

Komunikacja w systemach czasu rzeczywistego

Witold Paluszyński

Katedra Cybernetyki i Robotyki

Wydział Elektroniki, Politechnika Wrocławska

<http://www.kcir.pwr.edu.pl/~witold/>

2017



Ten utwór jest dostępny na licencji
**Creative Commons Uznanie autorstwa-
Na tych samych warunkach 3.0 Unported**

Utwór udostępniany na licencji Creative Commons: uznanie autorstwa, na tych samych warunkach. Udziela się zezwolenia do kopiowania, rozpowszechniania i/lub modyfikacji treści utworu zgodnie z zasadami w/w licencji opublikowanej przez Creative Commons. Licencja wymaga podania oryginalnego autora utworu, a dystrybucja materiałów pochodnych może odbywać się tylko na tych samych warunkach (nie można zastrzec, w jakikolwiek sposób ograniczyć, ani rozszerzyć praw do nich).

Rozproszone systemy czasu rzeczywistego

Systemy czasu rzeczywistego są często tworzone jako systemy rozproszone, z wykorzystaniem sieci komputerowych. Sprzyjają temu następujące czynniki:

- często taniej jest zbudować system z wielu niedrogich elementów, zamiast rozwiązania scentralizowanego,
- wiele systemów czasu rzeczywistego jest naturalnie rozproszonych, z czujnikami i urządzeniami wykonawczymi rozmieszczonymi w odległych lokalizacjach,
- łatwiej jest zapewnić odporność na błędy w systemie rozproszonym niż scentralizowanym.

Ostatni element może wydawać się kontrowersyjny, ponieważ w systemie zawierającym więcej elementów zawodność wzrasta. Jednak budowanie odporności na błędy w systemie rozproszonym często jest możliwe, i łatwiejsze, niż w systemie zcentralizowanym, gdzie jedynym sposobem jest dublowanie całego systemu.

Komunikacja czasu rzeczywistego

Pojęcie **komunikacji czasu rzeczywistego** może być zdefiniowane jako taki przypadek komunikacji, gdzie istnieją wymagania **jakości obsługi** (*quality of service*) takie jak: opóźnienie (*maximum delay*), stratność (*maximum loss rate*), itp.

Zwykłe sieci komputerowe zapewniają „najlepszą możliwą” wydajność (*best effort performance*), bez próby spełnienia żadnych określonych parametrów. To znaczy, gdy pojawia się przeciążenie, tworzone są kolejki, i wszystkie pakiety czekają w nich równo. Dla systemu czasu rzeczywistego takie zachowanie sieci jest w oczywisty sposób nieodpowiednie. Potrzebne są technologie, w których pewne parametry transmisji mogłyby być zagwarantowane.

Jakość obsługi

Jakość obsługi (*Quality of Service*, QoS) wymagana dla sieci komputerowej jest wyrażana za pomocą określonych parametrów.

Opóźnienie: (*delay*) czas dostarczenia pakietu sieciowego do odbiorcy.

Nierównomierność opóźnienia: (*delay jitter*) maksymalna zmienność czasu opóźnienia.

W niektórych zastosowaniach opóźnienie jest bardziej akceptowalne niż jego zmienność. Np. sieciowa transmisja radiowa lub telewizyjna.

Pasmo: (*bandwidth*) ilość danych przesyłanych w jednostce czasu.

Stratność: (*packet loss rate*) procent pakietów traconych w transmisji.

Prawdopodobieństwo blokowania: (*blocking probability*) prawdopodobieństwo, że próba połączenia zostanie odrzucona przez sieć.

Charakterystyka ruchu sieciowego

Ruch generowany w sieciach czasu rzeczywistego można podzielić na pewne ogólne kategorie:

CBR (Constant Bit Rate): pakiety o stałym rozmiarze są generowane w stałych odstępach czasu; transmisja jest ciągła i nie ma w niej spiętrzeń.

Taki rodzaj transmisji jest typowo generowany przez czujniki.

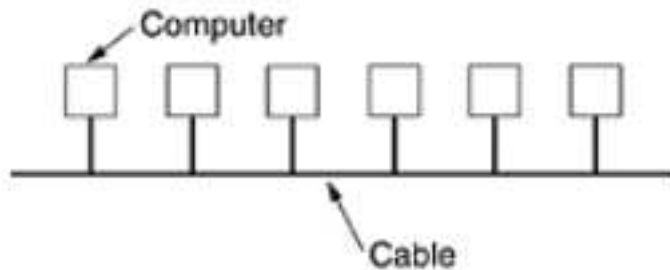
VBR (Variable Bit Rate): albo powtarzalna transmisja pakietów o stałym rozmiarze jest przedzielana okresami bezczynności, albo okresowo generowane są serie pakietów o zmiennym rozmiarze.

Taki rodzaj ruchu jest generowany przykładowo w transmisji skompresowanego dźwięku i/lub obrazu, gdzie redundancja obrazu i kodowanie MPEG powodują okresy bezczynności i okresy dużego natężenia ruchu.

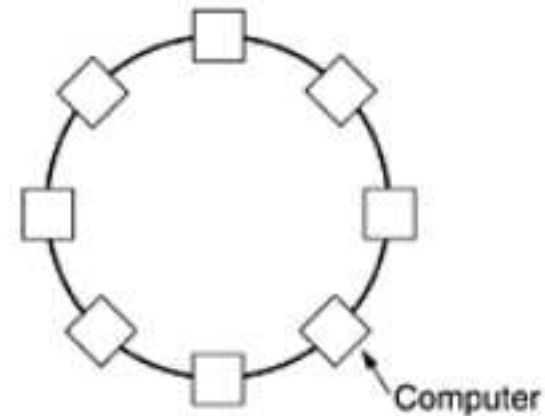
Ruch sporadyczny: czasami wyróżniany rodzaj ruchu, gdzie rzadko może pojawić się seria pakietów o zmiennej długości.

Przykładem może być ruch generowany w systemach alarmowych.

Topologie sieci komputerowych — LAN



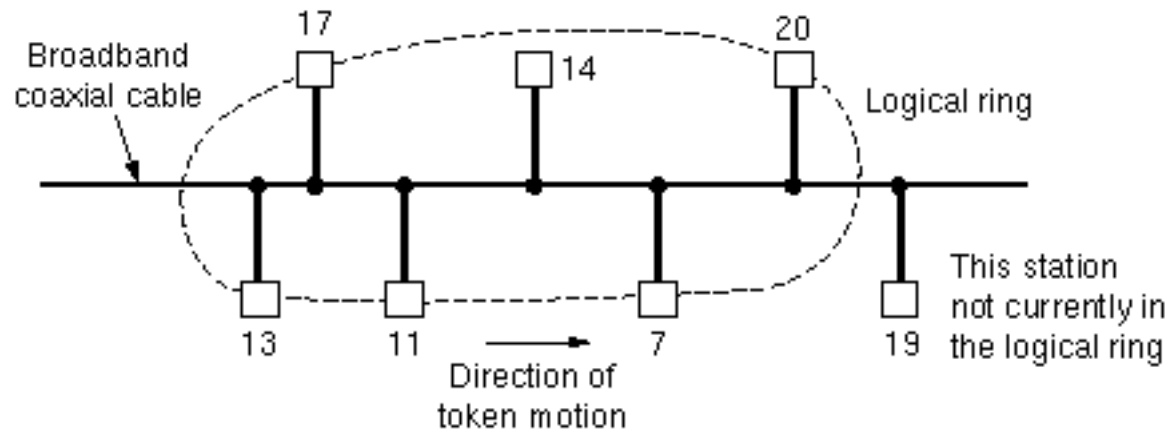
Topologia **magistrali** (ang. *bus*) łączy bezpośrednio wszystkich ze wszystkimi w wspólnym medium transmisyjnym pracującym w podziale czasu. W danej chwili tylko jeden węzeł może nadawać, pozostałe muszą czekać. Niezbędny jest mechanizm **arbitrażu** dla rozsądzenia konfliktów, zwanych **kolizjami**, kiedy dwa lub więcej węzłów chce nadawać jednocześnie. Mechanizm arbitrażu może być zcentralizowany lub rozproszony.



Topologia **pierścienia** łączy węzły sieci oddzielnymi segmentami. Pakiety krążą w jednym kierunku, a dostęp do sieci jest kontrolowany za pomocą specjalnego pakietu kontrolnego zwanego **żetonem** (*token*), przekazywanego od węzła do węzła. Żeton rozwiązuje problem arbitrażu w sposób rozproszony. Jednak awaria jednego węzła „kładzie” całą sieć. Kłopotliwa jest również rozbudowa i rekonfiguracja.

Topologie sieci komputerowych — LAN (cd.)

Topologie magistrali i pierścienia z żetonem posiadają pewne dopełniające się wady i zalety. Możliwe jest połączenie tych koncepcji w sieci o fizycznej topologii magistrali i logicznej topologii pierścienia z żetonem, zwanej **magistralą z żetonem** (*token bus*).



W powyższej sieci logiczną kolejnością przekazywania żetonu jest: 13-11-7-20-17. Węzły 14 i 19 nie należą do logicznego pierścienia.

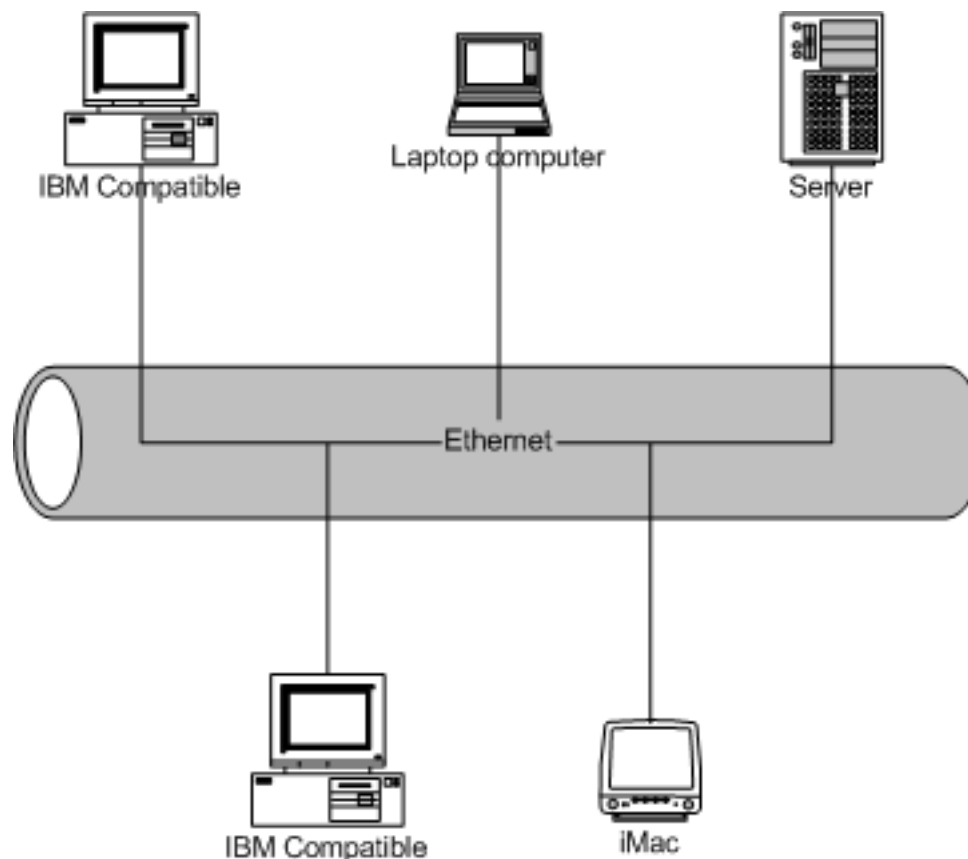
Technologie sieci LAN

W dalszej części tej prezentacji przedstawione zostaną wybrane technologie sieci LAN. Można je podzielić ze względu na powyższe topologie sieci, i związane z nimi standardy IEEE:

- technologie dla sieci magistralowych: CSMA/CD Ethernet, IEEE 802.3
- technologie dla sieci token bus, IEEE 802.4
- technologie dla sieci token ring, IEEE 802.5

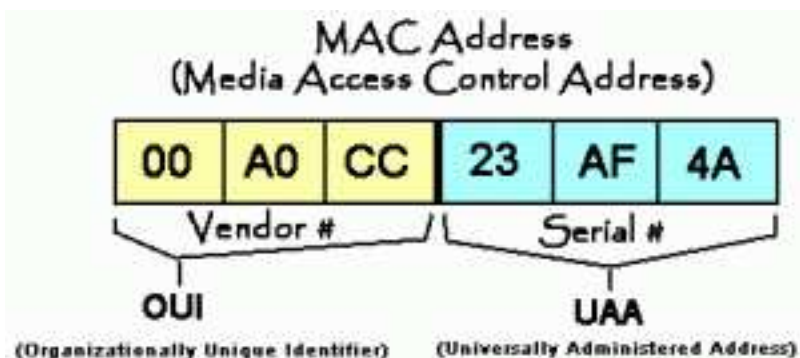
Technologia Ethernet

Ethernet jest niezwykle popularną technologią sieci komputerowych opracowaną w latach 1970-tych przez firmę Xerox. Jego specyfikacja została opublikowana w roku 1976. Powstało wiele unowocześnionych wersji, i technologia Ethernet jest jedną z najpopularniejszych również w dzisiejszych sieciach komputerowych LAN.



Adresowanie w sieci Ethernet

Unikalnym rozwiązaniem zastosowanym w sieci Ethernet jest przypisanie wszystkim urządzeniom 48-bitowych adresów, tzw. MAC (*Media Access Control*) unikalnych w skali światowej. Każde urządzenie sieci Ethernet ma na stałe przypisany własny adres, co pozwala włączać do sieci dowolne urządzenia i nie martwić się o przydział adresów. Administracją adresów początkowo zajmowała się firma Xerox, a obecnie przejęła ją organizacja IEEE. Bloki adresów przyznawane są firmom produkującym sprzęt Ethernet.



Celem takiej konstrukcji jest maksymalna prostota. Adres w sieci Ethernet nie ma żadnej struktury¹ i nie pełni żadnej roli poza identyfikacją. Jedyną operacją wykonywaną na adresach jest porównanie wszystkich 48 bitów. Ogranicza to przetwarzanie adresów do minimum i redukuje do zera problem administracji adresami.

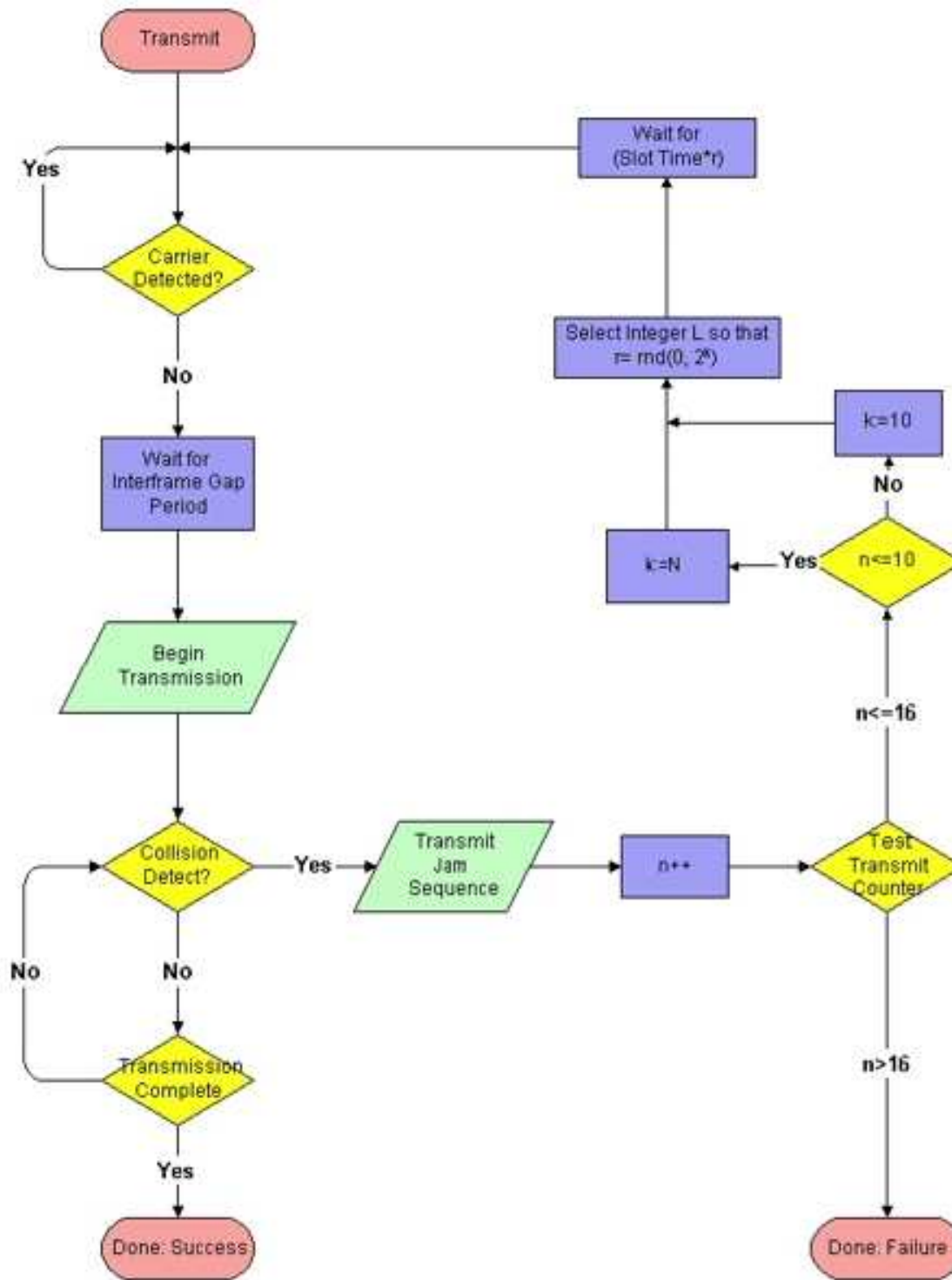
¹Nie jest to do końca prawda, ponieważ nawet w oryginalnej wersji Ethernet adres składający się z samych jedynek pełnił rolę adresu rozgłaszania (*broadcast*).

Arbitraż dostępu do sieci Ethernet

Jedną z kluczowych koncepcji działania Ethernet jest jego mechanizm arbitrażu dostępu do sieci. Ethernet jest siecią typu magistrali, to znaczy z jednoczesnym dostępem wszystkich węzłów do sieci, określaną jako CSMA/CD (*Carrier-Sense Multiple-Access/Collision Detection*).

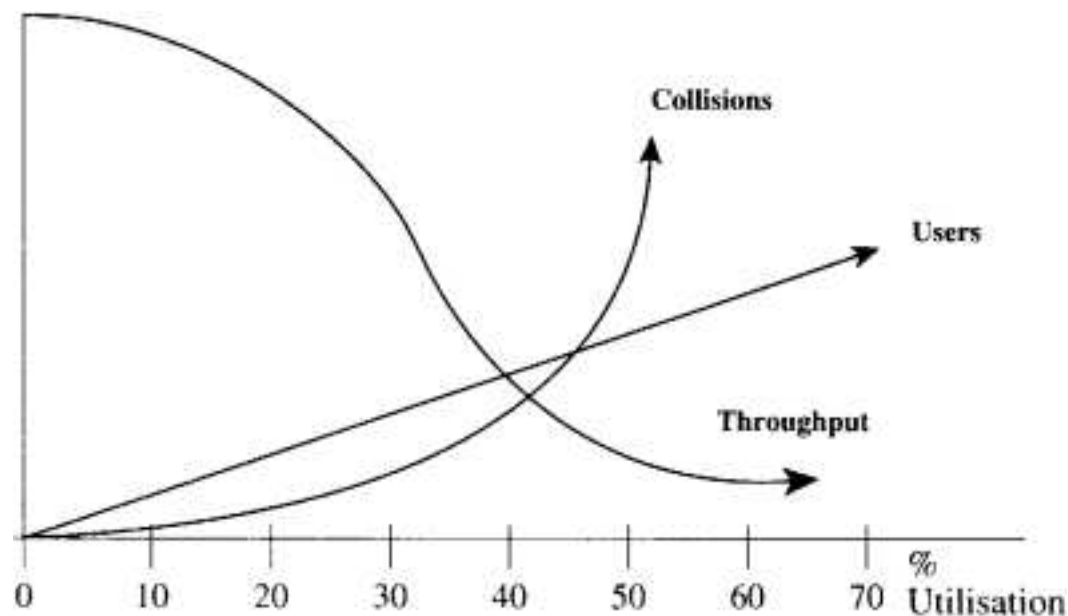
Arbitraż dostępu do magistrali jest rozproszony i polega na jednoczesnym nadłuchiowaniu przy próbie nadawania, w celu wykrycia ewentualnej kolizji. Rozwiązaniem w przypadku wykrycia kolizji jest wstrzymanie nadawania na losowy okres czasu, a w przypadku ponownej kolizji — wydłużanie tego okresu.

Jedną cechą tego algorytmu jest niesprawiedliwość arbitrażu. Węzeł, któremu zdarzą się dwie lub więcej kolizji, jest „karany” stopniowo wydłużanym czasem oczekania.



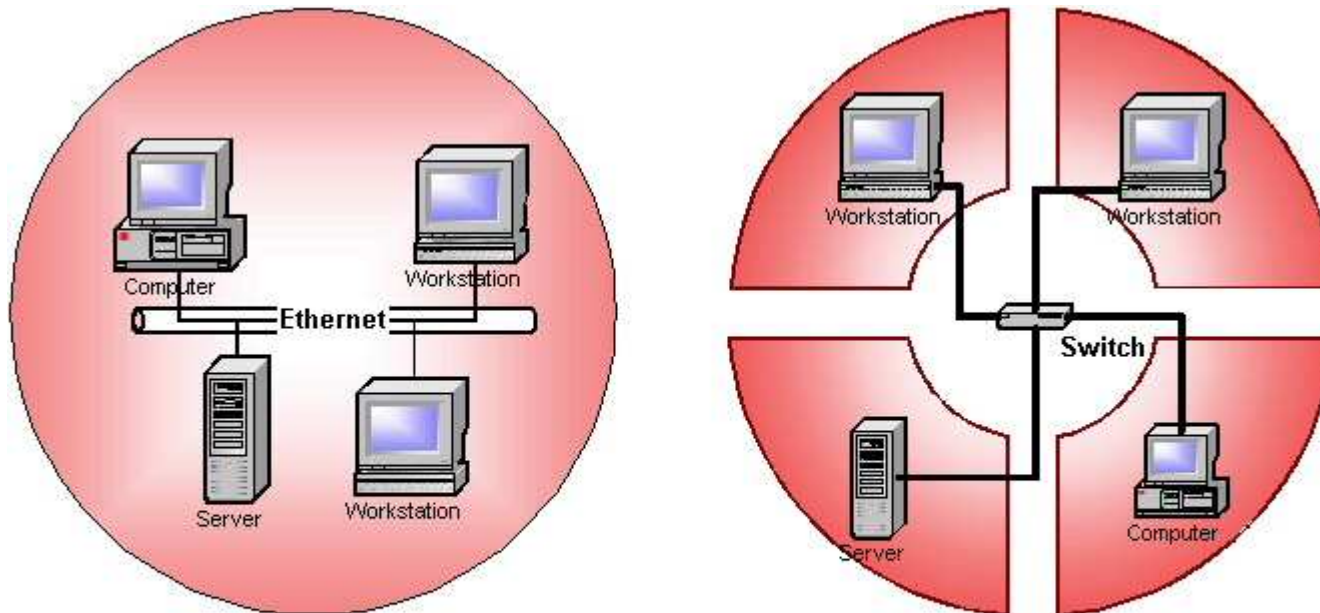
Zjawisko nasycenia w sieci Ethernet

Niekorzystnym efektem rozproszonego mechanizmu arbitrażu w sieciach Ethernet jest lawinowo narastająca liczba kolizji przy dużym obciążeniu przez wiele stacji. Ponieważ ten mechanizm arbitrażu powoduje marnowanie pasma, przy dużej liczbie kolizji duża część pasma jest marnowana. Zjawisko jest szczególnie nasilone przy małych pakietach. Powoduje to niemożność wykorzystania pełnego pasma fizycznego sieci. Przy wielu węzłach transmitujących małe pakiety trudno w sieci Ethernet uzyskać wykorzystanie więcej niż 50% pasma.



Przełączany Ethernet

Współczesne wersje sieci Ethernet w przeważającej większości realizują logiczną topologię magistrali przy fizycznym połączeniu w topologię gwiazdy.



Dzięki przełącznikom sieciowym ruch w poszczególnych segmentach sieci nie przenosi się do innych segmentów i zjawisko nasycania się sieci Ethernet w miarę wzrostu obciążenia w dużej mierze zostaje zniwelowane.

Między innymi, pozwoliło to na rozszerzenie technologii Ethernet do obsługi komunikacji sieciowej w czasie rzeczywistym.

Real-time Ethernet

Istnieją współczesne standardy „real-time” Ethernet przystosowane do obsługi ruchu sieciowego w czasie rzeczywistym. Większość z nich są technologiami własnościowymi wprowadzonymi przez poszczególne firmy:

- Ethernet/IP (*Industrial Protocol*) jest protokołem komunikacyjnym warstwy aplikacyjnej, wykorzystującym standardowe rozwiązania Ethernet warstw niższych (fizycznej, sieciowej i transportowej)
- EtherCAT (*Control Automation Technology*) wykorzystuje tryb Master/Slave i działa w sieciach o topologii gwiazdy jak również magistrali, jednak bazuje na specjalizowanym sprzęcie
- Ethernet Powerlink wykorzystuje standardowe urządzenia Ethernet w trybie Master/Slave i również nie wymaga izolacji ruchu przez przełączniki sieciowe
- PROFINet jest rodziną standardów opartych na Ethernet, z wersją PROFINet-SRT (*Soft Real-Time*) umożliwiającą elementarną obsługę systemów czasu rzeczywistego, i PROFINet-IRT (*Isochronous RT*) zapewniających pracę hard-RT
- IEEE 1588 definiuje standard urządzeń Ethernet zawierających zegar wykorzystujący protokół PTP (*Precision Time Protocol*) i umożliwiający synchronizację węzłów sieci

Technologie Real-time Ethernet są aktywnie rozwijane.

Sieci pierścieniowe Token Ring/IEEE 802.5

Praca sieci pierścienia z żetonem *Token Ring* (technologia firmy IBM), lub 802.5 (standard IEEE):

1. Żeton jest tylko jeden i tylko jego posiadacz może nadawać.
2. Żeton jest przekazywany od węzła do węzła w określonym kierunku.
3. W czasie bezczynności sieci żeton krąży po niej w kółko.
4. Jeśli węzeł posiadający żeton chce nadawać, to przekształca żeton w nagłówek komunikatu, dodaje adres nadawcy i odbiorcy, oraz treść komunikatu, i puszcza go w dalszą drogę zamiast żetonu.
5. Gdy komunikat dotrze do węzła odbiorcy, ten odbiera komunikat, podmienia bity kontrolne by poinformować nadawcę, że komunikat został odebrany, i puszcza ramkę dalej do sieci.
6. Gdy oryginalny nadawca otrzymuje swoją ramkę z potwierdzeniem odbiorcy, kasuje jej treść, przekształca ramkę z powrotem w żeton, i wprowadza go znowu do sieci.

Priorytetowa transmisja Token Ring/IEEE 802.5

Standard Token Ring/IEEE 802.5 przewiduje w żetonie/nagłówku komunikatu trzy bity przeznaczone na priorytet komunikatu. Mogą one być ignorowane w normalnej transmisji, lub mogą być wykorzystane do realizacji arbitrażu globalnego priorytetu, według poniższego schematu:

1. W żetonie/nagłówku komunikatu jest pole priorytetu, oraz bit trybu transmisji. Trybem transmisji może być: tryb rezerwacji/nadawania i tryb wolny.
2. W trybie rezerwacji trwa transmisja danych. Węzeł nadawcy tworzy ramkę komunikatu, a w nagłówku odpowiadającym żetonowi ustawia priorytet swojego komunikatu, oraz bit trybu na rezerwację.
3. Tryb rezerwacji oznacza, że w czasie propagacji komunikatu, każdy węzeł, który ma do nadania komunikat o wyższym priorytecie, wpisuje ten priorytet do nagłówka, ale propaguje dalej oryginalną ramkę danych. W ten sposób jednak rezerwuje własną transmisję na kolejny cykl.
4. Gdy ramka powraca do węzła jej nadawcy, ten zamienia ją z powrotem na żeton, ustawia bit trybu na tryb wolny, i wprowadza żeton do sieci.
5. Gdy żeton dotrze do węzła z komunikatem do nadania o priorytecie równym priorytetowi w żetonie, ten wie, że wygrał rezerwację, i rozpoczyna nadawanie.

Silna komunikacja czasu rzeczywistego w sieciach LAN

Silna komunikacja czasu rzeczywistego w sieciach często dotyczy ruchu CBR, jak np. okresowo transmitowane dane z czujników. W silnych systemach czasu rzeczywistego determinizm opóźnień komunikacji jest ważniejszy niż wykorzystanie pasma sieci.

Istnieje szereg rozwiązań zapewniających determinizm komunikacji w sieciach magistralowych. Można wyróżnić następujące klasy protokołów komunikacji:

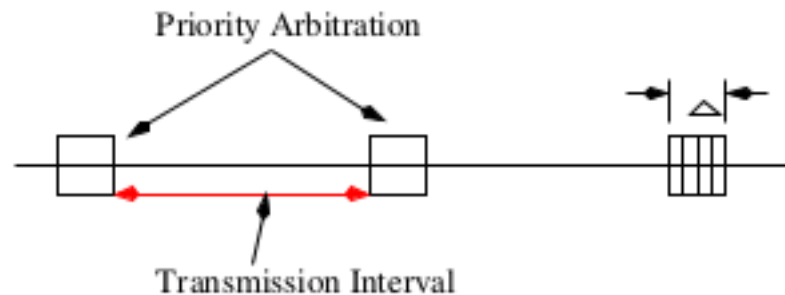
Protokoły globalnego priorytetu (*Global Priority*) — wszystkie komunikaty mają przydzielone globalne unikalne priorytety i rola algorytmu arbitrażu ogranicza się do wyboru komunikatu o najwyższym priorytecie.

Protokoły ograniczonego dostępu (*Bounded Access*) — czas dostępu do sieci każdego węzła jest ograniczony, zatem opóźnienie dostępu każdego węzła będzie również ograniczone; pozostaje problem wyboru pakietu do wysłania przez węzeł sieci w czasie jej dostępności.

Protokoły oparte na kalendarzu (*Calendar-Based*) — wszystkie węzły posiadają kopie kalendarza transmisji określającego co może być transmitowane w jakim przedziale czasu; jeśli pojawia się komunikat nieobjęty kalendarzem to może on być transmitowany w wolnym przedziale poprzedzony rezerwacją rozgłaszaną do wszystkich węzłów. Taki protokół jest prosty i skuteczny jeśli wszystkie komunikaty są okresowe i przewidywalne.

Protokół „Countdown”

Wyznaczanie komunikatu o najwyższym priorytecie w sieci magistralowej odbywa się na zasadzie podziału czasu na przedział arbitrażu i przedział transmisji. W czasie przeznaczonym na arbitraż wyznaczony zostaje komunikat o najwyższym priorytecie, i może on następnie zostać wysłany.



Protokół „Countdown” polega na podziale czasu arbitrażu na podprzedziały o stałej długości. W czasie kolejnych podprzedziałów transmitowane są przez wszystkie zainteresowane węzły kolejne bity priorytetu, począwszy od najstarszego. Jeśli sieć działa na zasadzie aktywnego stanu wysokiego, to realizuje logiczną operację OR, i wygrywa stan wysoki. Węzeł, który wytransmitował bit 0 a odczytał w tym czasie 1 wie, że przegrał arbitraż i musi się wycofać.

Przykład arbitrażu protokołu „Countdown”:

węzeł	priorytet	bity msb→lsb
N1	10	0 1 0 1 0
N2	16	1 0 0 0 0
N3	20	1 0 1 0 0

Po pierwszym bicie węzeł N1 wycofuje się z arbitrażu, a po trzecim bicie również N2. Po zakończeniu transmisji wszystkich bitów priorytetu węzeł N3 wie, że wygrał arbitraż i może transmitować.

Zauważmy, że aby ten algorytm działał poprawnie, podprzedział każdego bitu musi trwać co najmniej tyle ile czas propagacji sygnału w sieci od końca do końca.

Protokół czasu wirtualnego

W metodzie czasu wirtualnego nie ma określonych przedziałów arbitrażu. Arbitraż polega na odczekaniu najpierw na wolny stan magistrali, a następnie dodatkowo na odczekaniu okresu czasu odwrotnie proporcjonalnego do priorytetu komunikatu do wysłania. Im niższy jest ten priorytet, tym dłużej węzeł musi czekać po wykryciu wolnego stanu magistrali.

Po odczekaniu odmierzonego czasu węzeł upewnia się, że w sieci nie trwa transmisja innego węzła, i jeśli nie, to sam zaczyna transmitować. Węzeł, który odczekał swój czas, ale następnie wykrył rozpoczętą transmisję, wie, że ubiegł go komunikat o wyższym priorytecie, zatem musi wstrzymać się do następnego okresu bezczynności sieci.

Jaka powinna być różnica czasu czekania pomiędzy dwoma priorytetami różniącymi się o jedność? Gdyby była mniejsza niż czas propagacji sygnału w sieci od końca do końca, to węzeł mógłby nie być w stanie niezawodnie wykryć transmisji innego węzła o priorytecie większym zaledwie o 1. Zatem różnica priorytetów o 1 powinna odpowiadać różnicy czasu odczekania co najmniej równej czasowi propagacji sygnału w całym segmencie magistrali.

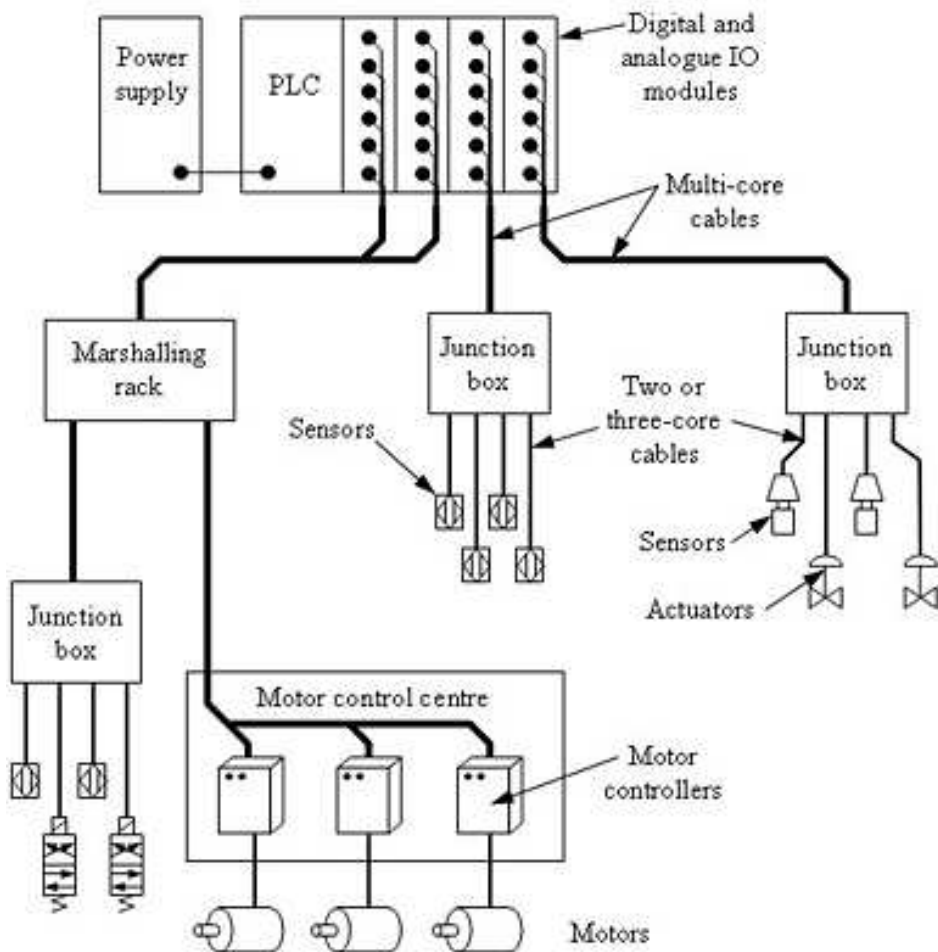
Przemysłowe sieci komputerowe

W odpowiedzi na zapotrzebowanie przemysłowe na standardy połączeń różnych urządzeń muszących się komunikować na terenie np. zakładu przemysłowego powstało wiele technologii i standardów odpowiadających takim potrzebom.

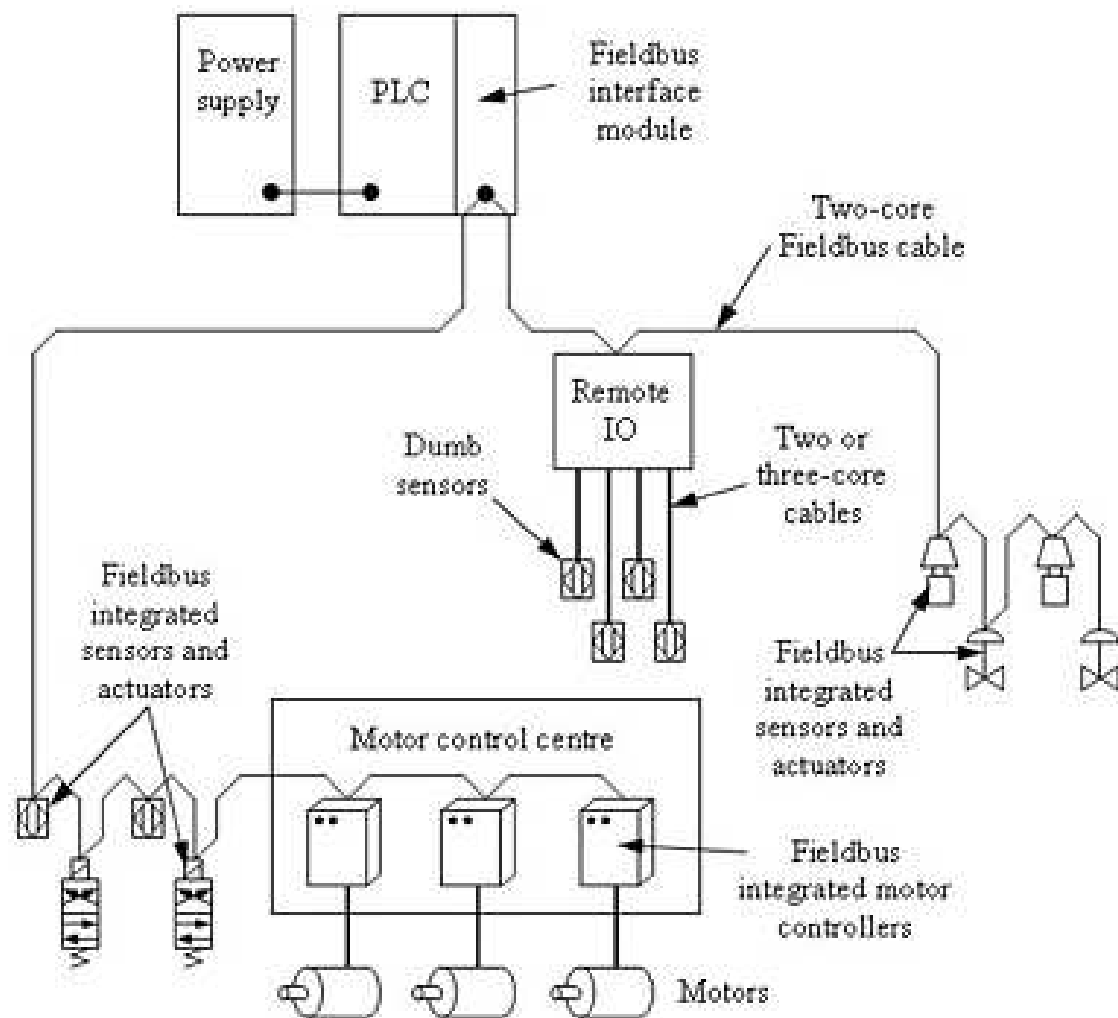
Wiele z nich łączy wspólne pojęcie **Fieldbus**, częściej pisane małą literą - **fieldbus**, ponieważ nie jest to nazwa własna żadnej konkretnej technologii ani urządzenia, ale raczej nazwa pewnej klasy sieci łączącej cyfrowe urządzenia elektroniczne. Nazwą **Field Devices** określa się czujniki i elementy wykonawcze, oraz ich sterowniki, działające na terenie zakładu.

Typowym trybem transmisji w sieciach fieldbus jest transmisja szeregową, z podziałem pasma w czasie. Dane są zwykle transmitowane w małych paczkach. Wybór trybu szeregowego ponad inne tryby wynika z pewnych zalet, na przykład kosztu okablowania.

Ogólnie, określenie fieldbus oznacza przemysłowe sieci sterowania.



Tradycyjne okablowanie systemu przemysłowego typu point-to-point wymaga ekstensywnego okablowania, złączy, i ma wiele punktów zawodności.



Architektura fieldbus ma minimalne wymagania dotyczące okablowania i złączy, a dzięki modularnym sterownikom wbudowanym w każde urządzenie pozwala na jednolite standardy komunikacji, monitorowania, zgłaszania błędów, itp.

Architektura Fieldbus

Sieci Fieldbus stanowi wiele technologii wykorzystywanych w przemyśle motoryzacyjnym, automatyce i robotyce, oraz sterowania procesami.

Istnieją sieci Fieldbus własnościowe objęte patentami, licencjami i zastrzeżeniami konkretnych firm.

Istnieją również sieci Fieldbus otwarte, których pełna specyfikacja jest opublikowana i dostępna (może być za niewygórowaną opłatą), dla której istnieją dostępne podstawowe komponenty (np. sterowniki), oraz istnieją dobrze opracowane, dostępne dla wszystkich użytkowników procedury walidacji.

Zalety architektury Fieldbus

Ogranicza złożoność systemów sterowania w sensie warstwy sprzętowej.

Prowadzi do ograniczenia złożoności systemów sterowania; projektowanie jest prostsze, bardziej wydajne, i w efekcie tańsze.

Wybór dobrze rozpowszechnionego systemu prowadzi do zamienności wyposażenia i podzespołów między różnymi dostawcami.

Ograniczone są również problemy z łączami, kompatybilnością, itp.

Przykłady technologii Fieldbus

- AS-Interface: Europa, standard otwarty
- CAN bus: Niemcy, standard początkowo opracowany i otwarcie publikowany przez Bosch, następnie wydany jako standard ISO
- Interbus: Niemcy, Phoenix Contact, otwarty system
- Profibus: Niemcy, Siemens
- ModBus: U.S.A., Modicon
- ControlNet: U.S.A., Allen-Bradley/Rockwell, otwarty system sterowania

CANbus

CANbus jest magistralą komunikacyjną opracowaną przez firmę Bosch dla potrzeb przemysłu samochodowego, która szybko zyskała akceptację w przemyśle motoryzacyjnym i lotniczym. CANbus jest szeregowym protokołem magistralowym do łączenia poszczególnych systemów i czujników oryginalnie pomyślanym jako alternatywa do konwencjonalnych wiązek wielożyłowych. Pozwala podzespołom na komunikowanie się po jedno- lub dwuprzewodowej magistrali danych do 1Mbps.

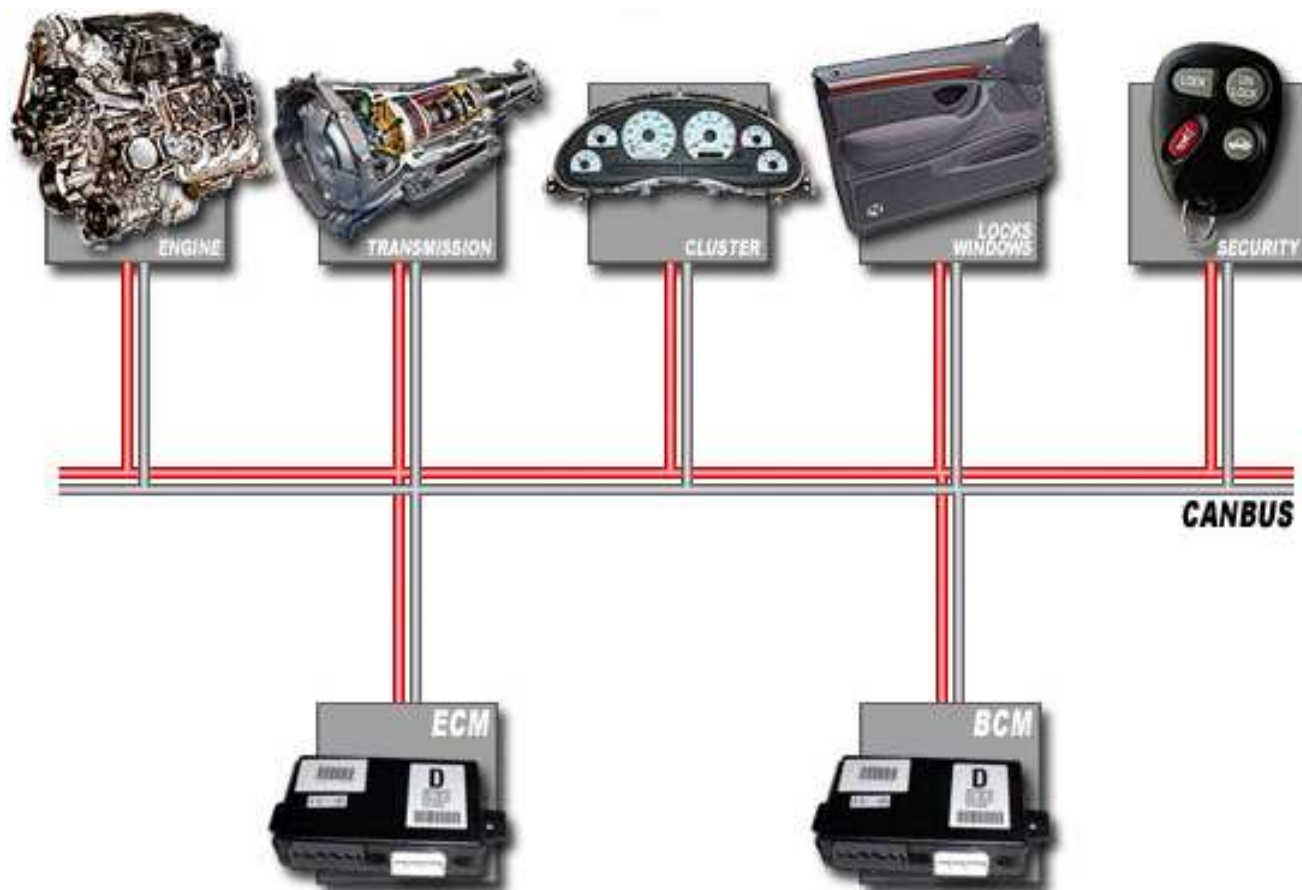
Motywacja wprowadzenia CANbus

Od połowy XX-go wieku technologie motoryzacyjne zwiększały udział elektroniki w konstrukcjach samochodów wprowadzając coraz większą liczbę komponentów elektronicznych, i związanych z nimi funkcji. Elementy elektroniczne zastępowały systemy mechaniczne i dostarczały dodatkowe udogodnienia i funkcje bezpieczeństwa. Powodowało to nieograniczony wzrost okablowania i związanych z tym kosztów oraz awaryjności.



Wprowadzenie magistrali CAN

BMW 850 coupe wprowadzony na rynek w 1989 roku był pierwszym pojazdem wyposażonym w magistralę CANbus (aczkolwiek bardziej rozbudowaną sieć CANbus — 5 węzłów — wprowadził Mercedes w 1992). Nowa technologia pozwoliła ograniczyć ilość okablowania o ponad 2 kilometry, wagę samochodu o ponad 50 kg, a liczbę złączy o połowę. Pozwoliła podzespołom komunikować się z prędkością 25kbps-1Mbps.



CANbus — masowy rozwój

Oczywiście wprowadzenie nowej technologii CANbus początkowo wymusiło również zwiększenie kosztu wszystkich podzespołów, które teraz musiały zawierać interfejs CANbus. Między innymi, wyeliminowało to dostawców niezależnych, którzy nie byli w stanie ponieść kosztów wdrożenia nowej technologii.

Jednak CANbus odniósł wielki sukces, przede wszystkim ze względu na to, że jest dobrą technologią, odpowiadającą potrzebom pojawiającym się w wielu konstrukcjach, jak również ze względu na opublikowanie specyfikacji przez firmę Bosch i objęcie jej standardami międzynarodowymi.

Obecnie wszystkie większe rynki samochodowe (U.S.A., U.E., i inne) wymagają aby wszystkie sprzedawane tam samochody były wyposażone w CANbus.

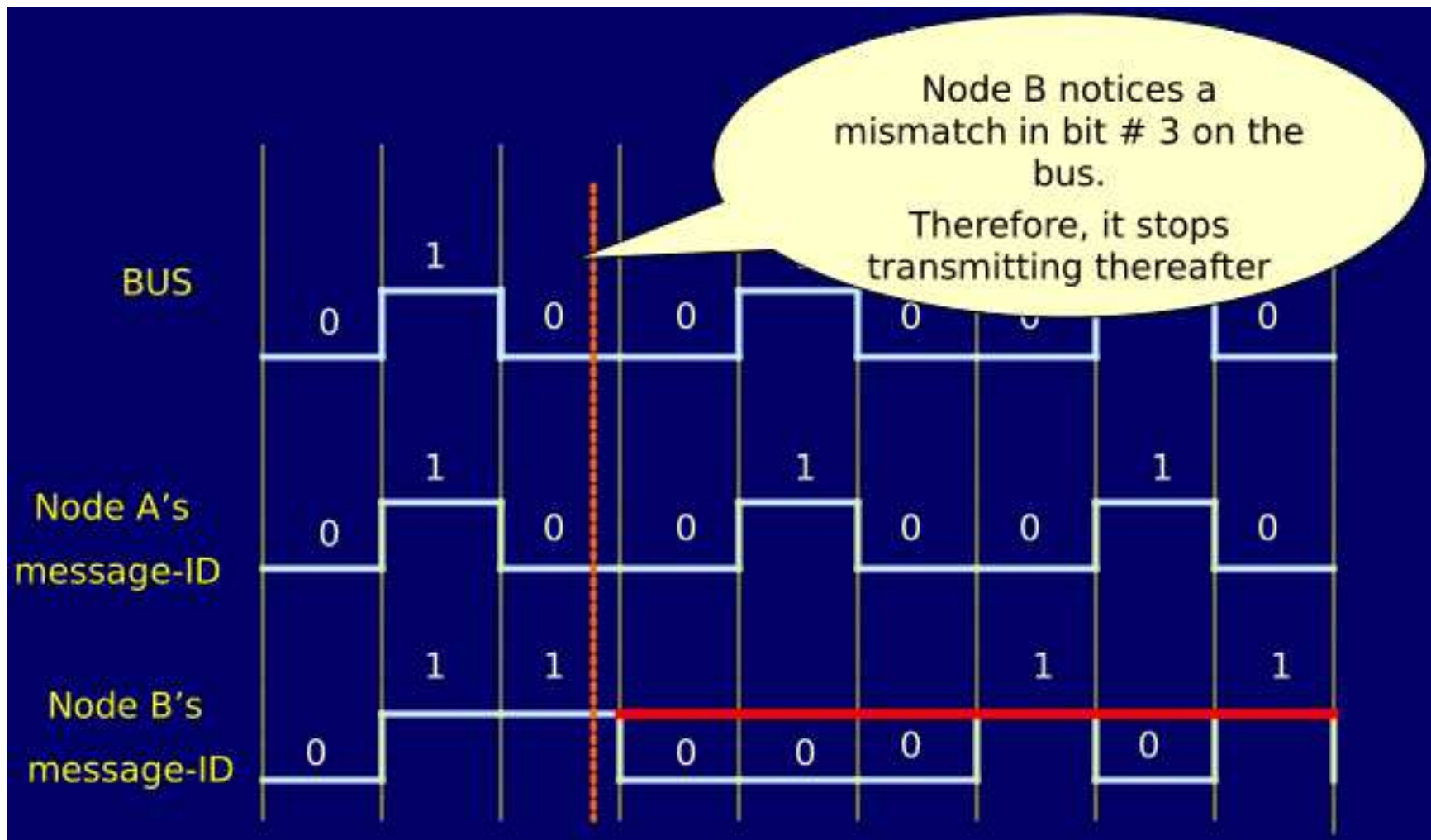
CANbus — szczegóły techniczne

W przeciwieństwie do tradycyjnych sieci, takich jak USB lub Ethernet, CANbus nie wysyła dużych bloków danych, z węzła A do węzła B pod nadzorem centralnego zarządcy sieci. W sieci CAN przesyłane są krótkie komunikaty, zawierające takie dane jak temperatura, albo liczba obrotów silnika. Te komunikaty są transmitowane do całej sieci, co zapewnia spójność danych w każdym węźle systemu.

Protokół komunikacyjny CAN jest protokołem magistralowym z jednoczesnym dostępem i wykrywaniem kolizji, podobnie jak Ethernet, dodatkowo uzupełnionym o arbitraż oparty na priorytecie komunikatu (CSMA/CD+AMP — *Carrier-Sense Multiple-Access Collision Detection with Arbitration on Message Priority*).

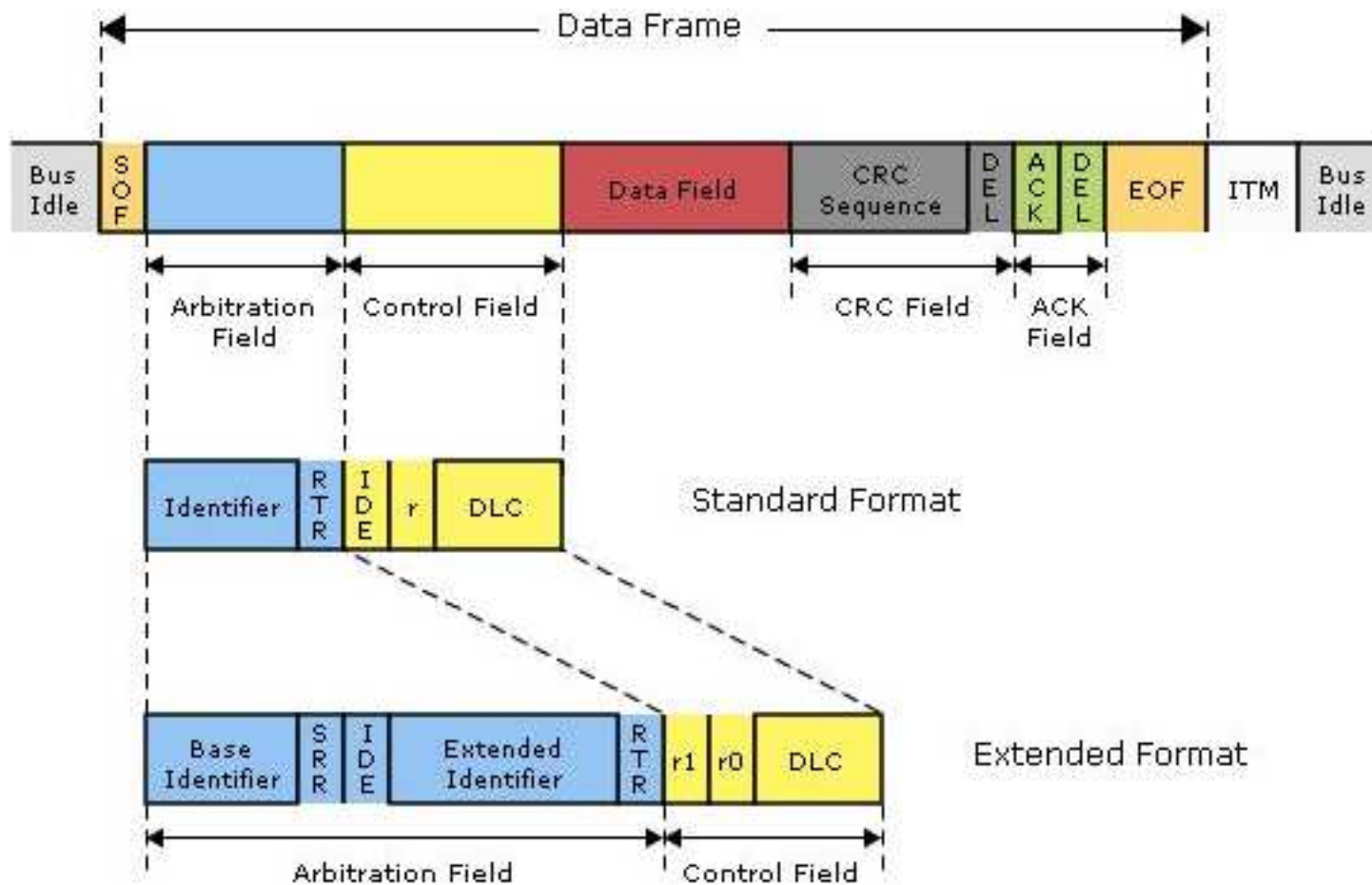
CSMA oznacza, że każdy węzeł magistrali musi czekać na okres bezczynności sieci przed przystąpieniem do transmitowania wiadomości. CD+AMP oznacza, że kolizje są wykrywane i automatycznie rozstrzygane przez arbitraż bitowy, w oparciu o zaprogramowany priorytet każdego komunikatu, określonego w polu identyfikatora komunikatu. Identyfikator o wyższym priorytecie zawsze wygrywa dostęp do magistrali.

Wyższy priorytet jest związany z identyfikatorem o mniejszej wartości. Porównanie jest wykonywane bitowo: nadawca który ma najwięcej zer na początku identyfikatora wygrywa. Działa to dzięki zastosowaniu elektroniki o wysokim stanie, gdzie połączenie stanu wysokiego (pasywnego) i niskiego (aktywnego) powoduje, że wygrywa niski.



Komunikaty CANbus

Standard ISO-11898:2003 definiuje podstawowy format komunikatu CANbus z 11-bitowym identyfikatorem (maks. $2^{11} = 2048$ różnych identyfikatorów). Został on później uzupełniony o format z rozszerzonym 29-bitowym identyfikatorem (2^{29} , czyli 537 miliony różnych identyfikatorów).



Komunikaty CANbus zawierają od jednego do ośmiu bajtów danych. Istnieje uzupełniona specyfikacja CAN FD dopuszczająca do 64 bajtów danych w komunikacie. Poza komunikatem danych możliwe są jeszcze inne komunikaty, które nie zawierają danych ale pełnią funkcję pomocnicze (np. potwierdzenie).

Bibliografia

Hermann Kopetz: Real-Time Systems: Design Principles for Distributed Embedded Applications, Kluwer 2002