg impedancji: (a) - z filtrem, bez węzłów *slave*, (b) - z filtrem i 30 węzłami typu *slave*; b) widmo częstotliwościowe typowego rozkazu *master*'a ASI

Jednakże w przypadkach wątpliwych należy przeprowadzić analizę spadków napięć i rozptyłów prądów w sieci. Techniczne względy pozwalają na stosowanie innych przekrojów przewodnika z zakresu 0,75-2,5 mm², przy czym mniejsze przekroje znajdują zastosowanie przy łączeniu sensorów o małym poborze prądu, przy zapewnieniu odpowiedniego napięcia zasilającego.

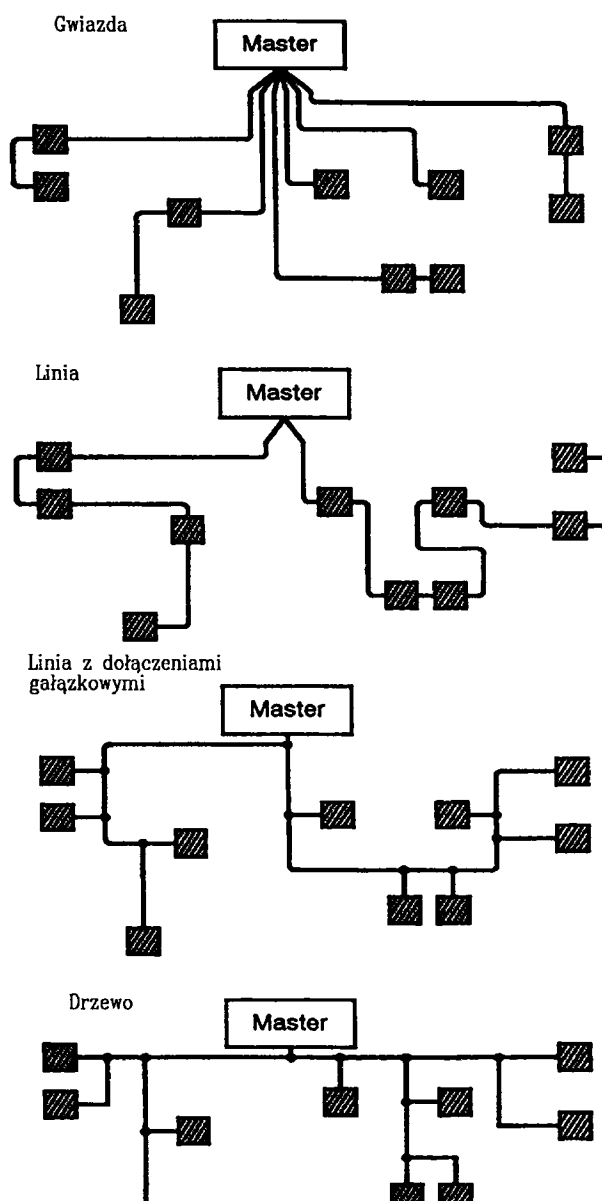
Oczywiście w przypadkach ekstremalnych, np. gdy wszystkie 31 urządzeń typu *slave* dołączone są na końcu 100 m magistrali, zachodzi konieczność stosowania przekrojów $2,5 \text{ mm}^2$.

Rysunek 5.30 przedstawia przebieg zmiany impedancji kabla ASI w funkcji częstotliwości. Dla częstotliwości 0,167 MHz wynosi ona $80\text{--}120 \Omega$, natomiast poza nią impedancja rośnie. Przebieg taki jest szczególnie korzystny dla eliminacji zakłóceń asymetrycznych.

W przypadku systemów transmisji narażonych na duże zakłócenia elektromagnetyczne zaleca się stosowanie kabli ekranowanych (np. *(N)YMHC-02X1,5*). Szczególnie istotne z uwagi na efektywność ekranowania jest prawidłowe „umasowienie” opłotu. Powinno ono być wykonane tuż przy źródle zasilania i w żadnym razie nie do jednej z linii transmisyjnych (ASI+, ASI-).

5.5.4.2 Topologia systemu ASI

Wybór topologii systemu ASI jest bardzo elastyczny (rys.5.5.5).



Rys.5.5.5 Przykłady topologii systemu ASI [1]

Umożliwia to swobodne dopasowanie struktury połączeń do wymagań obiektu. Może ona zatem stanowić gwiazdę, magistralę, drzewo, jak również być rozwiązaniem mieszanym. Odbiorniki mogą być pojedynczo dołączane do magistrali za pośrednictwem gałęzi lub tworzyć gwiazdźście dołączane grupy. Magistrala systemu nie wymaga zamknięcia impedancyjnego - terminatorów.

Głównym ograniczeniem wyboru konfiguracji systemu jest łączna długość linii, która nie może przekroczyć 100 m. Do długości tej należy zaliczyć również dołączenia poszczególnych gałęzi do magistrali głównej. W tych przypadkach gdy wymagana jest większa rozpiętość systemu należy stosować urządzenia regenerujące sygnał *repeater*'y, przy zastosowaniu których maksymalna długość linii zwiększa się do 300 m.

5.5.4.3 Metoda modulacji

Przy wyborze odpowiedniej techniki modulacji musiały zostać uwzględnione liczne wymagania:

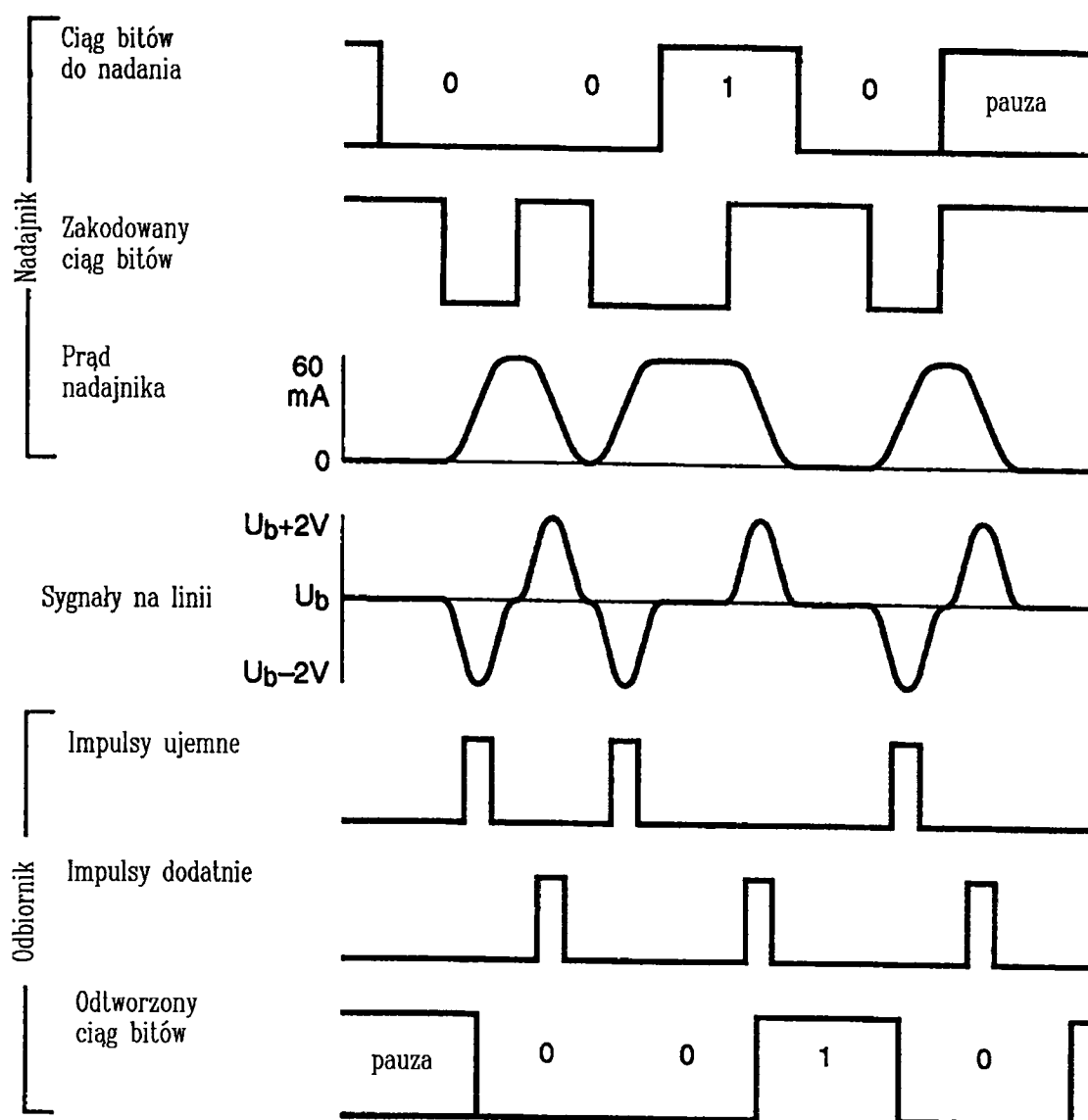
- **sygnał informacji** - ponieważ czujniki i elementy wykonawcze wymagają zasilania, które ma być dostarczane równocześnie tym samym medium transmisyjnym, to nie może ono zawierać składowej stałej,
- **nadajniki** urządzeń typu *slave* i master muszą być tanie w realizacji - zatem powinny generować proste sygnały,
- **zakres częstotliwości** - zgodnie z przedstawionym przebiegiem na rysunku 5.5.2, a przyjęty zakres częstotliwości powinien być relatywnie wąski,
- **emisyjność** pracującego systemu powinna być możliwie niska.

Ponieważ żadna ze znanych metod modulacji nie zapewniała realizacji tych wymagań, dlatego też zdecydowano się na opracowanie nowego rozwiązania dla systemu ASI.

Alternatywna modulacja fazy (ang. *Alternating Pulse Modulation APM*) jako szeregową transmisję w paśmie podstawowym została zobrazowana na rys.5.5.2.

Ciąg bitów do wysłania zostaje najpierw przekształcony w zakodowany ciąg, który przy każdej zmianie sygnału wprowadza zmianę fazy. Z sygnału tego uzyskiwany jest sygnał prądowy, który dzięki indukcyjności filtru separującego zasilacza, generuje różnicowy sygnał napięciowy. Każdy przyrost prądu powoduje zatem powstanie ujemnego impulsu napięciowego, natomiast jego spadek powstanie impulsu dodatniego. W ten sposób możliwe jest generowanie impulsów przez urządzenia typu *slave* o amplitudach przekraczających ich napięcia zasilania. Dzięki elementom indukcyjnym w filtrze separującym nie ma potrzeby ich stosowania w nadajnikach. Zmniejsza to ich gabaryty i obniża koszty.

Po stronie odbiorczej sygnały napięciowe zostają zdekodowane i przetworzone w ciąg bitów.



Rys.5.5.6 Alternatywna modulacja fazy [1]

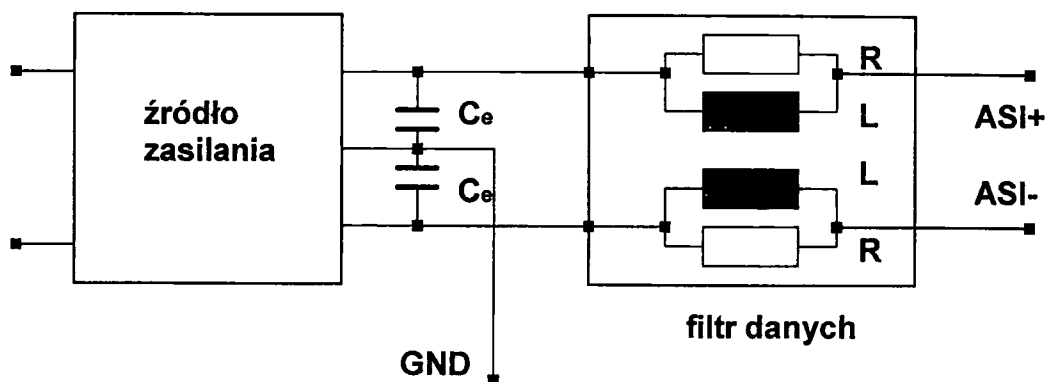
Jeśli sygnały napięciowe zbliżone są do przebiegu funkcji $\sin^2$ to jest spełniony warunek ograniczenia pasma jak również niskiej emisyjności. Taką postać sygnału uzyskuje się dzięki odpowiedniemu kształtowaniu impulsów przez całkowanie. Widmo częstotliwościowe typowego ciągu impulsów przedstawiono na rys.5.5.2 b, który w zestawieniu z rys.5.5.2 a, określającym charakterystykę częstotliwościową kabla ASI, potwierdza optymalne dopasowanie obu charakterystyk.

Opracowana metoda modulacji, przy zachowaniu wymagań topologii systemu, pozwala na uzyskanie czasów przesłania pojedynczego bitu na poziomie 6 μs .

5.5.6.4.4 Zasilanie energii i filtr separujący

Rysunek 5.5.7 przedstawia schematycznie źródło zasilania systemu ASI wraz z filtrem. Źródło zasilania systemu zapewnia napięcie w przedziale 29.5 V .. 31.6 V DC i obciążalności 0 A .. 2,2 A.

Źródło to zawiera również zabezpieczenie przeciwzwarciowe oraz przeciążeniowe. Wszystkie szczegóły dotyczące zasilania w systemie ASI są określone normą [3].



Rys.5.5.5.7 Źródło zasilania i filtr separujący w systemie ASI

Filtr separujący składa się z dwóch elementów indukcyjnych $50\ \mu\text{H}$ wraz z równoległymi rezystancjami $39\ \Omega$. Na elementach indukcyjnych następuje całkowanie przebiegu prądowego wymuszonego przez nadajniki, a wynikiem tego jest powstawanie przebiegu napięciowego. Elementy indukcyjne zapobiegają jednocześnie zwarceniu dla składowej zmiennej w systemie transmisji.

W celu optymalnego zabezpieczenia systemu przed zakłóceniami sumacyjnymi, wymagana jest symetryczna budowa układu połączeń. W tym celu wprowadzono symetryczne elementy pojemnościowe C_e , których wartości są tak dobrane, że stanowią pojemność dominującą w obwodzie (rys.5.5.7).

Gdy całkowite obciążenie wprowadzane przez węzły sieci jest większe od $2\ \text{A}$, to należy zastosować dodatkowe źródła zasilania dla urządzeń typu *slave* lub *repeater*'y z dodatkowymi źródłami.

5.5.4.5 Metody dostępu do magistrali

Ponieważ system ASI powinien umożliwić zastąpienie połączenia typu gwiazda (tradycyjna wiązka przewodów), to należało zaproponować taką metodę dostępu do medium transmisyjnego, przy której możliwe byłoby zachowanie logicznej struktury systemu i zapewnienie spełnienia wymagań czasu rzeczywistego. Wymagania takie spełnia metoda dostępu **master-slave** z cyklicznym odpytywaniem (ang. *polling*). Urządzenie zarządzające ruchem w sieci, czyli *master*, wysyła telegramy z zapytaniami bądź rozkazami dla wybranych urządzeń końcowych typu *slave*. W określonym czasie od otrzymania telegramu urządzenie odbiorcze musi nadać telegram potwierdzający (rys.5.5.8).

Z punktu widzenia systemu transmisji w jednej chwili może się komunikować tylko jeden *master* z jednym *slave*'m.

Takie alternatywne metody dostępu do magistrali jak:

- dostęp wielu urządzeń typu master z przekazywaniem znacznika (ang. *token passing*),

- z detekcją kolizji (*CSMA/CD*),
- dostęp priorytetowy (*CSMA/CA*) oraz
- z pierścieniowym przekazywaniem znacznika (*token ring*)

nie znalazły zastosowania w systemie ASI ze względu na:

- dużą złożoność konfiguracji i przydział praw,
- brak możliwości zapewnienia wymagań czasu rzeczywistego,
- długość telegramów.

Wady te nie pozwoliły na zastosowanie powyższych metod dostępu dla magistrali przeznaczonej dla zastosowań w warstwie czujników i elementów wykonawczych, gdzie informacje są wprawdzie krótkie, ale są przekazywane z dużą częstotliwością.

Wybrana metoda dostępu *master-slave* dla standardu ASI umożliwia budowę bardzo prostych (tanich) urządzeń odbiorczych *slave*, nie ograniczając przy tym elastyczności rozwiązania. Możliwe jest bowiem wysłanie dodatkowego telegramu z rozkazem w dowolnym czasie, bez istotnego zachwiania cyklu pracy systemu. W metodzie odpytywań cykl pracy zależy od liczby dołączonych do sieci urządzeń. Przy cyklicznym odpytywaniu sześciu odbiorników długość cyklu wynosi ok. 1 ms. Przy maksymalnej rozbudowie systemu (31 węzłów) czas ten nie przekracza zatem 5 ms. Metoda ta umożliwia również "wyjątkowe" odpytanie urządzenia w przypadku wystąpienia błędu transmisji, bez konieczności odczekania pełnego cyklu.

Całkowita szybkość transmisji w systemie ASI, łącznie z niezbędnymi czasami przerw, wynosi 0.167 Mbit/s. Dane netto przesyłane są z szybkością 53.3 kbit/s, co ustala efektywność protokołu na poziomie 32 %. W porównaniu z innymi standardami magistral przemysłowych efektywność protokołu ASI jest wysoka, co wynika z korzystnego stosunku przesyłanych danych do informacji dodatkowej. Do zalet tego rozwiązania należy również wysokie zabezpieczenie przed zakłóceniami elektromagnetycznymi (p. „Działanie systemu w przypadku zakłóceń i błędów”).

5.5.4.6 Budowa telegramów

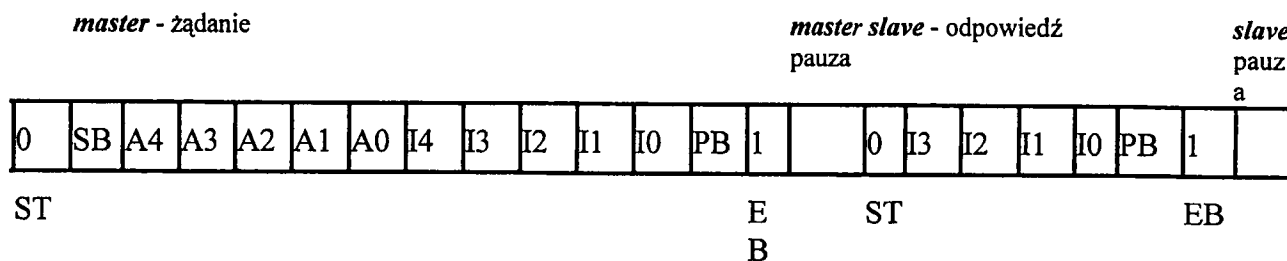
Telegramy systemu ASI składają się z następujących bloków:

- adres oraz zapytanie-rozkaz *master'a*,
- przerwa po telegramie *master'a*,
- odpowiedź-potwierdzenie *slave'a*,
- przerwa po telegramie *slave'a*.

Wszystkie telegramy pochodzące od *master'a* mają długość 14 bitów, od *slave'a* długość 7 bitów, natomiast czas przesłania jednego bitu został ustalony na 6 μ s.

Przerwa po telegramie *master'a* powinna być równa czasowi 3 bitów, nie dłuższa jednak niż 10 bitów. W przypadku gdy *slave* jest synchronizowany, to może on rozpocząć nadawanie odpowiedzi już po odczekaniu 3 bitów, natomiast w przypadku pracy asynchronicznej wymagane jest przedłużenie oczekiwania o dodatkowe 2 bity, co wynika z konieczności oczekiwania na

dodatkowe informacje przed zaakceptowaniem telegramu. Przerwa po telegramie *slave*'a została ustalona na długość 1 bita.



Rys.5.5.8 Struktura komunikatów w systemie ASI

Telegram *master*'a składa się z następujących pól:

ST	bit startu	sygnalizuje początek telegramu <i>master</i> 'a
		0: aktywny start
		1: niedozwolony start
SB	bit sterujący	wskazuje na typ telegramu
		0: dane / parametry albo adres
		1: rozkaz
A0..A4	adres	adres wywołanego <i>slave</i> 'a (5 bitów)
I0 .. I4	dane	5 bitów danych, które zawierają informację zależną od zapytania
PB	bit parzystości	dopełnienie do parzystej liczby jedynek w telegramie
EB	bit stopu	zaznacza koniec telegramu
		0: niedozwolony
		1: aktywny koniec

Telegram *slave*'a składa się z następujących pól:

ST	bit startu	sygnalizuje początek telegramu <i>master</i> 'a
		0: aktywny start
		1: niedozwolony start
I0 .. I3	dane	4 bity danych, które zawierają informację zależną od zapytania
PB	bit parzystości	dopełnienie do parzystej liczby jedynek w telegramie
EB	bit stopu	zaznacza koniec telegramu
		0: niedozwolony
		1: aktywny koniec

5.5.4.7 Rozkazy

Zostało zdefiniowanych jedynie dziewięć różnych rodzajów telegramów, z czego:

- 2 - dla przekazywania, wymiany danych i parametrów,
- 2 - do konfiguracji (ustalania, zamiany adresów),
- 5 - do identyfikacji odbiorników (np. odpytanie statusu).

data_exchange

		5 bitowy adres					5 bitowe pole informacyjne						
0	0	A	A	A	A	A	0	D	D	D	D	P	1
4	3	2	1	0	I	I	I	I	I	0	B		
ST	S						I4	I3	I2	I1	I0	E	B
	B												

write_parameters

0	0	A	A	A	A	A	1	D	D	D	D	P	1
4	3	2	1	0	I	I	I	I	I	0	B		
ST	S						I4	I3	I2	I1	I0	E	B
	B												

assign_address

0	0	0	0	0	0	0	A	A	A	A	A	P	1
4	3	2	1	0	I	I	I	I	I	I	0	B	
ST	S	A	A	A	A	A	I4	I3	I2	I1	I0	E	B
	B	4	3	2	1	0							

reset_ASI_slave

0	1	A	A	A	A	A	1	1	1	1	0	P	1
4	3	2	1	0	I	I	I	I	I	I	0	B	
ST	S						I4	I3	I2	I1	I0	E	B
	B												

read_IO_configuration

0	1	A	A	A	A	A	1	0	0	0	0	P	1
4	3	2	1	0	I	I	I	I	I	I	0	B	
ST	S						I4	I3	I2	I1	I0	E	B
	B												

read_ID_code

0	1	A	A	A	A	A	1	0	0	0	1	P	1
4	3	2	1	0	I	I	I	I	I	I	1	B	
ST	S						I4	I3	I2	I1	I0	E	B
	B												

read_status

0	1	A	A	A	A	A	1	1	1	1	0	P	1
4	3	2	1	0	I	I	I	I	I	I	0	B	
ST	S						I4	I3	I2	I1	I0	E	B
	B												

read_and_reset status

0	1	A	A	A	A	A	1	1	1	1	1	P	1
4	3	2	1	0	I	I	I	I	I	I	1	B	
ST	S						I4	I3	I2	I1	I0	E	B
	B												

Rys.5.5.9 Zestawienie rozkazów *master*'a w systemie ASI

Inne kombinacje kodów są niedozwolone i zarezerwowane dla dalszych rozszerzeń systemu ASI.

5.5.5 Wymiana danych (*data exchange*)

Najistotniejszym i najczęściej występującym rozkazem w sieci ASI jest **zapytanie o dane**. Rozkaz ten powoduje przekazanie maski bitowej do interfejsu 1, zaadresowanego *slave*'a lub odczytania stanu tego interfejsu i przekazania go do *master*'a. Działanie tego rozkazu zależy od konfiguracji portu, która może być następująca:

- jako wyjścia do procesu, które mogą zostać ustawione i wyzerowane,
- jako wejścia z procesu, których stan może być odczytany,
- jako mieszane wejścia / wyjścia lub kanały dwukierunkowe. Konfiguracja taka na stałe zapisana jest w 4 bitowym rejestrze konfiguracji (*I/O config*) *slave*'a.

Jeżeli porty skonfigurowane są jako wyjścia, to ostatnio przesłana do nich wartość zostaje zapamiętana (zatrzaśnięta) do czasu przyjęcia nowej lub „zresetowania”.

Po rozkazie *reset*, który może zostać wywołany również włączeniem napięcia zasilania systemu, *slave* przesyła stan swoich wejść dopiero po otrzymaniu pierwszego poprawnego zapytania.

5.5.5.1 Zapis parametrów (*write parameters*)

Rozkaz zapisania parametrów umożliwia zdalną konfigurację parametrów interfejsu 1. Rozkaz ten umożliwia rozszerzenie funkcji *slave*'a, np. przez przełączanie aktywnego czujnika, wyzwalanie zadań czasowych. Ostatnio przekazany parametr do węzła zostaje zatrzaśnięty do czasu nadejścia nowego, bądź „zresetowania” systemu.

Po sygnale *reset* parametry interfejsu 1 są ustawiane wg wartości zapisanych jako domyślne. Po pierwszym poprawnym rozkazie ustawienia parametrów interfejs jest przekonfigurowany.

5.5.5.2 Przydzielenie adresu (*assign address*)

Rozkaz przydzielenia adresu pozwala *master*'owi na nadanie nowego permanentnego adresu urządzeniu o adresie *00H*.

Slave potwierdza prawidłowe przyjęcie polecenia, odpowiadając komunikatem *6H* i tym samym rozpoczyna się proces przeprogramowywania pamięci EEPROM, który może trwać do 15 ms. Proces ten nie może zostać przerwany zanikiem napięcia zasilającego lub rozkazem *reset*.

Polecenie przydziału adresu jest wykorzystywane przez urządzenie adresujące w celu początkowego zaadresowania poszczególnych węzłów sieci podczas konfiguracji wstępnej. Dla *master*'a komenda ta umożliwia „wymianę” uszkodzonego *slave*'a.

5.5.5.3 Reset (*reset ASI slave*)

Po rozkazie *reset_ASI_slave* zostaje przywrócona początkowa konfiguracja *slave*'a. Rozkaz ten ma podobne działanie do *reset* po włączeniu napięcia zasilającego lub też rozkazu *reset* dla wejść interfejsu 1.

Telegram potwierdzający poprawne wykonanie polecenia zawiera odpowiedź *6H*, a czas wykonania rozkazu nie może być dłuższy jak 2 ms.

5.5.5.4 Kasowanie adresu operacyjnego (*delete operating address*)

Rozkaz kasowania adresu operacyjnego umożliwia czasowe kasowanie adresów operacyjnych urządzeń *slave* i jest stosowany jako poprzedzający dla rozkazu adresowania, gdyż ten jest dostępny jedynie dla urządzeń o adresie *00H*.

W przypadku konieczności przeprogramowania aktualnego adresu *slave*'a *15H* na *09H* należy zatem wydać następujący ciąg komend:

delete_operating_address

assign_adress 09H.

Po poprawnym wykonaniu rozkazu kasowania adresu operacyjnego, *slave* odpowiada komunikatem *6H* i od tego czasu jest dostępny pod nowym adresem *00H*. Kasowanie adresów za pomocą powyższej funkcji nie jest permanentne i możliwy jest powrót do adresu początkowego (stałego) za pomocą wywołania rozkazu *reset_ASI_slave*.

5.5.5.5 Odczytanie kodu identyfikacyjnego (*read ID code*)

Komenda ta umożliwia odczytanie identyfikatora urządzenia, co w połączeniu z rozkazem odczytu konfiguracji jednoznacznie identyfikuje *slave*'a.

Numer identyfikacyjny zostaje przydzielony i zapisany podczas produkcji i nie podlega on już późniejszej zmianie. Pozwala on na identyfikację przynależności do zdefiniowanego typu rozwiązań. Definicja taka ujęta jest w dokumentacji określonej jako profil (ang. *profile*). Wszystkie urządzenia dla których sporządzono specyfikację indywidualną, na podstawie umowy producenta z użytkownikiem, mają numer identyfikacyjny *FH*.

5.5.5.6 Odczytanie konfiguracji we/wy (*read I/O configuration*)

Pozwala ono na odczytanie aktualnej konfiguracji portów wejście / wyjście *slave*'a. W połączeniu z rozkazem odczytu kodu identyfikacyjnego *id_code_read* umożliwia to jednoznaczną identyfikację urządzeń końcowych.

Konfiguracja dotyczy interfejsu 1 i zdefiniowana jest w tab.5.5.2.

Tab.5.5.2 Możliwe konfiguracje portów WE/WY

Kod	D0	D1	D2	D3
0 Hex	IN	IN	IN	IN
1 Hex	IN	IN	IN	OUT
2 Hex	IN	IN	IN	I/O
3 Hex	IN	IN	OUT	OUT
4 Hex	IN	IN	I/O	I/O
5 Hex	IN	OUT	OUT	OUT
6 Hex	IN	I/O	I/O	I/O
7 Hex	I/O	I/O	I/O	I/O
8 Hex	OUT	OUT	OUT	OUT
9 Hex	OUT	OUT	OUT	IN
A Hex	OUT	OUT	OUT	I/O
B Hex	OUT	OUT	IN	IN
C Hex	OUT	OUT	I/O	I/O
D Hex	OUT	IN	IN	IN
E Hex	OUT	I/O	I/O	I/O
F Hex	TRI	TRI	TRI	TRI

Oznaczenia:

- IN: wejście od procesu,
 OUT: wyjście od procesu,
 I/O: port dwukierunkowy,
 TRI: brak możliwości konfiguracyjnej.

Stan TRI został przyjęty ze względu na przypadek braku możliwości konfiguracji w czasie *reset*, spowodowanego np. błędem odczytu danych ze *slave*.

Typowe, jednokierunkowe, skonfigurowanie portów dla czujników i elementów wykonawczych, zostało rozszerzone o możliwości transmisji dwukierunkowej. Np. czterobitowe pole danych, w rozkazie wymiany danych, może być traktowane jako podadres, co umożliwia odczytanie jednej z szesnastu 4 bitowych odpowiedzi.

Sposób ten umożliwia również przesyłanie kolejno, zakodowanej wartości analogowej, o długości 64 bitów.

5.5.5.7 Odczytanie statusu (*read status*)

Rozkaz ten umożliwia kontrolę stanu rejestru statusowego *slave'a*. Rejestr ten zawiera następujące flagi:

S0: zapamiętanie adresu

Flaga jest ustawiona w czasie przebiegu procesu programowania permanentnego nowego adresu. Proces może trwać do 15 ms i nie może zostać przerwany przez *reset* lub inny rozkaz przeprogramowania.

S1: błąd parzystości

Flaga jest ustawiona, jeśli od ostatniego rozkazu *reset* lub *read_and_reset_status* został rozpoznany błąd parzystości.

S2: błąd zakończenia

Flaga jest ustawiona, jeśli od ostatniego rozkazu *reset* lub *read_and_reset_status* zostało rozpoznane nieprawidłowe wykonanie komendy.

S3: błąd odczytu pamięci wewnętrznej.

Flaga jest ustawiona, jeżeli po rozkazie *reset* wystąpił błąd odczytu wewnętrznej konfiguracji (błąd czytania wewnętrznej pamięci).

Po rozkazie odczytu statusu zostaje przeprowadzone porównanie adresu operacyjnego ze stałym (bazowym). W przypadku niezgodności zostaje ustawiona flaga S0.

Informacje zawarte w rejestrze statusowym służą dla celów diagnostycznych.

5.5.5.8 Odczytanie i kasowanie statusu (*read and reset status*)

Działanie komendy rozszerzone zostało w stosunku do odczytu statusu o zerowanie flag rejestru statusowego.

5.5.6 Działanie systemu w przypadku zakłóceń i błędów

Szczególnie ważna w przypadku systemów komunikacyjnych o dużych szybkościach transmisji oraz wówczas, gdy stosuje się jako połączenia kabel nieekranowany, jest odporność na zakłócenia elektromagnetyczne.

Należy rozróżnić dwa przypadki:

- **emisję zakłóceń**, która nie powinna przekroczyć wartości określonych w normie EN 55011,
- **odporność na narażenia elektromagnetyczne**, która jest określona w normie IEC 801.

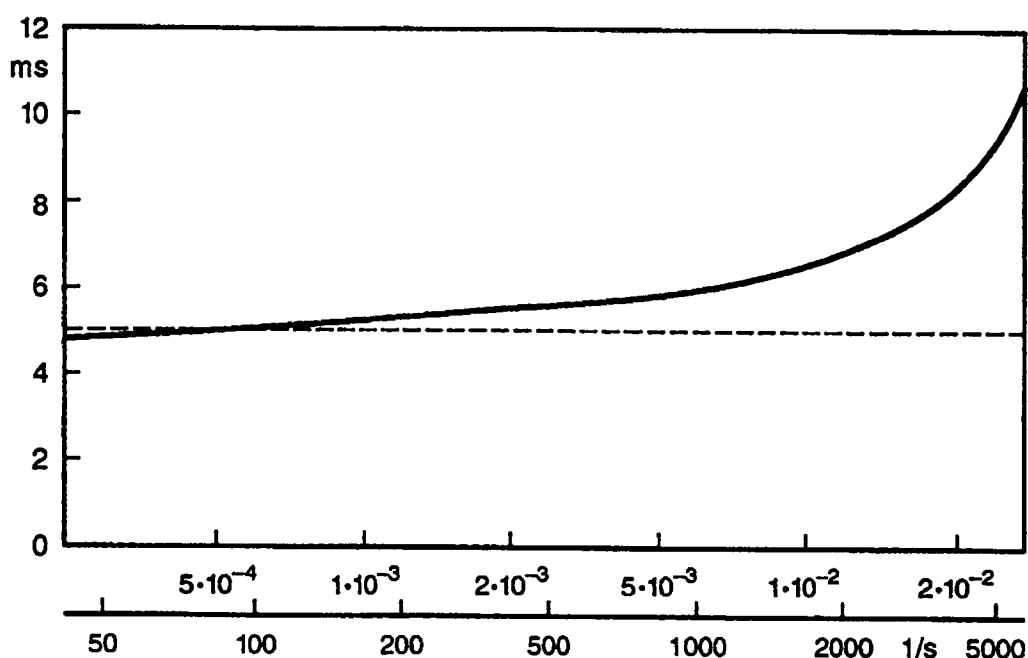
Pełne testy systemu ASI potwierdziły [1], że dzięki zastosowaniu impulsów $\sin^2$ emisja zakłóceń elektromagnetycznych leży poniżej wartości określonych w normie europejskiej dla tego standardu.

Odporność na narażenia elektromagnetyczne: wyładowania elektrostatyczne, pola elektromagnetyczne o zakresie częstotliwości 26 MHz do 1 GHz, jak również szybkie impulsy szpilkowe, określona jest na *poziomie 3*. Oznacza to, że w najbardziej niekorzystnym przypadku pojedyncze telegramy mogą ulec zakłóceniu. Jednakże dzięki mechanizmom kontroli parzystości błędy takie zostaną rozpoznane i nastąpi zażądanie powtórzenia telegramu.

Dzięki krótkim telegramom, pojedyncze powtórzenia nie wpływają zauważalnie na czas cyklu. W przypadku dużego natężenia powtórzeń cykl pracy ulega jednak zachwianiu.

Na podstawie rys.5.5.10 można zauważyć, że przy prawdopodobieństwie występowania zakłóceń na poziomie 5×10^{-4} , czas cyklu 5 ms jest zachowany. W przypadku krótkotrwałego wzrostu zakłóceń, powodowanego np. dołączeniem kolejnego urządzenia do pracującej sieci, system ASI pozostaje sprawny. Dopiero częstotliwość wystąpienia zakłóceń na poziomie wyższym

niż 5000 błędów/s nie pozwala zapewnić wymagań zachowania cyklu odpytań (czasu rzeczywistego).



Rys.5.5.10 Średni czas cyklu jako funkcja prawdopodobieństwa błędów i częstotliwości ich wystąpienia

Przy uszkodzeniu elektrycznym okablowania systemu ASI, np. złym kontakcie elektrycznym lub przerwie, urządzenia *slave* będące poza miejscem uszkodzenia są niewidoczne dla *master*'a, natomiast urządzenia leżące "bliżej" mogą się zwykle komunikować bez zakłóceń.

Zatem pomimo uszkodzenia linii transmisyjnej system ASI pozostaje częściowo czynny funkcjonalnie. W wielu przypadkach *master* może lokalizować miejsce wystąpienia niesprawności, efektywnie wspierając jej usunięcie. Podstawą takiego działania jest jednak jednakowy punkt dołączenia *master*'a, stanowiącego źródło informacji oraz urządzenia zasilającego. W przypadku wzajemnego odłączenia obu tych elementów następuje całkowite zaprzestanie działania systemu.

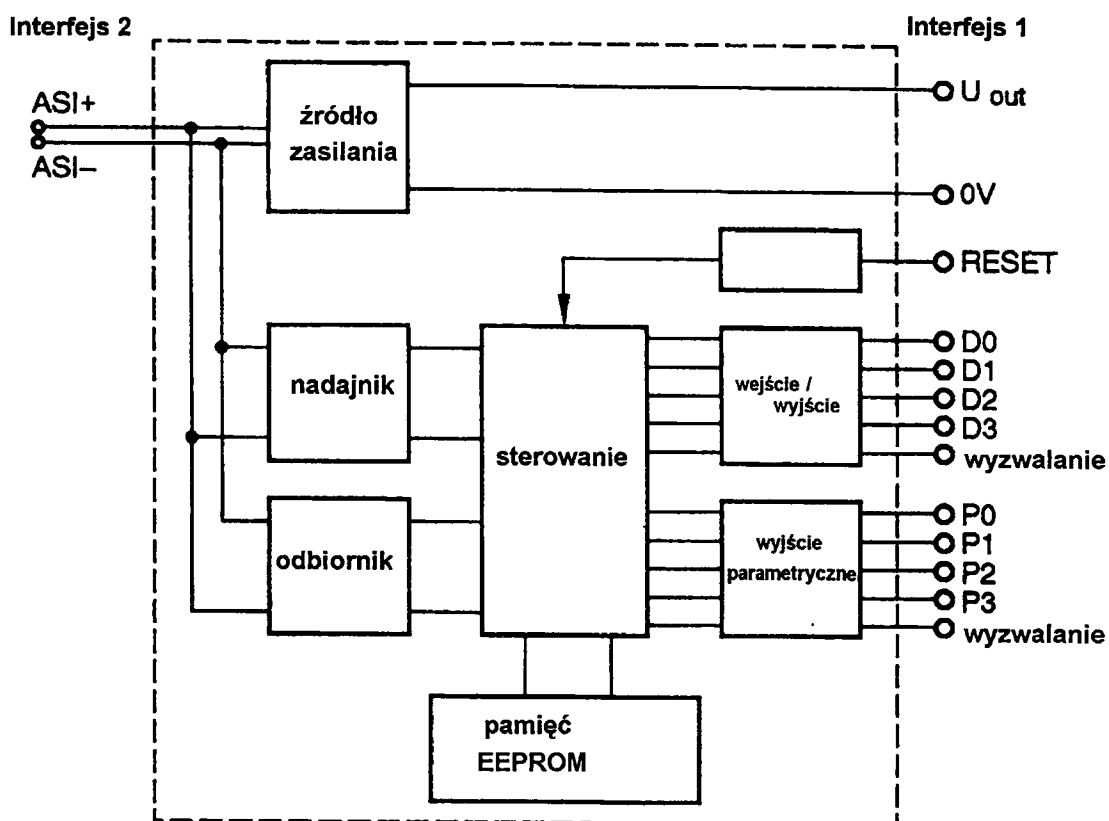
Sieć wyposażona w *repeater*'y jest odporna na zwarcia, a w przypadku ich zaistnienia wyłączony zostaje jedynie niesprawny jej segment.

5.5.6.1 ASI - *slave*

5.5.6.1.1 Opis schematu blokowego

Urządzenie *ASI - slave* stanowi połączenie między systemem transmisji ASI a interfejsem 1, do którego dołączone są czujniki i elementy wykonawcze. Z uwagi na zalety integracji *slave*'a z urządzeniami końcowymi zostało zaproponowane rozwiązanie w postaci zintegrowanego układu scalonego, co zapewniło małe gabaryty oraz niską cenę rozwiązania.

Na rysunku 5.5.11 przedstawiono główne bloki funkcjonalne zintegrowanego obwodu scalonego



Rys.5.5.11 Budowa blokowa ASI-IC

Z uwagi na jednoczesne przesyłanie energii oraz danych przez system transmisji ASI, na wejściu zastosowano układ rozdzielający.

Na wejściu **bloku odbiornika** układu ASI *slave* impulsy napięciowe zostają odfiltrowane, a następnie ich poziom podlega kwantyzacji. Po kontroli parzystości odebranego ciągu trafia on do rejestru odbiorczego. Z uwagi na polepszenie własności odbiornika zastosowano w nim adaptacyjny układ regulacji poziomu wyzwalania.

Blok nadajnika koduje informację zapisaną w rejestrze wyjściowym, wymuszając zmienny przebieg prądowy na liniach transmisyjnych. Dzięki zastosowaniu układów logiki (typu ang. *Jabber Inhibit*) zabezpieczono system transmisji ASI przed zakłóceniami w przypadku uszkodzenia nadajnika.

Blok sterujący dekoduje rozkazy *master'a*, sprawdza ich poprawność, formułuje odpowiedzi *slave'a* oraz zapisuje i odczytuje informacje do pamięci nie ulotnej EEPROM. Ma on również dostęp do wewnętrznych rejestrów:

- **rejestr adresowy:** długości 5 bitów, zawiera aktualny adres *slave'a* (adres operacyjny). W przypadku zgodności z adresem zawartym w odebranym rozkazie, nadawana jest odpowiedź do *master'a*. Po rozkazie *reset* rejestr adresowy zostaje załadowany adresem domyślnym zapisanym w pamięci EEPROM. Stan rejestru może zostać zmieniony rozkazami kasowania adresu operacyjnego (*delete operating address*) oraz przydzielenia adresu (*assign address*).

- **rejestry konfiguracji i kodu identyfikacyjnego:** mają długości 4 bitów, a dopuszczalne wartości dla rejestru konfiguracji definiuje tab.5.5.2. Stany rejestrów zapisane są w sposób permanentny przez producenta lub mogą być pobierane z pamięci EEPROM po rozkazie *reset*.
- **rejestr danych wyjściowych:** ma on długość 4 bitów i zawiera stan portu zachowany po ostatnim poprawnym rozkazie wymiany danych (*data exchange*). Używane w nim są jedynie pola zgodnie z konfiguracją we/wy (patrz rejestr konfiguracji). Po rozkazie *reset* zostaje wpisana do niego wartość domyślna FH.
- **rejestr parametryczny:** długości 4 bitów, zawiera informację o konfiguracji układu odczytanej po ostatnim rozkazie odczytu konfiguracji (*read I/O configuration*). Po rozkazie *reset* zostaje wpisana do niego wartość domyślna FH.
- **rejestr odbiorczy:** długości 12 bitów zawiera ostatnio odebrany telegram *master*'a, który jest przygotowany do dalszego przetworzenia przez blok sterujący.
- **rejestr nadawczy:** długości 5 bitów zawiera słowo przygotowane do nadania przez blok sterujący.
- **rejestr statusu:** zawiera 4 flagi określające aktualny stan układu (patrz opis rozkazu *odczytanie statusu*).
- **flaga synchronizacji:** po poprawnie odebranym telegramie od *master*'a, zdekodowaniu i wysłaniu odpowiedzi przez *slave*'a, zostaje ona ustawiona. W stanie synchronizacji przerwa po rozkazie *master*'a jest kontrolowana jedynie przez długość jednego bita, co pozwala na rozpoczęcie nadawania odpowiedzi już po 3 bitach.
- **flaga blokowania wymiany danych:** zostaje ustawiona po rozkazie *reset*, a jej zwolnienie następuje po poprawnym odebraniu telegramu odczytu parametrów. Nie pozwala ona na wystawienie nieaktualnego stanu portu.

5.5.6.1.2 Bezpieczeństwo danych i zachowanie systemu w przypadku wystąpienia błędów

Bezpieczeństwo danych systemu ASI opiera się na innej metodzie niż jest stosowana w większości systemów komunikacyjnych.

Telegram *master*'a zawiera 11, a *slave*'a 4 bity danych do ochrony. Chcąc zastosować konwencjonalny sposób ochrony, wykorzystujący kod Hamming'a, np. z odstępem $HD=4$, należałoby powiększyć długość telegramów *master*'a o 7 bitów (+64%), a *slave*'a o 4 bity (+100%). Doprowadziłoby to jednak do istotnego spadku efektywności transmisji. Ponadto, przy dużej ilości bitów kontrolnych w stosunku do ilości bitów danych, efektywność takiej metody jest bardzo niska, a prawdopodobieństwo wystąpienia zakłóceń w bloku kontrolnym i danych jest porównywalne.

Ze względu na powyższe rozważania zdecydowano się na zastosowanie metody polegającej na kontroli wybranych elementów transmisji:

- błąd bitu startu,
- błąd zmiany znaku,
- błąd pauzy,

- błąd informacji,
- błąd parzystości,
- błąd końca telegramu.

Każdy z rozkazów *master*'a podlega powyższej kontroli i w przypadku nieprawidłowości zostaje wymuszone powtórzenie telegramu.

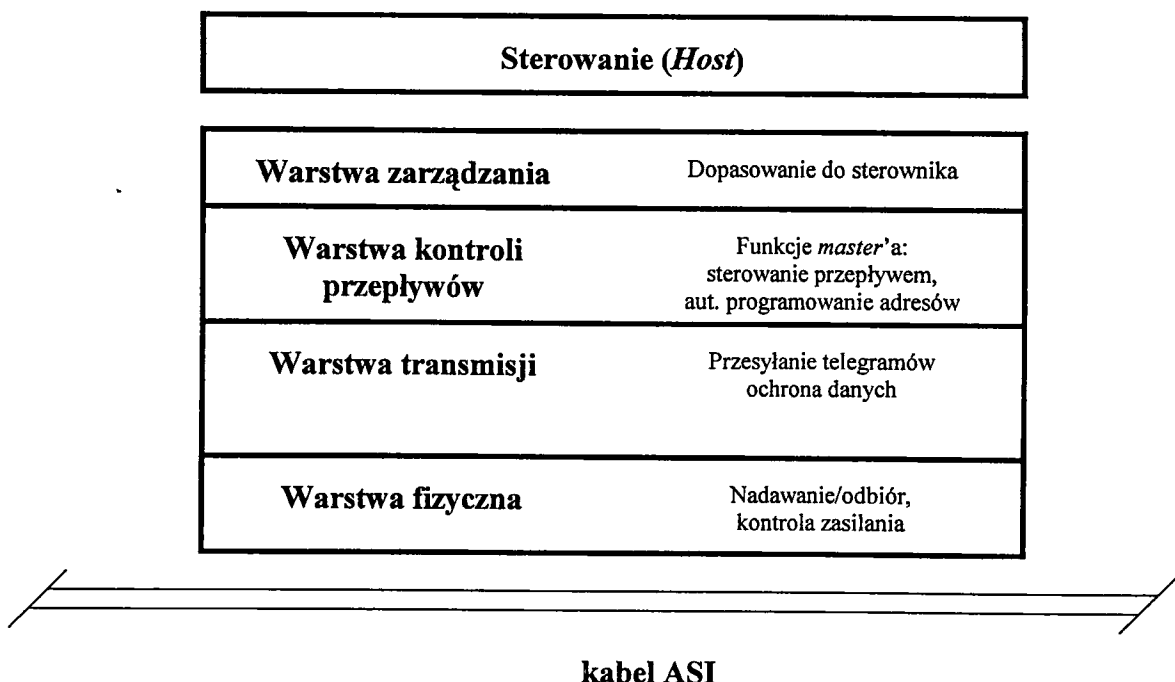
5.5.6.2 *ASI - master*

Master ASI stanowi interfejs między systemem transmisji ASI, a jądrem sterowania systemu (np. logicznym sterownikiem programowalnym PLC lub komputerem PC). Karta *master*'a zastępuje dotychczasowe moduły wejść/wyjść przeznaczone do komunikacji z czujnikami i elementami wykonawczymi w sposób tradycyjny, z wykorzystaniem topologii gwiazdy. Wszystkie sygnały wejścia/wyjścia sterownika zostają odwzorowane w słowach 4 bitowych. Dodatkowo master ASI oferuje możliwość precyzyjnej diagnostyki systemu, bez konieczności stosowania dodatkowych środków. Odbywa się to dzięki mechanizmom ciągłej kontroli stanu systemu, co daje możliwość łatwego uzupełniania i przekonfigurowania w przypadku uszkodzeń urządzeń typu *slave*.

Moduły master ASI pełnią również funkcje łączące z nadrzędnymi systemami komunikacji, tzw. magistralami przemysłowymi (ang. *Fieldbus*).

Specyfikacja *master*'a (*profile*) ma za zadanie ułatwić jego użycie przez użytkownika, ale również umożliwić pełne wykorzystanie możliwości nowoczesnych sterowników. Z tego względu dokonywana jest ona na pewnym stopniu abstrakcji, co uniezależnia ją od szczegółowych rozwiązań sprzętowych.

Specyfikacja *master*'a została przedstawiona w sposób warstwowy na rys.5.5.12.



Rys.5.5.12 Model warstwowy *master*'a ASI

Zadaniem każdej z warstw jest maskowanie, dla użytkownika, szczegółowego przepływu danych oraz dostarczenie podstawowych funkcji. Bardzo istotnym zadaniem każdej z warstw jest obsługa błędów.

Warstwa fizyczna: ma do spełnienia podobną funkcję, jak *ASI slave*. W warstwie tej zachodzi właściwa transmisja bitowa.

Warstwa transmisji: stanowi najniższą logiczną warstwę, która jest odpowiedzialna za przesyłanie i odbieranie telegramów. Dzięki mechanizmom kontroli poprawności oraz wymuszania powtórzeń, w przypadku wystąpienia błędów, jedynie poprawne telegramy przekazywane są do warstwy wyższej.

Warstwa kontroli przepływów: generuje zadania transmisji danych dla poniższej warstwy transmisyjnej, a jej działanie oparte jest na trzech podstawowych stanach:

- inicjowanie pracy,
- rozruch systemu,
- praca normalna.

Warstwa zarządzania: najwyższa, dopasowująca do jądra systemu sterowania (*host system*). Wprowadza ona ograniczenia dla nie zdefiniowanych rozkazów.

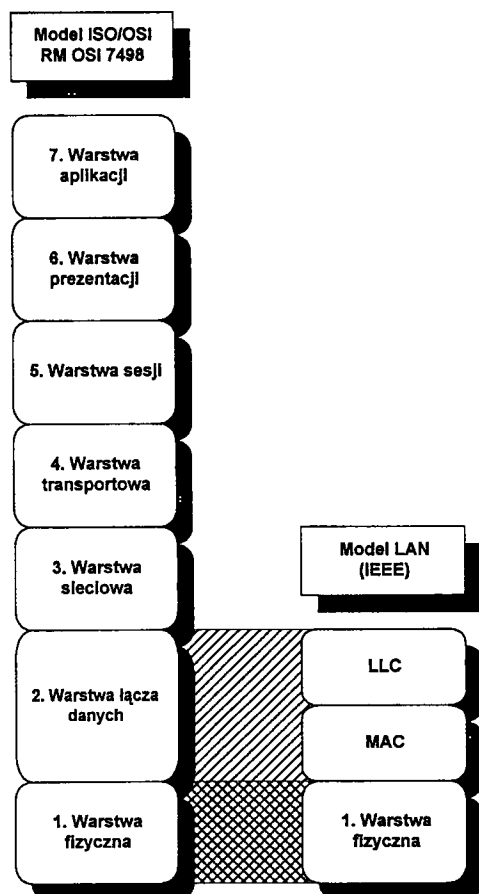
Literatura

- [1] Werner Kriesel, Otto W. Madelung (Hrsg), "*ASI. Das Aktuator - Sensor - Interface für Automation*", Carl Hanser Verlag München Wien, 1994 (dostępna jest również wersja anglojęzyczna wydania)
- [2] Siemens, *Aktuator - Sensor - Interface Systembeschreibung*, Erlangen 11/95
- [3] *Actuator Sensor Interface (ASI)*, Draft IEC, SC17B, WG3

6. Standardy sieci lokalnych LAN

6.1 Wstęp

W górnych oraz środkowych poziomach zakładu produkcyjnego od wielu lat stosuje się sieci lokalne LAN. Sieć LAN obejmuje swoim zasięgiem budynek i doskonale spełnia swoje zadania przy połączeniach komputerów w technicznych oraz ekonomicznych działach przedsiębiorstwa.



Rys.6.1. Model Odniesienia ISO i model odniesienia LAN

Model LAN wg IEEE 802 obejmuje dwie najniższe warstwy Modelu Odniesienia ISO (z pewnymi funkcjami warstwy sieciowej). Warstwę łącza danych (*Data Link Layer*) podzielono na dwie podwarstwy:

- Sterowanie dostępem do medium **MAC** (ang. *Medium Access Control*) oraz
- Sterowanie łączem logicznym **LLC** (ang. *Logical Link Control*).

Podwarstwa MAC definiuje rozwiązania sprzętowe i programowe dla konkretnego typu sieci, natomiast podwarstwa LLC tworzy pomost do wyższych warstw wg modelu OSI. Aktualna struktura komitetu IEEE Project 802 jest przedstawiona na rys.6.2. Poniżej przedstawione są podkomitety opracowujące sieci lokalne, a także sieci miejskie MAN (ang. *Metropolitan Area Network*). I tak podkomitet:

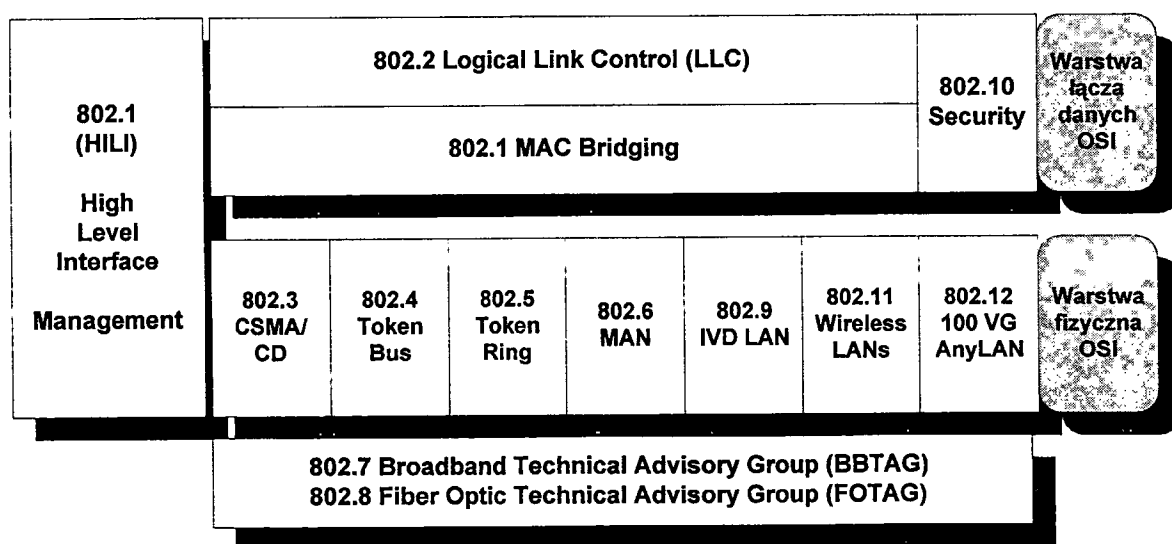
Standaryzacją lokalnych sieci komputerowych na szeroką skalę zajął się amerykański Instytut Inżynierów Elektryków i Elektroników IEEE (ang. *Institute of Electrical and Electronic Engineers*) oraz organizacja ISO (ang. *International Standard Organisation*). IEEE powołał komitet IEEE 802 (oznaczenie od daty powstania w lutym 1980r.), który zainicjował szereg projektów dotyczących interfejsów, protokołów i zagadnień implementacyjnych dla sieci LAN. Pierwsze prace studialne wykazały, że zadanie komunikacji w ramach sieci lokalnej wymaga podziału na szereg mniejszych, oddzielnych zadań oraz że nie ma jednego uniwersalnego rozwiązania spełniającego rozbieżne niekiedy wymagania. Ponadto należało uwzględnić istniejące rozwiązania firmowe, które znalazły już swoje miejsce na rynku.

Ponieważ bezpośrednie zastosowanie siedmio-warstwowego modelu odniesienia ISO/OSI (ang. *International Standard Organisation / Open System Interconnection*) dla sieci lokalnych okazało się niecelowe, dlatego też w ramach *IEEE Project 802* przyjęto Model Odniesienia dla Sieci Lokalnej (ang.

- 802.1 opracowuje warunki współpracy sieci (ang. *internetworking*) - określa relacje między standardami 802.x, a modelem ISO/OSI. Komitet 802.1 definiuje, na użytek wszystkich standardów 802, adres stacji w sieci lokalnej (adres 48 bitowy). Wszyscy producenci kart sieciowych są zobowiązani do stosowania unikatowych adresów kart sieciowych,
- 802.2 opracowuje podwarstwy: sterowania dostępem do medium MAC oraz sterowania łączem logicznym LLC (rys.5.). W odniesieniu do modelu ISO/OSI podwarstwy te znajdują się w warstwie łącza danych. LLC posługuje się adresami punktów dostępu do usługi SAP (ang. *Service Access Points*), natomiast podwarstwa MAC - adresami fizycznymi urządzeń sieciowych. LLC jest odpowiedzialne za usługi połączeniowe (ang. *Connection-oriented service*), usługi połączeniowe z potwierdzeniami (ang. *Acknowledged connection-oriented services*) i usługi połączeniowe bez potwierdzenia (ang. *Unacknowledged connection-oriented services*).
- 802.3 (ISO 8802.3) opracowuje sieci typu CSMA/CD (ang. *Carrier Sense Multiple Access/Collision Detection*). Standard ten określa szybkość transmisji danych 10Mbit/s i przewiduje użycie przewodu koncentrycznego, skrętki i światłowodu. Nowsze rozwiązania pozwalają na szybkość transmisji 100Mbit/s.
- 802.4 opracowuje szerokopasmową sieć typu *Token Bus* - opracowaną do wykorzystania w przemyśle wytwórczym. Standard ten wykorzystuje technikę przekazywania znacznika (ang. *Token passing*). *Token* przesyłany jest od stacji do stacji, a stacja mająca *token* jest uprawniona do transmisji danych. Przekazywanie znacznika tworzy pierścień logiczny, który nie musi pokrywać się z pierścieniem fizycznym wyznaczonym przez kolejność stacji.
- 802.5 sieć typu *Token Ring* - spopularyzowaną przez firmę IBM. Standard ten definiuje sieć o topologii gwiazdy, ale tworzącą pierścień logiczny. Wykorzystuje się tu technikę przekazywania znacznika (ang. *Token passing*). Poszczególne węzły są włączone do jednostki centralnej (koncentratora). Na protokole 802.5 bazuje również sieć światłowodowa FDDI, ale jej standard został opracowany przez komitet ASC X3T9. Sieć FDDI jest kompatybilna ze standardami 802 poprzez warstwę sterowania łączem określoną w 802.2.
- 802.6 opracowuje sieci miejskie MAN. Standard ten określa szybki protokół obsługujący stacje włączone do podwójnej magistrali światłowodowej. Zaletą tego standardu jest możliwość utrzymania połączenia nawet w przypadku uszkodzenia magistrali. W rozwiązaniu tym wykorzystywana jest metoda dostępu do medium określana jako DQDB (ang. *Distributed Queue Dual Bus*). Standard ten umożliwia także przesyłanie danych, głosu i obrazu na obszarze o zasięgu do 50 km z szybkością do 155Mbit/s. Przekaz danych następuje w formie komórek o stałej długości 53 bajtów, a zatem sieć ta jest kompatybilna z protokołem B-ISDN oraz ATM.
- 802.7 to grupa doradztwa technicznego do spraw szerokiego pasma w sieciach LAN (ang. *broadband LAN*).
- 802.8 to grupa doradztwa technicznego do spraw światłowodów w sieciach LAN oraz MAN.

- 802.9 to grupa robocza zajmująca się zintegrowaną transmisją głosu i danych w sieciach LAN i MAN. Specyfikacja węzłów takich sieci obejmuje komputery, telefony oraz kodery/dekodery video. Generalnie grupa 802.9 zajmuje się zintegrowanymi sieciami cyfrowymi ISDN.
- 802.10 to grupa doradcza do spraw bezpieczeństwa, ochrony danych i zasobów sieci.
- 802.11 opracowuje lokalne sieci bezprzewodowe. Komitet ten zajmuje się problemami transmisji radiowej wąsko- i szerokopasmowej, transmisją w podczerwieni oraz za pomocą linii energetycznych. Standard ten przewiduje stosowanie dwóch rodzajów sieci bezprzewodowych: rozproszonych i z punktem koordynacyjnym.
- 802.12 opracowuje opis funkcji warstwy MAC oraz warstwy fizycznej dla sieci 100VG AnyLAN. W standardzie tym przewidziano stosowanie techniki dostępu do medium transmisyjnego określanej jako priorytet na żądanie (ang. *Demand Priority*).

Jak wynika z rysunku 6.2, zdefiniowany został tylko jeden wariant warstwy LLC (802.2), który jest wspólny dla wszystkich typów sieci lokalnych. Typ sieci lokalnej jest określony przez warstwę MAC, przy czym danemu protokołowi MAC odpowiada wiele możliwych sposobów realizacji warstwy fizycznej. Określony protokół MAC oraz dopuszczone standardem rodzaje warstwy fizycznej są opisane we wspólnym dokumencie standaryzacyjnym.



Rys.6.2 Struktura komitetu IEEE 802

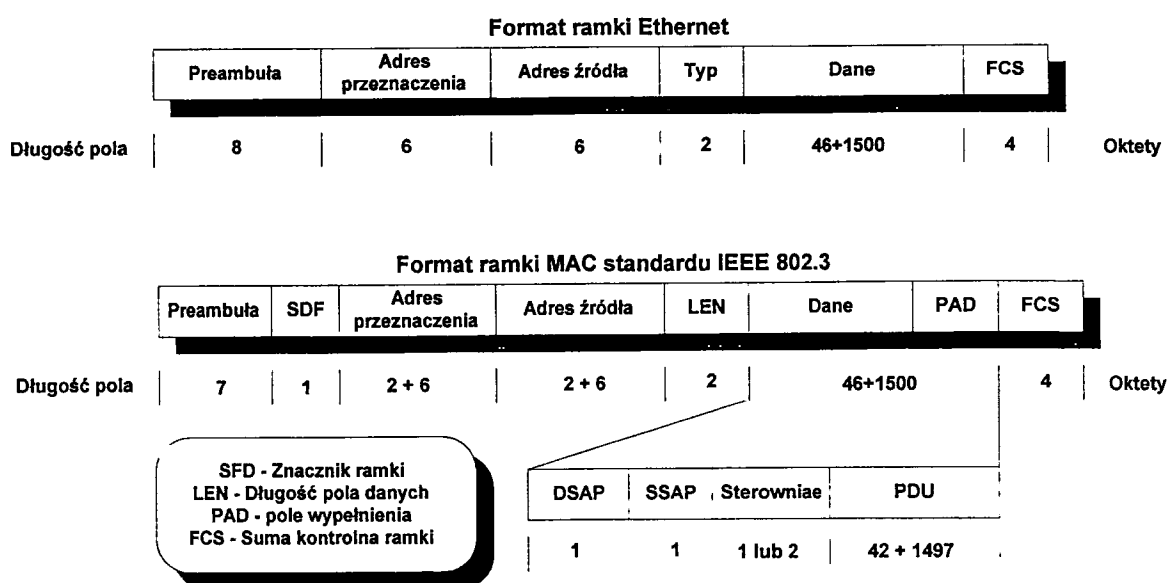
6.2 Sieć typu Ethernet (IEEE 802.3)

Chociaż Ethernet został opracowany przez firmę Xerox, to dopiero współpraca z firmami takimi jak Intel i DEC doprowadziła do upowszechnienia tej technologii na całym niemal świecie. Ethernet jest rozwiązaniem bardzo podobnym do standardu IEEE 802.3, lecz różniącym się formatem ramki (rys.6.3). Standard 802.3 został przyjęty przez międzynarodową organizację standaryzującą ISO. Format ramki Ethernet składa się z 6 następujących części:

- **Pola preambuły** - której zadaniem jest zapewnienie synchronizacji. Preambuła składa się z sekwencji ośmiu oktetów, z których pierwsze 6 zawiera ciąg jedynek 10101010 a ostatni ósmy ma postać 10101011.
- **Pola adresu przeznaczenia** - zawiera adres warstwy fizycznej lub adres grupowy stacji przeznaczenia.
- **Pola adresu źródła** - zawiera adres stacji nadającej ramkę.
- **Pola typu** - określa typ protokołu warstw wyższych np. TCP/IP.
- **Pola danych** - o długości 46 do 1500 oktetów, zawiera dowolne dane.
- **Pola FRC (Frame Redundancy Checksum)** - zawiera sumę kontrolną ramki.

Część producentów sprzętu sieciowego dostarcza swoje produkty (np. karty sieciowe, koncentratory), które przyjmują i obsługują zarówno formaty ramek Ethernet'owych jak i IEEE 802.3. Przepustowość sieci typu Ethernet wynosi 10 Mbit/s. W sieciach tych stosuje się metodę dostępu do medium transmisyjnego CSMA/CD (ang. *Carrier Sense Multiple Access with Collision Detection*). Dzięki tej metodzie dostępu stacje robocze mogą korzystać ze wspólnego medium - kabla sieciowego, ale w danym czasie może korzystać z niego tylko jedna stacja. Stacja, która zamierza rozpocząć transmisję danych, nadśłuchuje, czy aktualnie nie odbywa się transmisja między innymi stacjami. Jeśli tak, to stacja ta odczekuje na koniec transmisji i w przypadkowo wybranym czasie próbuje ponownie wysłać pakiet danych. Jednoczesne rozpoczęcie transmisji przez dwie lub więcej stacji jest wykrywane przez wbudowane mechanizmy detekcji kolizji. W przypadku wykrycia kolizji stacje wstrzymują wysyłanie danych i rozpoczynają ponownie nadawanie w różnych odstępach czasowych.

Na rysunku 6.3 pokazano formaty ramek sieci Ethernet.

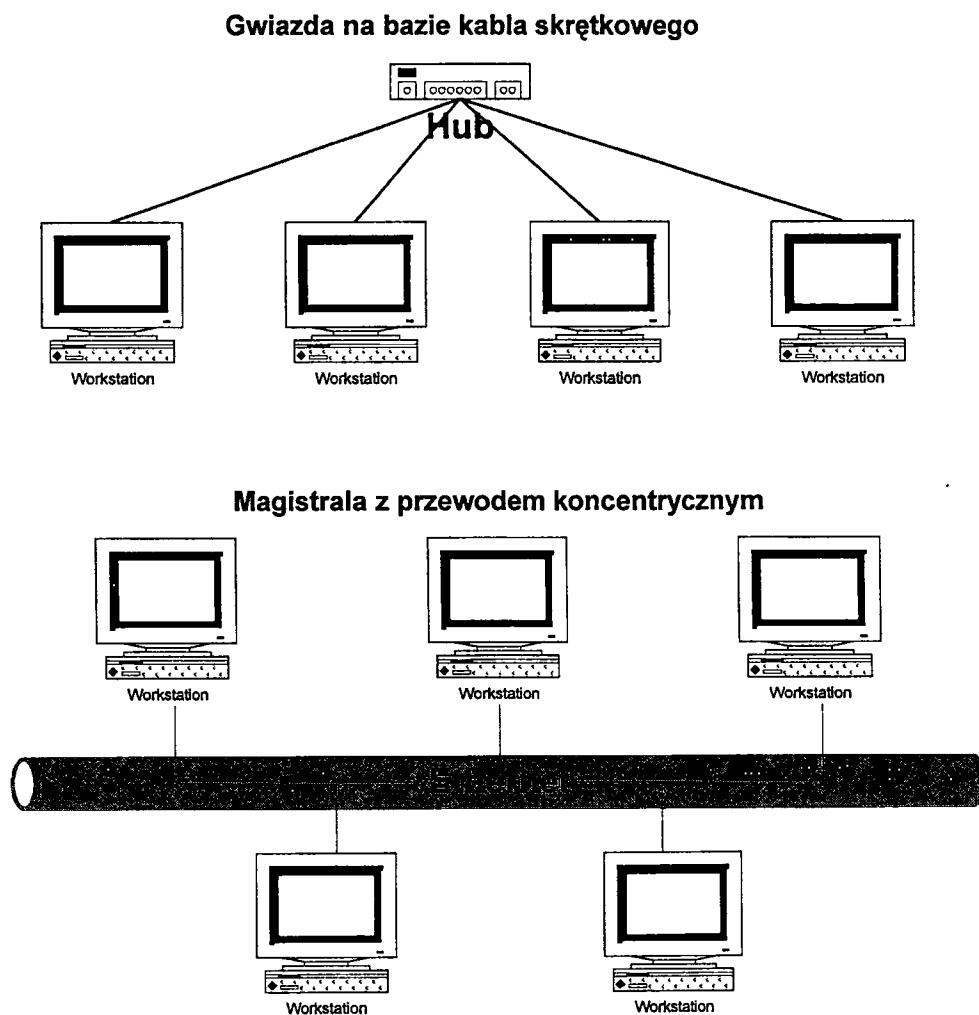


Rys.6.3 Formaty ramek Ethernet oraz MAC IEEE 802.3

Sieci Ethernet (IEEE 802.3) pozwalają także na wykorzystywanie różnorodnych mediów transmisyjnych, które określają także topologię sieci:

- 10Base5 - przewód koncentryczny RG-11,
- 10Base2 - przewód koncentryczny (RG-58A/U),
- 10BaseT - para przewodów skręconych,
- 10Broad36 - przewód koncentryczny (RG-59/A/U/CATV),
- 10BaseF - światłowód,
- 100BaseT4 - skrętka wieloparowa UTP kategorii 3,4 i5,
- 100BaseTX - para przewodów skręconych UTP kategorii 5,
- 100BaseFX - dwuwłóknowy światłowód.

Na rysunku 6.4 pokazano sieci Ethernet w konfiguracji z przewodem koncentrycznym i *skrętką*.



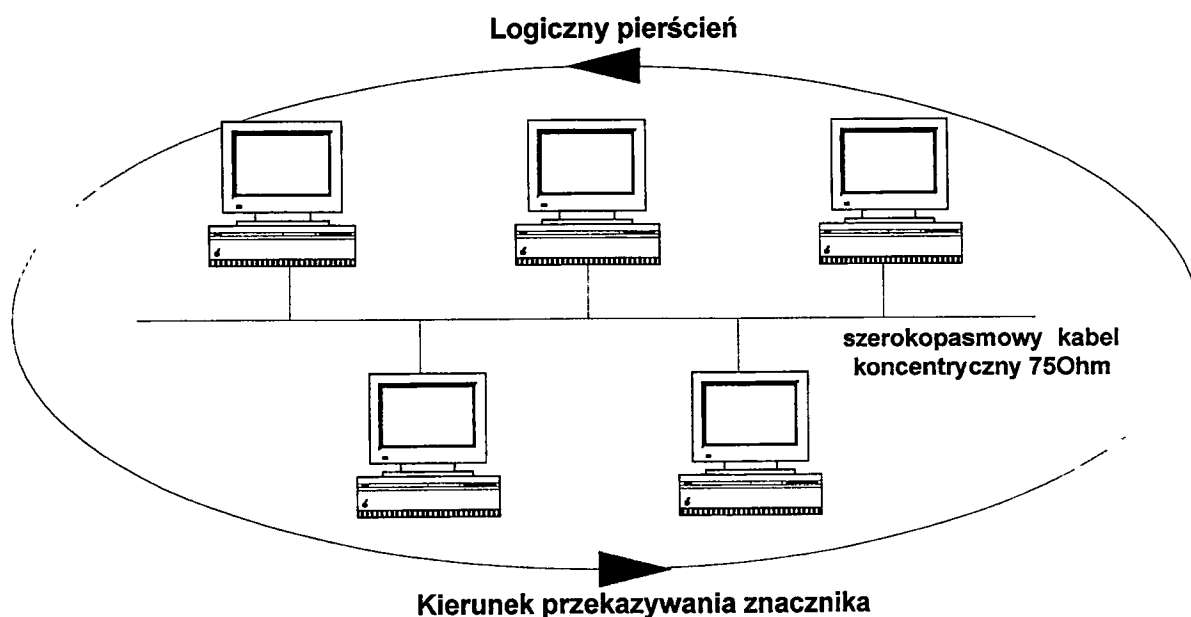
Rys.6.4 Sieć Ethernet w przykładowych konfiguracjach

6.3 Sieć typu *Token Bus* (IEEE 802.4)

W sieci typu *Token Bus* stacja może transmitować dane tylko wtedy, gdy jest w posiadaniu znacznika (ang. *token'a*). Standard ten został unormowany przez komitet IEEE 802.4 i przewiduje on, w odróżnieniu od sieci Ethernet, wykorzystanie transmisji szerokopasmowej (ang. *broadband*).

Sieci typu *Token Bus* charakteryzują się wykorzystaniem kabla koncentrycznego 75 Ω , stosowanego w telewizji kablowej, jako medium transmisyjnego. Wynika stąd topologia sieci, a mianowicie wspólna magistrala. Transmisja szerokopasmowa umożliwia przesyłanie informacji wieloma kanałami równocześnie z szybkością transmisji 10 Mbit/s. Z zastosowanej w tym standardzie techniki przekazywania znacznika wynika szereg wniosków co do fizycznego i logicznego rozmieszczenia poszczególnych stacji (rys.6.5):

- znacznik wędruje od stacji do stacji i jest przekazywany wzdłuż logicznego pierścienia, który nie musi odpowiadać fizycznemu rozmieszczeniu stacji sieciowych,
- kolejność, w której znacznik odwiedza stacje jest uwarunkowana adresami stacji w sieci,
- ostatnia stacja w sieci (tzn. o najniższym numerze) przekazuje znacznik do pierwszej stacji (tzn. o najwyższym numerze).



Rys.6.5 Zasada przekazywania znacznika w sieci *Token Bus*

Topologia sieci *Token Bus* uwzględnia grupy stacji podłączonych za pomocą długich kabli głównych (ang. *trunk cables*). Stacje te są podłączone do *hub'a*, tworząc konfigurację gwiazdy. Taka sieć ma w tym przypadku topologię magistrali i gwiazdy.

Przykładem sieci *Token Bus* jest sieć ARCNET, która jednak nie jest w pełni zgodna ze standardem IEEE 802.4. Standard ten ma zastosowanie w przemyśle wytwórczym i jest ujęty w specyfikacji MAP 3.0 (ang. *Manufacturing Automation Protocol*).

Na rysunku 6.6 pokazano format ramki standardu IEEE 802.4.

Format ramki MAC standardu IEEE 802.4

	P	SD	FC	DA	SA	Dane	FCS	ED
Długość pola	1	1	1	2 lub 6	2 lub 6	0-8182	4	1

P - Preambuła	SA - Adres źródła
SD - Znacznik początku	FCS - Suma kontrolna
AC - Sterowanie dostępem	ED - Znacznik końca
FC - Nadzór ramki	FS - Status ramki
DA - Adres przeznaczenia	

Rys.6.6 Format ramki standardu IEEE 802.4

6.4 Sieć typu Token Ring (IEEE 802.5)

Sieci typu *Token Ring* zostały ujęte standaryzacją przez komitet IEEE 802.5. Podobnie jak wspomniany już *Token Bus*, sieć *Token Ring* wykorzystuje technikę przekazywania znacznika (ang. *token'a*) (rys.6.8). Pierwsze rozwiązania sieci tego typu zaproponowała firma IBM w połowie lat 80-tych. Topologia fizyczna sieci w postaci gwiazdy wymaga zastosowania koncentratora, umożliwiającego dowolne przyłączanie i odłączanie stacji sieciowych. Ważne jest przy tym to, aby pierścień logiczny nie został naruszony w czasie przekonfigurowania sieci.

Format ramki znacznika IEEE 802.5

	Znacznik początku	Sterowanie dostępem	Znacznik końca
Długość pola	1	1	1

bajty

Format ramki MAC standardu IEEE 802.5

	SD	AC	FC	DA	SA	Dane	FCS	ED	FS
Długość pola	1	1	1	2 lub 6	2 lub 6	0-32	4	1	1

bajty

SD - Znacznik początku	SA - Adres źródła
AC - Sterowanie dostępem	FCS - Suma kontrolna
FC - Nadzór ramki	ED - Znacznik końca
DA - Adres przeznaczenia	FS - Status ramki

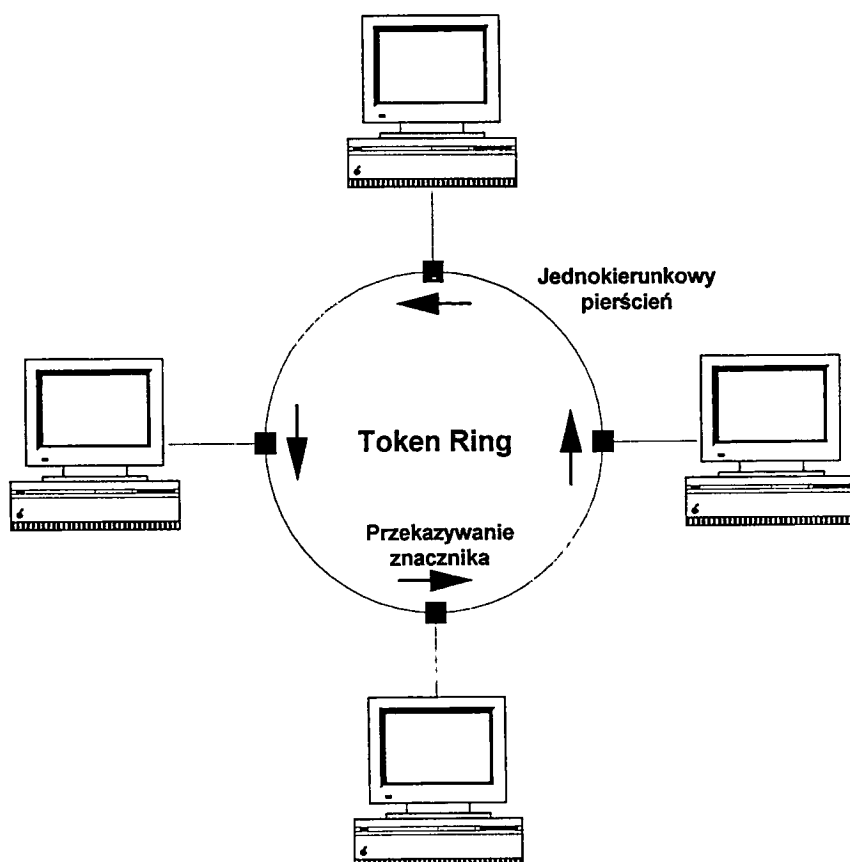
Rys.6.7 Formaty ramek w standardzie IEEE 802.5

W sieci tej wykorzystano trzy typu ramek (rys.6.7):

- **ramka znacznika** - jest ona odpowiedzialna za tworzenie pierścienia logicznego i przekazywanie uprawnień,
- **ramka informacyjna** - służąca do przekazywania danych i informacji sterujących oraz

- **ramka sekwencji odrzucania** - występuje w razie wystąpienia trwałego błędu w sieci.

Ramka znacznika ma długość trzech bajtów i składa się z pola znacznika początku wg DMC (*Differential Manchester Code*), pola sterownia dostępem, które zapewnia dostęp do sieci oraz pola końca ramki. Przekazywanie danych w sieci IEEE 802.5 odbywa się poprzez ramkę informacyjną, która jest złożona z: pola znacznika początku, pola sterownia dostępem, pola nadzoru ramki (definiuje rodzaj ramki), pola adresów przeznaczenia i źródła, pola informacji *routing*'u, pola danych, pola sumy kontrolnej i pola końca ramki.



Rys.6.8 Zasada przekazywania znacznika w sieci Token Ring

Stacje robocze przyłącza się do koncentratorów, które w tym standardzie nazywane są jednostkami MAU (ang. *Multistation Access Unit*). Aby utworzyć większą sieć możliwe jest łączenie MAU w kaskady. Pierścień logiczny w sieci *Token Ring* jest realizowany w jednostkach MAU, które także mają możliwość automatycznego omijania stacji nie włączonych oraz odłączania uszkodzonych stacji w celu zachowania pierścienia.

Jednostki MAU, obok gniazd do łączenia stacji sieciowych, wyposażone są w gniazda typu *ring-in* oraz *ring-out*, służące do połączeń z innymi jednostkami MAU w tzw. kaskady.

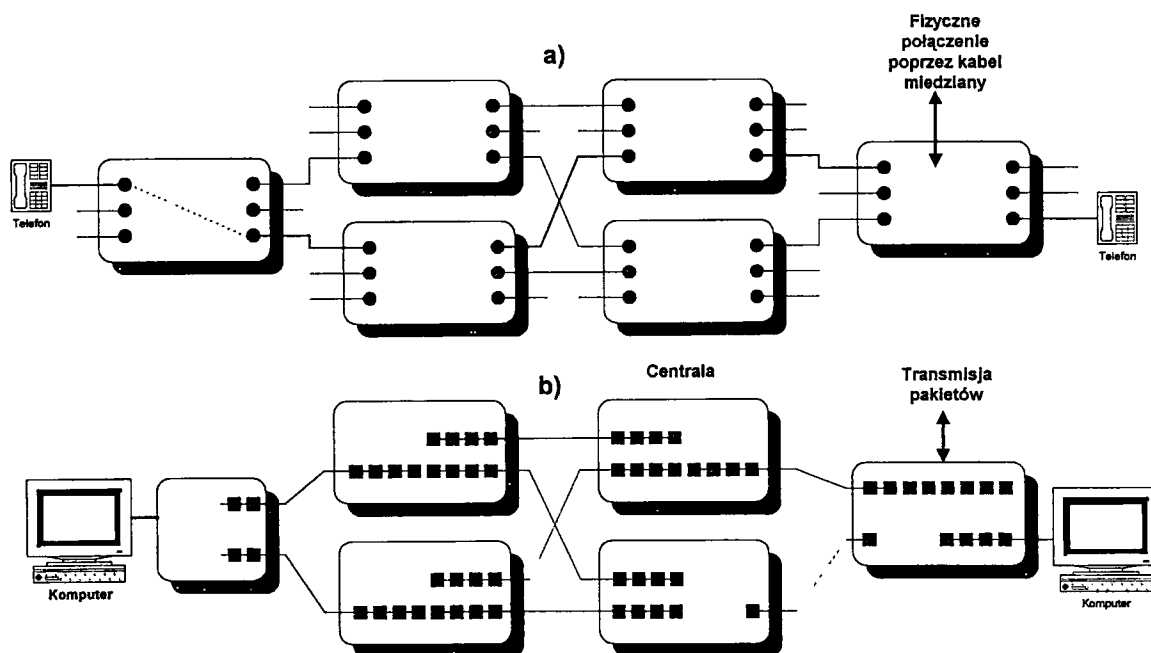
Obecnie na rynku oferuje się osprzęt sieciowy zgodny ze standardem 802.5, który jest rozwinięciem koncepcji IBM (4 Mbit/s, 16 Mbit/s). Najczęściej wykorzystuje się nie ekranowaną skrętkę i 16-portowe jednostki MAU. Specyfikacja *Token Ring* dopuszcza zastosowanie następujących kabli: typ 1, typ 2, typ 3, typ 5, typ 6, typ 8 i typ 9.

7. Rozległa sieć komputerowa - WAN

7.1 Wstęp

Rozległa sieć komputerowa WAN jest systemem telekomunikacyjnym, którego zadaniem jest połączenie oddalonych geograficznie od siebie systemów komputerowych w jedną sieć informatyczną. Łączy ona systemy komputerowe znajdujące się poza obszarem należącym do jednego lub wielu budynków, sięgając po obszary publiczne, krajowe i międzynarodowe. Połączenia w sieciach WAN są zazwyczaj realizowane za pomocą publicznej sieci telekomunikacyjnej - PSTN (ang. *Public Switched Telephone Network*), a także publicznej sieci pakietowej PSDN (ang. *Public Switched Data Network*).

Publiczna sieć telekomunikacyjna PSTN stanowi infrastrukturę telekomunikacyjną opartą o technikę komutacji łączy (linii telefonicznych) (rys.7.1a). Łączność realizowana w oparciu o linie komutowane nie różni się zbytnio od typowej rozmowy telefonicznej, gdzie pomiędzy dwoma punktami zestawiany jest - na wymagany czas - dedykowany kanał komunikacyjny. Przekazywanie danych w sieciach PSTN może odbywać się przez sieci analogowe lub sieci cyfrowe. Do przekazywania danych przez analogowe linie telefoniczne wykorzystywane są modemy. Szybkość transmisji realizowanej za pomocą modemu może osiągnąć wartość do 28.8 kbit/s (standard modemowy ITU/CCITT V.34). Połączenia tego typu nadają się dobrze do nawiązywania łączności okazjonalnych, np. w celu wymiany poczty elektronicznej lub zdalnego sterowania komputerem (usługa *telnet*).



Rys.7.1 Sieć PTSN: a) technika komutacji łączy, b) technika komutacji pakietów

Dynamicznie rozwijająca się od kilku lat cyfrowa sieć telefoniczna, taka jak ISDN, otwiera zupełnie nowe możliwości połączenia różnorodnego sprzętu, takiego jak: telefon, telefaks, komputer, wideotelefon itp.

Publiczna sieć pakietowa PSDN jest siecią transmisji danych działającą w trybie pakietowym, w której dane w formie pakietów mogą być transmitowane przez wielu uczestników jednocześnie (każdy z każdym). Sieć ta składa się z systemu połączeń, którymi przesyłane są zaadresowane pakiety z danymi. System wysyłający dane dzieli je na porcje umieszczane w pakietach o określonej długości, a następnie pakiety te są dostarczane do celu przy wykorzystaniu usługi połączeniowej lub bezpołączeniowej (rys.7.1b).

7.2 Sieć cyfrowa z integracją usług - ISDN

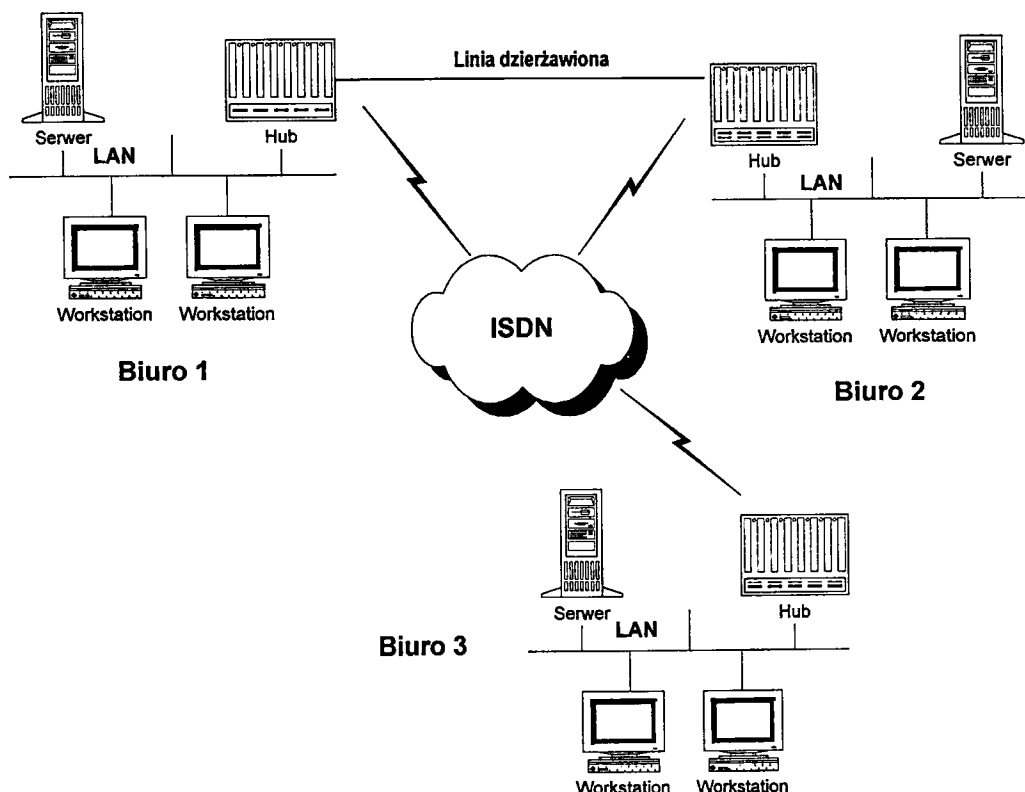
Sieć cyfrowa z integracją usług ISDN (ang. *Integrated Services Digital Network*) została opracowana dla zapewnienia w pełni cyfrowych połączeń telekomunikacyjnych. Standard ISDN powstał jako tzw. *wąskopasmowy ISDN* i zapewniał on pracę z szybkością do 2 Mbit/s, przy użyciu kabla miedzianego. W ostatnich latach pojawiło się rozwiązanie *szerokopasmowy ISDN (B-ISDN)*, zapewniające pracę z szybkościami rzędu 100 Mbit/s a nawet Gbit/s. Sieć ISDN składa się z szeregu kanałów transmisji danych, oznaczonych jako B i D. Kanał B umożliwia transmisję danych z szybkością 64 kbit/s i jest on wykorzystywany do przesyłania głosu i danych. Kanał D, o szybkości 16 lub 64 kbit/s, jest natomiast przeznaczony do wymiany sygnałów kontrolnych między centralą telefoniczną a urządzeniem końcowym ISDN (np. telefonem). Ponieważ przekaz informacji i sygnałów kontrolnych w kanale B dokonuje się w trybie komutacji pakietowej, dlatego też jest możliwe wykorzystanie tego kanału do transmisji pakietów danych w sieciach X.25 lub *Frame Relay*.

Dwa kanały typu B (64 kbit/s) oraz jeden kanał typu D (16 kbit/s) składają się na najmniejsze dostępne pasmo transmisyjne, tzw. 2B+D o maksymalnej przepustowości 144 kbit/s ($64+64+16$). Pasma to nosi nazwę BRA (ang. *Basic Rate Access*) i jest przeznaczone dla abonentów o niewielkim zapotrzebowaniu na binarną przepływność informacji. Pasma BRA umożliwia np. równoczesną realizację połączenia dwóch lub więcej urządzeń końcowych takich jak telefon, faks, komputer.

Dla abonentów wymagających zwiększonej przepustowości sieci ISDN są dostępne dodatkowe kanały komunikacyjne tzw. 30B+D (64 kbit/s). Kanały te noszą nazwę PRA (ang. *Primary Rate Access*) i umożliwiają transmisję danych w paśmie 1984 kbit/s. Dostęp do sieci typu PRA jest wykorzystywany do przyłączenia centralek abonenckich typu PABX lub koncentratorów sieci LAN.

Zastosowanie ISDN w sieciach komputerowych jest często alternatywą dla realizacji linii dzierżawionych lub korzystania z innych usług WAN takich jak X.25 lub *Frame Relay* (rys.7.2). Generalnie wyróżnia się zastosowanie ISDN w sieciach komputerowych jako:

- łącze awaryjne dla sieci WAN (ang. *Primary Line Backup*) - firmy korzystające z połączenia pomiędzy oddalonymi biurami poprzez linie dzierżawione, sieć X.25 lub *Frame Relay* mogą, w przypadku uszkodzenia tych linii, zestawiać połączenie korzystając z ISDN,
- łącze WAN na żądanie (ang. *Dial on Demand*) - jest stosowane w tych przypadkach, kiedy odległe oddziały firmy nie mają dedykowanego połączenia w postaci linii dzierżawionej, X.25 czy *Frame Relay*, a potrzebują okazjonalnie wymieniać dane ze swoimi biurami. Router zestawia połączenie z odległym oddziałem jedynie w momencie transmisji danych,
- dodatkowe pasmo na żądanie (ang. *Bandwidth on Demand*) - w przypadku tym odległe oddziały firm są połączone przy użyciu linii dzierżawionych, X.25 lub *Frame Relay*, jednak pasmo połączenia WAN jest w pewnych wyjątkowych przypadkach (chwilowe przeciążenie łączy komunikacyjnych) niewystarczające i dlatego też należy zwiększyć przepustowość łącza WAN poprzez ISDN.



Rys.7.2 Wykorzystanie sieci ISDN do połączenia oddalonych biur

7.3 Sieć pakietowa X.25

W celu zapobieżenia rozwojowi niekompatybilnych rozwiązań sieci pakietowych w różnych krajach na świecie, w roku 1974 organizacja CCITT (franc. *Comité Consultatif International de Télégraphique et Téléphonique*) zaproponowała międzynarodowy standard protokołu znany jako X.25. Standard ten był w kolejnych latach weryfikowany i uaktualniany. X.25 jest dobrze znanym i

powszechnie stosowanym standardem w sieciach PSDN, w szczególności do połączeń odległych terminali z *host*'ami komputerowymi.

Standard X.25 definiuje interfejs pomiędzy systemem *host* (np. komputerem), nazywanym DTE (ang. *Data Terminal Equipment*) a urządzeniami pośredniczącymi w przekazywaniu danych DCE (ang. *Data Circuit-terminating Equipment*). Standard X.25 definiuje również format oraz znaczenie informacji przekazywanej poprzez interfejs DTE-DCE. Ponieważ standard ten został opracowany wcześniej w stosunku do modelu protokołów OSI, dlatego też związana z nim terminologia jest w niektórych przypadkach inna niż terminologia przyjęta przez OSI. W standardzie X.25 zdefiniowano protokoły trzech warstw, tzn. 1, 2 oraz 3 wg modelu ISO/OSI.

Warstwa fizyczna (1 - *physical layer*) zajmuje się elektrycznym, mechanicznym, proceduralnym oraz funkcjonalnym interfejsem pomiędzy DTE a DCE. Aktualnie X.25 nie definiuje konkretnych rozwiązań dla tej warstwy lecz odnosi się do dwóch innych standardów, a mianowicie: X.21 i X.21 bis, które definiują cyfrowy i analogowy interfejs komunikacyjny.

Zadaniem warstwy dostępu do łącza (2 - *link access layer*) jest zapewnienie niezawodnej komunikacji pomiędzy DTE a DCE. Wykorzystywane są tu protokoły takie jak LAP (*Link Access Procedure*) i LAPB (*Link Access Procedure B*). LAP-B jest częścią protokołu HDLC (*High-level Data Link Control*) i został zaprojektowany do obsługi połączeń typu punkt z punktem. Definiuje on strukturę ramki, mechanizmy kontroli przepływu danych i mechanizmy kontroli błędów. LAP-B jest także odpowiedzialny za potwierdzanie przejścia pakietów przez każde łącze w sieci.

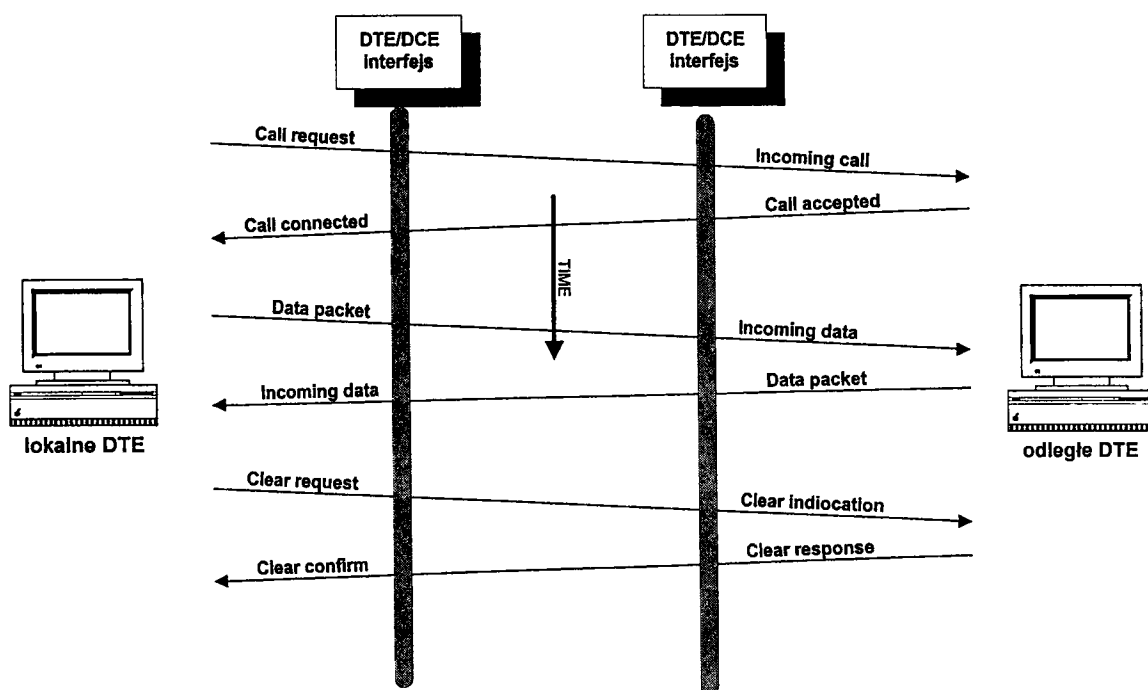
Warstwa pakietowa (3 - *packet layer*) określa niezawodne połączenia wirtualne pomiędzy urządzeniami DTE w sieci pakietowej. Warstwa ta jest często nazywana X.25 PLP (ang. *Packet Layer Protocol*) i pozwala przysyłać dane pomiędzy dwoma lub wieloma uczestnikami komunikacji.

Ważnym elementem w koncepcji sieci X.25 jest tzw. łącze wirtualne (*virtual circuit*), które ustanawia stały lub tymczasowy kanał komunikacyjny. Zastosowanie łącza wirtualnego gwarantuje przybycie pakietów do celu w tej samej kolejności w jakiej zostały wysłane. W standard X.25 uwzględnia dwa typy łączy wirtualnych:

- tymczasowe łącze wirtualne (*temporary call-based virtual circuits*), nazywane także połączeniem wirtualnym (*virtual circuits*) w praktyce jest podobne do typowego połączenia telefonicznego. Połączenie to jest realizowane wyłącznie na czas trwania sesji realizującej transfer danych i odbywa się w trzech fazach: ustanowienie połączenia, transfer danych i zerwanie połączenia.
- stałe łącze wirtualne (*permanent virtual circuits*), w przeciwieństwie do łącza tymczasowego, jest podobne do linii dzierżawionej tzn. połączenie jest utrzymywane stale pomiędzy dwoma węzłami końcowymi DTE. Połączenia tego typu są używane w sytuacjach, kiedy istnieje potrzeba transferu dużych ilości danych w różnym czasie.

Połączenia wirtualne wg standardu X.25 są dokonywane w trzech następujących fazach (rys.7.3):

1. Kiedy lokalny system DTE (np. komputer) chce nawiązać komunikację z innym, oddalonym systemem DTE, to w pierwszej fazie ustanawia logiczne połączenie. W tym celu wysyła do lokalnej stacji DCE pakiet o nazwie CALL REQUEST. Pakiet ten jest następnie przekazywany, poprzez system komunikacyjny, do stacji docelowej DTE. Jeśli stacja przeznaczenia DTE zaakceptuje żądanie ustanowienia połączenia, to wysyła ona pakiet zwrotny CALL ACCEPT. Wirtualne połączenie jest ustanowione w momencie otrzymania pakietu zwrotnego CALL ACCEPT.
2. Od tego momentu obie stacje DTE mogą używać połączenia do dwukierunkowej transmisji danych (*full-duplex*).
3. W fazie trzeciej stacja DTE, która chce zerwać połączenie posyła do stacji partnerskiej pakiet CLEAR REQUEST. Druga odbiorcza posyła natomiast w odpowiedzi pakiet potwierdzający zgodę na zerwanie połączenia CLEAR CONFIRMATION.



Rys.7.3 Trzy fazy połączenia X.25

W celu jednoznacznej identyfikacji połączenia wirtualnego inicjalizująca komunikację stacja DTE może wybrać jedynie wolny (nie obciążony) numer wirtualnego połączenia. Jeśli numer połączenia wirtualnego jest używany przez stację docelową DTE, to docelowa stacja DCE musi go zastąpić numerem nie używanym.

Interfejs X.25 współpracuje z liniami przesyłającymi dane z szybkością do 64 kbit/s, aczkolwiek organizacja CCITT w 1992 roku udoskonaliła standard, zwiększając szybkość do 2 Mbit/s.

Wykorzystanie standardu X.25 zostało przewidziane także w połączeniach wg standardu OSI, lecz usługi podstawowe warstwy transportowej tzw. prymitywy muszą być przekształcane na usługi X.25. Przekształcenia te zostały ujęte w normie ISO 8878.

7.4 Sieć pakietowa FR

Ponieważ połączenia w standardzie X.25 cechują się małą wydajnością, dlatego też w większości przypadków nie nadają się one do komunikacji między dwoma sieciami LAN, która jest realizowana w czasie rzeczywistym. W takim przypadku zaleca się zastosowanie innych standardów takich jak sieć z przełączaniem pakietów FR (ang. *Frame Relay*). Podstawowym założeniem przy projektowaniu sieci FR była dobra jakość łączy transmisyjnych i niezawodność linii telefonicznych. Pakietowa sieć transmisyjna FR z przełączaniem ramek (pakietów) wykorzystuje łącza cyfrowe wysokiej jakości, zwykle światłowody, o małym współczynniku błędów i zmiennej przepływności w zakresie szybkości transmisji 56 kbit/s do 45 Mbit/s. W sieci FR wyróżnia się trzy typy połączeń wirtualnych (w sieci X.25 występują dwa):

1. Stałe połączenie wirtualne PVC (*Permanent Virtual Circuit*) - dostępne przez dłuższy czas co w praktyce upodabnia ten typ połączenia do linii dzierżawionej.
2. Przełączane połączenie wirtualne SVC (*Switched Virtual Circuit*) - zestawiane tylko na żądanie abonenta.
3. Grupowe połączenia wirtualne (*Multicast*) - pozwalające na dostęp wielu abonentów sieci do tych samych zasobów sieciowych.

Dane w sieci FR są przesyłane postaci ramek o zmiennej długości (zalecenie Q921). Ramka FR jest ograniczona dwiema flagami, pomiędzy którymi są umieszczone: nagłówek (*header*), dane właściwe i pole kontrolne FCS (*Frame Check Sequence*). W sieci FR wyeliminowano kontrolę błędów transmisji w węzłach pośredniczących w transmisji, przez co znacznie zwiększono efektywność przekazywania pakietów z danymi. Redukcja kontroli i retransmisji uszkodzonych ramek w każdym węzle pośredniczącym, skróciła czas ich komutacji poniżej 2 ms. W sieci FR za odrzucanie ramek nie kompletnych oraz za ich korektę i retransmisję odpowiadają wyłącznie stacje końcowe (przełączniki, komputer użytkownika).

W praktyce przemysłowej sieć FR może mieć zastosowanie do :

- przekazywania plików danych między stacjami graficznymi CAD/CAM a komputerową bazą danych poprzez sieć WAN,
- do przesyłania poczty elektronicznej pomiędzy oddalonymi oddziałami przedsiębiorstwa,
- pracy interaktywnej pomiędzy terminalami a komputerem centralnym przedsiębiorstwa (*mainframe*) itp.

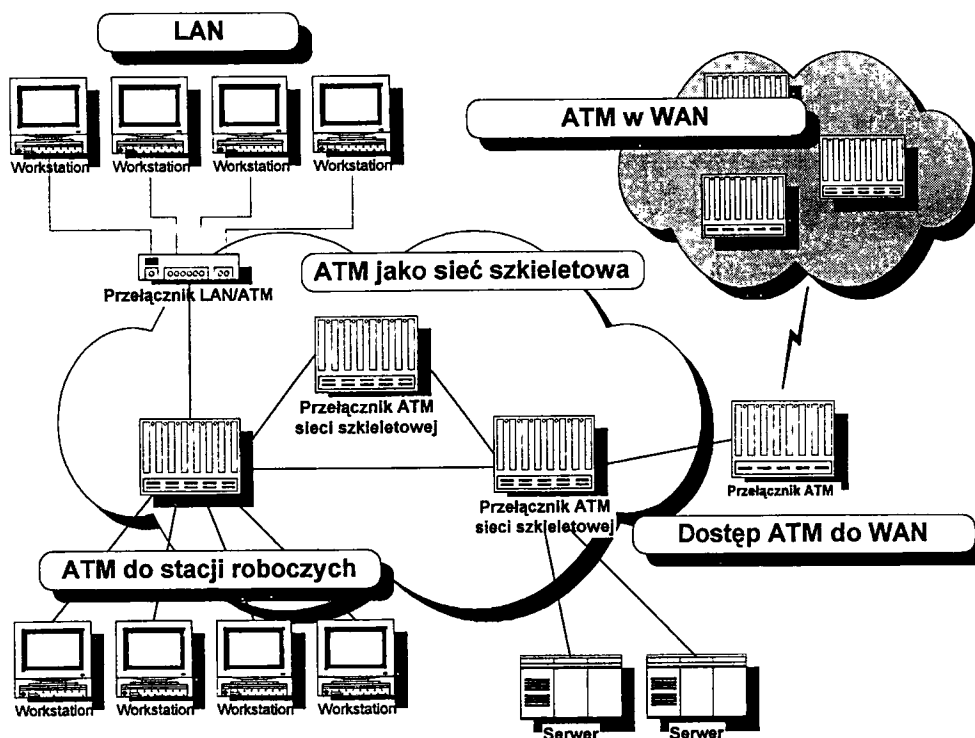
7.5 Sieć ATM

Jak się powszechnie uważa, przyszłość komunikacji w sieciach rozległych WAN będzie należała do sieci ATM (ang. *Asynchronous Transfer Mode*). ATM jest technologią o zintegrowanych usługach, uwzględniającą:

- nieograniczoną szerokość pasma na żądanie,
- dane, głos i wideo są przesyłane za pośrednictwem jednej, ekonomicznej infrastruktury,
- ciągłe łącza między sieciami LAN i WAN. Technologia ta może być stosowana zarówno w sieciach rozległych WAN jak i sieciach lokalnych LAN.

W zastosowaniach ATM wyróżniono cztery obszary rozwiązań (rys.8.4):

- lokalna sieć szkieletowa (ang. *local-area backbone*) - której zadaniem jest połączenie kilku sieci lokalnych LAN w jedną, zwartą strukturę logiczną. ATM eliminuje wąskie gardła w sieci szkieletowej oraz zapewnia dużą przepustowość wysokowydajnym serwerom. Do łączenia stacji roboczych pracujących w sieci LAN (np. Ethernet 10/100 Mbit/s) z siecią szkieletową ATM korzysta się z przełączników LAN, np. ATM 3Com LinkSwitch, wyposażonych z jednej strony w 12 portów Ethernet i 1 port ATM (155/622 Mbit/s) z drugiej strony.
- połączenia między stacjami roboczymi (ang. *desktop interconnect*) - realizowane za pomocą sieci ATM, mają na celu zwiększenie ich wydajności. Realizacja tego połączenia jest możliwa dzięki opracowaniu przez organizację IETF (*Internet Engineering Task Force*) specyfikacji LAN Emulation i Classical IP-over-ATM. Specyfikacje te zapewniają standardowe mechanizmy przechodzenia na przełączany ATM, nie eliminując dotychczasowych aplikacji czy oprogramowania sieciowego.



Rys.7.4 Obszary zastosowań sieci ATM

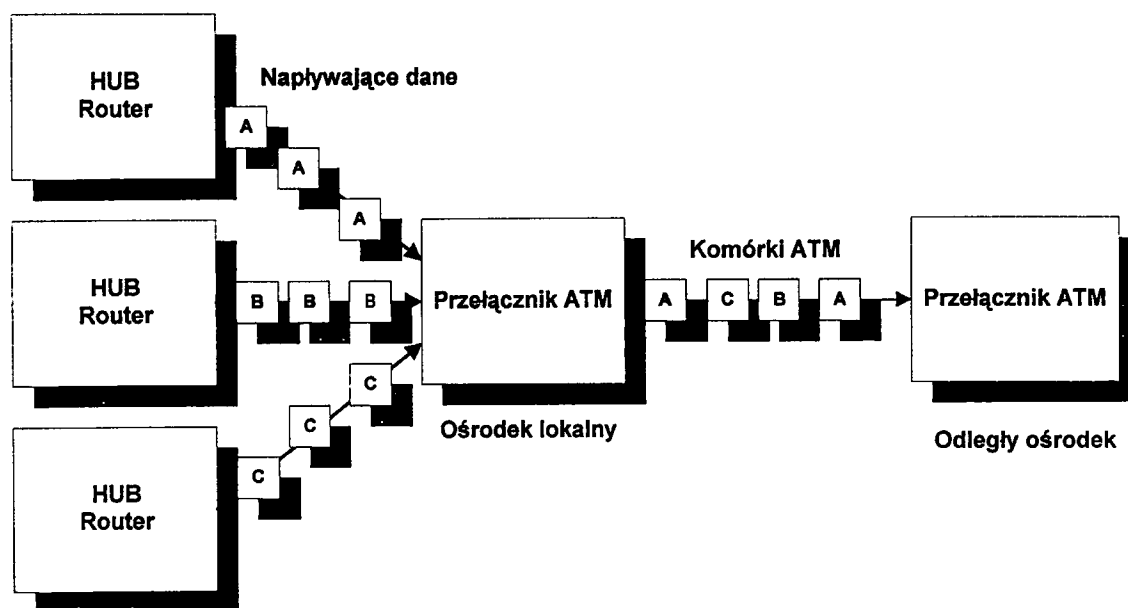
- dostęp do sieci rozległej (ang. *wide-area access*) - jest tematem prac organizacji ATM Forum. Organizacja ta zdefiniowała interfejs dla sieci rozległych, określany jako DS-1 UNI (1.5 Mbit/s) lub DS-3 (44 Mbit/s). Jednakże DS-3 jest rozwiązaniem drogim i wymaga zbyt szerokiego pasma dla większości sieci korporacyjnych. Obecnie są dostępne nowe specyfikacje ATM pozwalające na dostęp do sieci rozległych, przy ogólnodostępnych szybkościach (Japonia - specyfikacja UNI 6,3 Mbit/s, Europa - specyfikacja E1 2.0 Mbit/s). Specyfikacje te zostały już zaaprobowane przez producentów urządzeń sieciowych (przełączniki i multiplexery).
- transmisja w sieci rozległej (ang. *wide-area transport*) - ze względu na wysoki koszt sieci rozległych opartych na technologii ATM, pozostają one w dużej mierze w domenie operatorów publicznych. Aktualnie publiczne sieci ATM, podobnie jak sprzęt do ich realizacji, znajdują się w fazie rozwoju.

Sieć **ATM 155 Mbit/s** jest technologią pozwalającą na przesyłanie różnych rodzajów informacji takich jak: głos, obraz video oraz dane w czasie rzeczywistym. Rozwojem tego standardu zajmuje się organizacja ATM Forum. Technologia ta odpowiada wymaganiom zarówno sieci lokalnych LAN, jak i rozległych WAN. ATM wykorzystuje kable światłowodowe, pozwalające na osiągnięcie wysokich szybkości transmisji. Możliwa jest transmisja danych z szybkością od 155 Mbit/s do 622 Mbit/s. Tak duża szybkość jest możliwa dzięki pełnemu wykorzystaniu medium, jakim jest światłowód oraz techniki transmisji przyjętej w specyfikacji SONET (ang. *Synchronous Optical Network*). Sieci światłowodowe ATM budowane są głównie jako publiczne sieci telekomunikacyjne MAN, WAN, ale też nadają się one do wykorzystania w sieciach lokalnych LAN.

W technologii ATM zastosowano technikę komutacji pakietów, korzystając przy tym z rozszerzonego pasma transmisyjnego. Metoda ta nazywana jest często *cell relay* i charakteryzuje się stałą długością pakietów danych, tzw. komórek (techniki takie jak *frame relay* i IEEE 802.x korzystają natomiast ze zmiennej długości pakietów). Stały rozmiar komórek oraz zastosowanie multipleksacji (z wielokrotnienia) pozwalają na dynamiczne dopasowanie szerokości pasma do potrzeb aplikacji (rys.8.5). W przypadku zwiększenia aktywności jednej ze stacji przełączniki ATM automatycznie przydzielają odpowiednio większą liczbę komórek dla intensywniej napływającego strumienia danych.

ATM nie określa żadnego konkretnego medium transmisyjnego, pozwala więc na zastosowanie wielu różnych mediów stosowanych w innych systemach. Eksperci jednak zalecają wykorzystanie w charakterze fizycznego interfejsu sieci: FDDI (100Mbit/s), *Fiber Channel* (155 Mbit/s), OC3 SONET (155 Mbit/s) oraz T3 (45 Mbit/s).

Przewiduje się, że technologia ATM w nieodległej przyszłości rozpowszechni się zarówno w sieciach rozległych WAN, jak i w lokalnych LAN. Stanie się tak ponieważ takie firmy jak IBM inwestują rocznie setki milionów dolarów w rozwój produktów ATM.



Rys.7.5 Przełączanie komórek w ATM

Sieć ATM 25Mbit/s jest rozwiązaniem zaproponowanym jako tania wersja ATM. Zaprojektowano ją z myślą o komputerach PC i stacjach roboczych w sieciach LAN. Rozwiązanie to, jak się uważa powszechnie, nie ma jednak przyszłości ze względu na dużą konkurencję w zakresie przełączanego Ethernet'u, 100BaseX i 100VG-AnyLAN.

III. Systemy otwarte

9. Standardy systemów otwartych

Komunikacja w obszarze komputerowo zintegrowanego wytwarzania CIM odgrywa kluczową rolę. Wybór systemu komunikacyjnego określa w dużej mierze efektywność i niezawodność pracy w całym przedsiębiorstwie. Ponadto podczas prowadzenia działań zmierzających do informatycznego połączenia poszczególnych składników CIM istotne znaczenie mają związane z tym koszty.

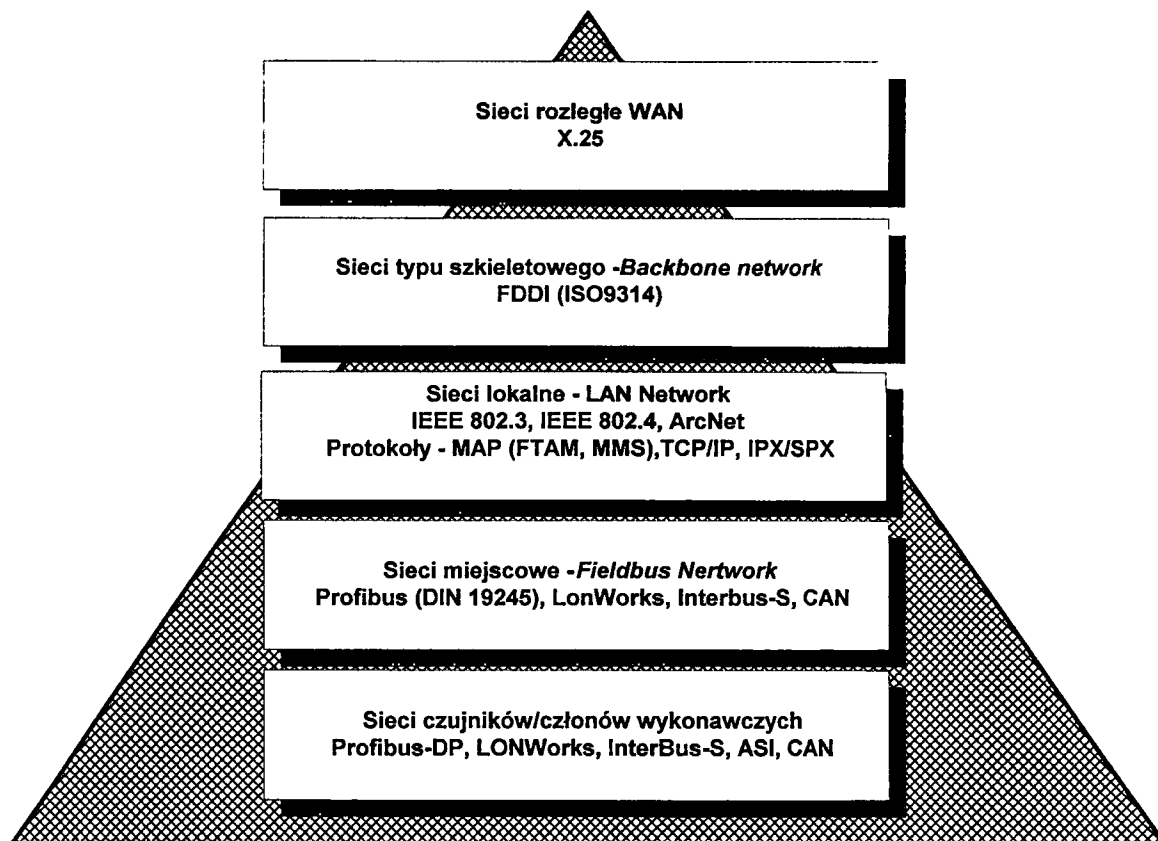
Ze względu na dużą ilość stosowanych w systemach CIM technik sprzętowych i programowych oraz różnorodnych wymagań w poszczególnych obszarach wytwarzania, powstaje konieczność zastosowania różnych, odpowiadających określonym wymaganiom sieci komunikacyjnych. I tak np. sieciom w obszarze komunikacji biurowej są stawiane inne wymagania aniżeli sieciom w obszarze wytwarzania.

Analizując rozwiązania różnorodnych systemów komunikacyjnych stosowanych w nowoczesnych przedsiębiorstwach w Europie Zachodniej, a także badając zapotrzebowanie na tego rodzaju techniki w przedsiębiorstwach krajowych, przyjęto hierarchiczny model przemysłowych, otwartych sieci komunikacyjnych w postaci piramidy (rys.3-1). Model ten dzieli sieci komunikacyjne przedsiębiorstwa na pięć podstawowych poziomów różniących się wymaganiami co do ilości przesyłanych danych oraz krytycznego, gwarantowanego czasu dostępu do danych systemowych. Przyjęto przy tym propozycje sprawdzonych już rozwiązań, proponowanych przez producentów sprzętu i oprogramowania dla zastosowań w przemyśle. Przykładem tego może być firma Siemens, która oferuje całą gamę produktów sieciowych pod nazwą SINEC, pod którą to nazwą kryją się także rozwiązania ogólnie przyjęte i ustandaryzowane przez gremia międzynarodowe. Model zaproponowany przez firmę Siemens uwzględnia jednak głównie interesy tej firmy i nie opisuje innych rozwiązań dotyczących otwartych sieci komunikacyjnych. Dlatego też został on uzupełniony o rozwiązania, które są równie dobre, a często nawet lepsze i nowocześniejsze.

Na rysunku 9.1 zamieszczono model otwartych sieci komunikacyjnych dla przedsiębiorstw produkcyjnych. Zamieszczone na tym rysunku propozycje te są kompromisem pomiędzy kosztami instalacji danego systemu, a możliwościami funkcjonalnymi, nowoczesnością i otwartością jego rozwiązania.

W zakresie rozległych sieci komputerowych WAN, spośród wielu nowoczesnych technik komunikacyjnych takich jak ATM czy ISDN, Frame Relay, X.25 i wielu innych należy wskazać na ostatnie rozwiązanie ze względu na istniejącą już w wielu przedsiębiorstwach infrastrukturę telekomunikacyjną oraz stosunkowo najniższe koszty instalacji i eksploatacji sieci.

Natomiast w obszarze sieci szkieletowych (ang. *Backbone*) uzasadnione jest proponowanie rozwiązania optymalnego ze względu na koszty instalacji, eksploatacji oraz zaawansowania technologicznego i otwartości. Sieć FDDI (ISO 9314) jest siecią doskonale nadającą się do łączenia poszczególnych jednostek i wydziałów produkcyjnych przedsiębiorstwa, zapewniając szybkość transmisji 100 Mbit/s. Aktualnie są wprowadzane proponowane nowsze rozwiązania, takie jak np. ATM, ale są jednak to rozwiązania, które mają szansę być upowszechnione dopiero za kilka lat.



Rys.9.1 Model warstwowy otwartych sieci komunikacyjnych przedsiębiorstwa

Sieci lokalne LAN są technologią znaną w naszym kraju już od wielu lat. Powszechnie przyjęły się rozwiązania oparte na standardzie Ethernet (IEEE 802.3) oraz *Token Ring* (IEEE 802.5), ale w obszarze wytwarzania w normach międzynarodowych przewidziano rozwiązanie sieci LAN oparte na standardzie IEEE 802.4 (*Token Bus*).

Ze względu na otwartość protokołów opartych na modelu ISO/OSI, protokoły te znajdują zastosowanie w budowie sieci komunikacyjnej wg rozwiązania MAP 3.0 z protokołami MMS i FTAM. Protokoły te są dostosowane do pracy w środowiskach przemysłowych i uwzględniają różnorodne platformy sprzętowo-programowe przedsiębiorstw. Protokoły te mają także szansę stać się w niedługim czasie powszechne w zastosowaniach w dużych przedsiębiorstwach produkcyjnych, a zwłaszcza w sferze wytwarzania. Jest to możliwe ze względu na duże poparcie agencji rządowych oraz stale malejących kosztów instalacji i eksploatacji tych systemów.

Biorąc pod uwagę ilość istniejących instalacji systemów z protokołami TCP/IP oraz IPX/SPX, należy je również brać pod uwagę. Należy także dodać, że protokół TCP/IP nie był wprawdzie projektowany dla zastosowań w przemyśle, ale okazało się, że spełnia on doskonale wymagania warunków pracy w przemyśle.

Sieci miejscowe typu *Fieldbus* oraz sieci czujników / członów wykonawczych są stosowane na najniższych poziomach wymiany danych w zautomatyzowanym wytwarzaniu. Na świecie jest obecnie wiele systemów spełniających wymagania stawiane w tym obszarze. Warto w tym miejscu dodać, że obserwuje się dynamiczny rozwój tych systemów oraz rosnące nimi zainteresowanie. Firmy światowe organizują się w stowarzyszenia, które mają na celu rozpowszechnianie i rozwój

tych standardów. Do takich standardów należą: Profibus, InterBus-S, LONWorks, ASI oraz CAN. Na rynku jest również coraz więcej urządzeń wyposażonych w interfejsy zgodne z wyżej wymienionymi standardami. Dzieje się tak z powodów organizowania się firm we wspomniane już stowarzyszenia użytkowników.

10. Protokoły warstwy transportowej

10.1 Protokół TCP/IP

Protokoły TCP/IP używane są przez dziesiątki tysięcy sieci i łączą ze sobą miliony komputerów. Największym takim zbiorem współpracujących razem systemów jest Internet, tzn. sieć, która rozwija się bardzo intensywnie i od roku 1988 systematycznie corocznie podwaja swój rozmiar. Globalna społeczność sieciowa nie ma swojej centralnej władzy. Sieć Internet działa na podstawie uzgodnień dotyczących stosowanych w niej protokołów sieciowych oraz współpracujących w niej użytkowników. Do sieci Internet przyłączane są coraz to nowe lokalne sieci komputerowe związane z prywatnymi i publicznymi organizacjami.

Podstawowymi pojęciami, przy pomocy których charakteryzuje się architekturę sieci są usługi i protokoły. Usługi dotyczą funkcji realizowanych przez warstwę oprogramowania i przekazywane są pomiędzy warstwami pojedynczego systemu za pomocą elementarnych jednostek usługowych (prymitywów), natomiast protokoły umożliwiają przekazywanie tych usług poprzez wzajemną komunikację odległych systemów, jako wymianę danych między partnerskimi warstwami. W takim rozumieniu protokoły stanowią implementację usług. Zestaw protokołów wypełniających kolejne warstwy modelu odniesienia ISO/OSI nosi nazwę stosu protokolowego.

10.2 Stos protokolowy TCP/IP

Stos protokolowy sieci wykorzystujących zestaw protokołów TCP/IP nie stanowi dokładnego odwzorowania siedmiu warstw modelu ISO. Jest to wynikiem tego, że model ISO/OSI został ustanowiony w czasie, gdy funkcjonowała już i zdobywała coraz większą popularność architektura TCP/IP. Stos protokolowy TCP/IP w odniesieniu do modelu OSI przedstawia rys.10.1.

aplikacyjna	FTP, Telnet, DNS, SMTP, WWW, RIP, inne		OSP F
prezentacji			
sesji			
transportowa	TCP	UDP	
sieciowa	ICMP	IP	
			ARP/RARP
liniowa	technologie sieciowe:		
fizyczna	Ethernet, Token Ring, FDDI, ATM, X.25		

Rys.10.1 Stos protokolowy TCP/IP

Najistotniejszą różnicą w obu tych modelach jest brak warstw sesji i prezentacji w TCP/IP. Nie oznacza to jednak, że funkcje tych warstw nie są realizowane, natomiast brak jest tylko wyraźnego

rozgraniczenia ich implementacji pomiędzy trzema najwyższymi warstwami modelu ISO. Tak więc praktycznie, w modelu TCP/IP warstwa aplikacji usytuowana jest nad warstwą transportową.

Warstwa fizyczna oraz warstwa połączeń liniowych związane są ze stosowanymi technologiami sieciowymi. Akceptowane są wszystkie dostępne protokoły tych poziomów, jakie są wykorzystywane w sieciach lokalnych lub rozległych.

Najważniejszymi w całym zestawie protokołów z rys. 1 są protokoły IP (*Internet Protocol*) oraz TCP (*Transmission Control Protocol*), od których pochodzi nazwa architektury internetowej.

10.3 Warstwa sieciowa

Ponieważ protokół warstwy sieciowej IP jest podstawowym protokołem komunikacyjnym w zestawie protokołów TCP/IP, dlatego też musi więc być implementowany w każdym hoście i routerze. IP dostarcza usług datagramowych, czyli bezpołączeniowych, których bezspornymi zaletami są elastyczność w przesyłaniu jednostki danych oraz odporność na uszkodzenia w sieci. Z drugiej jednak strony jest to usługa zawodna, co znaczy, że protokół nie gwarantuje przekazania odbiorcy danych w założonym czasie i właściwej sekwencji oraz dopuszcza niedostarczenie komunikatu do miejsca przeznaczenia. Nie znaczy to, że protokół IP pozbawiony jest jakichkolwiek mechanizmów kontroli błędów transmisji. W przypadku wystąpienia nieprawidłowości w transmisji (np. błędny adres lub brak buforów), datagram zostaje zniszczony z powiadomieniem o tym nadawcy (poprzez ICMP). Odtwarzanie po błędzie należy już do TCP. Protokół IP obsługuje każdy datagram jako indywidualną jednostkę, niezależną od innych datagramów.

Przenoszony przez sieć datagram identyfikowany jest nazwą, adresem oraz trasą dojścia do miejsca przeznaczenia. Wybór drogi przesyłania wspomagają statyczne lub dynamiczne tablice, znajdujące się w każdym hoście i routerze.

Popularność sieci Internet z jednej strony potwierdziła trafność koncepcji protokołu IP, a z drugiej stworzyła nowe problemy, których nie przewidzieli twórcy wersji podstawowej. Ograniczenia płynące ze sposobu adresowania stacji internetowych oraz rozmiary tablic routingowych okazały się czynnikiem wymuszającym modyfikacje istniejącego standardu *de facto*. Z drugiej strony, również naturalny rozwój sieci komputerowych wymusza akceptację w niej nowych technologii oraz ciągle poszerzanie jej zastosowań. Wszystko to spowodowało konieczność zmian w trzonie systemu internetowego, jakim jest protokół IP. Powstała więc nowa wersja protokołu IP znana pod nazwą IPv6.

Szósta wersja IP ma już niedługo całkowicie wyprzeć funkcjonującą obecnie z powodzeniem opisaną wyżej wersję czwartą protokołu (IPv4). IPv6 nie jest w swoim zamierzeniu nowym protokołem zastępującym dotychczas działający IP, lecz jest to tylko nowa wersja, zachowująca funkcje swojego poprzednika, wprowadzając równocześnie w sposób łagodny zmiany, tak aby obie wersje współpracowały w sieci bezkolizyjnie i nie stwarzały zagrożeń dla istniejących już rozwiązań.

Istotną poprawę w działaniu i architekturze sieci dają zmiany w stosunku do wersji czwartej takie jak:

- możliwość szybszego przetwarzania w kolejnym węźle przez uproszczenie formatu datagramu,
- rozszerzenie zakresu adresacji,
- wprowadzenie możliwości wyboru jakości usług,
- wprowadzenie mechanizmów bezpieczeństwa transmisji.

Zmiany te w dużym stopniu otwierają drogę dla rozwoju nowych aplikacji sieciowych.

10.4 Warstwa transportowa

W zestawie protokołowym TCP/IP nad zawodnym protokołem IP znajduje się niezawodny protokół TCP. Protokół ten daje możliwość utworzenia wirtualnego połączenia pomiędzy dwoma aplikacjami i przesyłania strumienia bajtów z *host'a* źródłowego do *host'a* przeznaczenia w dokładnie takiej kolejności, w jakiej były generowane przez aplikację w hoście źródłowym.

TCP jest protokołem pracującym w trybie połączeniowym o wysokim poziomie niezawodności oraz charakteryzuje się on dużą niezależnością od współpracującej z nim warstwy niższej. Wynikiem tego jest jego wysoka odporność na uszkodzenia sieci i systemu lokalnego, a tym samym umożliwienie niezawodnej wymiany danych między dwoma procesami aplikacyjnymi, rezydującymi w odległych *host'ach* środowiska otwartego. Proces w obrębie *host'a* jest identyfikowany przez port, który w połączeniu z międzysieciowym adresem *host'a* (dostarczonym przez IP) tworzy jednoznacznie adresowalny obiekt w całej sieci globalnej zwany gniazdkiem (*socket*). Para gniazdek tworzy połączenie, w którym procesy wymieniają w trybie pełnodupleksowym (w obu kierunkach równocześnie) seryjnie przesyłane dane. Gniazdko może obsługiwać więcej niż jeden proces, ponieważ TCP łączy procesy, a nie identyfikatory gniazdek.

Usługi TCP wyspecyfikowane są w postaci ogólnej i określają tylko podstawowe funkcje, jakie są konieczne do realizacji komunikacji między procesami. Pozostawia się swobodę w implementacji szczegółowego zbioru usług oraz ich formatu.

Użytkownik TCP (proces aplikacyjny) wysyła komendę żądania usług do swojej lokalnej jednostki TCP, a jednostka ta odpowiedzialna jest za kontakt ze swoim odległym partnerem, zrealizowanie żądanej usługi i przekazanie wyników swojemu lokalnemu użytkownikowi.

Drugim istotnym protokołem warstwy transportowej jest UDP (*User Datagram Protocol*), który jest prostym, zorientowanym datagramowo protokołem transportowym i stanowi rozszerzenie protokołu IP w kierunku minimalnych usług transportowych. Różni się on od strumieniowo zorientowanego protokołu TCP, gdzie liczba danych produkowanych przez aplikację nie ma nic wspólnego z zawartością datagramu IP. Podobnie jak IP, UDP nie daje gwarancji dostarczenia datagramu do miejsca przeznaczenia, ponieważ do warstwy sieciowej przekazuje on tylko dane od aplikacji. Stąd też protokół UDP używany jest tam, gdzie nie wymaga się poziomu usług oferowanego przez TCP, lub też gdy istnieje zapotrzebowanie na takie usługi, jakich TCP nie oferuje, takie jak np. komunikacja realizowana w trybie rozgłoszeniowym (*broadcasting*) i grupowym (*multicasting*),.

10.5 Protokoły wspomagające komunikację

Poza omówionymi podstawowymi funkcjami, jakie realizują protokoły internetowe IP i TCP, działanie sieci wymaga jeszcze realizacji szeregu funkcji pomocniczych. Funkcje te wykonywane są przez szereg protokołów pomocniczych. Zaliczyć do nich można protokoły wspomagające adresację ARP i RARP, obsługę błędów (ICMP) i ustalanie drogi (OSPF, RIP). Te ostatnie, aczkolwiek usytuowane są na poziomie aplikacji (rys.10.1), pełnią zasadniczą rolę w podejmowaniu decyzji odnośnie dalszej drogi datagramu w każdym węźle.

10.6 Aplikacje

Protokoły warstwy aplikacji wypełniają, w odniesieniu do modelu ISO, trzy wyższe warstwy i są bezpośrednio "widoczne" przez użytkownika sieci. Pozwalają one na dostęp do zdalnego terminala (*Telnet*), sprowadzanie lub wysyłanie plików (*FTP*), komunikację przez pocztę elektroniczną (*SMTP*), znalezienie adresu zawierającego np. żadaną usługę (*DNS*) oraz realizację najpopularniejszej usługi, jaką jest uzyskanie dostępu do każdej zarejestrowanej w sieci informacji lub bazy danych (*WWW*).

10.7 Ruch w czasie rzeczywistym

Wykorzystanie sieci bazujących na tak przedstawionym stosie protokołowym TCP/IP do aplikacji działających w czasie rzeczywistym nie jest możliwe. Czas przeznaczony na wymianę wiadomości dostarcza sporego narzutu komunikacyjnego, który jest nie do zaakceptowania w przypadku systemów ostro uwarunkowanych czasem. Dla aplikacji o łagodniejszych wymaganiach względem czasu, jakimi są np. systemy multimedialne, niewielkie opóźnienia w transmisji są dopuszczalne, jednak należy liczyć się z pogorszeniem jakości takiej usługi. Dlatego też architekturze sieci dla transmisji odbywającej się w czasie rzeczywistym stawia się inne wymagania niż w przypadku przesyłania zwykłych danych, nie uzależnionych od czasu. W systemach przesyłania danych tradycyjnych podstawową miarą wydajności transmisji jest przepustowość systemu, to znaczy liczba danych, jaka może być przesłana przez sieć od ich źródła do miejsca przeznaczenia w jednostce czasu. W systemach czasu rzeczywistego podstawową miarą jest natomiast prawdopodobieństwo dostarczenia wiadomości w określonym czasie. Tak przyjęta miara uwzględnia dwa czynniki: szybkość transmisji oraz prawdopodobieństwo zagubienia wiadomości (niedostarczenie wiadomości traktowane jest jako nieskończony czas jej transmisji). Dlatego też podstawowe wymagania stawiane sieciom dla transmisji w czasie rzeczywistym to odpowiednia szerokość pasma transmisji oraz minimalne opóźnienia.

Istnieje więc konieczność utworzenia nowej architektury, która nie burząc dotychczasowej konwencji internetowej może być lepiej dostosowana do potrzeb nowych użytkowników.

Architektura przystosowana do przenoszenia ruchu w czasie rzeczywistym mniej rygorystycznie traktuje modelową warstwowość na korzyść większej elastyczności we współpracy protokołów. Nie wszystkie aplikacje muszą wykorzystywać pełny zestaw usług oferowany przez

warstwy niższe. Do tej grupy należą właśnie aplikacje czasu rzeczywistego. Złożoność transportowego protokołu TCP, która stanowi dość istotne źródło opóźnienia transmisji może być do pokonania, jeśli z warstwy transportowej wyłoni się tylko te funkcje, które są konieczne dla aplikacji, pomijając inne. Aplikacje przesyłające głos, np. dość dobrze tolerują zagubienie danych, natomiast nie mogą funkcjonować przy dużych opóźnieniach. W protokołach transportowych błędy wynikłe z zagubienia danych (brak sekwencyjności w odbiorze jednostek danych protokołowych) koryguje się retransmisjami, które opóźniają dostęp danych do aplikacji. Aplikacja, która jest mało wrażliwa na niezgodność w sekwencji danych może sama obsłużyć tego rodzaju błąd, często przez zignorowanie sytuacji. Takiej możliwości nie dostarcza tradycyjna forma warstwy transportowej, która zgodnie ze swoim zadaniem nie może dostarczyć aplikacji segmentu danych, który nie będzie w sekwencji z pozostałymi.

Podstawą architektury dla ruchu rzeczywistego jest szkielet protokołu, który ma zdefiniowane tylko podstawowe funkcje, operacje i formaty wiadomości. Częścią takiej architektury jest protokół czasu rzeczywistego (RTP - *Real Time Protocol*). Zasadą jest tu, że kompletny protokół tworzy się przez dodawanie przez aplikację do szkieletu elementów specyficznych dla aplikacji. Funkcje RTP zostały uzupełnione przez brakujące w nim funkcje obsługi błędów i sterowanie, które jest realizowane przez ściśle z nim współpracujący protokół RTCP (*Real Time Control Protocol*).

Współpracujące ze sobą protokoły RTP i RTCP służą do wymiany danych, dając przy tym użytkownikom pełną kontrolę nad aplikacją. Pozwalają one na dynamiczne dopasowanie szybkości transmisji do zmieniającego się natężenia ruchu w sieci. Nie daje to jednak gwarancji terminowego dostarczenia danych do odbiorcy, a długie przerwy w transmisji pogarszają jej jakość. Mechanizmem mającym kluczowy wpływ na uzyskanie transmisji w czasie rzeczywistym o wysokiej jakości usług jest rezerwacja zasobów implementowana za pomocą protokołu RSVP (*Resource Reservation Protocol*). Funkcją jego jest utworzenie i utrzymanie ścieżki przepływu między źródłem danych i ich przeznaczeniem, przez co zostanie zarezerwowane pasmo sieciowe dla danego przepływu. Daje to gwarancję przydziału zasobów (szerokość pasma, bufor pamięci) dla takiej transmisji.

Protokoły RTP, RTCP oraz RSVP spełniają częściowo funkcję transportową, częściowo aplikacyjną, jednak zostaje zachowana podstawowa koncepcja architektury TCP/IP, a między IP i RTP działa UDP, realizując wszystkie swoje funkcje. RTP, RTCP oraz RSVP mogą więc działać niezależnie od implementowanej wersji protokołu IP (czwartej lub szóstej).

10.8 Protokoły warstwy transportowej systemu MAP 3.0

Podstawową funkcją warstwy transportowej jest aktywacja, utrzymywanie i deaktywacja połączenia transportowego między dwoma komunikującymi się systemami otwartymi. W szczególności funkcje warstwy transportowej można podzielić na następujące części:

- przesyłanie jednostek danych protokołu transportowego (oznaczonych zwyczajowo jako TPDU, ang. *Transport Protocol Data Units*),

- multipleksowanie wielu połączeń transportowych w jedno połączenie sieciowe (zwykle z powodów minimalizacji kosztów przesyłania danych),
- realizacja pojedynczego połączenia transportowego na bazie wielu połączeń sieciowych (zwykle z powodów wydajnościowych),
- wychodzenia ze stanów błędów i/lub utraty połączeń na poziomie sieciowym,
- konkatenacji i segmentacji TPDU przed ich dostarczeniem do przesłania do warstwy sieciowej i separacji oraz składania TPDU w odbiorczej stacji transportowej,
- przesyłania danych przyspieszonych (zwykle niewielkich porcji danych o priorytecie wyższym niż dane „zwykłe”).

W dokumencie standaryzacyjnym ISO 8073 opisano połączeniowy protokół transportowy. W protokole tym wyróżniono 5 klas implementacji:

- klasa 0 - klasa prosta,
- klasa 1 - klasa z odtwarzaniem po błędach,
- klasa 2 - klasa z multipleksowaniem,
- klasa 3 - klasa z odtwarzaniem po błędach i multipleksowaniem,
- klasa 4 - klasa z detekcją błędów, odtwarzaniem po błędach, multipleksowaniem etc.

W modelu MAP/TOP zastosowano protokoły warstwy transportowej klasy 4, który ma: możliwość sterowania przepływem danych, multipleksowania transmisji użytkowników, detekcję błędów oraz dysponuje mechanizmami wydobywania się z błędów poprzez kontrolę sekwencyjności, zagubień oraz uszkodzeń pakietów.

Specyfikacja MAP dopuszcza jako opcję stosowanie protokołu ISO bezpołączeniowego, ale może to być traktowane jako usługa dodatkowa, a nie podstawowa.

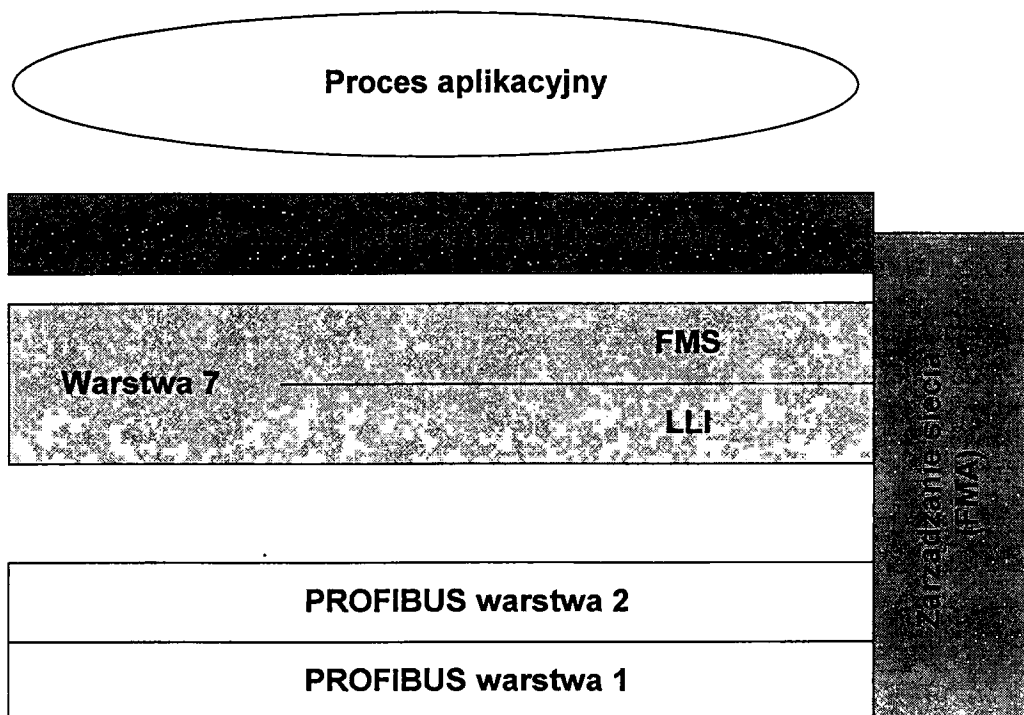
11. Protokoły warstwy aplikacyjnej systemów typu *Fieldbus* - FMS i PMS

11.1 PROFIBUS-FMS

PROFIBUS-FMS jest przeznaczony do realizacji połączeń komunikacyjnych na poziomie gniazd produkcyjnych. W tym obszarze komunikacja przebiega pomiędzy programowalnymi sterownikami PLC, komputerami PC oraz inteligentnymi urządzeniami automatyki. Obszar ten charakteryzuje się wyższym stopniem funkcjonalności, przy mniejszych wymaganiach czasowych.

11.2 Warstwa aplikacyjna PROFIBUS-FMS

Model systemu sieciowego PROFIBUS-FMS opiera się na siedmiowarstwowym modelu ISO/OSI. Jednak aby system ten mógł pracować w aplikacjach o określonych rygorach czasowych, zredukowano model siedmiowarstwowy do trzech warstw (1, 2 oraz 7) (rys.11.1).



Rys.11.1 Struktura warstwowa PROFIBUS'a-FMS

Warstwa połączenia aplikacyjnego ALI (ang. *Application Layer Interface*). Warstwa ta przekształca lokalne obiekty procesowe z aplikacji w standardowe obiekty komunikacyjne warstwy siódmej FMS. Odpowiada ona także za proces odwrotny. W module tym użytkownik podczas etapu konfiguracji sieci definiuje połączenia komunikacyjne oraz obiekty komunikacyjne.

Fieldbus Message Specification (FMS). Podwarstwa ta ma dwie podstawowe funkcje: *generowanie paczek danych w protokole oraz ich kodowanie, dekodowanie oraz interpretacja paczek danych protokołu.*

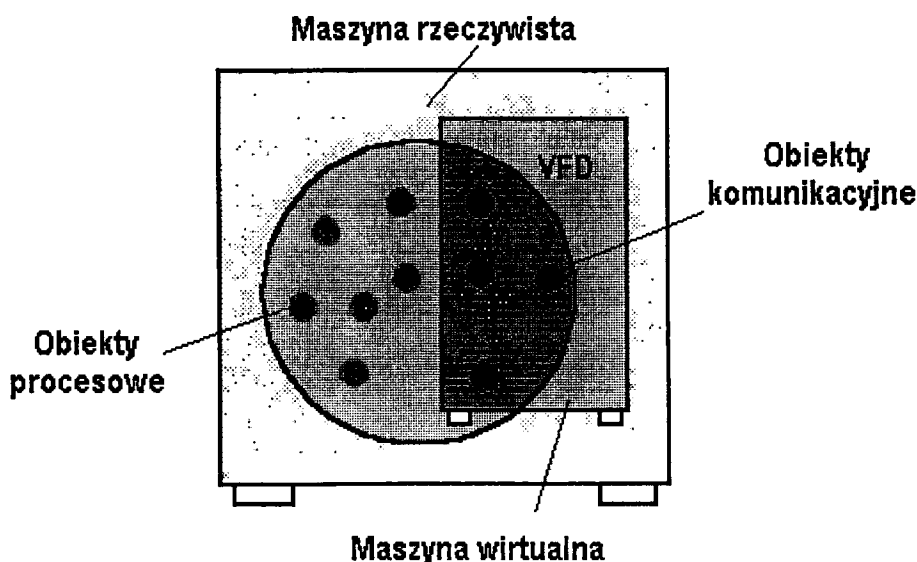
Lower Layer Interface (LLI). W tej części usługi warstwy siódmej są przekształcane na usługi warstwy drugiej.

Zarządzanie siecią FMA (ang. *Fieldbus Management Application*). Ta część obejmuje wszystkie warstwy modelu PROFIBUS. Udostępniane są w niej użytkownikowi funkcje zarządzania siecią.

11.3 Model komunikacyjny PROFIBUS-FMS

Za pomocą modelu komunikacyjnego FMS jest możliwe takie dopasowanie programów aplikacyjnych i zgrupowanie obiektów procesowych (zmiennych, programów itp.) aby odzwierciedlały one rzeczywisty proces technologiczny.

Aby system komunikacyjny miał dostęp do obiektów procesowych przekształca się je na obiekty komunikacyjne. Odbywa się to przez zdefiniowanie każdego obiektu w tzw. *słowniku obiektów*. Każdy *słownik obiektów* wraz z zebranymi w nim obiektami komunikacyjnymi tworzy odwzorowanie urządzenia widzianego teraz w sieci jako *urządzenie wirtualne* (ang. *Virtual Field Device* - VFD).



Rys.11.2 Urządzenia wirtualne (VFD)

VFD jest abstrakcyjnym modelem, który opisuje dane oraz zachowanie rzeczywistego urządzenia lub systemu (rys. 12.2).

11.4 Elementy komunikacyjne warstwy siódmej

Z punktu widzenia użytkownika (aplikacji) i jego funkcji, komunikacja realizowana jest pomiędzy dwoma stacjami. Jedna stacja jest *client*'em (wysyłająca żądania) a druga *server*'em (odpowiadająca na żądania *client*'a). I tak np. *client*'em może być sterownik, a *server*'em inteligentne czujniki lub elementy wykonawcze.

Do realizacji komunikacji pomiędzy *client*'em i *server*'em siódma warstwa PROFIBUS'a wykorzystuje trzy elementy komunikacyjne:

- relacje komunikacyjne,
- obiekty komunikacyjne oraz

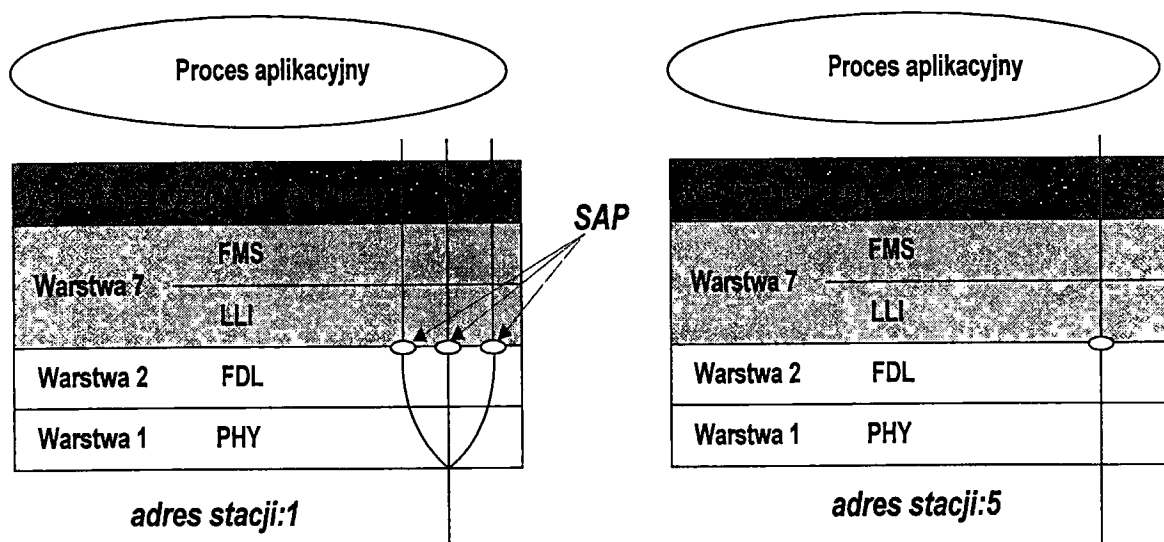
- usługi komunikacyjne.

Relacja komunikacyjna definiuje logiczne połączenie pomiędzy dwoma stacjami. Wszystkie *relacje komunikacyjne* zebrane są w jednej liście, tzw. *liście relacji komunikacyjnych* lub *liście połączeń CRL* (ang. *Communication Relationship List*).

Obiektem komunikacyjnym może być: dana pomiarowa, część programu lub parametry, które są wymieniane pomiędzy stacjami. Obiekty te są dostępne dla *usług komunikacyjnych* takich jak np. *Read* oraz *Write*.

11.5 Lista relacji komunikacyjnych

Komunikacja z procesem w innej stacji odbywa się poprzez logiczne kanały transmisji danych. Kanały te są definiowane w fazie konfigurowania sieci, a ich opis umieszczony jest w *liście referencji komunikacyjnych CRL* (ang. *Communication Relationship List*). Każda stacja zawiera listę referencji opisującą jej połączenia logiczne z innymi stacjami komunikacyjnymi.



Rys.11.3 Adresy stacji oraz porty dostępu do usług

Każde połączenie komunikacyjne jest opisane w formie parametrycznej w referencji komunikacyjnej. Najważniejsze parametry to:

- SSAP - jest to numer portu łączącego warstwę drugą z siódmą, poprzez który usługa ma dostęp do procesu aplikacyjnego stacji lokalnej,
- typ połączenia,
- atrybuty połączenia,
- RADR - adres stacji dołączonej,
- RSAP - to numer portu łączącego warstwę drugą z siódmą, poprzez który usługa ma dostęp do procesu aplikacyjnego stacji dołączonej,
- długość ramki (*PDU length*),

- wspierane usługi,
- inne.

Parametry te pozwalają stacji na jednoznaczne odnalezienie partnerów komunikacyjnych w sieci oraz nawiązanie z nimi połączenia odpowiedniego typu.

11.5.1 Adresy i porty komunikacyjne

Każda stacja musi mieć nadany unikatowy adres aby była możliwość jednoznacznego odwołania się do danego węzła w sieci. Poza zwykłym adresowaniem stacji, PROFIBUS-FMS ma porty – SAP (ang. *Service Access Point*), dzięki którym stacja może nawiązać wiele logicznych połączeń z innymi węzłami sieci (rys.11.3). Każde z tych połączeń może charakteryzować się innymi parametrami.

11.5.2. Typy połączeń

Innym parametrem definiowanym w liście referencji komunikacyjnych jest określenie typu połączenia. W zależności od rodzaju stacji oraz sposobu przesyłania danych, PROFIBUS-FMS rozróżnia następujące typy połączeń:

- MSAC – połączenie *master - slave* acykliczne,
- MMAZ – połączenie *master - master* acykliczne,
- MSAC_SI – połączenie *master – slave* acykliczne ze *slave*’em z inicjatywą,
- MSCY – połączenie *master – slave* cykliczne,
- MSCY_SI – połączenie *master – slave* cykliczne ze *slave*’em z inicjatywą,
- MULT – *multicast*,
- BRCT – *broadcast*.

Jednak nie wszystkie stacje wspierają połączenia typu MMAC_SI oraz MMCY_SI.

11.5.3 Atrybuty połączeń

Obok omówionych wcześniej typów relacji komunikacyjnych, w PROFIBUS’ie-FMS rozróżniane są dwa dodatkowe typy:

- **zdefiniowane** (w referencji oznaczane literą „D”) oraz
- **otwarte** (dwa typy oznaczane w referencji literą „O” oraz „I”).

Połączenie zdefiniowane jako „D” jest sparametryzowane przez użytkownika i jest ono ustalone na cały czas działania systemu komunikacyjnego. Połączenia otwarte nie wymagają definiowania adresu oraz portu stacji dołączonej. Połączenie typu „I” jest inicjowane podczas nawiązania komunikacji. Parametry połączenia są ustalane przez stację odpowiadającą (*server*’a). Połączenie otwarte „O” jest inicjowane również podczas nawiązania komunikacji. Jednak parametry połączenia są definiowane przez stację żądającą (*client*’a). Oba połączenia otwarte istnieją tylko na czas przesyłania danych. Po zakończeniu mogą być one udostępnione innym stacjom.

Typowym zastosowaniem połączenia otwartego typu „O” jest konfiguracja większości połączeń ze stacjami pasywnymi (*slave*’ami). Dzięki zdefiniowaniu połączenia otwartego konfiguracja *slave*’a ogranicza się tylko do nadania mu adresu.

11.5.4. Kodowanie wspieranych usług

Jednym z parametrów połączenia komunikacyjnego deklarowanym w liście referencji komunikacyjnych są usługi FMS wspierane przez obie stacje (*client*’a oraz *server*’a). Aby można było jednoznacznie zadeklarować wspierane przez daną stację funkcje, podzielono je na trzy sekcje osobno dla *client*’a oraz *server*’a (tab.12.1), a każdej funkcji przypisano odpowiedni kod (tab.12.2).

Tab.11.1 Funkcje *Client*’a

<i>Client</i>			<i>Server</i>		
FMS0	FMS1	FMS2	FMS3	FMS4	FMS5
**	**	**	**	**	**

** w miejsce gwiazdek wpisywane są kody usług.

Tab.11.2 Kody funkcji

Usługa	<i>Client</i>	<i>Server</i>	Wartość kodu
<i>GetOD</i>	FMS0	FMS3	80
<i>UnsolicitedStatus</i>	FMS0	FMS3	40
<i>PutOD</i>	FMS0	FMS3	20
<i>DomainDownload</i>	FMS0	FMS3	10
<i>DomainUpload</i>	FMS0	FMS3	08
<i>RequestDomainDownload</i>	FMS0	FMS3	04
<i>RequestDomainUpload</i>	FMS0	FMS3	02
<i>Create, DeleteProgramInvocation</i>	FMS0	FMS3	01
<i>Start, Stop, Resume, Reset</i>	FMS1	FMS4	80
<i>Kill</i>	FMS1	FMS4	40
<i>Read</i>	FMS1	FMS4	20
<i>Write</i>	FMS1	FMS4	10
<i>ReadWithType</i>	FMS1	FMS4	08
<i>WriteWithType</i>	FMS1	FMS4	04
<i>PhysRead</i>	FMS1	FMS4	02
<i>PhysWrite</i>	FMS1	FMS4	01
<i>InformationReport</i>	FMS2	FMS5	80
<i>InformationReportWithType</i>	FMS2	FMS5	40
<i>Define, DeleteVariableList</i>	FMS2	FMS5	20
<i>EventNotification</i>	FMS2	FMS5	10
<i>EventNotificationWithType</i>	FMS2	FMS5	08
<i>AcknowledgeEventNotification</i>	FMS2	FMS5	04
<i>AlterEventConditionMon.</i>	FMS2	FMS5	02
<i>Addres with Name</i>	FMS2	FMS5	01

11.6. Słownik obiektów

PROFIBUS-FMS jest zorientowany na działania z obiektami. Aby była możliwość komunikacji, obiekty komunikacyjne oraz ich struktura muszą stać się znane dla systemu komunikacyjnego. Umożliwia to *słownik obiektów OD* (ang. *Object Dictionary*). Opis obiektów jest definiowany w stacji, w której one istnieją (źródłowy słownik obiektów). Pozostałe stacje mają całkowitą lub częściową kopię tego samego opisu obiektów (słownik obiektów stacji dołączonej). W ten sposób każda ze stacji posiada źródłowy słownik obiektów, definiujący lokalne obiekty oraz jeden lub więcej słowników obiektów stacji dołączonych.

PROFIBUS-FMS posiada specjalne usługi do pozyskiwania i modyfikacji poprzez sieć źródłowych słowników obiektów. Każdy słownik obiektów ma określoną i niezmienną strukturę. Podzielony jest on na dwie główne części:

- część opisująca obiekty statyczne wraz z nagłówkiem oraz
- część opisująca obiekty dynamiczne.

Struktura obiektów statycznych jest definiowana przez użytkownika, jest ona tworzona w fazie inicjacji systemu i nie podlega ona zmianom w czasie przebiegu procesu. W odróżnieniu od obiektów statycznych, struktura obiektów dynamicznych może podlegać zmianom lub może zostać skasowana w czasie działania systemu.

11.6.1. Typy zmiennych

Do jednoznacznego zdefiniowania zmiennych udostępnianych w sieci, PROFIBUS-FMF używa następujących typów zmiennych:

- *boolean*,
- *integer* (8 bit, 16 bit oraz 32 bit),
- *unsigned* (8 bit, 16 bit oraz 32 bit),
- *floating point*,
- *octet string*,
- *visible string*,
- *date (data)*,
- *time of day* (godzina),
- *time difference* (różnica czasu),
- *bit string*.

11.6.2 Mechanizm dostępu do obiektów

Stacja aktywna (*master*) może odwoływać się do konkretnej zmiennej w stacji dołączonej poprzez różnego rodzaju adresy. PROFIBUS-FMS wyróżnia cztery różne rodzaje adresowania:

- *adresowanie logiczne* (poprzez indeksowanie). Adresowanie to pozwala na dostęp systemowi komunikacyjnemu do obiektów poprzez specjalny krótki adres zwany indeksem. Indeksy są wylistowane w *słowniku obiektów* i tworzą one poszczególne urządzenia logiczne. *Słownik obiektów* zawiera opis wszystkich obiektów znajdujących się w urządzeniu i dostępnych na

magistrali. Natomiast opis ten zawiera: indeks, nazwę obiektu, jego typ oraz długość. Wszystkie opisy obiektów mogą być przesyłane pomiędzy stacjami. Żeby można było przesłać dane użytkownika, muszą być wskazane indeksy przesyłanych zmiennych.

- *adresowanie fizyczne*. Adresowanie fizyczne może odbywać się tylko na obiektach mających dostęp fizyczny. Ten typ adresowania powinien być używany tylko w wyjątkowych sytuacjach, jak np. diagnozowanie występujących błędów.
- *adresowanie niejawne*. Adresem niejawnym w PROFIBUS'ie jest referencja komunikacyjna która jest używana do adresowania *urządzeń wirtualnych (VFD)*.
- *adresowanie za pomocą nazwy*. Obiekty komunikacyjne mogą być także adresowane za pomocą nazw złożonych z 32 znaków.

Jednak nie wszystkie stacje wspierają wymienione wyżej rodzaje odwołań, najczęściej używane jest adresowanie logiczne poprzez numer indeksu.

11.7 Usługi warstwy siódmej

Usługi PROFIBUS-FMS można podzielić na trzy grupy:

- *usługi aplikacyjne* - funkcje te umożliwiają dostęp do obiektów komunikacyjnych procesu aplikacyjnego. Zawierają one takie usługi jak np. *Read* do odczytywania zmiennych, *Write* do zapisywania zmiennych lub *Stop* za pomocą której zatrzymywane jest działanie programu.
- *usługi administracyjne* - usługi te są inicjalizowane przez użytkownika i odnoszą się one do obiektów oraz relacji komunikacyjnych. W tej grupie występują takie funkcje jak np. *GetOD*, która odczytuje zawartość słownika obiektów stacji dołączonej, *Initiate* lub *Abort* służące do inicjalizowania i zrywania połączenia komunikacyjnego.
- *usługi zarządzania siecią* - występują tu funkcje ustalające wielkość buforów, pozwalające modyfikować listę relacji komunikacyjnych.

Nie wszystkie jednak usługi są wspierane we wszystkich urządzeniach PROFIBUS'owych.

Usługi aplikacyjne. Wśród usług aplikacyjnych rozróżniamy następujące grupy funkcji:

Variable Access (dostęp do zmiennych) - usługi te pozwalają na dostęp do zmiennych prostych, rekordów, tablic oraz list zmiennych. Używając tych funkcji można dynamicznie tworzyć lub usuwać obiekty komunikacyjne z listy zmiennych.

Usługi należące do tej grupy to:

- *Read*,
- *Write*,
- *ReadWithType*,
- *WriteWithType*,
- *PhysRead*,
- *PhysWrite*,
- *InformationReport*,
- *InformationReportWithType*,
- *DefineVariableList*,
- *DeleteVariableList*.

Domain Access (dostęp do domen) - usługi te służą do zarządzania logicznie dołączonymi obszarami pamięci, np. załadowanie do pamięci lub pobranie z pamięci.

Do tej grupy należą następujące usługi:

- *InitiateDownloadSequence*,
- *DownloadSegment*,
- *TerminateDownloadSequence*,
- *RequestDomainSequence*,
- *InitiateUploadSequence*,
- *UploadSegment*,
- *TerminateUploadSequence*,
- *RequestDomainUpload*.

Program Invocation (zarządzanie programami) - funkcje te umożliwiają podstawowe działanie na programach, np. ich uruchamianie, zatrzymywanie lub kasowanie.

W tej grupie występują następujące usługi:

- *CreateProgramInvocation*,
- *DeleteProgramInvocation*,
- *Start*,
- *Stop*,
- *Resume*,
- *Reset*,
- *Kill*.

Event Management (zarządzanie zdarzeniami) - usługi te pozwalają na przesyłanie ważnych informacji pomiędzy dwoma stacjami.

Do tej grupy należą następujące usługi:

- *EventNotification*,
- *EventNotificationWithType*,
- *AcknowledgeEventNotification*,
- *AlterEventConditionMonitoring*.

Usługi administracyjne. Wśród usług administracyjnych rozróżniamy następujące grupy funkcji:

VFD support (wspieranie urządzeń wirtualnych) - za pomocą tych funkcji możliwe jest zarządzanie abstrakcyjnymi modelami urządzeń.

Do tej grupy usług należą następujące funkcje:

- *Status*,
- *UnsolicitedStatus*,
- *Identify*.

OD management (zarządzanie słownikiem obiektów) - usługi z tej grupy pozwalają wykonywać działania na słownikach obiektów.

Do usług z tej grupy należą takie funkcje jak:

- *GetOD*,
- *InitiatePutOD*,
- *PutOD*,
- *TerminatePutOD*.

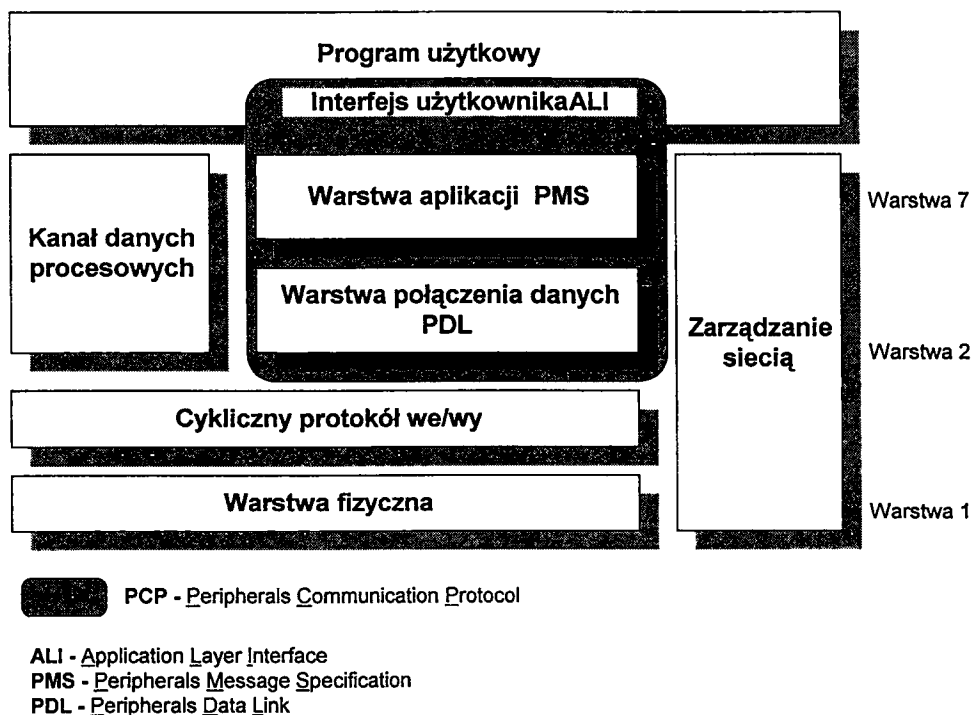
Context management (zarządzanie kontekstami) - usługi te inicjalizują połączenie komunikacyjne i przerywają je lub odrzucają nielegalne żądania.

Usługi z tej grupy to:

- *Initiate*,
- *Abort*,
- *Reject*.

11.8 Protokół PMS systemu InterBus-S

Aby zapewnić jednoznaczne i niezależne od producenta przesyłanie danych, muszą zostać ustalone reguły dotyczące rodzajów i typów wymiany danych. W systemie *InterBus-S* przesyłanie danych przez kanał danych parametrycznych odbywa się za pomocą specjalnego oprogramowania nazywanego *Peripherals Communication Protocol* (PCP) (rys.11.4)



Rys.11.4 Struktura protokołu PCP (*Peripherals Communication Protocol*)

Protokół PCP tworzą następujące elementy:

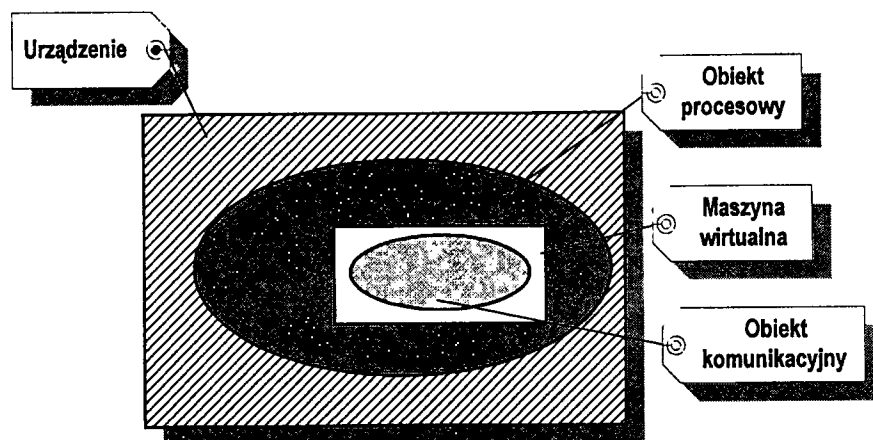
- warstwa interfejsowa aplikacji i użytkownika ALI (ang. *Application Layer Interface*),
- usługi PMS (ang. *Peripherals Message Specification*),
- warstwa łączenia z danymi PDL (ang. *Peripherals Data Link*).

11.8.1 Interfejs warstwy aplikacyjnej (ang. *Application Layer Interface* (ALI))

Interfejs warstwy aplikacyjnej ALI służy do połączenia programu użytkownika z warstwą siódmą systemu *InterBus-S*. Interfejs warstwy aplikacyjnej jest stałym elementem programu użytkownika i nie należy do siódmej warstwy systemu *InterBus-S* (rys.11.4). Interfejs warstwy aplikacyjnej ALI ma za zadanie udostępnić w warstwie użytkowej zdefiniowane usługi (PMS).

11.8.2. Model maszyny wirtualnej

Model maszyny wirtualnej opisuje rzeczywiste urządzenie w warstwie komunikacyjnej. W rzeczywistym urządzeniu istnieją tak zwane obiekty procesowe (rys.11.5).



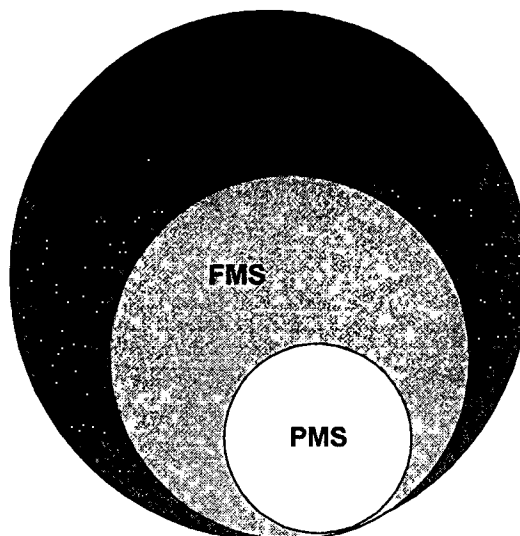
Rys.11.5 Obiekt komunikacyjny i obiekt procesowy

Celem komunikacji jest wymiana informacji pomiędzy partnerami komunikacyjnymi w obiektach procesowych. Aby dwa procesy użytkownika w różnych urządzeniach mogły wymieniać między sobą obiekty, to muszą one przekształcić dane procesowe na dane komunikacyjne. Aby to zrealizować, należy zarejestrować obiekt procesowy w tak zwanej liście obiektów (katalogu obiektów) jako obiekt komunikacyjny. Katalog obiektów jest znormalizowaną listą do której wprowadzane są obiekty komunikacyjne ze swoimi własnościami. Aby wymiana danych odbywała się swobodnie i bez przeszkód, to muszą być usystematyzowane (znormalizowane) nie tylko ogólnodostępne katalogi obiektów, ale także inne cechy urządzeń. Właśnie taka jednolita forma przedstawienia urządzenia nazwana jest modelem maszyny wirtualnej (rys.11.5).

11.8.3 *Peripherals Message Specification (PMS)*

Spełnienie wymagań użytkownika, dotyczących pionowego przesyłania danych pomiędzy użytkownikami sieci o różnych wymaganiach, sprzyja automatyzacji i integracji przedsiębiorstwa. Aby mogło to być zapewnione, musiał powstać jednolity system usług na wszystkich poziomach sieci.

W systemie InterBus-S do transmisji danych, używana jest struktura usług taka jak w MMS (ang. *Manufacturing Message Specification*). W hierarchicznej sieci komunikacyjnej, poniżej warstwy z usługami MMS występują jednak inne wymagania. Dlatego też zmniejszono zakres usług zgodnie z wymaganiami warstwy czujników i urządzeń wykonawczych (rys.11.6).



MMS - Manufacturing Message Specification

FMS - Fieldbus Message Specification

PMS - Peripherals Message Specification

Rys.11.6 Porównanie usług PMS z usługami MMS i FMS

Usługi PMS warstwy siódmej InterBus'a-S zawierają się w usługach Profibus'a FMS (ang. *Fieldbus Message Specification*). Wywołanie funkcji FMS i PMS jest identyczne, a jedynie przesyłanie danych przez magistralę odbywa się za pomocą innych mechanizmów.

InterBus-S w siódmej warstwie oferuje wiele usług, które można podzielić na trzy grupy:

Usługi użytkowe - funkcje wspomagające użytkownika ułatwiają dostęp do obiektów komunikacyjnych i procesowych. Do takich funkcji należą np. czytanie, pisanie w obiekcie komunikacyjnym, start, stop programu.

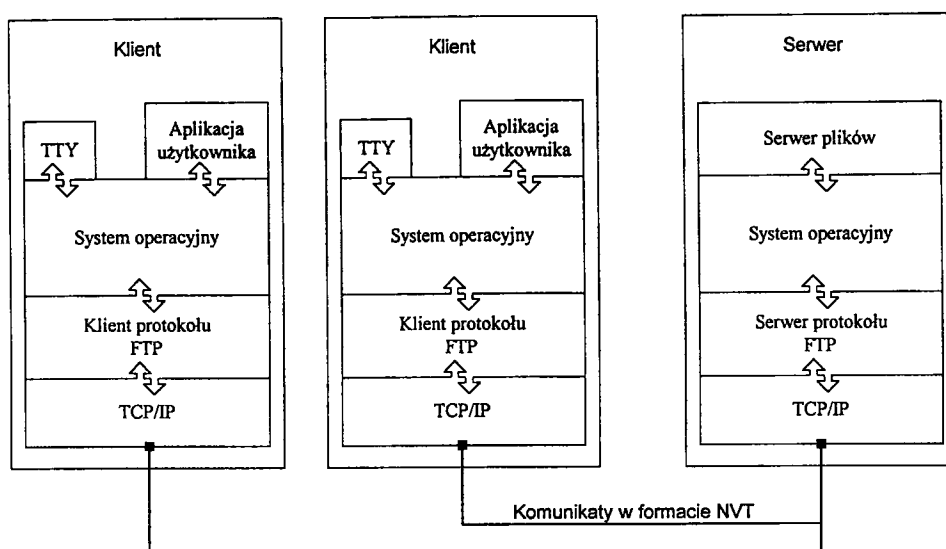
Usługi zarządzające - funkcje te służą do wymiany informacji przez obiekty maszyny wirtualnej. Usługi z tej grupy to np. czytanie rejestru z katalogu obiektów.

Usługi administrujące siecią - funkcje te ustalają pracę systemu sieciowego, magistrali np. inicjowanie związków komunikacyjnych.

12. Protokoły warstwy aplikacyjnej (FTP, Telnet)

12.1 FTP

Wykorzystanie dostępu do zdalnego systemu plików jest obecnie dominującym rozwiązaniem uzyskania zdalnej informacji. Do realizacji tego zadania w sieciach opartych na protokole TCP/IP najczęściej stosuje się protokół FTP (*File Transfer Protocol*). Klient FTP jest udostępniany użytkownikom przy terminalach lub ich aplikacjom przez system operacyjny (rys.13.1), przy czym najczęściej jeden klient może równocześnie obsługiwać kilku użytkowników. Pozwala on każdemu użytkownikowi w podobny sposób obsługiwać wiele różnych systemów plikowych. Za jego pomocą można zdalnie przeglądać kartotekę, tworzyć nowe zbiory, pobierać zawartość istniejących zbiorów, aktualizować je, kasować je itp.



Rys.12.1 Relacje *klient* – *serwer* w protokole FTP

Podobnie jeden serwer może równocześnie obsługiwać żądania od różnych klientów. Po otrzymaniu każdej komendy, serwer realizuje ją na lokalnym systemie w taki sposób jakby była na nim wygenerowana. Aby obsługiwać wiele żądań równocześnie, dla każdego nowego rozkazu serwer i klient (jeśli jest to możliwe) tworzy osobą sesję. Normalnie wszystkie nowe żądania przychodzą do pojedynczego procesu serwera, który tworzy nową sesję dla każdego połączenia. Proces potomny serwera sprawdza, na podstawie danych dostarczanych przez klienta, autoryzację dostępu oraz określa budowę i typ danych zbiorzonych do przesyłania. Wyróżnia się trzy typy budowy plików: bez określonej struktury, ciąg rekordów o stałej długości oraz ciąg rekordów o zmiennej długości z dostępem swobodnym. Jako typ danych zawartych w zbiorze może wystąpić: ciąg znaków ASCII, ciąg znaków EBCDIC, ciąg binarny oraz ciąg słów o zdefiniowanej długości.

Zbiory, które nie mają określonej struktury, mogą przedstawiać sobą dowolne informacje (dane binarne, teksty itp.) i są one przesyłane między dwoma użytkownikami protokołu FTP jako ciąg bitów. Odbiorca interpretuje ten ciąg i przekształca na dane według zdefiniowanego typu.

Zbiory zbudowane z rekordów o stałej długości, mogą być przesyłane w postaci bloków o ustalonej długości lub w formie skompresowanej. Kompresja jest właściwa dla plików zawierających ciągi powtarzających się bajtów. I tak np. dla zbiorów tekstowych, w których często występują długie łańcuchy tych samych znaków (np. znaki odstępu), przez zastąpienie informacją o kodzie i liczbie wystąpień, możliwe jest wyeliminowanie przesyłania powtarzających się bajtów. Ponieważ wspieranie kompresji nie jest obligatoryjne, dlatego też zastosowanie kompresji i jej algorytm muszą być wcześniej uzgodnione (zadanie takie pełni warstwa prezentacji w modelu referencyjnym OSI).

Zbiory z dostępem swobodnym są złożone z rekordów o zmiennej długości. Każdy rekord ma ze sobą skojarzony nagłówek, w którym zawarte są informacje o długościach pól i typach danych oraz pozycji rekordu względem początku pliku. Przesyłanie takiego pliku między dwoma jednostkami FTP odbywa się w takiej właśnie formie.

Gdy transfer dokonywany jest w postaci blokowej lub skompresowanej, to wykonywana jest kontrola postępu w transmisji, co pozwala na sterowanie przesyłaniem danych. Zadanie takie pełni usługa synchronizacji warstwy sesji w modelu referencyjnym OSI.

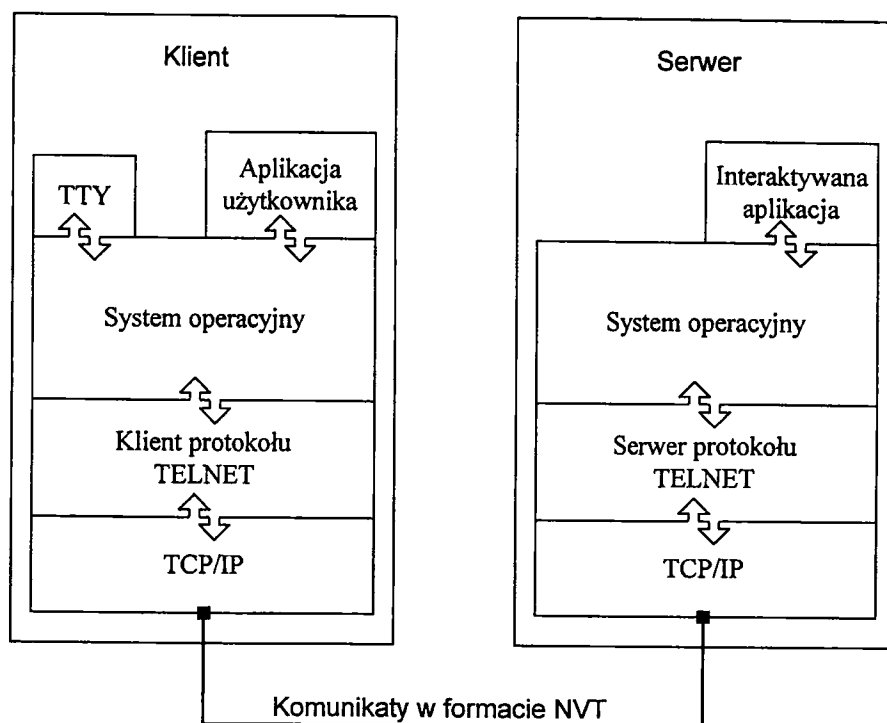
Pojedyncza sesja FTP otwiera dwa porty komunikacyjne. Jeden z nich jest przeznaczony do przesyłania komend sterujących a drugi do transferu pliku. Format przesyłania wiadomości związanych z zawartością zbioru jest determinowany zdefiniowaną budową pliku. Format wiadomości wymienianych pomiędzy dwoma sterującymi procesami FTP (*FTP protocol data units*) musi być przeprowadzany w uzgodnionym formacie, co zapewnia jednoznaczność komunikujących się komputerów. Uzyskuje się to przez stosowanie formatu NVT, przy czym nie jest wymagana negocjacja opcji.

12.2 Protokół TELNET

Klient protokołu TELNET jest udostępniany przez lokalny system operacyjny dla użytkowników przy terminalach. Dostarcza on usług pozwalających użytkownikowi logowanie się do zdalnego systemu z odległej maszyny i uruchamiania na nim interaktywnych procesów, np. *shell'a*, edytora tekstów, itp., w taki sposób jakby był uruchamiany na lokalnej maszynie. Wszystkie komendy (znaki sterujące) i dane wprowadzane na terminalu użytkownika lub generowane przez aplikację są przesyłane przez lokalny system operacyjny do klienta TELNET (rys.12.2), który korzystając ze strumienia niezawodnych komend dostarczanych usług przekazuje je przez TCP do korespondującego serwera TELNET. Tam są one wykonywane na rzecz użytkownika, a ich rezultat (dane wyjściowe) przesyłane są tą samą drogą na ekran terminala użytkownika lub do interpretacji przez jego aplikację.

Serwer TELNET jest często określany jako pseudo terminal. Klient i serwer TELNET komunikują się między sobą, korzystając z komend składających się z pojedynczych znaków lub łańcucha znaków, które są kodowane w standardowym formacie znanym jako sieciowy terminal wirtualny (NVT - *Network Virtual Terminal*). Zarówno komendy jak i dane wejścia - wyjścia dla programów interaktywnych są znakami ASCII. W tym przypadku gdy w lokalnym systemie używany jest inny

format, to konieczne jest stosowanie odpowiedniej funkcji odwzorowującej, przez co protokół TELNET pełni taką funkcję jak warstwa prezentacji w modelu odniesienia OSI.



Rys.12.2 Schemat relacji klient – serwer dla protokołu TELNET

W formacie NVT, wszystkie dane i komendy składają się z ośmiu bitów. Gdy najbardziej znaczący bit ma wartość 0, to bajt jest interpretowany jako standardowy znak ASCII (łącznie z kodami sterującymi), natomiast gdy bit ten ustawiony jest na 1, to traktowany jest jako komenda ze zbioru rozszerzającego. Opis tego zbioru zawarty jest w tab.12.1.

Tab.12.1 Bajty komend NVT

Komenda	Kod	Opis
IAC	FFh	Interpretuj następny bajt jako komendę
NOP	F1h	Bez operacji
EC	F7h	Kasuj poprzedni znak
EL	F8h	Kasuj linię
GA	F9h	Idź do przodu
AYT	F6h	Czy tam jesteś?
IP	F4h	Przerwij proces
AO	F5h	Przerwij wyjście
BRK	F3h	Wstrzymaj przesyłanie danych wyjściowych
DMARK	F2h	Wznów przesyłanie danych wyjściowych
SB	FAh	Star łańcucha negocjującego opcje
SE	F0h	Koniec łańcucha negocjującego opcje
WILL	FBh	Przyjęcie / żądanie opcji
WON'T	FCh	Odrzucenie żądania opcji
DO	FDh	Akceptacja żądania opcji
DON'T	FEh	Odrzucenie akceptacji opcji

Wszystkie komendy protokołu TELNET poprzedzone są bajtem IAC (FFh). Jeśli komenda składa się z więcej niż jednego bajta, to łańcuch znaków musi zostać poprzedzony komendą SB i zakończony przez komendę SE. Wszystkie dane normalnie są przesyłane na 7 bitach jako łańcuchy znaków ASCII. Dostępne jest przysyłanie łańcuchów danych 8 bitowych. Może to być stosowane np. do przysyłania specjalnych znaków ekranowych (znaki narodowe, itp.) lub przy wymianie bloków danych binarnych przez aplikację. Aby tego dokonać jedna ze stron musi wynegocjować opcje wysyłając ciąg komend (WILL, WON'T, DO, DON'T) wraz z kodem operacji. Inne funkcje dostępne w podobny sposób oraz ich kody operacji przedstawiono w tab.12.2.

Tab.12.2 Kody opcji NVT

Nazwa opcji	Kod	Opis
Transmisja binarna	0	Akceptacja / żądanie transmisji danych ośmiobitowych
Echo	1	Włączenie trybu „echo” (wysyłanie odebranego znaku z powrotem do wysyłającego)
Status	5	Żądanie / powtórzenie statusu odbioru
Znacznik czasowy	6	Wstaw znacznik czasowy w zwracanym ciągu
Typ terminala	24	Żądanie / odpowiedź o typ zdalnego terminala
Tryb liniowy	34	Wysyłanie całych linii danych

Sekwencja:

IAC,SB,WILL,'0',SE

wysyła żądanie wprowadzenia trybu przysyłania danych binarnych, jeśli operacja ta może być zaakceptowana, to respondent powinien wysłać:

IAC,SB,DO,'0',SE .

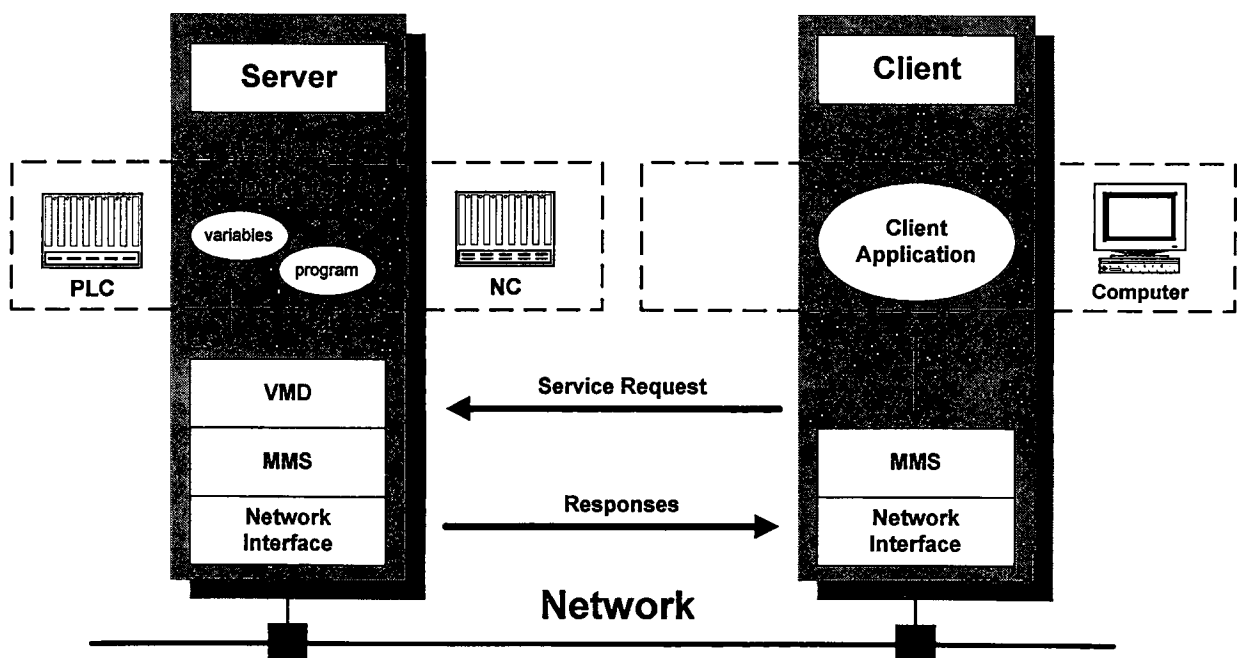
Jeśli przesyłane są dane binarne to każdy znak o kodzie 255 (IAC) nadawany jest podwójnie, a odbiorca interpretuje go jako bajt danych a nie komendę.

Operacja włączenia znaczników czasowych pełni podobną rolę jak funkcja synchronizacji.

13. Protokoły warstwy aplikacyjnej wg specyfikacji MAP/TOP

13.1 Specyfikacja komunikatów przemysłowych MMS

Przykładem protokołu komunikacyjnego w pełni zgodnego z modelem ISO/OSI jest protokół automatyzacji wytwarzania MAP (ang. *Manufacturing Automation Protocol*). Zadaniem specyfikacji MAP jest standaryzacja interfejsów komunikacyjnych komputerów i sterowników urządzeń wytwórczych, a co za tym idzie, ułatwienie prac i redukcja kosztów związanych z integracją poszczególnych środków automatyzacji w jednolity system produkcyjny. W warstwie aplikacyjnej specyfikacji MAP 3.0 występuje protokół MMS (ang. *Manufacturing Message Specification*). MMS jest systemem informatycznym służącym do wymiany znormalizowanych komunikatów pomiędzy różnymi sterownikami i komputerami w maszynach i urządzeniach przemysłowych podczas realizacji wspólnego zadania wytwórczego. MMS, podczas wymiany informacji między dwoma partnerami komunikacyjnymi (np. komputer - sterowniki CNC lub PLC), opiera się w zasadzie na dwóch uzupełniających się modelach. Są to: model Klient-Serwer (ang. *Client-Server*) i model Wirtualnego Urządzenia Wytwórczego (ang. *Virtual-Manufacturing-Device VMD*) (rys.14.1). Parę *Klient-Serwer* mogą na przykład stanowić komputer sterujący gniazda wytwórczego i sterownik CNC obrabiarki. Wirtualne urządzenie wytwórcze VMD może być przedstawione jako maska ukrywająca takie rzeczywiste cechy urządzenia, np. sterownika PLC, jak: zmienne lub programy.



Rys.13.1 Model VMD oraz technika Klient-Serwer w aplikacjach MMS

Specyfika komunikacji przemysłowej MMS polega na tym, że wszystkie urządzenia w środowisku przemysłowym muszą mieć możliwość jednakowej interpretacji każdej instrukcji jaka

ma być wykonana, co przy uwzględnieniu różnorodności stosowanego w przemyśle sprzętu powoduje, że MMS musi realizować względnie dużą liczbę usług. Komunikacja między dwoma urządzeniami polega na przesyłaniu komunikatów zawierających zbiór parametrów o ściśle wyspecyfikowanych wartościach i typach. MMS dostarcza standardowego języka komend, który umożliwia urządzeniom wysyłanie i jednoznaczną interpretację takich komunikatów. Specyfikacja komunikatów MMS jest prezentowana w notacji składni abstrakcyjnej za pomocą języka ASN.1.

MMS zezwala na różnego typu sterowania. W szczególności może to być zarówno sposób polegający na cyklicznym żądaniu od urządzeń (ang. *polling*) przesłania wartości określonych parametrów (np. stanu urządzenia), jak też i pracę w trybie sterowania zdarzeniami, polegającą na samoczynnym przekazywaniu informacji przez urządzenie w wyniku wystąpienia określonego zdarzenia. MMS zezwala również na korzystanie ze wspólnych zasobów w środowisku przemysłowym, wykorzystując mechanizmy semaforowe do synchronizacji i sterowania ich przydziałem.

Poza podstawową funkcją MMS, jaką jest wymiana wiadomości między urządzeniami przemysłowymi, MMS dostarcza ogólnych usług realizowanych w warstwie aplikacji, takich jak ustanowienie asocjacji (połączenia) oraz przesyłanie plików MMS.

Specyfikacja MMS definiuje operacje i semantykę w sposób na tyle ogólny, że aplikacja ta może mieć zastosowanie niemal we wszystkich gałęziach przemysłu. Uwzględnienie rozległych potrzeb potencjalnych użytkowników prowadzi do definicji dużej liczby różnorodnych funkcji i ich możliwości, co nie pozostaje bez wpływu na wysoki poziom złożoności protokołu MMS.

W samej koncepcji protokołu wymiany wiadomości skorzystano z modelu komunikacji *Klient-Serwer*, reprezentowanego przez niesymetryczną współpracę dwóch procesów (inicjatora i respondenta). Zaletą takiego rozwiązania jest koncentracja złożoności implementacyjnej w procesie inicjatora, pracującego zwykle na sterownikach o dużej mocy obliczeniowej, przy równoczesnym utrzymaniu prostych implementacji procesu *responder'a*. Klient żąda usługi, jaka ma być wykonana przez serwer, używając konwencjonalnych podstawowych jednostek usługowych typu *.request*, *.indication*, *.response* i *.confirm*.

W modelu *Klient-Serwer* MMS wychodzi się z założenia, że dla uzyskania celu wytwarzania, procesy użytkownika w wielu wspomaganych komputerowo systemach wytwórczych wykonują wspólne zadanie przetwarzania danych. Oznacza to, że przetwarzanie danych odbywa się w systemie rozproszonym. Do wymiany informacji, zawsze pomiędzy dokładnie dwoma procesami użytkownika, jest przy tym budowana w zasadzie czasowo nieograniczona relacja komunikacyjna. Obaj partnerzy komunikacyjni (procesy użytkownika) podczas wymiany informacji spełniają różnorodne role (relacja niesymetryczna). Zależnie od tej roli są oni określani jako „*Klient*” lub jako „*Serwer*”.

Klient jest tym uczestnikiem komunikacji, który u swojego partnera komunikacyjnego, tzn. *Serwera*, żąda wykonania określonego zlecenia, takiego jak np. „prześlij status urządzenia”, „załaduj wartość zmiennej”, „uruchom realizację programu”. Dla zażądania wykonania zlecenia *Klient* przesyła do *Serwera* żądanie - *Request-Telegram*. Serwer jest natomiast tym partnerem komunikacyjnym, który wykonuje przydzielone mu zlecenie, a zatem np. określa on status lub

uruchamia „wykonanie programu”. Serwer melduje o wyniku wykonania zlecenia do *Klienta* za pomocą odpowiedzi - *Reponse-Telegram*. Jeżeli wykonanie zlecenia nie jest możliwe, to zamiast *Reponse-Telegram* jest wysyłany komunikat o błędzie - *Error-Telegram*. Ponadto *Serwer*, w przypadku wystąpienia nieoczekiwanych wyników, może spontanicznie przesłać do *Klienta* takie informacje jak np. „Alarm” lub „nie zażądany status urządzenia”. Za pomocą tworzonych przez MMS relacji komunikacyjnych uczestnik komunikacji (proces użytkownika) w ramach swoich zadań może w zasadzie tworzyć dowolnie wiele innych „logicznych” relacji MMS do wielu innych partnerów komunikacyjnych. Dopuszczalne jest również to, że pomiędzy dwoma systemami sprzętowymi istnieje równocześnie wiele relacji komunikacyjnych. Sensowny może być również taki scenariusz, w którym np. w jednym komputerze wytwarzania są zaimplementowane dwa procesy użytkownika, z których jeden w sztywnej roli realizuje jako *Klient* sterowanie i nadzorowanie podporządkowanej mu maszyny NC (*Serwer*), podczas gdy drugi proces użytkownika w roli serwera zbiorów przygotowuje wymagane przez maszynę (*Klienta*) programy NC. W obu relacjach komunikacyjnych może mieć miejsce, dla użytkownika wydawałoby się równoczesna i niezależna od siebie, wymiana informacji.

Działanie procesu aplikacyjnego serwera MMS jest modelowane przez abstrakcyjną definicję operacji wykonywanych przez jednostkę zwaną wirtualnym urządzeniem przemysłowym VDM (ang. *Virtual manufacturing device*). Na definicję tą składa się szereg takich elementów jak np. domena, realizowana funkcja, zmienne itp. Metody modelowania rzeczywistego urządzenia jako VDM mogą się od siebie różnić i dlatego też np. jedno urządzenie może być reprezentowane jako dwa VDM.

Domena jest to abstrakcyjny obiekt reprezentujący możliwości VMD ukierunkowane na realizację określonego celu. Domena zawiera więc wszystkie zasoby potrzebne do wykonania specyficznego aspektu zadania aplikacji (np. sterowania lub monitorowania). Przykładem domeny może być zamodelowane ramię robota, na które składa się komplet zasobów koniecznych do realizacji operacji na rzeczywistym ramieniu wraz z danymi i kodem programu sterującego. Domena może być statyczna, jeśli istnieje zawsze w VMD lub dynamiczna, jeśli jest tworzona i usuwana w miarę potrzeb aplikacji.

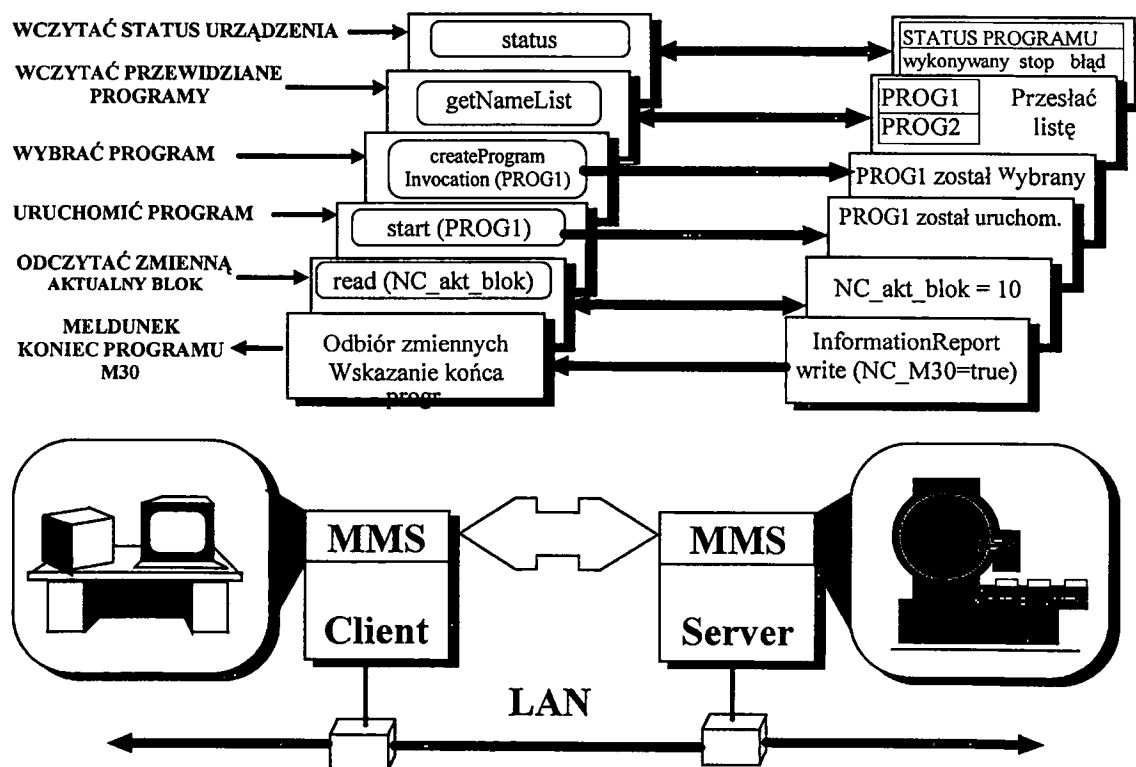
Specyfikacja MMS definiuje 87 usług, które wykorzystywane w różnych kombinacjach są źródłem tak dużej funkcyjności aplikacji. Usługi te można sklasyfikować w dziewięciu grupach:

- *Usługi środowiska i ogólnego zarządzania.* Do grupy tej zalicza się funkcje dostarczające mechanizmów ustanawiania i zwalniania środowiska MMS. Służą do tego usługi takie jak: *.initiate*, *.conclude*, *.abort*, *.cancel* oraz *.reject*.
- *Usługi zarządzania VDM.* Dostarczają one mechanizmów zarządzania VDM za pomocą takich funkcji jak zapis, pobranie listy obiektów w VDM, modyfikacja nazw obiektów oraz obsługi specyficznych atrybutów producenta.
- *Usługi zarządzania domenami.* Obejmują one modelowanie i utrzymanie domen w VMD. Usługi tej grupy definiują obiekt domeny z jej atrybutami oraz pozwalają na utworzenie i załadowanie obiektów domeny.

- *Usługi zarządzania inwokacją programu.* Usługi te grupują funkcje sterujące inwokacjami programów. Działają one na obiekcie inwokacji programu, pozwalając na jego utworzenie, usunięcie, wystartowanie, zatrzymanie, wznowienie itp.
- *Usługi dodatkowych zmiennych* wykorzystywane są do specyficznej wymiany informacji, pozwalając na podstawowe czynności manipulacyjne.
- *Usługi zarządzania semaforami.* Jeśli system wykorzystuje semaforey, muszą one być sterowane prostymi usługami tej grupy, np. definiowania i usuwania semafora, przejęcia sterowania czy też raportowania stanu semafora.
- *Usługi komunikacji operatorskiej* wspomagają lokalny interfejs człowiek - urządzenie.
- *Usługi zarządzania zdarzeniami* pozwalają klientowi na zdefiniowanie zdarzeń oraz otrzymanie powiadomienia, jeśli takie wystąpią.
- *Usługi zarządzania dziennikiem.* MMS pozwala na gromadzenie w dzienniku istotnych informacji. Usługi tej grupy pozwalają na inicjację dziennika, odczytanie, zapis oraz sprawdzenie stanu dziennika.

Chociaż w praktyce protokoły MAP wraz z MMS nie odniosły sukcesu komercyjnego, to dały one wzorzec dla projektowania sieci miejscowych. Obecnie niemal wszystkie liczące się standardy sieci miejscowych *Fieldbus* (np. Profibus, InterBus-S, LONWorks) mają opracowaną warstwę aplikacyjną (7) zgodną z MAP/MMS.

Na rysunku 13.2 pokazany jest przykładowo typowy scenariusz komunikacyjny pomiędzy komputerem kierowania lub komputerem gniazda i układem sterowania obrabiarki w systemie wytwórczym.



Rys.13.2 Przykład wykorzystania MMS

W tym przykładzie jest pokazane jakie usługi MMS mogą znaleźć zastosowanie, aby wykonać określony program NC w układzie sterowania obrabiarki. Najpierw przez odpytanie o status następuje sprawdzenie, czy obrabiarka jest gotowa do pracy. Na zakończenie za pomocą usługi *GetNameList* następuje wczytanie do obrabiarki przewidzianego programu NC. Istniejący w układzie sterowania obrabiarki program NC PROG1 jest wybierany przez usługę MMS *CreateProgrammInvocation* i uruchamiany za pomocą usługi *Start*. Podczas wykonywania programu, za pomocą zdefiniowanej w tym przypadku przez producenta obrabiarki zmiennej NC_akt_blok, można zameldować o aktualnie wykonywanym bloku programu NC. Za pomocą nie zażądanego meldunku (*InformationReport*) układ sterowania obrabiarki po zakończeniu programu NC przesyła zmienną NC_30, która komputerowi nadrzędnemu wskazuje, że wykonywanie programu NC zostało zakończone.

13.2 Normy stowarzyszone MMS - *Companion Standards do MMS*

Rdzeń MMS określa jednoznacznie strukturę generowanych klas obiektów i wykonywanych na nich usług. Pozostaje jednak wiele niejasnych miejsc odnośnie odpowiedzi na pytanie, jakie funkcje i właściwości urządzenia automatyzacji można i daje się odwzorować za pomocą obiektu określonej klasy. I tak np. funkcję obrabiarki sterowanej numerycznie, która na żądanie komputera wytwarzania powinna wykonać zjazd na punkt referencyjny, można teoretycznie odwzorować zarówno przez obiekt *Program-Invocation*, za pomocą obiektu *Event*, jak również za pomocą obiektu *Variable*. Implementatorowi pozostaje podjęcie decyzji o tym, który z tych obiektów wybierze on do odwzorowania najazdu na punkt referencyjny. Oznacza to, że w zasadzie każdy z implementatorów może zastosować własne rozwiązanie dla takiego samego problemu technicznego. Dlatego też sformułowano podstawowe wymaganie dotyczące otwartych interfejsów komunikacyjnych, które mówi, że takie same funkcje i właściwości na zewnątrz powinny być jednakowo przedstawione, jest w ten sposób nie spełnione. Otwarte obiekty MMS i usługi użytkownika są w ten sposób podstawą nowych specyficznych dla producentów interfejsów komunikacyjnych. Nie powinno to być oczywiście sensem MMS. W uzupełnieniu do rozszerzeń atrybutów i klas obiektów muszą być dlatego określone reguły, które podadzą za pomocą jakich klas obiektów powinny być reprezentowane określone właściwości i funkcje urządzenia automatyzacji.

Z wyżej wymienionych przyczyn wynika konieczność rozszerzenia wzgl. sprecyzowania generacyjnych konstrukcji rdzenia MMS. Ze względu na występowanie znacznych różnic we właściwościach urządzeń automatyzacji okazało się przy tym celowe, aby to uproszczenie i sprecyzowanie rdzenia MMS zorientować funkcjonalnie, tzn. przeprowadzić oddzielnie dla określonych kategorii urządzeń automatyzacji. Wynikające z koniecznych do tego celu prac normy są określane jako *Companion Standards* (normy towarzyszące) do MMS.

Wraz z opublikowaniem *Manufacturing Message Specification* (MMS) jako międzynarodowej normy w październiku 1990 r. został skutecznie wykonany ostatni krok na drodze do powstania niezależnych od producentów, tzn. otwartych interfejsów komunikacyjnych dla stosowanych w wytwarzaniu urządzeń automatyzacji. Na przestrzeni okresu historii transmisji danych w

wytwarzaniu dopiero dzisiaj jest możliwe wyposażanie urządzeń automatyzacji w neutralne od producentów interfejsy komunikacyjne w oparciu o stabilne normy. MMS odnośnie oferowanej przez niego funkcjonalności jest otwarty na życzenia i już dzisiaj można wyraźnie stwierdzić, że ze względu na jego zaawansowane i ukierunkowane na przyszłość koncepcje, może w następnych dziesięcioleciach decydować o rozwoju komunikacji w wytwarzaniu.

Zestaw norm dotyczących standardu ISO/MMS obejmuje:

- ISO/IEC 9506-1, Industrial Automation Systems - Manufacturing Message Specification - Part 1: Service Definition, 1990.
- ISO/IEC 9506-2, Industrial Automation Systems - Manufacturing Message Specification - Part 2: Protocol Specification, 1990.
- ISO/IEC 9506-3, Industrial Automation Systems - Manufacturing Message Specification - Part 3: MMS Companion Standard for Robotics, 1991.
- ISO/IEC 9506-4, Industrial Automation Systems - Manufacturing Message Specification - Part 4: MMS Companion Standard for Numerical Control, 1991.
- ISO/IEC 9506-5, Industrial Automation Systems - Manufacturing Message Specification - Part 5: MMS Companion Standard for PLC.
- ISO/IEC 9506-6, Industrial Automation Systems - Manufacturing Message Specification - Part 6: MMS Companion Standard for Process Industry.
- ISO/IEC 9506-1, Amendment 1: Data Exchange (Service), 1993.
- ISO/IEC 9506-2, Amendment 1: Data Exchange (Protocol), 1993.

13.3 Protokół transferu, dostępu i zarządzania zbiorami - FTAM

13.3.1 Wprowadzenie

Warstwowy model otwartych systemów komunikacyjnych OSI definiuje ustandaryzowaną architekturę sieci, która zapewnia komunikację połączeniową (ang. *connection-oriented*) pomiędzy urządzeniami pochodzącymi od różnych producentów. Użytkownicy sieci zgodnych z OSI mogą dokładnie przewidzieć jak zachowają się inne węzły sieci, a połączenie nawiązane z systemem oddalonym wg standardów OSI gwarantuje poprawną wymianę danych.

Międzynarodowa Organizacja Standaryzująca ISO (ang. *International Standards Organisation*) opracowała model OSI po to, aby zdefiniować działania i usługi siedmiu funkcji komunikacyjnych nazywanych warstwami. Warstwą siódmą tego modelu jest warstwa aplikacji (ang. *Application layer*) tzn. warstwa, na której jest osadzony program aplikacyjny użytkownika. Protokół FTAM (ang. *File Transfer, Access and Management*) jest protokołem umiejscowionym w najwyższej warstwie protokołów w modelu OSI, warstwie aplikacji (rys.13.3).

System FTAM określa technikę wymiany usług pomiędzy klientem (inicjatorem) a serwerem (odpowiadającym - w specyfikacji FTAM używa się określenia *responder*) w środowisku systemów otwartych. Jest on standardem zgodnym z założeniami otwartych systemów komunikacyjnych, jest zaimplementowany we wszystkich siedmiu warstwach modelu OSI oraz pretenduje do roli uniwersalnego systemu plików. Specyfikacja FTAM zawiera się w dokumencie ISO 8571 i składa

się z czterech części, z których część pierwsza zawiera informacje ogólne o systemie FTAM, część druga definiuje wirtualny system plików VFS (ang. *Virtual Filestore System*), natomiast część trzecia zawiera definicję usług natomiast czwarta definiuje protokoły.

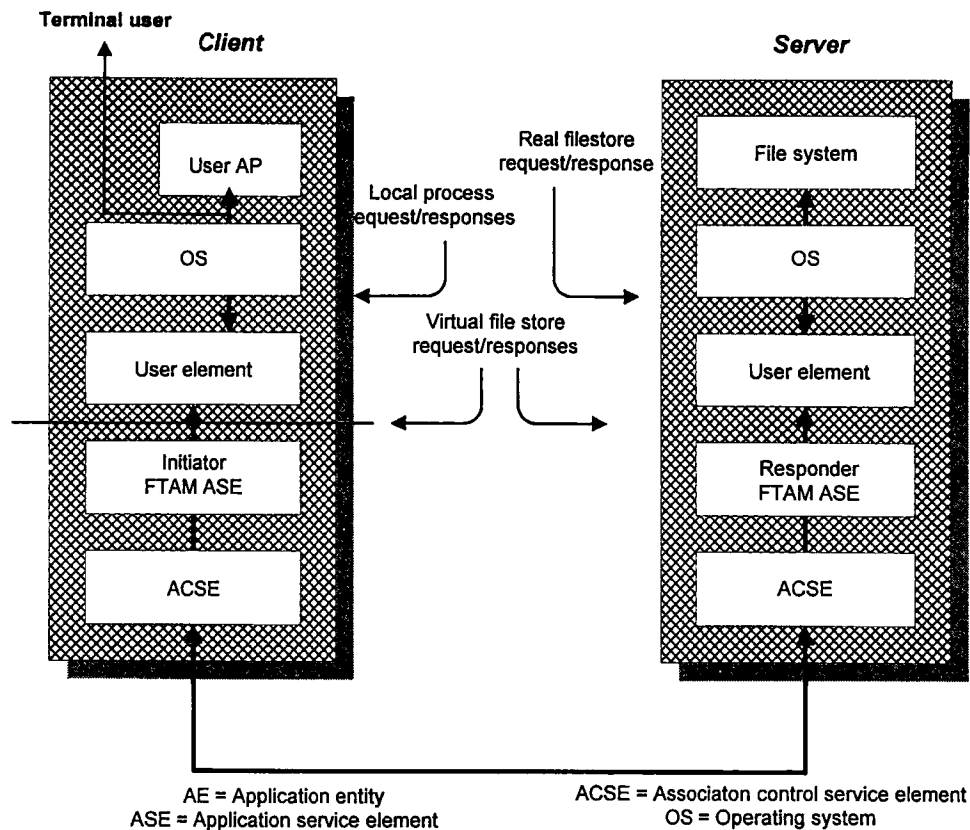
Warstwa	Protokoły
Aplikacji	VT DS FTAM CMIP CMIS MHS ACSE, RTSE, ROSE ASN.1
Prezentacji	Warstwa prezentacji OSI
Sesji	Warstwa sesji OSI
Transportowa	TP0, TP1, TP2, TP3, TP4
Sieciowa	Stacja-węzeł, Węzeł-węzeł X.25, PLP CLNP
Łącza danych	IEEE 802.2 HDL, LAPB
Fizyczna	IEEE 802.3 IEEE 802.4 IEEE 802.5 FDDI RS-232, RS-449, X.21, V.35, ISDN

Rys.13.3 Protokoły komunikacyjne w odniesieniu do modelu ISO/OSI

System FTAM zaprojektowano z myślą o użytkownikach wykorzystujących różnorodne platformy sprzętowe i systemowe po to, aby umożliwić im zarządzanie i wzajemną wymianę plików. W uproszczeniu jest on podobny do FTP (ang. *File Transfer Protocol*) oraz NFS (ang. *Network File System*), które działają w środowisku systemów wykorzystujących TCP/IP.

Generalnie protokół FTAM, określony w ISO 8571, definiuje jednolity model dla operacji na plikach w systemach otwartych, włączając w to strukturę plików, definicję usług oraz protokołów.

Wszystkie elementy usługowe ASE (ang. *Application Service Element*) warstwy aplikacji OSI zostały zaprojektowane i zbudowane z myślą o dostarczeniu środków oraz narzędzi, które pozwolą procesom aplikacyjnym użytkownika swobodnie się komunikować. FTAM jest przykładem takiego elementu usługowego warstwy aplikacji. Poprzez FTAM rozproszona grupa procesów klienta AP (ang. *Application Process*) może udostępniać i zarządzać oddalonym serwerem plików (ang. *file server*). Każdy proces klienta oraz jednostka aplikacji AE (ang. *Applicatin Entity*), z którym jest ona związana, jest nazywany inicjatorem (ang. *initiator* AP), natomiast proces serwera - responderem (od ang. *responder*) (rys.13.4).



Rys.13.4 Technika wirtualnego urządzenia FTAM

Funkcje pierwotne tzw. prymitywy (ang. *primitives*) używane przez procesy klienta AP do zarządzania i udostępniania plików na oddalonym systemie VFS, jak również funkcje używane do zarządzania i udostępniania plików w systemie rzeczywistym, są związane lokalnie z platformą sprzętową i systemem operacyjnym (rys.13.4)

HP FTAM/9000 jest przykładem implementacji standardu FTAM ISO 8571 na platformę HP 9000. Oferuje on interfejs programisty w języku "C" zdefiniowany w MAP 3.0 oraz zapewnia szereg usług takich jak:

- komunikację w środowisku sprzętu i oprogramowania pochodzącego od różnych producentów,
- transfer plików binarnych i tekstowych,
- udostępnianie plików (lokalizacja danych na poziomie rekordów oraz ich czytanie),
- zarządzanie plikami, włączając w to:
- tworzenie i kasowanie plików,
- manipulację atrybutami plików (czytanie i zmiana),
- mechanizmy zabezpieczenia takie jak: blokowanie pliku, zakładanie hasła na system plików, równoległe sterowanie oraz sterowanie dostępem do plików.

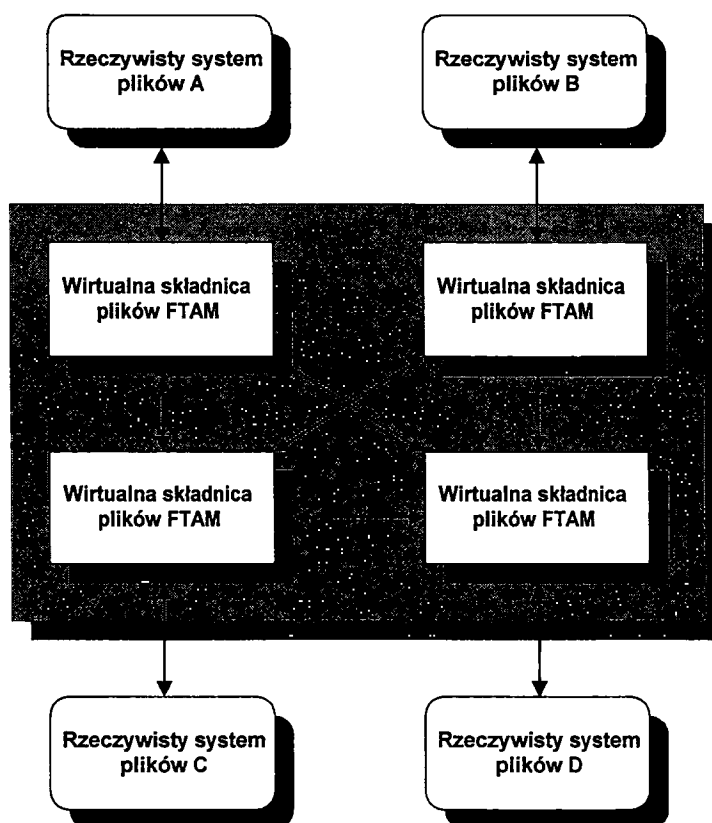
Implementacja FTAM na platformie HP-UX jest zgodna z następującymi standardami i specyfikacjami:

- Open System Interconnection (OSI) Model,
- ISO/IS 8571 FTAM,
- Manufacturing Automation Protocol MAP 3.0 Application Interface Specification,

- NBS (NIST) Phase III Implementors Agreements.
- Dodatkowo firma HP zaimplementowała profile T1.3, T2.3, A1.3 oraz M1.3 na podstawie NBS (NIST) Phase III Implementors Agreements.

13.3.2 Wirtualny system plików (VFS - *Virtual Filestore*)

Protokół FTAM został zaprojektowany tak, by ukryć przed użytkownikiem szczegóły implementacyjne systemu plików. W tym celu wprowadzono pojęcie wirtualnej składnicy plików (ang. *virtual file store*), która ma ukryć przed użytkownikiem specyficzne elementy rzeczywistego systemu plików, zwanego tu rzeczywistą składnicą plików (ang. *real file store*). Dzięki temu użytkownik może, za pośrednictwem usług FTAM, w jednakowy sposób korzystać z plików znajdujących się w różnego typu systemach komputerowych zarządzanych przez różne systemy operacyjne (rys.13.5).



Rys.13.5 Model wirtualnej składnicy plików FTAM

Rzeczywisty system plików wykorzystuje różnorodne mechanizmy po to, aby zbudować strukturę plików, zapisywać informacje na dysku oraz naprawiać i odzyskiwać utracone informacje w pliku. Na przykład pliki w systemie HP-UX są zapisywane na dysku oraz udostępniane jako strumień bajtów, natomiast w innych systemach plików może być stosowana technika zapisywania i udostępniania rekordów o zmiennej lub stałej długości lub też inne techniki.

Pojęcie wirtualnej składnicy plików leży u podstaw idei „przezroczystej” wymiany danych za pośrednictwem protokołu FTAM. Wirtualna składnica plików opisuje pliki, system ich przechowywania oraz dostępne, związane z plikami operacje. Jest ona tak zaprojektowana, aby

odwołania do niej można było przenieść na dowolną rzeczywistą składnicę plików, za pomocą minimalnej liczby funkcji odwzorowujących operacje na wirtualnej składnicy plików na operacje na rzeczywistej składnicy plików.

Wirtualna składnica plików jest adresowalną jednostką w sieci, z którą użytkownik może utworzyć połączenie sieciowe (asocjację). W danym momencie ze składnicą plików respondera może być połączona co najwyżej z góry określona liczba inicjatorów. Składnica plików składa się z plików, z których każdy ma określone atrybuty. Atrybuty określają: nazwę pliku, umożliwiającą jednoznaczne odwołania się do pliku; dozwolone akcje (np. odczyt, wstawienie elementu, zastąpienie zawartości); rodzaj dostępu, np. tylko do odczytu, do odczytu i zapisu; wielkość pliku; kontekst prezentacji (syntaktyka) zawartości pliku; nazwę twórcy pliku; datę i czas utworzenia; nazwę użytkownika ostatnio modyfikującego plik; datę i czas ostatniej modyfikacji; typ zawartości oraz klucz szyfrujący.

Implementacje FTAM przetwarzają wirtualną składnicę plików VFS w rzeczywistą składnicę plików. Funkcje przetwarzające (ang. *Mapping Function*) od systemów otwartych do rzeczywistych absorbują różnice występujące w specyfikacjach struktur plików. Uogólniając, od użytkownika FTAM nie wymaga się znajomości rozwiązań systemów operacyjnych czy też systemów plików, z którym chce się komunikować. Wymaga się natomiast znajomości pełnej nazwy pliku oraz jego ulokowania.

Informacje związane z plikiem wirtualnym mogą być podzielone na dwie klasy: zawartość pliku oraz jego atrybuty. Podczas gdy pierwsza klasa jest opcjonalna (plik może być pusty), to druga jest obowiązkowa, ponieważ np. atrybut nazwy plików musi istnieć.

Pliki FTAM składają się z dwóch części: atrybutu pliku oraz jego zawartości. Atrybuty opisują plik (wielkość, założyciela) i można wyróżnić zarówno atrybuty pliku (ang. *file attributes*) jak i atrybuty aktywności (ang. *activity attributes*),

Atrybuty pliku - definiują właściwości pliku, które są dostępne dla wszystkich inicjatorów. Atrybuty pliku są przechowywane i tworzone wraz z plikiem i są nieodłączną częścią pliku (np. wielkość pliku). Niektóre atrybuty pliku identyfikują strukturalne aspekty zawartości pliku.

Atrybuty aktywności - definiują właściwości, które istnieją tylko podczas używania specyficznych połączeń FTAM. Atrybuty te nie są częścią pliku (np. *file_store_password*). Zbiór atrybutów tej klasy jest związany z każdym reżimem (połączeniem) FTAM i może być podzielony na:

- **atrybuty aktywne** (ang. *active attributes*) - opisują one aktywne wartości atrybutów pliku, które są postrzegane przez inicjatora,
- **atrybuty bieżące** (ang. *current attributes*) - opisują zaistniałe fakty o inicjatorze i bieżącym stanie informacji podczas transferu pliku.

Lista atrybutów podzielonych na trzy wyżej opisane klasy i zdefiniowanych w FTAM jest pokazana w tab.13.1.

Tab.13.1 Lista atrybutów pliku wirtualnego FTAM

<i>Nazwa atrybutu</i>	<i>Typ atrybutu</i>	<i>Grupa</i>
<i>Filename</i>	<i>File attr.</i>	<i>Kernel</i>
<i>Permitted actions</i>	<i>File attr.</i>	<i>Kernel</i>
<i>Contents type</i>	<i>File attr.</i>	<i>Kernel</i>
<i>Active contents type</i>	<i>Activity attr.</i>	<i>Kernel</i>
<i>Current access request</i>	<i>Activity attr.</i>	<i>Kernel</i>
<i>Current initiator identity</i>	<i>Activity attr.</i>	<i>Kernel</i>
<i>Current location</i>	<i>Activity attr.</i>	<i>Kernel</i>
<i>Current processing mode</i>	<i>Activity attr.</i>	<i>Kernel</i>
<i>Current calling application title</i>	<i>Activity attr.</i>	<i>Kernel</i>
<i>Current responding application title</i>	<i>Activity attr.</i>	<i>Kernel</i>
<i>Storage account</i>	<i>File attr.</i>	<i>Storage</i>
<i>Date and time of creation</i>	<i>File attr.</i>	<i>Storage</i>
<i>Date and time of last modification</i>	<i>File attr.</i>	<i>Storage</i>
<i>Date and time of last read access</i>	<i>File attr.</i>	<i>Storage</i>
<i>Date and time of last attribute</i>	<i>File attr.</i>	<i>Storage</i>
<i>Modification</i>	<i>File attr.</i>	<i>Storage</i>
<i>Identity of creator</i>	<i>File attr.</i>	<i>Storage</i>
<i>Identity of last modifier</i>	<i>File attr.</i>	<i>Storage</i>
<i>Identity of last reader</i>	<i>File attr.</i>	<i>Storage</i>
<i>Identity of last attribute modifier</i>	<i>File attr.</i>	<i>Storage</i>
<i>Filesize</i>	<i>File attr.</i>	<i>Storage</i>
<i>Future filesize</i>	<i>File attr.</i>	<i>Storage</i>
<i>Current account</i>	<i>Activity attr.</i>	<i>Storage</i>
<i>Current concurrency control</i>	<i>Activity attr.</i>	<i>Storage</i>
<i>Current locking style</i>	<i>Activity attr.</i>	<i>Storage</i>
<i>Access control</i>	<i>File attr.</i>	<i>Storage</i>
<i>Legal qualification</i>	<i>File attr.</i>	<i>Storage</i>
<i>Current access passwords</i>	<i>Activity attr.</i>	<i>Storage</i>
<i>Active legal qualification</i>	<i>Activity attr.</i>	<i>Storage</i>
<i>Private use</i>	<i>File attr.</i>	<i>Private</i>

W ostatniej kolumnie tab.13.1 znajduje się typ grupy, do której należą poszczególne atrybuty pliku. Grupa *Kernel* zawiera cechy wspólne dla wszystkich plików i jest obowiązkowa dla wszystkich implementacji FTAM, grupa *Storage* jest zdefiniowana do czytania wartości parametrów ustawionych podczas ostatniego odwołania do pliku, grupa *Security* związana jest z prawami dostępu, natomiast grupa *Private* zawiera wyłącznie prywatne atrybuty.

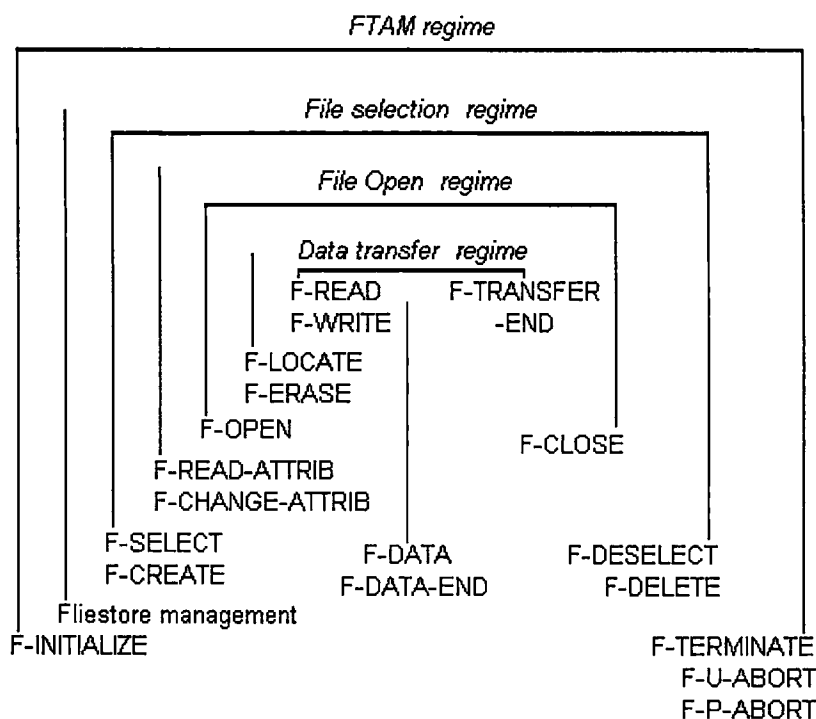
13.3.3 Reżimy (ang. *Regimes*)

Transakcja FTAM jest zakładana poprzez wywołanie funkcji FTAM w krokach (lub zgrupowanej sekwencji kroków) nazywanych reżimami. Reżim (ang. *regime*) jest okresem czasu,

podczas którego możliwe jest wywołanie określonej grupy funkcji FTAM. Reżimy określają działania, które mogą wystąpić w danym czasie i dlatego tworzą protokoły dla urządzeń FTAM (rys.13.6).

Reżimy są zgrupowane w porządku postępującym, tzn. aplikacje muszą przejść przez zewnętrzne reżimy po to, aby osiągnąć dostęp do wewnętrznych, a także muszą wprowadzać reżimy we właściwej kolejności podczas opuszczania wewnętrznych reżimów, w celu uzyskania dostępu do reżimów zewnętrznych. Wyróżnia się cztery podstawowe reżimy (tryby pracy):

- Reżimy FTAM (ang. *FTAM regime*). Reżimy FTAM żądają usług FTAM po to, aby nawiązać komunikację. Reżim FTAM, także nazywany asocjacją FTAM (ang. *FTAM Association*) lub reżimem inicjalizującym FTAM (ang. *FTAM Initialisation regime*) jest reżimem zewnętrznym.
- Reżim wyboru pliku (ang. *File Selection regime*) wyszukuje i wybiera określony, wyspecyfikowany plik. Jest możliwa zmiana oraz czytanie atrybutów wybranego pliku podczas tego reżimu.
- Reżim otwarcia pliku (ang. *File Open Regime*). Podczas reżimu otwarcia pliku zawartość pliku jest udostępniana do czytania, zapisywania, lokalizowania i wymazywania.
- Reżim transferu danych (ang. *Data Transfer Regime*) jest najbardziej wewnętrznym reżimem i odpowiada za transfer aktualnych danych.



Rys.13.6 Reżimy FTAM i związane z nimi usługi

Jak zdefiniowano w specyfikacji FTAM, operacje na plikach są możliwe wyłącznie wówczas, jeśli uprzednio przygotowane było odpowiednie środowisko pracy, nazywane reżimem. Zdefiniowane są cztery różnie zagnieżdżone reżimy, co pokazuje rys.13.6. Rysunek ten wskazuje także na operacje dozwolone w każdym z reżimów.

W zakresie jednego reżimu dozwolone jest tylko jedno żądanie (instancja), co oznacza, że inicjator (ang. *initiator*) otwierając plik w wirtualnym systemie plików (ang. *responder*) poprzez założenie asocjacji FTAM, reżimu wyboru pliku oraz reżimu otwarcia pliku, nie może otworzyć innego pliku w tym samym systemie plików (ang. *filestore*). Dlatego też w tym samym czasie pomiędzy tą samą parą uczestników (partnerów) inicjator - responder może być transferowany tylko jeden plik. Jednak różni inicjatorzy (ang. *initiators*) mogą transferować pliki w tym samym czasie z tym samym systemem plików (ang. *responders*) oraz ci sami użytkownicy mogą otwierać kilka plików na różnych wirtualnych składnicach plików.

Ustanowienie reżimów FTAM, lub inaczej założenie asocjacji, zapewnia aplikacjom odpowiednie środowisko pracy nazywane często środowiskiem operacyjnym. Środowisko to jest wymagane przy odwołaniach do usług FTAM zgrupowanych w jednostki funkcyjne FU (ang. *Functional Units*) i klasy usług (ang. *Service*). Asocjacja jest zakładana przez wywołanie usługi F-INITIATE ustanawiającej zadania dla inicjatora (ang. *initiator*) i responder'a, które są jednakowe dla tej usługi i niektórych innych parametrów takich jak: rodzaj dostępu, typ konta (ang. *account*) oraz hasło (ang. *password*). Reżimy mogą kończyć się w sposób uporządkowany, w przypadku wywołania funkcji F-TERMINATE, lub też mogą kończyć się przez przerwanie (ang. *abort*). Przerwanie może być zgłoszone przez jednego z dwóch użytkowników FTAM przez wywołanie usługi F-U-ABORT lub wywołanie F-P-ABORT, jak i też przez zgłoszenie równoważnego sygnału od warstwy prezentacji. Operacje na systemie plików w reżimie FTAM nie są w standardzie zdefiniowane.

Reżim zarządzania systemem plików (ang. *File selection regime*) pozwala inicjatorowi na wykonywanie ogólnych operacji na plikach bez udostępniania ich kontekstu. Wykorzystywany on jest w celu zidentyfikowania określonego pliku w systemie plików respondera. Reżim ten jest inicjalizowany zarówno przez użycie usługi:

- F-CREATE - jeśli jest tworzony nowy plik, lub niektóre specjalne atrybuty muszą być zmienione, jak i
- F-SELECT - jeśli plik istnieje.

W reżimie tym możliwe jest manipulowanie atrybutami plików (F-READ-ATTR oraz F-CHANGE-ATTR). Zamknięcie reżimu jest dokonywane przez wywołanie usługi F-DELETE, która wymazuje plik i wszystkie jego atrybuty (włączając w to nazwę pliku), jak i przez F-DESELECT, który zachowuje statusy pliku. Dostęp do zawartości pliku oraz informacji o pliku (ang. *structuring information*) jest dozwolony wyłącznie w reżimie otwarcia pliku (ang. *File open regime*). Inicjator jest upoważniony do wyboru odpowiedniego podzbioru plików (F-LOCATE), na których mogą być wykonywane operacje wymazywania (F-ERASE) wybranych podzbiorów plików oraz czytania lub zapisywania wybranych podzbiorów plików. *File open regime* jest otwierany wywołaniem usługi F-OPEN i zamykany przez wywołanie F-CLOSE. Błędy zgłaszane w czasie istnienia reżimu są wykorzystywane w procesie decydowania o rozpoczęciu akcji odzyskiwania utraconych informacji (w systemie FTAM są zaimplementowane standardowe mechanizmy odzyskiwania utraconych informacji).

Przesyłanie jednostek danych DU (ang. *Data Units*) jest możliwe wewnątrz reżimu transferu danych (ang. *Data transfer regime*). Dane mogą być przesyłane od inicjatora do *responder*'a lub w kierunku odwrotnym (F-READ, F-WRITE). Jednostki udostępniania danych pliku FADU, wskazujące na jednostki danych DU, są przekazywane jako parametry wyżej wymienionych usług.

13.5 Jednostki aplikacji AE (ang. *Application Entities*)

Jednostka aplikacji AE jest częścią składową programu użytkownika (aplikacji), mającą unikalny identyfikator i współdziałającą z sieciami komunikacyjnymi OSI. Jednostkami AE mogą być inicjator i responder FTAM.

Przed przystąpieniem do lokalizacji, udostępniania lub obsługi (czytanie, zapisywanie) plików FTAM, należy nawiązać komunikację (zainicjalizować reżimy FTAM) z jednostką aplikacji AE *responder*'a, który obsługuje pożądaną system plików. Identyfikowanie jednostek aplikacji AE odbywa się zarówno przez prezentacyjny adres jednostkowy (*P_address*) jak i przez jego wyróżnioną nazwę ścieżki dostępu (*Ae_dir_name*). Adresy te są definiowane podczas konfiguracji systemu oraz podczas dodawania nowych węzłów do sieci. Informację o adresach można uzyskać od administratora systemu.

Po ustanowieniu reżimów FTAM można przejść do obsługi i udostępniania plików w systemie. Możliwe jest wykonanie dwóch typów akcji na plikach VFS: działania na kompletnych plikach oraz akcje udostępniania plików np. możliwe jest utworzenie i kasowanie kompletnych plików lub udostępnianie części zawartości pliku.

13.6 Wirtualna struktura pliku FTAM

Plik wirtualny zdefiniowany w specyfikacji FTAM ma strukturę hierarchiczną bazującą na następujących aksjomatach:

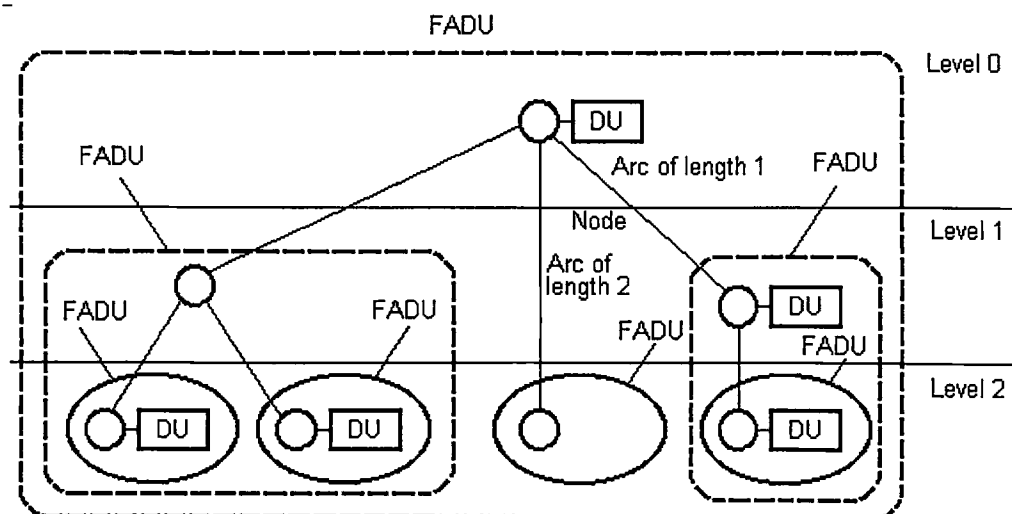
- struktura dostępu do pliku jest drzewem uporządkowanym,
- każdy węzeł w drzewie może mieć lub nie mieć związanej z nim jednostki danych DU (ang. *Data Unit*),
- każdy węzeł w drzewie umożliwia dostęp do niżej leżącego poddrzewa, przy czym każde poddrzewo jest związane z jednostką FADU (ang. *File Access Data Unit*), a dostęp do całego pliku jest możliwy poprzez węzeł podstawowy (ang. *Root Node*),
- węzły mogą mieć lub też nie mieć nazwy z nimi związanej,
- wysokość drzewa i liczba węzłów pochodnych (ang. *children*) dla każdego węzła jest nieograniczona.

Istnieją trzy powszechnie używane struktury plików: niestrukuralne U (ang. *Unstructured*), płaskie F (ang. *Flat*) oraz hierarchiczne H (ang. *Hierarchical*). Jak wskazuje nazwa, w plikach niestrukuralnych nie istnieje widoczna, uporządkowana struktura. Przykładem tego może być plik tekstowy, który ma strukturę z punktu widzenia użytkownika, ale jest ona niewidoczna dla systemu plików i wyłącznie pełny plik może być czytany lub zapisywany.

Plik typu F (płaski) zawiera uporządkowaną sekwencję rekordów (jednostek danych), które mogą mieć zmienną długość i typ. Normalnie każdy rekord zawiera identyfikator (ang. *label*), jakkolwiek może on zawierać także wiele elementów danych identyfikowanych przez użytkownika, lecz nie widocznych przez system plików.

Plik typu H (hierarchicznego) dodatkowo, obok identyfikowalnych rekordów, ma towarzyszącą mu strukturę. Jest to struktura drzewa z licznymi węzłami (rozgałęzieniami). Zazwyczaj, każdy węzeł gałęzi ma identyfikator oraz związany z nim rekord danych. Węzły są lokalizowane w znanym porządku, tak że jest możliwe zidentyfikowanie węzła poprzez jego pozycję względem pozostałych węzłów. Alternatywnie, jeśli każdy węzeł ma towarzyszący mu identyfikator, to węzeł może być również identyfikowany przez podanie nazwy ścieżkowej od głównego węzła (ang. *root*). Jak łatwo się domyśleć, pozostałe struktury są specjalnymi przypadkami struktury hierarchicznej. Wirtualny system plików wybrany przez ISO bazuje na strukturze hierarchicznej. Wzajemne powiązanie struktur oraz stosowana terminologia jest przedstawiona na rysunku 14.5 (MAP/TOP).

Struktura drzewa zawiera pojedynczy węzeł podstawowy tzw. *root* z wewnętrznym węzłem oraz rozgałęzieniami połączonymi bezpośrednimi łukami. Węzeł może należeć wyłącznie do jednego poziomu. Każdy węzeł daje dostęp do jednego poddrzewa, które jest znane jako FADU (ang. *File Access Data Unit*). Dane pliku w systemie plików są utrzymywane w jednym lub wielu jednostkach danych DU (ang. *Data Units*). Do węzła może być przypisana tylko jedna jednostka danych DU. Oznacza to, że jednostka DU może być udostępniona przez lokalizację węzła FADU. DU jest określonym obiektem danych (np. skalar, wektor itd.) i zawiera elementy danych atomowych nazywanych elementami danych DE (ang. *Data Elements*). Z każdym elementem danych DE stowarzyszona jest abstrakcyjna syntaktyka (ang. *character, octet, integer etc.*) i wszystkie elementy w DU są wzajemnie zależne. Normalna zależność elementów występuje w formie drzewa, chociaż alternatywnie mogą być używane pojedyncze elementy danych lub wektory.



Rys.13.7 Ogólna struktura pliku wirtualnego

Drzewo jest przecięte w poprzek w takiej kolejności jak to przedstawiono na powyższym rysunku (rys.13.7). Udostępnione jednostki danych DU są przenoszone do jednostek prezentacji PE

(ang. *Presentation Entity*) w tej samej kolejności jak pokazana na rysunku. Jednostki prezentacji traktują każdy element DU jako niezależny oraz używają odpowiedniej (pertraktowanej) syntaktyki transferu w celu zachowania ich względnego porządku.

Zależnie od systemu plików, poszczególne akcje są wywoływane przez odpowiednie usługi podstawowe (prymitywy). Usługi te zawierają: tworzenie, otwieranie, zamykanie oraz kasowanie kompletnych plików, a także lokalizowanie, czytanie, wstawianie, rozszerzanie oraz wymazywanie jednostek danych DU w pliku.

13.7 Typy dokumentów w systemie FTAM

Typ dokumentu FTAM odpowiada typom plików zaimplementowanych w większości współcześnie wykorzystywanych systemów operacyjnych. W rzeczywistych systemach plików powszechne typy plików składają się z tekstowych plików strumieniowych, tekstowych plików zorientowanych na rekordy, plików binarnych oraz katalogów, które zawierają pliki. Typy te są reprezentowane przez następujące typy dokumentów FTAM:

- FTAM-1 niestrukturyzowany, zorientowany strumieniowo plik tekstowy,
- FTAM-2 tekstowy plik zorientowany na rekordy,
- FTAM-3 niestrukturyzowany, zorientowany na strumień plik binarny,
- INTAP-1 pliki rekordowe,
- NBS-9 katalogi, które zawierają pliki.

Typy dokumentów definiują także semantykę dla określonych klas plików. Ponadto typ dokumentu definiuje rodzaj zawartości pliku przez określenie *ograniczenia zbioru* (ang. *constraint set*) oraz możliwych typów danych w pliku. Przykłady typów pliku zawierają pliki tekstowe, pliki binarne, pliki sekwencyjne oraz katalogi.

IV. Architektury sieci w CIM

14. Architektury sieci w zautomatyzowanym wytwarzaniu i automatyce przemysłowej

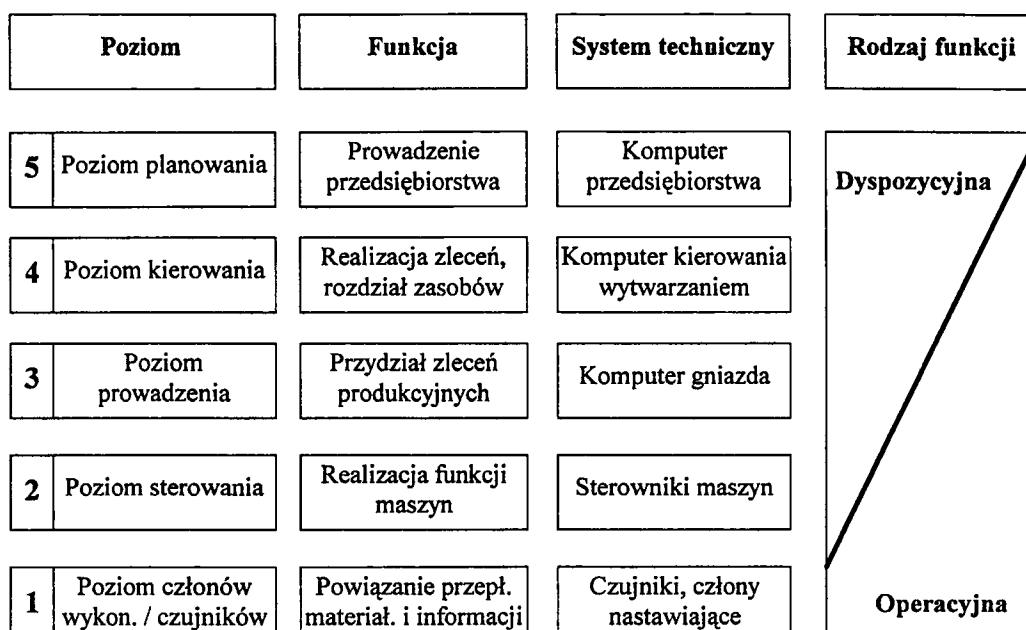
14.1 Wstęp

Komunikacja w obszarze wytwarzania, a szczególnie w produkcji zautomatyzowanej, odgrywa już obecnie i będzie odgrywała w przyszłości bardzo istotną rolę. System komunikacyjny decyduje bowiem w dużej mierze o efektywności i niezawodności działań w całym przedsiębiorstwie. Ze względu na dużą ilość stosowanych w systemach wytwarzania technik, dotyczących zarówno sprzętu jak i oprogramowania, a także ze względu na różnorodne wymagania w poszczególnych obszarach komputerowo wspomaganego wytwarzania, powstaje konieczność zastosowania różnych, odpowiadających tym wymaganiom sieci komunikacyjnych. I tak np. sieciom danych w obszarze komunikacji biurowej są stawiane zupełnie inne wymagania aniżeli sieciom komunikacyjnym w obszarze wytwarzania. W obszarze biurowym sieci komunikacyjne są bowiem stosowane w pierwszym rzędzie do umożliwienia dostępu do zbiorów z różnych komputerów oraz transmisji tych zbiorów. Te systemy komunikacyjne muszą zatem przede wszystkim spełniać wysokie wymagania w zakresie ochrony danych. Natomiast w obszarze komputerowo wspomaganego wytwarzania sieci komunikacyjne są stosowane głównie do sterowania programowalnych maszyn i urządzeń wytwórczych. Tym samym występują tutaj również wysokie wymagania czasowe i bezwzględnie konieczna jest bezbłędna transmisja danych.

Realizacja zautomatyzowanego wytwarzania dla zróżnicowanego wariantowo programu produkcyjnego wymaga zastosowania zintegrowanego sterowania komputerowego oraz informacyjnego powiązania wszystkich składników, które na różnych poziomach funkcyjnych obszaru wytwarzania realizują zadania sterowania, kierowania, kontroli i nadzorowania. Organizacja przebiegów przetwarzania informacji ma bowiem rozstrzygający wpływ na efektywność, elastyczność i niezawodność całego systemu wytwórczego, a ze wzrostem stopnia jego złożoności wyraźnie zwiększa się ilość możliwych form jego konfiguracji.

Informatyczną strukturę obszaru wytwarzania można dobrze zobrazować w postaci hierarchicznie podporządkowanych poziomów (rys.14.1). I tak na poziomie kierowania komputer kierowania wytwarzaniem koordynuje i nadzoruje niezbędne dla przebiegu wytwarzania przepływy materiałów i informacji pomiędzy poszczególnymi systemami wytwórczymi i gniazdami. Komputer nadrzędny wytwarzania wykonuje natomiast optymalizację realizacji zleceń, inicjuje strategie zastępcze w przypadkach awarii i dostarcza informacji o obciążeniu instalacji produkcyjnej. Z nadrzędnego systemu planowania i sterowania produkcji przesyłane są dane o planowanej realizacji zleceń. Na poziomie prowadzenia odbywa się natomiast koordynacja i sterowanie różnych maszyn i urządzeń wytwórczych poziomu sterowania, takich jak obrabiarki NC, systemy transportu materiałów i narzędzi, które muszą być zasilane w odpowiednie dane sterujące. Stosowane do tego celu komputery gniazd mogą być wykorzystywane również do obsługi poszczególnych składników

systemów wytwórczych, np. podczas ich ustawiania, zarządzania programami NC, zaopatrywania w narzędzia, uchwyty itp.



Rys.14.1 Model struktury podziału przedsiębiorstwa na poziomy

Na poziomie planowania realizowane są nadrzędne zadania całego przedsiębiorstwa. Zasadniczym zadaniem na tym poziomie jest spełnianie postawionych celów przedsiębiorstwa (planowanie długoterminowe). Systemem technicznym na poziomie planowania przedsiębiorstwa (zakładu) może być np. komputer zakładowy. Na tym poziomie realizowane są takie funkcje jak opracowanie konstrukcji (np. system CAD) lub planowanie i sterowanie produkcją (np. system PPC).

Na poziomie kierowania odbywa się natomiast koordynacja produkcji i wytwarzania, jak również nadzorowanie i sterowanie realizacji zleceń, łącznie z zapewnieniem i rezerwowaniem zasobów wymaganych dla poszczególnych zleceń. Typowymi zadaniami realizowanymi na poziomie kierowania są: planowanie produkcji, planowanie zdolności produkcyjnych, planowanie zasobów itp. Technicznym systemem na tym poziomie jest np. komputer kierowania wytwarzaniem. W obszarze biurowym na poziomie kierowania są realizowane także takie funkcje przedsiębiorstwa jak planowanie pracy (np. systemy CAP).

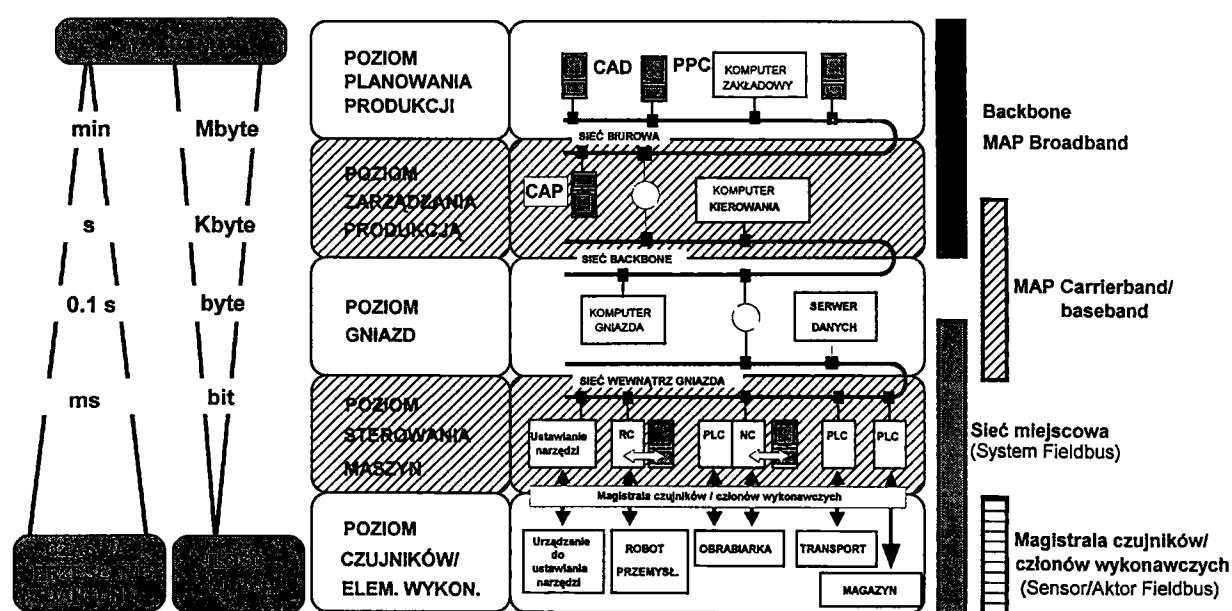
Analiza zasobów wymaganych dla realizacji poszczególnych zleceń wytwórczych i pomiarowych, żądanie tych zasobów, tworzenie kolejek oczekiwania zleceń, przydział zleceń do poszczególnych stanowisk, nadzorowanie realizacji zleceń oraz sporządzanie raportów o statusie zleceń dla poziomu kierowania są typowymi zadaniami poziomu prowadzenia, który w zasadzie odpowiada zadaniom mistrza. Realizacjami technicznymi poziomu prowadzenia, które są zestawione pod pojęciem CAM, są np. komputery gniazd, zintegrowane systemy DNC/PDA /MDA (ang. *Direct Numerical Control / Production Data Acquisition / Machine Data Acquisition*) i stanowiska kierowania wytwarzaniem.

Kierowanie, nadzorowanie i koordynowanie działalności jednej lub wielu zautomatyzowanych maszyn lub urządzeń produkcyjnych w celu obróbki przedmiotów, sterowania zleceniami montażowymi i pomiarowymi oraz meldowanie do systemu nadrzędnego o stanie urządzeń są natomiast zadaniami, które są przyporządkowane do poziomu sterowania. Technicznymi realizacjami na poziomie sterowania, który jest zawarty również w obszarze CAM, są układy sterowania zautomatyzowanych maszyn i urządzeń produkcyjnych.

Oddziaływanie pomiędzy procesem i układem sterowania, które jest realizowane na poziomie czujników i członów wykonawczych, odbywa się przez odpowiednie elementy wykonawcze i czujniki maszyn i urządzeń.

Porównanie pionowych i poziomych przepływów informacji na różnych poziomach przedsiębiorstwa wykazuje różne wymagania odnośnie do żądanej szybkości transmisji danych, przesyłanych ilości danych i przebiegów czasowych transmisji wzgl. reakcji (rys.14.2). Dlatego też w przemyśle wytwórczym ewolucyjnie utworzyły się cztery następujące segmenty komunikacyjne:

- Dla komunikacji na poziomie czujników / członów wykonawczych, tzn. do wymiany danych pomiędzy członami wykonawczymi, czujnikami i układami sterowania stosowane są tzw. sieci miejscowe członów wykonawczych / czujników (ang. *Aktor / Sensor Fieldbus*).
- Powiązanie informatyczne układów sterowania urządzeń automatyzacji z hierarchicznie nadrzędnymi systemami poziomu kierowania odbywa się natomiast za pomocą lokalnych sieci gniazd wzgl. systemowych sieci miejscowych (ang. *System Fieldbus*).
- Integracja różnych części zakładu (hal produkcyjnych, budynków biurowych) i ich urządzeń komunikacyjnych odbywa się na poziomie kierowania za pomocą lokalnych sieci szkieletowych (ang. *Backbone*).
- Do połączenia w sieć technicznych i administracyjnych działów biurowych poziomu planowania i kierowania są natomiast dzisiaj stosowane lokalne sieci biurowe.



Rys.14.2 Segmenty komunikacyjne w obszarze wytwarzania (wg Weck'a)

Jeżeli na poziomie czujników / członów wykonawczych w dyskretne obwody regulacji są włączone różne nastawniki i układy pomiarowe, to pomiędzy nimi i odpowiednimi urządzeniami sterującymi muszą być cyklicznie transmitowane dane. Czasy cyklu muszą być przy tym znacznie mniejsze aniżeli stałe czasowe charakterystyczne dla przebiegu regulacji. Dlatego też na poziomie członów wykonawczych / czujników są transmitowane i przetwarzane krótkie informacje o długości niewielu bitów (ok. 10 do 50 bitów) w czasie pojedynczych milisekund (np. < 10 ms). Jako urządzenia komunikacyjne w ostatnich latach na poziomie członów wykonawczych / czujników stosowane są specjalne sieci miejscowe (*Sensorbus*), które są też określane jako magistrale członów wykonawczych / czujników. W tab.14.1 zamieszczono ogólny przegląd właściwości segmentów komunikacyjnych sieci stosowanych w technikach wytwarzania.

Rozpatrując różne poziomy hierarchicznej struktury przedsiębiorstwa ze względu na ich powiązanie informacyjne, uzyskuje się różnorodne wymagania odnośnie ilości danych i przebiegów czasowych. Na poziomie sterowania muszą być transmitowane małe ilości danych w krótkim czasie. Na wyższych poziomach funkcyjnych zwiększają się natomiast strumienie danych przesyłanych w kierunku pionowym. Wraz ze wzrostem stopnia złożoności systemów wytwarzania następuje progresywny przyrost ilości danych. Ponadto żądania elastyczności systemu wytwórczego pociągają za sobą również wzrost wymagań czasowych dotyczących aktualności tych danych. Dlatego też dla uwzględnienia tego rodzaju wymagań konieczny jest wybór takich sieci komunikacyjnych, które umożliwią zintegrowany przepływ informacji pomiędzy poszczególnymi podsystemami.

Tab.14.1 Porównanie właściwości systemów komunikacyjnych stosowanych w wytwarzaniu

	Sieć czujników/ członów wykonaw. (<i>Sensorbus</i>)	Sieć gniazda (sieć miejscowa <i>Fieldbus</i>)	Lokalna sieć szkieletowa	Lokalna sieć biurowa
Znamionowa szybkość transmisji danych	300 kbit/s do 2 Mbit/s	9.6 kbit/s do 1 Mbit/s (wyj. 10 Mbit/s)	do 100 Mbit/s	4 Mbit/s do 16 Mbit/s
Czas reakcji	1.4 do 10 ms (wyjątkowo 40 ms)	>20 ms	-	-
Długość kabli	do 500 m	500m do 120 m	3 km do 10km	150 m do 2 km
Szacunk. koszt przyłączenia użytkownika	8 zł do 100 zł	800 zł do 3400 zł	430 zł do 1700 zł	170 zł do 1700 zł
Przykłady	SERCOS, ABUS, PROFIBUS-DP InterBus-S, ASI, CAN	SINEC H1 (Eth.) MEßBUS, PROFIBUS FMS, FIP i FAIS, MAP/MMS	sieć FDDI, protokół MAP	protokół TOP, sieć Token-Ring, Ethernet (TCP/IP), Novellnet (IPX/SPX)

Na poziomie sterowania, pomiędzy różnymi użytkownikami komunikacji, wymieniane są takie informacje jak np. komendy i alarmy, których długość zawiera się przeciętnie w zakresie od 100 do 500 Byte. Zwykle urządzeniom komunikacyjnym nie są tutaj stawiane wysokie wymagania odnośnie do przebiegów czasowych transmisji, ponieważ ze względu na dużą inteligencję układów sterowania, operacje krytyczne czasowo lub też istotne dla bezpieczeństwa są wykonywane lokalnie. I tak np. obrabiarka NC, po rozpoznanej i zgłoszonej do komputera gniazda kolizji, nie oczekuje na rozkaz wyłączenia lecz wyłącza się samoczynnie. Dlatego też transmisja alarmu nie musi się odbywać w przeciągu pojedynczych milisekund, lecz może się zawierać w obszarze wielu dziesiątych sekundy. Dlatego dla transmisji danych pomiędzy układem sterowania NC i komputerem gniazda ze względów technologicznych na poziomie sterowania wymagana jest nominalna szybkość transmisji danych rzędu 40 kbit/s. Obecnie stosowanymi sieciami gniazd są np. SINEC H1 firmy Siemens, wzgl. takie systemowe sieci miejscowe jak niemieckie sieci MEßBUS i PROFIBUS, a także sieć francuska FIP i japońska FAIS.

Szkieletowe sieci lokalne łączą natomiast różne urządzenia komunikacyjne danych w przedsiębiorstwie. Jako sieci szkieletowe stosowane są np. lokalna sieć SINEC H3 firmy Siemens lub MODNET 3 firmy AEG. Do tych lokalnych sieci szkieletowych są przyłączane sieci biurowe i sieci gniazd wzgl. systemowe sieci miejscowe za pomocą elementów sprzęgających (*Gateway's*, *Router*, *Bridge's*).

W obszarach biur większych przedsiębiorstw wymiana informacji nie jest zwykle zorientowana na telegramy, tak jak na poziomie *człony wykonawcze / czujniki* i na poziomie *sterowania*, lecz składa się przeważnie z transmisji danych i wysyłania poczty elektronicznej (E-Mail). Dlatego też poziome i pionowe strumienie danych w obszarze sieci biurowej są znacznie obszerniejsze. Natomiast w przeciwieństwie do tego spadają tutaj znacznie wymagania odnośnie do przebiegu czasów reakcji systemu transmisji danych.

Jako rozwiązania sieci biurowych są np. stosowane takie produkty jak DECnet, IBM Token-Ring lub Novellnet (w połączeniu z Ethernet), które mogą pracować z maksymalną szybkością transmisji danych 16 Mbit/s i dopuszczają przesyłanie telegramów o maksymalnej długości 8192 byte.

Przy wyborze sieci komunikacyjnej do zastosowania w określonym przedsiębiorstwie produkcyjnym, oprócz uwzględnienia wymagań funkcjonalnych, należy się kierować również względami otwartości systemu oraz jego kosztami. I tak np. uzyskanie możliwości zestawiania poszczególnych składników automatyzacji (komputerów, sterowników itp.), które pochodzą od różnych producentów, w zintegrowany system sieciowy, wymaga zachowania jednakowych reguł podczas przebiegu komunikacji danych w przedsiębiorstwie. Różnorodne wymagania komunikacyjne na poszczególnych poziomach przedsiębiorstwa znajdują swoje uwzględnienie w odpowiednich standardach komunikacyjnych i protokołach takich jak np. TOP, MAP, RS 485 itp. Ocena i wybór odpowiedniego systemu komunikacyjnego jest ważnym zadaniem podczas

planowania sieci, co jest związane ze szczegółową analizą wymagań wynikających z zastosowania w wytwarzaniu. Należy tutaj także zaznaczyć, że dobór systemu komunikacyjnego musi być zawsze okrojony do określonego zakresu zastosowań, ponieważ w zasadzie nie istnieją systemy uniwersalne.

Oprócz stosowanych w obszarze biurowym lokalnych sieci komputerowych LAN coraz większego znaczenia nabierają przemysłowe sieci komunikacyjne typu *Fieldbus*, które są stosowane na najniższych poziomach struktury hierarchicznej przedsiębiorstwa tzn. w obszarze czujników i członów wykonawczych oraz sterowników (NC, PLC, RC) i komputerów gniazd. Termin *Fieldbus* został wprowadzony przez IEC i jest on cyfrowym odpowiednikiem rozpowszechnionego w świecie analogowego interfejsu 4-20 mA, który to interfejs jest obciążony taką wadą, że może transmitować każdorazowo tylko jedną jedyną analogową wartość sygnału w jednym kierunku. Wskutek wprowadzenia inteligentnych czujników i członów wykonawczych w obszarze procesów powstało zapotrzebowanie na bezpośredni, dwukierunkowy, cyfrowy system komunikacyjny, umożliwiający włączenie poszczególnych urządzeń z tego obszaru do nadrzędnego systemu sterowania. Ten system komunikacyjny powinien spełniać takie wymagania jak: zwiększona szybkość i niezawodność transmisji, zredukowanie nakładów na okablowanie, umożliwienie zdalnej kalibracji, a także umożliwiać realizację funkcji diagnostycznych i nadzorowania.

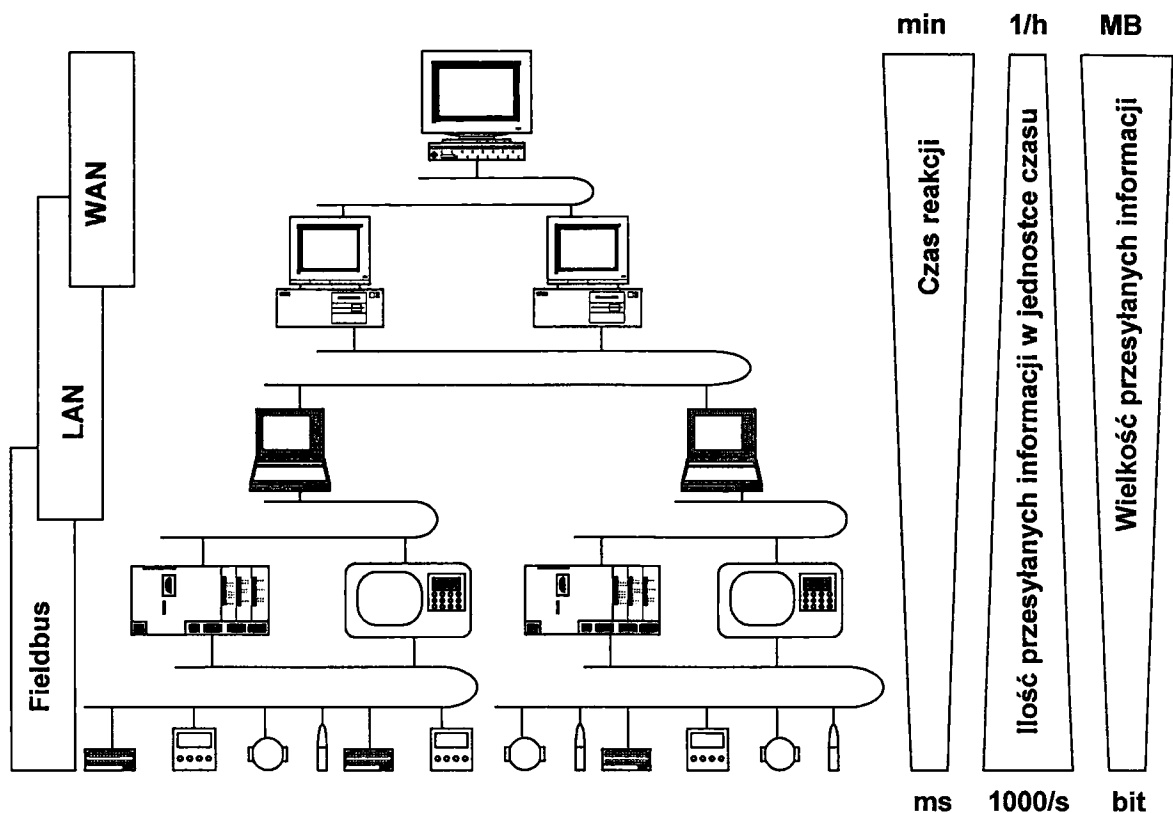
Wielu producentów systemów stosowanych w automatyzacji procesów wytwarzania i automatyce przemysłowej opracowało firmowe, cyfrowe protokoły komunikacyjne, które jednak przez użytkowników są stosowane niechętnie, gdyż powodują ich uzależnienie od tych producentów. Dlatego też również w tym obszarze są inicjowane krajowe i międzynarodowe działania w celu utworzenia jednolitego standardu komunikacyjnego dla obszaru procesów, przy czym powinna być w nim zapewniona integracja ze strukturami komunikacyjnymi nadrzędnych systemów automatyzacji.

14.2 Architektura sieci w komunikacyjnej w zautomatyzowanym wytwarzaniu

Architektura sieci w zautomatyzowanym wytwarzaniu i automatyce przemysłowej jest zdeterminowana wymaganiami stawianymi systemom komunikacyjnym. Podstawowymi kryteriami wyboru sieci oraz budowy kompleksowych systemów komunikacyjnych są:

- czas reakcji,
- wielkości przesyłanych informacji,
- częstotliwość występowania informacji,
- funkcjonalność,
- oddziaływanie zakłóceń,
- bezpieczeństwo danych,
- ilość urządzeń,

- koszt urządzeń.



Rys.14.3 Ogólny model struktury sieciowej w przedsiębiorstwie przemysłowym

Rysunek 14.3 przedstawia uogólnioną ideę struktury sieciowej w przedsiębiorstwie przemysłowym. Począwszy od samego szczytu piramidy – obszaru zarządzania firmą, do najniższego poziomu – urządzeń wykonawczych i czujników, pojawiają się specyficzne wymagania związane z poszczególnymi obszarami.

Obszar zarządzania firmą charakteryzuje się: niekrytycznymi czasami reakcji (rzędu 10 s - 10 min), dużymi wielkościami przesyłanych pakietów danych (rzędu 1-100 MB), przy niskiej częstotliwości ich występowania (rzędu 1/h). W obszarze tym nie występują silne zakłócenia elektromagnetyczne, a bezpieczeństwo danych przesyłanych na tym poziomie przedsiębiorstwa również nie jest bardzo krytyczne. Działania wykonywane na: danych, obszarach pamięci oraz plikach w tym obszarze wymagają dużej funkcjonalności protokołu oraz różnorodnych usług.

W odróżnieniu do najwyższego poziomu przedsiębiorstwa, obszar czujników i urządzeń wykonawczych charakteryzuje się całkiem innymi wymaganiami. Czasy reakcji w tym obszarze są bardzo krytyczne i wahają się na poziomie kilku milisekund, pakiety przesyłanych danych są niewielkie (rzędu kilku do kilkudziesięciu bitów), przy bardzo częstym ich występowaniu. W obszarze tym mogą także występować bardzo silne

zakłócenia elektromagnetyczne, przy jednoczesnych wysokich wymaganiach bezpieczeństwa danych.

W obszarach znajdujących się pomiędzy warstwą zarządzania przedsiębiorstwem a warstwą czujników i urządzeń wykonawczych, wymagania związane z kryteriami wymienionymi wyżej, przyjmują wartości pośrednie.

Tak różnorodne i skrajne wartości parametrów pojawiających się w całym obszarze wytwarzania, zmuszają użytkowników oraz producentów urządzeń automatyki przemysłowej do podzielenia systemów komunikacyjnych stosowanych w przedsiębiorstwie na kilka warstw. Każda warstwa ma swoje własne zoptymalizowane rozwiązania sieciowe. Na rysunku 1 przedstawiono trzy typy systemów komunikacyjnych obejmujących cały obszar przedsiębiorstwa, tzn. sieci rozległe WAN (ang. *Wide Area Network*), sieci lokalne LAN (ang. *Local Area Network*) oraz sieci miejscowe FAN (ang. *Field Area Network*). W obrębie poszczególnych typów pojawiają się standardy dedykowane do jeszcze węższych zastosowań, np. w obszarze działania sieci *Fieldbus* powstały takie standardy jak PROFIBUS-DP dedykowany do zbierania bardzo szybko prostych zmiennych, czy PROFIBUS-FMS, który jest przeznaczony do łączenia sterowników i inteligentnych urządzeń sterujących, natomiast InterBus-S jest szczególnie przydatny do sterowania silnikami, a AS-Interface (ASI) do zbierania sygnałów binarnych.

Jak zatem widać struktura systemu sieciowego w przedsiębiorstwie jest ściśle związana z wymaganiami stawianymi w danym obszarze jego działalności i jest ona strukturą heterogeniczną.

15. Integracja heterogenicznych systemów komunikacyjnych

Heterogeniczna struktura systemu komunikacyjnego w przedsiębiorstwie wymaga stosowania elementów integrujących poszczególne podsystemy sieciowe. Integracja taka pozwala na budowę kompleksowego systemu komunikacyjnego, w którym przepływ danych jest swobodny zarówno w poziomie jak i w pionie.

Przepływ danych w obu kierunkach (pionie i poziomie struktury przedsiębiorstwa) musi być również ekonomicznie oraz funkcjonalnie uzasadniony. W rozdziale 14.2 zasygnalizowano zagadnienie wąskiej specyfiki i naturalnych zalet (determinujących obszar stosowania) poszczególnych rozwiązań sieciowych. Aby wykorzystać w pełni zalety poszczególnych rozwiązań sieciowych, logicznie jednak zachowując ciągłość systemu komunikacyjnego, producenci zaproponowali kilka elementów integrujących poszczególne standardy sieciowe. Przykładem tego mogą być elementy wiążące sieci PROFIBUS-DP i AS-Interface, czy InterBus-S i AS-Interface, które pozwalają wykorzystać efektywność sieci AS-Interface przy sygnałach binarnych oraz możliwości efektywnego zbierania sygnałów analogowych lub cyfrowych z większego obszaru, czy też synchronizowania napędów przy zastosowaniu sieci PROFIBUS-DP lub InterBus-S. Inne rozwiązania elementów integrujących poszczególne standardy sieciowe, oferowane przez producentów to: moduł integrujący sieć PROFIBUS-DP z siecią PROFIBUS-PA (dedykowaną dla automatyzacji procesów) czy sieć InterBus-S z siecią TCP/IP. W aplikacjach spotyka się bardzo często moduły inteligentne z dwoma a nawet kilkoma procesorami komunikacyjnymi, odpowiadającymi za komunikację z poszczególnymi sieciami. Przykładowym rozwiązaniem takich modułów może być połączenie jednego sterownika PLC lub komputera PC poprzez dwie karty komunikacyjne, np. do sieci PROFIBUS-DP oraz PROFIBUS-FMS lub do sieci PROFIBUS-FMS oraz sieci InterBus-S lub TCP/IP itp. W innych takich rozwiązaniach obiekty komunikacyjne mogą być przekazywane między oboma systemami komunikacyjnymi za pomocą:

- wspólnego obszaru pamięci,
- *server'a* danych np. DDE czy
- dedykowanego *driver'a*.

W takich rozwiązaniach bardzo często dane przechodząc z jednego systemu komunikacyjnego do drugiego są poddawane obróbce „w locie” (pakowaniu, grupowaniu czy obróbce statystycznej). Najczęściej takie rozwiązania stosowane są w przekazywaniu danych do systemów nadrzędnych np. SCADA (wizualizacji i kontroli).

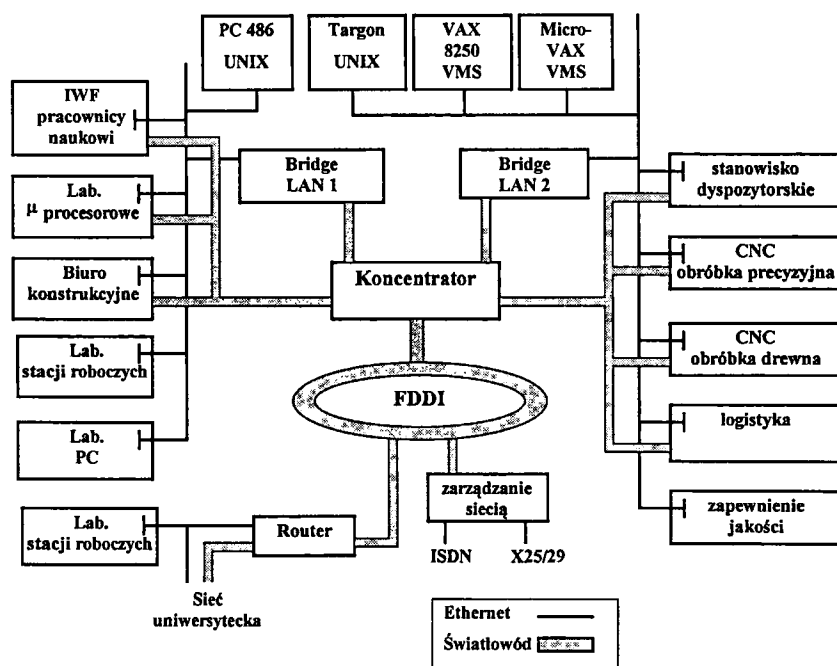
16. Przykłady pilotowych instalacji CIM

Komputerowo zintegrowane wytwarzanie CIM szczególnie dynamicznie rozwija się dzięki realizacji projektów badawczych, które są prowadzone w tym zakresie na wyższych uczelniach technicznych. W celu szybkiego wykorzystania w praktyce przemysłowej rozwiązań, które są proponowane w ramach tych projektów, znaczącą rolę odgrywają instalacje pilotażowe. Choć prezentowane tutaj przykłady, w odniesieniu do rozległości zagadnień jakie występują w CIM, są jedynie wycinkiem dużych rozwiązań przemysłowych, to zawierają one jednak szczególnie takie istotne ich elementy, jak możliwości łączenia i współpracy różnych standardów oraz otwartości i niezawodności.

W zamieszczonych tutaj przykładach instalacji pilotażowych ograniczono się jedynie do zagadnień komunikacji. Z uwagi na małe doświadczenia praktyczne z eksploatacji instalacji pilotażowych CIM, w wielu opublikowanych informacjach występujące problemy zostały potraktowane w sposób bardzo ogólny. Ich cechą charakterystyczną jest również duża zmienność proponowanych rozwiązań, która zawsze towarzyszy fazie planowania i projektowania systemów.

Instytut Obrabiarek i Technik Wytwarzania IWF - Brunszwik

Architektura sieci komunikacyjnej w Centrum Transferu Technologii (CTT) w Brunszwiku wskazuje na jej dużą przepustowość, ponieważ tradycyjnie stosowane rozwiązanie zostało rozszerzone o szybkie magistrale światłowodowe (FDDI) (rys.16.1).



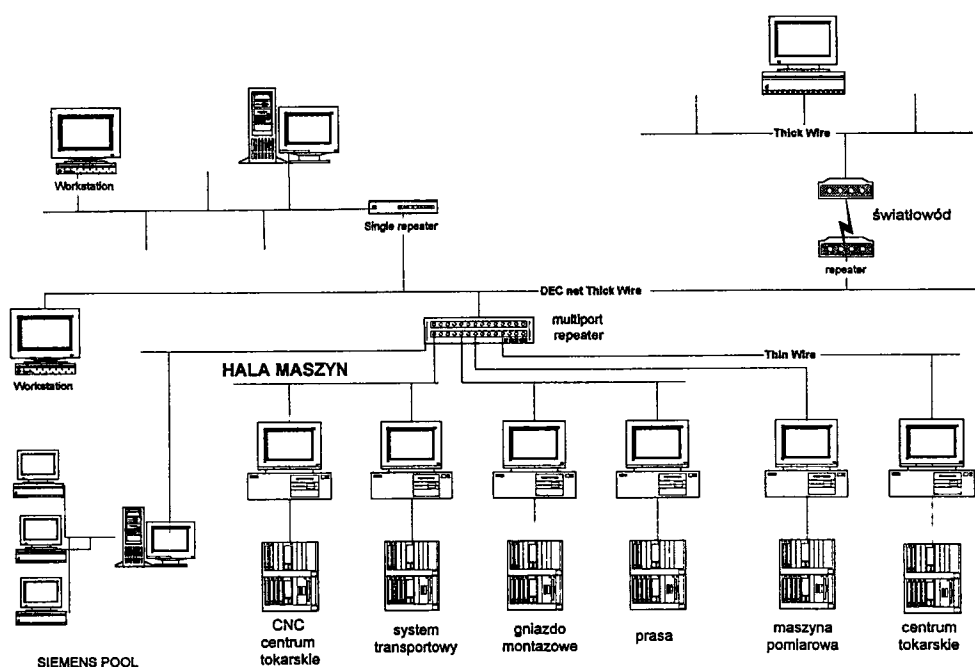
Rys.16.1 Architektura sieci komputerowej w CTT w Brunszwiku

Wyraźnie wyróżnione zostały tutaj dwa poziomy struktury CIM: poziom planowania i zarządzania (jako biura konstrukcyjne, laboratoria i komputery pracowników) oraz poziom produkcji

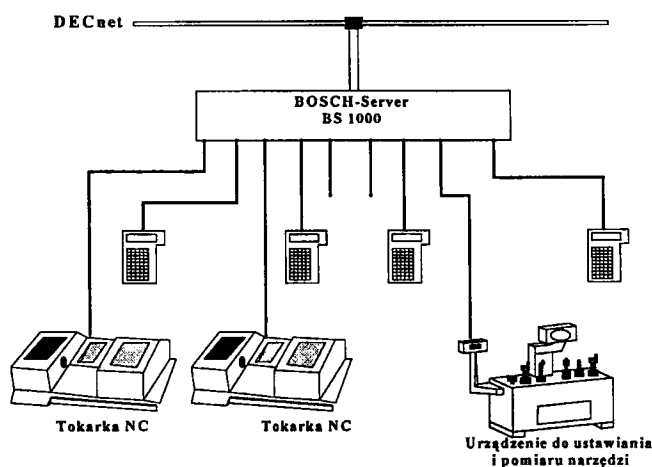
zawierający stanowisko zarządzania, sterowniki CNC, komputery logistyki i zapewnienia jakości oraz bazy danych.

Centrum Transferu Technologii - Drezno

Sieć komunikacyjna Centrum Transferu Technologii (CTT) w Dreźnie została oparta na standardzie DECnet (rys.16.2). Zastosowanie wielokanałowego *repeater*'a pozwoliło na uzyskanie wielu segmentów podsieci, które znacznie zwiększają jej efektywność. Sterowniki maszyn produkcyjnych przyłączone zostały do sieci nadrzędnej za pośrednictwem komputerów gniazd, pełniących funkcje terminali DNC (rys.16.3). Terminale te obsługują jednocześnie wiele sterowników (CNC, PLC, RC) oraz terminali PDA (ang. *Production Data Acquisition*). Sieć komputerowa w CTT w Dreźnie wykorzystuje takie różne media transmisyjne jak: „gruby Ethernet”, „cienki Ethernet” i światłowód.



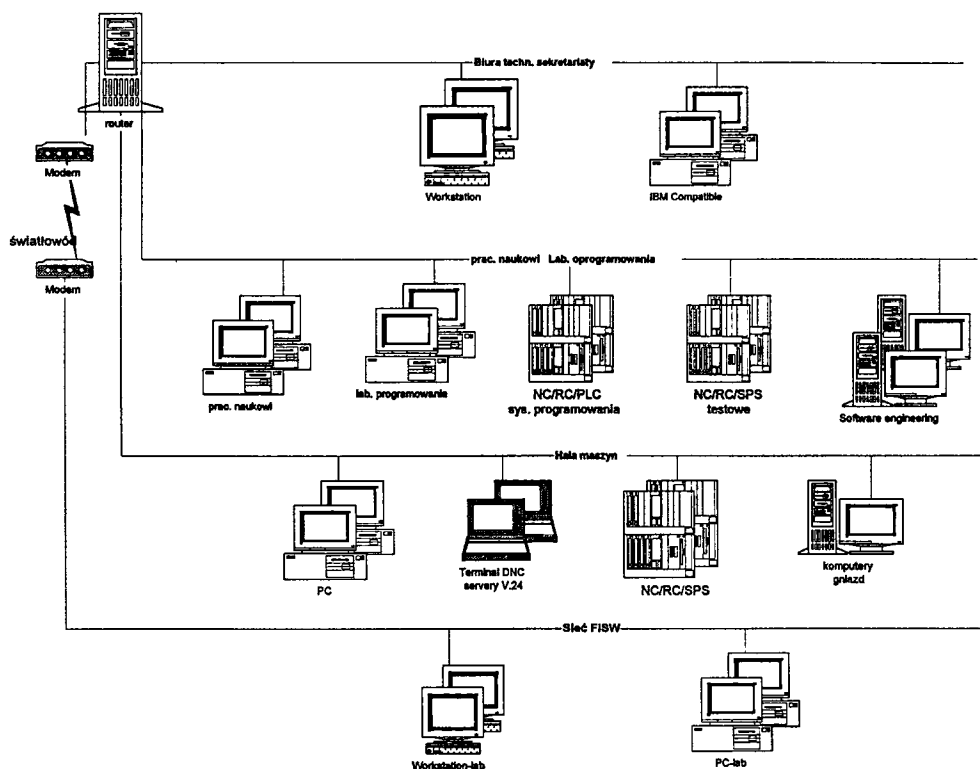
Rys.16.2 Schemat sieci wyższego poziomu



Rys.16.3 Schemat podsieci gniazda produkcyjnego

Instytut Sterowania Obrabiarek i Urządzeń Wytwórczych (ISW) - Stuttgart

Instalacja pilotażowa w ISW Uniwersytetu w Stuttgarcie, z uwagi na duże doświadczenie pracowników tego Instytutu w zagadnieniach CIM, stanowi bardzo rozbudowane rozwiązanie (rys.16.4).

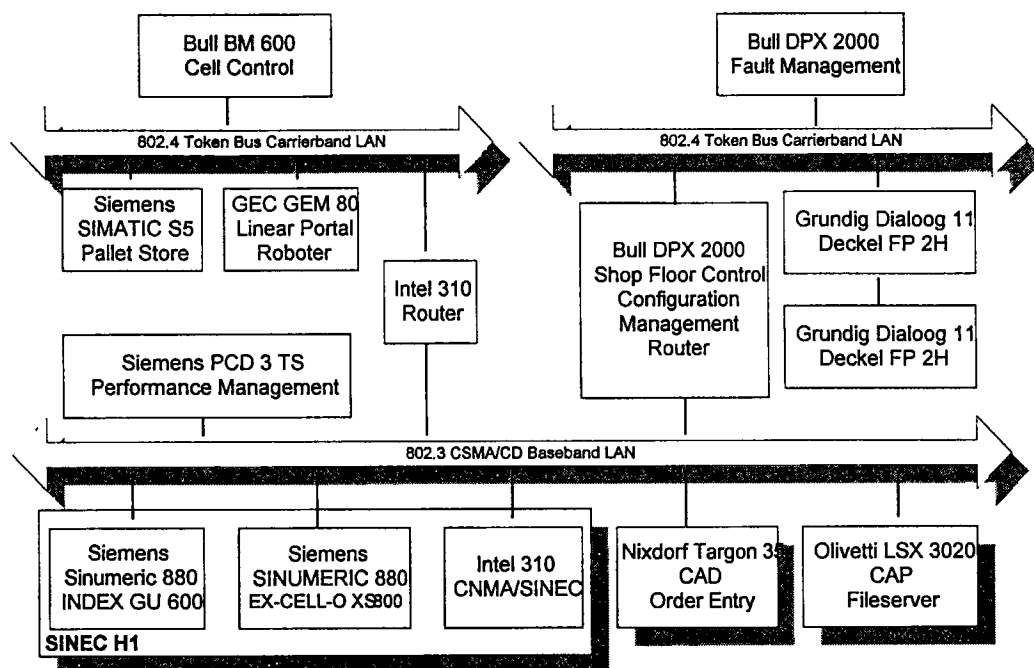


Rys.16.4 Sieć komputerowa w ISW Stuttgart

Szczególnie rozwinięta jest warstwa wyższa sieci komunikacyjnej, która wykorzystuje standardy 802.3 (DECnet, MAP, Sinec, AP, TCP/IP) oraz 802.4 (MAP).

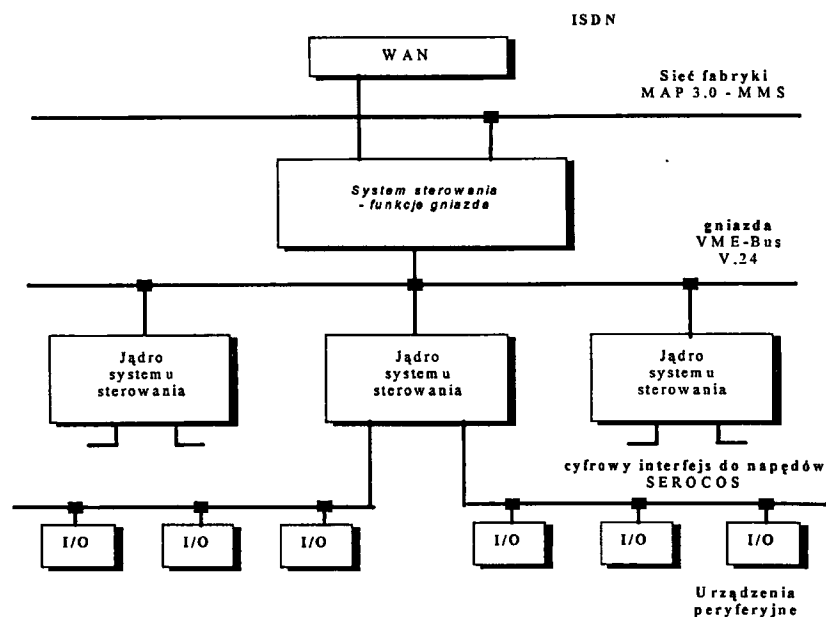
W warstwie gniazd i sterowników dominującym rozwiązaniem jest wykorzystanie usług MMS. Dla sterowników (PLC, NC i RC), nie mających możliwości bezpośredniej komunikacji z magistralą standardu MAP, interfejs do włączenia w sieć zapewniają terminale DNC, które wykorzystują standard V.24. Magistrale miejscowe (*Fieldbus*) łączą natomiast elementy poziomu czujników i członów wykonawczych.

W instalacji pilotażowej CNMA (ang. *Communication Network for Manufacturing*), wykonanej w ramach projektu Esprit, zwrócono szczególną uwagę na zastosowanie różnych sposobów dostępu do sieci (Token Bus - 802.4 i CSMA/CD - 802.3) w ramach architektury MAP (rys.16.5). Sprzęg między różnymi segmentami sieci stanowią *router*'y INTEL 310 i Bull DPX2000. Instalacja opracowana w ramach tego projektu zawiera również segment sieci SINEC-H1, która jest obsługiwana dzięki mostowi połączeniowemu CNMA/SINEC.



Rys.16.5 Instalacja pilotażowa CNMA

W ramach projektu Esprit - FICIM (ang. *Fieldbus Integration in to CIM*), zwrócono również uwagę na zastosowanie magistral przemysłowych (Fieldbus). Podstawowym ich zadaniem jest zapewnienie komunikacji między sterownikami oraz ułatwienie komunikacji z czujnikami i elementami wykonawczymi. Zastosowana magistrala SERCOS spełnia wymagania czasu rzeczywistego (rys.16.6).



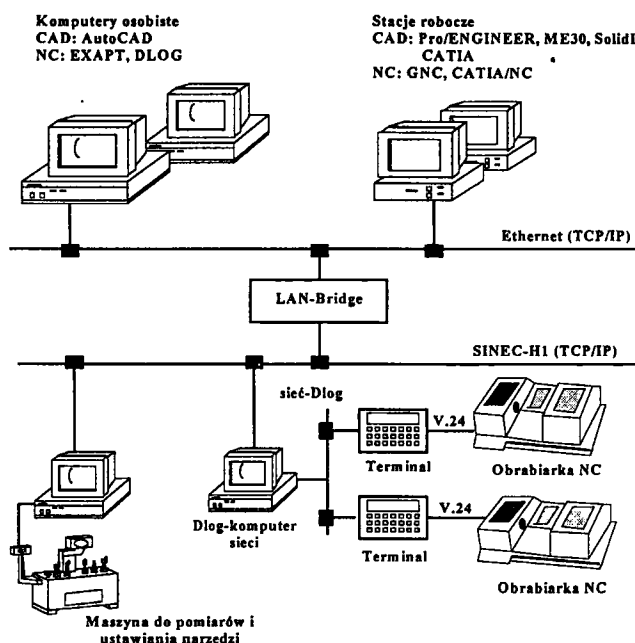
Rys.16.6 Instalacja pilotażowa FICIM

Centrum Transferu Technologii - Chemnitz

Instalacje pilotażowe CTT w Chemnitz (rys.16.7 i 16.8) pokazują główne kierunki badawcze Instytutu, a mianowicie:

- techniki produkcji,
- systemy fabryczne oraz
- komunikację w obszarze procesów.

Zestawiona sieć komunikacyjna ma strukturę hierarchiczną. W warstwie najwyższej (planowania produkcji) zastosowano Ethernet z protokołem TCP/IP. Poziom gniazd wiązań sieci standardu SINEC-H1 z usługami STF, natomiast na poziomie sterowania zastosowano standardy SINEC L2/PROFIBUS (z usługami FDL oraz FMS), Arcnet i V.24. Warstwa czujników i elementów wykonawczych wykorzystuje natomiast magistralę miejscową CAN.



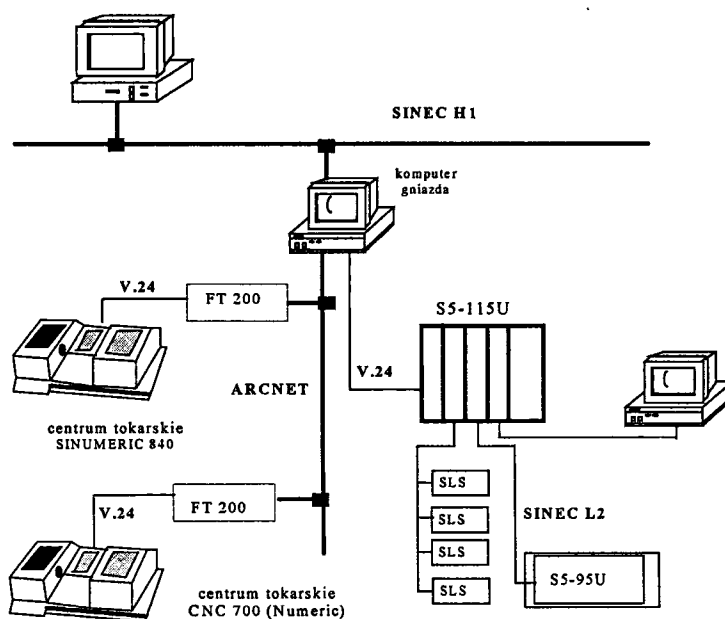
Rys.16.7 Sieć komputerowa instalacji pilotażowej w CTT Chemnitz

Dzięki bogatemu wyposażeniu laboratoriów (SUN SPARCstation 4/60; HP9000/370; HP9000/720; DECsystem 5500; VAX 3100; CYBER 910-400; WX-200; AT 386/486/Pentium; SICOMP PC32; sterowniki: SINUMERIK 840M, Heidenhein TNC415, Bosch CC200M, INDEX C200T, FANUC, SIMATIC S5U135, SIMATIC S5 U115, system transportowy TS4, terminale FT100, FT200, frezarka MAHO typu MH500 E2/432CNC, robot przemysłowy KUKA IR 161/60 z RCM-3B, maszyna do pomiarów i ustawiania narzędzi KOMEG TH 620 i wiele innych), wzbogaconemu o modele techniki Fischer (np. magazyn wysokiego składowania) istnieje możliwość rozpatrywania bardzo złożonych zagadnień.

W systemie komunikacji zwrócono szczególną uwagę na obszary autonomiczne oraz diagnostykę techniczną w celu kontroli maszyn podczas pracy. Rozwiązywane problemy, to:

- dobór przemysłowych systemów sterowania z uwzględnieniem wymagań czasu rzeczywistego,

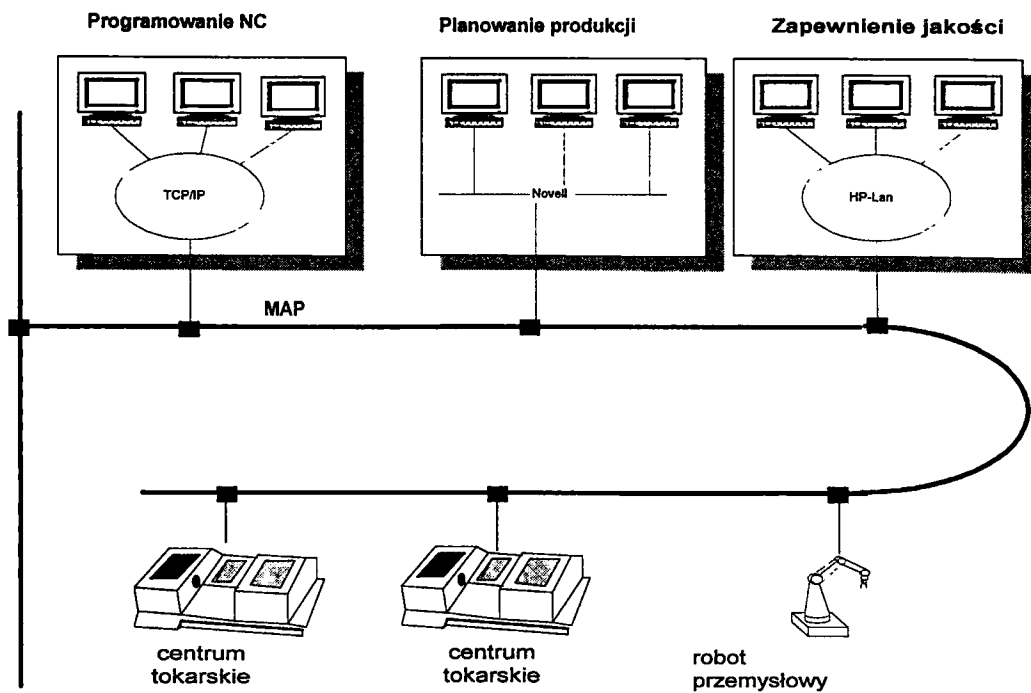
- wymiana danych z wykorzystaniem transmisji szeregowej,
- wizualizacja procesu,
- koordynacja realizacji zlecenia z jego technicznym przebiegiem,
- diagnostyka techniczna metodami klasycznymi oraz z uwzględnieniem nowoczesnych technik (np. *Fuzzy-Logic*).



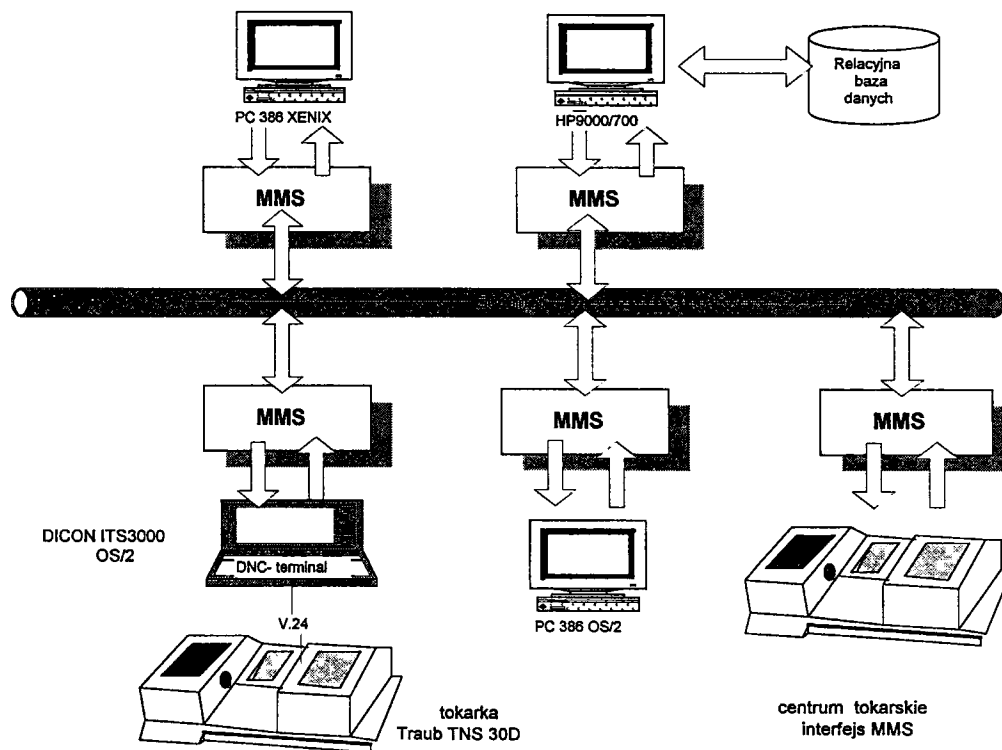
Rys.16.8 Schemat sieci gniazda wytwórczego w CCT Chemnitz

Instytut Urządzeń Produkcyjnych i Technik Konstrukcji (WOP)- Berlin

Komunikację w instalacji pilotażowej w WOP Berlin zrealizowano wg standardu MAP z wykorzystaniem usług MMS (rys.16.9 oraz 16.10). Za pomocą bram dołączone zostały do sieci głównej segmenty podsieci Novell, ICP/IP, HP-Lan. Komunikację w warstwie niższej (procesu) dla sterowników bez usług MMS zapewniają natomiast systemy DNC z połączeniami V.24. Sieć pilotażowa stanowi przykład integracji nie tylko różnych standardów komunikacyjnych, ale również platform komputerowych (HP9000/700, HP9000/370, Micro VAX, VAX-Station, PC 386/486/Pentium), systemów operacyjnych (UNIX/XENIX, VMS, OS/2, MS-DOS) oraz maszyn wytwórczych i pomiarowych z układami sterowania CNC (Jung JF 520 - Sinumeric 810G; Traub TNS30D - TX8D; Deckel FP 4NC - Dialog 11; Gildemeister CT40 - EPL 1; Nassovia Optimat 505; Zeiss FMC600). Dużą uwagę zwraca się również na integrację graficznych środowisk pracy, programów narzędziowych, baz danych i programów użytkowych.



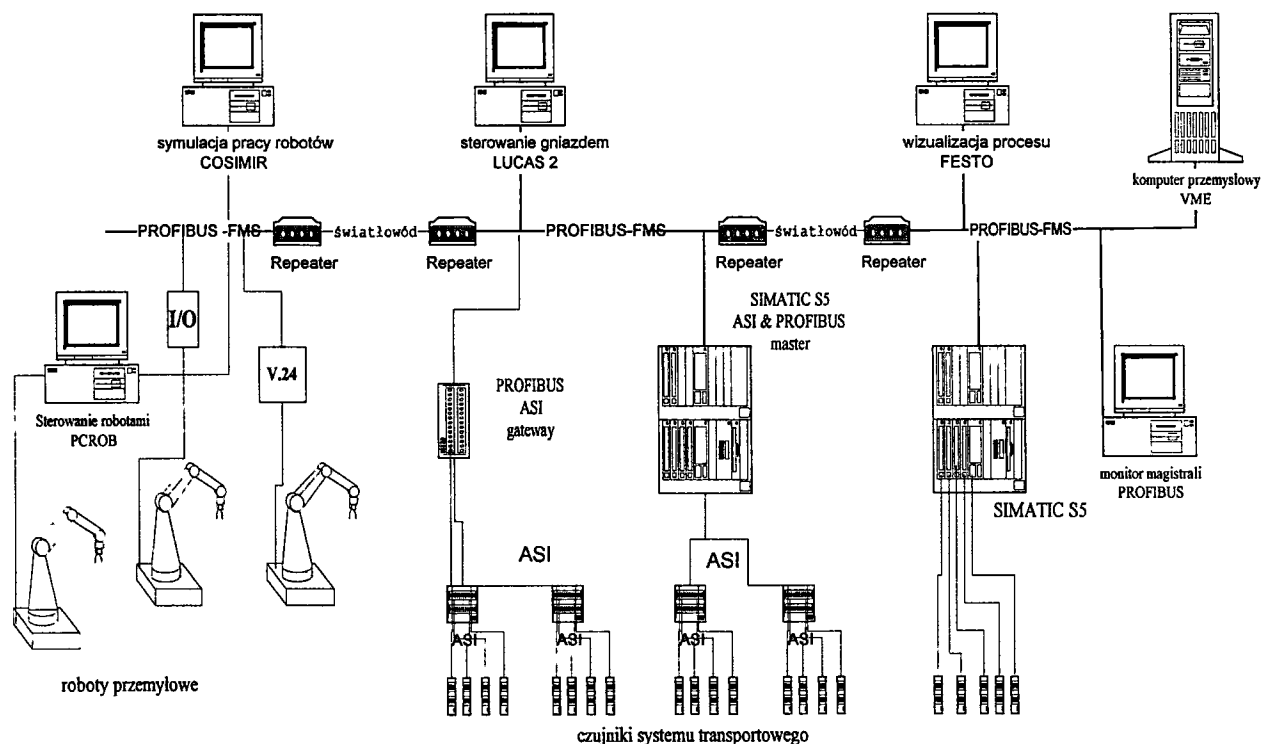
Rys.16.9 Schemat sieci szkieletowej WOP Berlin



Rys.16.10 Wykorzystanie usług MMS w sieci pilotażowej WOP Berlin

Instalacja pilotażowa Instytutu Robotyki (IRF) - Dortmund

Instalacja pilotażowa IFR w Dortmundzie (rys.16.11) obejmuje elastyczne gniazdo montażowe, w skład którego wchodzi paletowy system transportowy oraz trzy roboty montażowe różnych producentów. W celu zapewnienia koordynacji i synchronizacji zadań między różnymi urządzeniami zastosowano magistralę komunikacyjną PROFIBUS-FMS. Uzupełnieniem jej, dla warstwy czujników i elementów wykonawczych, jest magistrala standardu ASI. Dzięki unormowanemu systemowi komunikacji uzyskano dużą elastyczność rozwiązania, co pozwala na instalację elementów pochodzących od różnych producentów. W systemie komunikacyjnym zastosowano jednocześnie różne media łączące (światłowód i skrętka). W celu pokazania różnic między tradycyjnym połączeniem równoległym (punkt - punkt), a rozwiązaniem magistralowym zachowano fragment konwencjonalnej instalacji osieciowania binarnych czujników i elementów wykonawczych systemu transportowego. Połączenia między dwoma standardami PROFIBUS/ASI zrealizowano z wykorzystaniem sterownika SIMATIC S5 oraz bramy.



Rys.16.11 Schemat instalacji pilotażowej w IRF Dortmund

Sterowanie gniazdem montażowym za pomocą programu LUCAS2 i robotami montażowymi PCROB zostało rozszerzone o programy symulacji pracy robotów COSIMIR oraz system przetwarzania obrazów. Dzięki spełnieniu warunku otwartości zastosowanych rozwiązań komunikacyjnych, uzupełnieniem systemów sterowania i wizualizacji są programy różnych producentów.

Planowanym rozszerzeniem tej instalacji jest nadbudowa warstwy wyższej wg standardu MAP z usługami MMS.