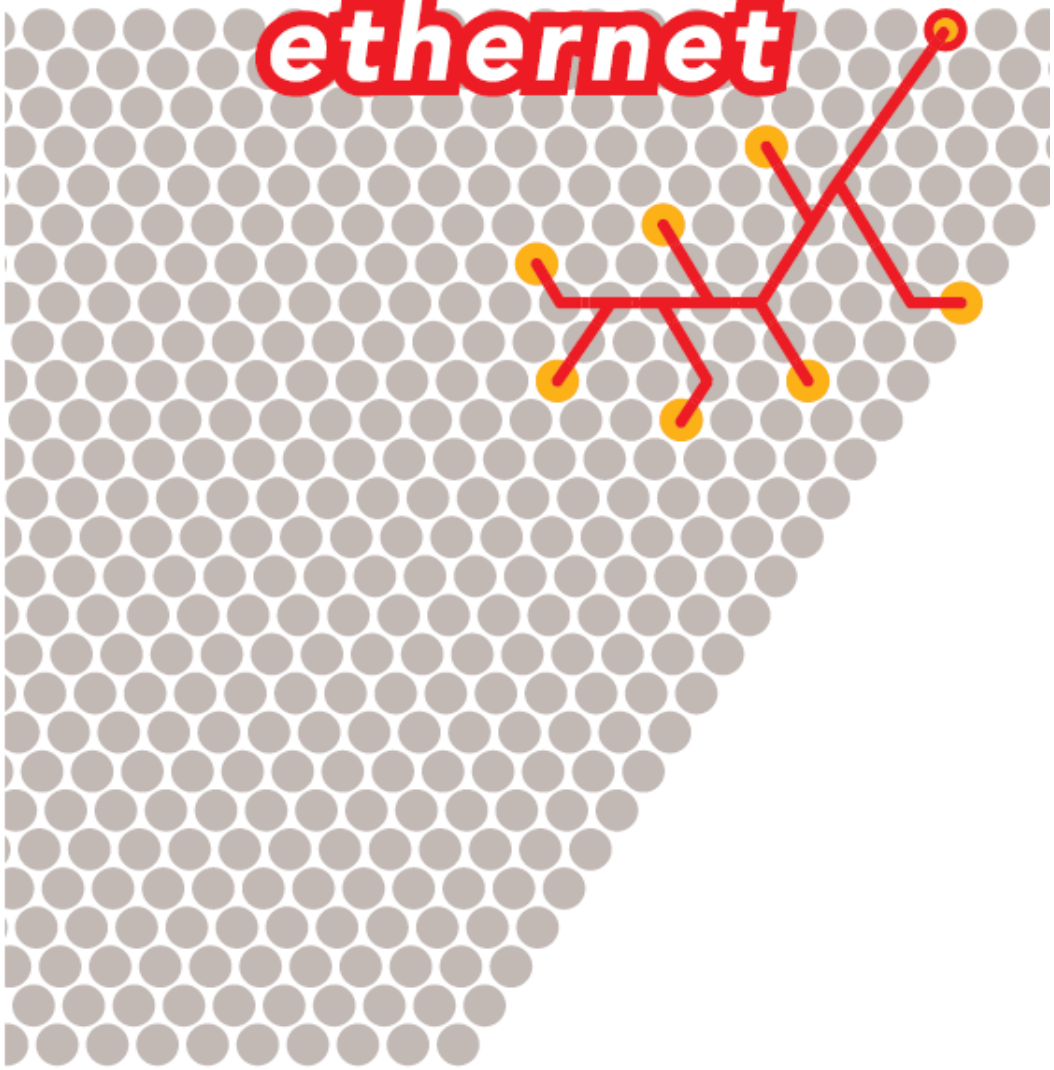
R.A. Hulsebos

industrial ethernet



PHOENIX CONTACT

INNOVATION IN INTERFACE

industrial ethernet

industrial ethernet

R.A.Hulsebos

PHOENIX CONTACT

Inhoud

INHOUD

Voorwoord Phoenix Contact

Industriële communicatietechniek is sinds 20 jaar een competentie van Phoenix Contact en een onmisbaar onderdeel van de huidige industriële automatiseringstechniek. Veldbussen zoals Interbus, Profibus, DeviceNet en CANopen zijn hiervan de belangrijkste voorbeelden.

De communicatiestandaard is er helaas nog steeds niet. Sinds een aantal jaren speelt Ethernet een steeds belangrijkere rol in de industrie. De verwachting is dat de komende jaren de markt voor industrieel Ethernet met 80% per jaar zal toenemen.

Of het Ethernet wel lukt om de communicatiestandaard in de industriële automatisering te worden is nog de vraag en zal afhangen van nieuwe ontwikkelingen zoals ProfiNet en Ethernet IP. Voor velen redenen om zich in industrieel Ethernet te verdiepen. Vandaar deze unieke uitgave.

Jaap Dijsselhof
Phoenix Contact bv

Zevenaar, september 2004

Voorwoord

Het gebruik van Ethernet is in elke kantooromgeving eigenlijk standaard. Binnen de industriële automatisering kennen we echter veel meer variatie in netwerktechnologieën; zo'n 500 verschillende industriële netwerken beconcurreren elkaar. Dit is een enorme verspilling van R&D geld en marketinginspanningen, kennisverdunning en aansluitperikelen voor gebruikers.

Het is daarom toe te juichen dat ook Ethernet steeds verder ingezet wordt voor industriële toepassingen. Men kan dit splitsen in *Gebruik van Ethernet in een industriële applicatie* en *Industrieel gebruik van Ethernet*. Alhoewel dit oppervlakkig hetzelfde lijkt te betekenen, zijn er toch wel de nodige verschillen.

Het gebruik van Ethernet in industriële applicaties is al gangbaar sinds ca. 1985, toen de eerste Ethernet-interfaces op de markt beschikbaar kwamen. In veel gevallen zal hier "standaard" Ethernet hardware en software gebruikt worden: de TCP/IP protocolfamilie met ARP, DHCP, BOOTP, IGMP, RIP, SNMP, (Fast)STP, Internet-koppelingen, firewalls, redundantie, routers, etc. Hier kan men al zeer veel literatuur, hardware, software, websites enz. over vinden. Nóg een publicatie hierover heeft weinig toegevoegde waarde, en de genoemde onderwerpen komen in deze publicatie dan ook niet aan de orde.

Industrieel gebruik van Ethernet gaat echter veel verder, omdat het systeem zich ontwikkelt als een nieuwe veldbus. Dat is een heel ander toepassingsgebied dan de administratieve automatisering, en er worden dus ook geheel nieuwe eisen aan Ethernet gesteld. De interne details van Ethernet, die in de kantoorautomatisering geen enkele gebruiker nog boeien, zijn voor industriële automatiseerders juist wél heel erg interessant omdat vaak precies bekend moet zijn hoe een netwerk opereert qua opstarten, snelheid, foutafhandeling, veiligheid, storingsgevoeligheid, enz. Alhoewel er in de (Duitse en Amerikaanse) vakpers veel over gepubliceerd is, was er tot halverwege 2003 maar één (kleinschalig) Engelstalig boekje over dit onderwerp beschikbaar.
De behoefte aan een diepgaandere bespreking van industrieel Ethernet zonder marketing bla-bla heeft geleid tot deze publicatie, welke niet alleen in het Nederlandse taalgebied maar zelfs daarbuiten een zekere primeur te noemen is. Ik hoop daarom ook dat deze publicatie in een behoefte voorziet.

R.A. Hulsebos
Nuenen, mei 2004
rahulsebos@cs.com

1. Dertig jaar Ethernet

1.1 Still going strong...

Onlangs heeft Ethernet zijn 30e verjaardag gevierd. Dat is een respectabele leeftijd voor een netwerktechnologie, die dus ontstaan is in een tijdperk dat het geheugen van de gemiddelde microprocessor nog in honderden bytes werd gemeten, kloksnelheden in tientallen kHz'en en software in assembly werd gemaakt. Des te verbazingwekkender is het dat Ethernet nog steeds niet met pensioen gestuurd is en vervangen door een jongere technologie; integendeel, in 2003 is de specificatie voor de 10 Gbit/s variant uitgekomen en er wordt al weer gepraat over nóg hogere bitrates.

In de administratieve automatisering ("op kantoor") heeft Ethernet inmiddels een marktaandeel van meer dan 95% behaald. Des te opvallender is het dat het gebruik in industriële applicaties altijd beperkt is gebleven. Het echte werk wordt in dit vakgebied gedaan door de veldbussen, waarvan (in scherp contrast met de LAN-wereld) er zéér veel varianten ontwikkeld zijn.

1.2 Richting industrieel gebruik

Vanaf ca. 1998 is een ontwikkeling gestart waarbij steeds meer gekeken wordt naar Ethernet als de "veldbus van de toekomst". Deze ontwikkelingen kwamen voornamelijk uit de VS, waar de acceptatie van Europese (voornamelijk Duitse) veldbustechnologie niet zo groot. Ethernet zelf is (van origine) een Amerikaanse technologie, en "industrial Ethernet" is dan ook door de Amerikanen met veel bombarie gelanceerd.
Dit heeft geleid tot zeer veel media-aandacht en hype-verschijnselen in de markt. Ironisch is het om te zien dat nu, na circa 5 jaar, het toch voornamelijk Duitse bedrijven (en een enkeling van elders) zijn die de kar met nieuwe ontwikkelingen rondom industrieel Ethernet trekken.

De ontwikkelingen rondom industrieel Ethernet zijn veel langzamer gelopen dan in eerste instantie gedacht. De hype van de eerste jaren (1999-2001) is geheel verdwenen. Deze hype nam zodanige vormen aan dat gedacht werd (en door sommige bedrijven ook zo gepropageerd) dat Ethernet de "veldbus van de toekomst" zou worden, waardoor de honderden 1e generatie veldbussystemen van de markt zouden gaan verdwijnen en vervangen worden door één systeem: industrieel Ethernet, en dan zouden forse schaalvoordelen te behalen zijn. Op dit moment heeft Ethernet echter geen enkel ander systeem van de markt verdreven, en dit zal ook nooit gebeuren. Ethernet is (simpel gezegd) primair een manier van bekabelen. Bestaande veldbussystemen zijn op dit moment bezig om zichzelf uit te

breiden zodanig dat ook op Ethernet bekabeld kan worden. Het is dus eerder een fusie van netwerktechnologieën, in plaats van een overname of verdringing.

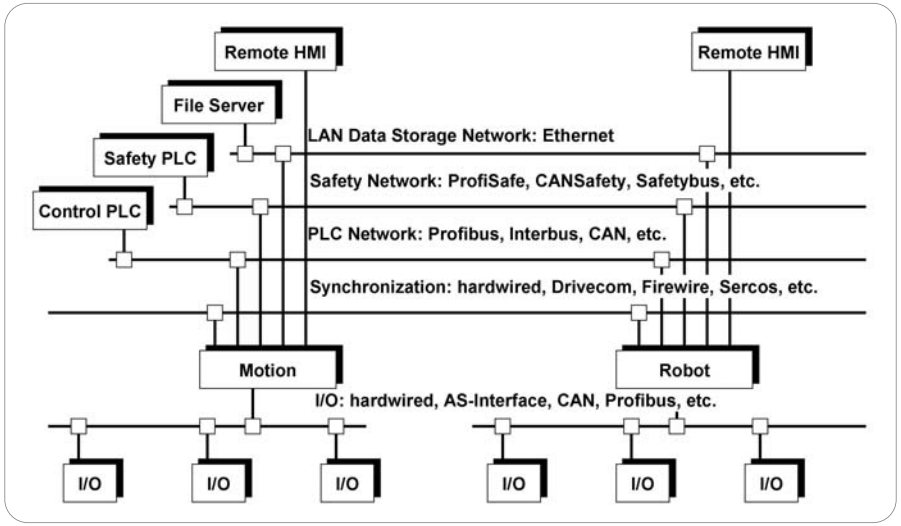
Inmiddels wordt er nog steeds gewerkt aan het industriëler maken van Ethernet. Het 'industrieel Ethernet' is voor het grootste deel identiek aan het Ethernet dat we op kantoor (en ook thuis!) hebben liggen, maar is in sommige details toch afwijkend. Deze nieuwe technologie wordt niet in een paar jaar ontwikkeld; net zoals bij de al bestaande industriële netwerken zal dit c.a. tien jaar kosten (van eerste idee tot systemen waarmee ook niet-beursdemo's op te bouwen zijn). De eerste resultaten zullen vanaf ca. 2005 op de markt komen.

Dat wil niet zeggen dat er nu niets is – integendeel, al vanaf de introductie van Ethernet op PC's (ca. 1983) wordt het al gebruikt voor industriële toepassingen. De meeste leveranciers van PLC's en andere industriële besturingen hebben Ethernet in hun catalogus staan. Er zijn genoeg interfaces, switches, protocollen etc. te krijgen op basis van Ethernet. In de meeste gevallen betreft het echter nog bedrijfsspecifieke (proprietary) oplossingen, of aanpassingen van bestaande protocollen die nu ook op Ethernet bekabeld kunnen worden. De nieuwe generatie ontwikkelingen op het gebied van industrieel Ethernet zijn veel geavanceerder, meer geoptimaliseerd, en passen beter bij moderne ontwikkelingen uit de IT (o.a. gebruik van XML, component-gebaseerd ontwikkelingen, objectoriëntatie, UML, etc.). Dit maakt een koppeling van Ethernet op de plantvloer naar het Ethernet op kantoorniveau ook makkelijk, want de software uit beide werelden sluit beter op elkaar aan.

1.3 Positie van Ethernet ten opzichte van bestaande veldbussen

De bestaande veldbussen en Ethernet zijn niet per definitie concurrenten van elkaar. Dit wordt veroorzaakt door de technische optimalisaties die zowel aan Ethernet als aan de veldbussen zijn uitgevoerd. Ethernet is immers ooit ontwikkeld als LAN, waarop grote hoeveelheden data efficiënt getransporteerd moeten kunnen worden naar apparatuur (PC's) waarin veel hardware-aanwezig is (CPU, disk, RAM, powersupply, etc.) en dus niet op een paar Euro gekeken hoeft te worden. Bij veldbussystemen gaat de optimalisatie vooral richting eenvoudige hardware, het goedkoop kunnen bekabelen en het optimaal transporteren van kleine hoeveelheden data. Beide types netwerken hadden daarom hun eigen inzetgebied, en waren geen concurrenten van elkaar. Bij Ethernet speelde verder het complete gebrek aan standaardisatie nog een rol; elke leverancier had zijn eigen netwerkprotocollen.

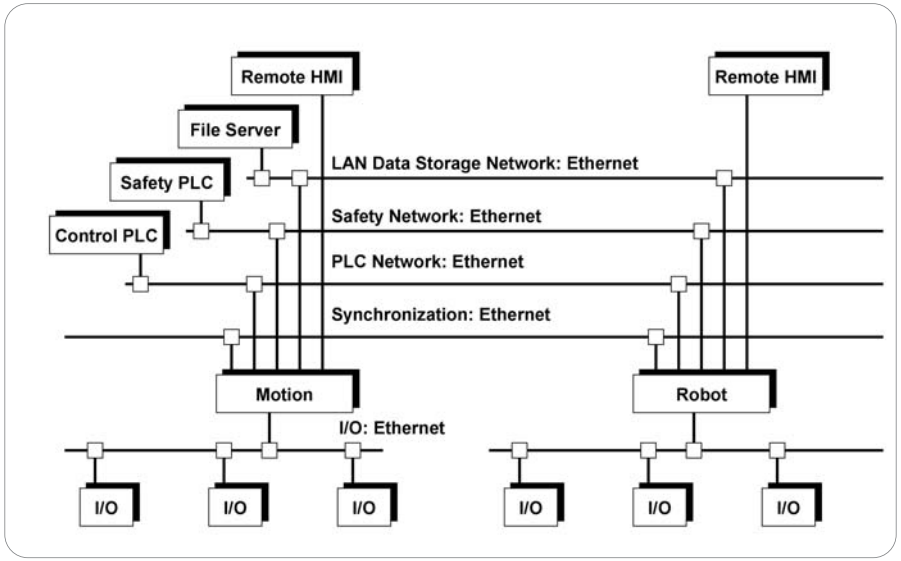
Als men een typische machine beschouwt, dan blijken daarin vaak een aantal bekabelingsstructuren (al dan niet in de vorm van een netwerk) parallel te lopen. Er zijn netwerken voor communicatie met de buitenwereld, netwerken voor communicatie tussen besturingen onderling, remote I/O systemen, bekabeling voor safety (eventueel kan dit ook via een netwerk), misschien nog een motion netwerk, en voor de supersnelle I/O enkele digitale signalen. Uiteraard is dit geen goedkope oplossing, maar tot nu toe is het niet mogelijk het anders te doen (figuur 1-1).



Figuur 1-1: In een systeem kunnen parallel aan elkaar verschillende netwerken en/of bekabelingsstructuren lopen. Voor elk niveau zijn verschillende oplossingen mogelijk.

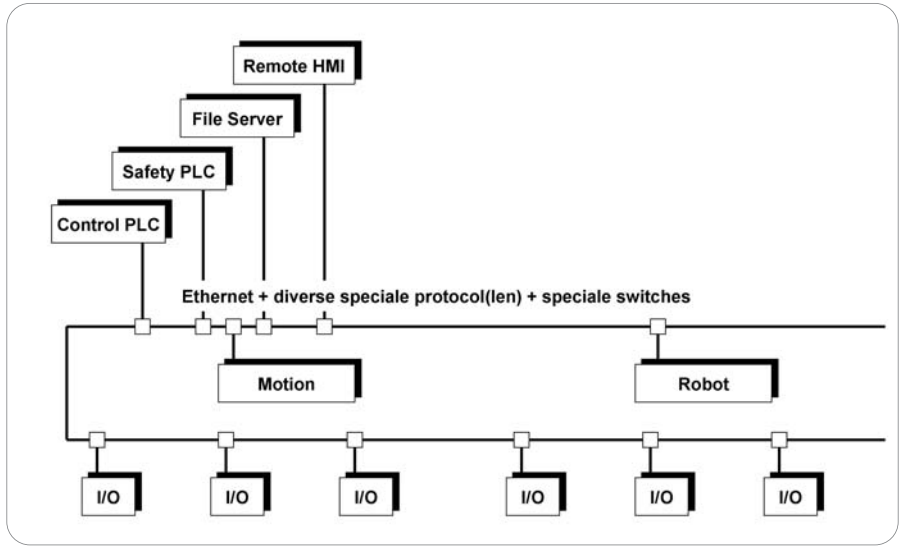
In een industrieel systeem waarin zowel Ethernet als veldbussystemen voorkwamen fungeerden de PLC's vaak als koppelvlak, doordat deze voorzien werden van twee netwerkinterfaces met dus een koppeling naar 'boven' (bv. SCADA) en naar 'beneden' (remote I/O). Het gebruik van Ethernet op deze manier is niet eens zo vreemd, want tegenwoordig heeft elke PC een eigen Ethernet-interface (bij Unix-systemen was dit al eind jaren '80 zo). Het betreft hier dus wel een industriële toepassing van Ethernet, maar eigenlijk ook weer niet – er waren meer raakvlakken met de kantoor-IT dan met de industriële IT. Niemand zou het echter in zijn hoofd halen om veldbussen te gebruiken op de hogere niveaus, of Ethernet op de lagere niveaus. De opkomst van industrieel Ethernet maakt dat dit laatste toch steeds gebruikelijker wordt.

Het zich wagen aan een toekomstvoorspelling is, net zoals in de waarzeggerij, ook voor een techneut turen in een kristallen bol gebaseerd op een extrapolatie van de technische ontwikkelingen van de afgelopen vijf jaar. Maar laat ik toch een poging wagen. Ook al heeft het gebruik van Ethernet als veldbus nu soms nog wel eens nadelen, dat wil niet zeggen dat dat volgend jaar ook nog zo is. De afgelopen jaren zijn er diverse innovaties (lees: uitbreidingen en aanpassingen aan Ethernet) gelanceerd die de nadelen een-voor-een wegnemen. In figuur 1-1 is een situatie geschetst waarbij nog verschillende bekabelingsstructuren naast elkaar bestaan. Indien industrial Ethernet een succes wordt, is het niet ondenkbaar dat op elk niveau Ethernet bruikbaar is (figuur 1-2).



Figuur 1-2: De toenemende penetratie van Ethernet, in combinatie met speciale netwerk-protocollen, maakt het misschien mogelijk om op alle niveaus Ethernet te gebruiken.

Er is dan geen reden meer om aparte bekabelingsstructuren naast elkaar te hebben – misschien is één netwerk dan voldoende voor MMI / SCADA, communicatie tussen besturingen, remote I/O, high-speed motion én safety (figuur 1-3).
Dat wil nog niet zeggen dat overal hetzelfde netwerkprotocol gebruikt moet worden – Ethernet biedt immers de mogelijkheid om meerdere protocollen parallel te verwerken, die dus onafhankelijk van elkaar een eigen bestaansrecht kunnen hebben.



Figuur 1-3: Omdat op Ethernet meerdere protocollen tegelijkertijd uitgevoerd kunnen worden, is het niet meer nodig om meerdere netwerk- en bekabelingsstructuren parallel aan te leggen. Wel zijn speciale switches nodig om het high-speed motion netwerkverkeer voorrang te geven (hier niet getekend).

1.4 Concurrentie

Welk Ethernet-gebaseerd systeem we ook beschouwen, het krijgt te maken met concurrentie van de al bestaande industriële netwerken. En dat is niet direct een gewonnen wedstrijd voor Ethernet. Als we puur kijken naar snelheid, toch vaak een zeer belangrijk aspect voor remote I/O systemen, dan kunnen de bestaande bussystemen de concurrentie met 10 Mbit/s Ethernet zéér goed aan. Met 100 Mbit/s Ethernet wordt het al veel lastiger, maar enkel als het Ethernet-systeem beschikt over goede software die geen vertragende factor vormt in de uitvoering van de gebruikte protocollen.

Ook op gebied van bekabeling, voeding voor I/O, kostprijs, omvang van de elektronica, etc. hebben bestaande bussystemen veel voordelen. Dit geldt zeker voor sensor/actuator-netwerken, in mindere mate voor remote I/O netwerken, en eigenlijk al niet meer voor bussystemen die gebruikt worden voor communicatie tussen besturingen onderling en van/naar visualisatiesystemen (HMI, SCADA).

Het industrieel Ethernet zoals we dat nu kennen is ook maar een momentopname gebaseerd op de huidige stand van de techniek. De komende jaren zullen aan de lopende band innovaties op de markt gebracht worden die de genoemde nadelen een voor een zullen wegnemen. De bestaande bussystemen zijn al verder geëvolueerd natuurlijk (omdat ze al 10...15 jaar op de markt zijn) maar industrieel Ethernet heeft nog niet zo'n lange historie. Het is daarom te verwachten dat Ethernet een steeds grotere concurrent wordt van remote I/O systemen (zoals Profibus/DP, Sercos, CAN, Interbus), en misschien ook wel van sommige sensor/ actuatornetwerken (zoals bv. AS-Interface).

Daarnaast zijn er nog andere technologieën op de markt, zoals bijvoorbeeld USB en FireWire. Omdat deze in elke moderne PC ingebouwd worden, zijn dit twee even goedkope alternatieven voor een insteekkaart voor een industrieel netwerk als Ethernet.
USB is echter speciaal ontwikkeld voor consumententoepassingen en wordt niet serieus als industrieel netwerk opgevat. Toegegeven, het is mogelijk om er I/O mee aan te sturen (een bit is immers een bit). Firewire heeft een al wat meer industriële inslag, omdat het ook veel voor high-speed motion-toepassingen gebruikt wordt. Men kan er echter maar zeer korte afstanden (enkele tientallen meters) mee overbruggen. Ik heb sterk de indruk dat als de bedrijven die nu met Firewire bezig zijn de kans zouden krijgen om nog eens opnieuw te beginnen, ze voor Ethernet zouden kiezen. Tien jaar geleden is misschien ooit een vergelijking tussen Ethernet en Firewire gemaakt en destijds zag waarschijnlijk niemand Ethernet als serieus bruikbaar in industriële applicaties, terwijl bij Firewire een hoge snelheid en real-time eigenschappen direct beschikbaar waren. De opvattingen over Ethernet zijn sindsdien gewijzigd. De protocollen voor ProfiNet, IDA en Powerlink (zie hoofdstuk 6) zijn óók sterk gericht op high-speed motion toepassingen. Als ze binnenkort uitontwikkeld zijn en hun kinderziektes overwonnen hebben, kan Firewire hier nog wel eens stevige concurrentie van krijgen.

1.5 Eén standaard voor real-time Ethernet?

Op dit moment (begin 2004) is er geen standaard voor real-time toepassingen met Ethernet. Dit is altijd een van de grote manco's geweest die het gebruik van Ethernet in industriële toepassingen in de weg stond. Desalniettemin is Ethernet al zeer lang in gebruik voor real-time systemen, echter men kon hiervoor meestal geen kant-en-klare hardware en software kopen, of men was gebonden aan één leverancier.

In 2003 is men in de IEC (International Electrotechnical Commision) begonnen met het schrijven van een norm voor real-time Ethernet. Deze wordt geacht in 2005 klaar te zijn,

en de definitieve acceptatie (na stemming door alle aangesloten landen) in 2007. Vooruitlopend op de vrijgave van deze norm worden delen ervan nu al geïmplementeerd in de nieuwe netwerkprotocollen (zie hoofdstuk 6).

2. Basis Ethernet

2.1 Ontstaan

De eerste ideeën over Ethernet komen van Xerox omstreeks 1970. Ene Bob Metcalfe bedacht een netwerk om werkstations aan elkaar te koppelen. Hier is continue verder aan ontwikkeld totdat omstreeks 1980 een snelheid van 10 Mbit/s mogelijk was. Dat lijkt nu peanuts, maar in die tijd liep de gemiddelde PC nog maar op 4,77 Mhz en derhalve leek 10 Mbit/s een "oneindig snel" netwerk.

De standaardisatie rondom Ethernet is eerst uitgevoerd door de drie bedrijven Digital Equipment, Intel en Xerox. Genoemd naar hun beginletters heette de norm destijds "DIX Ethernet"; de DIX Ethernet 2.0 uit 1982 is de basis onder de Ethernet zoals we die nu nog kennen.

De IEEE 802.3 commissie

In 1980 werd de ontwikkeling van de Ethernet-norm overgenomen door de IEEE (Institute of Electrical and Electrotechnical Engineers), en er werd een commissie opgericht voor het standaardiseren van alle technologische ontwikkelingen van LAN's. Omdat deze commissie in 1980 is opgericht en in dat jaar de tweede nieuwe commissie was, kreeg deze het volgnummer "802". Subcommissie 3 behandelt specifiek Ethernet LAN's. De volledige 'naam' van de commissie is derhalve IEEE 802.3. In veel brochures, documentatie en boeken worden "Ethernet" en "IEEE 802.3" als elkaars synoniemen gezien.

Nieuwe ontwikkelingen aan Ethernet worden (nog steeds door) de 802.3 commissie uitgevoerd. Regelmatig worden uitbreidingen op de standaard gepubliceerd. Dit wordt aangegeven met één of twee volgletters, b.v. "IEEE 802.3ah". Regelmatig worden alle uitbreidingen samengenomen en geconsolideerd in een nieuwe versie van de 802.3 norm (dit is voor het laatst gebeurd in 2002). De norm wordt overigens ook als IEC- en ISO-norm uitgebracht, als IEC / ISO 8802.3.

802.3a	1985	10BASE-2 (Thin Ethernet)
802.3c	1985	10 Mbit/s repeater specificatie
802.3d	1987	Fiber-optic repeaters
802.3i	1990	10BASE-T (Twisted-pair Ethernet)
802.3j	1993	10BASE-F (Fiber-optic Ethernet)
802.3u	1995	100BASE-T (Fast Ethernet)
802.3x	1997	Full-duplex versie
802.3z	1998	1000BASE-X (Gigabit Ethernet)
802.3ab	1999	1000BASE-T (Gigabit Ethernet via twisted-pair)
802.3ac	1999	Berichtgrootte verhoging ('frame size extension')
802.3ad	2000	Parallelle verbindingen ('link aggregration')

Tabel 2-1: Enkele uitbreidingen op de originele 802.3

Andere Ethernets

Er bestaan enkele andere netwerken die ook "Ethernet" in hun naam hebben: isoEthernet, Ethernet/IP en Safe Ethernet.

IsoEthernet (isochronous Ethernet) is een aanpassing van de 10 Mbit/s variant. Door de data op de kabel op een andere manier elektrisch te versturen is het mogelijk geworden om 16 Mbit/s data te transporteren. De "extra" 6 Mbit/s wordt gebruikt voor 97 ISDN kanalen ("96B+D" in ISDN-jargon). Hiermee zou het dan mogelijk worden om ook beeldtelefoon e.d. te gaan invoeren, c.q. telefoongesprekken via het netwerk en een PC te gaan voeren. In de praktijk komt men dit echter niet meer tegen.

Ethernet/IP is een protocol van Allen-Bradley, dat op een normaal Ethernet kan werken. Het wordt in hoofdstuk 6 in detail behandeld.

Safe Ethernet is een protocol van HIMA, dat speciaal bedoeld is voor safety-applicaties die van een netwerk (Ethernet in dit geval) gebruik willen / moeten maken. Door de netwerkbelasting laag te houden (< 10%) kan gegarandeerd worden dat het systeem altijd binnen een bepaalde tijd reageert, hetgeen zeer belangrijk is voor safety-netwerken.

2.2 Bekabelingsvarianten

Er bestaan een groot aantal bekabelingsvarianten van Ethernet. Oorspronkelijk kenden we enkel de 10Base5 versie, die gebruik maakt van een dikke (1 cm doorsnede), stugge, coaxkabel met een maximale lengte van 500 meter. Met tussenafstanden van 2,5 meter (of veelvouden daarvan) kunnen hierop maximaal 100 deelnemers aangesloten worden. De bijnaam voor deze variant is ook wel "Thick Ethernet"

De benaming van Ethernet-varianten volgt overigens een vast schema (zie ook tabel 2-1, rechterkant). Eerst wordt de snelheid in Mbit/s gegeven. Daarna volgt de transmissietechnologie (tegenwoordig altijd BASEband, vroeger ook nog BROADband). Tenslotte volgt het gebruikte type bekabeling: T en Tx = twisted pair, F en Fx = glasvezel, etc. In de eerste versies werd nog de maximale lengte (in veelvouden van 100 meter) opgegeven. Voorbeeld 1: de "100BaseF" variant werkt op 100 Mbit/s, baseband, en op glasvezel. Voorbeeld 2: "10Broad36"werkt op 10 Mbit/s, breedband, en een maximale lengte van 3600 meter. In latere versies is men hier mee gestopt, omdat dankzij het gebruik van routers, switches en hubs er eigenlijk geen maximale afstand meer is.

CheaperNet

Het gebruik van de stugge coaxkabel leidde tot veel klachten. Al snel werd een Ethernetvariant ontwikkeld die gebruik kan maken van dunne, veel flexibelere coax. Dit is de 10Base2 versie, met maximaal 30 deelnemers en een maximale lengte van 185 meter. De manier van aansluiten is ook veel eenvoudiger dan bij de 10Base5 versie. De bijnaam voor deze variant is "Thin Ethernet" of ook wel "CheaperNet".

Toch was nog niet iedereen tevreden met Ethernet. Beide manieren van bekabelen leveren immers een busstructuur op, in principe één lange leiding waarop alle deelnemers aangesloten zijn. Dit houdt ook in dat problemen op het netwerk (van kabeltechnische aard, of vanwege storingen) voor iedereen zichtbaar zijn. Tevens moet de netwerkbeheerder in de gaten houden dat het netwerk niet te lang wordt, er niet teveel deelnemers op aangesloten zijn, aan het begin en aan het eind twee terminators aangekoppeld zijn, er geen aardings- en afschermingsproblemen zijn, etc. Het is moeilijk de fysieke bron van elektrische problemen te vinden, anders dan het nalopen van de gehele bekabeling en alle aangesloten apparatuur.

Vanuit beheersstandpunt is coax-gebaseerd Ethernet dus niet praktisch, en dit zorgt ervoor dat problemen niet altijd snel opgelost kunnen worden en het netwerk voor langere tijd slecht of soms geheel niet kan functioneren, en dit leidt dan tot een lagere "uptime" en ontevreden gebruikers.

Twisted-pair

Dit is veranderd door de introductie van een nieuwe Ethernet-variant, de 10BaseT. Dit is gebaseerd op twisted-pair bekabeling (nóg goedkoper en eenvoudiger dan coax), én de introductie van een "hub". Elke deelnemer wordt, via zijn eigen kabel, op de hub aangesloten. Deze fungeert als een soort netwerk-telefooncentrale, en schakelt alle netwerkberichten door naar de eindbestemming. Omdat iedereen nu een eigen aansluiting op de hub heeft, kunnen elektrische storingen altijd maar één deelnemer treffen. Ook het uitbreiden van het netwerk is makkelijker, omdat geen rekening gehouden hoeft te worden met de maximale lengte van alle bekabeling (uiteraard is er wel een maximum: 100m per deelnemer) en de totale hoeveelheid deelnemers wordt enkel bepaald door de aansluitmogelijkheden (het aantal "poorten") op een hub. Tenslotte is het mogelijk om intelligentie in een hub in te bouwen, men krijgt dan een zgn. "manageable hub" (welke wel duurder zijn). Hiermee kan elke poort bewaakt worden, bijvoorbeeld op het aantal netwerkfouten. Indien deze een bepaald maximum overschrijden, kan men besluiten om (op afstand) die poort af te sluiten. Dit kan men via het netwerk zelf doen, door een commando naar de hub te sturen. Een bekend protocol hiervoor is SNMP (Simple Network Management Protocol). Bij elkaar genomen geeft 10BaseT de netwerkbeheerder dus de mogelijkheid om het netwerk fysiek onder controle te houden, én om alle apparatuur op afstand (= via het netwerk zelf) te kunnen beheren. De introductie van 10BaseT heeft daarom sterk bijgedragen aan de populariteit van Ethernet, dankzij de resulterende hoge betrouwbaarheid.

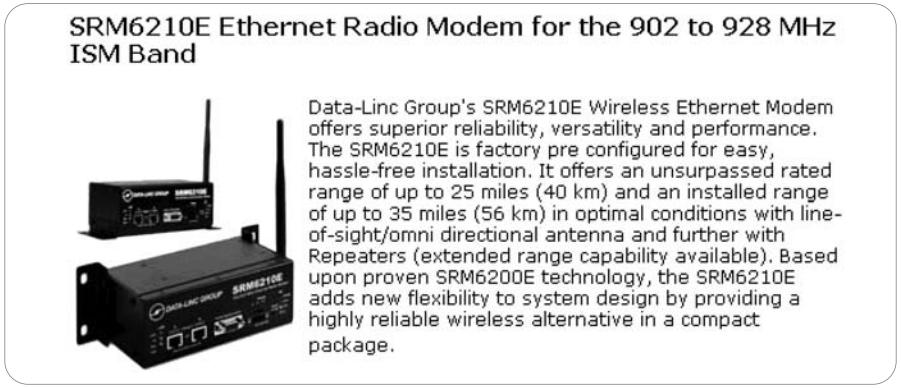
Na de 10 Mbit/s versies van Ethernet kwam een 100 Mbit/s versie. Ook deze is beschikbaar in diverse bekabelingsvarianten, maar niet meer in coax. De twisted-pair variant 100BaseTX is de meest gebruikte variant. Het lijkt zeer sterk op 10BaseT, maar er moet wel een betere kwaliteit kabel gebruikt worden: de zgn. "CAT5", welke in staat is om de hogere frequenties goed te transporteren. Voor 10 Mbit/s kan volstaan met goedkopere "CAT3" bekabeling. Het wordt echter aanbevolen ook 10BaseT netwerken aan te leggen op CAT5, dan kan men later altijd overschakelen naar 100BaseTX zonder dat alle bekabeling vervangen moet worden.

Glasvezel

Glasvezel kan bij Ethernet ook gebruikt worden (10BaseFL, 100BaseFX, etc.). De te overbruggen afstanden zijn afhankelijk van de kwaliteit van de kabel, en kunnen variëren van 400m tot enkele tientallen kilometers, zelfs op snelheden van Gbit/s en hoger. Het gebruik van glasvezel is ideaal in sterk gestoorde omgevingen, bv. lasrobots, magnetrons, zware elektromotoren, en andere elektrisch vervuilde omgevingen. Tevens is men niet gevoelig voor blikseminslag.

Draadloos

Tenslotte kennen we nog een draadloze variant van Ethernet, ook wel bekend onder de naam IEEE 802.11. Eigenlijk is het geen Ethernet-variant (geen IEEE 802.3!) maar de ontwikkelende bedrijven in de 802.11 commissie hebben besloten hun draadloos netwerk compatibel te maken met Ethernet, en het dan ook gelijk maar "draadloos Ethernet" te noemen, om zodoende te kunnen meeliften op de populariteit van het 'gewone' Ethernet.

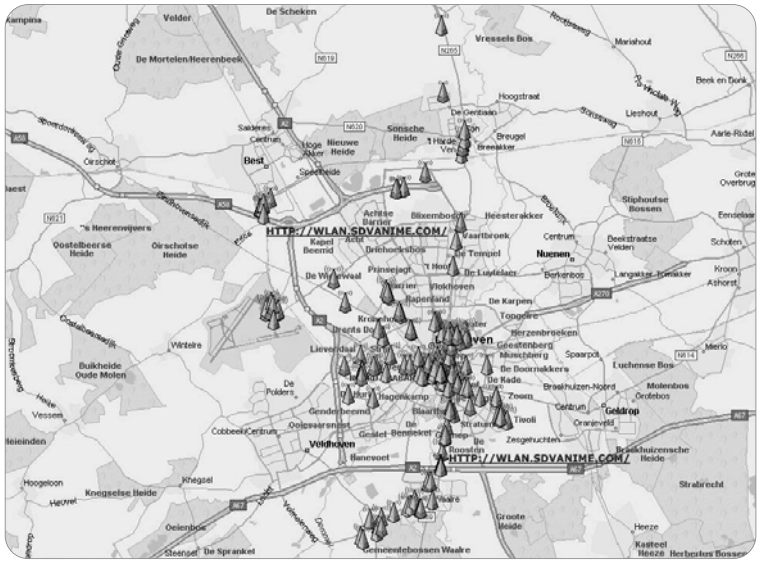


Figuur 2-1: De populariteit van Ethernet heeft ervoor gezorgd dat diverse bedrijven al heel vroeg hun eigen producten hebben ontwikkeld voor een draadloze variant. Deze zijn echter niet gebaseerd op de WiFi standaard, maar bieden daarom wel vaak extra mogelijkheden (o.a. qua afstand).

In sommige tijdschriften is wel eens gespeculeerd op het compleet draadloos zijn van een industrieel netwerk. Dit is echter een utopie. Waar krijgen alle deelnemers hun voeding vandaan? Het is geen optie om alle deelnemers in een netwerk via batterijen te voeden, althans niet als men een levensduur van enkele jaren vraagt. Alleen in geselecteerde toepassingen is batterij- of accuvoeding misschien een haalbare kaart. Bovendien heeft Ethernet hier stevige concurrentie van andere draadloze netwerken (bv. Zigbee) die juist ontwikkeld zijn voor low-power toepassingen. Alhoewel de snelheid dan meestal ook vrij laag ligt, hoeft dit voor industriële toepassingen niet altijd een probleem te zijn.

Verder moet nadrukkelijk gedacht worden aan de veiligheid; elke "hacker" kan het radiosignaal oppikken. Ondanks de aanwezigheid van een encryptiealgoritme ("WEP"- Wired Equivalent Privacy) is het met eenvoudige middelen: een PC met een netwerkkaartje en een softwarepakketje, mogelijk om het beveiligingsalgoritme in een tijdsbestek van niet meer dan ca. 15 minuten te kraken (deze 'sport' heet ook wel 'wardriving',

zie *http://wlan.sdvanime.com/map.php* voor een overzicht van alle gevonden draadloze netwerken in Nederland). De laatste jaren is er gelukkig veel aandacht voor de beveiligingsaspecten van draadloze netwerken.



Figuur 2-2: Kaart van de met "wardriving" gevonden draadloze netwerken in Eindhoven en omgeving.

2.3 Snelheid

Ethernet is ooit begonnen op een snelheid van 10 Mbit/s. Daarna kwam de 100 Mbit/s versie, en vervolgens een 1 Gbit/s versie. In 2003 is ook de 10 Gbit/s versie uitgekomen. Voor industrieel gebruik, zeker op de lagere niveaus in de industriële automatisering zal 100 Mbit/s in de meeste gevallen snel genoeg zijn. De 10 Mbit/s versie is op zich ook goed bruikbaar, maar de moderne industriële netwerken kunnen de concurrentie makkelijk aan, zijn makkelijker te bekabelen en goedkoper.

Een Ethernet op 1 Gbit/s is voor veel industriële applicaties complete overkill. Overigens krijgt men die hoge snelheid niet cadeau; bij transport van kleine hoeveelheden data moet rekening gehouden worden met een factor 8 extra overhead. Juist op de lagere niveaus in de industriële automatisering wordt vaak met kleine hoeveelheden data gewerkt.

Het gebruik van een 1 Gbit/s Ethernet lijkt daarom enkel zinvol op de hogere niveaus, waarbij grotere hoeveelheden data getransporteerd worden, of data van lagere niveaus gebundeld wordt aangeboden. Uiteraard biedt een 1 Gbit/s Ethernet goede mogelijkheden voor transport van audio- en video-signalen. Het gebruik van de 1 Gbit/s versie wordt wel steeds makkelijker, nu er ook PC-insteekkaarten komen (o.a. van Intel) die deze snelheid aankunnen.

Net zoals bij alle andere netwerken moet ervoor gezorgd worden dat alle aangesloten apparaten op een gemeenschappelijke bitrate kunnen werken. Bij de beide coax-varianten van Ethernet is dit geen probleem, want die werken enkel op 10 Mbit/s. Bij de UTP-varianten is er keuze uit 4 verschillende snelheden en dit kan lastig worden. Als alle aangesloten apparatuur echter voorzien is van "auto-negotiation" mogelijkheden dan kan elk apparaat zelf uitzoeken op welke snelheid gewerkt moet worden. Dit maakt het ook vrij makkelijk om een netwerk te migreren naar een hogere snelheid, bv. van een 10 naar een 100 Mbit/s netwerk, eventueel met gedurende een bepaalde tijd een mix van 10 en 100 Mbit/s apparatuur op hetzelfde netwerk. Let er op dat ook de hubs voorzien moeten zijn van auto-negotiation mogelijkheden, dit is namelijk niet standaard aanwezig.

2.4 Netwerkadressen

Net zoals de meeste andere netwerken moet elke deelnemer op een Ethernet een eigen, uniek, netwerkadres[1] hebben. Dit wordt ook wel het "MAC-adres" genoemd (MAC=Medium Access Control).

Ethernet heeft een zeer ruim adresbereik: een 48-bits getal. Hiermee zijn ruim 281 biljoen adressen te maken, ruimschoots genoeg dus om wereldwijd unieke adressen te hebben. Om dit te realiseren moet natuurlijk nog wel voorkomen worden dat twee verschillende bedrijven bij toeval hetzelfde adres gebruiken. Daarom worden de 48 bits in twee groepen van 24 bits opgedeeld:

[1] Het Ethernet netwerkadres wordt alleen gebruikt voor ethernet zelf. Voor de hogerliggende protocollen kan het nodig zijn om ook daar unieke netwerkadressen te hebben. Bijvoorbeeld, overal waar TCP/IP gebruikt wordt is een uniek "IP-adres" nodig. de toewijzing, nummering en formaat staat verder los van het Ethernet netwerkadres. Zo is een Ethernet adres 48 bits groot, een IP Versie 4 netwerkadres maar 32 bits, en een Versie 6 netwerkadres 128 bits.

- De hoogste 24 bits geven de leverancier aan. Dit heet ook wel de "Organisationally Unique Identifier" (OUI).
- De laagste 24 bits zijn vrij te bepalen door elke leverancier zelf.

De leverancier is vrij in de toewijzingen van 'zijn' 24 bits, zolang deze ook telkens maar eenmalig gebruikt worden. Met 24 bits geeft dit altijd nog 16 miljoen mogelijkheden, en mochten die op zijn, dan kan bij de IEEE een nieuwe groep aangevraagd worden (zie *http://standards.ieee.org/ regauth/oui/forms/*).

Als een leverancier een netwerkmodule bouwt, moet het Ethernet-netwerkadres er in geprogrammeerd worden. Dit gaat vaak door middel van een kleine (E)PROM of iets dergelijks. Om later terug te kunnen vinden welk adres er in geprogrammeerd is, wordt er vaak een sticker opgeplakt waar het 48-bits adres op staat. Dit wordt meestal in hexadecimale code afgedrukt, dus 6 getallen (6*8=48 bits), bijvoorbeeld "03-80-86-12-20-04". De eerste drie getallen (03-80-86) zijn dan de OUI, en de "12-20-04" een leveranciers-specifiek volgnummer.

Er zijn een aantal speciale waardes van het MAC-adres mogelijk. Dit levert dan een zgn. "broadcast adres" of een "multicast adres" op (zie hieronder). In alle andere gevallen hebben we steeds een MAC-adres van één deelnemer op een netwerk.

De "Organisationally Unique Identifier"
De eerste drie bytes (24 bits) van het MAC-adres vormen de zgn. "Organisational Unique Identifier" (OUI), soms ook wel "Company ID" genoemd. De OID wordt op aanvraag (na betaling van een klein bedrag) toegewezen door de IEEE. Van de 8 miljoen mogelijke OUI's is nog maar een klein deel toegewezen; de lijst is op te vragen op de website van de IEEE. Enkele voorbeelden:

00-A0-45	Phoenix Contact
00-09-5C	Philips Medical Systems
00-05-1A	3Com

Sommige bedrijven hebben meerdere OUI's gekregen; dit is niet zozeer omdat ze die perse nodig hadden maar is meestal een gevolg van overnames of fusies, zoals bijvoorbeeld Advantech China (met OUI 00-0B-AB), Advantech Taiwan (met OUI 00-D0-C9) en Advantech Canada (00-E0-02). Anderzijds zullen er ook OUI's zijn die nooit meer gebruikt worden omdat het bijbehorende bedrijf niet meer bestaat.

Het eerste byte van de OUI zal altijd een even waarde zijn. Dit komt omdat het laatste bit van het eerste byte altijd een '0' moet zijn. Dit bit wordt namelijk niet gebruikt voor het bepalen van de MAC-adressen van individuele deelnemers op een netwerk, maar voor het toewijzen van een zgn. "multicast" adres (zie onder).

Voor gebruikers van het PPP-protocol is er nog een speciale waarde van de OUI (CF-xx-yy) mogelijk; zie RFC-2153 (*http://www.faqs.org/rfcs/rfc2153.html*) voor verdere details hierover.

Indien men zelf MAC-adressen wil bepalen, is het natuurlijk altijd mogelijk om een eigen waarde voor een OUI in te vullen, zonder dit bij de IEEE aan te vragen. Dit is technisch geen probleem; het netwerk zal functioneren. Uiteraard draagt men dan wel zelf de consequenties als er dubbele MAC-adressen ontstaan op hetzelfde netwerk.

Individual Address Blocks (IAB)
Voor die bedrijven die geen behoefte hebben aan 16 miljoen MAC-adressen is er de mogelijkheid om een groep van 4096 MAC-adressen bij de IEEE aan te vragen. Dit wordt een zgn. "Individual Address Block" (IAB) genoemd. Van het 48-bits Ethernet-adres kan men dan alleen de laagste 12 bits zelf toekennen; het restant wordt door de IEEE bepaald. IAB MAC-adressen beginnen altijd met 00-50-C2. Enkele voorbeelden:

00-50-C2-00-30-00	t/m	00-50-C2-00-3F-FF	Microsoft (Redmond, US)
00-50-C2-01-10-00	t/m	00-50-C2-01-1F-FF	Bihl & Wiedemann GmbH (Mannheim, D)
00-50-C2-13-C0-00	t/m	00-50-C2-13-CF-FF	NBG Industrial (Nederweert, NL)

Het nut van zulke kleine blokken MAC-adressen ligt niet zozeer in het gebruik in LAN's, maar eerder in het gebruik van Ethernet in embedded applicaties. Een bedrijf kan dan zijn 'eigen' MAC-adressen op een netwerk herkennen, en eventueel speciale acties hierop ondernemen.

Voor de werking van een Ethernet maakt het geen verschil of er IAB MAC-adressen gebruikt worden of niet – voor Ethernet is het een 'gewone' rij van 48 bits; de speciale betekenis wordt alleen door de IEEE er aan gegeven om administratief-technische redenen (zuinig omgaan met de MAC-adressen).

Broadcast adressen

Indien het bestemmings MAC-adres de waarde FF-FF-FF-FF-FF-FF heeft, dan geeft de zender van het netwerkbericht aan dat het een 'broadcast' bericht is, en door iedereen op het netwerk geaccepteerd zou moeten worden. Dit gebeurt overigens automatisch door elke Ethernet controller-chip. Aan de hand van het 'type' veld kan de zender nog aangeven om welk protocol het gaat; enkele voorbeelden:

0600	XNS protocol (Xerox Network Services)
0800	IP (e.g. RWHOD via UDP) as needed
0804	CHAOS protocol
0806	ARP (for IP and CHAOS) as needed
0BAD	Banyan protocol
1600	VALID protocol
8035	Reverse ARP (RARP) protocol
807C	Merit Internodal (INP) protocol
809B	EtherTalk protocol

Elke deelnemer zal alleen die broadcast-berichten verwerken als het aangegeven protocol lokaal ondersteund is. Indien niet, dan zal het broadcast-bericht op deze deelnemer genegeerd worden. Deze filtering zal normaliter in software uitgevoerd worden. Het gebruik van broadcasts dient daarom beperkt te blijven, omdat het immers een belasting vormt voor alle aangesloten deelnemers, ook voor diegenen die geen interesse hebben in de broadcast-netwerkberichten. Bijvoorbeeld, in het XNS-protocol zal elke deelnemer 24 netwerkberichten per minuut per broadcast sturen. In grotere netwerken kan dit al gauw een aanzienlijk percentage van de netwerkbelasting vormen, of (waarschijnlijker) een nog groter percentage van de CPU-belasting van de aangesloten deelnemers.

De zender van een netwerkbericht kan ook FF-FF-FF-FF-FF-FF in het afzender-veld plaatsen. De consequentie is dat een antwoord door iedereen ook weer per broadcast verstuurd zal worden. Dit kan aanleiding geven tot een zgn. "broadcast storm".

Multicast adressen

Indien men wel broadcast-achtige functionaliteit gebruiken wil maar toch niet elke deelnemer op een netwerk wil bezwaren, kan het gebruik van "multicast" MAC-adressen een oplossing zijn. Hierbij kan een netwerkbericht naar een *groep* deelnemers op het netwerk gestuurd worden.

Een multicast MAC-adres is herkenbaar aan de waarde van bit 0 van het eerste byte in het MAC-adres: de waarde moet '1' zijn. Waarom de vreemde locatie van het multicast bit? Het zou logischer lijken om het multicast bit in bit 7 van het eerste byte te plaatsen i.p.v. in bit 0. Echter, op een ander netwerkniveau gebeurt dat ook. De transmissie van een byte in Ethernet begint altijd met bit 0, dan bit 1, etc. tot bit 7. Elke ontvanger van een netwerkbericht weet na het eerste bit van het MAC-adresveld dus al of het om een multicast-adres gaat of niet.

Multicast over Ethernet kent een directe relatie met het IP (van TCP/IP) protocol. IP netwerkadressen zijn 32 bits groot, en een (klein) deel hiervan is gereserveerd voor IP multicasts. Deze zgn. "GDA's" (Group Destination Networkaddress) liggen in het zgn. "Klasse D" adresbereik van IP: 224.0.0.0 t/m 239.255.255.255. Binair gezien zijn hiervan de hoogste 4 bits steeds '1' om aan te geven dat het om een IP multicast gaat; de laagste 28 bits geven een zgn. "multicast adres" aan. De GDA's hebben op zich niets te maken met Ethernet, het is immers ook mogelijk om het IP-protocol uit te voeren op andere netwerken dan Ethernet. Maar om het gebruik van multicasts via IP in combinatie met een Ethernet zo makkelijk en efficient als mogelijk te maken, krijgt elke GDA een bijbehorend Ethernet multicast adres.

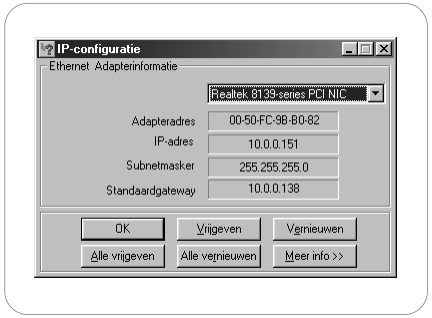
De conversie van een GDA naar een Ethernet MAC-adres is als volgt:

- De eerste 3 bytes krijgen de waarde 01-00-5E. Dit is een speciaal hiervoor gereserveerde OUI (zie hierboven). Dit maakt het voor switches en routers vrij makkelijk om multicasts uit te filteren, hetgeen vooral interessant is voor "IGMP Snooping".
- De resterende 3 bytes worden gevuld met de laagste 23 bits van het GDA. De resterende 5 bits worden niet meegenomen (er is eenvoudigweg geen ruimte voor).

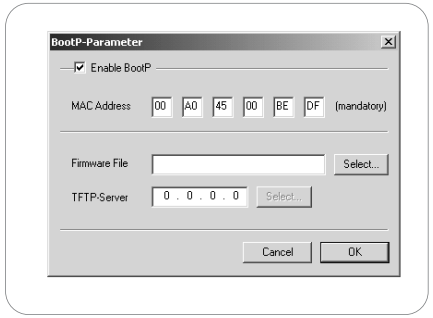
Bijvoorbeeld, een IP-adres 224.0.0.1 wordt geconverteerd naar MAC-adres 01-00-5E-00-00-01, en IP-adres 225.10.20.30 naar IP-adres 01-00-5E-0A-14-2D. Let op dat een volledige conversie van een GDA naar een MAC-adres niet mogelijk is; 32 GDA's leveren immers steeds hetzelfde MAC-adres op. Het is dus niet 100% mogelijk om op basis van MAC-adressen multicasts te filteren. Dit kan alleen door IP zelf gedaan worden (in software).

Er is één zeer speciaal multicast adres: 01-80-C2-00-00-01. Eigenlijk heeft het niets te maken met multicast transmissies, omdat het enkel wordt gebruikt voor flow-control toepassingen tussen twee apparaten onderling. Het principe achter flow-control wordt in detail uitgelegd in hoofdstuk 5.

Opzoeken van het MAC-adres.
Omdat het niet altijd makkelijk is (of sowieso mogelijk) om een apparaat te openen en de EPROM / flashprom uit te lezen, plakken veel leveranciers een sticker op hun apparaat waar het Ethernet netwerkadres dan op afgedrukt is. Meestal is het ook wel mogelijk om het MAC-adres via software op te vragen, alhoewel men er niet op moet rekenen dat dat altijd kan, zeker niet bij de exotischer besturingssystemen (hetgeen in de industriële automatisering nog wel eens voorkomt).



Figuur 2-3: Een voorbeeld van de opgave van het MAC-adres ("Adapteradres") van een PC-netwerkkaart



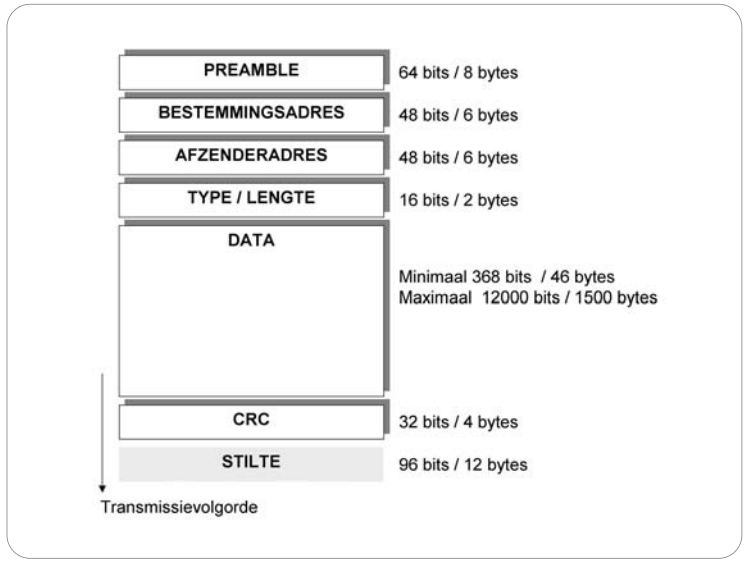
Figuur 2-4: Een voorbeeld van een softwarepakket voor het "BOOTP" protocol, waarbij alléén die deelnemers herkend worden waarvan men het MAC-adres ingegeven heeft.

Figuur 2-3 en 2-4 geven twee voorbeelden van het werken met MAC-adressen.
Op een PC kan men het MAC-adres van de eigen netwerkkaart eenvoudig te weten komen via de menu's Start -> Control Panel / Configuratiescherm -> Settings / Instellingen, en ga dan verder met de gewenste netwerkdriver. De MAC-adressen van alle communicatiepartners zijn op te vragen via het commando "arp –a", dat overigens ook op Unix-varianten aanwezig is.

Let op dat er allerlei verschillende benamingen gebruikt worden ("MAC-ID", "fysisch adres", "Ethernet address", "EA", "Adapter adres", "INC", etc.). Het feit dat ze altijd zes bytes groot zijn, maakt het MAC-adres meestal snel herkenbaar. De officiële schrijfwijze is met streepjes tussen de 6 hexadecimale getallen, maar ook dubbele punten komen veel voor.

2.5 Berichtstructuur

Elk Ethernet-bericht heeft altijd een standaard opbouw, waarin zes velden te herkennen zijn (Figuur 2-5).



Figuur 2-5: Opbouw van een Ethernet-bericht, welke altijd uit zes velden bestaat. Na afloop volgt gedurende een bepaalde tijd de "Interpacket spacing" stilte, welke geen deel is van het feitelijke netwerkbericht, maar wel van belang is bij snelheidscalculaties.

Elk bericht begint met een preambule (preamble, aanhef) die 64 bits groot is, en bestaat uit een afwisselende combinatie van 0- en 1-bits (behalve het laatste bit, dat is ook een 1). Deze wordt gebruikt om de kloksnelheid van de ontvanger precies gelijk te laten lopen (synchroniseren) met die van de zender: ook al zijn beiden op dezelfde snelheid ingesteld (10, 100, .. Mbit/s), door kleine verschillen in elektronische onderdelen en ook onder invloed van temperatuurswisselingen kunnen er toch kleine verschillen optreden. Door per netwerkbericht te synchroniseren worden snelheidsverschillen gecompenseerd.

Daarna volgt het 48-bits netwerkadres (MAC-adres) van de bestemming, waar het netwerkbericht dus afgeleverd moet worden. Direct daarna komt het eigen netwerkadres.

Het vierde veld is 16 bits groot, en geeft aan hoeveel bytes applicatiedata er in het data-veld komen. Soms staat er ook in wat voor soort data er in het dataveld staat; dit heeft te maken met een van de eerste versies van Ethernet die nu eigenlijk niet meer voorkomt. Met 16 bits kunnen waardes tussen de 0 en 65535 aangegeven worden, maar er mag toch nooit meer dan 1500 bytes data per netwerkbericht verstuurd worden. De enige uitzondering hierop zijn de zgn. "jumbo frames". Het vijfde veld heeft een variabele opbouw, en bevat applicatiedata. Wat hier precies in staat maakt voor Ethernet niet uit, de data wordt 'zoals het is' verstuurd en op de bestemming afgeleverd, die wel weet wat er met die data moet gebeuren (vergelijk het met de PTT, die ook niet weet wat er precies in een brief staat, en er gewoon voor zorgt dat het in de juiste brievenbus belandt).

Het dataveld is minimaal 46 bytes groot, en maximaal 1500 bytes. Heeft men minder dan 46 bytes data te sturen, dan moeten er extra bytes aan toegevoegd worden. Waarom 46? Dit heeft te maken met de minimale berichtgrootte van 64 bytes (64 bytes minus CRC-veld á 4 bytes, minus lengte-veld á 2 bytes, minus twee adresvelden á 6 bytes elk = 46 bytes). Dus ook al heeft men maar één byte data te sturen, toch zullen er 46 databytes de lijn over moeten gaan.

Het laatste veld in elk netwerkbericht is de "Cyclic Redundancy Check" (CRC), die gebruikt wordt voor de gegevensbeveiliging. Ethernet kan hieraan zien of er tijdens een transmissie een storing is opgetreden, bv. dat een 0-bit in een 1-bit gewijzigd is (of andersom), en zal het beschadigde bericht dan weggooien. Het is dan aan de software om dit te detecteren, en het bericht nogmaals te versturen (dit is bv. een van de taken van TCP).
Als laatste zit er tussen twee Ethernet-berichten altijd 96 bits stilte, de zgn. "Interpacket Spacing". Hiermee wordt de ontvanger de tijd gegund om een bericht intern te verwerken en om klaar te zijn voor een ontvangst van een nieuw netwerkbericht. Dit veld is geen onderdeel van het feitelijke netwerkbericht. Echter, bij snelheidsberekeningen kan het wel degelijk van belang zijn. In veel publicaties in de vakpers blijkt dat men dit veld veelal vergeet, en dat scheelt toch zo'n 15%.

Efficiency
Als men de lengte van alle velden bij elkaar optelt, is er altijd minimaal 64+48+48+16+46*8+32+96 = 672 bits transmissietijd (= 67,2 µsec op 10 Mbit/s, 6,72 µsec op 100 Mbit/s) nodig, zelfs als men maar 1 byte aan data stuurt (met 45 bytes opvulling). Dit is dus een efficiency van 8 / 672 = 1,2%. Dit toont aan dat Ethernet niet zo vreselijk geschikt is om kleine hoeveelheden data mee te sturen, iets dat juist in industriële netwerken (veldbus) gebruikelijk is.

Als gebruik gemaakt wordt van IEEE 802.1P of van "Virtual LAN" mogelijkheden in Ethernet, dan komt er één extra veld van 4 bytes (32 bits) in het netwerkbericht bij. Dit is echter niet van belang bij de meeste efficiencyberekeningen, omdat de minimumlengte van het dataveld dan afneemt van 46 naar 42 bytes. Er blijven dus altijd minimaal 672 bits transmissietijd nodig.

Jumboframes
In 1998 heeft de fa. Alteon (nu onderdeel van Nortel) voor hun eigen producten zgn. "Jumbo frames" ingebouwd. Dit zijn qua opbouw normale Ethernet berichten, maar met een dataveld van maximaal 9000 bytes. Het idee achter jumbo frames is het verminderen van de overhead die de verwerking van elk netwerkbericht kost. Op een snelheid van 1 of 10 Gbit/s kunnen er zoveel netwerkberichten per seconde binnenkomen (maximaal 19000000 kleine netwerkberichten, of 800000 berichten met 1500 byte data) dat geen enkele processor dit nog verwerken kan. Door zes keer zoveel data in een netwerkbericht te sturen, wordt de belasting van de processor met 5/6 verlaagd.
Jumbo frames zijn geen officieel onderdeel van Ethernet, maar worden wel ondersteund door veel leveranciers. Via 'auto negotiate' zal run-time worden uitgezocht of beide partijen op een full-duplex lijn dit ondersteunen.

Let er op dat een jumbo frame in kleinere stukken wordt opgedeeld indien zo'n frame bij een router komt naar een netwerk waarop jumbo frames niet ondersteund zijn. Eenmaal opgesplitst zullen deze kleine stukken (= normale grootte Ethernetberichten!) niet meer tot een nieuw jumbo frame geassembleerd worden, ook niet als een volgend netwerksegment weer wél jumbo frames kent.

Maximum Transmission Unit (MTU)
Indien men met TCP/IP werkt, dan moet in de configuratie van IP ingesteld worden wat de maximum hoeveelheid data kan zijn die per netwerkbericht verzonden kan worden. Dit wordt de "Maximum Transmission Unit" (MTU, MaxMTU) genoemd. Aangezien in een Ethernet-bericht ruimte is voor 1500 bytes data, zal de MTU dus ook op 1500 ingesteld kunnen worden. Indien een applicatie een grotere hoeveelheid data zal willen sturen, dan wordt dit door IP automatisch opgesplitst in kleinere stukken van hooguit MTU bytes.

Het is mogelijk om de MTU op een kleinere waarde dan 1500 in te stellen. Dit heeft enkel zin indien in het bekabelingstraject tussen twee apparaten ergens gewerkt wordt met langzame of storingsgevoelige segmenten, zoals b.v. een telefoonlijn, ADSL, wireless Ethernet, of bij sommige internetproviders. Als namelijk in een netwerkbericht een storing

optreedt, dan zal dat gehele netwerkbericht nog eens verstuurd moeten worden. Hoe langer het netwerkbericht is, des te groter de kans dat het getroffen kan worden door een storing, en dan betaalt men nóg eens een prijs door de hertransmissie. Het kan dan beter zijn om MTU op een lagere waarde in te stellen, hetgeen de communicatiesnelheid soms aanzienlijk kan verhogen, eenvoudigweg omdat minder tijd wordt besteed aan hertransmissie van lange netwerkberichten.

Op zich heeft dit eigenlijk niets met Ethernet te maken, maar bij veel systemen is het wel zo dat als MTU verlaagd wordt, dit geldt voor *alle* netwerkberichten, ook die het onderliggende Ethernet in het geheel niet verlaten.

De Maximum *Transmission* Unit is tevens de Maximum *Receive* Unit, alhoewel men deze parameter nergens zal tegenkomen (het komt wel eens voor dat er Ethernet-drivers zijn die ontvangen netwerkberichten met een lengte groter dan de MTU geheel negeren).

2.6 Ontvangst van netwerkberichten

Een Ethernet netwerkkaart zal allerlei netwerkberichten ontvangen. Welke dit exact zijn hangt af van de manier van bekabelen: coax of twisted-pair, gebruik van een hub of een switch. De netwerkkaart "weet" wat zijn eigen MAC-adres is, want dit is door de leverancier ingeprogrammeerd (of door de gebruiker veranderd). De netwerkkaart zal dus alléén die berichten filteren (doorlaten) waarvoor geldt dat:

- Het eigen MAC-adres is identiek aan het bestemmings MAC-adres in de ontvangen netwerkberichten.
- Het bestemmings MAC-adres in de ontvangen netwerkberichten is gelijk aan het speciale broadcast MAC-adres (48 maal '1').

Alle andere netwerkberichten worden verder geheel genegeerd.

Promiscuous mode

Bij sommige netwerkkaarten is het mogelijk om deze in "promiscuous mode" in te stellen. Dit is een speciale optie die het mogelijk maakt om zo'n netwerkkaart te gebruiken voor een netwerkanalyzer. Deze is dan in staat om al het netwerkverkeer te volgen en te gebruiken voor verdere analyse (zie ook hoofdstuk 7).

Het kunnen werken in promiscuous mode van een deelnemer kan een gevaar opleveren voor de veiligheid in een netwerk, omdat immers alle dataverkeer gevolgd kan worden. Soms kan van buitenaf vastgesteld worden of een bepaalde deelnemer in promiscuous mode ingesteld staat; op internet is hiervoor enige programmatuur te vinden.

Verschil tussen coax en twisted-pair bekabeling

Op een coax-bekabeld Ethernet zal elke deelnemer automatisch alle netwerkberichten voorbij zien komen; coax is immers een gedeeld medium waarop iedereen meeluistert. Voor het gebruik van een netwerkanalyzer is dit zeer handig. Dit geldt ook indien men gebruik maakt van twisted-pair bekabeling (10BaseT of 100BaseTX) en een hub; ook dan ontvangt elke deelnemer alle netwerkberichten. In deze gevallen werkt het MAC-filter dus precies zoals bedoeld.

Op een twisted-pair bekabeld Ethernet met een switch zal de MAC-filtering al door de switch worden uitgevoerd; een deelnemer ontvangt zowiezo enkel de netwerkberichten die voor hem bestemd zijn. Toch zal de netwerkkaart deze filtering nog eens herhalen. Dit is in feite overbodig, maar is niet nadelig voor de snelheid. Wel heeft het instellen van promiscuous mode dan weinig zin.

2.7 Virtuele netwerken

VLAN's (Virtual LAN's) is een uitbreiding op Ethernet die oorspronkelijk uit de kantoorwereld komt, en is bedoeld voor het scheiden van netwerkverkeer van groepen deelnemers die elkaar niet mogen 'zien' op het netwerk. U kunt hierbij o.a. denken aan: afscherming van confidentiële informatie die niet op andere afdelingen terecht mag komen, voor het uitvoeren van een scheiding tussen 'vaste' PC's en laptops, afkoppeling van testsystemen, aansluiten van apparatuur van bezoekers of inhuur, of het gewoon opdelen van een bedrijf in subnetwerken i.v.m. bescherming tegen virussen en wormen. Anderzijds is het natuurlijk ook wel makkelijk als iedereen in een bedrijf op hetzelfde netwerk is aangesloten; uiteraard kan men best meerdere onafhankelijke netwerken aanleggen, maar bij elke verhuizing of uitbreiding kan zowat opnieuw begonnen worden. Met VLAN's kan men flexibeler hierop inspelen - men heeft één infrastructuur, maar kan toch meerdere 'virtuele' netwerken maken die verder onafhankelijk van elkaar opereren. Dit alles wordt via software ingesteld.

Hoe gaat zoiets in zijn werk? Via een uitbreiding op het Ethernet-protocol genaamd IEEE 802.1q krijgt iedereen die op het netwerk aangesloten is, ingesteld op welk virtueel net-

werk hij moet werken. Er kan alleen gecommuniceerd worden met andere deelnemers die ook voor hetzelfde VLAN ingesteld zijn. Van welk VLAN men lid is is te zien in de "tag", een veld van 4 bytes dat elk netwerkbericht extra wordt meegestuurd. Men zou het enigzins kunnen vergelijken met het hebben van meerdere 'kanalen' op een radioband, en als men de juiste frequentie kent kan men meedoen (zenden/ontvangen) op dát ene kanaal.

Technisch gezien is het werken met VLAN's voor 'gewone' netwerkberichten niet echt anders dan bij een standaard Ethernet – iedereen voor wie het netwerkbericht niet bestemd is, ziet het ook niet. Anders wordt het als er sprake is van broadcasts: een transmissie van één netwerkbericht bedoeld voor alle aangesloten apparaten op het netwerk. Broadcasts worden veel gebruikt voor netwerkbeheer, maar ook voor het 'zoeken' naar servers (applicatieservers, fileservers, printerservers). Door alleen maar te luisteren op een netwerk krijgt men dus vanzelf te horen hoe de infrastructuur er uit ziet, en dit kan dan als basis van verdere inbraaktactieken dienen. Het is hier waar VLAN's kunnen helpen – een broadcast in een VLAN zal alléén maar doorgestuurd worden naar iedereen die in dit VLAN actief is, maar niet naar de rest (eigenlijk is het dus –qua definitie- geen echte broadcast meer).

Een switch (of router) in een netwerk speelt een belangrijke rol bij het gebruik van VLAN's - deze apparaten 'weten' immers wie er allemaal op het netwerk aangesloten zijn, en ook op welk VLAN iedereen actief is. Indien men VLAN-ondersteuning nodig heeft, moeten de switches dit ook kunnen.

Na de bovenstaande opsomming van toepassingsgebieden lijkt het gebruik van VLAN's in veel industriële toepassingen overbodig. Toch kan men er zeer nuttig gebruik van maken, niet zozeer om categorieen gebruikers te scheiden, maar om functionele scheidingen in een netwerk aan te brengen. Denk hierbij o.a. het real-time verkeer dat een besturing genereert, en het netwerkverkeer van / naar HMI (Human/Machine Interface) of hogerliggende bedrijfsprocessen. Bij de diverse netwerkprotocollen voor industrieel Ethernet die op dit moment allemaal in ontwikkeling zijn (zoals ProfiNet, IDA, Powerlink, etc.) wordt heel veel gewerkt met broadcasts. Dat zou dus een enorme belasting geven voor de resterende deelnemers op het netwerk, want die krijgen die broadcasts ook allemaal binnen en alhoewel ze er niets aan hebben, is toch steeds enige rekenkracht nodig om een broadcast te verwerken. Door met VLAN's te werken zijn al die broadcasts te kanaliseren: wie interesse heeft meldt zich aan in het desbetreffende VLAN en kan meedoen; de rest ziet er niets van en heeft er dus ook geen last van.

2.8 Software

Uit het grote marktaandeel van Ethernet zou geconcludeerd kunnen worden dat Ethernet een defacto "standaard" is op netwerkgebied. Enerzijds is dit juist, maar anderzijds ook weer niet. Ethernet zelf regelt namelijk niets over de op het netwerk te gebruiken protocollen, een puur software-aspect tegenover de meer hardware-matige aspecten van Ethernet zelf.
In de praktijk blijkt dat een andere defacto standaard zeer veel in combinatie met Ethernet gebruikt wordt: de TCP/IP protocolfamilie. Deze combinatie is zelfs zo sterk dat velen Ethernet en TCP/IP als synoniemen zien, maar dat is niet juist – beiden kunnen zonder elkaar gebruikt worden (hetgeen men thuis ook doet als men gaat 'internetten' via de kabel, ADSL, ISDN of een telefoonmodem).
In de industriële automatisering is het gebruik van TCP/IP ook wijdverbreid, maar het is zeker niet zo'n standaard als in de kantoorautomatisering. Sterker, veel nieuwe ontwikkelingen op het gebied van industrieel Ethernet kiezen ervoor om geen gebruik te maken van TCP/IP, vanwege de slechte performance voor bepaalde types applicaties (high-speed motion bijvoorbeeld).

Net zoals bij alle andere netwerken moet dus ook bij Ethernet aandacht besteed worden aan: welk protocol gebruiken we op het netwerk? Indien twee apparaten geen gemeenschappelijke protocollen ondersteunen, kunnen ze toch niet met elkaar communiceren (ook TCP/IP helpt hier niet bij). In hoofdstuk 6 wordt een overzicht gegeven van de actuele ontwikkelingen op protocolgebied.

Ethernet heeft wel het voordeel dat het mogelijk is om meerdere protocollen tegelijkertijd over hetzelfde netwerk uit te voeren (protocolinvariantie). Bij veel industriële netwerken is dit niet mogelijk; bijvoorbeeld op een Profibus-netwerk kan enkel het Profibus-protocol uitgevoerd worden, maar verder niets. Indien men later over wil schakelen naar een ander industrieel netwerk, dan moet ook alles opnieuw bekabeld worden.
Bij Ethernet is dit niet zo; men heeft daarom meer vrijheid m.b.t. de keuze van een protocol. Bijvoorbeeld, men kan beginnen met het invoeren van een 2e protocol op het netwerk zonder dat dit merkbaar is voor het 1e protocol. De consequenties van een verkeerde keuze zijn dus minder erg, omdat de bekabeling in principe kan blijven liggen. Tevens worden migraties makkelijker.

3. Ethernet als industrieel netwerk

Het is niet vanzelfsprekend dat Ethernet als industrieel netwerk wordt ingezet.
In dit hoofdstuk bespreken we een aantal ontwikkelingen, zowel hardwaretechnisch als softwaretechnisch, die relevant zijn voor industrieel Ethernet. Hierbij dient opgemerkt te worden dat alle ontwikkelingen op dit moment (begin 2004) nog in volle gang zijn, en dat waarschijnlijk nog veel innovaties zullen volgen waarmee bepaalde bestaande nadelen opgeheven worden.

3.1 Bekabelingsaspecten

In een modern LAN wordt Ethernet altijd bekabeld met UTP, en het onafgeschermde aspect hiervan roept bij industrieel gebruik altijd de nodige vragen op. STP (Shielded Twisted Pair) lijkt beter te zijn dan UTP, vanwege de aanwezigheid van het scherm dat storingen kan afschermen van de signaallijnen in de kabel. Dat is echter niet het hele verhaal. Het gebruik maken van twisted-pair kabels is juist één van de methodes om de kans op storingen zo klein mogelijk te houden. Niet alleen Ethernet, maar ook de meeste industriële netwerken maken gebruik van twisted-pair bekabeling. Door de twists in de kabel heffen bepaalde stoorsignalen elkaar op, waardoor er geen corrumperende invloed meer is op het datatransport.

Afgeschermde kabels kunnen bij elk type netwerk soms voor meer problemen zorgen dan ze oplossen. Het scherm werkt als een antenne, en pikt dus juist EMI op, zelfs als de kabel goed geaard is. De stroompjes die over het scherm lopen veroorzaken op hun beurt een gelijkwaardige maar tegenovergesteld lopende stroompjes in de signaallijnen. Zolang beide symmetrisch zijn heffen ze elkaar op, maar elke discontinuïteit in het scherm of de signaallijnen veroorzaakt een kleine asymmetrie, en dat resulteert in een kleine storing op de signaallijnen.

Een ander nadeel van STP is de verzwakking van signalen op hogere frequenties.
De aanwezigheid van een scherm moet bij de ontwikkeling van de elektronica meegenomen worden. De effectiviteit van het scherm is zelf ook afhankelijk van het materiaal, de dikte, het soort EMI, de frequentie van de EMI, de afstand tussen de storingsbron en de kabel, onderbrekingen of gaten in het scherm, en de gevolgde aardingsmethodiek.

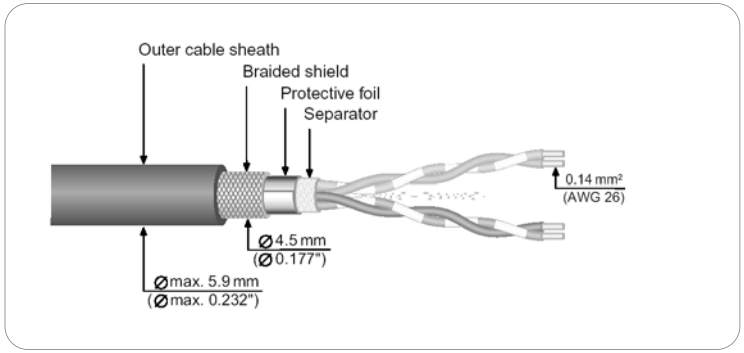
Qua uitvoering van een scherm bestaan er diverse vormen. Goedkopere kabels hebben alleen een gevlochten scherm (Shielded Twisted Pair of STP genaamd), duurdere kabels hebben hier ook nog een folie omheen (Foiled Twisted pair of FTP genaamd), en nog duurdere kabels zijn een combinatie hiervan: Shielded Foiled Twisted Pair (SFTP) of Pair In Metail Foil (PIMF). Bij deze laatste is elk aderpaar apart omvlochten.

In alle gevallen hebben de kabels een minimum buigstraal die bij de installatie strikt in acht genomen moet worden, anders kan er een scheur in het scherm ontstaan. Ook zijn er (kleine) afwijkingen in de dikte van kabels; als deze in een te kleine connector wordt aangesloten kan het scherm na een bepaalde tijd ook scheuren (men wordt daarom wel geadviseerd de kabels en de connectoren bij dezelfde leverancier te kopen).

Bij STP/FTP/SFTP-bekabelde systemen moet niet alleen de afscherming van de kabel in orde zijn, maar ook die van de gebruikte connectoren, behuizingen, en apparatuur.
Aan de aarding van alle apparatuur moet ook voldoende aandacht besteed worden.
Bij hoge frequentie-signalen moet de bekabeling tweezijdig geaard zijn. Maar dan kunnen er, vanwege potentiaalverschillen op de aardpunten, wel vereffeningsstromen gaan lopen. Dit moet met speciale elektrische maatregelen voorkomen worden. Men kan eenvoudigweg besluiten om de kabel dan maar aan één zijde te aarden, maar een eenzijdig geaard scherm werkt al helemaal niet tegen magnetische interferentie. Ook de locatie van de fysieke aarding is nog van belang; als die te ver weg is functioneert hij niet. Daarom is elk bekabeld systeem uniek; het is moeilijk om goede regels op te schrijven.

UTP, FTP en STP kabels dienen normaliter in metalen kabelgoten gelegd te worden. SFTP is veel minder storingsgevoelig, en kan daarom in een kunststof kabelgoot gelegd worden. PIMF is zo goed als storingsongevoelig, en mag daarom ook in een kunststof kabelgoot.

Ook op het gebied van EMI-emissies zijn er grote verschillen tussen STP/FTP/SFTP en UTP. In tegenstelling tot eerste vermoedens voldoen UTP-bekabelde systemen vaker aan de geldende Europese richtlijnen dan STP-bekabelde systemen (gebaseerd op de ervaringen van www.evolution.nl).



Figuur 3-1: Opbouw van de industriële Ethernetkabel van Phoenix Contact.

Standaard UTP-kabels zijn ook niet bestand tegen te lage temperaturen. Indien deze toch optreden, kan de isolatie verbrokkelen en als gevolg hiervan mogelijk sluiting optreden. Diverse leveranciers bieden dan ook industrieel-kwaliteit kabel aan (figuur 3-1). In zeer storingsrijke omgevingen kan ook nog gekozen worden voor gebruik van glasvezel. Dit heeft tegens nog het voordeel van de galvanische scheiding, en (afhankelijk van de kwaliteit) kan een grotere afstand overbrugd worden dan met koperen bekabeling (100m).

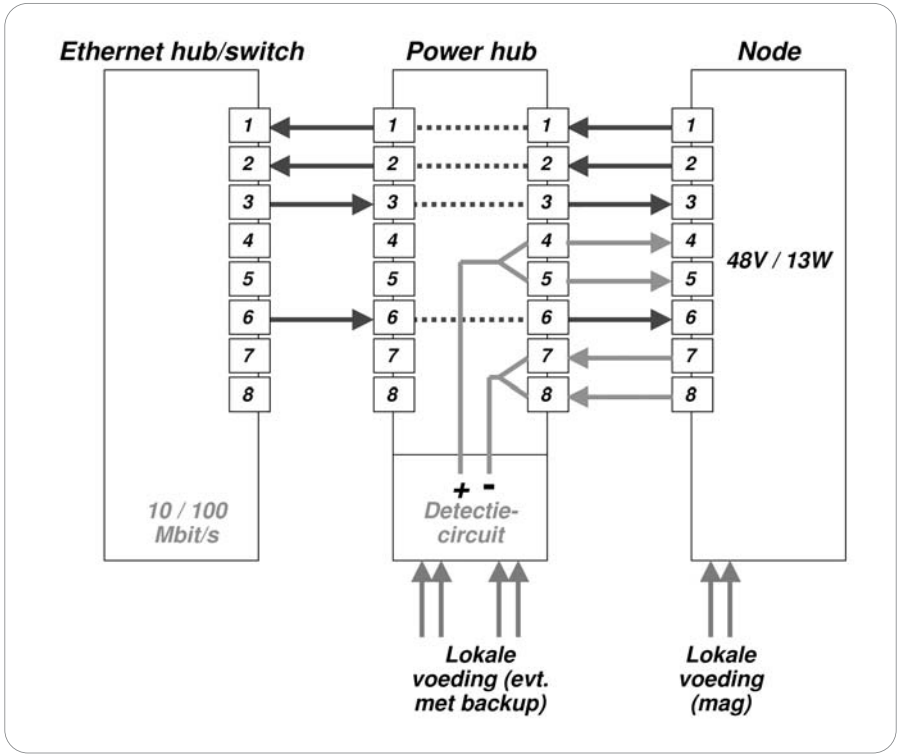
3.2 Voeding voor I/O modules

In een kantooromgeving is altijd wel ergens een 230V aansluiting aanwezig. In veel industriële toepassingen niet, en bovendien wordt ook met andere voedingsspanningen voor I/O gewerkt: meestal 24V. Bij bestaande industriële netwerken zijn soms voorzieningen aanwezig om de remote I/O elektronica via het netwerk zelf van voeding te voorzien (enkele voorbeelden):

- DeviceNet heeft 2 extra aders in de netwerkkabel (24V).
- AS-Interface moduleert het netwerksignaal bovenop de voedingsspanning (30V).
- Foundation Fieldbus en Profibus/PA moduleren het netwerksignaal boven op de voedingsspanning (15V, i.v.m. gebruik in intrinsiek-veilige toepassingen).

Bij Ethernet is het nu **niet** mogelijk om remote I/O via het netwerk van voeding te voorzien. Dit is een belangrijk nadeel voor gebruik van Ethernet in sommige applicatiegebieden waarbij het zeer gebruikelijk is dat apparatuur centraal van voeding voorzien wordt (bv. in de procesindustrie).

Een nog lopende ontwikkeling (die overigens al grotendeels uitgekristalliseerd is) wordt uitgevoerd in de werkgroep IEEE 802.3af. Het Israëlische bedrijf PowerDsine (*www.powerdsine.com*) is medeontwikkelaar van deze technologie, en heeft inmiddels de eerste producten op de markt gebracht. Hierbij wordt het wél mogelijk om apparatuur via de Ethernet-bekabeling van voeding te voorzien. Deze voeding wordt geleverd door een zgn. "power hub" (figuur 3-2) welke geschakeld wordt tussen het aan te sluiten apparaat, en een hub / switch. De power-hubs zijn eigenlijk ontwikkeld voor IP-telefonie, waarbij men telefoneert via het LAN, en niet via een telefooncentrale. Waar een telefoontoestel normaliter 48V uit de telefooncentrale krijgt, moet die nu via het netwerk aangeleverd worden.



Figuur 3-2: Schema van voeding van apparatuur via een "power hub". Getekend is de variant waarbij de voeding via de normaliter ongebruikte pinnen 4/5 en 7/8 wordt aangeboden.

De power hub zal alle netwerksignalen ongewijzigd doorgeven. Op 4 tot nu toe ongebruikte pinnen wordt 48V / 13W aangeboden (op elke poort van de hub). Er is discussie ontstaan over de voor industriële toepassingen ongebruikelijke spanning van 48V, maar dit komt omdat power hubs zijn ontwikkeld voor de zakelijke en consumentenmarkt, en niet voor de industriële markt. Inmiddels gaan er stemmen op om de tekst in de norm zo te lezen dat er staat 'maximaal 48V'. Dit biedt dan ook ruimte voor 24V voedingen.

Een tweede variant van IEEE 802.3af biedt de voeding aan via dezelfde aders als al gebruikt wordt voor de netwerksignalen. Deze oplossing moet zowiezo gekozen worden voor de Gbit Ethernet-varianten, want daarbij zijn alle 8 pinnen op de RJ45-connector al in gebruik.

Om 'backwards compatible' te zijn met bestaande Ethernet apparatuur, zal de power hub alléén 48V aanbieden als een deelnemer dit ook vraagt. Hiervoor is een speciaal detectie-circuit aanwezig. Om schade te voorkomen aan apparatuur die toevallig wel elektronica op de ongebruikte pinnen heeft aangesloten moet de te voeden deelnemer stroom gaan afnemen volgens een bepaalde curve. De power hub ziet dit, en zal de spanning dan opvoeren. Bij een te hoog stroomverbruik wordt de spanning weer afgeschakeld.

Het voordeel van een power hub betreft dus voornamelijk de besparing op bekabeling voor voeding van apparatuur. Dit zal niet in alle toepassingen belangrijk zijn (bv. in kleinere machines), maar kan wel eens heel interessant zijn in de gebouwautomatisering. Het voordeel van het werken met een power hub is tevens dat er sprake is van een gecentraliseerde voeding. Dit maakt het ook makkelijker om back-up systemen (UPS) aan te sluiten, want dat hoeft nu maar op één locatie, en niet op alle netwerkdeelnemers.

Uiteraard werken power hubs enkel met UTP bekabeling, en niet met glasvezel. Toch zijn er ook hier ontwikkelingen gaande; het Amerikaanse bedrijf Photonic kan met fotocellen 6V / 10 mA genereren uit een glasvezelaansluiting. Deze ontwikkelingen lopen echter niet onder de vlag van de IEEE en het betreft dus geen aanstaande uitbreiding op Ethernet.

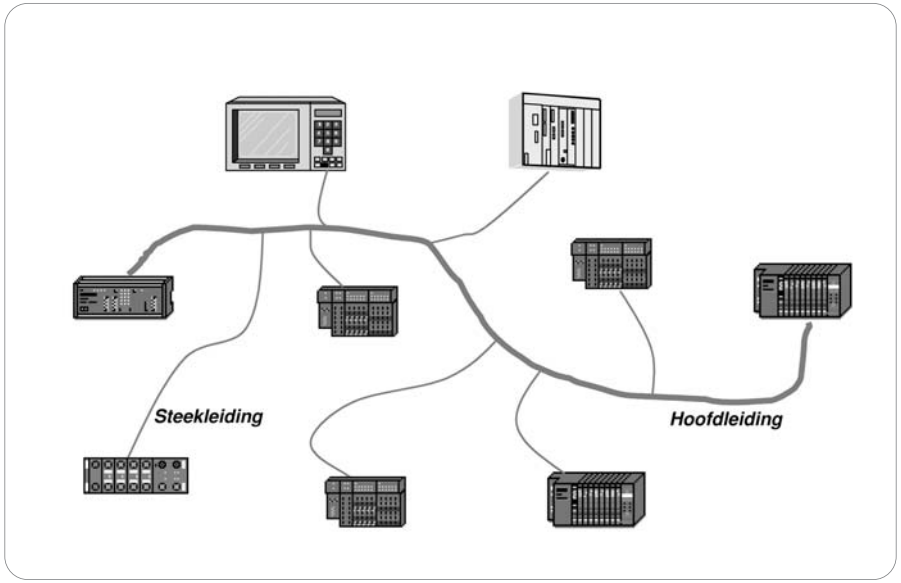
3.3 Bekabelingsstructuren

Bij Ethernet is het mogelijk om een netwerk op verschillende manieren te bekabelen: volgens de bus-, ster-, lijn- of ringstructuur. Elk heeft zijn eigen voor- en nadelen.

Busstructuur

Bij veel industriële netwerken wordt de bekabeling aangelegd volgens een busstructuur[1]. In principe is dit een lange kabel, waarop via aftakkingen alle deelnemers worden aangesloten. Het is niet toegestaan om aftakkingen te maken waarop meer dan een deelnemer is aangesloten. Het is een goedkope, eenvoudige en flexibele manier van bekabelen en daarom ook vrij populair (figuur 3-3).

[1] Let er op dat het woord "bus" twee betekenissen heeft: 1) een manier van bekabelen, 2) een netwerk. Er zijn bussystemen op de markt die als bus bekabeld moeten worden (bv. Profibus), maar er zijn ook bussystemen op de markt die niet als bus bekabeld moeten worden (zoals Interbus, dat als ring bekabeld wordt).



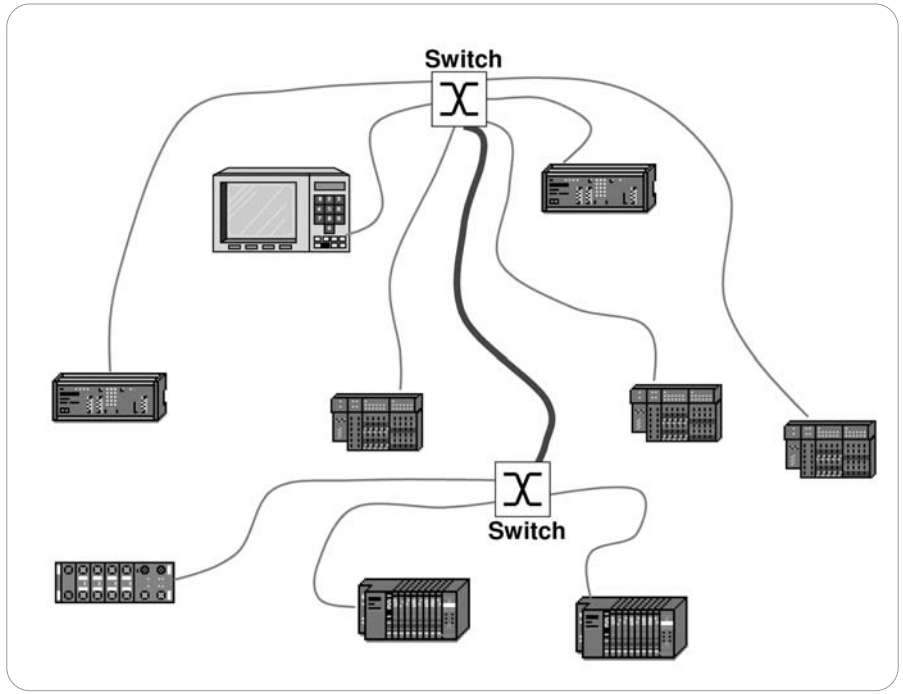
Figuur 3-3: Een "bus" bekabelingsstructuur. Hierbij is er één hoofdleiding, via steekleidingen kunnen deelnemers worden aangesloten. Tevens kunnen twee deelnemers aan het begin en eind van de hoofdleiding worden aangesloten.

Een bus-bekabeld netwerk bestaat altijd uit een hoofdleiding, welke als "rode draad" door het gehele systeem loopt. Op de hoofdleiding kunnen via steekleidingen deelnemers worden aangesloten. Per steekleiding mag steeds één deelnemer worden aangesloten. Het is dus onmogelijk dat via een steekleiding meerdere deelnemers worden aangesloten; aftakkingen kunnen dus nooit voorkomen. Op het begin en eind van de hoofdleiding kunnen ook twee deelnemers worden aangesloten. Tevens is het bij veel systemen verplicht dat op het begin en eind van de hoofdleiding met zgn. "terminators" of "afsluitweerstanden" wordt gewerkt. De maximale lengtes van de hoofleiding en de steekleiding zijn altijd begrensd; in principe geldt: hoe hoger de snelheid (bitrate/baudrate), des te korter de leidingen moeten zijn.

De ironie wil dat het bij Ethernet ook mogelijk is om op deze manier te bekabelen: de beide (dikke en dunne) coax-versies (10Base2 en 10Base5). Deze varianten worden echter nauwelijks nog gebruikt.

Sterstructuur

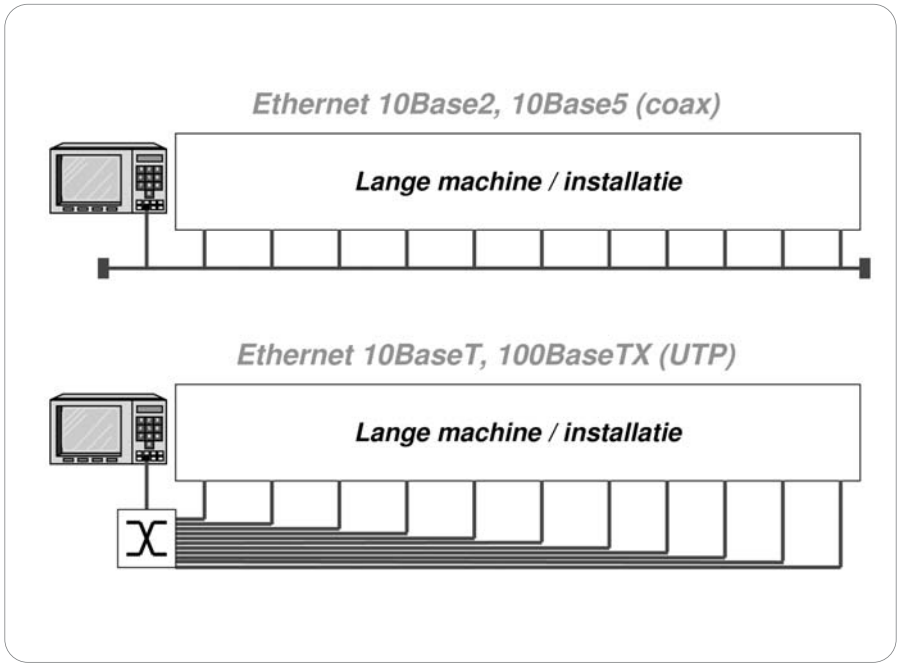
De twisted-pair varianten van Ethernet (10BaseT en 100BaseTX) staat echter niet toe dat volgens een busstructuur bekabeld wordt. Hier moet volgens een sterstructuur bekabeld worden (figuur 3-4). Dit wordt veroorzaakt door de hub / switch. Elke deelnemer wordt via zijn eigen kabel op de hub aangesloten. Het maximum aantal deelnemers wordt begrensd door het aantal beschikbare poorten, maar als dit te weinig mocht zijn kunnen hubs ook aan elkaar gekoppeld worden om zodoende grotere netwerken mogelijk te maken.



Figuur 3-4: Een stervormig bekabeld netwerk, zoals dat bij Ethernet 10BaseTX en 100BaseTX voorkomt. Hierbij zijn twee hubs ingezet, omdat er meer deelnemers zijn dan beschikbare poorten (8) per hub. De hubs zijn aan elkaar gekoppeld via een eigen kabel (midden-verticaal).

Een stervormige bekabelingsstructuur is niet zo vreselijk praktisch bij zeer lange machines of installaties. In vergelijking met een busstructuur is namelijk heel veel extra bekabeling nodig; en bij een kostprijs van ca. 1 Euro per meter kan dit toch nog behoorlijk aantikken.

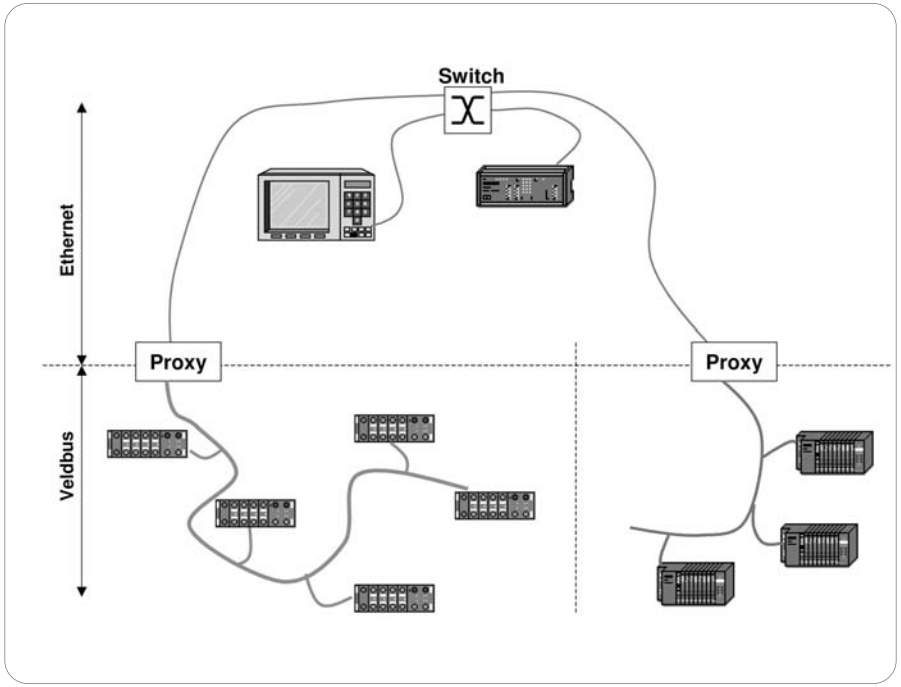
Verder is het lastig om een machine modulair op te bouwen, want er moet rekening gehouden worden met alle te trekken kabels. Een busstructuur is wat dat betreft veel makkelijker en flexibeler.



Figuur 3-5: Een stervormige manier van bekabelen is onpraktisch in elke lineaire structuur (onder). Bij de coax-varianten van Ethernet (boven) is het nog goed mogelijk, maar deze worden nauwelijks nog toegepast.

Combinatie tussen bus- en sterstructuur

Een tussenoplossing is ook nog mogelijk. Men krijgt dan een netwerk op twee niveaus: enerzijds een stervormig bekabeld Ethernet op het hoogste niveau, anderzijds een bestaand industrieel netwerk met een busstructuur (of iets anders) op het laagste niveau. Tussen beide niveaus wordt een "buskoppelaar", "proxy", "linking device" of "gateway" geplaatst; dit apparaat regelt de conversie tussen Ethernet en het andere netwerk (figuur 3-6).

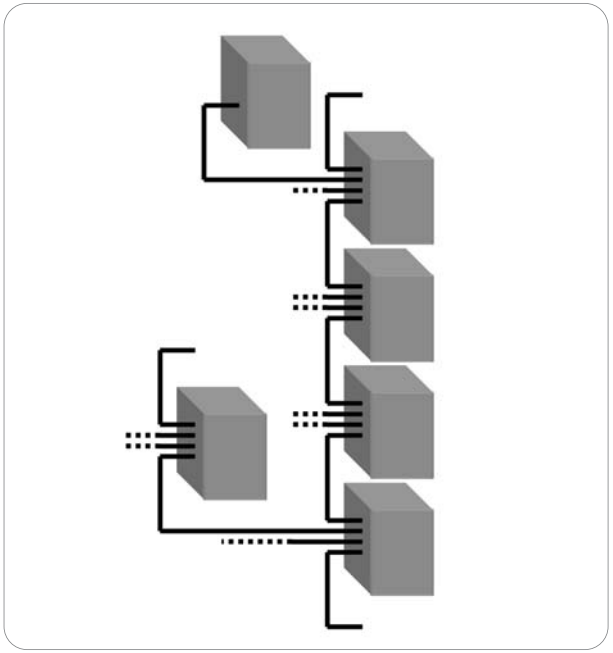


Figuur 3-6: Een netwerk dat uit 2 niveaus is opgebouwd: Ethernet op het hoogste niveau, en een bestaande veldbus op het laagste niveau, met een gateway / proxy / linking device / buskoppelaar als intermediair tussen beide types netwerk.

Indien meerdere hubs / switches gebruikt worden, kan een mix tussen een ring-, bus- en stertopologie opgebouwd worden. Dit wordt des te sterker naarmate er slechts enkele deelnemers op een hub aangesloten zijn. Niet toevallig bevatten industriële hubs meestal maar 4 of 8 poorten, véél minder dan de hubs die normaal in een kantooromgeving gebruikt worden, waarbij meestal geldt: hoe meer poorten des te beter (24, 32, 48, 64, etc.). Helaas zijn industriële hubs vrij prijzig, en als dan ook nog steeds 2 poorten nodig zijn om de buur-hubs linksom en rechtsom te koppelen, dan is het een vrij dure oplossing.

Lijnstructuur

Bij ProfiNet (zie pagina 50) is een andere oplossing in de maak, waarbij elke deelnemer op het netwerk wordt voorzien van een eigen, interne switch (figuur 3-7).



Figuur 3-7: In ProfiNet versie 3 krijgt elke deelnemer een eigen, interne switch met 4 poorten. Daarmee wordt een andere manier van bekabelen mogelijk: busvormig in plaats van stervormig.

In de ProfiNet controller chip is namelijk een 4-poorts switch ingebouwd. Hiermee kan een lijnvormig netwerk worden opgebouwd, door de switches aan elkaar te koppelen; twee poorten blijven nog beschikbaar voor het maken van aftakkingen. Elk willekeurig netwerk kan op deze manier opgebouwd worden zolang een telegram niet meer dan 20 switches hoeft te passeren (dit i.v.m. de optelsom van de interne verwerkingstijden van elke switch). Omdat elk ProfiNet apparaat straks een eigen switch heeft, hoeft men deze dus ook niet meer aan te schaffen; tevens is de bekabeling minder. Een extra voordeel van een 'eigen' switch is dat deze het eigen protocol kan beoordelen boven de andere protocollen die parallel op Ethernet uitgevoerd worden. Dit verbeterd de real-time eigenschappen van het eigen (ProfiNet) protocol. Let wel op de specifieke nadelen van de nieuwe manier van bekabelen: indien een deelnemer uit het netwerk wordt gehaald of uit wordt gezet, zal het netwerk onderbroken worden en in 2 onafhankelijke helften opgedeeld raken. Dit is bij de stervormige manier van bekabelen niet zo.

Deze uitvinding van Profibus zal ongetwijfeld gekopieerd gaan worden door andere bussystemen. Het is een zeer interessante manier van bekabelen voor Ethernet, die het grote nadeel van de stertopologie opheft, en het misschien mogelijk maakt dat Ethernet beter concurrerend wordt met bestaande remote I/O bussystemen.

Ringstructuur

Ringvormige bekabelingsstructuren zijn ook mogelijk met Ethernet. Deze zijn te realiseren met switches, maar daarbij moet wel rekening gehouden worden dat 'eindeloze lussen' niet voor mogen komen. Speciale netwerkprotocollen zoals STP (Spanning Tree Protocol) zijn ontwikkeld om dit te detecteren, en via een softwarematige 'knip' in de lus wordt deze onderbroken. Het voordeel van een ring is dat er redundantie aanwezig is: een kabelbreuk, of uitval van een switch, kan opgevangen worden. De omschakeltijd is afhankelijk van de omvang van het netwerk, en uiteraard ook van het gebruikte protocol (meer informatie hierover in hoofdstuk 5).

3.4 Is Ethernet nu real-time of niet?

Over het feit of Ethernet nu "real-time" is of niet wordt al tientallen jaren gediscussieerd, en het antwoord op deze vraag kan zowel "ja" als "nee" zijn. Dit komt omdat een vraag "Is Ethernet geschikt voor real-time toepassingen?" lezen als "Is Ethernet snel?", terwijl anderen het lezen als "Is Ethernet deterministisch?".

Of Ethernet "snel" is of niet, hangt uiteraard af van de eisen die de applicatie stelt. Net zoals bij alle andere netwerken zijn er systemen te bedenken waarbij Ethernet ruimschoots snel genoeg is, maar er zijn ook systemen te bedenken waarbij Ethernet véél te langzaam is. Om de discussies hierover meer te standaardiseren, heeft de IAONA werkgroep "Real-time systemen" vier verschillende soorten real-time systemen onderscheiden (tabel 3-1).

Klasse	1	2	3	4
Max. jitter	> 1 ms	0,1 .. 3 ms	10 .. 400 µs	0,5 .. 15 µs
Hardware	Standaard	Standaard	Standaard	Speciaal
SW Stack	Standaard	Aangepast	Speciaal	Speciaal
Protocol	Standaard	Standaard	Speciaal	Speciaal

Tabel 3-1: de IAONA real-time klassen

Klasse 1 systemen zijn te realiseren met standaard Ethernet hardware, standaard software (zonder verder enige speciale voorzieningen), en met bestaande protocollen.
Klasse 2 systemen maken al wel gebruik van speciale software, bijvoorbeeld een real-time kernel of een real-time extensie voor een bestaand operating systeem (o.a. voor Windows).
Klasse 3 systemen maken nog steeds gebruik van standaard Ethernet hardware, maar hebben wel speciale netwerkprotocollen en de daarbij behorende software nodig (diverse hiervan zijn in hoofdstuk 6 beschreven). Klasse 4 systemen, die de zwaarste eisen stellen, hebben tenslotte ook nog speciale hardware nodig in de vorm van switches (die b.v. IEEE 802.1P ondersteunen), speciale randapparatuur (zoals voor IEEE 1588), of speciale Ethernet interfaces (zoals de in elk apparaat ingebouwde switches voor ProfiNet V3).

10 of 100 Mbit/s ?
De snelheid van Ethernet is nog te kiezen: 10 of 100 Mbit/s? Tien megabit lijkt snel genoeg, zeker in vergelijking met veel moderne industriële netwerken die opereren op veel lagere bitrates (zoals CAN op maximaal 1 Mbit/s, Interbus op 500 Kbit/s, etc.). Maar het is slechts een bruto-snelheid, waarbij geen rekening gehouden is met de overhead per netwerkbericht. Als men daarmee wel rekening houdt, dan blijkt dat Ethernet slecht presteert bij transport van kleine hoeveelheden data. Bestaande bussystemen zoals CAN en Profibus/DP kunnen op veel lagere snelheden de concurrentie met Ethernet goed aan (zie de snelheidsvergelijkingen in hoofdstuk 4).

Indien op Ethernet een master/slave protocol gebruikt wordt, kan de totale netto snelheid nog lager worden bij gebruik van een switch. In tegenstelling tot een hub slaat een switch alle ontvangen netwerkberichten eerst intern op, en stuurt ze daarna door. In zulke gevallen moet dan rekening gehouden worden met een dubbele transmissietijd.

Een geheel ander aspect bij Ethernet is de invloed van software. De in hoofdstuk 4 getoonde berekeningen gaan uit van 0% software-overhead. Bij bestaande bussystemen zoals bijvoorbeeld Interbus, CAN en Profibus/DP is dit mogelijk, want deze protocollen kunnen geheel "in hardware" uitgevoerd worden (met speciale chips). Bij Ethernet is dit (nog) niet mogelijk, er is altijd een processor nodig die de hogere protocollagen (OSI-laag 3 t/m 7) uitvoert. De invloed van de software-overhead kan zodanig zijn dat een bruto-snelheidsverhoging van een 10 naar 100 Mbit/s maar een kleine netto snelheidsverhoging oplevert.

3.5 Is Ethernet nu deterministisch of niet?

Het begrip "determinisme" is, net zoals "real-time", een weinig begrepen kreet. De uitspraak van veel leveranciers is "Ethernet is niet deterministisch", en zonder verdere kennis over wat determinisme is, lijkt het allemaal heel erg slecht, puur vanwege de implicatie dat als er iets ontbreekt, dat Ethernet dan 'dus' niet goed is voor industriële toepassingen.

Indien een netwerk deterministisch is, dan wil dat niet meer zeggen dat het gedrag van het netwerk geheel voorspelbaar is, en dat dus ook bekend is hoeveel tijd het maximaal kost om een bepaald netwerkbericht te versturen. Bij Ethernet is dit niet enkel afhankelijk van de omvang van het netwerkbericht zelf, maar ook van de netwerkbelasting (= wat doen de andere deelnemers op dat moment) en of er een hub dan wel een switch gebruikt wordt.

Indien men met een hub werkt, dan kan het voorkomen dat twee (of meer) deelnemers tegelijkertijd een netwerkbericht willen sturen. Uiteraard kan dit niet, maar deelnemers weten van elkaar niet wat de ander doet. Er ontstaat dan een zgn. "collision" omdat er twee transmissies tegelijk actief zijn (te detecteren dankzij de aanwezigheid van een ongeldig spanningsniveau). Beide deelnemers moeten stoppen met hun transmissie, een random tijd wachten, en het daarna nog eens opnieuw proberen. Als er opnieuw een collision ontstaat herhaalt het geheel zich nog eens, net zo lang totdat beide partijen hun netwerkbericht verstuurd hebben. In het zeldzame geval dat er meer dan 16 collisions achter elkaar optreden zal Ethernet de transmissie van een netwerkbericht afbreken. Het is dan aan het netwerkprotocol op een hoger niveau (meestal in OSI-laag 4, b.v. dus TCP) om de fout te herstellen.

De random tijd die gewacht moet worden, wordt gekozen uit een interval dat na elke collision steeds 2x zo groot wordt: 1, 3, 7, 15, etc. Het maximum is echter vastgesteld op 1023. De kans dat twee deelnemers exact dezelfde random waarde uit het interval kiezen wordt dus steeds 2x zo klein. Het zal duidelijk zijn dat de kans dat twee apparaten precies dezelfde tijd wachten steeds kleiner wordt, en dus zal één van de twee zijn bericht eerder kunnen sturen dan de ander. Na de eerste collision is er 50% kans dat het daarna goed gaat, en mocht het fout gaan dan is er de volgende keer 75% kans dat het goed gaat, en mocht het voor de derde keer fout gaan dan is de kans dat het de vierde keer goed gaat 87.5%, etc.

Collision	Interval-grenzen	Max. wachttijd op 10 Mbit/s (msec)	Max. cumulatieve wachttijd op 10 Mbit/s (msec)	Max. cumulatieve wachttijd op 100 Mbit/s (msec)
1	0..1	0,0512	0,0512	0,0051
2	0..3	0,1536	0,2048	0,0204
3	0..7	0,3584	0,5632	0,056
4	0..15	0,7680	1,331	0,133
5	0..31	1,587	2,918	0,291
6	0..63	3,226	6,144	0,614
7	0..127	6,502	12,64	1,264
8	0..255	13,05	25,70	2,570
9	0..511	26,16	51,86	5,186
10	0..1023	52,38	104,2	10,42
11	0..1023	52,38	156,6	15,66
12	0..1023	52,38	209,0	20,90
13	0..1023	52,38	261,4	26,14
14	0..1023	52,38	313,8	31,38
15	0..1023	52,38	366,1	36,61
16	0..1023	52,38	418,5	41,85

Tabel 3-2: Maximale wachttijd per collision, en de cumulatieve wachttijd na 'n' collisions.

Tabel 3-2 geeft de maximale tijdsduur van het interval na elke collision. Zoals te zien is er op een 10 Mbit/s netwerk een kans dat men na een collision zeer lang moet wachten. Bijvoorbeeld, na 10 collisions is er een kans van 1/1023 dat men 52,38 msec moet wachten. Gemiddeld gezien zal de wachttijd slechts de helft hiervan bedragen. Echter voordat men zover is kan al 104,2 msec verstreken zijn. De kans op steeds de maximale wachttijd bij 10 collisions is wel zeer klein: 1/2 * 1/4 * 1/8 * 1/16 * 1/32 * 1/64 * 1/128 * 1/256 * 1/512 = 1/245 oftewel 1 op de 35 biljoen. Maar voor gebruik in industriële applicaties wordt toch vaak met dit soort worst-case kansberekeningen omgegaan.

In de meeste gevallen zal de wachttijd niet zo lang zijn, maar enige garantie krijgt men niet. In netwerkkringen wordt dan gezegd dat Ethernet daarom niet "deterministisch" is: het is niet voorspelbaar, meestal zal het snel genoeg zijn, maar héél soms toch niet. Op 100 Mbit/s gaat het afhandelen van collisions uiteraard wel 10x zo snel, maar desalniettemin blijft er toch nog eens kans op een maximale wachttijd van 41,8 msec. Voor industrieel gebruik is dit een "eeuwigheid", zeker voor machinebouwers.

Het feit dat een hub-gebaseerd Ethernet soms héél veel tijd nodig kan hebben om een netwerkbericht te sturen is niet te betwisten. De vraag is echter, hoe vaak zal men in zo'n situatie verzeild raken? Het is net als met auto's op snelwegen; telkens als men over het asfalt zoeft is er een bepaalde kans dat men betrokken raakt bij een ongeluk en overlijdt. Toch maken wij ons daar geen zorgen om, omdat de kans dat zoiets gebeurt heel erg klein is. Een gelijksoortige houding kan men ook aannemen bij Ethernet. OK, het is dan misschien wel niet 100% deterministisch, maar is het *in mijn situatie* deterministisch genoeg? Is het een probleem als er gemiddeld slechts één keer per 1000 jaar de aflevering van een netwerkbericht vertraagd raakt? Of eens per 10000 jaar?

Netwerk snelheid (Mbit/s)	Bericht-grootte (bytes)	Berichten per sec.	Deadline (max ms)	Levensduur (jaar)	Netwerk-belasting	Kans op succes
100	128	1000	3	5	1%	1
100	128	1000	2	5	1%	0.99999983
100	128	1000	1	5	1%	0.99995591
100	128	1000	0.5	5	1%	0.99432787
100	1024	1000	2	5	8%	0.99991674
100	1024	1000	1	5	8%	0.97882959
100	1024	2000	1.5	5	16%	0.99874754
100	1518	2000	2	5	24%	0.99615729
100	1518	5000	2	5	61%	0.89125139
10	128	100	2	5	1%	0.96370139
10	128	250	2	5	3%	0.23984533
10	128	250	4	5	3%	0.97827393
10	128	500	4	5	5%	0.70837748
10	128	500	8	5	5%	0.99733087
10	128	1000	8	5	10%	0.95967719
10	1024	100	8	5	8%	0.99786252
10	1024	250	8	5	20%	0.92655956
10	1024	500	8	5	41%	0.35021843

Tabel 3-3: De spreadsheet van RTI geeft voor een aantal scenario's direct aan wat de kans is dat nooit een bepaalde deadline overschreden zal worden, gegeven de belasting van het netwerk, de deadline, en het aantal jaar dat een applicatie succesvol moet kunnen blijven werken.

Het Amerikaanse bedrijf Real Time Innovations (*www.rti.com*) heeft een spreadsheet (tabel 3-3) ontwikkeld waarmee men de kans kan berekenen dat, gegeven een bepaalde netwerkbelasting, een te grote vertraging ontstaat bij het verzenden van een netwerkbericht. Men kan zelf de deadline invoeren, en de spreadsheet rekent dan uit hoe groot de kans is dat men te laat reageert. Een voorbeeld: gegeven een netwerk van 10 Mbit/s, waarop 500x per seconde 1K data wordt verstuurd. Deze applicatie heeft een deadline van 8 msec, welke gedurende 5 jaar niet overschreden mag worden. Wat is de kans dat het goed gaat? De spreadsheet geeft aan dat de kans slechts ca. 35% is. Dit zal niet acceptabel zijn. Wat kan dan nog gedaan worden om de slaagkans te vergroten? Een mogelijke optimalisatie is om het applicatieprogramma aan te passen zodat slechts 250 berichten per seconde hoeven te worden verstuurd. De slaagkans wordt dan 92,6%. Dat is al een stuk beter; het aardige van deze spreadsheet is dat men op deze manier snel een aantal scenario's door kan rekenen.

Ook in het boek van Fuller (zie hoofdstuk 9) wordt uitgebreid aandacht besteed aan dit aspect.

Voorkomen van collisions

Alle hierboven getoonde berekeningen gaan uit van een manier van netwerkgebruik waarbij iedereen op elk willekeurig moment een netwerkbericht zou kunnen sturen. Dit is inderdaad het geval op veel kantoornetwerken, waar achter elke PC een gebruiker zijn eigen werk zit uit te voeren onafhankelijk van wat de andere collega's doen. Een te groot aantal collisions, veroorzaakt door een te hoge belasting, gaat dan op een bepaald moment contraproductief werken, omdat Ethernet alleen nog maar bezig is met het afhandelen van collisions en er nauwelijks nog een netwerkbericht normaal verstuurd en afgeleverd wordt. Over waar dit kantelpunt precies ligt, circuleren veel sprookjes: volgens de ene expert begint Ethernet al bij 8% belasting overbelast te raken, de ander legt het kantelpunt ergens bij de 50%, en de volgende expert claimt dat Ethernet tot meer dan 90% te belasten is (men leze hoofdstuk 19 van het boek van Spurgeon).

Onafhankelijk van wie er nu gelijk heeft, zijn er nog andere methodes om geen last te hebben van (teveel) collisions. Dit kan zowel geregeld worden met speciale hardware, als met speciaal ontworpen netwerkprotocollen:

- Gebruik geen hubs maar switches. Elke deelnemer heeft zijn eigen transmissiekanaal naar de switch. Omdat er per poort altijd maar één deelnemer is aangesloten, kunnen er nooit collisions ontstaan.
- Hou de belasting van het netwerk zodanig laag dat de kans zeer klein is dat

  er ooit een collision optreed. Mocht dit toch nog eens voorkomen, dan is de kans op nog een collision zeer klein. De totale vertraging blijft dan beperkt. Dit is de aanpak die Hima volgt in het "SafeEthernet" protocol.
- Maak gebruik van een master/slave protocol. Hierbij is óf de master, óf een van de slaves bezig met een transmissie. Omdat er dus altijd maar één deelnemer tegelijk bezig is, kunnen er in het geheel geen collisions optreden.
- Maak gebruik van een token-bus protocol. Slechts één van de deelnemers heeft het token in bezit, en mag daarom een transmissie uitvoeren. Omdat er dus altijd maar één deelnemer tegelijk bezig is, kunnen er in geen collisions optreden.
- Maak gebruik van een time-triggered protocol. Elke deelnemer krijgt een eigen tijdslot waarin hij zijn transmissies uit kan voeren. Dit tijdslot komt regelmatig terug. Omdat er dus altijd maar één deelnemer tegelijk actief kan zijn, kunnen er geen collisions optreden.

Uiteraard is het ook mogelijk een combinatie van deze maatregelen te gebruiken. In de praktijk wordt meestal een switch gebruikt, samen met een van de genoemde protocolvarianten. Dit zien we dan ook precies zo terugkomen in moderne netwerkprotocollen (hoofdstuk 6).

Let er op dat daar waar in de bovenstaande opsomming gesproken wordt over protocollen, er geen gebruik gemaakt kan worden van TCP/IP. Ook al maakt men een master/slave protocol op basis van TCP/IP, dan nog is er een zekere kans op collisions. Dit komt omdat TCP/IP ook autonoom gedrag heeft waarbij het zelf netwerkberichten verstuurt.

3.6 De connector

UTP-gebaseerde Ethernetten maken gebruik van de "RJ45" connector (Registered Jack). Deze voldoet uitstekend voor gebruik op kantoor en thuis, maar voor industrieel gebruik kleven er toch enkele nadelen aan deze connector:

- Geen trekontlasting;
- Plastic lipje breekt makkelijk af;
- Connector kan bewegen in de plug;
- Niet trillingsbestendig;
- Niet stof- en spatwaterdicht;
- Niet bestand tegen bepaalde chemicaliën;
- Etc.

De trillingsbestendigheid is uitgebreid door Rockwell onderzocht. Het blijkt dat door de voortdurende bewegingen van de veertjes in de RJ45 connector na enige tijd de goudlaag afschuurt. Het onderliggende metaal komt dan bloot en kan gaan oxideren, wat de kwaliteit van de signaaloverdracht negatief beïnvloedt.

Gegeven al deze nadelen is de gedachte ontstaan dat de RJ45 niet bruikbaar zou zijn voor industrieel Ethernet, en dat er een alternatief zou moeten komen. Dat is ook gebeurd, en wel in een zodanig mate dat er nu ca. 8 verschillende connectoren op de markt gebracht zijn. Deze zijn als volgt te groeperen:

- Inzet van al bestaande connector (Sub-D, M12);
- Geheel nieuw type connector;
- RJ45-compatibele connector.

Deze connectoren zijn echter niet zo goedkoop als de RJ45. Inmiddels is dan ook een tegenbeweging op gang gekomen: waarom moet *elke* industriële applicatie van een speciale, dure connector gebruik maken? In de meeste gevallen voldoet de gewone RJ45 uitstekend, dat is de afgelopen 10 jaar wel bewezen. We zien dan ook dat het gebruik van RJ45 mogelijk blijft. In figuur 3-8 zijn twee voorbeelden gegeven van industriële connectoren.

Figuur 3-8: Twee voorbeelden van industriële connectoren.

Bestaande types connectoren

Het ontwikkelen van een connector is geen eenvoudige materie, en daarom is het beter om soms gebruik te maken van connectoren die zich al hebben bewezen in andere toepassingsgebieden. Twee voorbeelden hiervan zijn de 9-pins sub-D connector, en de M12 connector. Deze laatste lijkt een defacto standaard te gaan worden.

RJ45 compatible connectoren

In deze groep connectoren wordt steeds uitgegaan van de standaard RJ45, meestal uitgebreid met een steviger behuizing, trekontlasting, IP65 beveiliging, etc. Daarnaast zijn er connectoren ontwikkeld welke extra pennen hebben (naast de RJ45) welke gebruikt worden voor het aansluiten van de voeding. Dit is dus een afwijkend concept van wat elders in Ethernet gebruikelijk is (zie paragraaf 3.2), maar het voordeel is dat er grotere stromen geleverd kunnen worden.

Het gebruik van de RJ45-compatibele connectoren heeft voor de gebruiker het voordeel dat hij nog steeds een standaard RJ45 plug kan gebruiken. Dit maakt het aansluiten vrij makkelijk als men geen speciale eisen aan de connector stelt. Voor de leverancier heeft dit het voordeel dat men met één connector twee categorieën klanten tevreden kan stellen, en dus ook niet twee varianten van hetzelfde type apparaat op de markt hoeft te brengen.

Toekomstverwachtingen

De grote diversiteit aan connectoren voor industrieel Ethernet is uiteraard vervelend, en een typisch voorbeeld van de verdeeldheid in de industriële automatisering. Toch zijn er tot nu toe geen klachten over te horen. Dit komt omdat gebruikers nog niet geconfronteerd zijn met de verscheidenheid aan connectoren; deze zijn al wel enkele jaren op de markt, maar nog niet ingebouwd in apparatuur. Dit begint nu langzaam te komen, en dan zullen er meer klachten komen.

Uiteraard beseffen de leveranciers van de speciale connectoren zich dit ook. Allen claimen goede contacten te hebben met de IEEE, en samen te werken met de IAONA-vereniging. Uiteindelijk zal de markt toch moeten beslissen welke connectoren overblijven en welke weer zullen verdwijnen. Tot die tijd zullen verloopstukjes e.d. nodig blijven.

3.7 Gebruik van speciale protocollen

In de administratieve automatisering wordt zeer veel gebruik gemaakt van de uitgebreide TCP/IP protocolfamilie. TCP (Transmission Control Protocol) en IP (Internet Protocol), eventueel nog met UDP (User Datagram Protocol) zijn dan de fundamenten van een zeer uitgebreide set toepassingen:

TELNET	Remote Terminal
FTP	File Transfer Protocol
NFS	Networked File System (disk)
HTTP	Hypertext Transport Protocol (html)
POP	Post Office Protocol (email)
SNMP	Simple Network Management Protocol
BOOTP	Bootstrap Protocol
NTP	Network Time Protocol
RTP	Real-Time Protocol
DHCP	Dynamic Host Configuration Protocol
etc.	

De ervaren Internetter zal deze protocollen herkennen als bouwstenen die op Windows en Linux aanwezig zijn (vooral als ze niet werken!). Het zijn echter geen van allen protocollen die gebruikt worden om de communicatie tussen besturingen, en van en naar I/O, uit te voeren. Toegegeven, een besturing kán wel een email sturen naar een andere besturing met daarin het verzoek om een bepaalde actie uit te voeren, maar dit is een onpraktische (en langzame) manier van werken.
Helaas ontbreekt het aan een standaard protocol voor industrieel gebruik op basis van TCP/IP. Nu bestaan zulke protocollen wel, want veel leveranciers hebben deze zelf ontwikkeld voor gebruik op eigen apparatuur en producten. Al deze eigen protocollen zijn echter incompatible met de rest van de wereld. Dit zorgde ervoor dat TCP/IP en Ethernet al wel veel gebruikt werd in de industrie, maar het waren geen "open" protocollen, en de gebruikers waren derhalve gebonden aan één leverancier.

Actuele ontwikkelingen rondom TCP/IP

Veel lopende ontwikkelingen op het gebied van industrieel Ethernet maken ook gebruik van TCP/IP. Het betreft echter altijd die protocolonderdelen waarbij snelheid en real-time gedrag niet van belang zijn, zoals bij initialisatie, netwerkbeheer, configuratie en diagnostiek. Voor snelle remote I/O is TCP/IP niet bruikbaar; de bestaande bussystemen

kunnen hier de concurrentie uitstekend aan. Dit komt dankzij een aspect van TCP/IP dat uitstekend functioneert in een kantooromgeving, maar minder in een industriële omgeving: het transporteren van kleine hoeveelheden data (meestal enkele bytes) is uit TCP/IP weggeoptimaliseerd, ten gunste van een optimaal transport van grote(re) hoeveelheden data. Dat past dus niet goed bij remote I/O. Daarom worden nieuwe protocollen ontwikkeld, die rechtstreeks gebruik maken van Ethernet.

Op zich is het 'verlies' van TCP/IP niet zo'n probleem; niemand gaat immers remote I/O uitvoeren over een intranet of het internet. Een tussenoplossing is het gebruik van het "UDP" protocol (User Datagram Protocol). Dit is een zeer simpel en eenvoudig protocol, dat eigenlijk niet veel meer is dan een heel dunne schil om IP. Echter, omdat UDP zoveel simpeler is, is het ook veel sneller dan TCP. Daarnaast heeft het nog twee belangrijke voordelen: UDP is berichtgeoriënteerd, en men kan gebruik maken van "broadcasts": één netwerkbericht tegelijk naar alle deelnemers op een netwerk sturen. Nadelen zijn o.a. dat men geen garantie heeft op correcte aflevering op de eindbestemming, en dat berichten in een andere volgorde kunnen arriveren dan ze verzonden zijn. Tenslotte heeft het werken met broadcasts in combinatie met switches niet altijd zin.

Maar ook UDP wordt steeds vaker beschouwd als "te langzaam". Zo worden in IDA, ProfiNet en Powerlink speciale protocollen voor afhandeling van I/O gemaakt die rechtstreeks op Ethernet werken, en in feite dus parallel aan TCP/IP en UDP werken.
Een belangrijke snelheidswinst is het gevolg, en tevens wordt het mogelijk om zeer nauwkeurig te synchroniseren. Dit gaat dan wel weer ten koste van de flexibiliteit die dankzij TCP/IP verkregen werd en kan potentiële problemen opleveren als op hetzelfde netwerk nog andere protocollen gebruikt worden. Tevens is het niet altijd meer mogelijk om van standaard randapparatuur uit te gaan (switches, routers, netwerkanalyzers, etc.); koppelingen naar intranetten en/of Internet worden moeilijker, en softwarepakketten voor netwerkbeheer kunnen niet met de speciale protocollen omgaan.

Tunnelling

Een geheel andere ontwikkeling is de mogelijkheid om TCP/IP berichten over een (bestaande) veldbus te sturen. Dit wordt ook wel 'tunnelling' genoemd, en heeft eigenlijk helemaal niets met Ethernet te maken; het is in feite een koppeling van een bestaand veldbussysteem met een extra schil software voor de afhandeling van TCP/IP berichten. Op zich is het idee heel eenvoudig: een TCP/IP bericht bestaat uit een aantal bytes, en als de veldbus deze bytes ongewijzigd kan transporteren, dan is de klus al geklaard. De moeilijkheid zit meestal in de beperkte maximale omvang van netwerkberichten op een veldbussysteem

(enkele tientallen bytes, of nog minder, soms enkele honderden bytes). De oplossing is dan om grote TCP/IP berichten in stukken op te splitsen, deze een voor een te verzenden, en aan de ontvangende kant alle stukken weer samen te voegen tot het originele TCP/IP bericht. Daarmee is het onzichtbaar geworden dat er ooit een veldbus nodig is geweest om dat bericht te transporteren.

Wat er daarna met het TCP/IP bericht gebeurt, is afhankelijk van het applicatieprotocol. Zoals hierboven genoemd zijn er een groot aantal mogelijkheden (applicatieprotocollen). Eén interessante is het inbouwen van een webserver in een field device; van buiten af kunnen dan webpagina's opgevraagd worden. Een andere is de mogelijkheid dat een field device zelf een email stuurt, aangestuurd door de eigen diagnostische functies. Alhoewel sommige bedrijven deze mogelijkheden al bieden, is niet iedereen er van overtuigd dat dit allemaal zinvol is. Uiteraard komt het uitvoeren van de eigen functies van een apparaat op de eerste plaats. Meestal is daarvoor een zo simpel mogelijke processor (CPU) ingebouwd die, in tegenstelling tot kantoor-PC's met hun GigaHertz Pentium's, best goed op een klok-snelheid van 30 MHz (soms nog minder) kunnen lopen. Het uitrekenen en opsturen van webpagina's kost dan veel tijd, en dat mag uiteraard niet ten koste gaan van de I/O, de regellus, etc.

3.8 Prijs

Dankzij de enorme populariteit is Ethernet-hardware zeer goedkoop. Insteekkaarten voor een PC hoeven niet meer te kosten dan een paar Euro. En ook deze zijn steeds minder nodig nu Ethernet-interfaces standaard in elke PC ingebouwd zijn.

Hier kan geen industrieel netwerk aan tippen. De prijsstelling wordt daarom wel eens gebruikt als voordeel van "industrieel Ethernet" boven de bekende industriële netwerken. Maar dit is niet helemaal terecht; de uitgave van de paar Euro voor een Ethernet-kaartje zijn slechts een deel van de kosten die men maakt om een netwerk te kunnen bouwen. Het netwerkkaartje is alleen maar een elektrische interface, en doet verder niets. Het echte werk gebeurt op de PC zelf. Om die te laten werken heeft men op voorhand al geld moeten uitgegeven voor het volgende:

- Ondersteuning op het moederbord van de PC;
- Gebruik van de powersupply van de PC;
- Ontwikkeling van de driver door de leverancier;
- Betaling van de ontwikkeling van de netwerkprotocollen;

- Ontwikkeling van de netwerksoftware door Microsoft;
- Opslagruimte op de harddisk voor deze software;
- RAM geheugen nodig voor de uitvoering van deze software;
- Gebruik van de capaciteit van de Pentium-CPU voor uitvoering van de netwerkprotocollen;

Desondanks blijven de totale kosten van het gebruik van Ethernet in een PC, indien we niet verder kijken dan één enkele module, altijd lager[2] dan die van een willekeurige veldbus, waarvan PC-insteekkaarten toch al gauw *minimaal* 100 Euro kosten, nog afgezien van softwarelicenties.
Het implementeren van TCP/IP in een niet-PC zal aanzienlijk duurder zijn dan in Windows, omdat er geen verborgen subsidies zijn en derhalve de volle kosten terugverdiend zullen moeten worden. Dit ziet men uiteraard terug in de kostprijs van de apparatuur.

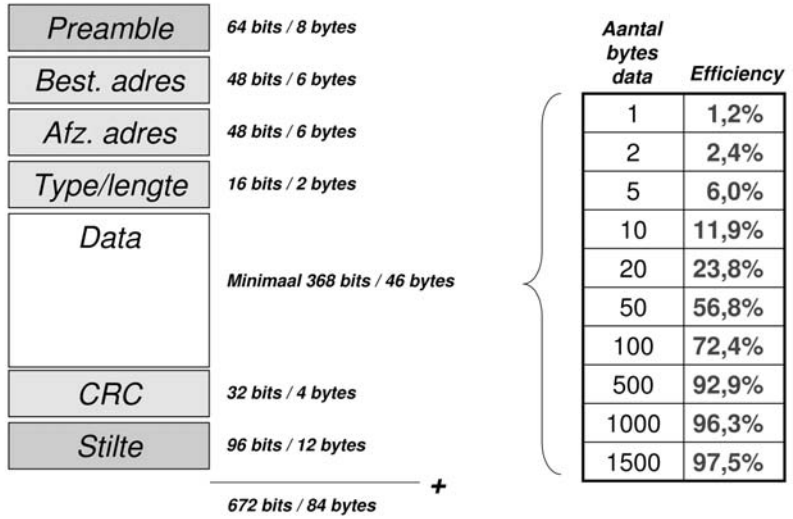
[2] De Ethernet-interfacekaart in de PC van de auteur kostte 6 Euro.

4. Snelheid

Net zoals bij de meeste industriële netwerken is snelheid ook bij Ethernet belangrijk. In eerste instantie lijkt een snelheid van 10 of 100 Mbit/s zeer hoog, zeker in vergelijking met industriële netwerken die veelal op snelheden van enkele Mbit/s werken. Maar de bitrate is eigenlijk slechts een 'bruto' snelheid, terwijl de 'netto' snelheid bepaald wordt door de overhead van Ethernet in combinatie met het gebruikte netwerkprotocol.

4.1 Overhead per netwerkbericht

De overhead van Ethernet is niet gering: elk netwerkbericht heeft een minimumomvang van 84 bytes, waarin ruimte is voor 46 bytes applicatiedata. Zeker bij transmissie van kleine hoeveelheden data is Ethernet zeer inefficiënt. Maar dit is een situatie die juist voorkomt bij aansturing van remote I/O; één analoog kanaal is meestal 2 bytes groot, en 24 digitale I/O kanalen hebben ook maar 3 bytes data nodig. Figuur 4-1 geeft een overzicht van de efficiency van Ethernet bij verschillende hoeveelheden data. Let er op dat bij grote hoeveelheden data Ethernet juist zéér efficiënt is.



Figuur 4-1: De opbouw van elk Ethernet bericht (links) maakt dat transmissie van kleine hoeveelheden data (iets dat vaak voorkomt in industriële toepassingen!) vrij inefficiënt is.

4.2 Snelheidsvergelijking met CAN

Indien we Ethernet nu vergelijken met CAN (Controller Area Network), een bekend industrieel netwerk voor remote I/O en real-time applicaties, dan blijkt dat beide netwerken goed aan elkaar gewaagd zijn, maar wel in andere toepassingsgebieden (tabel 4-1).

Een eerste vergelijking betreft de transmissie van 2 bytes data. Elk CAN-netwerkbericht heeft een overhead van 47 bits, en de totale hoeveelheid verstuurde data is dan 47+2*8 = 63 bits. Op een snelheid van 1 Mbit/s kost dit dus 63 µsec. Dezelfde 2 bytes op Ethernet kost in totaal 672 bits transmissietijd, en dit kost (op 10 Mbit/s) 67 µsec. CAN is dus toch sneller dan Ethernet. Hoe komt dit nu? CAN heeft een hogere efficiency (ca. 25%) dan Ethernet (ca. 2,3%), en daardoor wordt de 10x lagere bitrate van CAN ruimschoots gecompenseerd.

Aspect	CAN	Ethernet
Snelheid	1 Mbit/s	10 Mbit/s
Max. data	8 bytes	1500 bytes
Min. data	0 bytes	46 bytes
Overhead	47 bits	38 bytes
2 bytes data sturen	1 bericht nodig	1 bericht nodig
Aantal bits transmissie	47 + 2*8 = 63	(46+38)*8 = 672
Transmissietijd	63 µsec	67 µsec
Efficiency	(2*8)/63 = 25,3%	(2*8)/672 = 2,3%
1500 bytes sturen	188 berichten nodig (waarvan de laatste met 4 bytes data)	1 bericht nodig
Aantal bits transmissie	187*(8*8+47)+(4*8+47) = 20836	(1500+38)*8 = 12304
Transmissietijd	20,8 msec	12,3 msec
Efficiency	(1500*8)/20836 = 57,5%	(1500*8)/12304 = 97,5%

Tabel 4-1: Snelheidsvergelijking tussen CAN (op 1 Mbit/s) en Ethernet (op 10 Mbit/s), in 2 situaties.

Interessanter wordt de snelheidsvergelijking nog als we gebruik maken van de mogelijkheid van Ethernet om full-duplex te kunnen werken. Bij CAN is dit niet mogelijk, met kan netwerkberichten zenden of ontvangen, maar niet tegelijk. Juist bij de afhandeling van remote I/O is het zinvol om full-duplex te kunnen werken, omdat de outputs aangestuurd moeten worden en de actuele inputs ingelezen. De benodigde netwerkberichten hiervoor kunnen dus parallel met elkaar verstuurd worden. Bij CAN kan dit niet; in zo'n geval is Ethernet dus duidelijk sneller.

Een tweede vergelijking betreft de transmissie van 1500 bytes data. Bij CAN is dit niet mogelijk, want per netwerkbericht kunnen maar 8 bytes data verstuurd worden. De 1500 bytes moeten dan in losse stukken worden opgesplitst; dit kost 187 netwerkberichten met 8 bytes data, en als afsluiter één netwerkbericht met 4 byte data. In totaal kost dit dan 20836 bits transmissietijd (ca. 20,8 msec op 1 Mbit/s). Bij Ethernet passen de 1500 bytes precies in één netwerkbericht, en er gaan 12304 bits over de lijn. Ethernet wint het in deze situatie dus van CAN. Opvallend is dat CAN een 10x lagere bitrate heeft (bruto), maar wel bijna 17x langzamer is (netto). Dit komt omdat de 188 netwerkberichten steeds 47 bits overhead hebben. Dit tikt dus zeer hard aan. De efficiency van CAN heeft zijn plafond bereikt (ca. 57,5%), maar bij Ethernet ligt die op 97,5%.

Uit deze vergelijking kan de conclusie getrokken worden dat CAN het efficiëntst werkt bij transmissie van kleine hoeveelheden data, en Ethernet juist bij grote hoeveelheden data. Beide werken suboptimaal in de omgekeerde situatie. Maar daarvoor zijn ze ook niet ontworpen.

Vergelijkbare berekeningen kan men uiteraard maken voor alle andere industriële netwerken, en dan blijkt dat veel industriële netwerken de concurrentie met Ethernet op 10 Mbit/s uitstekend aankunnen. Een zelfde vergelijking tussen CAN en een 100 Mbit/s Ethernet valt uiteraard altijd in het nadeel van CAN uit, puur door de brute kracht van Ethernet. Uiteraard moet niet enkel naar snelheid gekeken worden, er zijn nog veel andere aspecten waarop men beide netwerken kan vergelijken. Bijvoorbeeld, op 1 Mbit/s is met CAN maar een afstand van ca. 20 meter te overbruggen, tegenover 200 meter op een Ethernet in een basisconfiguratie (met slechts één hub / switch).

4.3 Snelheidsvergelijking met een commercieel product

Wat in bovenstaande berekeningen nog niet meegenomen is, is de (vertragende) invloed van software. In een netwerk zal immers méér gebeuren dan de transmissie van slechts één netwerkbericht. En hierbij komt de software kijken. Elke verwerking van een netwerkbericht kost enige tijd. Hoe sneller het netwerk, des te belangrijker dat wordt.

Een voorbeeld hiervan zijn de performance-getallen die leverancier S voor zijn Ethernet-gebaseerd remote I/O systeem geeft. Het gaat hierbij om een netwerk met 6 remote I/O modules, die elk 16 bits (2 bytes) inputs hebben en 16 bits (2 bytes) digitale outputs. Het gebruikte protocol stuurt elke module eerst de waardes voor de nieuwe outputs op, waarna de waardes van de actuele inputs teruggestuurd worden. Dit is dus een half-duplex methode van communicatie. Theoretisch heeft een besturing hiervoor nodig aan zuivere transmissietijd van netwerkberichten (op 10 Mbit/s):

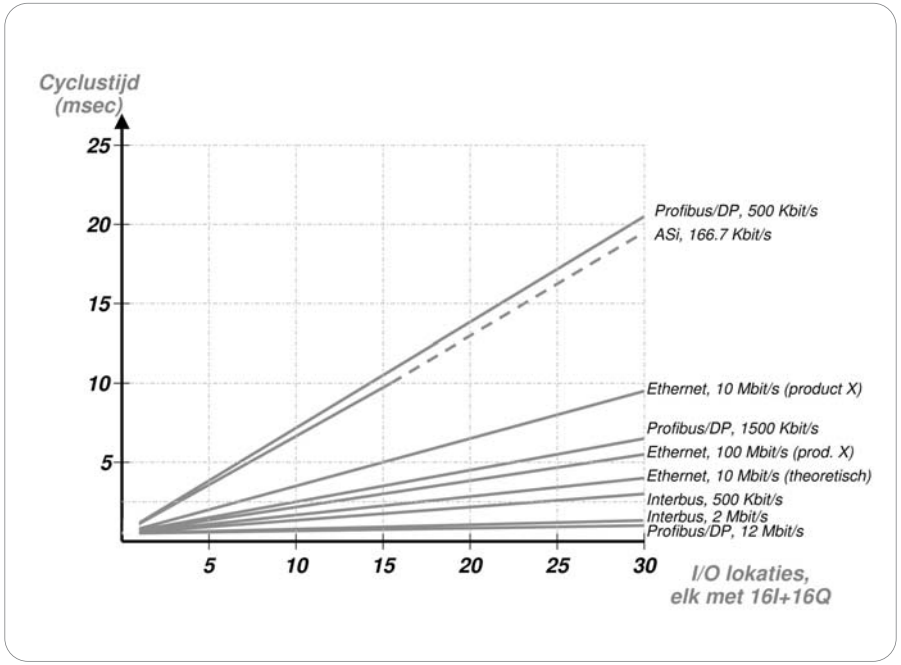
6 x 2 (= inputs en outputs) x 672 bits / 10 Mbit/s = 0,8 msec.

Sneller is gewoonweg niet mogelijk op 10 Mbit/s Ethernet. Langzamer kan natuurlijk wel, en dat blijkt ook uit de opgave van de leverancier: deze claimt een verwerkingstijd voor de 6 remote I/O modules van 1,9 msec. Dit is dus aanzienlijk meer dan de transmissietijd nodig voor alle netwerkberichten. Het verschil moet 'dus' zitten in de software-verwerkingstijden op de besturing en andere vertragingen (1,9 – 0,8 msec = 1,1 msec).

Indien men met de tijd van 1,9 msec niet tevreden is, zou gepoogd kunnen worden om het netwerk op 100 Mbit/s te laten lopen (indien dit ondersteund wordt door alle hardware). In tegenstelling tot wat eerst gedacht wordt, zal het netwerk nu niet 10x sneller worden (0,19 msec). Immers, enkel de transmissietijd van de netwerkberichten zakt naar 0,08 msec. De softwareverwerkingstijden op de besturing worden niet automatisch 10x sneller, en blijven waarschijnlijk 1,1 msec. De totaal benodigde verwerkingstijd is dan 1,1 + 0,08 = ca. 1,2 msec, dus ca. 1/3 sneller en geen factor 10.

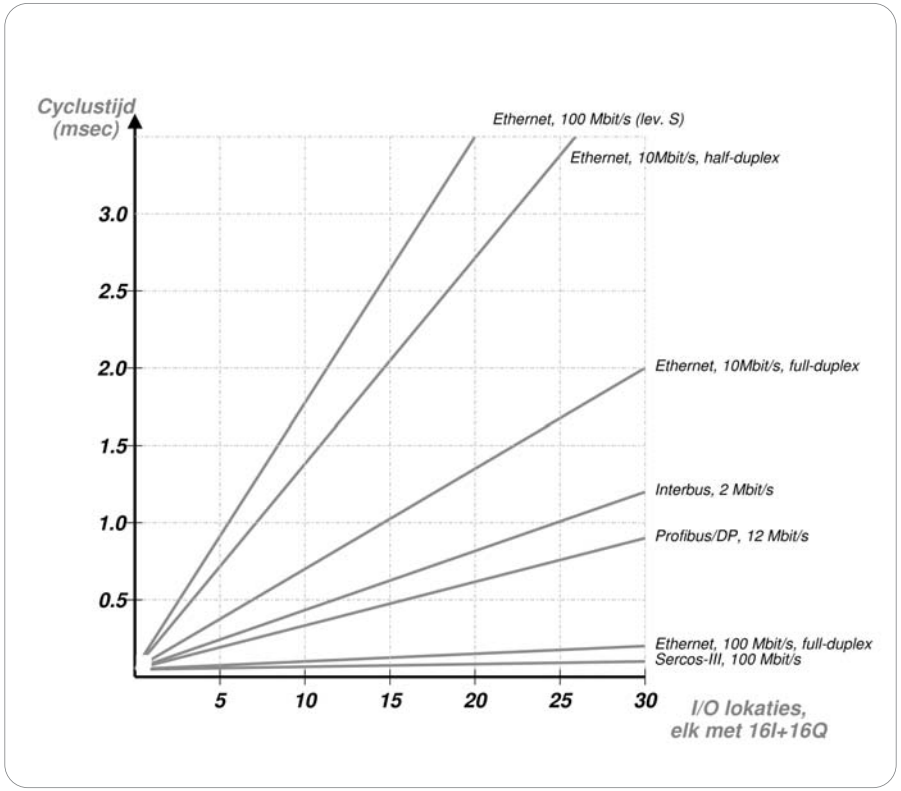
De inzet van een 100 Mbit/s Ethernet op maximale capaciteit vereist dus de nodige aandacht voor de software, welke 'uitgeperst' moet zijn op maximale performance. Dit is zeker **niet** vanzelfsprekend. Het kan dus zéér zinvol zijn om performance-vergelijkingen te maken tussen Ethernetproducten van verschillende leveranciers! Bij bestaande industriële netwerken speelt dit veel minder, omdat daar het protocol veelal geheel in hardware uitgevoerd kan worden (speciale chips of ASIC's). Bij Ethernet is dit (nog) nauwelijks leverbaar.

Figuur 4-2 geeft een snelheidsvergelijking tussen Ethernet en een aantal bekende industriële netwerken. De X-as geeft het aantal remote I/O modules op het netwerk (elk met 16 bits voor inputs en 16 bits voor outputs) weer; de Y-as de benodigde tijd om alle inputs eenmalig te lezen en om alle outputs eenmalig aan te sturen.



Figuur 4-2: Cyclustijden van een aantal bekende industriële netwerken en industrieel Ethernet op 10 Mbit/s op zijn theoretisch maximum snelheid, en op 10 en 100 Mbit/s zoals te koop bij leverancier S.

In deze figuur is ook weergegeven wat de snelheid is van de in deze paragraaf besproken producten op 10 en 100 Mbit/s half-duplex (of dankzij de gekozen Ethernet-variant, of dankzij het gebruikte netwerkprotocol), en wat de theoretische maximum snelheid zou zijn. In figuur 4-3 wordt ingezoomd op een deel van figuur 4-2, en tevens is hier onderscheid gemaakt tussen half- en full-duplex Ethernet-varianten. Uiteindelijk blijkt dan dat 100 Mbit/s Ethernet altijd wint van de bestaande bussystemen. Alleen Sercos-III is nog sneller, maar dat komt omdat dit systeem geen standaard Ethernet is (zie hoofdstuk 6), waardoor het efficiënter werkt.



Figuur 4-3: Uitvergroting van figuur 4-3, tevens zijn hier de theoretische cyclustijden voor 100 Mbit/s full-duplex Ethernet en Sercos-III ingetekend.

4.4 Vergelijking tussen Sercos, ProfiNet en Powerlink

In hoofdstuk 6 worden diverse applicatieprotocollen behandeld die speciaal ontwikkeld zijn voor high-speed motion toepassingen. Dit type applicaties stelt zeer hoge snelheidseisen, en daarom is het voor leveranciers nodig om bij Ethernet echt het onderste uit de kan te halen qua snelheid van protocol (hardware/software), en dat levert een interessante snelheidsvergelijking op tussen diverse protocollen, waaruit nogmaals het verschil tussen een 'bruto' snelheid en een 'netto' snelheid blijkt.

Volgens de Sercos gebruikersvereniging zijn de volgende cyclustijden haalbaar op een netwerk met 10 assen en 12 bytes data per as:

Sercos-III	100 Mbit/s Ethernet	64 µsec
ProfiNet V3	100 Mbit/s Ethernet	106 µsec
Powerlink V2	100 Mbit/s Ethernet	205 µsec

Het verschil tussen Powerlink en ProfiNet is grotendeels te verklaren door het feit dat Powerlink een half-duplex Ethernet eist (wordt anders in Powerlink V3). Ter vergelijking nog enkele niet-Ethernet gebaseerde protocollen:

Sercos-II	16 Mbit	195 µsec
Profibus-DP	12 Mbit/s	1056 µsec

Opvallend hierin is dat zowel Sercos-III, ProfiNet als Powerlink zich allen baseren op een 'bruto' 100 Mbit/s Ethernet, maar dankzij de verschillende protocollen met elk hun eigen overhead is de uiteindelijke nettosnelheid dus steeds anders. De genoemde tijden zijn gebaseerd op implementaties die eind 2003 op de markt beschikbaar waren, en enige verbetering is dus nog te verwachten; geen enkel genoemd protocol (behalve Profibus-DP en Sercos-II) is immers al uitontwikkeld.

5. Hubs en switches

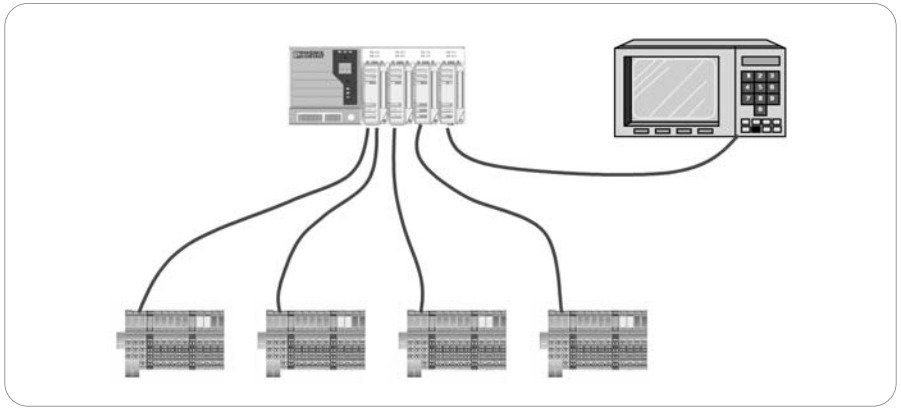
In elk modern Ethernet systeem wordt gebruik gemaakt van minstens één hub of switch, en meestal wel meerdere exemplaren. Aan de buitenkant is weinig verschil te zien, en in veel gevallen maakt het ook weinig uit of een hub of switch gebruikt wordt. In dit hoofdstuk bespreken we de essentiële verschillen tussen een hub en een switch, en de diverse varianten van elk.

5.1 Functie

Bij de coax-gebaseerde Ethernet-varianten is geen hub / switch nodig, maar bij de twisted-pair en glasvezelvarianten wel. Dit komt omdat bij deze varianten gebruik gemaakt wordt van aparte "Receive" en "Transmit" (rx, tx) transmissiepaden (aparte aders in de netwerkkabel).

Een zeer simpel netwerk tussen twee deelnemers kan eenvoudigweg gemaakt worden door de tx van de ene deelnemer aan te sluiten op de rx van de andere deelnemer (en vice-versa). Maar een derder (vierde, vijfde..) deelnemer kan al niet meer aangesloten worden zonder dat men ergens meerdere rx- of tx-signalen aan elkaar moet koppelen. En ook al zou dit werken, dan is het alleen nog maar mogelijk bij koperen kabels; bij glasvezel is dit geheel onmogelijk.

Een tussenoplossing zou nog kunnen zijn dat het netwerk geheel wordt opgebouwd uit point-to-point communicatiepaden. Indien een netwerk 5 deelnemers moet hebben, dan krijgen deze allen 4 netwerkkaarten, en per deelnemer worden 4 kabels getrokken. Dit levert dan een spinneweb-achtige configuratie op. Hoewel dit technisch zeer goed mogelijk is (er zijn ook leveranciers die netwerkkaarten leveren met 4 aansluitingen), is het geen financieel haalbare manier van werken bij grote netwerken. Ook de hoeveelheid werk bij uitbreidingen is gigantisch.



Figuur 5-1: Een hub waarop 5 deelnemers zijn aangesloten, elk via een eigen kabel.

Ethernet heeft gekozen voor het werken met een "hub" (Engels: "naaf"). Dit is een extra deelnemer in het netwerk, welke gaat fungeren als een soort van telefooncentrale. Alle deelnemers op het netwerk worden op de hub aangesloten. Iedereen stuurt zijn netwerkberichten dus altijd naar de hub. De hub regelt vervolgens het doorsturen van het netwerkbericht naar de uiteindelijke bestemming. Hoe een hub dit doet, maakt voor de zender van een netwerkbericht op zich niet zo veel uit; deze weet in feite niet eens dat elk netwerkbericht een of meerdere hub(s) passeert op weg naar de eindbestemming.

5.2 Terminologie

In de loop der jaren zijn twee soorten hubs op de markt gekomen: zo kennen we "hubs", maar er zijn ook "switches". Zuiver datacommunicatietechnisch gezien is een switch ook een hub. De officiële benamingen zijn namelijk:

- Switching hub; en
- Repeating hub.

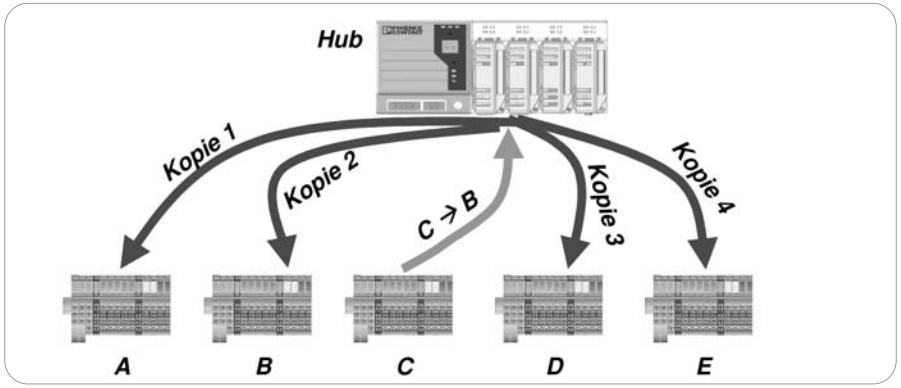
In de praktijk wordt dit echter afgekort tot:

- Switching hub tot "switch";
- Repeating hub tot "hub".

In deze publicatie zal deze conventie ook zoveel mogelijk gevolgd worden.

5.3 Werking van een hub

De werking van een hub is in principe eigenlijk zeer simpel. Op de hub, die voorzien is van een aantal aansluitingen of "poorten" (8, 12, 16, 24, 32.. afhankelijk van fabrikaat en type) kunnen evenzoveel deelnemers worden aangesloten. Indien een deelnemer een netwerkbericht wil sturen, dan zal dit op de hub ontvangen worden. Deze zal dit netwerkbericht dan doorsturen naar alle andere (n-1) deelnemers.

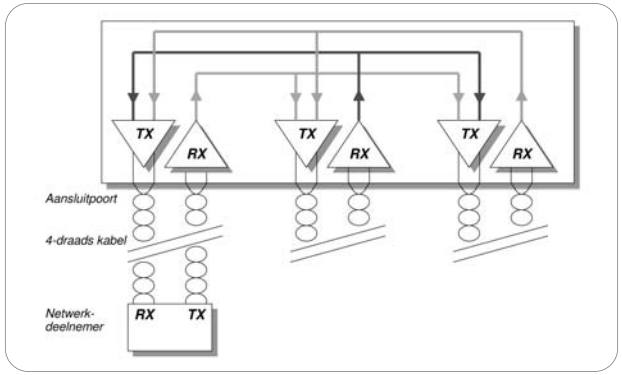


Figuur 5-2: Een hub stuurt elk ontvangen netwerkbericht door naar alle aangesloten deelnemers

Figuur 5-2 toont een hub met daarop aangesloten 5 deelnemers. C wil een netwerkbericht aan B sturen. Dit komt dan zowel op A, B, D en E aan. Dit leidt niet tot problemen, want A, D en E zien dat het netwerkbericht niet voor hen bestemd is, en negeren het dan verder. Deze filterfunctie is integraal onderdeel van Ethernet, en daarom kan een hub zo werken als beschreven. Elk netwerkbericht bevat immers het MAC-adres van de bestemming. Enkel indien het eigen MAC-adres overeenkomt met het MAC-adres in het netwerkbericht, dan wordt het bericht geaccepteerd.

De interne opbouw

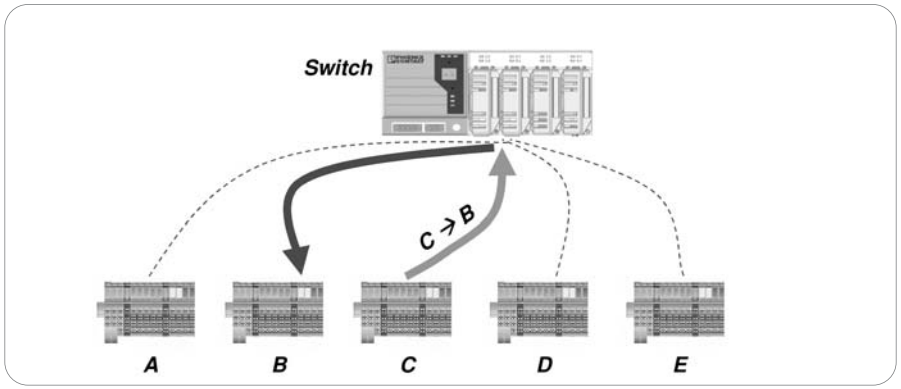
De interne opbouw van een hub is bepalend voor de beschreven manier van werken. Figuur 5-3 toont (schematisch) de interne opbouw van een hub met drie poorten. Hieruit blijkt dat er altijd maar één transmissie tegelijkertijd actief kan zijn, anders zal een collision optreden. Daarom kan met een hub alleen half-duplex (óf zenden of ontvangen) gewerkt worden, zoals ook bij de oudere coax-varianten van Ethernet het geval was.



Figuur 5-3: Interne opbouw van een hub met drie poorten.

5.4 Werking van een switch

Een switch gaat wat subtieler om met ontvangen netwerkberichten dan een hub. Met lokaal aanwezige intelligentie kan ook "in" een Ethernet netwerkbericht gekeken worden. Daardoor wordt het mogelijk te bepalen welke deelnemer de bestemming is. Omdat de switch ook "weet" op welke poort deze deelnemer is aangesloten, zal het ontvangen netwerkbericht doorgestuurd worden op alleen die poort. Op alle andere poorten gebeurt niets. Uiteraard worden broadcast netwerkberichten wel doorgestuurd op alle poorten.

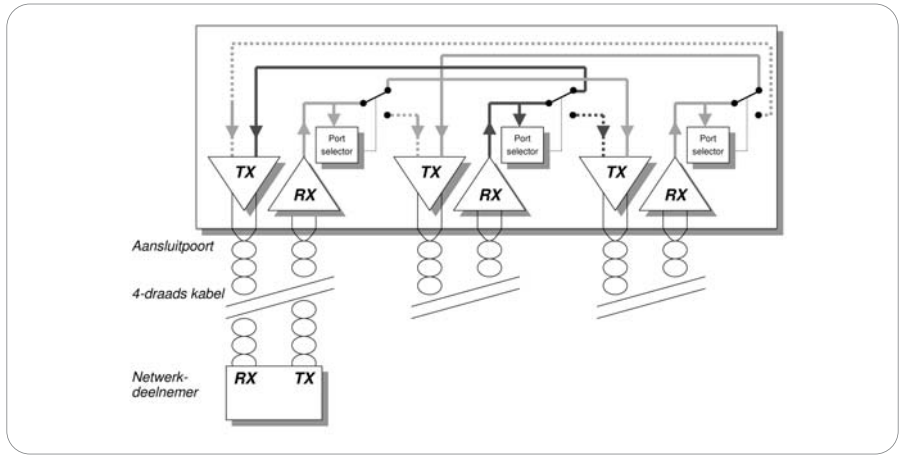


Figuur 5-4: Een switch stuurt een ontvangen netwerkbericht enkel door naar de deelnemer voor wie het bestemd is.

Figuur 5-4 toont een netwerk opgebouwd uit 5 deelnemers. Deelnemer C wil een netwerkbericht sturen naar B. De switch zal het netwerkbericht dan ook alleen naar B doorsturen; deelnemers A, D en E horen niets. Het zou ook geen zin hebben om het netwerkbericht aan hen te sturen, want ze zouden het zelf verder toch negeren (omdat het niet voor hen bestemd is).

De interne opbouw

Figuur 5-5 toont (schematisch) de interne opbouw van een switch met drie poorten. Via de "port selector" schakelaar wordt een netwerkbericht alleen doorgesluisd naar een uitgaande poort. Dat is hier gesymboliseerd getekend met een schakelaar, maar dat is in realiteit uiteraard een elektronische schakelfunctie.



Figuur 5-5: Interne opbouw van een switch met drie poorten.

Omdat de andere poorten niet gebruikt worden, zijn deze vrij voor transmissie van netwerkberichten die wél voor de op die poort aangesloten deelnemer bedoeld zijn. Met een switch is het dus mogelijk om meerdere transmissies van netwerkberichten tegelijk actief te hebben: in principe één per aangesloten deelnemer. Daarnaast mag elke deelnemer op elk moment een transmissie actief hebben *naar* de switch. Dit wordt ook wel "full-duplex" genoemd.

Dit is een van de belangrijkste voordelen van een switch t.o.v. een hub. De totale bandbreedte van het netwerk wordt immers aanzienlijk verhoogd. Op een 10 Mbit/s netwerk met 3 deelnemers heeft men een bandbreedte van 3 * 2 * 10 Mbit/s = 60 Mbit/s (en dat zonder iets te hoeven doen aan bekabeling). Let op: dit getal geldt voor het netwerk als geheel, niet voor elke individuele deelnemer. Elke deelnemer is (in dit voorbeeld) immers nog steeds aangesloten op een 10 Mbit/s verbinding. Wel is er een voordeel omdat hij tegelijkertijd netwerkberichten kan zenden en ontvangen.

Let er wel op dat het full-duplex voordeel alléén geldt als het netwerkverkeer gelijkelijk verspreid is tussen alle netwerkdeelnemers. Is er in de applicatie één centrale deelnemer, zoals in veel systemen waarbij gebruik gemaakt worden van een master/slave protocol, dan heeft men geen voordeel van een switch boven een hub. Ook indien protocollen gebruikt worden waarbij alle netwerkberichten gebroadcast worden, heeft een switch minder zin, want deze zal zich dan (dankzij de broadcast) net zo gaan gedragen als een hub – iedereen krijgt elk netwerkbericht.

Werking van de port selector
De port selector kan zijn werk op twee verschillende manieren doen:

- Zodra het MAC-bestemmingsadres uit het ontvangen netwerkbericht binnen is (dit staat vooraan in elk Ethernet-netwerkbericht) kan uitgerekend worden welke uitgaande poort gebruikt moet worden. Terwijl de rest van het netwerkbericht nog niet eens binnen is, kan al begonnen worden met de transmissie van het deel dat al wel binnen is. Dit is dus een zeer efficiënte manier om netwerkberichten door te sluizen. Switches die zo werken worden "Cut-through" switches genoemd.
- Eerst wordt het gehele netwerkbericht ingelezen en lokaal opgeslagen. Daarna wordt bepaald (via het MAC-bestemmingsadres) wat de uitgaande poort moet zijn. Dan pas wordt begonnen met de transmissie van het ontvangen netwerk bericht. Switches die zo werken worden "Store-and-Forward" switches genoemd.

Het nadeel van cut-through switches is dat ze geen foutdetectie op ontvangen netwerkberichten kunnen doen, en alle fouten dus ook mee doorsturen (deze worden uiteindelijk wel uitgefilterd en gedetecteerd op de bestemming). Omdat in de loop der jaren store-and-forward switches ook steeds sneller geworden zijn, komen cut-through switches niet zo vaak meer voor.

Non-blocking switches
Zeker op snelheden van 100 Mbit/s en hoger kan een switch met een groot aantal poorten het intern aanzienlijk druk krijgen. Dit kan er toe leiden dat deze het aanbod van netwerkberichten niet kan verwerken, en sommige netwerkberichten dus zal moeten negeren. De bandbreedte die zo'n switch biedt is dus niet evenredig met het aantal poorten.

Er zijn ook switches op de markt die geen enkel probleem hebben met het verwerken van elk willekeurig berichtaanbod. Zulke switches worden "non-blocking" genoemd.

Interne buffering
Ook al is een switch in staat om parallelle transmissies van netwerkberichten af te handelen (zowel voor ontvangst als voor versturen), een probleem kan ontstaan als twee of meer deelnemers tegelijk netwerkberichten naar dezelfde eindbestemming willen sturen. Naar een bepaalde deelnemer kan op elk moment toch maar één transmissie tegelijk actief zijn. Dit kan door de switch op verschillende manieren afgehandeld worden:

- *Negeren*: Het netwerkbericht dat toevallig als eerste weggestuurd kan worden, wordt gewoon verstuurd. Totdat deze transmissie klaar is, worden alle netwerkberichten voor diezelfde deelnemer eenvoudigweg weggegooid. Het tenietgaan van deze berichten moet dan door het netwerkprotocol op een hoger niveau (TCP doet dit o.a.) "gerepareerd" worden.
- *Tijdelijke opslag*: het netwerkbericht dat toevallig als eerste weggestuurd kan worden, wordt gewoon verstuurd. Totdat deze transmissie klaar is, worden alle netwerkberichten voor diezelfde deelnemer opgeslagen in intern geheugen, en alsnog verstuurd zodra het mogelijk is. De beschikbare hoeveelheid intern geheugen is leveranciersafhankelijk; indien het geheugen vol is, moeten ontvangen netwerkberichten alsnog weggegooid worden. Dit leidt dan tot hetzelfde gedrag als hierboven beschreven. Tijdelijke pieken in netwerkbelasting voor een bepaalde deelnemer kunnen dus worden opgevangen, langdurige piekbelastingen niet.

Raadpleeg uw leverancier indien het nodig is om exact te weten hoe een switch intern functioneert.

De switchtabel
Een switch moet ontvangen netwerkberichten doorsturen op die poort waarachter de netwerkdeelnemer voor wie het bericht bestemd is, is aangesloten. Maar hoe weet een switch dat? De netwerkbeheerder zou dit natuurlijk kunnen configureren,

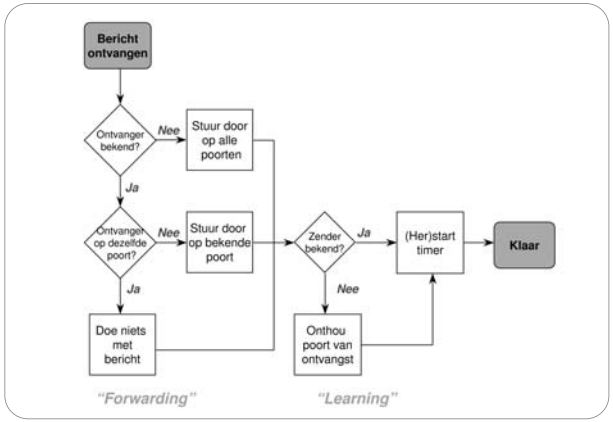
maar bij grotere netwerken is dit niet werkbaar. Daarom zijn switches "zelflerend": tijdens bedrijf wordt geleerd welke netwerkdeelnemer op welke poort is aangesloten (figuur 5-6). Dit gaat als volgt:

1. Zodra de switch een netwerkbericht ontvangt op een poort, is automatisch bekend wie de afzender is. Dit staat namelijk in het netwerkbericht zelf: het zgn. MAC-adres (Medium Access Control). De switch onthoudt dit netwerkadres, en ook op welke poort het ontvangen is. Deze informatie wordt in een "switchtabel" bijgehouden, met een maximale omvang (leveranciersafhankelijk) van bv. 1000 adres/poort combinaties.

2. In elk ontvangen netwerkbericht is ook aangegeven wat de bestemming is. In de tabel van stap 1 wordt opgezocht of bekend is op welke poort de bestemming is aangesloten.

3. Is de poort bekend, dan wordt het netwerkbericht op die poort doorgestuurd. Is de poort niet bekend, dan wordt het netwerkbericht op alle poorten van de switch doorgestuurd. Dan komt het vanzelf op de eindbestemming aan.

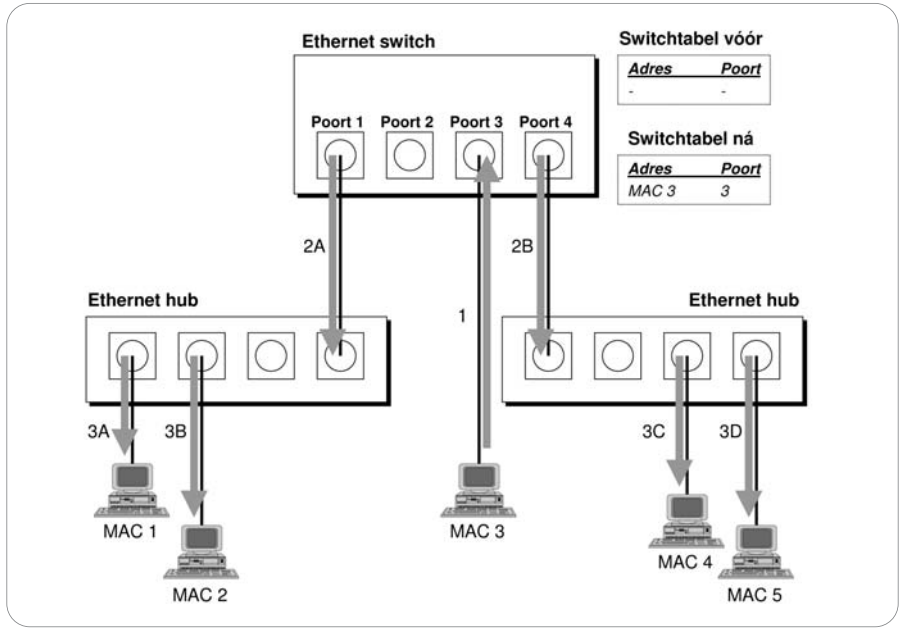
4. Hierop is één uitzondering. In grotere netwerken kan het voorkomen dat de bestemming achter dezelfde poort is aangesloten als de zender van het netwerkbericht, bijvoorbeeld in een hiërarchie van een switch met meerdere hubs. In zo'n geval heeft de bestemming het bericht al ontvangen. De switch doet daarom niets als een bericht op dezelfde poort verstuurd zou moeten worden als dat ze ontvangen is.



Figuur 5-6: Stroomschema van de werking van een switch, en het gebruik van de switchtabel.

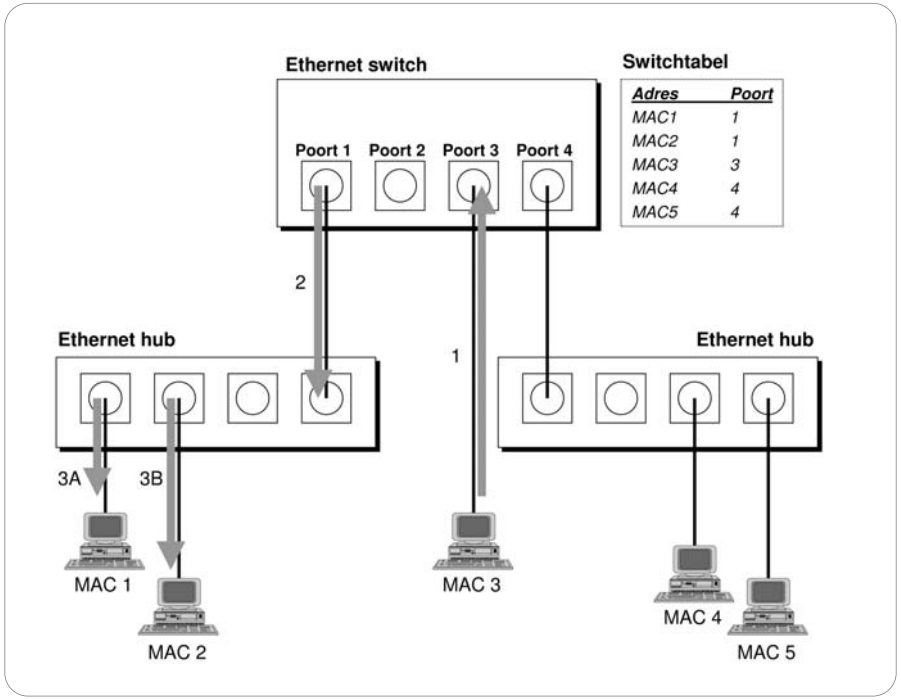
In de volgende figuren tonen we hoe een netwerk, bestaande uit 1 switch, 2 hubs en 5 deelnemers functioneert. Initieel (na spanning aan) is de switchtabel uiteraard leeg. Een transmissie van deelnemer 3 naar deelnemer 2 zorgt er dan voor dat de switch zich eigenlijk ook als een hub gedragen moet, omdat de switch (nog) niet weet via welke poort deelnemer 2 bereikbaar is. De switch stuurt het ontvangen netwerkbericht eenvoudigweg op alle poorten naar buiten, zodat het zeker op deelnemer 2 aankomt.

Toevallig zijn op de switch alleen beide hubs aangesloten; deze sturen elk netwerkbericht altijd door op alle poorten. Uiteindelijk krijgen alle deelnemers (1, 2, 4 en 5) een kopie van deelnemer 3 zijn netwerkbericht. Omdat het enkel bestemd is voor deelnemer 2 (zichtbaar via het bestemmings MAC-adres dat 3 opgegeven heeft) zullen deelnemer 1, 4 en 5 dit netwerkbericht verder negeren (figuur 5-7). Na afloop van deze actie "weet" de switch nu ook dat deelnemer 3 op poort 3 aangesloten is. De switchtabel begint dus te groeien!



Figuur 5-7: Deelnemer 3 stuurt een bericht naar deelnemer 2. Omdat de switchtabel nog leeg is, stuurt de switch het netwerkbericht op alle poorten door. De hierop aangesloten hubs doen dit ook (altijd). Het netwerkbericht komt dan uiteindelijk bij deelnemer 2 uit; de andere deelnemers ontvangen het bericht ook maar negeren het verder.

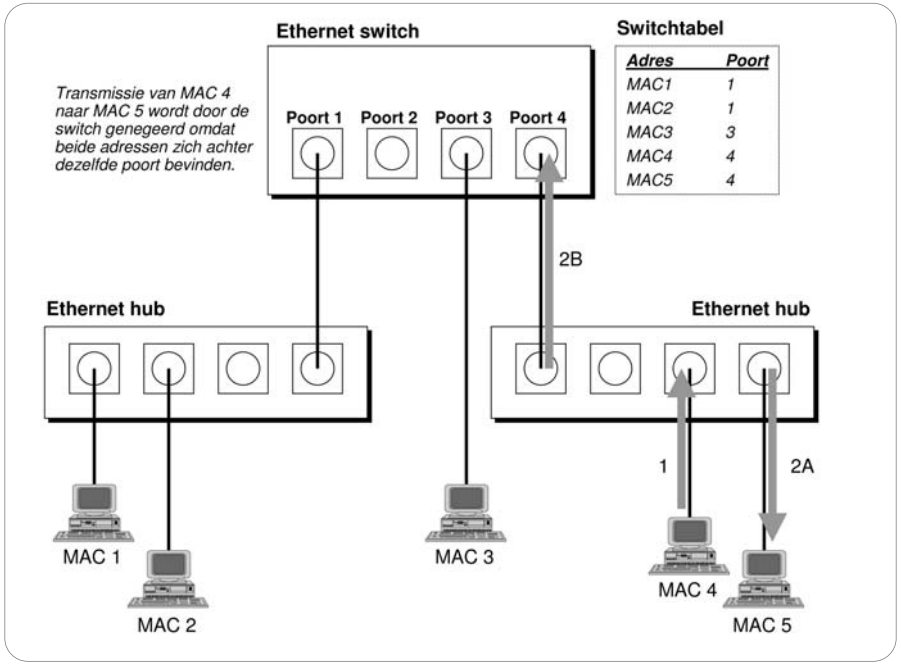
Als enige tijd later de switchtabel geheel gevuld is (van alle aangesloten deelnemers is bekend via welke poort ze bereikbaar zijn) wordt de verwerking van een netwerkbericht van deelnemer 3 naar 2 anders uitgevoerd. De switch "weet" nu immers (figuur 5-8) dat deelnemer 2 via poort 1 bereikbaar is. Het netwerkbericht wordt nu dus alléén via poort 1 doorgestuurd, en niet meer via poort 4. Uiteraard zal de hub het netwerkbericht ook nog steeds doorsturen naar deelnemer 1, maar deze zal het nog steeds negeren aangezien het niet voor hem bestemd is.



Figuur 5-8: Als de switchtabel uiteindelijk geheel gevuld is, zal een transmissie van deelnemer 3 naar 2 ervoor zorgen dat op poort 4 niets meer wordt verstuurd, maar alleen op poort 1.

Als laatste voorbeeld een situatie waarbij deelnemer 4 iets stuurt naar deelnemer 5 (figuur 5-9). Dit netwerkbericht komt allereerst op de hub binnen, en deze stuurt het netwerkbericht door naar de switch én naar deelnemer 5. Deze is dus direct bediend. Maar wat moet de switch nu nog doen met dit netwerkbericht? Volgens de switchtabel zit deelnemer 5 op poort 4, maar het bericht is ook binnengekomen via deze poort 4.

Hierdoor "weet" de switch dat het geen zin heeft om het netwerkbericht nóg eens te versturen, en het netwerkbericht wordt verder genegeerd. Zou de switch het bericht wel doorsturen, dan zou deelnemer 5 twee maal hetzelfde netwerkbericht ontvangen.



Figuur 5-9: Een transmissie van deelnemer 4 naar 5 zal door de switch niet doorgestuurd worden, want deelnemer 5 zit op dezelfde poort als 4, en zal het bericht dus al ontvangen hebben (via de hub).

Omvang van de switchtabel

De omvang van de switchtabel is meestal ruim voldoende voor een normaal netwerk; en zal minimaal zo'n 1000 MAC-adressen kunnen bevatten. Het is echter nergens gespecificeerd dat er een minimale omvang moet zijn; bij twijfel kunt u dit altijd bij de leverancier navragen. Velen vermelden het automatisch in hun documentatie of op de website. Tevens zal men hier kunnen vinden hoe omgegaan wordt met het beheer van de switchtabel: lengte van timers, mogelijkheid tot voorgeprogrammeerde instellingen, blokkades van bepaalde MAC-adressen, etc.

5.5 Switch mogelijkheden

De in de vorige paragraaf beschreven werking van een switch is slechts de standaardfunctionaliteit. Bij sommige types switches is nog véél meer functionaliteit mogelijk. In deze paragraaf bespreken we enkele van de mogelijkheden.

L2..L7 switches

De in de vorige paragraaf geschetste werking van een switch is de meest eenvoudige. Er bestaan diverse types switches, die afhankelijk zijn van hoeveel informatie de switch uit het netwerk-bericht leest. Op basis van deze informatie besluit de switch dan welke poort gebruikt moet worden. De benaming van de types switches correspondeert met het OSI 7-lagen model. In elk netwerkbericht is administratie aanwezig voor de lagen 2 t/m 7 (laag 1 beschrijft de fysieke bekabeling, en daarvoor is geen administratie nodig).

Een switch die de administratie van OSI-laag 2 uit het netwerkbericht leest en daar op reageert, is dan een zgn. "laag 2 (L2) switch". Bij Ethernet bevat deze administratie: het MAC-adres van de afzender, het MAC-adres van de bestemming, het gebruikte protocol van OSI-laag 3, en/of de lengte (in bytes) van de gebruikersdata. Een switch die de administratie van OSI-laag 3 leest en gebruikt is een "laag 3 (L3) switch", etc. en zo kunnen we dus ook een L4, L5, L6 en L7 switch onderscheiden. Hoe hoger in het OSI-model, des te meer kennis is aanwezig over het gebruikte netwerkprotocol en de gebruikersapplicatie. Dan kan de switch dus ook intelligentere beslissingen nemen over wat met elk netwerkbericht gedaan moet worden.

Een L2 switch functioneert zoals in de vorige paragraaf beschreven. Omdat ze alleen maar hoeven te beslissen op basis van de netwerkadressen in elk bericht zijn ze vrij simpel, maar daarom ook zeer snel. De zgn. "latency" of doorvoervertraging is een belangrijke concurrentiefactor tussen leveranciers.

Een L3 switch is eigenlijk meer een "router", een ander randapparaat dat op (grotere) netwerken gebruikt wordt. L3 switches werken sneller dan routers, maar ze ondersteunen minder protocollen en toepassingen. L3 switches worden doorgaans gebruikt in lokale netwerken, terwijl routers in de regel worden gebruikt om remote netwerken (bv. vestigingen van een bedrijf) via een WAN (Wide Area Network) met elkaar te verbinden.

Een L4 switch heeft kennis van de TCP en UDP protocollen. Op basis van de administratie hiervan kan gezien worden wat de gebruiker aan het doen is: telefoneren? websurfen?

email? filetransfer? iets anders? Op basis hiervan kan de switch geconfigureerd worden om bepaalde vormen van netwerkgebruik voorrang te geven, bijvoorbeeld IP-telefonie krijgt voorrang boven email, en dit weer voorrang op het websurfen. Dit is de zgn. "Quality of Service" (zie onder). Tevens bieden L4 switches de mogelijkheid om netwerkverkeer uit te splitsen. Dit is zeer praktisch bij zwaar belaste websites. Als één webserver al het werk niet meer aankan en een tweede webserver er bij wordt gezet, kan de switch zo ingesteld worden dat beide webservers elk gemiddeld 50% van de aanvragen te verwerken krijgen, en dit alles zonder dat naar deze website surfende persoon (elders op de wereld) dit merkt.

De verschillen tussen L5 en L7 switches zijn wat onduidelijk en afhankelijk van het merk switch. L6 switches komen bijna niet voor. Met een technologie die aangeduid wordt als "webswitchen" kijkt de switch mee naar web-verkeer. Bijvoorbeeld, de switch kan zien welke URL (http://www....) opgevraagd wordt, en het netwerkbericht dan zelf doorsluizen naar de juiste server.

De meeste industriële switches zijn L2 switches. De extra snufjes van L4..L7 switches zijn goed bruikbaar voor webservers en LAN-servers, maar zijn niet nodig voor de meerderheid van de industriële applicaties.

Quality of service

Bij standaard Ethernet is het niet mogelijk om voorrang te verlenen aan bepaalde netwerkberichten. Meestal gaat dit op basis van "First In, First Out". De IEEE 802.1p norm geeft de mogelijkheid om voorrang te geven aan bepaalde klasses Ethernetberichten. Hierdoor kan kritieke (real-time) en vertragingsgevoelige data (audio, video) voorrang krijgen boven de rest van het verkeer op het netwerk. Het werkt als volgt: de oorspronkelijke verzender van het netwerkbericht kent een "prioriteitsklasse" aan het netwerkbericht toe, dat hierdoor 4 bytes groter wordt. Deze vier bytes, de zgn. "tag", worden door de switch gelezen om te weten te komen wat de prioriteit van het netwerkbericht is. Berichten met een hoge prioriteit worden dan eerder verstuurd dan berichten met lage(re) prioriteit.

802.1p heeft acht prioriteitsklassen, waarbij "1" de laagste en "8" de hoogste is. De waarde "0" betekent "normale verzending" oftewel "geen prioriteitsklasse toegekend". Het aantal feitelijk ondersteunde prioriteitsklassen is echter afhankelijk van het merk van de apparatuur. Omdat er veel apparatuur is die 802.1p niet aankan, moet de switch per poort hiervoor geconfigureerd worden. Als er op een poort apparatuur is aangesloten die geen 802.1p kent, dan zal de switch de tag weer verwijderen. Het is dan weer een "normaal" Ethernet-bericht geworden, en elke prioriteitsafhandeling gaat verloren.

Let op dat er veel apparatuur op de markt is die claimt wel ondersteuning te hebben voor 802.1p, maar vaak blijkt dat deze dan niet 8 maar slechts 2 prioriteitsklassen onderscheiden: 0..3 en 4..7. Bij de toewijzing van prioriteitsklassen moet hier dus rekening mee gehouden worden.

Het ondersteunen van 802.1p wordt vaak genoemd als de reden dat Ethernet geschikt is voor real-time applicaties. In veel gevallen zal het dat ook al zijn zonder 802.1p, maar uiteraard kan 802.1p hierbij helpen. Ga er echter niet van uit dat elke industriële switch 802.1p aan kan. En afgezien daarvan moet de 802.1p tag worden toegevoegd door de oorspronkelijke zender van het bericht, en dat is ook niet altijd mogelijk.

Port trunking

Switches mogen aan elkaar gekoppeld worden. Als dit via één kabel gebeurt die op beide switches aangesloten wordt, dan geeft dit dus een bandbreedte van 10 of 100 Mbit/s. Soms is dit te weinig, maar men wil toch niet overgaan op (veel duurdere) 1 Gbit/s switches. Dan kan "port trunking" helpen. Bij switches die dit ondersteunen moeten dan meerdere kabels tussen beide switches aangesloten worden. Bijvoorbeeld, bij 2 kabels krijgt men een bandbreedte van 20 of 200 Mbit/s, bij drie kabels 30 of 300 Mbit/s, etc. Uiteraard moeten beide switches wel 'weten' dat dit gebeurt, anders zullen ze maar één kabel gaan gebruiken en heeft de hele exercitie dus geen zin. Aangezien niet alle switches port trunking ondersteunen, moet men hier bij aanschaf dus wel specifiek op letten.

Een andere reden om port trunking te gebruiken is vanwege de redundantie die het biedt. Als men drie kabels tussen twee switches heeft liggen en een ervan valt uit, dan heeft men nog steeds 2/3 van de oorspronkelijke capaciteit over. En omdat de switches dit helemaal zelf uitzoeken, zijn er geen speciale protocollen (zoals het langzame Spanning Tree) nodig om dit alles te regelen.

Port Mirroring

In tegenstelling tot een hub zal een switch alle inkomend netwerkverkeer meestal maar naar precies een van de uitgaande poorten doorsturen. Dat is soms lastig als men een netwerk-analyzer of –monitor aan wil sluiten om te volgen welke netwerkberichten er door wie wanneer worden verstuurd. De analyzer / monitor / sniffer zal alleen de broadcast-netwerkberichten ontvangen en de rest niet, want geen enkel netwerkbericht is immers specifiek voor hem bestemd. Dat maakt het analyseren van het netwerkverkeer dus zo goed als onmogelijk.

Een switch die "mirroring" ondersteunt, heeft de mogelijkheid om alle uitgaande berichten van een bepaalde poort te kopiëren naar een andere poort, zodat men dus precies kan volgen wat er allemaal op het netwerk rondgaat.

Afhankelijk van de implementatie is het soms niet mogelijk om op een poort die "mirroring" aan heeft staan ook netwerkberichten te sturen. Dit is meestal geen probleem omdat een analyzer / monitor / sniffer zelfs niets zal sturen.

Flow control

Ook al is een switch in staat om op volle snelheid te communiceren met een deelnemer op elke poort, dan wil dat nog niet zeggen dat de switch in staat is om met alle deelnemers tegelijk op volle snelheid te communiceren, of dat elke deelnemer snel genoeg is om alle door de switch verzonden netwerkberichten te verwerken. Indien men de netwerkberichten niet snel genoeg kan verwerken, dan is bij de meeste apparatuur de reactie vrij primair: weggooien. Men kan dit meestal straffeloos doen, omdat netwerkprotocollen op hogere niveaus (zoals bv. TCP) dit weer corrigeren.

Maar het is uiteraard een verspilling van netwerkcapaciteit en CPU-capaciteit door een netwerkbericht eerst te sturen om het daarna te laten negeren. Beter zou het zijn om het netwerkbericht zowiezo niet "zomaar" te sturen, maar pas op een moment dat de ontvanger er iets mee kan doen (op zijn minst opslaan in het geheugen voor latere verwerking). Deze mechanismes bestaan, en worden "flow control" genoemd.

Flow-control kan op verschillende niveaus uit het OSI 7-lagen model uitgevoerd worden. Bij RS232 is flow-control mogelijk via speciale handshake-lijnen (OSI-laag 1). Ethernet kent dit echter niet. Een switch functioneert op OSI-laag 2, en heeft dus het probleem dat deze niet in de netwerkberichten kan kijken, omdat er geen kennis van de gebruikte netwerkprotocollen aanwezig is. De switch kan dus niet zelf de taak van (bijvoorbeeld op OSI-laag 4) TCP overnemen, welke flow-control kan uitvoeren op het gehele traject tussen twee met elkaar communicerende apparaten. Wat een switch wél kan doen is: op de eigen Ethernet kabelsegmenten flow-control uitvoeren. Dit betreft dus steeds één kabelsegment, derhalve tussen de switch zelf en het apparaat aan de andere kant van de kabel, of tussen twee switches onderling. Er is dus géén flow-control over het gehele traject van bron van een netwerkbericht tot eindbestemming ervan.

De flow-control per kabelsegment is een optie die een leverancier in zijn apparatuur in kan bouwen. Of een apparaat flow-control ondersteunt of niet wordt uitgezocht tijdens

de auto-negotiation fase. Het zgn. "PAUSE" bit geeft aan dat een apparaat flow-control ondersteunt. Indien beide apparaten dit kunnen, en er wordt op een full-duplex gewerkt, dan wordt flow-control gebruikt.

De flow-control functie zelf wordt uitgevoerd door de "MAC Control" functie, welke onderdeel is van de datalinklaag (OSI-laag 2) van Ethernet. Op dit moment heeft de MAC Control sublaag maar één taak, en dat is het uitvoeren van flow-control; maar misschien worden er in de toekomst meer taken aan de MAC Control sublaag gegeven. Deze sublaag kan zelfstandig Ethernet netwerkberichten sturen. Deze voldoen aan het normale bericht-formaat. Sommige velden hierin krijgen echter een speciale betekenis.

Het MAC-bestemmingsadres is (hexadecimaal) 01-80-C2-00-00-01. Dit is dus niet het MAC-adres van de feitelijke bestemming, maar een multicast adres. Op de ontvanger moet het dus wel mogelijk zijn om dit type netwerkberichten te kunnen ontvangen! Feitelijk is het werken met een multicast adres niet zinvol, omdat er altijd maar precies één deelnemer met één MAC-adres aan het andere eind van de kabel aangesloten kan zijn. De reden dat van een multicast adres gebruik gemaakt wordt is dat men dan het MAC-adres van de andere partij niet hoeft te weten. Het type-veld krijgt de (hexadecimale) waarde 88:08. Van het dataveld (46 bytes) worden alleen de eerste 4 bytes gebruikt. De eerste twee bytes krijgen de waarde (hexadecimaal) 00:01. Dit is de zgn. "Opcode", waarmee in de toekomst nieuwe functies van de MAC Control kunnen worden onderscheiden. De volgende twee bytes geven het "Pauzegetal" aan, in stappen van 512 bits. De resterende 42 bytes in het dataveld krijgen de waarde 00.

De ontvanger van een flow-control netwerkbericht kan aan de hand van het pauzegetal zien gedurende welke tijd geen netwerkberichten dienen te worden verstuurd. Bijvoorbeeld, als het pauzegetal de waarde 10 heeft, en er wordt op 10 Mbit/s gewerkt, dan is de pauzetijd 10 * 512 / 10M = 0,512 msec. De maximale pauzetijd bedraagt 65535 * 512 / 10M = 3,355 sec. Op hogere bitrates zijn deze tijden natuurlijk evenredig korter. Als een switch eenmaal een bepaalde pauzetijd heeft opgegeven maar tot de ontdekking komt dat deze te lang is, dan kan door het sturen van een netwerkbericht met waarde 0 voor het pauzegetal de (resterende) pauzetijd worden afgebroken, zodat direct weer met transmissie van netwerkberichten kan worden begonnen. Het is ook mogelijk dat een andere pauzetijd wordt opgegeven. Let op dat de flow-control netwerkberichten dus zelf niet onderworpen zijn aan de pauzetijd, ze mogen op elk moment verstuurd worden.

Flood control

Met "flood control" wordt een eigenschap van een switch aangegeven die helpt bij het indammen van excessief netwerkverkeer door één of meerdere deelnemers.

Het meest extreme voorbeeld hiervan is een zgn. "broadcast storm", die initieel in gang kan worden gezet door één deelnemer en daarna door iedereen op het netwerk in gang gehouden wordt. Het begint bij een deelnemer die (meestal per expres) een netwerkbericht stuurt aan iedereen (bestemmings MAC-adres is FF-FF-FF-FF-FF-FF), en als het afzender MAC-adres niet zijn eigen MAC-adres gebruikt maar óók het broadcast-adres. De inhoud van het netwerkbericht wordt zodanig gevuld dat het een ongeldig netwerkbericht van een hoger protocol lijkt. Alle ontvangers voelen zich dan geroepen hierop een foutmelding te sturen, maar dat wordt óók weer per broadcast verstuurd (want het afzender MAC-adres had deze waarde). Hierop herhaalt de cyclus zich weer. Iedereen op het netwerk is dan bezig om broadcasts met foutmeldingen aan de anderen te sturen, die hierop weer reageren aan iedereen, etc. De netwerkbelasting stijgt zo sterk dat normaal netwerkverkeer geen kans meer krijgt; dit wordt ook wel een "network meltdown" genoemd. Ook de aangesloten deelnemers moeten zoveel CPU-tijd aan de verwerking van de (nutteloze) broadcasts besteden dat er verder geen zinnig werk meer kan worden uitgevoerd. Dit kan dan het netwerk zo zwaar kan gaan belasten dat men over moet gaan tot het uitschakelen of loskoppelen van alle aangesloten deelnemers.

Broadcast storms kunnen ook op een legitieme wijze ontstaan, meestal omdat er fouten gemaakt zijn in de netwerkconfiguratie. Zo' n broadcast storm wil wel eens vanzelf weer verdwijnen, om na een bepaalde tijd weer opnieuw te beginnen.
Voor gebruikers is dit zichtbaar doordat af-en-toe het netwerk zeer langzaam lijkt, maar even later is alles weer OK.

Een broadcast storm kan door een switch tegengewerkt worden door per poort limieten te stellen aan het aantal netwerkberichten dat per seconde gebroadcast mag worden. Als die limiet bereikt wordt, kan het volgende gebeuren:

- De excessieve broadcasts worden verwijderd; of
- Geen enkele broadcast wordt nog doorgestuurd; of
- De poort wordt (tijdelijk) uitgeschakeld.

Wat er precies gebeurt is leveranciersafhankelijk, evenals de instelling(en) van de limieten.

Naast broadcast storms bestaan er ook nog multicast storms en unicast storms. Ook hieraan kan een switch limieten stellen. In tegenstelling tot broadcast storm filters komt men multicast en unicast storm filters veel minder in apparatuur tegen.

IGMP Snooping

Dit is een feature die in switches ingebouwd kan zijn om te zorgen voor een vermindering van het aantal multicast-transmissies. Deze kunnen worden opgezet via het "IGMP" protocol (Internet Group Management Protocol). Dit is o.a. nodig voor het kunnen volgen van real-time audio- en video-transmissies die door meerdere deelnemers tegelijk ontvangen kunnen worden. In principe zal de server dan voor elke deelnemer álle netwerkberichten opnieuw sturen, ook al zijn die feitelijk voor alle deelnemers steeds hetzelfde (alleen de eindbestemming is steeds anders). Dit kan een verspilling van netwerkcapaciteit opleveren, zeker als meerdere eindbestemmingen op dezelfde switch zijn aangesloten.

Indien in de switch "IGMP Snooping" is ingebouwd, dan zal de switch bespioneren ("snoopen") welke deelnemers gebruik maken van IGMP. Mochten deze deelnemers dan ook van dezelfde webserver gebruik willen maken, dan zal de switch alle verzoeken van de 2e (3e, 4e,...) deelnemer naar de webserver blokkeren. Echter, de switch zal wel alle data die door de webserver naar de 1e deelnemer wordt gestuurd, ook naar de andere deelnemers sturen. Deze zien verder geen enkel verschil.
Het voordeel is dus dat de webserver zelf alles maar één keer hoeft te sturen, en dat de tussenliggende netwerkinfrastructuur veel minder belast wordt. Alle kopieën worden immers pas zo laat mogelijk gemaakt.

Het per multicast sturen van data is ooit ontwikkeld voor het per Internet transporteren van audio en video. In de meeste industriële Ethernets zal dit type gebruik niet voorkomen, en op het eerste gezicht lijkt het daarom zinloos voor leveranciers om toch IGMP Snooping in hun producten in te bouwen. Toch is er een zeer zinvol gebruik van IGMP mogelijk: in combinatie met publish/subscribe (ook wel producent/consument) netwerkprotocollen. Bij dit type protocollen worden wijzigingen in data aan alle geinteresseerden doorgestuurd. De meest eenvoudige vorm van implementatie doet dit via een Ethernetbroadcast, maar dit heeft de volgende nadelen:

- Broadcasts belasten ook de deelnemers die géén interesse in de data hebben.
- Broadcasts worden niet doorgegeven door routers (wel door hubs en switches).

Door gebruik te maken van multicasts in combinatie met IGMP snooping gelden deze nadelen niet. Iedereen die interesse heeft in de broadcasts meldt zich aan als lid van de multicast groep.

Een "multicast groep" bestaat uit een server (als bron) en één of meerdere ontvangers. Een deelnemer op het netwerk kan lid worden van een multicast groep door zich aan te melden; uiteraard kan hij zich ook weer afmelden. Op een netwerk kunnen meerdere multicast groepen tegelijkertijd actief zijn; iedereen kan ook lid zijn van meerdere multicast groepen tegelijk. De administratie hiervan wordt gevoerd via het "IGMP" protocol. Via "IGMP Report Group" en "IGMP Leave Group" commando's kunnen deelnemers zich aan- en afmelden uit multicasts groepen. Hoe dit precies geschiedt, is afhankelijk van de IGMP variant: 1, 2 of 3. Versie 2 is nog steeds zeer gebruikelijk; b.v. Microsoft ondersteunt IGMP versie 3 pas vanaf Windows XP, en ook in de diverse Unix / Linux varianten is IGMP versie 3 niet altijd standaard aanwezig.

Zónder IGMP snooping kan een switch multicasts alleen maar op dezelfde manier behandelen als broadcasts: doorsturen op elke uitgaande poort. Mét IGMP snooping zal een switch multicasts naar een bepaalde groep deelnemers alleen doorsturen naar poorten waarop leden van die groep zijn aangesloten.

Waarom zal een switch zonder IGMP snooping altijd alle multicasts doorsturen op elke poort? Dit komt dankzij het speciale "Group Destination Address" (GDA), een MAC-adres met de speciale waarde 01-00-5E-XX-YY-ZZ (zie hoofdstuk 2); de waardes voor XX, YY en ZZ bepalen om welke multicast-groep het hier gaat. In tegenstelling tot 'gewone' MAC-adressen wordt dit MAC-adres nóóit ingevuld in het afzenderveld van een Ethernetbericht. Daarom kan de switchtabel nooit leren op welke poort dit MAC-adres te bereiken is. En de switch zal dan, net zoals met gewone Ethernet-berichten waarvan onbekend is op welke poort de bestemming te bereiken is, het bericht op alle uitgaande poorten wegsturen.

Een switch mét IGMP snooping lost dit probleem elegant op, door alle IGMP berichten te onderscheppen en de inhoud te bekijken. Als een deelnemer lid wil worden van een multicast groep, dan dient een "IGMP Report Group" bericht naar de gewenste server gestuurd te worden. In dit bericht is uiteraard het afzender MAC-adres gegeven. De switch weet ook op welke poort dit bericht ontvangen is, én van welke multicast groep (XX YY ZZ) de deelnemer lid geworden is. Zal de switch nu multicast-netwerkberichten ontvangen, dan kan meteen bepaald worden op welke poorten er deelnemers zijn aangesloten die er interesse in hebben (= lid zijn van de multicast groep voor wie het netwerkbericht bestemd is).

Alleen op die poorten zal het netwerkbericht doorgestuurd worden, en op de andere poorten (uiteraard) niet. De switch zal in de switchtabel onthouden dat er een deelnemer op een bepaalde poort is die interesse heeft in alle multicasts voor het opgegeven GDA.

Het omgekeerde gebeurt als een apparaat zich wil afmelden uit een multicast groep. Hiertoe dient een "ICMP Leave Group" bericht naar de gewenste server gestuurd te worden. Ook dit bericht zal door de switch onderschept worden. In tegenstelling tot het aanmelden is het niet 1-2-3 mogelijk om te stoppen met het doorsturen van multicast verkeer naar de betreffende poort. Het is immers mogelijk dat op de poort een andere switch is aangesloten, en dat één apparaat op de andere switch de "ICMP Leave Group" verstuurd heeft. De switch kan dus alleen stoppen met doorsturen als ook de laatste deelnemer zich afgemeld heeft.

Switch opstarttijd
Nadat een switch aangezet of gereset is, zal enige tijd nodig zijn om op te starten. Hoelang dit duurt hangt o.a. van het feit of STP (Spanning Tree Protocol) wel of niet ondersteund is. Is STP niet nodig, dan hoeft het opstarten hooguit een seconde te kosten. Als STP wel nodig is, dan moet eerst het STP protocol afgerond zijn, omdat de nieuwe bekabelingstopologie van het netwerk uitgerekend moet worden; deze kan immers gewijzigd zijn nu deze switch "in de lucht" komt. Dit kan in totaal 30..60 seconden kosten.

Switch firmware update
Een switch kan intern voorzien zijn van een aanzienlijke hoeveelheid software. Deze software is niet alleen nodig om de interne hardware van de switch te besturen, maar ook voor het uitvoeren van diverse netwerkprotocollen, zoals SNMP (Simple Network Management Protocol) voor netwerkbeheer, Telnet voor remote login, (Fast) STP (Spanning Tree Protocol), etc. Een leverancier kan met een bepaalde regelmaat, of na ontdekking van fouten, nieuwe software ter beschikking stellen (let wel: niet alle leveranciers doen dit). Deze kan vaak via het netwerk zelf geladen worden. Nadat dit gebeurd is, zal de switch opnieuw opstarten, en wordt de nieuwe software actief.

5.6 Aansluiting van apparatuur

Op 10BaseT en 100BaseTX Ethernet wordt de RJ45 connector gebruikt, welke in tegenstelling tot de coax-connectoren van 10Base2 en 10Base5 niet over een bidirectioneel transmissiekanaal beschikt. Integendeel, van de 8 pinnen van de RJ45 worden er twee voor "transmit" (TX+ en TX-) en twee voor "receive" (RX+ en RX-) gebruikt.

Een verbinding tussen twee Ethernet-apparaten kan dan uiteraard alleen tot stand komen als de TX+ aan de RX+, en de TX- aan de RX- wordt gekoppeld.

De signalen TX+, TX-, RD+ en RD- zijn altijd te vinden op pinnen 1, 2, 3 en 6 van de RJ45 connector. Echter, de exacte locatie verschilt tussen "gewone" Ethernet-apparaten enerzijds, en hub / switches anderzijds. Dit geeft dan de volgende bekabelingsmogelijkheden:

- Een Ethernet-apparaat aan een hub / switch (zie hieronder: "A");
- Twee Ethernet-apparaten rechtstreeks aan elkaar (zie "B");
- Twee hubs / switches aan elkaar (zie "C").

A. Een Ethernet-apparaat aan een hub / switch.
Deze situatie zal in de praktijk het meest voorkomen. Op beide apparaten zijn de Ethernet-signalen als volgt op de RJ45 aangesloten:

Ethernet-deelnemer		Hub / Switch	
Pin 1	RD+	Pin 1	TD+
Pin 2	RD-	Pin 2	TD-
Pin 3	TD+	Pin 3	RD+
Pin 6	TD-	Pin 6	RD-

Een "1:1" kabel is in deze omstandigheden dus voldoende om communicatie mogelijk te maken.

B. Twee Ethernet-apparaten rechtstreeks aan elkaar.
Het is mogelijk om twee Ethernet-apparaten rechtstreeks aan elkaar aan te sluiten, zonder gebruik te maken van een hub of switch. Men kan dan wel geen netwerk maken met meer dan 2 deelnemers, maar soms is dit geen probleem – het kan een aanzienlijk financieel voordeel opleveren als geen hub of switch aangeschaft hoeft te worden. Op beide apparaten zijn de Ethernet-signalen als volgt op hun RJ45 aangesloten:

Ethernet-deelnemer 1		Ethernet-deelnemer 2	
Pin 1	RD+	Pin 1	RD+
Pin 2	RD-	Pin 2	RD-
Pin 3	TD+	Pin 3	TD+
Pin 6	TD-	Pin 6	TD-

Het is dus niet mogelijk om met een 1:1 kabel beide apparaten aan elkaar te koppelen: pin 1 moet immers naar pin 3 van de andere partij. Men krijgt dan een "kruising" (crossing) van de signaalparen in de kabel, dit levert een zgn. "kruiskabel" (cross-over cable) op. Het is altijd verstandig dit type kabels goed te merken, omdat men anders (later) soms lang zit te zoeken naar een oorzaak van communicatieproblemen.

C. Twee hubs / switches aan elkaar.

Soms is het nodig om twee hubs of switches aan elkaar aan te sluiten, bijvoorbeeld als het aantal poorten te klein is, of als men een hiërarchie in een netwerkstructuur aan wil brengen. Op beide hubs / switches zijn de Ethernet-signalen als volgt op de RJ45 aangesloten:

Hub / Switch 1		Hub / Switch 2	
Pin 1	TD+	Pin 1	TD+
Pin 2	TD-	Pin 2	TD-
Pin 3	RD+	Pin 3	RD+
Pin 6	RD-	Pin 6	RD-

Ook hier is dus een "kruiskabel" nodig om communicatie mogelijk te maken.

Veel leveranciers voorzien vaak al in de wens van gebruikers om hubs / switches aan elkaar aan te sluiten. Omdat kruiskabels op zich lastig zijn, wordt vaak één poort van een hub / switch als "uplink" poort ter beschikking gesteld. De Ethernet-signalen op de RJ45 van de uplink poort zijn dan omgedraaid. Er is dan geen kruiskabel nodig, maar er kan een gewone 1:1 kabel gebruikt worden.

Hub / Switch 1 uplink poort		Hub / Switch 2	
Pin 1	RD+	Pin 1	TD+
Pin 2	RD-	Pin 2	TD-
Pin 3	TD+	Pin 3	RD+
Pin 6	TD-	Pin 6	RD-

Let er wel op dat de uplink poort vaak geen extra poort is, maar enkel een extra connector voor een andere poort (vaak de hoogst of laagst genummerde poort). Men heeft dan wel twee connectoren ter beschikking, maar er mag er maar één van gebruikt worden, omdat intern dezelfde elektronica aangesloten is.

MDI / MDI-X

Er zijn ook leveranciers die geen extra connector gebruiken, maar een schakelaar. Men kan dan kiezen of men een poort als "normaal" of als "uplink" wil gebruiken. Deze schakelaar wordt ook wel aangeduid met de term "MDI / MDI-X" (Medium Dependent Interface normaal of "crossed").

Indien een hub / switch maar één uplink poort heeft, maar men wil toch een aansluiting maken naar twee of meer andere hubs / switches, dan zit er niets anders op dan toch weer met kruiskabels te werken, die dan op de gewone poorten aangesloten moeten worden.

Auto-crossing

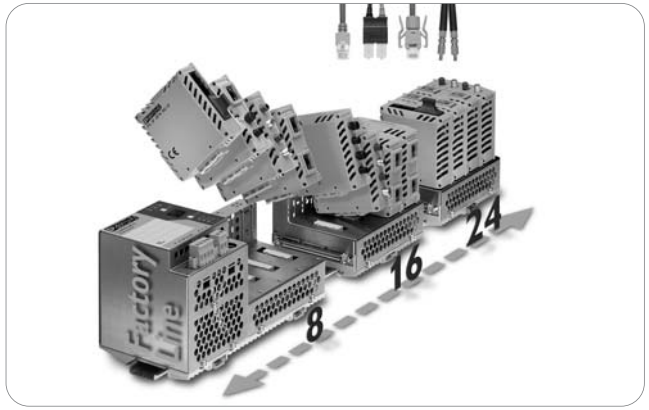
Sommige leveranciers bieden ook een "auto-crossing" functie op hun switches aan. Dit houdt in dat de switch zelf uitzoekt op welke pinnen de TX+ en TX- signalen van de andere deelnemer ontvangen worden, en zal zichzelf daarop instellen. Het is dan niet meer nodig om een MDI/MDI-X schakelaar te bedienen. Ook hoeft de switch zelf geen onderscheid te maken tussen 'gewone' en 'uplink' poorten. Voor de gebruiker biedt dit vooral gemak.

5.7 Industriële switches

Industriële switches zijn in principe (qua werking) 'gewone' switches, maar zoals de naam al aangeeft hebben ze een aantal speciale eigenschappen:

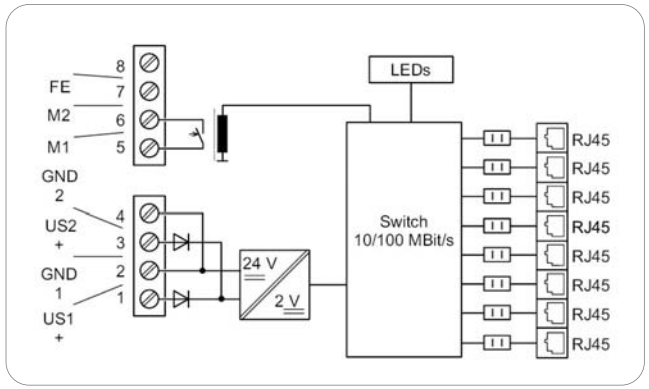
- Steviger behuizing;
- Gebruik van speciale connectoren;
- Voeding 24V, vaak ook redundant;
- Minder poorten (4/8/12/16);
- Hogere MTBF;
- Uitgebreider temperatuurbereik (bv. –40...+60 graden Celsius);
- Geen noodzaak voor geforceerder koeling (ventilator);
- Beter bestand tegen EMC / trillingen / schokken etc.;
- Potentiaalvrij contact voor doorgeven events.

Industriële switches worden door zeer veel bedrijven geleverd. Volgens een in 2003 verschenen rapport van ARC (Automation Research Council) zal de komende 5 jaar de markt voor industrieel Ethernet jaarlijks zo'n 84% groeien. Met zulke optimistische voorspellingen is het niet verwonderlijk dat steeds meer bedrijven deze markt betreden.



Figuur 5-10: De modulaire "MMS" switch van Phoenix.

Indien een netwerk een hoge beschikbaarheid moet hebben, is de voeding van een industriële switch van belang – als de voeding uitvalt, stopt de switch en daarmee dus ook het netwerk. Diverse leveranciers bieden daarom de mogelijkheid hun apparatuur van een dubbele voeding te voorzien. Meestal wordt een 24V voeding voorzien, omdat deze spanning in veel industriële toepassingen gebruikelijk is. Figuur 5-11 geeft een blokschema van de interne opbouw van een switch van Phoenix. Via een diode-schakeling kunnen twee voedingen aangesloten worden.



Figuur 5-11: Blokschema van een Phoenix switch. De aansluiting voor de redundante voeding (US1 en US2) zit linksonder.

5.8 Spanning tree

Door gebruik te maken van hubs en switches mogen allerhande bekabelingsstructuren ontstaan, behalve: een ring. Indien men dit (soms per ongeluk) doet zal het netwerk ophouden met functioneren totdat de ring weer verbroken wordt.

Uiteraard kan een ring soms per ongeluk ontstaan, meestal in die gevallen waarbij de documentatie over de netwerkbekabeling en aangesloten apparatuur niet up-to-date is. In de meeste gevallen zal men echter een ring juist expres maken, vanwege een zeer belangrijk voordeel die dit biedt: redundantie. Een netwerkbericht kan dan óf linksom, óf rechtsom zijn eindbestemming bereiken. Een kabelbreuk in de ring of de uitval een hub / switch is dan niet fataal.

Maar het aanleggen van een ring met standaard hubs of switches is niet mogelijk. Indien men een ring bouwt met hubs, dan zal dit resulteren in het oneindig doorsturen van hetzelfde netwerkbericht. Elk netwerkbericht wordt immers uitgestuurd op alle andere poorten, komt bij de volgende hub aan, waarna dit zich herhaalt, en na enige tijd is het netwerkbericht terug bij 'zijn' oorspronkelijke hub, waarna het geheel zich nogmaals herhaalt. Het aantal collisions zal fenomenaal hoog worden, de netwerkbelasting stijgt ook flink, maar in concreto gebeurt er niets meer op dit netwerk totdat de ring onderbroken wordt.

Indien men een ring bouwt met switches, dan zal (direct nadat de spanning aangezet is) elke switch zich als een hub gedragen omdat de switchtabel nog leeg is. Ook hier ontstaat een continue herhaling van het eerste netwerkbericht. Er zijn wel scenario's te bedenken waarbij het netwerk uiteindelijk toch in een stabiele situatie terechtkomt, maar dit is meestal niet te realiseren met standaard software en is daarom geen optie.

Indien men toch vast wil houden aan de ringstructuur in de bekabeling, maar wel op een functionerend systeem uit wil komen, dan moet de ring onderbroken worden. Dit kan op de volgende manieren:

- Via het handmatig onderbreken van de ring, eenvoudigweg door het ver wijderen van een van de kabels in de ring uit de poort op de hub / switch. Indien de ring elders onderbroken raakt, dan steekt men deze kabel weer in de oorspronkelijke poort terug. Het voordeel van deze manier van werken is dat er geen software voor nodig is, maar het nadeel is de lange reactietijd (minuten...uren) voordat iemand in de gaten heeft wat er aan de hand is,

en tevens omdat iemand fysiek een actie uit moet voeren.
Dit is derhalve niet geschikt voor industriële systemen.

- Via het softwarematig onderbreken van de ring, door op een hub / switch één van de poorten waarop een kabel uit de ring is aangesloten uit te schakelen. Sommige leveranciers bieden deze functionaliteit op hun apparatuur aan. Het voordeel van deze manier van werken is dat er geen software voor nodig is, maar het nadeel is de handmatige actie (die wel via het netwerk zelf uitgevoerd kan worden) en de tijd die nodig is voordat iemand in de gaten heeft wat er aan de hand is.

Indien men gebruik maakt van switches, dan is er nog een derde mogelijkheid:

- Via het softwarematig onderbreken van de ring, door in alle switches gebruik te maken van STP. Dit netwerkprotocol kan zelf uitzoeken waar in het netwerk een ring aanwezig is, en deze wordt dan softwarematig onderbroken c.q. weer in bedrijf genomen indien dit nodig mocht zijn. Geen menselijke interventie is verder nodig.

Switches die dit kunnen hebben het zgn. "Spanning Tree Protocol " (STP) ingebouwd (heeft niets te maken met Shielded Twisted Pair bekabeling!). STP is oorspronkelijk bedacht door Radia Perlman van Sun. De werking van STP is vastgelegd in de norm IEEE 802.1d. Dit is geen Ethernet-specifieke norm, omdat STP ook kan werken op andere types netwerken dan Ethernet; maar in de praktijk zal men het toch vaak gecombineerd zien met Ethernet.

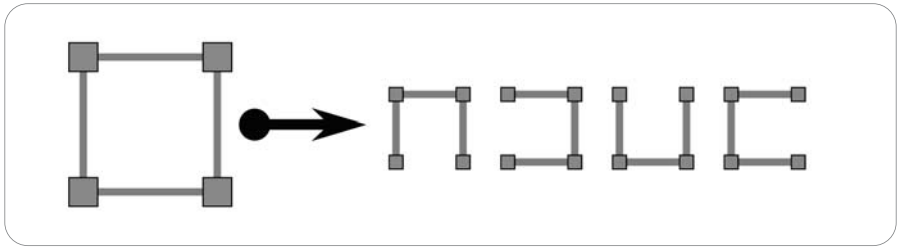
Werking van STP

De switches op een netwerk die STP ondersteunen communiceren met elkaar, en bouwen op deze manier kennis op over wie aan wie gekoppeld is. Het wordt dan ook duidelijk welke redundante bekabelingsmogelijkheden aanwezig zijn. Dit hoeft niet altijd perse een ring-structuur te zijn; men kan b.v. ook twee parallelle kabels aanleggen tussen twee switches.

Na enige tijd (gemiddeld 30..50 seconden) zullen alle switches weten hoe het netwerk in elkaar zit, en dus ook waar alle redundantie aanwezig is. Deze lange tijd is het resultaat van interne verwerkingsstappen van STP, dat ontwikkeld is in een tijd dat netwerken nog niet de snelheid hadden zoals nu, en er dus ook niet van kon worden uitgegaan dat sommige verwerkingsstappen snel uitgevoerd zouden kunnen worden. Inmiddels is de snelheid van netwerken fors gestegen, en sommige leveranciers bieden dan ook STP-implementaties aan waarvan de interne verwerkingsstappen sneller kunnen worden uitgevoerd.

Om het netwerk operationeel te krijgen moet nu besloten worden welke van de redundante bekabelingsstukken niet gebruikt gaan worden. Er ontstaat dan, in wiskundige termen, een "spanning tree", waarbij de 'bladeren' de switches zijn, en de 'takken' gevormd worden door de bekabeling. Iedereen op het netwerk is dan bereikbaar op precies één manier. De analogie met een boom is op dit moment nog niet helemaal correct, want een echte boom heeft alleen maar vertakkingen en geen lussen en ringen in zijn takken.

Een voorbeeld van een mogelijke configuratie is links in figuur 5-12 gegeven; een netwerk met 4 switches en 4 kabels daartussen. Daar zit een redundant pad in, via de lus. Na afloop van het STP protocol is bekend dat er 4 mogelijke configuraties zijn waarmee het netwerk kan functioneren (rechts). Daarbij moet dus één verbinding uitgeschakeld worden. In elk geval, iedereen kan nog steeds met iedereen communiceren.

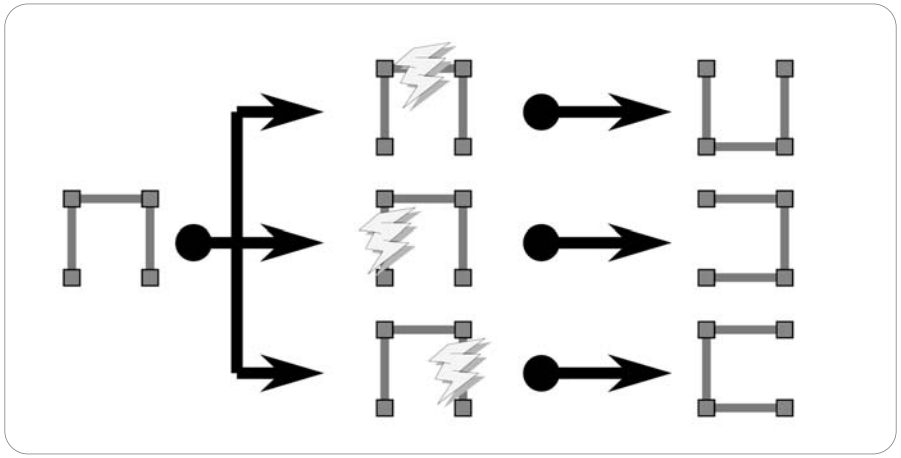


Figuur 5-12: een netwerk met een ring opgebouwd uit 4 switches (links). Rechts zijn de vier mogelijke "spanning trees" weergegeven; de 4e verbinding is er nog wel maar niet actief in gebruik.

Als gebruiker kan men nog beïnvloeden wat de meest gewenste uitkomst zal zijn. Bijvoorbeeld, men kan instellen wat de snelheid van elke netwerkverbinding is, en STP kiest dan die configuratie uit de 4 mogelijkheden die de beste performance biedt. Zo kan men dus een langzamere (of duurdere) netwerkverbinding als back-up laten bestaan, in plaats van een snellere netwerkverbinding als back-up te kiezen.

Als een netwerk operationeel is en er een van de mogelijke spanning trees gekozen is, hoeft er verder niets meer te gebeuren totdat een van de netwerkverbindingen uitvalt. Er moet dan een nieuwe spanning tree gemaakt worden zodat iedereen weer met iedereen kan communiceren.

Een voorbeeld hiervan is het eerder getoonde netwerk met 4 switches (figuur 5-12). Er zijn drie verbindingen tussen deze 4 switches, dus er kunnen ook 3 netwerkverbindingen uitvallen. Per mogelijke fout kan het spanning tree algoritme een nieuwe configuratie vinden waarmee alle communicatie hersteld kan worden. Dat kan dan met activering van de voorheen passieve netwerkverbinding (figuur 5-13).



Figuur 5-13: Na uitval van een verbinding zal de spanning tree opnieuw opgebouwd worden door het inbedrijfnemen van het ongebruikte kabelsegment. Afhankelijk van waar de storing optrad zal het netwerk anders geherconfigureerd worden.

Hierboven is uitgegaan van een ringstructuur. In principe kan STP echter elke willekeurige bekabelingstopologie aan.

Snelheid van STP

De snelheid waarmee STP reageert op wijzigingen in de netwerktopologie (meestal meer dan 30 seconden) voldoet niet (meer); zelfs niet in administratieve omgevingen, laat staan voor een industriële toepassing. De norm IEEE 802.1w beschrijft een snellere variant van STP, "Fast STP", "Rapid Reconfiguration STP" of "Rapid STP" (RSTP) genaamd. Deze is in om onder normale omstandigheden binnen 1..10 seconden te kunnen herconfigureren.

Echter, zelfs 1 seconde is nog steeds een eeuwigheid voor veel industriële applicaties. Daarom komen leveranciers met switches/bridges die nóg sneller kunnen reconfigureren.

Nog snellere herconfiguratie

Diverse leveranciers bieden STP-varianten aan die aanzienlijk sneller zijn dan de 'standaard' STP of Fast STP. Men moet zich dan vaak wel beperken tot een ring, dit in tegenstelling tot STP dat elke mogelijke bekabelingstopologie aankan. Ook moeten alle switches bij dezelfde leverancier gekocht worden.

Met "HiperRing" wordt een omschakeltijd van ca. 500 msec geclaimd. Deze tijd is vrijwel onafhankelijk van het aantal switches en de lengte aan bekabeling. Dit staat in tegenstelling tot RSTP welke bij een klein aantal switches (< 4) sneller omschakelt dan HiperRing, maar bij een groter aantal switches langzamer omschakelt; bovendien is de omschakeltijd niet voorspelbaar (deterministisch). Verder vermeldingswaardig is dat tijdens het omschakelen van RSTP er een kans bestaat dat netwerkberichten in een andere volgorde worden afgeleverd dan dat ze verzonden zijn, en dat ook een kans bestaat op duplicering van netwerkberichten. Bij veel netwerkprotocollen is dit geen probleem (b.v. bij gebruik van TCP) maar bij andere protocollen kan dit wel een probleem opleveren.

Met het "TurboRing" protocol wordt een herconfiguratietijd geclaimd van minder dan 300 msec bij 120 deelnemers. De claim voor de snelste herconfiguratie wordt afgegeven voor "FRNT" (Fast Reconfiguration of Network Topology), waarvoor tijden van 30 msec opgegeven worden. Het is bij dit type claims altijd zeer de vraag onder welke condities dit allemaal gegarandeerd wordt; publiekelijk verkrijgbare informatie is schaars.

Alternatief

Indien men niet voor alle bekabelingstrajecten redundante bekabeling nodig heeft, kan het soms zinvol zijn switches te gebruiken die "port trunking" ondersteunen. Omdat de switches e.e.a. zelf uitzoeken, is geen extra netwerkprotocol (zoals STP) nodig, en dit kan een belangrijk snelheidsvoordeel bieden.

Nadeel van STP

Een nadeel van (Fast) STP is dat sommige bekabeling en sommige poorten op switches niet gebruikt kunnen worden, omdat ze 'stand-by' moeten staan. Dit is uiteraard een extra kostenpost. Ook hier biedt "port trunking" een voordeel, omdat alle capaciteit van bekabeling en randapparatuur gebruikt wordt.

5.9 Gebruik van switches

Veel leveranciers van switches bevelen deze graag aan[1] boven het gebruik van hubs. Dit is echter technisch niet altijd zinvol; in een aantal omstandigheden worden de voordelen van een switch teniet gedaan door het gedrag van het (software) applicatieprotocol. Enkele voorbeelden:

• Een switch kan een bandbreedteverhoging bieden vanwege de mogelijkheid om full-duplex te werken (zenden en ontvangen tegelijk). Bij protocollen die master/slave gedrag vertonen (zenden óf ontvangen) heeft full-duplex dus geen nut.

• Een switch staat toe dat meerdere transmissies tegelijkertijd actief kunnen zijn, en dit biedt ook een bandbreedteverhoging. Bij applicaties waarbij één centrale deelnemer (besturing) met alle andere deelnemers communiceert, en deze verder onderling niet, kan er op elk moment toch maar één transmissie naar de centrale besturing actief zijn.

• Een switch stuurt een netwerkbericht enkel door via de poort waarop de bestemming is aangesloten. Bij netwerkprotocollen die werken volgens het publish/subscribe (of producent/consument) model, waarbij alle netwerkberichten naar iedereen moeten worden gestuurd, gaat een switch zich precies zo gedragen als een hub, tenzij deze voorzien is van VLAN-functionaliteit.

Het is dus afhankelijk van het gebruikte applicatieprotocol (zie ook hoofdstuk 6 voor een opsomming hiervan) en de werking van de eigen applicatie of een switch wel zoveel voordelen biedt boven een hub. Uiteraard kan men zich afvragen of het dan toch niet beter is een switch te kopen, omdat dit enige ruimte biedt naar de toekomst toe. Maar het is altijd mogelijk om een hub te vervangen door een switch indien dit echt nodig is; er zijn verder geen software- of protocolconsequenties.

[1] Switches zijn soms nog wel duurder dan hubs, alhoewel het prijsverschil de afgelopen drie jaar heel snel kleiner geworden is. Intussen begint het aanbod aan hubs bij sommige leveranciers al wel minder te worden.

6. Applicatieprotocollen

Zoals eerder besproken is Ethernet slechts een klein deel van een compleet netwerk, namelijk: 1) de bekabeling, en 2) de mogelijkheid om berichten te versturen en te ontvangen. Hogere niveaus van standaardisatie zijn nodig vóórdat twee apparaten met elkaar kunnen communiceren. Op dit gebied lopen er al enkele jaren zeer veel ontwikkelingen, onder andere bij:

- ProfiNet van de Profibus gebruikersvereniging;
- Ethernet/IP van de Devicenet gebruikersvereniging;
- Powerlink van de Powerlink standaardisatiegroep;
- IDA van de Modbus/IDA-group;
- Modbus/TCP van de Modbus/IDA-group;
- FF HSE van de Foundation Fieldbus gebruikersvereniging;
- Sercos-III van de Sercos gebruikersvereniging;

Daarnaast zijn er nog de nodige leverancierssspecifieke oplossingen, o.a.:

- SRTP (Service Request Transfer Protocol);
- EGD (Ethernet Global Data);
- Safe Ethernet;
- RTEthernet (Real Time Ethernet);
- NDDS (Network Distributed Data Services);
- Etc.

Al deze systemen claimen "Ethernet" te zijn, maar dus steeds met een eigen invulling van de hogere protocolniveaus (lees: OSI-lagen 3 t/m 7), en soms ook wel met een eigen invulling op de lagere OSI-niveaus. Men is dus niet compatibel met elkaar. Hoogstens kunnen twee apparaten met verschillende protocollen op (fysiek) hetzelfde Ethernet worden aangesloten, maar communiceren met elkaar is onmogelijk. Soms is het zelfs onmogelijk om twee protocollen tegelijk te gebruiken. Dit aspect wordt door leveranciers vaak verzwegen, omdat het een acceptatie van Ethernet in de weg zou kunnen staan.

In de volgende secties worden enkele van de meer bekende systemen besproken. Paragraaf 6.9 behandelt tenslotte de kloksynchronisatieprotocollen, in het bijzonder IEEE-1588. Alhoewel dit niet specifiek een Ethernet-protocol is, wordt het wel gebruikt als onderdeel van Powerlink en Ethernet/IP, een variant ervan wordt gebruikt in JetSync, en vergelijkbare technieken in andere protocollen.

6.1 Modbus/TCP

Van Modbus/TCP wordt wel gezegd dat het op dit moment (2004) het meest gebruikte industrieel Ethernet protocol is. De redenen hiervoor zijn dat Modbus/TCP al vrij lang beschikbaar is (sinds 1999), eenvoudig te implementeren is (ca. 1 week werk voor iemand die goed thuis is op TCP/IP), de specificatie voor iedereen vrij ter beschikking staat (*zie www.modbus.org*), en daarnaast ook snel te leren is (in ca. 1 dag).

Zoals de naam al aangeeft is Modbus/TCP lid van de Modbus-familie van protocollen (Modbus/ASCII, Modbus/RTU, Modbus+, J-Bus). Er zijn dan ook grote overeenkomsten te vinden. Sterker nog, in feite is Modbus/TCP voor 95% gelijk aan Modbus/RTU; de resterende 5% betreft aanpassingen om met TCP/IP te kunnen werken, en het plaatsen van enkele puntjes-op-de-i in de Modbus/RTU specificatie. In de loop der jaren is gebleken dat deze op een aantal punten vaag was, en dit leidde tot verschillende implementaties, hetgeen in de praktijk wel eens tot (interoperabiliteits-)problemen leidde bij gebruikers.

Modbus/TCP maakt gebruik van het bekende TCP/IP protocol dat de feitelijke transmissie van netwerkberichten verzorgt. Meestal zal men dit via een Ethernet uitvoeren. Een voordeel van TCP/IP is dat deze niet persé via Ethernet hoeven te werken; men kan ook TCP/IP uitvoeren via een seriële lijn (RS232), ISDN, kabel, mobieltje (GSM/UMTS/GPRS), etc. Modbus/TCP kan dus óók van deze media gebruikmaken!

De voordelen van het gebruik van Modbus/TCP zijn:

- Bestaande berichtstructuur is gehandhaafd;
- Snellere communicatie mogelijk;
- Master/slave beperking vervalt.

Een nadeel:

- Broadcasts zijn niet mogelijk.

De bestaande Modbus/RTU berichtstructuur is gehandhaafd gebleven, maar een nieuwe 6-byte header is toegevoegd. Deze is nodig om de koppeling naar TCP/IP goed uit te voeren, en blijft verder onzichtbaar voor de applicatie. Verder is het niet meer nodig om elk byte een pariteitsbit te geven, en het hele bericht een (16-bits) CRC. TCP/IP heeft zelf al een 32-bits checksum, en Ethernet nog een 32-bits CRC, en het is dus niet nodig om nog

een derde niveau van gegevensbeveiliging toe te voegen. Als dit op het laagste netwerkniveau wordt uitgevoerd, zal het automatisch door hardware worden gedaan en hoeft het dus niet via software berekend te worden. Dat is voordelig voor de snelheid van het geheel, want het berekenen van een CRC is zeer tijdsintensief.

Normaliter wordt Modbus altijd bekabeld op basis van RS232 of RS485, en de snelheid is meestal beperkt tot bitrates niet hoger dan 38.4 Kbit/s, eenvoudigweg omdat een seriële poort niet sneller kan. Het gebruik van een 10 of 100 Mbit/s Ethernet geeft dus al een zeer forse snelheidsverhoging.

Van origine is Modbus een master/slave netwerk, zo opgedrongen door de manier van werken op RS485. Het werken volgens een master/slave structuur, waarbij de master alléén met de slaves, maar de slaves onderling *niet* mogen communiceren, stelt dus duidelijke beperkingen. Op TCP/IP is dit niet meer nodig, en Modbus/TCP kan zo dus een multi-master netwerk worden, waarbij iedereen aan iedereen commando's kan sturen (edoch dit is niet altijd mogelijk omdat er bij eenvoudige implementaties geen rekening mee gehouden is). Dit maakt Modbus/TCP opeens veel geschikter voor moderne gedistribueerde applicaties, waarbij elke besturing gegevens uitwisselt met een of meerdere andere.

Het gebruik van TCP zorgt er wel voor dat broadcasts (van de master naar alle slaves) niet meer kunnen. TCP is immers een 'point-to-point' protocol, en een bericht komt dus altijd maar bij precies één slave aan. Een oplossing zou nog kunnen zijn door hetzelfde bericht n maal te kopiëren en het dan apart naar alle n slaves te sturen, maar dat geeft wel een n maal hogere netwerkbelasting, en de gelijktijdige aankomst van het bericht op alle slaves is daarmee ook niet gegarandeerd.

In Modbus-kringen wordt daarom wel gesproken over een broadcast-versie die op basis van UDP (het simpele broertje van TCP) kan werken, maar dat is (nog lang) niet tot standaard verheven en men moet er ook niet van uit gaan dat dit al ergens te koop is.

Medio 2002 is Modbus/TCP aangemeld bij de Internet-autoriteit (IETF) als voorstel voor een nieuwe Internet-standaard. Het doel hiervan is wat onduidelijk, want wat moet het Internet nu met Modbus/TCP? Waarschijnlijk is het enige doel om netwerkbeheerders bekend te maken met het protocol, zodat men niet bevreesd is om Modbus/TCP netwerkverkeer toe te laten; dit is zeer interessant voor remote monitoring en –diagnostics gebruik. Om dit Internet-technisch mogelijk te maken, moet op firewalls e.d. een TCP/IP port opengezet worden (anders wordt alle Modbus/TCP verkeer geblokkeerd).

Volgens de specificatie is hiervoor poort 502 gereserveerd, en dit is iets wat netwerkbeheerders dus moeten weten. Inmiddels (medio 2003) is gebleken dat Modbus/TCP *niet* als Internet-standaard is geaccepteerd door de IETF, omdat niemand moeite heeft gedaan om genoeg "ja" stemmen te verzamelen. Waarschijnlijk zal een nieuwe poging ondernomen worden.

6.2 ProfiNet

ProfiNet is de Ethernet-variant uit de Profibus-familie. Eigenlijk is het woord "variant" niet helemaal van toepassing, want ProfiNet lijkt in het geheel niet op de Profibus-protocollen (FMS, DP, PA) zoals we die al 15 jaar kennen.

De ontwikkeling van ProfiNet is in 1999 in gang gezet, en volgens de eerste planning zou in 2003 het systeem op de markt komen. Naderhand is besloten om tussenversies op de markt te brengen, zo kennen we nu ProfiNet versies 1, 2 en 3. Als laatste is de specificatie voor remote I/O gemaakt. De hiervoor benodigde hardware is pas eind 2004 leverbaar; dan zal het zeker nog een jaar duren voordat producten hiermee op de markt gebracht zijn (en nog langer voordat alle kinderziektes er uit zijn gehaald). In de tussentijd kan men al gebruik maken van ProfiNet V2.

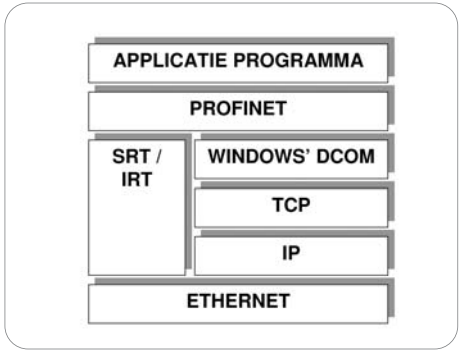
ProfiNet versies

In ProfiNet 1 ging men nog van de gedachte uit dat alle Ethernet-communicatie via het TCP/IP protocol uitgevoerd zou moeten worden. Technisch kan dit ook wel, alleen moeten geen hoge eisen gesteld worden aan de snelheid: cyclustijden van tientallen milliseconden zijn mogelijk, met een zeer grote variatie. Dit komt vanwege de manier waarop TCP/IP protocolstacks opgebouwd zijn; in het algemeen is geen enkele aandacht besteed aan real-time eigenschappen. Snelheid is niet eens het belangrijkste in real-time systemen, maar wel de voorspelbaarheid ('determinisme'): is te *garanderen* dat het netwerk altijd binnen een bepaalde tijd een taak kan afhandelen?

In ProfiNet 2 is daarom TCP/IP deels verlaten; de real-time communicatie wordt nu afgewikkeld door een eigen protocol (parallel aan TCP/IP) dat rechtstreeks op Ethernet toegrijpt. Dit heet "SRT" (Soft Realtime), en wordt gebruikt voor alle cyclische I/O, alarmen en hoge-snelheids communicatie. Cyclustijden in het bereik van 5..10 msec zijn haalbaar. TCP/IP wordt verder gebruikt tijdens het opstarten, configurering, download van programmatuur, webdiensten, email, etc. Een extra voordeel is nog dat SRT niet afhankelijk is van Microsoft's DCOM (Distributed Component Object Model) technologie, welke al heel lang

in Windows aanwezig is, maar niet in veel andere operating systemen. Op het gebruik van DCOM is altijd veel commentaar geweest, vanwege de afhankelijkheid van Microsoft. Vanaf ProfiNet 2 is het ook mogelijk om voor de communicatie tussen twee ProfiNet-deelnemers een ander protocol dan DCOM te gebruiken.

ProfiNet 2.0 is al aanzienlijk sneller dan 1.2, maar op zich niet spectaculair snel ondanks de 100 Mbit/s van Ethernet; een 'gewoon' Profibus/DP netwerk biedt een vergelijkbare performance, hoewel de bitrate hiervan veel lager is (maar de efficiency in gebruik van het netwerk evenredig hoger). ProfiNet 2 is in 2002 op de Hannover Messe aangekondigd, en staat sinds juni 2003 ter beschikking aan leden van de Profibus gebruikersvereniging. De eerste producten zijn begin 2004 op de markt gekomen.



Figuur 6-1: De opbouw van de ProfiNet protocolstack. Let op de aanwezigheid van Microsoft's DCOM, en het eigen remote I/O protocol dat geen gebruik (meer) maakt van TCP/IP.

In een poging om ProfiNet ook bruikbaar te maken voor high-end motion-toepassingen, zoals in gebruik in drukpersen, spuitgietmachines, verpakkingsmachines, CNC besturingen etc. is het niet mogelijk om het netwerkprotocol geheel in software uit te voeren. Dit soort toepassingen stellen twee speciale eisen aan een industrieel netwerk: zeer hoge snelheid, én een zeer constante snelheid. De zgn. "jitter", de variatie in snelheid, moet dan ook zeer klein zijn. Profibus noemt dit "Isochronous Real Time" (IRT), en dit is de basis van ProfiNet 3.0; men claimt dat een jitter < 1 microseconde te realiseren is.
Eén manier om dit te doen is het gebruik van zo weinig mogelijk software; een speciaal controller-IC zal de IRT (en SRT) protocollen uit gaan voeren. Dit IC zal in de tweede helft van 2004 op de markt komen.

Met dit IC wordt het ook mogelijk om een ProfiNet systeem volgens een bustopologie op te bouwen; elke IC is in feite een 4-poorts switch (ook wel "machine distributor" genoemd in ProfiNet documentatie). Het is dan niet meer nodig om een losse switch aan te schaffen. Dit geeft ProfiNet een belangrijk financieel voordeel in vergelijking met andere systemen.

Intern zal ProfiNet een netwerkcyclus uitvoeren die steeds uit twee delen bestaat: eerst wordt de real-time data getransporteerd, en daarna is er ruimte voor alle andere transmissies (bv. via TCP/IP). Deze cyclus is dus steeds even lang; mocht er meer tijd benodigd zijn voor de niet-real-time transmissies dan moet gewacht worden op een volgende netwerkcyclus. Deze manier van werken is overigens niet uniek voor ProfiNet, ook Powerlink en Sercos-III implementeren deze manier van netwerkgebruik.

Bekabeling
Voor ProfiNet wordt uitgegaan van 100 Mbit/s Ethernet-varianten, alhoewel uitdrukkelijk beschreven is dat gebruik van 10 Mbit/s Ethernet ook mogelijk is. De bekabeling dient uitgevoerd te worden met STP (Shielded Twisted Pair) of glasvezel (max. 2 km afstand voor multimode, 14 km voor single mode). De aanbevolen connector is de RJ45, waarvan tevens een IP67 variant van beschikbaar is. Ook kan de M12 connector gebruikt worden. Tenslotte is er een hybride connector bestaande uit een RJ45-deel en 4 extra pinnen welke ingezet kunnen worden voor voeding.

Systeemopbouw
In tegenstelling tot de eerdere varianten uit de Profibus-familie wordt in ProfiNet niet alleen naar het netwerkprotocol beschreven, maar ook de manier van programmeren van applicaties. Men programmeert hier niet meer door het heen-en-weer sturen van bits en bytes, maar door het aan elkaar koppelen van "componenten". Een component kan een bepaald stuk hardware zijn op het netwerk, maar ook een softwaremodule die een bepaalde taak doorrekent. Het maken van een applicatie bestaat dan feitelijk uit het op het scherm plaatsen van afbeeldingen van componenten, en deze aan elkaar te koppelen. In feite tekent men dan hoe de datastromen tussen componenten lopen. Een softwarepakket dat dit allemaal mogelijk maakt heet een "Interconnection Editor".

Deze editor 'weet' welke componenten er allemaal zijn omdat deze zijn opgeslagen in een bibliotheek. Deze wordt door de gebruiker gevuld met componentbeschrijvingen. Deze zijn opgeslagen in de zgn. "ProfiNet Component Description" (PCD) bestanden. De PCD bestanden moet men krijgen van de leveranciers van de modules op het netwerk.

Eventueel kan men ook zelf PCD bestanden maken via de "ProfiNet Component Editor", die door de Profibus gebruikersvereniging aan leden ter beschikking gesteld wordt.

Uiteindelijk heeft men een complete applicatie gemaakt waarin allerlei componenten aan elkaar gekoppeld zijn. Nu moet nog vastgelegd worden welke component waar op het netwerk te vinden is: de netwerkconfiguratie. Daarna kan elke component zijn eigen configuratie geladen krijgen, alsmede zijn eigen deel van de "Interconnection Data", waardoor hij weet door wie data wordt aangeleverd, en aan wie de eigen resultaten doorgestuurd moeten worden. De "engineering fase" van ProfiNet is nu afgerond, en de "run-time fase" begint. Deze functioneert dan anders tussen ProfiNet V1, V2 en V3 (zie hierboven).

Integratie
ProfiNet vindt op veel gebieden het wiel niet steeds opnieuw uit; besloten is om zoveel mogelijk van bestaande netwerkprotocollen gebruik te maken. Zo zagen we al het gebruik van DCOM en TCP/IP, maar ook DHCP en SNMP zijn geïntegreerd. Tevens is er ondersteuning voor HTTP, HTML en XML.

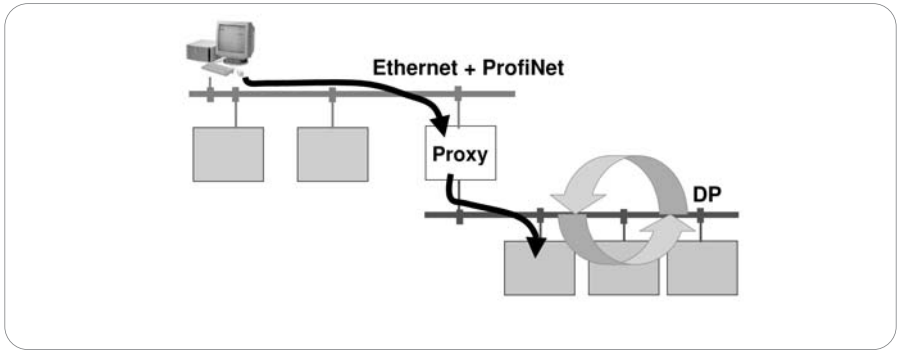
Het configureren van ProfiNet-deelnemers kan op twee verschillende manieren gebeuren: via het eigen "DCP" (Discovery and Basic Configuration Protocol), of via het bekende "DHCP" (Dynamic Host Configuration Protocol). DCP is verplicht voor alle ProfiNet-deelnemers, terwijl DHCP optioneel geïmplementeerd mag zijn.

SNMP (Simple Network Management Protocol) wordt gebruikt voor netwerkbeheer.
Het is mogelijk om ProfiNet-deelnemers te monitoren, diagnostische gegevens uit te lezen, en bedrijfsparameters in te stellen. Er is voor gekozen om alleen de SNMP-standaard parameters te ondersteunen. Alle ProfiNet-specifieke gegevens zijn echter niet via SNMP benaderbaar.

Web integratie is eenvoudig mogelijk door ondersteuning van HTTP (Hypertext Transport Protocol). Webpagina's kunnen dan worden opgevraagd en in HTML of XML teruggegeven, zodat met elke standaard browser gewerkt kan worden. Elke ProfiNet-deelnemer kan een webserver zijn, hoewel dit overigens niet verplicht is.

Koppelingen naar andere systemen
Een koppeling van ProfiNet naar OPC is vrij makkelijk te maken. De technologische basis van beide is namelijk Microsoft's DCOM-technologie. Een groot verschil is echter wel dat ProfiNet op basis van echte objecten werkt, terwijl OPC uitgaat van "tags" (namen) voor

velden. In principe kan elke ProfiNet-deelnemer als een OPC-server benaderd worden. Tevens is in ProfiNet ondersteuning voor OPC/DX ingebouwd; hiermee kan men communiceren met OPC-servers voor andere types netwerken. Anderzijds is het ook mogelijk om aan een willekeurige OPC-server een ProfiNet-aansluiting te geven. De OPC-server moet dan aan een speciale softwareconverter (een "OPC Objectizer") gekoppeld worden, welke van OPC naar ProfiNet converteert.



Figuur 6-2: De koppeling van ProfiNet en Profibus/DP wordt mogelijk door middel van een "proxy".

Uiteraard is het mogelijk om koppelingen te maken tussen ProfiNet en Profibus/DP. Dit is zowiezo gewenst om de invoering van ProfiNet in bestaande applicaties mogelijk te maken, hetgeen de acceptatie zal versnellen. Uiteraard kan niet rechtstreeks DP-apparatuur op Ethernet worden aangesloten, vanwege de verschillende fysieke interfaces: 100 Mbit/s full-duplex 4-draads Ethernet versus max. 12 Mbit/s half-duplex 2-draads RS485. Een conversiemodule is nodig (figuur 6-2). In ProfiNet-termen heet dit een "proxy" (plaatsvervanger). Men krijgt dan een 2-laags netwerkarchitectuur, met op het hoogste niveau ProfiNet en daaronder Profibus/DP. Het is mogelijk om meerdere proxies aan te sluiten. De proxy zelf fungeert als een normale DP-master, en communiceert met alle DP-slaves. Van buitenaf is dit echter niet zichtbaar. De proxy simuleert als het ware alsof hij meerdere ProfiNet-devices tegelijk is (nl. één per DP-slave). Via ProfiNet kan dan tegen een van de gesimuleerde DP-slaves worden gepraat. De proxy handelt dan de echte communicatie met de DP-slave af, en stuurt het antwoord terug.

Deze manier van werken is ook goed bruikbaar voor andere types bussystemen. Naar verwachting van de auteur zullen er hiervoor dan ook speciale proxies op de markt komen.

Phoenix Contact heeft begin 2004 besloten om Interbus te integreren met ProfiNet. Hiervoor is een aparte werkgroep opgericht die dit technisch vorm moet gaan geven. Ten tijde van publicatie waren hierover nog geen verdere details bekend.

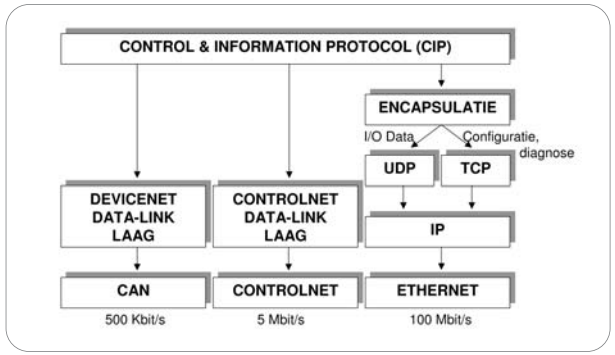
Documentatie

Documentatie over ProfiNet is te verkrijgen via *www.profibus.com*. Hier is enige algemene informatie over ProfiNet te vinden, alsmede de specificatie welke ook voor niet-leden van de Profibusvereniging te verkrijgen is. Het kunnen begrijpen van dit document vereist wel een diepgaande kennis van Microsoft's DCOM-technologie. Eenvoudiger te begrijpen is de "ProfiNet Technology & Application System Description" van 21 pagina's. Tenslotte is er de "ProfiNet Installation Guideline", welke regels geeft m.b.t. de installatie van een netwerk.

6.3 Ethernet/IP

Ethernet/IP (Industry Protocol) is een ontwikkeling van Rockwell / Allen-Bradley. Het vertoont zeer veel overeenkomsten met de twee andere protocollen van deze firma, namelijk DeviceNet en ControlNet. Ethernet/IP is nu onder beheer van de DeviceNet gebruikersvereniging (ODVA).

Hoewel de bekabeling van DeviceNet en ControlNet anders is, hadden deze twee systemen al een gemeenschappelijk deel, genaamd CIP (Control & Information Protocol). Dit komt ook weer bij Ethernet/IP terug. Wat in feite gedaan is, is het koppelen van CIP aan TCP/IP.



Figuur 6-3: De netwerkarchitectuur van DeviceNet, ControlNet en Ethernet/IP met elkaar vergeleken. Alle drie hebben CIP als gemeenschappelijk protocol; de onderliggende protocollen en de bekabeling zijn wel steeds anders.

Op de website *www.ethernet-ip.org* (of via *www.odva.org*) is de specificatie van Ethernet/IP kostenloos op te halen. Dit is enigszins een farce, want op zeer veel plaatsen wordt verwezen naar de specificatie van ControlNet, en die is niet gratis. Wel kan men source-code ophalen van een vrij eenvoudige Ethernet/IP implementatie.

Het interessante aan de drieling DeviceNet, ControlNet en Ethernet/IP is dat ze alle drie CIP als applicatielaag (OSI laag 7) kennen. Dit maakt het voor de gebruiker zeer aantrekkelijk, want op alle drie netwerkniveaus kan dus met het hetzelfde protocol gewerkt worden (alleen de bekabeling is anders). Dit scheelt inleertijd. Daarnaast kan makkelijker overgestapt worden naar een andere variant uit de drieling. Tenslotte kunnen uitbreidingen op CIP direct ook op alle netwerkniveaus ingezet worden. Een voorbeeld hiervan is de (nog niet uitontwikkelde) safety-variant van CIP, genaamd CIPSafety. Alle andere Ethernet-varianten bieden deze eenduidigheid niet. Bijvoorbeeld, Siemens biedt op de drie netwerkniveaus straks ProfiNet, Profibus en AS-Interface aan, en deze ook nog in drie safety-varianten. Dit maakt het voor eindgebruikers allemaal niet makkelijker.

6.4 Powerlink

Powerlink is een ontwikkeling van de (Oostenrijkse) firma B&R. In eerste instantie is Powerlink ontwikkeld als protocol voor het eigen merk PLC's, maar later is een gebruikersvereniging opgezet (EPSG – Ethernet Powerlink Standaardisatie Groep). Powerlink implementaties kunnen nu geleverd worden door een Zwitserse hogeschool. Alhoewel men beloofd had de specificatie voor Powerlink openbaar te maken, is dit medio 2004 nog steeds niet gebeurd.

Bijzonder aan Powerlink is, dat er geen switches gebruikt mogen worden, maar enkel hubs. Dit is noodzakelijk om de real-time eigenschappen van Powerlink te garanderen. Dit gaat overigens veranderen met de introductie van het IEEE 1588 protocol, waarmee zeer nauwkeurige synchronisatie tussen deelnemers op Ethernet mogelijk is. Indien een switch gebruikt wordt die ook IEEE-1588 ondersteunt, dan is gebruik in Powerlink toegestaan.

Powerlink maakt op een speciale manier gebruik van Ethernet, zodat "collisions" voorkomen worden. Powerlink lijkt sterk op een master/slave systeem omdat één deelnemer de rol van "SCNM" (Slot Communication Network Manager) krijgt, welke bepaalt wie van de andere deelnemers op het netwerk een netwerkbericht mag sturen. Alle netwerkberichten worden overigens via "broadcast" verstuurd, waardoor toch iedereen met iedereen kan communiceren.

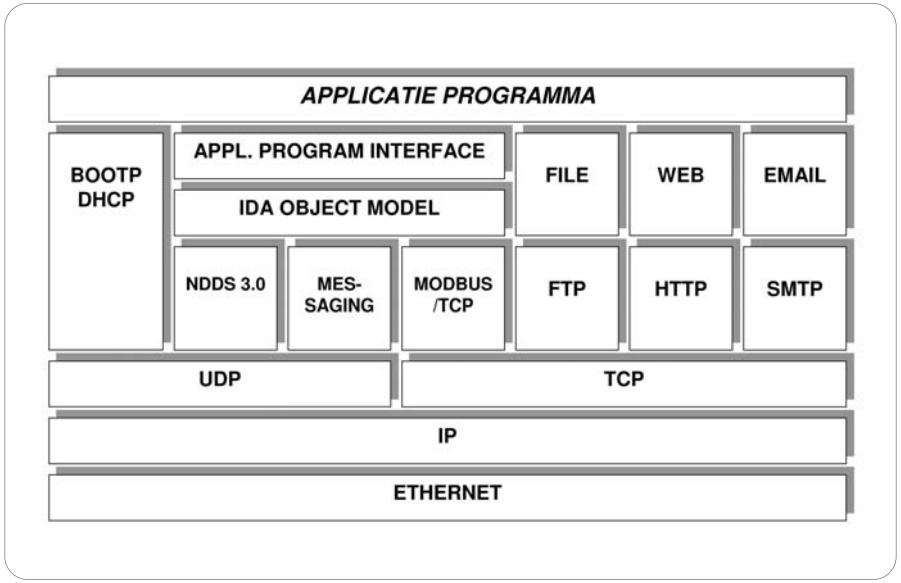
De eis dat er geen Ethernet switches mogen worden gebruikt, maar alleen hubs (repeaters) heeft te maken met de interne doorschakelvertraging van switches – deze moeten immers elk netwerkbericht compleet inlezen voordat het weer doorgestuurd kan worden. Dit kost een dubbele transmissietijd, en de interne verwerkingstijd van de switch moet hier nog bij opgeteld worden. Normaliter is dit geen probleem, maar voor hoge-snelheid motion-applicaties is dit onbruikbaar. Het is immers niet voorspelbaar hoe lang een netwerkbericht onderweg is (tussen zender en ontvanger). Deze "jitter" zorgt dan voor een hogere positie-onnauwkeurigheid. Als dit niet acceptabel is, zijn er twee oplossingen mogelijk: of het netwerk aanpassen, of het applicatieprogramma langzamer laten lopen (= minder output van een machine). In Powerlink is gekozen voor de eerste oplossing.

Elke Ethernet-gebruiker is bekend met de regel dat er nooit meer dan 4 repeaters (en dus ook hubs) in serie geschakeld mogen worden. Deze regel beperkt de maximale omvang van een netwerk, en dat heeft te maken met de interne detectie van collisions. Maar omdat Powerlink zo opgezet is dat er geen collisions mogelijk zijn, vervalt ook de bovenstaande regel. Volgens opgave van B&R kunnen tot 10 hubs in serie geschakeld worden.

In het Powerlink protocol is per cyclus een tijdslot ingeruimd voor andere vormen van communicatie en de bijbehorende netwerkprotocollen. Daarbij wordt veelal gewerkt met TCP/IP. Omdat elke TCP/IP implementatie zelf besluit wanneer netwerkberichten verzonden moeten worden, wordt het basisprincipe van Powerlink geschonden omdat er dan toch weer collisions kunnen ontstaan. Daarom worden TCP/IP berichten via Powerlink zelf verstuurd; Powerlink splitst deze op in stukken van maximaal 256 bytes, en assembleert ze aan de ontvangende kant weer. Daarvoor is wel een speciale Powerlink Ethernet-kaart nodig. Op PC's is dit niet echt praktisch; een PC kan daarom wel met een Powerlink-module communiceren, maar deze module mag dan het Powerlink-protocol niet meer uitvoeren.

6.5 IDA

Het Duitse "Interface for Distributed Automation" is de enige geheel nieuwe ontwikkeling rondom industrieel Ethernet. Met 'nieuw' wordt dan bedoeld: er is geen historie in de vorm van oudere versies van IDA. Men kan zich dus helemaal richten op het ontwikkelen van een zo modern en efficiënt mogelijk protocol. Figuur 6-4 schetst de architectuur van IDA, welke is gebaseerd op TCP/IP en Ethernet, in combinatie met al bestaande applicatieprotocollen (FTP, http, SMTP, BOOTP, etc.) en een nieuwe set functionaliteit voor de afhandeling van remote I/O (NDDS), inter-PLC communicatie (Modbus/TCP) en safety-applicaties. Hierboven draait dan het applicatieprogramma.



Figuur 6-4: De netwerkarchitectuur van IDA, welke veel bekende elementen bevat.

Het deel van IDA dat verantwoordelijk wordt voor de snelle afhandeling van remote I/O is gebaseerd op een al langer bestaand commercieel verkrijgbaar product: NDDS van het Amerikaanse bedrijf RTI (Real Time Innovations, *www.rti.com*). NDDS is een zgn. "publish/subscribe" (p/s) protocol (ook wel "producent/consument" genoemd), een manier van werken op een netwerk die nogal afwijkend is van wat in de meeste bestaande bus-systemen gebruikelijk is. Daar moet data waar men interesse in heeft zélf opgehaald worden bij diegene die de data in beheer heeft, bijvoorbeeld via een "Read" commando. Het basisprincipe bij een p/s protocol is precies omgekeerd: alle data op alle deelnemers wordt bij elke wijziging naar álle andere deelnemers op het netwerk rondgestuurd. Elke deelnemer heeft dus een lokale kopie van alle data die in het systeem aanwezig is. Omdat het steeds rondsturen van alles naar iedereen redelijk veel overhead geeft, tenslotte is niet iedereen in alle data geïnteresseerd, wordt vaak uitgegaan van een abonnementen-systeem. Een deelnemer die interesse heeft in bepaalde data moet zich bij de producent van die data aanmelden als abonnee (subscriber). Als subscriber krijgt men dan automatisch enkel die data binnen waar (applicatietechnisch gezien) interesse in is, en een producent hoeft geen data te sturen waarvoor geen abonnees aangemeld zijn.

Een deel van de specificatie van IDA's p/s protocol is inmiddels vrijgegeven, en te vinden als "RTPS Wire Protocol Specification". Het document is ook op te vragen door een email te sturen naar *info@rti.com*, met een verwijzing naar het genoemde document. Het ligt in de bedoeling dat IDA's protocol t.z.t. ook een Internet-standaard gaat worden (maar dat ligt nog wat verder in de toekomst).

Naast het gebruik van NDDS voor I/O is een ander belangrijk protocol binnen IDA een zeer bekende: Modbus/TCP. Dat dit onderdeel is van IDA heeft niet zozeer technische achtergronden maar vooral politieke: Groupe Schneider, de intellectuele eigenaar van Modbus, is een van de grootste leden van de IDA-vereniging en heeft dus een vinger in de pap bij het bepalen van de technische inhoud van IDA. Eind 2003 is de IDA gebruikersgroep opgegaan in de Modbus groep. Wat dit zal betekenen voor de verdere ontwikkeling van IDA is niet duidelijk.

6.6 FF HSE

De "High-Speed Ethernet" van de Foundation Fieldbus is een van de eerste Ethernet-varianten van een modern industrieel netwerk. Dit is vrij eenvoudig ontwikkeld, door het bestaande FF protocol te nemen, en dit te transporten via TCP/IP. Via zgn. "linking devices" kan dan een koppeling gemaakt worden tussen een HSE-netwerk en (een of meerdere) FF H1 netwerken, die op 31,25 Kbit/s lopen. Deze snelheid, ooit gekozen in 1988, is inmiddels te langzaam voor alle speciale eisen die aan een netwerk gesteld worden op het gebied van diagnostiek, embedded webservers, onderhoud, etc. De FF gebruikersvereniging heeft daarom gekozen voor een grote stap voorwaarts; de ontwikkeling van de eigen FF H2 versie is geschrapt en in plaats daarvan is gekozen voor 100 Mbit/s Ethernet.

De combinatie van FF HSE / H1 levert een tweelaags netwerkarchitectuur op, gekoppeld via meerdere linking devices. Deze voorzien ook in de voeding van de aangesloten apparatuur, want dat is met Ethernet niet mogelijk.

6.7 EtherCat

EtherCAT is een Duitse ontwikkeling welke in 2004 op de markt gekomen is (*www.ethercat.org*). Speciaal aan EtherCAT is dat het twee van de meest genoemde nadelen van Ethernet aanpakt, namelijk de lastige stervormige bekabeling, en de interne overhead.

EtherCAT heeft geen enkele overeenkomst met de andere Ethernet-gebaseerde systemen. Integendeel, EtherCat lijkt qua interne werking eerder enigzins op Interbus. Bij beiden wordt via een "total frame protocol" zeer efficiënt de I/O afgehandeld. De truc die uitgehaald wordt is vrij simpel: in plaats van het sturen van een nieuw netwerkbericht met data voor één specifieke deelnemer, wordt nu maar één netwerkbericht verstuurd met daarin alle data voor alle deelnemers tegelijk. Iedereen haalt hier dan zijn eigen veld data (= outputs) uit. De inputs worden ook allemaal verzameld in 1 netwerkbericht, en dan op de besturing afgeleverd. Dit is makkelijk haalbaar omdat in een Ethernet netwerkbericht toch ruimte is voor 1500 bytes data, waarmee dus 9000 digitale kanalen of 750 analoge kanalen (of een mix hiervan) afgehandeld kunnen worden.

In de commerciële literatuur worden een aantal voorbeelden van mogelijke snelheden gegeven:

- 1000 digitale I/O in 30 µs;
- 200 analoge I/O kanalen in 50 µs;
- 100 servoassen in 100 µs (= 20 kHz sample rate) met een synchronisatieafwijking < 1 µs;
- 12000 digitale I/O in 350 µs;
- Algemeen: 10 Kbyte / msec.

Omdat alle EtherCAT I/O modules voorzien zijn van 3 aansluitingen (in, uit, aftakking), kan vrij flexibel bekabeld worden. In principe heeft men een daisy-chain structuur (ketting, doorlussing), maar het is mogelijk om aftakkingen te maken via de 3e aansluiting. Hiermee wordt de daisy-chain minder kwetsbaar, want het verwijderen of uitvallen van een deelnemer in de ketting maakt dat alle deelnemers verderop onbereikbaar worden.

In totaal kunnen 65536 deelnemers worden aangesloten, over een afstand van ca. 500 km. De afstand tussen 2 deelnemers is echter beperkt tot 100 meter (CAT5 bekabeling, UTP) of ca. 2 km als van glasvezel gebruik gemaakt wordt. Er zijn geen hubs / switches nodig, maar die mogen er wel zijn.

Elke deelnemer krijgt een eigen "shared memory" van maximaal 64 Kbyte; de totale geheugenruimte wordt dan 4 GByte. Iedereen kan zelf beslissen welke geheugendelen lokaal aanwezig zijn (als kopie); dit maakt ook 'broadcast' van data mogelijk (= iedereen die interesse heeft, krijgt op hetzelfde moment een update). De snelheid die EtherCat haalt bij 1500 deelnemers is 10 Kbyte/msec (ruwweg de 100 Mbit/s van Ethernet gedeeld door

1000 msec en dan nog wat naar beneden afgerond). De uitvoering van het EtherCat protocol kan geheel worden overgelaten aan speciaal ontwikkelde chips (ASIC's) genaamd FMMU – Fieldbus Memory Management Unit, waardoor geheel geen software nodig is. Dit is een belangrijk voordeel qua snelheid, omdat (hoe men het ook bekijkt) software altijd een enigzins vertragende werking heeft (en niet in elke I/O module een 4 GHz Pentium met een 150W voeding ingebouwd kan worden). In de besturing zelf is geen FMMU nodig, hier kan men een gewone, goedkope Ethernet-kaart gebruiken.

Via 'tunneling' van EtherCat-berichten in het UDP-protocol is het mogelijk om een koppeling te maken via TCP/IP naar besturingen die het EtherCat-protocol niet kunnen uitvoeren. De hoge snelheid van EtherCat gaat dan wel verloren.

6.8 Sercos-III

Sercos is een bekende veldbus, die begin jaren 90 op de markt gekomen is. In tegenstelling tot veel andere bussystemen poogt Sercos niet een universeel netwerk te zijn dat voor alle applicatiegebieden bruikbaar is, maar wél om zo optimaal mogelijk inzetbaar te zijn voor high-speed motion applicaties. De manier van bekabelen (ringstructuur), gebruikte bekabeling (glasvezel) en de functionaliteit van het netwerkprotocol zijn hierop aangepast. Op dit moment is de 2e versie van Sercos (Sercos-II) op de markt, ook wel bekend als norm IEC-61491.

De hype rondom de opkomst van industrieel Ethernet enkele jaren geleden heeft, in eerste instantie, niet geleid tot enige paniek bij de Sercos gebruikersvereniging (*www.sercos.de* en *www.sercos.org*). Door de inefficiency van Ethernet, zijn de grote overhead, en de afwezigheid van speciale hardware, was Sercos-II uiteindelijk altijd veel sneller dan Ethernet, ook al liep Sercos-II 'slechts' op 16 Mbit/s en Ethernet op 100 Mbit/s. Vanuit de Sercos gebruikersvereniging werd de wereld dan ook opgeroepen niet al te veel af te gaan op de *bruto* snelheid van een netwerk, maar vooral te kijken naar de *netto* snelheid. Dit gezondboerenverstand is echter ondergesneeuwd onder de marketinghype van veel leveranciers ("100 Mbit/s is 6x zo snel als 16 Mbit/s").

De opkomst van Ethernet-protocollen die ook speciaal bedoeld zijn voor high-speed motion toepassingen heeft bij de Sercos gebruikersvereniging wél geleid tot een heroriëntatie op industrieel Ethernet. Deze nieuwe protocollen (oa. ProfiNet V3, Powerlink V3, EtherCAT) worden immers directe concurrenten van Sercos-II. Eind 2003 is daarom het "Sercos-III" project gestart, dat moet leiden tot een Ethernet-variant van Sercos.

De belangrijkste eigenschappen van Sercos-III zijn:

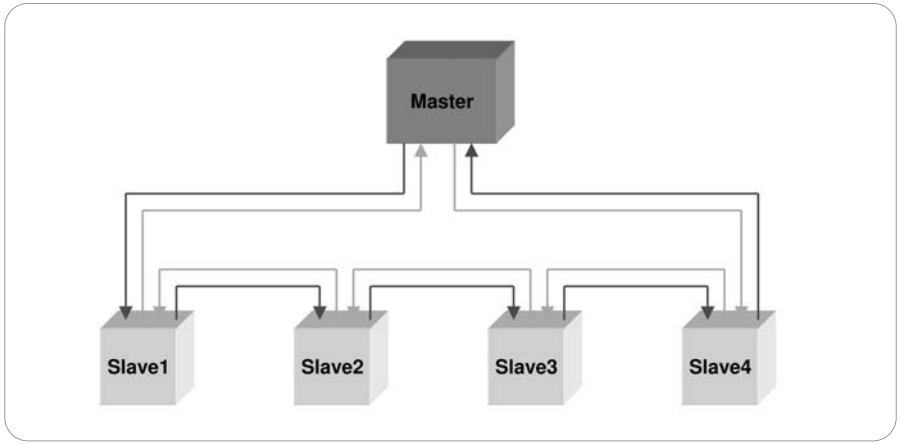
- Gebruik van Ethernet bekabeling en connectoren.
- Redundantie dankzij gebruik van een dubbele-ring bekabelingsstructuur (niet verplicht).
- "Hot plugging" van deelnemers mogelijk (indien dubbele ring gebruikt wordt).
- Bruto snelheid stijgt van 16 naar 100 Mbit/s.
- Minimale cyclustijd daalt van 62,5 naar 31,25 µsec.
- Mogelijkheid tot transport van TCP/IP netwerkberichten.
- Communicatie tussen deelnemers onderling wordt mogelijk.
- Gebruik van safety-functies.
- Hardware-gestuurde synchronisatie.

De interne werking van Sercos-III is zoveel mogelijk hetzelfde gebleven als bij eerdere versies. Wel heeft men, er wordt immers met Ethernet gewerkt, het mogelijk gemaakt om het TCP/IP protocol parallel aan Sercos-III uit te voeren. Wel een kleine kanttekening: er is wel een uitbreiding op Ethernet aangebracht, zodat het niet meer mogelijk is om "gewone" Ethernet-apparatuur in hetzelfde netwerk op te nemen.

Bekabeling

Sercos-III heeft ervoor gekozen om gebruik te maken van standaard 'koperen' Ethernet-kabels met de bijbehorende connectoren. In vergelijking met Sercos-II, waarbij nog glasvezel werd gebruikt met FSMA connectoren, levert dit een kostenvoordeel op. Ook de elektronica in een apparaat kan veel eenvoudiger worden; met de speciale "Single chip drive" Sercos-III controller wordt een netwerkinterface de helft goedkoper dan bij Sercos-II.

Net zoals bij eerdere versies van Sercos wordt gebruik gemaakt van een ringstructuur. Deze is echter grotendeels onzichtbaar, omdat de ring "platgeslagen" is in de gebruikte kabels. Verder kan nog gekozen worden uit het gebruik maken van een enkelvoudige ring, of van een dubbele ring. In het laatste geval is men bestand tegen uitval van één kabel of één deelnemer. Dit biedt tevens de mogelijkheid tot "hot plugging": het toevoegen of verwijderen van deelnemers op een netwerk, dat verder gewoon operationeel blijft. Bij de enkelvoudige ring van Sercos-II was dit niet mogelijk (een bekend nadeel van ringvormige netwerken).



Figuur 6-5: De dubbele ring van Sercos

In tegenstelling tot veel andere industrieel-Ethernet systemen hoeft geen gebruik gemaakt te worden van een hub of switch, en dit levert nogmaals een kostenvoordeel op in vergelijking met andere Ethernetprotocollen. Deelnemers worden steeds doorgelust; aan het 'hoofd' van het netwerk moet de master (besturing) geplaatst worden. In feite ziet een Sercos-III netwerk er dus uit als een bussysteem. Intern echter is wel een ringstructuur aanwezig; dit komt doordat elke kabel zowel een heengaand als teruggaand signaal voert (dit is standaard Ethernet). Elke deelnemer is uiteraard ook een Ethernet-repeater. Bij standaard Ethernet geldt een regel dat er nooit meer dan 3 repeaters achter elkaar geschakeld mogen zijn. Bij Sercos-III geldt deze regel niet, omdat er geen "collisions" (gelijktijdige transmissie van 2 netwerkberichten) kunnen optreden.

Aanpassing op Ethernet berichten

Ethernet is op zich niet zo vreselijk goed geschikt voor high-speed motion toepassingen. Dit komt door de overhead die Ethernet kent: elk telegram is minimaal 84 bytes transmissietijd groot, ook als men maar een paar bytes data wil sturen. Dit is de paradox van industrieel Ethernet: het is zeer optimaal als men grote hoeveelheden data wil sturen (zoals in een kantooromgeving), maar industriële toepassingen sturen vaak maar zeer kleine hoeveelheden data: enkele bytes.

Bij andere protocollen voor industrieel Ethernet-gebruik (zoals ProfiNet, Powerlink en EtherCAT) heeft men al besloten om speciale aanpassingen uit te voeren om minder last te

hebben van de overhead. Ook voor Sercos-III is een aanpassing bedacht: speciale kleine netwerktelegrammen met véél minder overhead dan gebruikelijk:

- Een Ethernet "Destination" adres is 6 bytes groot, een Sercos-III adres één byte.
- Een Ethernet "Source" adres is 6 bytes groot, een Sercos-III adres één byte.
- Ethernet heeft 2 bytes nodig om de lengte van een netwerkbericht aan te geven.
- Ethernet eist *minstens* 46 bytes data in elk netwerkbericht, Sercos-III niet.
- Voor het detecteren van transmissiefouten heeft Ethernet 4 bytes extra data nodig, Sercos-III maar twee extra bytes.

Gesteld dat we effectief 8 bytes data willen sturen, dan is op Ethernet in totaal transmissietijd voor 84 bytes nodig. De efficiency is dan slechts ca. 10% (8/84). Op Sercos benodigen 8 bytes data in totaal slechts 24 bytes transmissietijd, de efficiency is dan 8/24 = 33%. Met andere woorden, op eenzelfde brutosnelheid (100 Mbit/s) kan Sercos netto 3x zo snel zijn dan zijn concurrenten. Dit vertaalt zich direct in kortere cyclustijden, en dus ook een snellere aansturing van alle assen, zodat nauwkeuriger gestuurd kan worden. In vergelijking met Sercos-II zakt daarom de kleinste cyclustijd van 62,5 naar 31,2 µsec. Dit is dus 2x zo snel, terwijl men misschien een factor 6 zou verwachten. Hier zien we toch iets van de inefficiency van Ethernet terug: zelfs de kleine netwerktelegrammen hebben toch nog relatief veel overhead. Maar de 6x zo grote bitrate (van 16 naar 100 Mbit/s) compenseert dit weer.

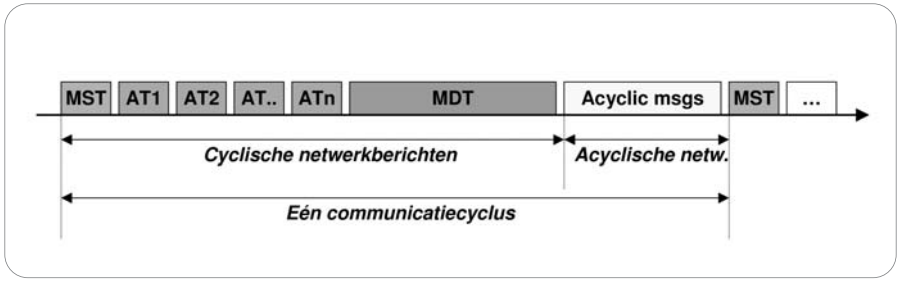
Uiteraard hebben de speciale Sercos-III netwerkberichten wel een consequentie: niet-Sercos III apparatuur zal deze netwerkberichten niet kunnen verwerken. Het zijn immers geen 'gewone' Ethernetberichten meer en worden dus genegeerd. Op een Sercos-III netwerk mag dan ook enkel Sercos-III apparatuur worden aangesloten. Alleen met een speciale converter is het mogelijk om een Sercos-III netwerk aan een 'normaal' Ethernet aan te sluiten. Als die er niet is moet de besturing als 'gateway' functioneren.

De cyclus

Een Sercos-III netwerk functioneert altijd op dezelfde manier: door het constant uitvoeren van "communicatiecycli". In feite is dit helemaal hetzelfde gebleven als bij Sercos-II. Nieuw is echter dat elke communicatiecyclus opgedeeld is in twee stukken: allereerst wordt de "cyclische data" uitgewisseld, en daarna is nog tijd beschikbaar voor de "niet-cyclische TCP/IP data". De benodigde tijd voor het afhandelen van de cyclische data is uiteraard afhankelijk van het aantal assen en de hoeveelheid data per as; de beschikbare tijd voor de niet-cyclische data wordt ingesteld door de gebruiker. Elke communicatiecyclus duurt dus even lang. Indien er in een communicatiecyclus te weinig tijd beschikbaar is voor TCP/IP,

dan moet deze bewaard worden tot de volgende communicatiecyclus. De applicatiebesturing krijgt dus in zeer constant tempo data van alle assen binnen, en stuurt ze in hetzelfde tempo aan.

Voor het versturen van cyclische data wordt gebruik gemaakt van de 'korte' netwerkberichten; de niet-cyclische communicatie wordt verstuurd via standaard Ethernet-berichten. Een cyclus begint steeds doordat de besturing een "MST" (Master Synchronisatie Telegram) verstuurd naar de 1e as in de ring. Deze as zal het MST bericht doorsturen, en direct daarop volgend zijn eigen "AT" (As Telegram) sturen naar de 2e as in de ring. Ook deze as zal zijn AT sturen, direct ná de AT van de eerste as. Zo ontstaat er dus een treintje AT's; en omdat de bekabeling in een ring loopt krijgt de master dus alle AT's binnen. De master 'weet' hoeveel assen er zijn, en dus ook hoeveel tijd het kost om alle AT's te versturen. Direct daarna zal de master zijn "MDT" (Master Data Telegram) sturen; deze wordt door álle assen ontvangen en elke as peutert de voor hem bestemde gegevens uit het MDT. De master is nu op de hoogte van de actuele stand van zaken op alle assen, en alle assen hebben nieuwe instructies gehad. Tot zover eigenlijk niets bijzonders, Sercos-II werkt ook op deze manier.



Figuur 6-6: De opbouw van een Sercos-III communicatiecyclus, welke zich continue herhaalt.

Na de cyclische data is er ruimte voor transmissie van "gewone" Ethernetberichten. Dit zal vaak TCP/IP verkeer zijn, in combinatie met de daarbij horende protocollen. Deze extra communicatiemogelijkheden kunnen o.a. gebruikt worden voor remote diagnostiek via ingebouwde webservers, laden van grote applicatieprogramma's, communicatie met de besturing, etc. De voor TCP/IP beschikbare tijd is instelbaar. De totale tijdsduur van een communicatiecyclus ligt dus altijd vast. Een plotselinge toename in de belasting van TCP/IP heeft dus geen effect op de afhandeling van de cyclische data. Alle assen blijven altijd in hetzelfde tempo aangestuurd worden.

Deze manier van werken op Sercos-III vereist dus geen enkele run-time coördinatie tussen de deelnemers; als iedereen goed geconfigureerd is kan het nooit voorkomen dat twee of meer deelnemers elkaar storen met een gelijktijdige transmissie. Normaliter zou dit op Ethernet een zgn. "collision" geven, en dit is een van de redenen waarom standaard Ethernet niet zo bruikbaar is voor industriële toepassingen. Bij Sercos-III wordt dit probleem dus geheel voorkomen. Een extra voordeel is nog dat er geen bandbreedte wordt verspeeld aan het afhandelen van collisions, en dit vertaalt zich dus in een hogere snelheid (lees: hogere nauwkeurigheid).

Snelheid

De concurrentie van de andere Ethernet-protocollen voor high-speed motion heeft er toe geleid dat Sercos-III een uitbreiding op Ethernet heeft moeten aanbrengen. Had men dit niet gedaan, dan was Sercos-III qua snelheid niet onderscheidend genoeg van zijn concurrenten, en dit zou m.i. het einde van Sercos betekend hebben. Door echter kleinere netwerktelegrammen (met veel minder overhead) te gebruiken is een veel efficiëntere aansturing mogelijk, en dus blijft Sercos een voordeel bieden in vergelijking met de andere protocollen. Tabel 6-1 geeft de haalbare cyclustijden voor een aantal verschillende situaties.

Hoeveelheid data per as	Aantal assen	Cyclustijd	Opmerking
8	8	31,25µsec	Koppel / positie
12	16	62,5 µsec	Snelheid / positie
16	30	125 µsec	Snelheid / positie / I/O
12	72	250 µsec	Snelheid / positie
32	36	250 µsec	"Brede" interface
12	150	500 µsec	Snelheid / positie
53	100	1 msec	100 Assen
32	153	1 msec	"Brede" interface
16	254	1 msec	Max. aantal assen
40	254	2 msec	Max. aantal assen
65	254	4 msec	Max. aantal assen

Tabel 6-1: Snelheid van Sercos-III in een aantal variaties (bron: Sercos gebruikersvereniging).

De genoemde getallen gelden enkel indien geen TCP/IP netwerkverkeer verstuurd wordt. Een "maximaal" (qua omvang) Ethernetbericht kost 127 µsec transmissietijd. Indien men in een Sercos-III cyclus hiervoor zoveel ruimte (tijd) reserveert, wordt de cyclustijd natuurlijk wel aanzienlijk langer.

6.9 Kloksynchronisatieprotocollen

Een man met twee horloges...weet nooit zeker hoe laat het is, een man met één horloge wel. Zo luidt het gezegde, en wat geldt voor mensen met twee klokken geldt ook voor systemen waarin meerdere besturingen zitten die elk een eigen (real-time) klok hebben. Het is niet eenvoudig om op alle besturingen exact dezelfde tijd op hun klok te krijgen, laat staan te houden (dankzij de onvermijdelijke verloop vanwege oscillatordrift).

In veel applicaties is het niet zo belangrijk dat alle besturingen dezelfde tijd op hun klok hebben staan; meestal wordt vanuit één hoofdbesturing de rest aangestuurd, en alleen de klok op deze besturing wordt gebruikt tijdens de uitvoering van het applicatieprogramma, voor het bijhouden van logboeken, en voor het starten en stoppen van extern opgedragen activiteiten.

Echter in veel motion-applicaties, besturing van elektriciteitsnetten, test- en meetsystemen, en softwareontwikkelings-werkzaamheden zijn nauwkeurig gesynchroniseerde klokken nodig. Bij dit laatste moeten in gedistribueerde systemen vaak lastige fouten ('bugs') uit de software gehaald worden, en indien de logboeken van de besturingen niet de juiste tijdstippen aangeven, is het vaak zeer lastig tot onmogelijk om de oorzaak en de gevolgen van een fout te vinden. Nu lijken twee klokken die op 1 milliseconde synchroon lopen wel heel nauwkeurig, maar dit is het toch niet: met een (bv.) moderne Intel-processor die op 3 GHz loopt wordt in 1 msec meer werk verzet dan de allereerste PC (1980) in 1 seconde afhandelen kon. Met zo'n minuscuul tijdverschil is het dan al vaak onmogelijk om te zeggen welke besturing het eerst de fout in ging.

Drift

Ook al worden de klokken van verschillende besturingen gelijkgezet, ze beginnen direct weer te verlopen onder invloed van onvermijdelijke kleine verschillen ('drift') tussen de elektronica van alle besturingen. Het is eventueel mogelijk om via een parallelle bus alle besturingen aan te sluiten op een-en-dezelfde klok, maar dit is zowiezo geen goedkope oplossing vanwege de hoeveelheid bekabeling, en onpraktisch/onmogelijk indien grotere afstanden dan enkele meters overbrugd moeten worden. Het is wél goed mogelijk om een (industrieel) netwerk te gebruiken om de klokken van besturingen met elkaar te synchroniseren, op een zodanige manier dat het maximale tijdsverschil tussen twee klokken niet groter is dan enkele seconden, milliseconden, microseconden of zelfs nanoseconden. Hoeveel exact hangt af van het gebruikte "netwerk kloksynchronisatieprotocol". In de loop der jaren zijn hiervan verschillende varianten ontwikkeld.

Kloksynchronisatie via een netwerk is geen simpel onderwerp, alhoewel met eenvoudige middelen toch wel een redelijk resultaat behaald kan worden als men niet kijkt op een minuutje verschil. De meeste eenvoudige vorm van kloksynchronisatie is standaard beschikbaar in Windows. Het commando **"net time \\serverpc /yes"** zal de lokale klok gelijkzetten aan die van de opgegeven PC met naam "serverpc". Indien het tijdsverschil groter is dan 1 minuut, dan wordt de eigen klok direct gelijkgezet aan die van de opgegeven server. In andere gevallen wordt de lokale klok iets versneld of vertraagd, zodanig dat het tijdsverschil steeds kleiner wordt. Naast deze eenvoudige kloksynchronisatie zijn er in Windows nog veel uitgebreidere mogelijkheden.

Een identiek mechanisme is ook aanwezig in Unix. Door op deze manier te werken, wordt voorkomen dat er grote 'gaten' ontstaan in de voortschrijdende tijd, en tevens blijft de tijd monotoon vooruitlopend toenemen.

Synchronisatie tot op milliseconden nauwkeurigheid

Het gebruik van een netwerk om klokken op deze manier te synchroniseren is praktisch als men niet kijkt op enkele tientallen seconden afwijking (zoals bv. in een kantooromgeving). Wil men echter een afwijking die niet groter is dan enkele tientallen *milli*seconden, dan wordt het gebruik van een netwerk lastig. Indien men een telegram met een nieuwe tijd voor een klok naar een andere besturing wil sturen, dan moet rekening gehouden met de snelheid van het netwerk, de netwerkbelasting op het moment dat men dat telegram stuurt, en de (software) verwerkingstijden van de netwerk interface modules. Indien men hier niet voor corrigeert zal de ontvanger meteen al weer achterlopen. Bijvoorbeeld, als men een netwerkbericht stuurt "Het is nu 21:17:10.556" maar het duurt 70 milliseconden voordat dit ontvangen en verwerkt is, dan is er nog een synchronisatieverschil van 70 milliseconden tussen beide systemen. Daarom moet altijd rekening gehouden worden met vertragingen door software, en door het netwerk zelf. Lastig hierbij is dat de vertraging van een netwerk niet constant is.

Hiervoor zijn dan ook verschillende protocollen ontwikkeld, zoals bv. NTP (Network Time Protocol) en SNTP (Simple NTP). NTP is overigens dé standaard op het gebied van kloksynchronisatie. Men kan via internet de lokale klok synchroniseren met een of meerdere "time servers", waarvan sommigen voorzien zijn van atoomklokken.
Bij veel bedrijven wordt dan één time server gebruikt om via internet te synchroniseren, en alle besturingen synchroniseren dan weer met de eigen time server. Op deze manier is een hiërarchie van time servers op te bouwen. Een nadeel van NTP is wel dat het vrij langzaam is, en gedurende langere tijd lopen klokken dus niet synchroon.

Verder is een beveiliging ingebouwd tegen time servers die expres verkeerde informatie aanleveren, hetgeen op internet natuurlijk niet moeilijk is. Het protocol is verder vrij complex, maar voor PC's en voor Unix is gelukkig zeer veel software op dit gebied verkrijgbaar. De website *www.ntp.org* is hét startpunt voor verdere informatie over dit protocol, en *http://tf.nist.gov/service/its.htm* helpt u verder aan software.

Synchronisatie tot op microseconde nauwkeurigheid

De meest recente ontwikkeling op het gebied van tijdssynchronisatie is de IEEE-1588 norm (PTP – "Precision Time Protocol"). Het is oorspronkelijk ontwikkeld door Hewlett-Packard (nu Agilent) in een consortium met voornamelijk Amerikaanse bedrijven. De eerste versie werkte op Ethernet, maar PTP is verder netwerkonafhankelijk. PTP is speciaal ontwikkeld voor industriële toepassingen waarbij klokken in relatief kleine netwerken binnen een microseconde synchroon moeten lopen. Het protocol is verder vrij eenvoudig (in vergelijking met NTP), zodat geen zware processor nodig is. Ook genereert het bijna geen netwerkbelasting: er hoeft maar één keer per seconde één netwerkbericht verstuurd te worden. Tenslotte heeft PTP geen configuratie nodig, en is dus heel eenvoudig in gebruik te nemen.

Een verschil met NTP is dat software-vertragingen zoveel mogelijk voorkomen moeten worden. Het idee is dan ook dat de afhandeling van PTP berichten in Ethernet-switches en -routers met prioriteit moet gebeuren, zodat deze berichten zo snel mogelijk doorgestuurd worden naar de eindbestemming, in plaats van achteraan in een wachtrij voor uitgaande netwerkberichten geplaatst te worden. Zulke switches bestaan op dit moment echter nog niet. Een tussenoplossing is om een PTP time server op meerdere subnetwerken tegelijk aan te sluiten, zodat de berichtenstroom niet via routers hoeft te lopen.

Op dit moment is nog geen apparatuur op de markt die met PTP werkt. Eind september 2003 is het eerste congres van leveranciers en gebruikers over de implementatie van deze norm gehouden, met o.a. lezingen over industriële toepassingen van NTP door diverse leveranciers. Een van de eerste grootschalige toepassingen van PTP is in de besturing van hoogspanningsleidingen, waarvoor nauwkeurigheden in de orde van 1 microseconde nodig zijn. Een eerste serie metingen over de kwaliteit van PTP is gedaan door Hewlett-Packard in een eenvoudig netwerk; een gemiddelde afwijking van 22 nanoseconde (standaarddeviatie 99 ns) is daarbij gemeten.

Van de industriële netwerken hebben tot nu toe alleen ProfiNet, het trio EthernetIP/DeviceNet/ControlNet en Powerlink uitgesproken dat ze IEEE-1588 gaan leveren in hun eigen protocollen. Verder is een implementatie op LON bekend.
De website *http://ieee1588.nist.gov/* geeft verdere verwijzigingen over actuele ontwikkelingen en beschikbare producten.

Gebruik van GPS

Uiteraard zijn er nog andere methodes om klokken te synchroniseren, bijvoorbeeld met ontvangst van GPS (Global Positioning System) signalen of via de speciale langegolf radiozenders. Indien correct geïmplementeerd kunnen klokken met GPS tot op 100 nanoseconde nauwkeurig met elkaar gesychroniseerd worden. Het kan echter een kostbare oplossing worden indien elke deelnemer in een lokaal netwerk een ontvanger moet krijgen. Een tussenoplossing, waarbij één deelnemer een ontvanger krijgt, en de anderen via NTP of PTP met hem synchroniseren is dan aantrekkelijker. GPS is vooral interessant omdat het ook de klokken van geografisch gescheiden systemen met elkaar kan synchroniseren.

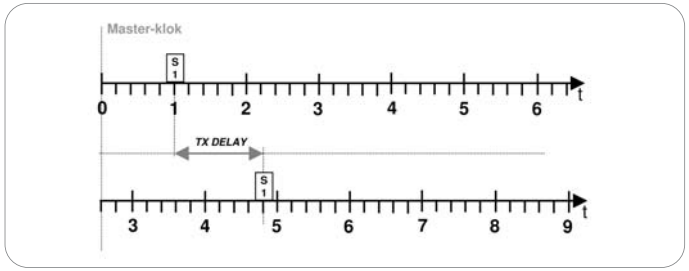
Nadelen

Overigens is het niet altijd goed als besturingen exact gelijklopen. De auteur maakte dit onlangs mee in een systeem waarbij PC's gebruikt werden die nog op MSDOS liepen. Na installatie van kloksynchronisatie-software bleek opeens dat soms bepaalde PC's niet meer wilden opstarten. Uiteindelijk bleek dit te liggen aan het netwerkprotocol DHCP, dat niet om kan gaan met MSDOS PC's die tot op de milliseconde exact dezelfde tijd hebben (Windows PC's hebben met DHCP hier overigens geen last van). Dit komt omdat bij veel operating-systemen en programmeertalen de functie die een random getal moet uitrekenen gebruik maakt van de interne klok (DHCP heeft een random getal nodig in zijn netwerkberichten om ze van elkaar te kunnen onderscheiden).

De (vereenvoudigde) werking van een kloksynchronisatieprotocol

In de onderstaande figuren (bron: IAONA Zwitserland) zijn vereenvoudigd de synchronisatiestappen tussen twee netwerkdeelnemers beschreven; één deelnemer is de master-klok, en de andere deelnemer wil zijn eigen klok met die van de master synchroniseren.

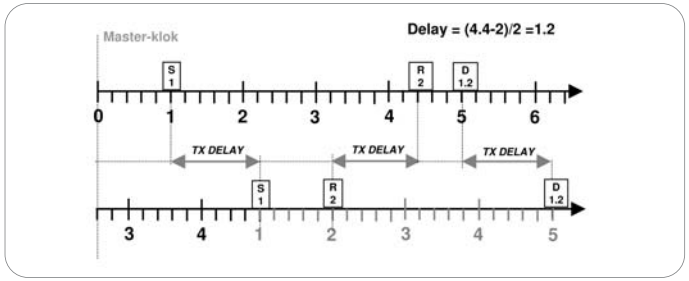
De synchronisatie van beide klokken geschiedt in een aantal stappen, waarna na afloop de slave dezelfde tijd op zijn klok heeft staan als de master. In het begin (figuur 6-7) is dat nog niet geval; de slave loopt ca. 2.6 tijdseenheden voor op de master.



Figuur 6-7: De eerste stap in de synchronisatie van de klok van een slave met die van de master.

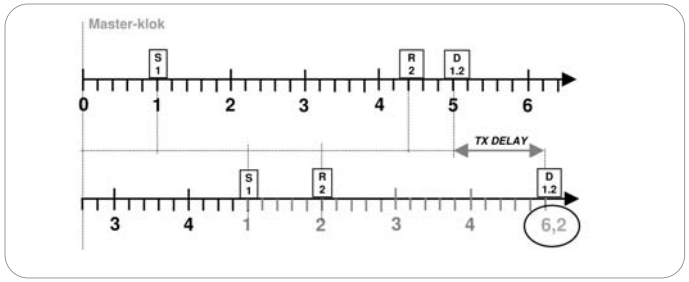
De eerste stap in de kloksynchronisatie tussen een master en een slave wordt genomen door de master. Deze stuurt een netwerkbericht naar de slave, bevattende het tijdstip waarom dat netwerkbericht verstuurd is: in dit geval t=1, namelijk de klok van de master zelf. De slave ontvangt dit netwerkbericht dan, voor hem op tijdstip t=4.8. De tijd tussen het moment van verzending en moment van ontvangst is nodig voor de uitvoering in software, en de transmissietijd van het netwerkbericht ("tx delay").

Nadat de slave op zijn kloktijd t=4.8 het bericht van de master ontvangen heeft, zal de slave zijn eigen klok gelijkzetten met de in het netwerkbericht opgegeven tijd (=1).
Dit heeft meteen al een positief effect, want waar beide klokken eerst 2,8 eenheden uit elkaar liepen, is dat nu nog maar 1,2 eenheid (zijnde de transmit-vertraging van het netwerkbericht).



Figuur 6-8: De tweede stap in het kloksynchronisatieproces. De slave's klok is nu al wel geactualiseerd, maar nog niet gesynchroniseerd met de master.

Na een interne verwerkingstijd zal de slave een bericht sturen aan de master, met hierin de lokale tijd (=2). Dit komt op de master binnen, en deze weet nu hoeveel tijd er verstreken is tussen zijn eerste bericht en dit tweede bericht. Hieruit valt ook af te leiden wat de transmit vertraging is. Het bericht komt binnen op t=4,4 en is verstuurd op t=1,0, dus 3,4 eenheden tijd nodig voor een complete cyclus. De slave had zelf 1 eenheid nodig om een antwoord te geven (= het verschil tussen de het verstuurd tijdstip t=2 en het ontvangen tijdstip t=1). Blijven over 2,4 eenheden tijd, en hierin zijn twee netwerkberichten verstuurd, dus de gemiddelde transmit vertraging is 1,2 eenheid tijd (figuur 6-8). Echter, de slave weet dit nog niet. Daarom stuurt de master een bericht naar de slave, bevattende de gemiddelde transmitvertraging (figuur 6-9). De slave zal nu nogmaals een synchronisatie uitvoeren, ditmaal van 1,2 eenheid. De beide klokken lopen nu gelijk.



Figuur 6-8: Na de derde stap loopt de klok van de slave gelijk met die van de master.

7. Randapparatuur

7.1 Gateways

In veel systemen wordt gebruik gemaakt van Ethernet én een ander type (industrieel) netwerk. De koppeling tussen beide types netwerken kan uitgevoerd worden door een "gateway", letterlijk te vertalen met "poort", maar "tolk" is een vertaling die de feitelijke werking van een gateway beter weergeeft. Overigens is de term zelf niet gestandaardiseerd, allerlei soorten apparatuur worden door hun leveranciers gateway genoemd, terwijl het datacommunicatie-technisch geen echte gateways zijn. Andersom zijn er ook producten die "protocol converter", "proxy" of "linking device" worden genoemd, maar feitelijk ook gateways zijn.

Grosso modo zijn er twee soorten gateways te onderscheiden:

- De "shared memory" gateways, voor koppeling tussen Ethernet en een remote I/O systeem.
- De "tolkende" gateways, voor koppeling tussen Ethernet en een ander type netwerk.

Het verschil tussen beide soorten gateways zit in de interne manier van functioneren.

Tolkende gateways

Een tolkende gateway vertaalt elk netwerkbericht van het ene netwerk in een gelijksoortig netwerkbericht voor het andere netwerk, met doorgave van de meegestuurde data. Dit lijkt makkelijk, maar is het niet altijd. De verschillen tussen twee protocollen kunnen vrij groot zijn, en daarom kunnen tolkende gateways vrij complex zijn. Dit maakt ze moeilijk in het gebruik, omdat veel kennis van beide aangesloten netwerken nodig is. Ook ziet men vaak dat niet alle mogelijkheden van beide protocollen ondersteund worden, vaak alleen een subset: enkel de meest gebruikte commando's, of enkel de gemeenschappelijke commando's. Het is daarom verstandig om op voorhand uitgebreid de documentatie van de gateway te bestuderen alvorens tot aanschaf van een product over te gaan.

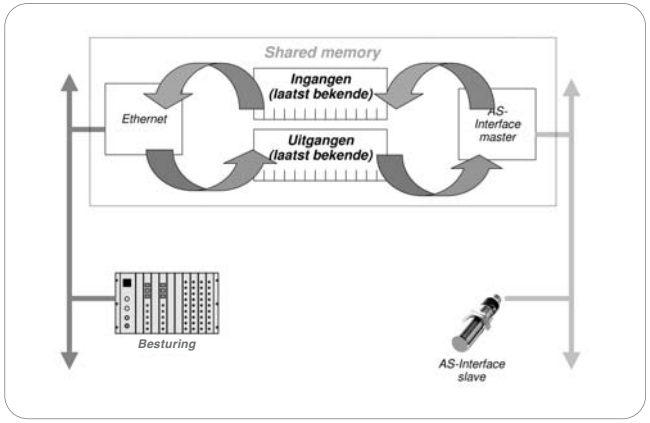
Het nadeel van dit type gateways is dat elk netwerkbericht geconverteerd dient te worden. Dit kost conversietijd (in de gateway) en transmissie van twee netwerkberichten (op beide netwerken). Gateways tussen Ethernet en een remote I/O netwerk kunnen daarom beter van het andere type zijn (zie de volgende paragraaf).

Shared memory gateways

Puur datacommunicatie-theoretisch gezien zijn shared memory gateways eigenlijk geen echte gateways, omdat ze geen protocolvertaling tussen beide aangesloten netwerken uitvoeren. Dit theoretische aspect neemt niet weg dat shared memory gateways zeer populair zijn, omdat ze eenvoudig te begrijpen en in gebruik te nemen zijn. Ook voor leveranciers is een shared memory gateway makkelijk te ontwikkelen.

Uiteraard heeft de gateway wel twee protocolstacks ingebouwd. De communicatie tussen beide wordt geregeld door middel van een stuk shared memory (figuur 7-1). Dit geheugen mag door beide partijen gelezen en geschreven worden. Meestal wordt er nog een opsplitsing gemaakt. Protocolstack A mag schrijven in de eerste helft van het geheugen, en B mag dit lezen. B mag zelf schrijven in de tweede helft van het geheugen, en A mag dit lezen. Op deze manier wordt voorkomen dat A en B elkaars gegevens overschrijven.

Deze manier van werken is zeer praktisch voor de afhandeling van I/O. Via beide helften uit het shared memory worden dan de inputs en de outputs verwerkt. Waar die precies in het geheugen staan, is meestal te configureren. Zodra dit gebeurd is, kunnen beide protocolstacks autonoom werken; ze hoeven nooit op elkaar te wachten want het shared memory is altijd beschrijfbaar en te lezen.



Figuur 7-1: De interne werking van een shared memory gateway Ethernet/AS-Interface. De besturing schrijft de nieuwe waardes voor de outputs in de onderste helft van het shared memory. De AS-Interface master gebruikt deze om de AS-Interface slaves aan te sturen. De inputs gaan de omgekeerde weg en kunnen allen in één Ethernet-bericht worden uitgelezen.

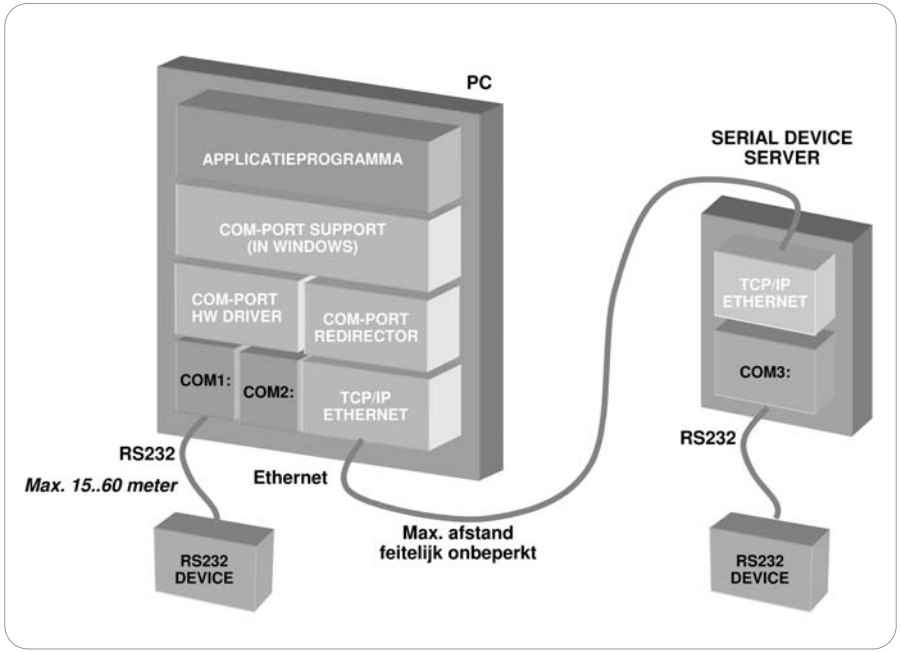
Het nadeel van het werken met shared memory is dat het geen geheugen heeft. Dat klinkt uiteraard enigszins vreemd, maar het klopt wel: indien nieuwe data in het geheugen geschreven wordt, is de oude data weg. Indien dus data sneller in het shared memory geschreven wordt dan de andere kant lezen kan, is er dus sprake van een synchronisatieprobleem. Dat is bij I/O meestal geen probleem, omdat enkel met de meest actuele I/O status gewerkt wordt. Maar indien men commando's, configuratie, foutmeldingen, statusmeldingen e.d. via shared memory door gaat geven, dan ontstaat altijd een synchronisatieprobleem. Te snel schrijven is niet goed voor de andere partij, en te snel lezen is voor de lezer niet goed, want die krijgt dan meerdere malen dezelfde data. In beide gevallen is er een soort van 'handshake' nodig.

Het voordeel van shared memory gateways voor de afhandeling van remote I/O is dat gebruik gemaakt kan worden van de hoge snelheid van de bestaande remote I/O bussystemen. Ook Ethernet wordt efficiënt gebruikt, want in plaats van heel veel netwerkberichten met weinig data en heel veel overhead, kan alle I/O nu afgehandeld worden met twee (voor inputs en outputs) netwerkberichten met veel data en dus weinig overhead. Indien men ook nog meerdere gateways op hetzelfde Ethernet heeft aangesloten, dan is er ook nog een snelheidsvoordeel dankzij de parallelle werking van alle gateways.

7.2 Serial Device Servers / RS232 Gateways

RS232 is altijd een van de werkpaarden geweest van de industriële communicatie, omdat het eigenlijk altijd 'gratis' op elke PC en besturing aanwezig is, en zeker niet te vergeten de eenvoud van werking. Dit zal binnenkort gaan veranderen, want de seriële (en de parallelle) poort zullen van de PC gaan verdwijnen en vervangen worden door USB. Dit is besloten door Microsoft in overleg met de PC-bouwers, en in de consumentenmarkt ziet men dan ook steeds meer USB-apparatuur op de markt komen.

Gelukkig blijft het mogelijk om vanuit een PC-applicatie met RS232-poorten te werken, maar dit moet dan via een USB/RS232 gateway, óf via Ethernet. Dit wordt dan gerealiseerd via extra hardware (een zgn. "serial device server", rechts op figuur 7-2) en extra software op een PC (zgn. "COM poort redirector", links op figuur 7-2).



Figuur 7-2: Een RS232-interface kan in een PC zelf zitten, of via een "serial device server" kilometers verder, zonder dat een applicatieprogramma dit ziet.

Op de locatie waar de extra RS232 aansluiting fysiek nodig is, wordt de serial device server geplaatst. Deze kan via het LAN (bijna altijd Ethernet dus) communiceren met de PC. Deze kan zelf voorzien zijn van standaard COM-poorten (COM1: en COM2:), maar dankzij de COM-poort redirector lijkt het net alsof de PC ook nog een COM3: heeft. Die is er wel, maar niet in de PC zelf, maar in de serial device server. Het applicatieprogramma op de PC hoeft hier echter niets van te zien, als via de standaard Windows functies met COM3: gewerkt wordt. Deze worden allemaal opgevangen (gesimuleerd) door de redirector, via TCP/IP netwerkberichten naar de serial device server gestuurd, en die doet het echte werk. De data die terugkomt wordt naar de PC gestuurd, en op de COM-manier teruggegeven aan het applicatieprogramma.

Uiteraard mag alleen gebruik gemaakt worden van standaard Windows functies. Applicaties die via zij- en achterdeurtjes speciale trucs uithalen van de PC's COM-poorten zullen nu natuurlijk niet meer werken, want de COM3: hardware zit elders in de wereld.

Meestal komt men hier pas achter door e.e.a. een keer uit te proberen! Let erop dat het niet enkel gaat om receive/transmit van data, maar dat óók de mogelijkheid aanwezig moet zijn om alle modemsignalen (RTS, CTS, RI, DTR, etc.) aan te kunnen sturen. Elk apparaat met een RS232-interface gebruikt die signalen immers op een andere manier, en een serial device server moet dat aankunnen.

Het gebruik van een COM-port redirector is natuurlijk alleen mogelijk op PC's. Om het gebruik met andere soorten apparatuur mogelijk te maken, is het meestal mogelijk om rechtstreeks via TCP/IP de device server aan te spreken. Op de besturing moet dan de mogelijkheid aanwezig zijn om rechtstreeks op "sockets" te werken. Sockets is een benaming voor een interface naar een TCP/IP protocolstack in een apparaat; het is oorspronkelijk ontwikkeld voor Unix maar later ook overgenomen door Microsoft en heeft eigenlijk alle andere interfaces naar TCP/IP van de markt verdreven. Het werken met sockets vereist wel wat kennis van TCP/IP en de bijbehorende familie van protocollen, maar echt moeilijk is het niet. Het gebruik van Ethernet voor vervanging van RS232 poorten ook om andere redenen nog aantrekkelijk, omdat men op deze manier de beperking van de korte afstand (10...60m) die normaliter mogelijk is met RS232 kan omzeilen. Ook is het mogelijk om vrij makkelijk veel RS232 poorten aan een PC te koppelen, omdat er geen insteekkaarten nodig zijn; vooral bij embedded PC's maakt dit een kleine behuizing mogelijk.

De mogelijkheden van een serial device server zijn nog veel uitgebreider als hierboven in kort bestek genoemd. Het valt echter buiten de scope van deze publicatie om hier dieper op in te gaan.

7.3 Analyzers, monitors en sniffers

Iedereen die met een veldbus aan de slag gaat, loopt wel eens tegen problemen op waarvan het niet meteen duidelijk is waar ze vandaan komen. Net zoals een elektronicus een multimeter en een oscilloscoop gebruikt, heeft de netwerkexpert een zgn. "netwerkanalyzer", "sniffer" of "netwerkmonitor" nodig. Deze kunnen laten zien wie er op het netwerk aangesloten zijn, wat voor netwerkberichten ze elkaar sturen en wanneer, wat de inhoud van die berichten is, etc. Analyzers voor Ethernet zijn in ruime mate op de markt te vinden. Vraag maar aan uw LAN-netwerkbeheerder op kantoor of hij de mogelijkheden hier eens van kan laten zien! Ook voor het controleren van de bekabeling van Ethernetnetwerken is genoeg apparatuur te verkrijgen (o.a. van Fluke).

Foundation Fieldbus analyzer

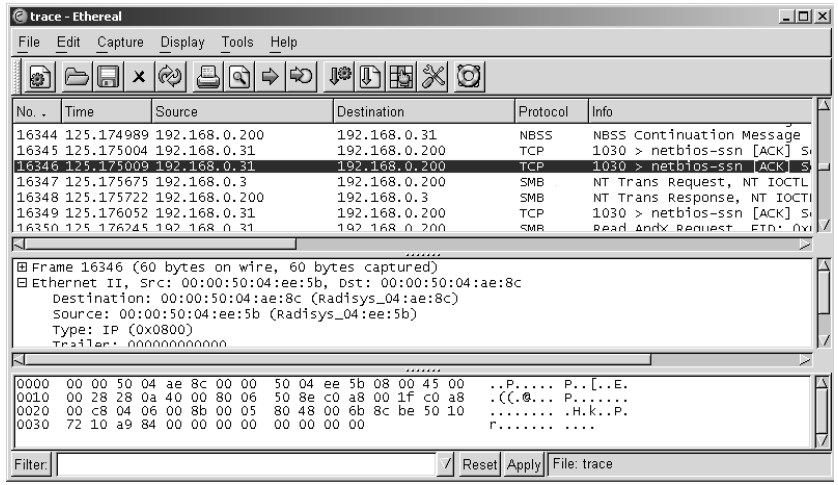
Het nadeel van al deze producten is dat ze gemaakt zijn voor de protocollen die gangbaar zijn in de industriële automatisering. Slechts in een enkel geval zijn er ook speciale analyzers ontwikkeld voor industriële Ethernet protocollen, zoals bijvoorbeeld voor de Foundation Fieldbus HSE (High-speed Ethernet) variant.

Omdat netwerkanalyzers niet in grote aantallen verkocht worden, zijn ze niet goedkoop. Een licentie voor de HSE Toolkit kost initieel $6750, en nog eens $1000 voor elke extra licentie. Zelfs voor een netwerkanalyzer zijn dit prijzen die aan de forse kant zijn (en dan is er niet eens een speciale netwerkkaart nodig). Maar de kostprijs moet afgewogen worden tegen de kosten voor downtijd; en tevens kan een analyzer goede diensten bewijzen voor het opsporen van fouten in applicatieprogrammatuur, het bepalen van responsetijden, en het uitzoeken van interoperabiliteitsproblemen tussen producten van verschillende leveranciers. Uit eigen ervaring blijkt dat een netwerkanalyzer zich snel terugverdient. Ik beschouw het als onmisbare meetapparatuur voor iedereen die professioneel met een netwerk bezig is. Een elektronicus ontzegt men immers ook niet zijn multimeter en oscilloscoop (al zijn die wel wat goedkoper).

De AT-440 software kan op een "gewone" PC met Windows geïnstalleerd worden, maar deze moet wel voorzien van een Ethernet-interface (welke op moderne PC's steeds vaker standaard aanwezig is). Dit is allemaal mogelijk omdat FF HSE gebruik maakt van standaard Ethernet. Hou er wel rekening mee dat de standaard Ethernet-interface van een PC niet in staat is om de bekabeling van het netwerk door te meten en/of de kwaliteit ervan te bepalen. Indien het netwerk zodanig slecht is aangelegd dat een bepaald netwerkbericht niet aankomt op de PC, dan kan het dus ook niet aan de gebruiker getoond worden.

Ethereal

In sommige gevallen is het niet erg om met een meer 'standaard' netwerkanalyzer te werken. Zeker voor software-ontwikkelaars, die vaak toch al gewend zijn om in bytes te rekenen is het niet erg om op laag niveau tegen een netwerk aan te kijken.
Een voorbeeld van een analyzer die dit mogelijk maakt is Ethereal, welke als freeware kostenloos op te halen is vanaf *www.ethereal.com*, en op de meeste PC-Ethernetkaarten kan werken (figuur 7-3). Interessant is tevens dat het pakket beschikt over een module voor Ethernet/IP, Powerlink en Modbus/TCP.



Figuur 7-3: Screendump van Ethereal

Om goed met Ethereal te kunnen werken, moet men de beschikking kunnen hebben over een hub. Alleen op deze manier zal de analyzer alle netwerkberichten kunnen opvangen. Als men van een switch gebruik maakt, zal Ethereal alleen de broadcast-netwerkberichten opvangen, maar alle andere berichten niet. Eventueel kan men een switch gebruiken die beschikt over "port mirroring" functionaliteit (zie hoofdstuk 5). Voordat men met Ethereal aan de slag gaat, moet eerst op de switch de port mirroring functie aangezet worden (dit kan niet met Ethereal zelf). Verder moet de gebruikte netwerkkaart nog de mogelijkheid hebben om in "promiscuous mode" ingesteld te worden, waardoor ook alle netwerkberichten die niet het MAC-adres van de netwerkkaart hebben worden doorgegeven aan Ethereal.

Microsoft Network Monitor

Microsoft levert voor de 'server' variant van Windows NT, XP en 2003 een eigen "Network Monitor". Deze is als optie te verkrijgen bij diverse Windows-varianten.

8. Enkele praktijkaspecten

8.1 Extra deelnemers toevoegen

Het is vrij makkelijk om extra deelnemers op het netwerk toe te voegen: zoek een hub / switch met een vrije poort, en plug de kabel voor de nieuwe deelnemer hier in. Omdat elk Ethernet-apparaat een wereldwijd uniek MAC-adres heeft, zijn geen problemen met dubbel-gebruikte netwerkadressen mogelijk (tenzij men zelf de MAC-adressen toewijst). Indien zowel de hub / switch als de nieuwe deelnemer voorzien zijn van "autonegotiation" mogelijkheden (of eventueel "autosensing", maar dit is minder krachtig) dan kiezen beide partijen hun hoogste gemeenschappelijke bitrate, en vanaf dat moment kan er, wat Ethernet betreft, gewerkt worden. Dat wil nog niet zeggen dat ook een applicatieprogramma meteen netwerkberichten kan gaan versturen, want ook de hogere protocollagen moeten nog geconfigureerd worden. Dit is per protocol steeds anders, het is moeilijk hiervoor algemene richtlijnen te geven.

8.2 Deelnemers verwijderen

Het is mogelijk om op elk moment een deelnemer van het netwerk te verwijderen, zelfs onder spanning. Ethernet zelf voert geen enkele controle uit de aanwezigheid van deelnemers, en geeft dus ook geen foutmelding. Protocollen op hogere niveaus kunnen dit wel doen; hoe dit exact gebeurt is per protocol anders.

8.3 Vervanging van defecte apparatuur

Uiteraard zal er een keer een deelnemer kapot gaan in een netwerk, en deze dient dan vervangen te worden door een nieuw exemplaar. Soms kan dan even niet gecommuniceerd worden. Dit wordt zichtbaar als men de kabel van de nieuwe deelnemer in *een andere* poort prikt dan waar de kabel oorspronkelijk zat. Men kan dan te maken krijgen met een subtiel verschil in werking tussen een hub en een switch.

Indien met een hub gewerkt wordt, dan zal de nieuwe deelnemer onmiddellijk weer op het netwerk actief kunnen worden. Hij ontvangt immers alle aan hem gerichte netwerkverkeer.

Indien met een switch gewerkt wordt, dan zal de nieuwe deelnemer gedurende enkele minuten geen enkel aan hem gericht netwerkbericht kunnen ontvangen. De switch "weet" immers nog niet dat de nieuwe deelnemer op een andere poort zit, en zal dus alle voor hem bestemde netwerkberichten doorsturen op de voorheen gebruikte poort. Pas als de timeout in de switchtabel afloopt (na enkele minuten) zal het foute poortnummer

verwijderd worden; het eerstvolgende netwerkbericht wordt weer op alle poorten doorgestuurd, en dan pas komt het aan op de nieuwe deelnemer.

Deze situatie is te voorkomen door de nieuwe deelnemer direct na zijn opstarten een (willekeurig) netwerkbericht te laten sturen. De switch is dan meteen op de hoogte van de nieuwe poort. Indien met TCP/IP gewerkt wordt, kan men (bv.) met het commando "ping" iets laten sturen. Wat en naar wie is niet belangrijk, zolang de switch het maar ontvangt.

Er zijn echter situaties mogelijk waarbij de nieuwe deelnemer niet in staat is om zelf een netwerkbericht te sturen. Dit komt vooral voor bij master/slave protocollen; een slave moet wachten op een commando van de master voordat deze iets mag sturen. Bestudeer daarom de documentatie van de switch; Ethernet legt niet vast hoe lang een timeout van de switchtabel moet zijn (dit kan per leverancier verschillen). Deze mate van geduld zal men dan moeten uitoefenen na het aansluiten van de nieuwe deelnemer. Indien men dit geduld niet heeft, en de applicatie er tegen kan, zou men eventueel de switch nog even snel kunnen uit- en weer aanzetten. Dan is de switchtabel namelijk ook gewist.

8.4 Toewijzing van IP-adressen

Indien van apparatuur gebruik gemaakt wordt die het TCP/IP protocol uitvoert, dan moeten deze apparaten IP-adressen toegewezen krijgen. Hiervoor bestaan verschillende methodes:

- Statische toewijzing via dipswitches, keyboard, etc.
- Dynamische toewijzing via het BOOTP protocol;
- Dynamische toewijzing via het DHCP protocol;
- Toewijzing via ingebouwde webserver via het TELNET protocol.

Indien van BOOTP of DHCP gebruik gemaakt moet worden, dan moet men ook een BOOTP of DHCP server op het netwerk installeren. Meestal zal dit een PC zijn, die dan voorzien wordt van BOOTP / DHCP server software. Deze is op Internet goed te vinden (overigens ook onderdeel van Windows Server).

Indien van Telnet gebruik gemaakt wordt, krijgt men toegang tot het apparaat via een command-line interface (zoals vroeger bij DOS). Het apparaat heeft een door de leverancier ingesteld default IP-adres. Via een commando is dit dan te wijzigen in iets anders. Uiteraard is er wel eerst een Telnet client programma nodig.

Dat is aanwezig op moderne Windows versies en op alle Unix / Linux versies, en indien niet, dan zijn er op Internet legio andere implementaties te vinden.

DHCP / BOOTP

Indien van TCP/IP gebruik gemaakt wordt, kan van DHCP (Dynamic Host Configuration Protocol) of de voorloper BOOTP (Bootstrap Protocol) gebruik gemaakt worden.
Deze twee protocollen worden vaak gebruikt voor het configureren van een deelnemer *via het netwerk zelf*. Direct na het opstarten stuurt de deelnemer een verzoek om configuratie naar alle aangesloten apparaten op het netwerk. Op één hiervan zal een DHCP of BOOTP server draaien, en deze zal dan de gewenste configuratie opsturen. Aan de hand van het MAC-adres is te bepalen wie er om zijn configuratie vraagt; iedereen krijgt dus zijn "eigen" configuratie.

Voor de gebruiker houdt dit in dat in de DHCP / BOOTP server een lijst aanwezig is van alle op het netwerk voorkomende MAC-adressen. Echter, indien er een deelnemer vervangen wordt, dan zal deze een ander MAC-adres ingebouwd hebben (elke Ethernet-interface heeft immers een eigen MAC-adres). Men kan de nieuwe deelnemer dan wel op het netwerk aansluiten, maar aangezien de DHCP / BOOTP server het nieuwe MAC-adres niet kent, zal er ook geen configuratie opgestuurd worden en de nieuwe deelnemer kan niet actief worden op het netwerk. Wat men dus éérst moet doen is de MAC-tabel van de DHCP / BOOTP server uitbreiden.

Omdat dit soms best lastig is, bieden sommige leveranciers de mogelijkheid dat men zelf een MAC-adres op een apparaat kan instellen. Net voor het aansluiten van de nieuwe deelnemer wordt eerst het 'oude' MAC-adres gekopieerd. Dit vraagt echter wel om accuratesse. Maakt men hier fouten bij, en worden MAC-adressen dubbel gebruikt, dan kan het netwerk ophouden met functioneren (afhankelijk van het gebruikte protocol). Ook de 'oude' deelnemer mag niet meer op het netwerk aangesloten worden zonder eerst een ander MAC-adres toegewezen te zijn. Een strakke administratie van uitgegeven MAC-adressen is derhalve noodzakelijk!

Binnen de industrie is er geen overeenstemming m.b.t. het gebruik van BOOTP of DHCP; de ene leverancier gebruikt BOOTP, en de andere DHCP. Dit is voor gebruikers lastig, aangezien zij dan in de situatie verzeild kunnen raken dat twee servers nodig zijn (voor beide protocollen).

8.5 Uitval van een voeding van een hub of switch

Indien een hub of switch uitvalt, dan heeft dit belangrijke consequenties voor (een deel van) het netwerk. Als spin-in-het web regelt de hub / switch immers alle netwerkverkeer; en dat valt dus geheel stil. Het uitvallen kan ook veroorzaakt worden door de voeding. Daarom hebben veel industriële hubs / switches de mogelijkheid om een dubbele (meestal 24V) voeding aan te sluiten, zodat bij uitval van één voeding toch nog doorgewerkt kan worden.

Het is voor de besturing soms interessant te weten of één of beide voedingen operationeel zijn. In veel gevallen wordt dit aangegeven via potentiaalvrije contacten, die men dan moet verbinden met een digitale input op een I/O module. Het zou echter goed mogelijk zijn om de voedingsstatus ook via het netwerk zélf aan de besturing door te geven (het netwerk ligt er immers voor). Dit is echter niet erg gebruikelijk. De reden hiervoor is dat de hub / switch dan moet weten welk netwerkprotocol er op het netwerk gebruikt wordt; aangezien er heel veel mogelijkheden zijn op protocolgebied zou dit betekenen dat de leverancier evenzoveel varianten van zijn product zou moeten ontwikkelen. Dit is geen haalbare kaart, en daarom wordt dit ook niet gedaan.

Een oplossing die waarschijnlijk wel beter aan zal slaan is het gebruik van SNMP (Simple Network Management Protocol) voor het doorgeven van de status van de voeding(en). Eén van de leveranciers die dit in zijn switches inbouwt is Phoenix Contact. Om goed met SNMP om te kunnen gaan is wel een SNMP-client nodig op de besturing. Meer informatie over de werking van SNMP in de praktijk is waarschijnlijk wel te vinden bij de netwerkbeheerder van uw eigen bedrijf.

8.6 Redundantie

Een ander gevolg van de uitval van een switch is dat een deel van het netwerk stilvalt. Dit is niet altijd acceptabel. Al zeer lang zijn switches op de markt waarbij het mogelijk is om (met een aantal andere switches) een redundante ring te bouwen. De switches zelf coördineren met elkaar hoe de ring moet werken (linksom of rechtsom), en bij een kabelbreuk of switch-uitval zal al het netwerkverkeer de andere kant op gestuurd worden waardoor het toch nog op zijn eindbestemming uitkomt. Hiervoor zijn speciale netwerkprotocollen ontwikkeld, zoals de "STP" (Spanning Tree Protocol) familie en leverancierssspecifieke varianten (zie ook hoofdstuk 5).

Let er op dat niet alle switches voorzien zijn van STP(-achtige) functionaliteit. Indien men dit nodig heeft moet er speciaal op gelet worden bij de aanschaf!

8.7 Hub vervangen door een switch

Het gebruik van een switch in plaats van een hub kan (hoeft niet altijd, zie hoofdstuk 5) belangrijke voordelen bieden m.b.t. de snelheid van een netwerk. Het kan voorkomen dat men eerst een netwerk gebouwd heeft met een hub, maar na enige tijd in performance-problemen komt. Het is dan mogelijk om de hub direct te vervangen door een switch, zonder dat dit verder zichtbaar is voor de applicatie. Of, met andere woorden: in sommige gevallen is het financieel aantrekkelijker om met een hub te beginnen, en alleen als het echt nodig is een switch te gebruiken.

8.8 Netwerkbeheer

Standaard Ethernet heeft geen functionaliteit voor netwerkbeheer, afgezien van ledjes die meestal in de buurt van elke RJ45 connector staan te knipperen. Een geavanceerdere vorm van netwerkbeheer kan natuurlijk in een bepaald protocol zijn ingebouwd, maar vaak wordt ook gebruik gemaakt van SNMP (Simple Network Management Protocol), dat uit de LAN-wereld komt.

Zoals de naam al zegt is SNMP speciaal voor netwerkbeheer ontwikkeld. Oorspronkelijk was het inderdaad "simpel", maar door de jaren heen is het gegroeid en wat minder simpel geworden, maar dat heeft het niet weerhouden om uit te groeien tot 'het' protocol voor netwerkbeheerders. Het protocol is overigens niet gekoppeld aan Ethernet; eigenlijk hoort het meer bij TCP/IP thuis. Maar in veel "managed switches" is het ingebouwd, reden om er enige aandacht aan te besteden.

Met SNMP is het mogelijk om managementinformatie uit apparatuur op te vragen, en om apparatuur te configureren. Dit geschiedt via zgn. "objecten", eigenlijk een foutieve benaming want het heeft niets met object-oriëntatie te maken; ik zal daarom het woord "variabele" gebruiken. Via SNMP kunnen alle variabelen gelezen worden, en sommige ook van waarde veranderd worden. Bijvoorbeeld, de MMS-switches van Phoenix Contact ondersteunen de door SNMP verplichte variabelen (ca. 150), de variabelen voor netwerkbeheer (ca. 60 velden), de variabelen voor switchbeheer (ca 50 velden) en tenslotte nog enige Phoenix-specifieke variabelen (ca. 130 velden). Sommige variabelen worden speciaal ingezet voor het instellen van de bedrijfsparameters van de switch:

denk daarbij o.a. aan de snelheid op elke poort, welk VLAN er gebruikt wordt, of een poort misschien uitgezet moet worden, gebruik van redundante bekabeling, omschakeltijden in geval van fouten, etc. Overigens is men zelden verplicht om van SNMP gebruik te maken: meestal kunnen ook via een RS232-interface of een ingebouwde webserver of een Telnet-server alle variabelen opgevraagd c.q. ingesteld worden.

Het kunnen aanleveren van de benodigde informatie vereist natuurlijk wel software-ondersteuning in een switch. Deze functionaliteit heet: "SNMP Agent" ('agent' is een informatica-term voor een autonoom opererend stuk software). De taak van de agent is om de benodigde managementinformatie op te vragen uit de hardwarecircuits van de switch, deze bij te houden, en op te sturen aan iedereen die er om vraagt. Hiervoor biedt het SNMP-protocol uitgebreide mogelijkheden. Tevens kan de agent ongevraagd (event-driven dus) informatie opsturen, dit wordt een zgn. "SNMP Trap" genoemd. Ergens op het netwerk moet natuurlijk nog wel een SNMP-ontvanger aanwezig zijn, die iets doet met de ontvangen traps. De realisatie hiervan is de verantwoordelijkheid van de gebruiker; bv. men zou SNMP traps kunnen gebruiken als diagnostisch middel in een applicatie.

Welke variabelen een SNMP agent aanlevert, is deels genormaliseerd, en deels te bepalen door de leverancier. De structuur van deze informatie is ook genormaliseerd in een zgn. "MIB" – Management Information Base. De MIB is eigenlijk niets meer dan een opsomming van de velden in een apparaat die via het netwerk te benaderen (lezen/schrijven) zijn. Per veld is vermeld wat zijn betekenis is, hoe groot het is, wat de specifieke kenmerken zijn, etc. Bijvoorbeeld: veld (object) 1.3.6.1.4.1.4346.11.11.4.1.5 is een Integer32, read-only, en geeft aan of de IP-protocol parameters wel (waarde 1) of niet (waarde 2) opgeslagen zijn in niet-vluchtig geheugen. Elke leverancier van een apparaat met een SNMP-agent dient deze "MIB" aan te leveren. De exacte inhoud kan men terugvinden in de documentatie van de leverancier. Enige kennis van SNMP wordt altijd wel verondersteld aanwezig te zijn bij de lezer. Wilt u hierover meer weten, kijk dan op *www.snmp.org/protocol/*. Iets dichter bij huis: aan de TU Twente wordt ook het nodige onderzoekswerk aan SNMP verricht, zie hiervoor *www.simpleweb.org*.

9. Meer informatie

9.1 Literatuur

Ethernet norm
De norm voor Ethernet staat officieel bekend als de IEEE 802.3. De IEEE (Institute of Electrical and Electronic Engineers) heeft in 2001 besloten deze, en alle andere IEEE 802 normen, gratis aan geïnteresseerden ter beschikking te stellen.
De documenten zijn zes maanden na hun (papieren) publicatie beschikbaar op *http://standards.ieee.org/getieee802* in de vorm van een PDF-bestand. Deze beslissing moet de verspreiding van de IEEE 802 normen bevorderen, waardoor hun intrinsieke waarde nog verder toeneemt. Het gaat hier overigens wel om een proef.

Ethernet definitive guide
Beter leesbaar is het boek "Ethernet, the definitive guide" van Charles Spurgeon (ISBN 1-56592-660-9, prijs ca. $50), welke vrij uitvoerig ingaat op de diverse bekabelingsvarianten van Ethernet met alle hebbelijkheden en onhebbelijkheden. Het is een van de betere boeken op dit gebied. Helaas wordt er niets verteld over shielded twisted pair en andere industriële specialiteiten.

Pocket Guide
Een klein boekje over industrieel Ethernet komt va de hand van Perry Marshall en is getiteld "Industrial Ethernet: A Pocket Guide". volgens de inhoudsopgave wordt een indrukwekkende lijst onderwerpen besproken, maar dat is gezien de omvang van het boek (200 pagina's op A7 formaat) niet al te diepgravend (de benaming "pocket guide" dient in dit geval letterlijk opgevat te worden!) ISA Press, ISBN 1-55617-774-7, prijs $34.95.
Zie ook *http://www.perrymarshall.com/ethernet/*.

Fuller
Een van de eerste boeken over industrieel Ethernet is geschreven door F.J. Fuller, getiteld "Ethernet-TCP/IP für die Industrieautomation". Door middel van simulaties poogt de auteur aan te tonen of Ethernet deterministisch genoeg is, en hoe het worst-case presteert. Uitgever is Hüthig Verlag, ISBN 3-7785-2641-3. In de tweede druk (ISBN 3-7785-2779-3) wordt ook nog dieper ingegaan op "quality of service" mogelijkheden, IPV6, redundantie, safety en software-architecturen.

Ethernet Planning & Installation Guide
De vereniging IAONA (Industrial Automation Open Networking Association, waarbij Open = Ethernet) heeft een document uitgegeven genaamd "Industrial Ethernet Planning & Installation Guide". Zoals de titel al aangeeft beschrijft het document de aanleg van alle onderdelen van een Ethernet. Dit is overigens geen nieuwe materie; de uit de LAN-wereld bekende normen EN 50173 en IEC 11801 kunnen grotendeels overgenomen worden voor de planning en aanleg van een industrieel Ethernet. De LAN-achtergrond is voor een deel ook wel in het document terug te zien: blijkbaar is het gebaseerd op een ander document dat vooral de aanleg van een Ethernet in kantoorgebouwen en universiteiten beschrijft.

In een viertal hoofstukken worden achtereenvolgens de factoren beschreven die de opbouw van een netwerk bepalen, gevolgd door de installatieregels, hoe het netwerk doorgemeten moet worden, hoe de maximale omvang uitgerekend kan worden, en hoe met twisted-pair of glasvezel omgegaan moet worden. Het IAONA document is kostenloos op te halen vanaf *www.iaona.org*. Er wordt wel vanuit gegaan dat de lezer bekend is met Ethernet in al zijn aspecten en varianten, en wat hubs / switches / routers / ISO / IEEE802.3p etc. zijn.

Bundeling artikelen
De Duitse uitgever Vogel Verlag geeft in 2004 voor de 4e keer een bundel uit met daarin alle artikelen over industrieel Ethernet die in het afgelopen jaar in het tijdschrift "Praxis Profiline" verschenen zijn. De oorspronkelijk Duitstalige artikelen zijn ook in het Engels afgedrukt. Het is een snelle manier om alle informatie over hetzelfde onderwerp te vinden. Omvang ca. 100 pagina's, ISBN 3-8259-1918-8, prijs 21 Euro.

ProfiNet specificatie
De Profibus gebruikersvereniging (PI – Profibus International, *www.profibus.com*) staat bij wijze van uitzondering toe dat de ProfiNet 2.0 specificatie ook door niet-leden van PI opgevraagd kan worden. De genoemde specificatie is geen document dat voor eindgebruikers bedoeld is, want er wordt een stevige kennis van het Microsoft Windows platform vereist.

Industrial Ethernet Book
Inmiddels is ook al weer de 18e uitgave verschenen van het "Industrial Ethernet Book" (zie *http://ethernet.for-industry.com*). Dit is, in tegenstelling tot de naam, een tijdschrift dat elk kwartaal verschijnt. Het is gevuld met een aantal redactionele artikelen over nieuwe ontwikkelingen rondom Ethernet, en een productencatalogus.

Vanwege de sterke groei in het aantal producten wordt de catalogus niet meer in zijn geheel afgedrukt. Het tijdschrift is gratis, en via de genoemde website kan men zich aanmelden.

9.2 Verenigingen

Er is op dit moment slechts één vereniging die zich bezighoudt met de ontwikkelingen rondom industrieel Ethernet, en dat is de IAONA (Industrial Automation Open Networking Association), welke oorspronkelijk een Europese en een Amerikaanse vestiging had, maar deze laatste bestaat niet meer. Een andere vereniging, de "Industrial Ethernet Association", is inmiddels ook ter ziele.

De website van de IAONA is te vinden op *www.iaona.org* en hierop kan men ook de bekabelingsrichtlijnen vinden, die geschreven zijn door een van de werkgroepen. Eind 2003 is de vierde versie verschenen. Ook wordt nog gewerkt aan de beveiligingsrichtlijnen.

Een Nederlandse IAONA is sinds medio 2003 in oprichting, maar heeft nog geen activiteiten ontplooid. De Zwitserse IAONA (*www.iaona.ch*) is zeer actief, en op de website is veel materiaal te vinden.

De IAONA voert zelf geen ontwikkelwerk uit, maar poogt de ontwikkelingen bij andere verenigingen (Profibus, ODVA, Foundation Fieldbus, Powerlink, Modbus, etc.) te coördineren. Tot nu toe met redelijk succes, alhoewel e.e.a. alleen op basis van vrijwilligheid te regelen valt. Dat leidt wel eens tot tegenslagen, bijvoorbeeld de werkgroep "Real-time Protocollen" is door de leveranciers aan alle kanten gepasseerd, en de werkgroep heeft zichzelf min-of-meer opgeheven.

9.3 Trainingen en cursussen

In Nederland worden bedrijfsonafhankelijke trainingen op het gebied van industrieel Ethernet verzorgd door het Mikrocentrum te Eindhoven (*www.mikrocentrum.nl*), en bedrijfs- c.q. productspecifieke trainingen door alle grote leveranciers. De duur van deze trainingen kan van één tot enkele dagen variëren, en de diepgang is daarom ook zeer verschillend. Ook is er veel verschil m.b.t. een cursus voor gebruik van Ethernet op hogere niveaus in de automatiseringshiërarchie, en cursussen voor gebruik van Ethernet op de laagste niveaus (rechtstreeks als concurrent van bestaande industriële netwerken).

Trainingen over Ethernet worden ook gegeven door bedrijven die actief zijn in de administratieve automatisering en LAN's. Hier komen uiteraard geen industriële aspecten aan bod, en zal sterk de nadruk op de software, protocollen, configuratie van Windows etc. gelegd worden.

In cursussen over Ethernet komen vaak onderwerpen aan bod komen die niet rechtstreeks Ethernet-gerelateerd zijn maar er wel veel mee te maken hebben, zoals TCP/IP en de hoger liggende protocollen. Indien men specifiek in TCP/IP geïnteresseerd is, kan men beter een cursus volgen voor dit onderwerp alleen, dan als onderdeel van een sessie over industrieel Ethernet.

Tenslotte is net over de grens, in Aken, ComConsult (*http://www.comconsult-akademie.de*) actief, welke veel trainingen op het gebied van (industrieel) Ethernet en netwerkprotocollen verzorgt. Ook kan men zich abonneren op een technisch goed gevulde email-nieuwsbrief.

9.4 Websites

Informatie over industrieel Ethernet op het www is er genoeg te vinden (probeer op Google maar eens te zoeken met de zoektermen "Ethernet industrial ppt"!).

Het probleem met het opsommen van websites is dat zulke lijsten sterk verouderen. Een actuele versie van de gegeven lijst is daarom te vinden op *http://ourworld.cs.com/rahulsebos.*

De mogelijke inzet van Ethernet als veldbus is een technologie die nog in volle ontwikkeling is. Het gebruik van Ethernet in industriële applicaties is natuurlijk al langer gebruikelijk, maar niet op de lagere netwerkniveaus. Hierin gaat snel verandering komen. De fusie van al bestaande technieken uit de IT in combinatie met real-time netwerkprotocollen en innovatieve manieren van bekabelen levert een geheel nieuw type Ethernet op.

Rob Hulsebos (1961, HTS Informatica), is in het dagelijkse leven R&D medewerker bij Assembléon. Hij is al sinds 1990 actief betrokken bij de ontwikkeling en het gebruik van industriële netwerken, in eerste instantie bij een PLC-leverancier, maar vanaf 1996 puur als "gebruiker". Zijn praktijkervaring op dit gebied heeft geleid tot meer dan 130 publicaties in de Nederlandse, Duitse en Engelse vakpers, waaronder de maandelijkse column in het tijdschrift *Automatie*. Tevens is hij de auteur van het standaardwerk *Veldbussen* en het *Dossier Industrieel Ethernet*, mede-auteur van het 2000 pagina's dikke naslagwerk *Industriële Netwerken*, mede-auteur van het *Sensorenboek*, en auteur van de veldbuscursussen bij de Open Universiteit, SBC en Dirksen.
Daarnaast verzorgt hij open- en in-company trainingen over industriële netwerken in het algemeen, en over AS-Interface, Profibus, industrieel Ethernet en Modbus in het bijzonder.