Global Network System

Sam Kelley

GLOBAL NETWORK SYSTEM

GLOBAL NETWORK SYSTEM

Sam Kelley

BIBLIOTEX
Digital Library

Global Network System
by Sam Kelley

www.bibliotex.com

Ebook ISBN: 9781984664105

Published by:
Bibliotex
Canada
Website: www.bibliotex.com

Contents

1

Global Network

The Internet is a global network of computers. Every computer that is connected to the Internet is considered a part of that network. This means even your home computer. It's all a matter of degrees, you connect to your ISP's network, then your ISP connects to a larger network and so on. At the top of the tree is the high-capacity backbones, all of these interconnect at 'Network Access Points' 'NAPs' at important regions around the world. The entire Internet is based on agreements between these backbone providers who set in place all the fibre optics lines and other technical aspects of the Internet. The first high speed backbone was created by the 'National Science Foundation' in 1987.

The Internet was first created by the Advanced Research Projects Agency (ARPA) of the U.S. government in 1960's,

and was first known as the ARPANet. At this stage the Internet's first computers were at academic and government institutions. They were mainly used for accessing files and to send e-mail. From 1983 onwards the Internet as we know it today started to form with the introduction of the communication protocol TCP/IP to ARPANet.

Since 1983 the Internet has accommodated alot of changes and continues to keep developing. The last two decades has seen the Internet accommodate such things as network LANs and ATM and frame switched services. The Internet continues to evolve with it becoming available on mobile phones and pagers and possibly on televisions in the future.

The actual term "Internet" was finally defined in 1995 by FNC (The Federal Networking Council). The resolution created by the The Federal Networking Council (FNC) agrees that the following language reflects our definition of the term "Internet". "Internet" refers to the global information system that,

EVOLUTION

The underpinnings of the Internet are formed by the global interconnection of hundreds of thousands of otherwise independent computers, communications entities and information systems. What makes this interconnection possible is the use of a set of communication standards, procedures and formats in common among the networks and

the various devices and computational facilities connected to them. The procedures by which computers communicate with each other are called "protocols." While this infrastructure is steadily evolving to include new capabilities, the protocols initially used by the Internet are called the "TCP/IP" protocols, named after the two protocols that formed the principal basis for Internet operation.

On top of this infrastructure is an emerging set of architectural concepts and data structures for heterogeneous information systems that renders the Internet a truly global information system. In essence, the Internet is an architecture, although many people confuse it with its implementation. When the Internet is looked at as an architecture, it manifests two different abstractions. One abstraction deals with communications connectivity, packet delivery and a variety of end-end communication services. The other abstraction deals with the Internet as an information system, independent of its underlying communications infrastructure, which allows creation, storage and access to a wide range of information resources, including digital objects and related services at various levels of abstraction.

Interconnecting computers is an inherently digital problem. Computers process and exchange digital information, meaning that they use a discrete mathematical "binary" or "two-valued" language of 1s and 0s. For communication

purposes, such information is mapped into continuous electrical or optical waveforms. The use of digital signaling allows accurate regeneration and reliable recovery of the underlying bits. We use the terms "computer," "computer resources" and "computation" to mean not only traditional computers, but also devices that can be controlled digitally over a network, information resources such as mobile programs and other computational capabilities.

The telephone network started out with operators who manually connected telephones to each other through "patch panels" that accepted patch cords from each telephone line and electrically connected them to one another through the panel, which operated, in effect, like a switch. The result was called circuit switching, since at its conclusion, an electrical circuit was made between the calling telephone and the called telephone.

Conventional circuit switching, which was developed to handle telephone calls, is inappropriate for connecting computers because it makes limited use of the telecommunication facilities and takes too long to set up connections. Although reliable enough for voice communication, the circuit-switched voice network had difficulty delivering digital information without errors.

For digital communications, packet switching is a better choice, because it is far better suited to the typically "burst" communication style of computers. Computers that

communicate typically send out brief but intense bursts of data, then remain silent for a while before sending out the next burst. These bursts are communicated as packets, which are very much like electronic postcards. The postcards, in reality packets, are relayed from computer to computer until they reach their destination.

The special computers that perform this forwarding function are called variously "packet switches" or "routers" and form the equivalent of many bucket brigades spanning continents and oceans, moving buckets of electronic postcards from one computer to another. Together these routers and the communication links between them form the underpinnings of the Internet.

Without packet switching, the Internet would not exist, as we now know it. Going back to the postcard analogy, postcards can get lost. They can be delivered out of order, and they can be delayed by varying amounts. The same is true of Internet packets, which, on the Internet, can even be duplicated. The Internet Protocol is the postcard layer of the Internet. The next higher layer of protocol, TCP, takes care of re-sending the "postcards" to recover packets that might have been lost, and putting packets back in order if they have become disordered in transit.

Of course, packet switching is about a billion times faster than the postal service or a bucket brigade would be. It also has to operate over many different communications systems,

or substrata. The authors designed the basic architecture to be so simple and undemanding that it could work with most communication services.

Many organizations, including commercial ones, carried out research using the TCP/IP protocols in the 1970s. E-mail was steadily used over the nascent Internet during that time and to the present. It was not until 1994 that the general public began to be aware of the Internet by way of the World Wide Web application, particularly after Netscape Communications was formed and released its browser and associated server software.

Thus, the evolution of the Internet was based on two technologies and a research dream. The technologies were packet switching and computer technology, which, in turn, drew upon the underlying technologies of digital communications and semiconductors. The research dream was to share information and computational resources. But that is simply the technical side of the story. Equally important in many ways were the other dimensions that enabled the Internet to come into existence and flourish. This aspect of the story starts with cooperation and far-sightedness in the U.S. Government, which is often derided for lack of foresight but is a real hero in this story.

It leads on to the enthusiasm of private sector interests to build upon the government funded developments to expand the Internet and make it available to the general public.

Perhaps most important, it is fueled by the development of the personal computer industry and significant changes in the telecommunications industry in the 1980s, not the least of which was the decision to open the long distance market to competition. The role of workstations, the Unix operating system and local area networking (especially the Ethernet) are themes contributing to the spread of Internet technology in the 1980s into the research and academic community from which the Internet industry eventually emerged.

Many individuals have been involved in the development and evolution of the Internet covering a span of almost four decades if one goes back to the early writings on the subject of computer networking by Kleinrock, Licklider, Baran, Roberts, and Davies.

The ARPANET, described below, was the first wide-area computer network. The NSFNET, which followed more than a decade later under the leadership of Erich Bloch, Gordon Bell, Bill Wulf and Steve Wolff, brought computer networking into the mainstream of the research and education communities. It is not our intent here to attempt to attribute credit to all those whose contributions were central to this story, although we mention a few of the key players

ROLE OF DARPA

Modern computer networking technologies emerged in the early 1970s. In 1969, The U.S. Defence Advanced Research Projects Agency (variously called ARPA and DARPA), an

agency within the Department of Defence, commissioned a wide-area computer network called the ARPANET. This network made use of the new packet switching concepts for interconnecting computers and initially linked computers at universities and other research institutions in the United States and in selected NATO countries. At that time, the ARPANET was essentially the only realistic wide-area computer network in existence, with a base of several dozen organizations, perhaps twice that number of computers and numerous researchers at those sites. The program was led at DARPA by Larry Roberts. The packet switches were built by Bolt Beranek and Newman (BBN), a DARPA contractor. Others directly involved in the ARPANET activity included the authors, Len Kleinrock, Frank Heart, Howard Frank, Steve Crocker, Jon Postel and many many others in the ARPA research community. Back then, the methods of internetworking (that is interconnecting computer networks) were primitive or non-existent. Two organizations could interwork technically by agreeing to use common equipment, but not every organization was interested in this approach. Absent that, there was jury-rigging, special case development and not much else. Each of these networks stood on its own with essentially no interaction between them-a far cry from today's Internet.

In the early 1970s, ARPA began to explore two alternative applications of packet switching technology based on the

use of synchronous satellites (SATNET) and ground-based packet radio (PRNET). The decision by Kahn to link these two networks and the ARPANET as separate and independent networks resulted in the creation of the Internet program and the subsequent collaboration with Cerf. These two systems differed in significant ways from the ARPANET so as to take advantage of the broadcast and wireless aspects of radio communications.

The strategy that had been adopted for SATNET originally was to embed the SATNET software into an ARPANET packet switch, and interwork the two networks through memory-to-memory transfers within the packet switch. This approach, in place at the time, was to make SATNET an "embedded" network within the ARPANET; users of the network would not even need to know of its existence. The technical team at Bolt Beranek and Newman (BBN), having built the ARPANET switches and now building the SATNET software, could easily produce the necessary patches to glue the programs together in the same machine.

Indeed, this is what they were under contract with DARPA to provide. By embedding each new network into the ARPANET, a seamless internetworked capability was possible, but with no realistic possibility of unleashing the entrepreneurial networking spirit that has manifest itself in modern day Internet developments. A new approach was in order.

The Packet Radio (PRNET) program had not yet gotten underway so there was ample opportunity to change the approach there. In addition, up until then, the SATNET program was only an equipment development activity. No commitments had been obtained for the use of actual satellites or ground stations to access them. Indeed, since there was no domestic satellite industry in the U.S. then, the only two viable alternatives were the use of Intelsat or military satellites. The time for a change in strategy, if it was to be made, was then.

NETWORKING

For such an extensive and involved subject, which includes so many different technologies, hardware devices and protocols, the definition of networking is actually quite simple. A network is simply a collection of computers or other hardware devices that are connected together, either physically or logically, using special hardware and software to allow them to exchange information and cooperate. Networking is the term that describes the processes involved in designing, implementing, upgrading, managing and otherwise working with networks and network technologies.

Networks are used for an incredible array of different purposes. In fact, the definitions above are so simple for the specific reason that networks can be used so broadly, and can allow such a wide variety of tasks to be accomplished.

While most people learning about networking focus on the interconnection of PCs and other "true" computers, you use various types of networks every day. Each time you pick up a phone, use a credit card at a store, get cash from an ATM machine, or even plug in an electrical appliance, you are using some type of network.

In fact, the definition can even be expanded beyond the world of technology altogether: I'm sure you've heard the term "networking" used to describe the process of finding an employer or employee by talking to friends and associates. In this case too, the idea is that independent units are connected together to share information and cooperate.

The widespread networking of personal computers is a relatively new phenomenon. For the first decade or so of their existence, PCs were very much "islands unto themselves", and were rarely connected together. In the early 1990s, PC networking began to grow in popularity as businesses realised the advantages that networking could provide. By the late 1990s, networking in homes with two or more PCs started to really take off as well.

This interconnection of small devices represents, in a way, a return to the "good old days" of mainframe computers. Before computers were small and personal, they were large and centralized machines that were shared by many users operating remote terminals. While having all of the computer power in one place had many disadvantages, one benefit was that all

users were connected because they shared the central computer. Individualized PCs took away that advantage, in favour of the benefits of independence. Networking attempts to move computing into the middle ground, providing PC users with the best of both worlds: the independence and flexibility of personal computers, and the connectivity and resource sharing of mainframes. in fact, networking is today considered so vital that it's hard to conceive of an organization with two or more computers that would not want to connect them together!

TYPES

The word "networking" strikes fear into legions of otherwise intrepid job hunters. It conjures up one of three images: calling everyone in your address book to see whether anyone knows anyone who may know anyone who may give you an informational interview, attending club meetings with other job hunters or having your friends come up with ideas and contacts instead of advising you to get out there and network. But there is no need to fear because there are many different types of networking techniques you can use to help with your job search and references. Obviously number three doesn't always work. But the other two aren't so appealing either. Who enjoys calling total strangers and asking them to meet with you when there's nothing in it for them? If they're successful in what they do-and why else would you want to meet with them?-they're likely to be busy doing it.

STRONG-CONTACT NETWORKS

Structured explicitly to pass business referrals among members, they allow only one member per profession. Strong-contact networks are particularly good for developing in-depth relationships because you see the same members week after week and pass referrals as a part of each meeting.

CASUAL-CONTACT NETWORKS

Bring businesspeople together in a less-structured context than strong-contact networks, but for many they are a primary source of referrals; membership is not limited by profession. These groups are good for developing breadth in your network, but deep, long-lasting relationships can be formed as well.

SERVICE ORGANIZATIONS

Associations that exist to provide and support humanitarian efforts and good works in the community and larger venues. They also bring people together in settings that facilitate referral and knowledge networking. Like casual-contact groups, they help you add breadth and diversity to your network.

PROFESSIONAL ASSOCIATIONS

Established to exchange information and ideas among those in a given industry, as well as to promote and support that industry. These networks often include direct competitors, but they also provide contacts in related, non-competing businesses as well.

SOCIAL/BUSINESS ORGANIZATIONS

Combine social activities with business networking and can provide a variety of networking opportunities; many tend to resemble singles bars.

WOMEN'S NETWORKING GROUPS

Still important networking organizations, but are slowly disappearing as women enter the business mainstream, especially as professionals, entrepreneurs, and small-business owners.

ONLINE NETWORKS -

A new phenomenon covering a wide range of interests.

TYPES OF COMPUTER NETWORKS

A computer network is two or more computers connected together using a telecommunication system for the purpose of communication and sharing resources". Ask any computer network expert to simplify this definition to you and you will start a debate on how it should not be just two computers but three.

Simply put a network is a means of communication between computers.

Within a given network, computers can send files, e-mails and other correspondence to each other. Even things like instant messaging, is set up within a computer's network.

LAN

LAN or Local Area Network is the most common kind of network set up. There are two ways to connect a LAN network. The simplest and easiest way is the peer-to-peer connection network. This is when two or more computers are directly connected to each other. For example if there were four computers in the network, computer 1 would be connected to computer 2, computer 2 would be connected to computer 3 and computer 3 would be connected to computer 4. This means each computer is dependent on the other. And if there were a network problem with any one computer, all of them would be affected. The other type if the client server connection. This is the type of connection where all the computers in a given network are connected to one central computer. This is a more complicated network but one that is much more efficient that peer-to-peer.

A local area network, or LAN, is a network of connected computers in a room, building, or set of buildings. Local area networks have been around since the beginning of computer use. A LAN is defined as a user network whereby data is sent at high rates between people located relatively close to each other. LANs do not usually make use of leased communication lines, but only means of communication that are provided by the installer of the network.

The Internet is a wide area network, or WAN, which is distinct from a LAN. In contrast to the term Internet, local

area networks are often called intranets, though sometimes this term refers to a cluster of LANs associated with a particular company or organization but not connected to the larger Internet. A local area network uses a hub or router to connect computers together. The means of communication is the omnipresent Ethernet cable or wireless wi-fi technology. These technologies offer data transfer rates running between 10 to 10000 Mbit/s.

Larger, more important LANs have redundant lines or other backup protocols. In networked computers, the most popular communication protocol is TCP/IP. Smaller LANs may be temporary and used between friends to play computer games over the network. Over a network, users can share files, view files, make changes to data on other computers if permitted, play movies or music on multiple computers at once, chat with instant messaging, send e-mails to each other, play games, and so on. All the advantages of the Internet apply, although they only include others on the LAN, and the data transfer rates are high.

Perhaps the most frequently employed use of a LAN is to connect users to the Internet with only one connected router. In modern times, we use broadband cable or DSL modems to connect to the Internet, and it would be clumsy to have a modem associated with every computer, so we simply plug the modem into a router and link the router to computers with Ethernet cables. Configuring a LAN can be intimidating

at first, but contemporary operating systems have programs that do most of the necessary configurations automatically, so setting up a local area network is pretty easy.

A local area network (LAN) supplies networking capability to a group of computers in close proximity to each other such as in an office building, a school, or a home. A LAN is useful for sharing resources like files, printers, games or other applications. A LAN in turn often connects to other LANs, and to the Internet or other WAN.

Most local area networks are built with relatively inexpensive hardware such as Ethernet cables, network adapters, and hubs. Wireless LAN and other more advanced LAN hardware options also exist.

Examples

The most common type of local area network is an Ethernet LAN. The smallest home LAN can have exactly two computers; a large LAN can accommodate many thousands of computers.

Many LANs are divided into logical groups called subnets. An Internet Protocol (IP) "Class A" LAN can in theory accommodate more than 16 million devices organized into subnets.

The High Performance Network

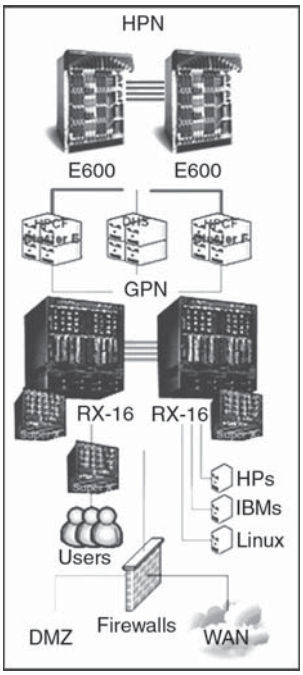
The HPN is used for the exchange of large amounts of operational data. Two Force10 E600 Routers, interconnected via 4-way 10-Gigabit Ethernet aggregated

links, provide connectivity between the High Performance Computing Facility (HPCF) and the Data-Handling System (DHS). The HPCF network nodes are connected via 10-Gigabit Ethernet and all DHS nodes via Gigabit Ethernet aggregated links.

The General Purpose Network

The GPN is used for all other traffic. It has at its core two Foundry BigIron RX-16 routers and at the edge seven Foundry Super-X switches.

The core routers are interconnected via 4-way 10-Gigabit Ethernet aggregated links and have multiple Gigabit Ethernet uplinks to the edge routers. The core also includes two further Super-X switches that are dual-attached to the RX-16s via 10-Gigabit Ethernet.



The GPN provides connectivity to:

- The HPCF, the DHS and additional servers via Gigabit Ethernet ports in the core.
- The user desktops and laptops via Gigabit Ethernet ports on the edge switches.
- The firewalls (for the Wide Area Network and the Demilitarized Zone (DMZ) via Gigabit Ethernet ports in the core. The DMZ includes ECaccess, web servers, the mail gateway and DNS (Domain Name Servers).

The Hardware

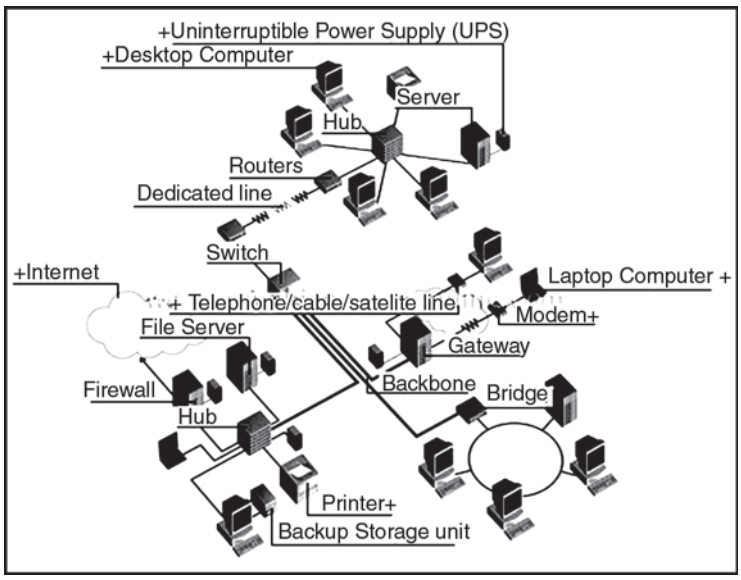
Both the Force10 E600 chassis are populated with 24 10-Gigabit-Ethernet and 144 Gigabit Ethernet ports. For resiliency there are four power supplies, two CPU modules and nine switching fabric modules. Both the RX-16 chassis are populated with 8 10-Gigabit Ethernet ports and 144 Gigabit Ethernet ports. For resiliency there are seven power supplies, two CPU modules and four switching fabric modules. The Super-X chassis each contain up to 156 Gigabit Ethernet ports and dual power-supplies.

WAN

Definition

The wide area network, often referred to as a WAN, is a communications network that makes use of existing technology to connect local computer networks into a larger working network that may cover both national and

international locations. This is in contrast to both the local area network and the metropolitan area network, which provides communication within a restricted geographic area. Here is how the wide area network functions, and why it is so important to communications today.



The concept of linking one computer network with another is often desirable, especially for businesses that operate a number of facilities. Beginning with the local area network and going up to the wide area network, this is most easily accomplished by using existing telephony technology. Essentially, fibre optics are used to create the link between networks located in different facilities.

Often, this means using standard phone lines, referred to as POTS, or employing PSTN (public switched telephone network) technology. During the 1990s, a third option, that of ISDN (integrated services digital network) solutions for

creation a wide area network gained a great deal of popularity, mainly because the concept made it more cost effective to extend the network beyond national boundaries.

With coverage in a broad area, a wide area network allows companies to make use of common resources in order to operate. For example, many retail drugstores make use of a wide area network as part of their support to customers who fill prescriptions with one of their stores. Once in the common customer database for the pharmacy, the client is free to fill a prescription at any of the company's locations, even while vacationing in another state.

Companies also make good use of the wide area network as well. Internal functions such as sales, production and development, marketing and accounting can also be shared with authorized locations through this sort of broad area network application. The concept of a wide area network as a means of taking individual location based computer networks and using them to create a unified computer network for the entire corporation means that employees can work from just about anywhere. Should one facility be damaged or rendered inaccessible due to natural disaster, employees simply move to another location where they can access the unified network, and keep on working.

OVERVIEW

WAN or Wide Area Network is when several LANs or independent computers are connected to a single, wider

network. The Internet is the perfect example of WAN. E-mails, Chat Rooms and IMs all connect to the WAN of the Internet. WAN is much more complex and requires connecting devices or hubs from all over the world.

The term Wide Area Network (WAN) usually refers to a network which covers a large geographical area, and use communications circuits to connect the intermediate nodes. A major factor impacting WAN design and performance is a requirement that they lease communications circuits from telephone companies or other communications carriers. Transmission rates are typically 2 Mbps, 34 Mbps, 45 Mbps, 155 Mbps, 625 Mbps (or sometimes considerably more).

Numerous WANs have been constructed, including public packet networks, large corporate networks, military networks, banking networks, stock brokerage networks, and airline reservation networks. Some WANs are very extensive, spanning the globe, but most do not provide true global coverage. Organizations supporting WANs using the Internet Protocol are known as Network Service Providers (NSPs). These form the core of the Internet.

Wide Area Networks, or WANs, connect a geographically diverse group of computers within a state, country, or even across several states or countries. WANs typically are connected by telephone lines, other types of communication lines, or radio waves. Quite often, smaller local area networks (LANs) are linked together to form a WAN. This is

accomplished via dedicated private lines, leased from telecommunications firms like Sprint and ATandT, or by Switched Multi-Megabit Data Services (SMDS) technology, developed in 1995 to eliminate the need for a leased line.

WAN technology has been refined over a period of several decades. It first emerged in the mid-twentieth century with the advent of networks like ARPAnet. Developed in 1969 by the Department of Defence, ARPAnet and several other networks eventually evolved into the Internet, the largest WAN in the world. The packet switching technology most commonly used with WANs surfaced in the 1960s, and standard packet switching protocol, known as X.25, was developed in 1976. To increase network speed, packet switching allows for the parceling of data into smaller chunks, known as packets, prior to transmission. These packets can travel independently via alternate routes, and they are reassembled once they reach their target. Although X.25 remained the most popular WAN packet switching protocol for years, other packet switching protocols used with increasing frequency by WAN developers and administrators include the Internet standard, Transmission Control Protocol/Internet Protocol (TCP/IP), and Frame Relay, used most often by WANs connected via high speed T-1 and T-3 lines.

WANs are used for a variety of purposes. A corporation with offices in several locations may use a WAN to form an

intranet. Quite often, the individual offices will use their own LANs for things like internal messaging, data processing functions, and hardware and software sharing. When these LANs are joined together to form a WAN, similar data sharing and messaging capabilities become possible across a much broader geographic area.

Businesses wanting to link up with their suppliers or distributors may create a WAN as a means of establishing an extranet. For example, an extranet could provide a sales representative with electronic access to information in about the time it might take to deliver a product, or the availability of a product. Some WANs bring together various types of communications, such as data, video, and voice. Some organizations, including companies, universities, research centers, hospitals, and libraries, use WANs to connect to the Internet.

By connecting the NSP WANs together using links at Internet Packet Interchanges (sometimes called "peering points") a global communication infrastructure is formed. NSPs do not generally handle individual customer accounts (except for the major corporate customers), but instead deal with intermediate organizations whom they can charge for high capacity communications. They generally have an agreement to exchange certain volumes of data at a certain "quality of service" with other NSPs. So practically any NSP can reach any other NSP, but may require the use of one or

more other NSP networks to reach the required destination. NSPs vary in terms of the transit delay, transmission rate, and connectivity offered.

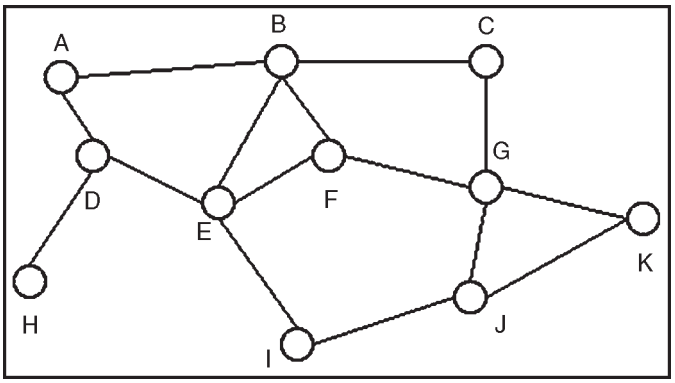
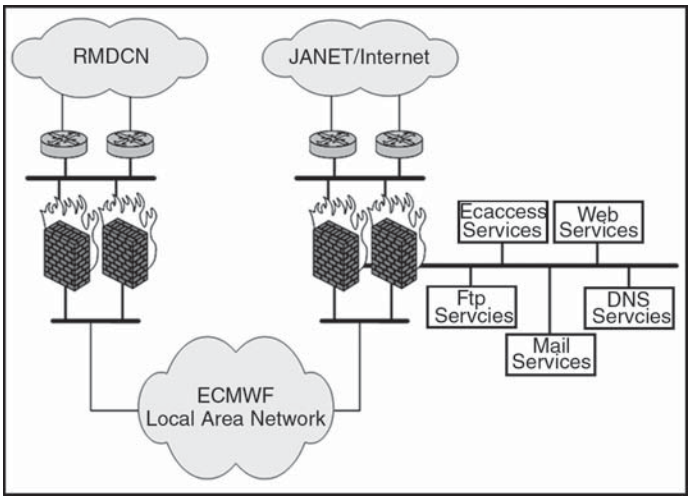


Fig. Typical "mesh" connectivity of a Wide Area Network

A typical network is shown in the figure above. This connects a number of End Systems (ES) (*e.g.* A, C, H, K) and a number of Intermediate Systems (IS) (*e.g.* B, D, E, F, G, I, J) to form a network over which data may be communicated between the End Systems (ES).

The characteristics of the transmission facilities lead to an emphasis on efficiency of communications techniques in the design of WANs. Controlling the volume of traffic and avoiding excessive delays is important. Since the topologies of WANs are likely to be more complex than those of LANs, routing algorithms also receive more emphasis. Many WANs also implement sophisticated monitoring procedures to account for which users consume the network resources. This is, in some cases, used to generate billing information to charge individual users.



The size of a network is limited due to size and distance constraints. However networks may be connected over a high-speed communications link (called a WAN link) to link them together and thus becomes a WAN.

WAN links are usually:

- Dial up connection
- Dedicated connection-It is a permanent full time connection. When a dedicated connection is used, the cable is leased rather than a part of the cable bandwidth and the user has exclusive use.
- Switched network-Several users share the same line or the bandwidth of the line.

There are two types of switched networks:

- Circuit switching-This is a temporary connection between two points such as dial-up or ISDN.
- Packet switching-This is a connection between multiple points. It breaks data down into small packets to be sent across the network.

A virtual circuit can improve performance by establishing a set path for data transmission. This will shave some overhead of a packet switching network. A variant of packet switching is called cell-switching where the data is broken into small cells with a fixed length.

WAN CONNECTION TECHNOLOGIES

- X.25-This is a set of protocols developed by the CCITT/ITU which specifies how to connect computer devices over an internet work. These protocols use a great deal of error checking for use over unreliable telephone lines. They establish a virtual communication circuit. It uses a store and forward method which can cause about a half second delay in data reception when two way communications are used. Their speed is about 64Kbps. Normally X.25 is used on packed switching PDNs (Public Data Networks). A line must be leased from the LAN to a PDN to connect to an X.25 network. A PAD (packet assembler/disassembler) or an X.25 interface is used on a computer to connect to the X.25 network. CCITT is an abbreviation for International Telegraph and Telephone Consultative Committee. The ITU is the International Telecommunication Union.
- Frame Relay-devices at both sides of the connection handle Error checking. Frame relay uses frames of varying length and it operates at the data link layer

of the OSI model. A permanent virtual circuit (PVC) is established between two points on the network. Frame relay speed is between 56Kbps and 1.544Mbps. Frame relay networks provide a high-speed connection up to 1.544Mbps using variable-length packet switching over digital fibre-optic media. Frame relay does not store data and has less error checking than X.25.

- Switched Multi-megabit Data Service (SMDS)-Uses fixed length cell switching and runs at speeds of 1.533 to 45Mbps. It provides no error checking and assumes devices at both ends provide error checking.
- Telephone connections
  - Dial up
  - Leased lines-These are dedicated analog lines or digital lines. Dedicated digital lines are called digital data service (DDS) lines. A modem is used to connect to analog lines, and a Channel Service Unit/Data Service Unit or Digital Service Unit(CSU/DSU) is used to connect to digital lines. The DSU connects to the LAN and the CSU connects to the line.
  - T Carrier lines-Multiplexors are used to allow several channels on one line. The T1 line is basic T Carrier service. The available channels may be used separately for data or voice transmissions or they may be combined for more transmission bandwidth. The 64Kbps data transmission rate

is referred to as DS-0 (Digital Signal level 0) and a full T1 line is referred to as DS-1.

Signal	**System**	**Total Kbps**	**Channels lines**	**Number of equivalent T1**
DS-1	T1	1544	24	1
DS-2	T2	6312	96	4
DS-3	T3	44736	672	28
DS-4	T4	274760	4032	3668

- T1 and T3 lines are the most common lines in use today. T1 and T2 lines can use standard copper wire. T3 and T4 lines require fibre-optic cable or other high-speed media. These lines may be leased partially called fractional T1 or fractional T3, which means a customer, can lease a certain number of channels on the line. A CSU/DSU and a bridge or router is required to connect to a T1 line.
- Integrated Services Digital Network (ISDN)-Comes in two types and converts analog signals to digital for transmission. It is a dial up service

• Basic Rate ISDN (BRI)-Two 64Kbps B-channels with one 16Kbps D channel. The D-channel is used tor call control and setup.

• Primary Rate ISDN (PRI)-23 B-channels and one D channel. A device resembling a modem (called an ISDN modem) is used to connect to ISDN. The computer and telephone line are plugged into it.

- Switched-56-A switched line similar to a leased line where customers pay for the time they use the line. Speed is 56Kbps. It is not dedicated and will not work to connect a WAN.

- Asynchronous Transfer Mode (ATM)-May be used over a variety of media with both baseband and broadband systems.

It is used for audio, video, and data. It uses fixed length data packets of 53 8 bit bytes called cell switching. 5 bytes contain header information. The cell contains path information that the packet is to use. It uses hardware devices to perform the switching of the data. Speeds from 155Mbps to 622 Mbps are achieved. Error checking is done at the receiving device, not by ATM. A permanent virtual connection or circuit (PVC) is established.

It may also use a switched virtual circuit (SVC). Service classes:

- Constant bit rate for data.
- Variable bit rate for audio or video.
- Connection less for data.
- Connection oriented for data.

ATM can be embedded in other protocols such as ATM-25, T1, T3, OC-1, OC-3, OC-12, and OC-48.

Some ATM technologies include:

- ATM-25-25Mbps speed.
- STS-3-155Mbps on fibre or category 5 cable.
- STS-12-620 Mbps on fibre cable for campus wide network.
- STS-48-2.2 Gbps on fibre cable on a MAN.
- STS-192-8.8 Gbps on fibre cable on intercity long distance. Phone companies normally use this.

Synchronous Optical Network (SONET)-A physical layer standard that defines voice, data, and video delivery methods over fibre optic media. It defines data rates in terms of optical carrier (OC) levels. The transmission rate of OC-1 is 51.8 Mbps. Each level runs at a multiple of the first. The OC-5 data rate is 5 times 51.8 Mbps which is 259 Mbps. SONET also defines synchronous transport signals (STS) for copper media which use the same speed scale of OC levels. STS-3 runs at the same speed of OC-3. Mesh or ring topology is used to support SONET. SONET uses multiplexing. The ITU has incorporated SONET into their Synchronous Digital Hierarchy (SDH) recommendations.

2

Integrated Services Digital Network

WHAT IS ISDN

Integrated Services Digital Network(ISDN) is a state-of-the-art Public Switched Digital Network for provisioning of different services – voice, data and image transmission over the telephone line through the telephone network.

ISDN, which stands for Integrated Services Digital Network, is a system of digital phone connections which has been available for over a decade. This system allows voice and data to be transmitted simultaneously across the world using end-to-end digital connectivity.

With ISDN, voice and data are carried by bearer channels (B channels) occupying a bandwidth of 64 kb/s (bits per second).

Some switches limit B channels to a capacity of 56 kb/s. A data channel (D channel) handles signaling at 16 kb/s or 64 kb/s, depending on the service type. Note that, in ISDN terminology, "k" means 1000 (10^3), not 1024 (2^{10}) as in many computer applications (the designator "K" is sometimes used to represent this value); therefore, a 64 kb/s channel carries data at a rate of 64000 b/s. A new set of standard prefixes has recently been created to handle this. Under this scheme, "k" (kilo-) means 1000 (10^3), "M" (mega-) means 1000000 (10^6), and so on, and "Ki" (kibi-) means 1024 (2^{10}), "Mi" (mebi-) means 1048576 (2^{20}), and so on. There are two basic types of ISDN service: Basic Rate Interface (BRI) and Primary Rate Interface (PRI). BRI consists of two 64 kb/s B channels and one 16 kb/s D channel for a total of 144 kb/s. This basic service is intended to meet the needs of most individual users.

PRI is intended for users with greater capacity requirements. Typically the channel structure is 23 B channels plus one 64 kb/s D channel for a total of 1536 kb/s. In Europe, PRI consists of 30 B channels plus one 64 kb/s D channel for a total of 1984 kb/s. It is also possible to support multiple PRI lines with one 64 kb/s D channel using Non-Facility Associated Signaling (NFAS).

H channels provide a way to aggregate B channels. They are implemented as:

- H0=384 kb/s (6 B channels)
- H10=1472 kb/s (23 B channels)

- H11=1536 kb/s (24 B channels)
- H12=1920 kb/s (30 B channels)-International (E1) only

To access BRI service, it is necessary to subscribe to an ISDN phone line. Customer must be within 18000 feet (about 3.4 miles or 5.5 km) of the telephone company central office for BRI service; beyond that, expensive repeater devices are required, or ISDN service may not be available at all. Customers will also need special equipment to communicate with the phone company switch and with other ISDN devices. These devices include ISDN Terminal Adapters(sometimes called, incorrectly, "ISDN Modems") and ISDN Routers.

ISDN HISTORY

The early phone network consisted of a pure analog system that connected telephone users directly by a mechanical interconnection of wires. This system was very inefficient, was very prone to breakdown and noise, and did not lend itself easily to long-distance connections. Beginning in the 1960s, the telephone system gradually began converting its internal connections to a packet-based, digital switching system. Today, nearly all voice switching in the U.S. is digital within the telephone network. Still, the final connection from the local central office to the customer equipment was, and still largely is, an analog Plain-Old Telephone Service (POTS) line.

A standards movement was started by the International Telephone and Telegraph Consultative Committee (CCITT), now known as the International Telecommunications Union (ITU). The ITU is a United Nations organization that coordinates and standardizes international telecommunications. Original recommendations of ISDN were in CCITT Recommendation I.120 (1984) which described some initial guidelines for implementing ISDN.

Local phone networks, especially the regional Bell operating companies, have long hailed the system, but they had been criticized for being slow to implement ISDN. One good reason for the delay is the fact that the two major switch manufacturers, Northern Telecom (now known as Nortel Networks), and AT&T(whose switch business is now owned by Lucent Technologies), selected different ways to implement the CCITT standards. These standards didn't always interoperate. This situation has been likened to that of earlier 19th century railroading. "People had different gauges, different tracks... nothing worked well."

In the early 1990s, an industry-wide effort began to establish a specific implementation for ISDN in the U.S. Members of the industry agreed to create the National ISDN 1 (NI-1) standard so that end users would not have to know the brand of switch they are connected to in order to buy equipment and software compatible with it. However, there were problems agreeing on this standard. In fact, many

western states would not implement NI-1. Both Southwestern Belland U.S. West (now Qwest) said that they did not plan to deploy NI-1 software in their central office switches due to incompatibilities with their existing ISDN networks.

Ultimately, all the Regional Bell Operating Companies (RBOCs) did support NI-1. A more comprehensive standardization initiative, National ISDN 2 (NI-2), was later adopted. Some manufacturers of ISDN communications equipment, such as Motorola and U S Robotics (now owned by 3Com), worked with the RBOCs to develop configuration standards for their equipment. These kinds of actions, along with more competitive pricing, inexpensive ISDN connection equipment, and the desire for people to have relatively low-cost high-bandwidth Internet access have made ISDN more popular in recent years.

Most recently, ISDN service has largely been displaced by broadband internet service, such as xDSL and Cable Modem service. These services are faster, less expensive, and easier to set up and maintain than ISDN. Still, ISDN has its place, as backup to dedicated lines, and in locations where broadband service is not yet available.

ADVANTAGES OF ISDN

Speed

The modem was a big breakthrough in computer communications. It allowed computers to communicate by

converting their digital information into an analog signal to travel through the public phone network. There is an upper limit to the amount of information that an analog telephone line can hold. Currently, it is about 56 kb/s bidirectionally. Commonly available modems have a maximum speed of 56 kb/s, but are limited by the quality of the analog connection and routinely go about 45-50 kb/s. Some phone lines do not support 56 kb/s connections at all. There were currently 2 competing, incompatible 56 kb/s standards (X2 from U S Robotics (recently bought by 3Com), and K56flex from Rockwell/Lucent). This standards problem was resolved when the ITU released the V.90, and later V.92, standard for 56 kb/s modem communications.

ISDN allows multiple digital channels to be operated simultaneously through the same regular phone wiring used for analog lines. The change comes about when the telephone company's switches can support digital connections. Therefore, the same physical wiring can be used, but a digital signal, instead of an analog signal, is transmitted across the line. This scheme permits a much higher data transfer rate than analog lines. BRI ISDN, using a channel aggregation protocol such as BONDING or Multilink-PPP, supports an uncompressed data transfer speed of 128 kb/s, plus bandwidth for overhead and signaling. In addition, the latency, or the amount of time it takes for a communication to begin, on an ISDN line is typically about half that of an

analog line. This improves response for interactive applications, such as games.

Multiple Devices

Previously, it was necessary to have a separate phone line for each device you wished to use simultaneously. For example, one line each was required for a telephone, fax, computer, bridge/router, and live video conference system. Transferring a file to someone while talking on the phone or seeing their live picture on a video screen would require several potentially expensive phone lines.

ISDN allows multiple devices to share a single line. It is possible to combine many different digital data sources and have the information routed to the proper destination. Since the line is digital, it is easier to keep the noise and interference out while combining these signals. ISDN technically refers to a specific set of digital services provided through a single, standard interface. Without ISDN, distinct interfaces are required instead.

Signaling

Instead of the phone company sending a ring voltage signal to ring the bell in your phone ("In-Band signal"), it sends a digital packet on a separate channel ("Out-of-Band signal"). The Out-of-Band signal does not disturb established connections, no bandwidth is taken from the data channels, and call setup time is very fast. For example, a V.90 or V.92

modem typically takes 30-60 seconds to establish a connection; an ISDN call setup usually takes less than 2 seconds.

The signaling also indicates who is calling, what type of call it is (data/voice), and what number was dialed. Available ISDN phone equipment is then capable of making intelligent decisions on how to direct the call.

INTERFACES

In the U.S., the telephone company provides its BRI customers with a U interface. The U interface is a two-wire (single pair) interface from the phone switch, the same physical interface provided for POTS lines. It supports full-duplex data transfer over a single pair of wires, therefore only a single device can be connected to a U interface. This device is called an Network Termination 1 (NT-1). The situation is different elsewhere in the world, where the phone company is allowed to supply the NT-1, and thereby the customer is given an S/T interface.

The NT-1 is a relatively simple device that converts the 2-wire U interface into the 4-wire S/T interface. The S/T interface supports multiple devices (up to 7 devices can be placed on the S/T bus) because, while it is still a full-duplex interface, there is now a pair of wires for receive data, and another for transmit data. Today, many devices have NT-1s built into their design. This has the advantage of making the devices less expensive and easier to install, but often reduces flexibility by

preventing additional devices from being connected. Technically, ISDN devices must go through an Network Termination 2 (NT-2) device, which converts the T interface into the S interface (Note: the S and T interfaces are electrically equivalent). Virtually all ISDN devices include an NT-2 in their design. The NT-2 communicates with terminal equipment, and handles the Layer 2 and 3 ISDN protocols. Devices most commonly expect either a U interface connection (these have a built-in NT-1), or an S/T interface connection.

Devices that connect to the S/T (or S) interface include ISDN capable telephones and FAX machines, video teleconferencing equipment, bridge/routers, and terminal adapters. All devices that are designed for ISDN are designated Terminal Equipment 1 (TE1). All other communication devices that are *not* ISDN capable, but have a POTS telephone interface (also called the R interface), including ordinary analog telephones, FAX machines, and modems, are designatedTerminal Equipment 2 (TE2). A Terminal Adapters (TA) connects a TE2 to an ISDN S/T bus.

Going one step in the opposite direction takes us inside the telephone switch. Remember that the U interface connects the switch to the customer premises equipment. This local loop connection is called *Line Termination* (LT function). The connection to other switches within the phone network is called *Exchange Termination* (ET function). The

LT function and the ET function communicate via the V interface.

LAYER 1-PHYSICAL LAYER

The ISDN Physical Layer is specified by the ITU I-series and G-series documents. The U interface provided by the telco for BRI is a 2-wire, 160 kb/s digital connection. Echo cancellation is used to reduce noise, and data encoding schemes (2B1Q in North America, 4B3T in Europe) permit this relatively high data rate over ordinary single-pair local loops.

2B1Q

2B1Q (2 Binary 1 Quaternary) is the most common signaling method on U interfaces. This protocol is defined in detail in 1988 ANSI spec T1.601.

In summary, 2B1Q provides:

- Two bits per baud
- 80 kilobaud (baud = 1 modulation per second)
- Transfer rate of 160 kb/s

Bits	QuaternarySymbol	VoltageLevel
00	–3	–2.5
01	–1	–0.833
10	+3	+2.5
11	+1	+0.833

This means that the input voltage level can be one of 4 distinct levels (note: 0 volts is not a valid voltage under this scheme). These levels are called Quaternaries. Each quaternary represents 2 data bits, since there are 4 possible ways to represent 2 bits, as in the table above.

Frame Format

Each U interface frame is 240 bits long. At the prescribed data rate of 160 kb/s, each frame is therefore 1.5 ms long.

Each frame consists of:

- Frame overhead-16 kb/s
- D channel-16 kb/s
- 2 B channels at 64 kb/s-128 kb/s

Sync bits	12 * (B_1 + B_2 + D) 6 bits	Maintenance6	18 bits	216

- The Sync field consists of 9 Quaternaries (2 bits each) in the pattern +3 +3 –3 –3 –3 +3 –3 +3 –3.
- (B_1 + B_2 + D) is 18 bits of data consisting of 8 bits from the first B channel, 8 bits from the second B channel, and 2 bits of D channel data.
- The Maintenance field contains CRC information, block error detection flags, and "embedded operator commands" used for loopback testing without disrupting user data.

Data is transmitted in a superframe consisting of 8 240-bit frames for a total of 1920 bits (240 octets). The sync field of the first frame in the superframe is inverted (*i.e.* –3 –3 +3 +3 +3 –3 +3 –3 +3).

LAYER 2-DATA LINK LAYER

The ISDN Data Link Layer is specified by the ITU Q-series documents Q.920 through Q.923. All of the signaling on the D channel is defined in the Q.921 spec.

LAP-D

Link Access Protocol-D channel (LAP-D) is the Layer 2 protocol used. This is almost identical to the X.25 LAP-B protocol.

Here is the structure of a LAP-D frame:

Flag	Address	Control	Information	CRC	Flag

Flag (1 octet)-This is always $7E_{16}$ ($0111\ 1110_2$)

Address (2 octets)							
1	2	3	4	5	6	7	8
		SAPI (6 bits)				C/R	EA0
	TEI(7bits)					EA1	

SAPI (Service access point identifier), 6-bits. C/R (Command/Response) bit indicates if the frame is a command or a response. EA0 (Address Extension) bit indicates whether this is the final octet of the address or not . TEI (Terminal Endpoint Identifier) 7-bit device identifier. EA1 (Address Extension) bit, same as EA0:

- *Control (2 octets):* The frame level control field indicates the frame type (Information, Supervisory, or Unnumbered) and sequence numbers (N(r) and N(s)) as required.
- *Information*: Layer 3 protocol information and User data
- *CRC (2 octets):* Cyclic Redundancy Check is a low-level test for bit errors on the user data.
- *Flag (1 octet):* This is always $7E_{16}$ ($0111\ 1110_2$)

SAPIs

The Service Access Point Identifier (SAPI) is a 6-bit field that identifies the point where Layer 2 provides a service to Layer 3.

See the following Table.

SAPI	Description
0	Call control procedures
1	Packet Mode using Q.931 call procedures
16	Packet Mode communications procedures
32-47	Reserved for national use
63	Management Procedures
Others	Reserved for Future Use

TEIs

Terminal Endpoint Identifiers (TEIs) are unique IDs given to each device (TE) on an ISDN S/T bus. This identifier can be dynamic; the value may be assigned statically when the TE is installed, or dyn amically when activated.

TEI	Description
0-63	Fixed TEI assignments
64-126	Dynamic TEI assignment (assigned by the switch)
127	Broadcast to all devices

Establishing the Link Layer

The Layer 2 establishment process is very similar to the X.25 LAP-B setup, if you are familiar with it.

- The TE (Terminal Endpoint) and the Network initially exchange Receive Ready (RR) frames, listening for someone to initiate a connection
- The TE sends an Unnumbered Information (UI) frame with a SAPI of 63 (management procedure, query network) and TEI of 127 (broadcast)

- The Network assigns an available TEI (in the range 64-126)
- The TE sends a Set Asynchronous Balanced Mode (SABME) frame with a SAPI of 0 (call control, used to initiate a SETUP) and a TEI of the value assigned by the network
- The network responds with an Unnumbered Acknowledgement (UA), SAPI=0, TEI=assigned.

At this point, the connection is ready for a Layer 3 setup.

LAYER 3-NETWORK LAYER

The ISDN Network Layer is also specified by the ITU Q-series documents Q.930 through Q.939. Layer 3 is used for the establishment, maintenance, and termination of logical network connections between two devices.

SPIDs

Service Profile IDs (SPIDs) are used to identify what services and features the telco switch provides to the attached ISDN device. SPIDs are optional; when they are used, they are only accessed at device initialization time, before the call is set up. The format of the SPID is defined in a recommendation document, but it is only rarely followed. It is usually the 10-digit phone number of the ISDN line, plus a prefix and a suffix that are sometimes used to identify features on the line, but in reality it can be whatever the telco decides it should be. If an ISDN line requires a SPID, but it is not

correctly supplied, then Layer 2 initialization will take place, but Layer 3 will not, and the device will not be able to place or accept calls.

Information Field Structure

The Information Field is a variable length field that contains the Q.931 protocol data.

Information Field							
1	2	3	4	5	6	7	8
Protocol Discriminator							
0	0	0	0	Length of CRV			
Call Reference Value (1 or 2 octets)							
0		Message Type Mandatory and Optional Information Elements (variable)					

These are the fields in a Q.931 header:

- *Protocol Discriminator (1 octet):* Identifies the Layer 3 protocol. If this is a Q.931 header, this value is always 08_{16}.
- *Length (1 octet):* Indicates the length of the next field, the CRV.
- *Call Reference Value (CRV) (1 or 2 octets):* used to uniquely identify each call on the user-network interface. This value is assigned at the beginning of a call, and this value becomes available for another call when the call is cleared.
- *Message Type (1 octet):* identifies the message type (*i.e.*, SETUP, CONNECT, etc.). This determines what additional information is required and allowed.

- *Mandatory and Optional Information Elements (variable length)*: are options that are set depending on the Message Type.

Layer 3 Call Setup

These are the steps that occurs when an ISDN call is established. *In the following example, there are three points where messages are sent and received*:

1. The Caller,
2. The ISDN Switch, and
3. The Receiver.
   - Caller sends a SETUP to the Switch.
   - If the SETUP is OK, the switch sends a CALL PROCeeding to the Caller, and then a SETUP to the Receiver.
   - The Receiver gets the SETUP. If it is OK, then it rings the phone and sends an ALERTING message to the Switch.
   - The Switch forwards the ALERTING message to the Caller.
   - When the receiver answers the call, is sends a CONNECT message to the Switch
   - The Switch forwards the CONNECT message to the Caller.
   - The Caller sends a CONNECT ACKnowledge message to the Switch
   - The Switch forwards the CONNECT ACK message to the Receiver.
   - *Done*: The connection is now up.

TERMINAL HANDLING

The example application TSTHLP works in the simplest way possible. Most real-life applications will be more sophisticated:

- The output routine HLP_OUTSUB used in TSTHLP outputs lines of text to the terminal without regard to screen management. The output routines of real applications may count lines and issue a "press return for more" message after a certain number, then wait for a response before going on; on appropriate terminals there may be screen erasures; and so on.
- Some sort of "abort" feature is useful, to provide a quick exit from the help session. On the VAX, detection of <CTRL>/Z in the application's input routine (by means of the END= feature of the READ statement), followed by simulation of multiple <CR> responses and ignoring of lines supplied to the output routine, is one way to arrange this.
- In the VMS DCL command HELP, a "?" response requires no <CR>, and it is possible an application may wish to provide this feature.

REFERENCE MODEL

HISTORY

Looking at the origins of the OSI Reference Model takes us back to several issues that were discussed in the Networking

Fundamentals chapter of this Guide; specifically, talking about standards and standards organizations. The idea behind the creation of networking standards is to define widely-accepted ways of setting up networks and connecting them together. The OSI Reference Model represented an early attempt to get all of the various hardware and software manufacturers to agree on a framework for developing various networking technologies.

In the late 1970s, two projects began independently, with the same goal: to define a unifying standard for the architecture of networking systems. One was administered by the *International Organization for Standardization (ISO)*, while the other was undertaken by the *International Telegraph and Telephone Consultative Committee*, or *CCITT* (the abbreviation is from the French version of the name). These two international standards bodies each developed a document that defined similar networking models.

In 1983, these two documents were merged together to form a standard called *The Basic Reference Model for Open Systems Interconnection*. That's a mouthful, so the standard is usually referred to as the *Open Systems Interconnection Reference Model*, the *OSI Reference Model*, or even just the *OSI Model*. It was published in 1984 by both the ISO, as standard ISO 7498, and the renamed CCITT (now called the *Telecommunications Standardization Sector of the International Telecommunication Union* or *ITU-T*) as standard

X.200. (Incidentally, isn't the new name for the CCITT *much* catchier than the old one? Just rolls off the old tongue, doesn't it. J) One interesting aspect of the history of the OSI Reference Model is that the original objective was *not* to create a model primarily for educational purposes—even though many people today think that this was the case. The OSI Reference Model was intended to serve as the foundation for the establishment of a widely-adopted suite of protocols that would be used by international internetworks—basically, what the Internet became. This was called, unsurprisingly, the OSI Protocol Suite.

However, things didn't quite work out as planned. The rise in popularity of the Internet and its TCP/IP protocols met the OSI suite head on, and in a nutshell, TCP/IP won. Some of the OSI protocols were implemented, but as a whole, the OSI protocols lost out to TCP/IP when the Internet started to grow.

The OSI model itself, however, found a home as a device for explaining the operation of not just the OSI protocols, but networking in general terms. It was used widely as an educational tool—much as I use it myself in this Guide—and also to help describe interactions between the components of other protocol suites and even hardware devices. While most technologies were not designed specifically to meet the dictates of the OSI model, many are described in terms of how they fit into its layers. This includes

networking protocols, software applications, and even different types of hardware devices, such as switches and routers. The model is also useful to those who develop software and hardware products, by helping to make clear the roles performed by each of the components in a networking system.

STANDARDS

The OSI reference model specifies standards for describing "Open Systems Interconnection" with the term 'open' chosen to emphasise the fact that by using these international standards, a system may be defined which is open to all other systems obeying the same standards throughout the world. The definition of a common technical language has been a major catalyst to the standardization of communications protocols and the functions of a protocol layer.

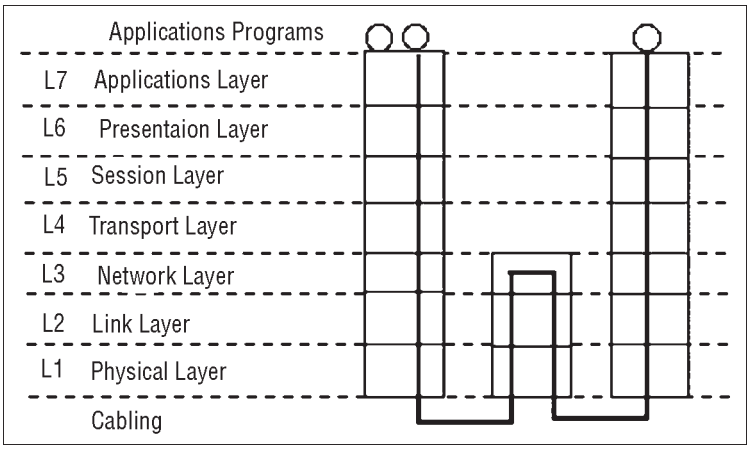


Fig. The Seven Layers of the OSI Reference Model

The structure of the OSI architecture is given in the figure above, which indicates the protocols used to exchange data

between two users A and B. The figure shows bidirectional (duplex) information flow; information in either direction passes through all seven layers at the end points. When the communication is via anetwork of intermediate systems, only the lower three layers of the OSI protocols are used in the intermediate systems.

OSI (Open Systems Interconnection) is a standard description or "reference model" for how messages should be transmitted between any two points in a telecommunication network. Its purpose is to guide product implementors so that their products will consistently work with other products. The reference model defines seven layers of functions that take place at each end of a communication. Although OSI is not always strictly adhered to in terms of keeping related functions together in a well-defined layer, many if not most products involved in telecommunication make an attempt to describe themselves in relation to the OSI model. It is also valuable as a single reference view of communication that furnishes everyone a common ground for education and discussion.

Developed by representatives of major computer and telecommunication companies beginning in 1983, OSI was originally intended to be a detailed specification of interfaces. Instead, the committee decided to establish a common reference model for which others could develop detailed interfaces, that in turn could become standards. OSI was

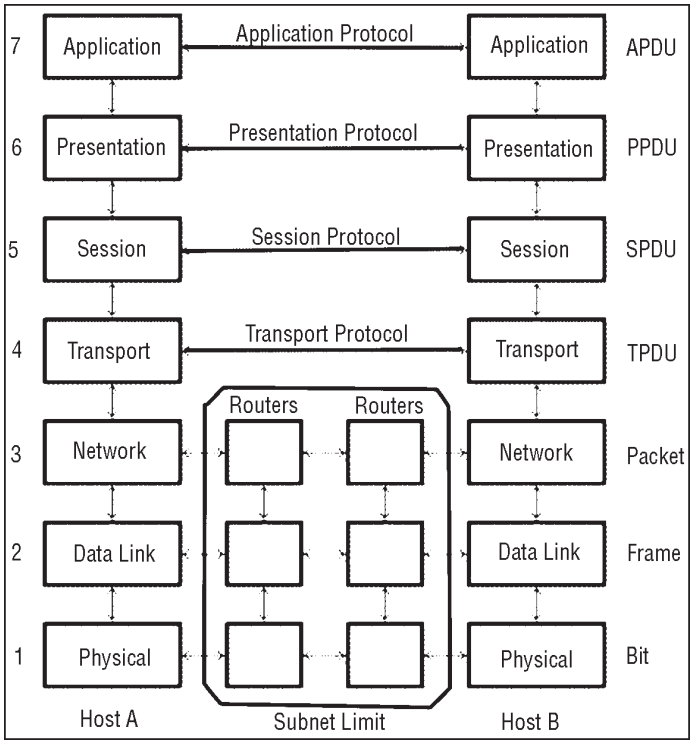
officially adopted as an international standard by the International Organization of Standards (ISO). Currently, it is Recommendation X.200 of the ITU-TS.

The main idea in OSI is that the process of communication between two end points in a telecommunication network can be divided into layers, with each layer adding its own set of special, related functions. Each communicating user or program is at a computer equipped with these seven layers of function. So, in a given message between users, there will be a flow of data through each layer at one end down through the layers in that computer and, at the other end, when the message arrives, another flow of data up through the layers in the receiving computer and ultimately to the end user or program. The actual programming and hardware that furnishes these seven layers of function is usually a combination of the computeroperating system, applications (such as your Web browser), TCP/IP or alternative transport and network protocols, and the software and hardware that enable you to put a signal on one of the lines attached to your computer.

OSI divides telecommunication into seven layers. The layers are in two groups. The upper four layers are used whenever a message passes from or to a user. The lower three layers (up to the network layer) are used when any message passes through the host computer. Messages intended for this computer pass to the upper layers. Messages destined for

some other host are not passed up to the upper layers but are forwarded to another host.

The seven layers are:



Physical Layer

Provides electrical, functional, and procedural characteristics to activate, maintain, and deactivate physical links that transparently send the bit stream; only recognizes individual bits, not characters or multicharacter frames.

The physical layer (layer 1) of the OSI reference model serializes the frame (*i.e.* converts it to a series of bits) and sends it across a communications circuit (*i.e* cable) to the destination (or an intermediate) system. There are a number of types of circuit/cable (sometimes also called

"media") that may be used.

These include:

- Open wire circuits
- Twisted pair cables
- Coaxial cables
- Fibre optic cables

Signaling of Bits

The physical layer defines the representation of ecah bit as a voltage, current, phase, or frequency.

Basic schemes are used:

1. RZ, Return to Zero (Pulse signalling)
2. NRZ, Non Return to Zero transmission (Level signalling)
3. Manchester encoding (Edge/phase signalling)

In NRZ transmission, each data bit is represented by a level. A high level may represent a logic 1, where as a low level may represent a logic 0. The term is derived from the earlier transmission technique of sending pulses to represent bits (called Return to Zero, RZ) in which a logic 1 is represented by a pulse and a logic 0 by the absence of a pulse. (AMI and HDB3 are techniques derived from RZ). Manchester encoding uses a still different scheme where a logic 1 is represnted by a transition in a particular direction (usually a rising edge) in the centre of each bit. A transudation in the opposite direction is used to represent a logic 0.

Timing of Bits

At the receiver, the remote system reassembles the series of bits to form a frame and forwards the frame for processing by the link layer. A clock (timing signal) is needed to identify the boundaries between the bits (in practice it is preferable to identify the centre of the bit-since this usually indicates the point of maximum signal power). There are two systems used to providing timing:

- Asynchronous Communication (independent transmit and receive clocks)
  - Simple interface (limited data rate < 19.2 kbps)
  - *Used for connecting*: Printer, Terminal, Modem
  - No clock sent (Tx & Rx have own clocks)
  - Requires start/stop which provides byte timing and increases overhead
  - Parity often used to validate correct reception
- Synchronous Communication (synchronised transmit and receive clocks)
  - More complex interface (high data rates supported upto ~ 1 Gbps)
  - *Used for*: LANs, MANs, WANs, Telephony
  - Clock sent with data (more configuration options)
  - Bit timing from transmission or by link protocol
  - Cyclic Redundancy Check (CRC) used to validate correct reception

Data Link Layer

Provides functional and procedural means to transfer data between network entities and (possibly) correct transmission

errors; provides for activation, maintenance, and deactivation of data link connections, grouping of bits into characters and message frames, character and frame synchronisation, error control, media access control, and flow control (examples include HDLC and Ethernet)

The MAC Sublayer

The *medium access control (MAC) sublayer* is closely associated with the physical layer and defines the means by which the physical channel (medium) may be accessed. It coordinates the attempts to seize a shared channel by multiple MAC entities, much as a school teacher must arbitrate between pupils' conflicting desires to speak. The MAC layer commonly provides a limited form of error control, especially for any header information which defines the MAC-level destination and higher-layer access mechanism.

Ethernet is a prime example of a shared medium with a defined MAC sublayer functionality. The shared medium in Ethernet has traditionally consisted of a coaxial cable into which multiple entities were "tapped," as depicted in Figure. Although this topology still applies conceptually, a hub and spoke medium is now typically used, in which the earlier coaxial cable has been physically collapsed into a *hub* device.

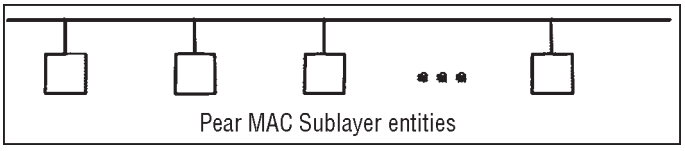


Fig: Ethernet MAC System

As a *contention* medium, Ethernet defines how devices *sense* a channel for its availability, wait when it is busy, *seize* the channel when it becomes available and *back-off* for a random length of time following a *collision* with another simultaneously transmitting device. On a shared channel, such as Ethernet, only a single entity can transmit at a time or messages will be garbled.

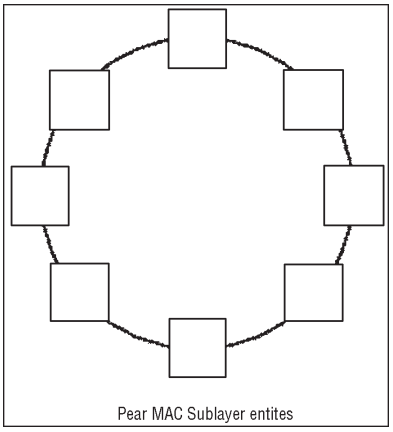


Fig. Token Ring MAC System

Not all shared channels involve contention. A prime example of a *contentionless* shared medium is *token ring (IEEE 802.5)*, in which control of the channel is rotated between the devices sharing the channel in a deterministic round-robin manner. Conceptually, control of the channel is given to the entity currently possessing a "token." If the device has nothing to transmit, it passes the token to the next device attached to the topological "ring,"

IEEE-defined MAC sublayer addresses are six bytes long and permanently assigned to each device, typically called a *network interface card* or *NIC. The* IEEE administers the

assignment of these addresses in blocks to manufacturers to assure the global uniqueness that the MAC sublayer protocols rely on for "plug Ôn play" network setup. Each manufacturer must assure individual device identifier uniqueness within their assigned block.

The LLC Sublayer

The *logical link control (LLC) sublayer* is responsible for reliable transfer of messages-called *frames* or, more formally, *link protocol data units (LPDUs)*-between two directly-connected Layer 2 entities. Functions needed to support this reliable transfer include *framing* (indicating where a Layer 2 message begins and ends), sequence control, error control and flow control.

The degree to which sequence, error and flow control are provided by the LLC sublayer is determined by whether the link protocol is connection-oriented or connectionless. A connectionless link protocol provides little if any support for these functions. A connection-oriented link might use a windowing technique for these functions, in which frames are individually numbered and acknowledged by their sequence number, with only a few such frames outstanding at any time.

The connection-oriented functions of sequencing, error and flow control provide a foundation for services provided by higher layers. As mentioned earlier, not all layer or sublayer functions are explicitly designed or implemented in any given

system. Provision of these functions depends on the services required by higher layers.

If the connection-oriented functions of the LLC sublayer are not implemented, they must be performed by higher layers for reliable end-to-end communication. If these functions are provided by several layers, they might be somewhat redundant and add unnecessary overhead (inefficiency) to the system. In the worst case, redundant provision of these functions at multiple layers could serve cross purposes and actually degrade overall system performance. An example of a connectionless LLC protocol is *frame relay (T1.617, 618)*, which defines point-to-point links with *switches* connecting individual links in a mesh topology. In a frame relay network, endpoints are connected by a series of links and switches. Because frame relay is defined in terms of the links between frame relay access devices (FRADs) and switches, and between switches themselves, it is an LLC protocol.

Connectionless Layer 2 protocols are best suited for high quality transmission media. With high quality transmission media, errors are rarely introduced in the transmission between network layer entities and discovery of and recovery from errors is most efficiently handled by the communicating hosts. In this case, it is better to move the packets quickly across the traversed subnetworks from source to destination rather than checking for errors at Layer 2.

Frame relay is derived from the *X.25 (ISO 8208)* protocol which spans Layers 2 and 3. X.25 is a connection-oriented packet-switching technology which defines how neighbouring *packet switches* exchange data with one another in a reliable manner from end-to-end. Frame relay simply removes the connection-oriented functions of error and sequence control; however, *congestion control* functions are provided in frame relay, to prevent the total traffic seen at any point in the network from overwhelming it.

Connection-oriented Layer 2 protocols are best suited for low quality transmission media where it is more efficient and cost-effective to discover and recover from errors as they occur on each hop than to rely on the communicating hosts to perform error recovery functions. With ever-increasing quality of transmission facilities and decreasing costs of computation capability at hosts, the need for connection-oriented network layer protocols is diminishing. However, X.25 remains popular outside of North America, where it has been tariffed at levels which encourage its use.

End-to-end communications may be via shared or dedicated facilities or *circuits*. Shared facilities involve the use of *packet switching* technology to carry messages from end-to-end; messages are subdivided as necessary into packets, which share physical and logical channels with packets from various sources to various destinations. Packet switching is almost universally used in data communications

because it is more efficient for the bursty nature of data traffic. On the other hand, some applications require dedicated facilities from end-to-end because they are isochronous (*e.g.*, voice) or bandwidth-intensive (*e.g.*, large file transfer). This mode of end-to-end circuit dedication is called *circuit switched* communication. Because the facilities are dedicated to a single user, this tends to be much more expensive than the packet switched mode of communication. But some applications need it-it is an economic trade-off.

Dedicated circuits are a rather extreme form of connection-oriented protocol, requiring the same setup and tear-down phases prior to and following communication. If the circuit setup and tear-down is statically arranged (*i.e.*, out-of-band), it is referred to as a *permanent virtual circuit* or *PVC*. If the circuit is dynamically setup and torn-down in-band, it is referred to as a *switched virtual circuit* or *SVC*.

Network layer

Provides independence from data transfer technology and relaying and routing considerations; masks peculiarities of data transfer medium from higher layers and provides switching and routing functions to establish, maintain, and terminate network layer connections and transfer data between users.

The Internetwork Protocol (IP)

The IP (Internet Protocol) is a protocol that uses data grams to communicate over a packet-switched network. The

IP protocol operates at the network layer protocol of the OSI reference model and is a part of a suite of protocols known as TCP/IP. Today, with over 1.5 billion users worldwide, the current Internet is a great success in terms of connecting people and communities. Even though the current Internet continues to work and is capable of fulfilling its current missions, it also suffers from a relative œossification, a condition where technological innovation meets natural resistance, as exemplified by the current lack of wide deployment of technologies such as multicast or Internet Protocol version 6 (IPv6).

The Internetwork Protocol (IP) [RFC791] provides a best effort network layer service for connecting computers to form a computer network. Each computer is identified by one or more gloablly unique IP addresses. The network layer PDUs are known as either “packets” or “data grams”. Each packet carries the IP address of the sending computer and also the address of the intended recipient or recipients of the packet. Other management information is also carried.

The IP network service transmits datagrams between intermediate nodes using IP routers. The routers themselves are simple, since no information is stored concerning the datagrams which are forwarded on a link. The most complex part of an IP router is concerned with determining the optimum link to use to reach each destination in a network. This process is known as “routing”. Although this process

is computationally intensive, it is only performed at periodic intervals. An IP network normally uses a dynamic routing protocol to find alternate routes whenever a link becomes unavailable. This provides considerable robustness from the failure of either links or routers, but does not guarentee reliable delivery. Some applications are happy with this basic service and use a simple transport protocol known as the User Datagram Protocol (UDP) to access this best effort service.

Most Internet users need additional functions such as end-to-end error and sequence control to give a reliable service (equivalent to that provided by virtual circuits). This reliability is provided by the Transmission Control Protocol (TCP) which is used end-to-end across the Internet.

In a LAN environment, the protocol is normally carried by Ethernet, but for long distance links, other link protocols using fibre optic links are usually used. Other protocols associated with the IP network layer are the Internet Control Message Protocol (ICMP) and the Address Resolution Protocol (arp).

IP the Next Generation, IPv6

The IPv4 protocol although widely used, is slowly being superceded by IPv6 [RFC2460], a next-generation network-layer protocol. IPv6 is now widely implemented, and deployed in many networks. The gradual transition from IPv4 towards majority IPv6 deployment will take many years and IPv4 may never itself

be phased out completely. In the meantime the two protocols can co-exist and be used together in various ways.

IPv4 Packet Header

The Internet Protocol (IP) uses a Datagram service to transfer packets of data between end systems using routers. The IPv4 packet header consists of 20 bytes of data. An option exists within the header that allows further optional bytes to be added, but this is not normally used (with the occasional exception of something called "Router Alert").

The header fields are discussed below:

- Version (always set to the value 4 in the current version of IP)
- IP Header Length (number of 32 -bit words forming the header, usually five)
- Type of Service (ToS), now known as Differentiated Services Code Point (DSCP) (usually set to 0, but may indicate particular Quality of Service needs from the network, the DSCP defines the way routers should queue packets while they are waiting to be forwarded).
- Size of Datagram (in bytes, this is the combined length of the header and the data)
- Identification (16-bit number which together with the source address uniquely identifies this packet-used during reassembly of fragmented datagrams)
- Flags (a sequence of three flags (one of the 4 bits is unused) used to control whether routers are allowed

to fragment a packet (*i.e.* the Don't Fragment, DF, flag), and to indicate the parts of a packet to the receiver)

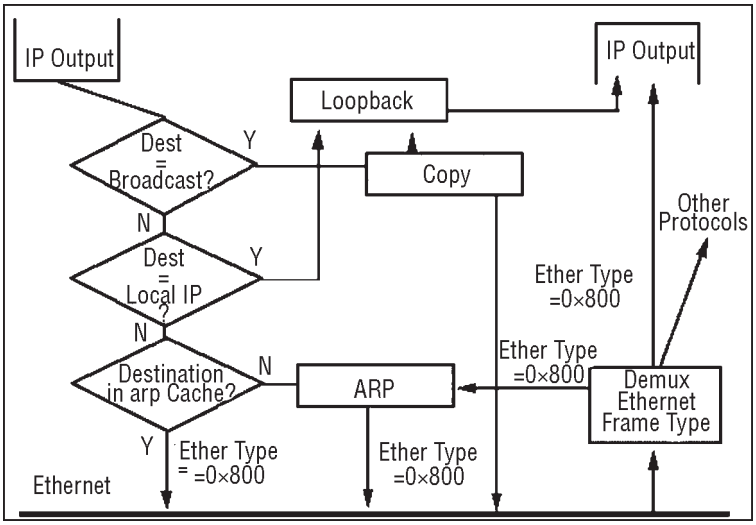
- Fragmentation Offset (a byte count from the start of the original sent packet, set by any router which performs IP router fragmentation)
- Time To Live (Number of hops/links which the packet may be routed over, decremented by most routers - used to prevent accidental routing loops)
- Protocol (Service Access Point (SAP) which indicates the type of transport packet being carried (*e.g.* 1 = ICMP; 2= IGMP; 6 = TCP; 17= UDP).
- Header Checksum (A 1's complement checksum inserted by the sender and updated whenever the packet header is modified by a router - Used to detect processing errors introduced into the packet inside a router or bridge where the packet is not protected by a link layer cyclic redundancy check. Packets with an invalid checksum are discarded by all nodes in an IP network)
- Source Address (the IP address of the original sender of the packet)
- Destination Address (the IP address of the final destination of the packet)
- Options (not normally used, but, when used, the IP header length will be greater than five 32-bit words to indicate the size of the options field)

IP Packet Processing

Transmission of a frame over Ethernet.

The IP packet is placed in an Ethernet frames as follows:

- *IP Broadcast/Multicast Address*: The IP destination address is checked to see if the system should also receive a copy of the packet. This happens if this is an IP network broadcast address (or a multicast address is used that matches one of the registered IP multicast filters set by the IP receiver). If a copy is required, it is sent to the loopback interface. This directly delivers the packet to the IP input routine. the original packet continues to be processed



- *IP Unicast Address*: The IP destination address is checked to see if the address is the unicast (source) IP address of the sending system. Such packets are sent directly to the loopback interface (*i.e.* never reach the physical Ethernet interface)

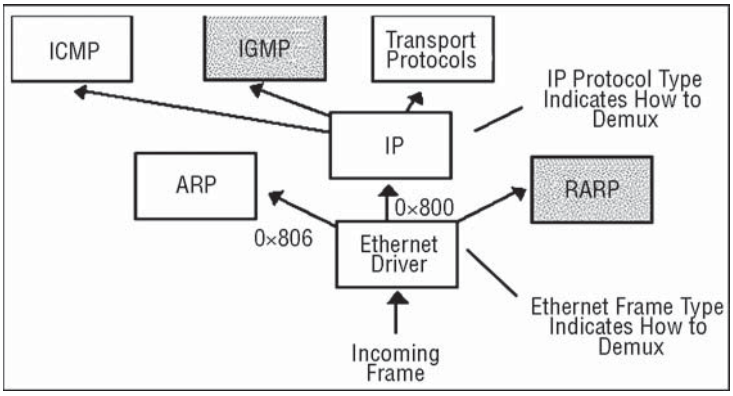
- *Next Hop IP Address*: The sender then determines the next hop address-that is the IP address of the next Intermediate System/End System to receive the packet. Once this address is known, the Address Resolution Protocol (arp) is used to find the appropriate MAC address to be used in the Ethernet frame.

  This is a two stage process:
  - The arp cache is consulted, to see if the MAC address is already known, in which case the correct address is added and the packet queued for transmission.
  - If the MAC address is not in the arp cache, the arp protocol is used to request the address, and the packet is queued until an appropriate response (or timeout) occurs.
- *MTU*: The size of the packet is checked against the MTU of the link on which it is to be sent. (Note the MTU of the loopback interface may be different to that of Ethernet). If required, IP fragmentation is performed, or an ICMP error message is returned, which may trigger Path MTU Discovery at the sending End Host.
- *Encapsulation*: The Ethernet frame is completed, by inserting the Destination, Source and Ethernet Type fields. When Tags are used, the appropriate 802.1pQ Tag is inserted following the MAC header (the Priority field in the Tag may be set based on the IP DSCP value).

- *Transmit*: The frame is transmitted using the MAC procedure for Ethernet.

Reception of a frame from Ethernet

The following summary shows the processing performed by an end system in an IP network. It is assumed that the system is connected to an Ethernet network.



The received frames are processed as follows:

- *MAC Protocol*: The Ethernet controller in the network interface card verifies that the frame is:
  - Not less than the minimum frame length not greater than the maximum length (1500 B)
  - Contains a valid CRC at the end
  - Does not contain a residue (*i.e.* extra bits which do not form a byte)
- *MAC Address: The frame is then filtered based on the MAC destination address and accepted only if*:
  - It is a broadcast frame (*i.e.* all bits of the destination address field are set to 1)
  - It is a multicast frame to a registered MAC group address

- It is a unicast frame to the node's own MAC address
- Or the interface is acting in promiscuous mode (*i.e.* as a bridge)

- *MAC SAP: The frame is then demultiplexed based on the specified MAC packet type (SAP):*
  - Frames carrying an IEEE 802.1pQ Tag will have their Virtual LAN information checked and processed, before skipping the Tag field and reading the following EtherType field.
  - It is passed to the appropriate protocol layer (*e.g.* LLC, ARP, IP)
  - Frames carrying Packets destined for IP have a type field of 0x0800 and those for arp have a value 0x0806.
- *IP Check:* The IP packet header is checked, including:
  - By checking the protocol type =4 (*i.e.* current version of IP)
  - By verifying the header checksum
  - By checking the header packet length
- *IP Address: The destination IP network address is then checked*:
  - If it matches an IP address of the node then it is accepted
  - If it is network broadcast packet to the node's network it is accepted
  - If it is a multicast packet to an IP multicast address which is in use then it is accepted
  - If it is none of these, it is forwarded using the routing table (if possible) or discarded

- *IP Fragmentation*: Packets for the node are then checked concerning whether reassembly is required:
  - The fragmentation offset value and more flags are inspected
  - Fragments are placed in a buffer until other fragments are received to complete the packet.
- *IP SAP*: The IP protocol field (SAP) is checked:
  - The SAP field identifies the transport protocol (*e.g.* 1 = ICMP; 6 = TCP; 17= UDP)
  - The complete packet is passed to the appropriate transport layer protocol.

Transport Layer

Provides transparent transfer of data between systems, relieving upper layers from concern with providing reliable and cost effective data transfer; provides end-to-end control and information interchange with quality of service needed by the application program; first true end-to-end layer.

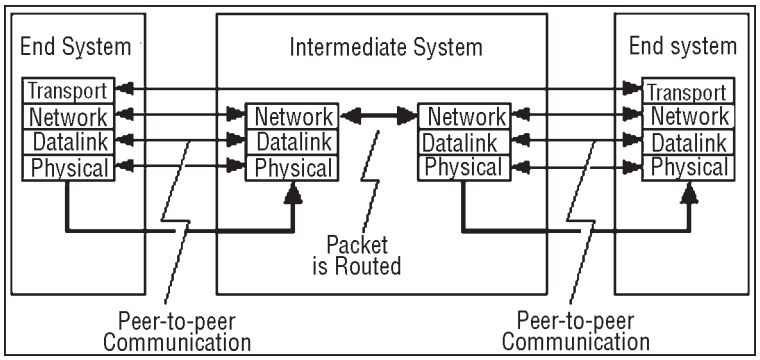
Transport Layer Protocol

The transport layer is the fourth layer of the OSI reference model. It provides transparent transfer of data between end systems using the services of the network layer (*e.g.* IP) below to move PDUs of data between the two communicating systems.

The transport service is said to perform "peer to peer" communication, with the remote (peer) transport entity. The data communicated by the transport layer is encapsulated in

a transport layer PDU and sent in a network layer SDU. The network layer nodes (*i.e.* Intermediate Systems (IS)) transfer the transport PDU intact, without decoding or modifying the content of the PDU. In this way, only the peer transport entities actually communicate using the PDUs of the transport protocol.

Two End Systems Connected by an Intermediate System. The Figure Shows the Various Protocol Layers Drawn with Reference to the OSI Reference Model:



The transport layer relieves the upper layers from any concern with providing reliable and cost effective data transfer. It provides end-to-end control and information transfer with the quality of service needed by the application program. It is the first true end-to-end layer, implemented in all End Systems (ES).

Session Layer

Provides mechanisms for organising and structuring dialogues between application processes; mechanisms allow for two-way simultaneous or two-way alternate operation,

establishment of major and minor synchronisation points, and techniques for structuring data exchanges.

Presentation Layer

Provides independence to application processes from differences in data representation, that is, in syntax; syntax selection and conversion provided by allowing the user to select a "presentation context" with conversion between alternative contexts.

Application Layer

Concerned with the requirements of application. All application processes use the service elements provided by the application layer. The elements include library routines which perform interprocess communication, provide common procedures for constructing application protocols and for accessing the services provided by servers which reside on the network. The communications engineer is concerned mainly with the protocols operating at the bottom four layers (physical, data link, network, and transport) in the OSI reference model. These layers provide the basic communications service. The layers above are primarily the concern of computer scientists who wish to build distributed applications programs using the services provided by the network.

"HOP-BY-HOP" "NETWORK-WIDE" AND "END-TO-END" COMMUNICATION

The two lowest layers operate between adjacent systems connected via the physical link and are said to work "hop by

hop". The protocol control information is removed after each "hop" across a link (*i.e.* by each System) and a suitable new header added each time the information is sent on a subsequent hop. The network layer (layer 3) operates "network-wide" and is present in all systems and responsible for overall co-ordination of all systems along the communications path. The layers above layer 3 operate "end to end" and are only used in the End Systems (ES) which are communicating. The Layer 4-7 protocol control information is therefore unchanged by the IS in the network and is delivered to the corresponding ES in its original form. Layers 4-7 (if present) in Intermediate Systems (IS) play no part in the end-to-end communication.

SERVICES

WEB SERVICES PROTOCOL STACK

Web Services are a set of protocols based on XML (Extensible Markup Language). Many readers will be familiar with the following base protocols that formed the initial specification for Web Services.

- *Simple Object Access Protocol (SOAP)*: defines the runtime message that contains the service request and response. SOAP is independent of any particular transport and implementation technology.
- *Web Services Description Language (WSDL)*: describes a Web Service and the SOAP Message. It provides a

programmatic way to describe what a service does, paving the way for automation.

- *Universal Discovery, Description, Integration (UDDI)*: UDDI is a cross industry initiative to create a standard for service discovery together with a registry facility that facilitates the publishing and discovery processes. These have effectively become de facto standards, with effectively universal acceptance and widespread implementation by vendors.

These base protocols have enabled many companies to put straightforward Web Services into production. However, to improve the security and reliability of Web Services and to address more complex business scenarios, a wide range of additional protocols have since been proposed. Some of these have since been merged with others or morphed into new proposals.

CBDI ASSESSMENT

Additional Protocols Required

Taking all the proposals in consideration, the set of protocols required for secure, reliable 'Enterprise' Web Services is largely complete.

Areas not fully addressed are:

- *Management*: The OASIS WSDM Technical Committee is still in its early stages. WSDM will provide standard protocols for the Management Of Web Services

(MOWS), and for Management Using Web Services (MUWS).

- *Service and Business Level Agreements*: These are identified by the W3C Web Service Architecture working group as part of the description layer, but as yet no proposals have been made in this area.
- *WS-Security*: The specifications for some elements of the WS-Security architecture have yet to be published. These are WS-Authorization and WS-Privacy.

Alternative Proposals

The degree of industry consensus on Web Service protocols has been significant. Though alternative proposals have been made in some areas, the formation of an appropriate working group in either W3C or OASIS has usually seen the subsequent convergence of all interested parties.

There are currently some areas where alternative proposals remain, namely in the areas of Reliable Messaging, Orchestration, and Transaction Coordination. These alternatives have generally reflected an IBM/Microsoft led initiative on one side, and one led by Sun/Oracle on the other.

However, Microsoft and Sun earlier this year agreed to settle various antitrust and other issues, and announced greater cooperation in Web Services. Since then Sun has joined with BEA, IBM, Microsoft and SAP AG to submit the latest version of the WS-Addressing specification to the W3C. This will merge

with the WS-MessageDelivery specification which Sun supported earlier along with Iona Nokia and Oracle. This was followed by BEA, CA, IBM, Microsoft, Sun and TIBCO jointly publishing an update to the WS-Eventing specification, which proposes a way of communicating about events within and between Web services. This update sees CA, IBM and Sun joining BEA, Microsoft and TIBCO, who proposed the spec originally, and likely signals the prospect of greater interoperability with related specifications such as WS-Notification. Ca, Sun and WebMethods also joined with BEA, IBM, Microsoft, and SAP, in publishing the 2nd version of WSMetadataExchange.

These are welcome moves, and it augurs well for further cooperation in other pockets of overlapping proposals such as transaction co-ordination and choreography. Consequentl, it looks like any concerns over competing Web Service protocols delaying standardization, and hence adoption, should now disappear.

There has also been some overlap between Web Services and the ebXML initiative. ebXML uses SOAP at the transport level, but has its own registry and orchestration. Though ebXML is an approved, robust standard, its applicability is far narrower than Web Services. As an evolution of EDI, it primarily addresses the B2B domain only. As such we believe the Web Service protocols that are designed to address multiple requirements and usage scenarios will prove more valuable in time and that ebXML will probably evolve to adopt

additional Web Service protocols as they mature and are approved. Part of the work of the OASIS ebSOA TC is to evolve the ebXML architecture and address the transition to the adoption of more Web Service protocols.

Standardization Process

Though the proposal of various Web Services protocols has been a fast moving area, their transition into actual open standards is inevitably much slower.

There are only a few protocols that have, or a close to completing the standards process proper. Some key proposals have yet to be submitted to any standards body. We advise continuous monitoring of what are currently the two main standards groups involved in Web Services,

- *World Wide Web Consortium (W3C)*:
- Organization for the Advancement of Structured *Information Standards (OASIS)*: Web service protocols summary table indicates the current status of the various protocols in the standards process.

WS-Interoperability (WS-I)

WS-I is an open, industry group that was formed in 2002 to promote Web services interoperability across platforms, operating systems, and programming languages. Though this would appear to be the basic premise of Web Services and the role of standards bodies, WS-I still has a useful role to play, for example,

- Standards specifications are always open to interpretation to some extent. WS-I will provide guidelines and tools to help measure the conformance of various implementations, and to enable their interoperability
- As standards evolve, there is a need to understand what different versions might interoperate
- Publishing interoperability profiles to reflect the above, one of the key deliverables of WS-I.

Adoption

The current status of these protocols is shown in Figure.

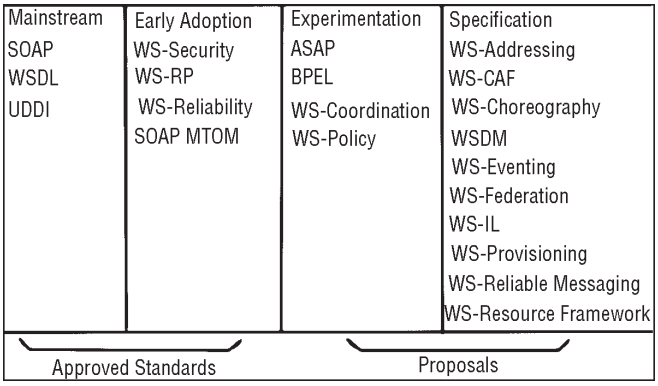


Fig. Adoption of Web Service Protocols

- *Specification*: Exists only as draft specification. Any usage requires hand coding.
- *Experimentation*: early implementations provided by vendors permit experimentation, but are not recommended for production use. (*e.g.* technologies available from IBM Alphaworks do not support production use)

- *Early adoption*: More robust implementations available and protocol well into standards process, encourages production usage by end user organizations
- *Mainstream*: standard ratified or wide scale de facto adoption

ROADMAP ACTIONS

Apart from infrastructure and tools vendors, and early experimentation, organizations should avoid handcrafting the use of Web Service protocols wherever possible. It should not be necessary for developers to learn the low-level XML syntax of Web Services, delegating the generation of it instead to the infrastructure products and development tools. Organizations should establish a policy for compliance with standards, paying particular attention to evolving versions, and using WS-I profiles wherever relevant.

Monitor progress of protocols through key standards bodies Establish policy on protocol usage. Adopt protocols as WS-Profiles become available to ensure standards based interoperability. Create local profiles only where necessary, and plan to upgrade to WS-I as they are published. Coordinate use of protocols to ensure consistent implementation of versions and profiles. Publish best practices Plan for phased implementation of emerging protocols with local extensions where necessary. Wherever possible wait for implementation of protocols in products.

DELAY ANALYSIS

TECHNIQUES

My first article on delay analysis (CJ 1 March) emphasised the importance of establishing that an event falls on the critical path of a project in order to give rise to an entitlement to extension of time. This article considers one of a number of techniques used by delay analysts to set out and present a case for delay.

It is not intended through these articles to express any particular views as to which method is the best, since the approach very much depends on the facts, the nature of the events being analysed, the nature and extent of the available as-built evidence, and the available programme and progress data. Complex projects can be very difficult to analyse, but above all, the case of Skanska -v- Egger (2004) illustrated the importance of focusing on the facts and keeping the analyses and the presentation of delays as simple as possible.

This second article considers the merits of the as-planned approach versus as-built approach to modeling delay.

The Method

The as-planned versus as-built method of analysis can be carried out without the need for complex computer project management software, although such software is usually used for its presentation. This method is described by a number of English and American commentators as well as

being referred to in SCL Delay and Disruption Protocol. The method merely compares the durations of the planned activities with the as-built durations, and attributes the variance between the planned construction period and the as-built construction period as an entitlement to an extension of time. It therefore assumes that the as-built situation arises by reason of variations or other reasons for which the contractor is entitled to an extension of time and not due to its own culpability.

Selection Criteria

This method is well suited for relatively simple projects where it is easy to identify where the main delays arose. Clearly it requires the planned programme and as-built information in sufficient detail to compare as-built activities with the same planned activities. This method is useful to identify the difference between the planned and actual durations and sequence of events in order to focus the further investigation on areas which appear to have gone particularly wrong and which appear to have been critical. Because this method requires as-built information it can only be used for retrospective delay analysis.

Issues

This is obviously a very simplistic method of analysis. Indeed it is questionable whether this can really be called a method of analysis in the way that it is described by

commentators. It is unlikely that it can credibly be concluded that the difference between the planned and as-built programmes gives rise to an extension of time.

To do so assumes that the planned programme was realistic and achievable, the work was appropriately resourced to enable that plan to be achieved and that the only reason for the delay was because of an event for which the contractor is entitled to an extension of time. What is needed is a careful consideration of the facts from which one should seek to determine an actual period of delay, which either will or will not sufficiently explain the difference between the planned and as-built programmes.

For example, if one looks at the programme above, it may be that the planned duration for Activity 1 was an under-estimate and there was no event affecting it, that the same applies to Activity 2, and Activity 3 was an overestimate, but was subject to a delay of four days, thus explaining four of the seven days of delay to completion. None of this would be apparent from the simple comparison shown in the chart, but would become apparent from an investigation of the facts.

Further, the planned programme may well not be in sufficient detail to enable a credible conclusion that the difference between the planned and actual activity durations is explainable by the events relied upon. If this method is used without the use of corporate performance management (CPM) software then it may be difficult to

demonstrate the critical path and the extent of the effect on the completion date.

Advantages

This method of analysis is generally the cheapest and simplest to perform, although assuming a planned programme exists, it still requires the production of an as-built programme. This approach can be useful in relation to events that are clearly on the critical path.

Disadvantages

It has been observed that the fact that this is not a CPM-based method means that it is impossible to demonstrate with any precision the effects of concurrency or parallel delays; unproductive working; the effects of secondary or consequential delay; acceleration; resequencing or mitigation. This is not necessarily so, since many of these issues are a question of fact and how they are presented.

If this method is modified to reflect and present the actual delays determined from a proper investigation of the facts, then this approach has its place. However, the use of this method of analysis in its simplest form is easily criticised and is unlikely to be reliable in dispute resolution.

BACK BONE DESIGN

In early data networking, the topology for the network backbone was relatively simple: Operations were centralized,

so a star topology made the most sense—and, in some cases, this was the only topology the technology would support. This did cause the center of the star to become a single point of failure, but because no real traffic flows existed between spokes on the star, this was not a major cause for concern. With the move towards multiple client-server and peer-to-peer relationships, the choice of core network topology is not as clear.

The purpose of the backbone is to connect regional distribution networks and, in some instances, to provide connectivity to other peer networks. A national infrastructure usually forms a significant part of the operational cost of the network. Given its position at the top of the network hierarchy, two requirements of the backbone topology are clear: it must be reliable and it must scale.

MAKING THE BACKBONE RELIABLE

Reliability can be acquired by employing two methods. First, you can create more reliable routers through the use of "carrier-class" characteristics, such as multiple CPUs, power supplies, and generators; and even redundant routers. Ultimately, however, any backbone will include WAN links that rely on a great deal of equipment and environmental stability for their operation, which represents a real risk of ultimate failure. If the carrier's up-time guarantees are not sufficient, you have no choice but to design a backbone that is resilient to link failure. The second option is to simply connect all distribution networks with a full mesh.

However, in terms of minimizing hop count within the network, the full mesh approach has several drawbacks:

- First, given N regional distribution networks, you must have N(N-1)/2 backbone links in the core. This creates expense in WAN circuitry, as well as in router and WAN switch hardware (channelized or ATM technology can reduce these issues).
- Moreover, PVC sizing requires that the traffic levels between any two distribution networks should be well understood, or that the network has the capability to circumvent congestion. Although traffic engineering calculations and circumventing congestion are common in the telephone network, common IP networks and their associated routing protocols do not provide this capability as readily. One good reason is that the resources required by any TCP/IP session are not known *a priori*, and IP networks are traditionally engineered as best-effort. Chapter 14 explores how to bypass best-effort by providing differentiated service in IP networks.
- A full PVC mesh can also obviate one of the benefits of multiplexing, or *trunking*, in a best-effort network. Round-trip time and TCP window size permitting, any user can burst traffic up to the full line rate of the trunk. Furthermore, the routing complexity in a full

mesh can consume bandwidth, computational, and operational management resources.

Most backbone topologies are, therefore, initially designed based on financial constraints, such as user population density, or application requirements; and WAN service availability. This initial design can be subsequently refined quite effectively by statistical analysis of traffic levels after the backbone is operational, and the availability of new WAN technologies is known. Data network requirements analysis is a relatively new art.

BUILDING THE BACKBONE TOPOLOGY

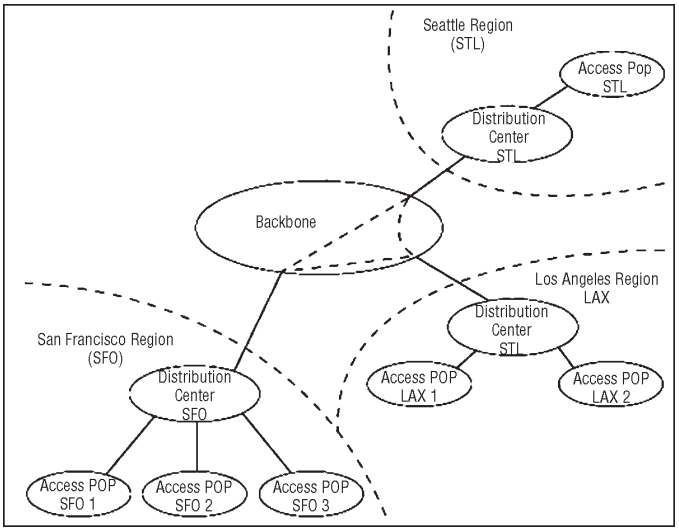
Because you have a basic need for resilience in the backbone, a good starting point for the backbone topology is a ring connecting all distribution networks. This ring could represent the minimum cost of WAN circuits, compromised by an initial estimate of major traffic flows, and possibly some very particular delay requirements (although this is rare, with notable exceptions being high-performance networks).

Next, existing links can be fattened, or direct connections between backbone routers can be added as required or as is cost-effective. This incremental approach should be considered when selecting WAN technologies, routing nodes, and interface types.

Backbone routing protocols, such as IBGP, properly coupled with OSPF, IS-IS, and Enhanced IGRP, can rapidly circumvent failures by simple link-costing mechanisms.

However, the bandwidth allocations with the core topology should consider failure modes. What happens when the ring is broken due to WAN or node failure? Is the re-routed path sufficient to carry the additional traffic load? Although TCP performs extremely well in congested environments compared with other protocols, it is still possible to render the network useless for most practical applications. Analysis of historical traffic levels, captured by SNMP, for example, provides for a relatively accurate estimation of the consolidated load on the remaining links during various failure modes. Traditionally, the use of a ring topology made it difficult to estimate the traffic levels between individual distribution networks. SNMP statistics, for example, provided only input and output byte counts for WAN interfaces, making it difficult to determine the appropriate sizing for new direct links between distribution networks.

Typically, this had to be accomplished using a cumbersome approach, such as "sniffers" on WAN links, or through accounting capabilities within routers that scaled rather poorly. However, IP accounting facilities, such as Netflow, now provide a scalable way for network managers to collect and analyse traffic flows, based on source and destination addresses, as well as many other flow parameters. This significantly eases traffic engineering and accounting activities. It is now possible to permanently collect and archive flow data for network design or billing purposes.



NOTE

Netflow is a high-performance switching algorithm that collects comprehensive IP accounting information and exports it to a collection agent. Load sharing is possible on the backbone network. With Cisco routers, this can be either on a per-packet or a per-flow basis. The latter usually is recommended because it avoids possible packet re-ordering, is efficiently implemented, and avoids the potential for widely varying round-trip times, which interfere with the operation of TCP. This is not a problem for per-packet load sharing over parallel WAN circuits, but it can be a problem when each alternate path is one or more routed hops.

It is possible to connect regional networks directly, avoiding the backbone altogether and possibly providing more optimal routing. For example, the DCs in SFO and LAX could be connected by a direct link. Traffic between the SFO and LAX regional networks could then travel over this link rather than

over the backbone. However, this exercise should be viewed as the effective consolidation of two regional distribution networks, and the overall routing architecture for the newly combined regions should be re-engineered to reflect this. On an operational note, the backbone network and routers may be under different operational management teams to the regional networks. In this situation, the routing relationship between the backbone and the distribution networks is likely to be slightly different because an Exterior Gateway Protocol such as BGP will be used. In this book, the operators of the backbone and regional networks are generally considered to be the same, which makes it possible for the two to share a hierarchical IGP.

3

The Role of Computer Networks in Development

The good news is that the Internet has grown like a weed, and many welcome it as a tool for productivity and enlightenment; the bad news is that it is almost unknown in developing nations. This chapter offers the hypothesis that computer networks can improve life in developing nations at a relatively low cost. We begin with a brief discussion of development, followed by some of the ways computer networks might help, and conclude with a look at what can be done.

DIMENSIONS OF DEVELOPMENT

"Development" is an imprecise concept. Economists once equated it with economic productivity—GDP per capita—but that is too simple a formulation. Rising GDP might be accompanied by environmental damage, anger over growing

disparity in income distribution, disappointment when expectations rise faster than they are fulfilled, displacement of traditional values and customs, crowded cities, and so forth. Furthermore, GDP counts many painful transactions as positive, for example, bypass surgery, buying a second home after a divorce, or the paycheck of a housewife who is forced into the labour market to make ends meet. A broader conception of human development is used in the United Nations Development Programme (UNDP) Reports on Human Development, published annually since 1992, e. g.. Their concept of development includes human autonomy and breadth of choice, equity, sustainability, and empowerment as well as productivity. In an attempt to capture this multidimensional concept of development, UNDP computes a comprehensive Human Development Index (HDI) as a function of productivity, health, and education. It is reassuring that this index shows less variance among nations than GDP per capita, still there are major disparities between nations.

With the comprehensive, UNDP concept of development in mind, let us turn to some of contributions networks might make in economic productivity, health, education, democracy, and quality of life.

Economic Productivity

Communication pays. In recent telecommunication investments in developing nations, the World Bank expects

rates of return between 13 and 20 percent, averaging about 20 percent. In addition to return on investment, they estimate 15-30 percent return to the general economy. They also find "very large economic returns" from the telecommunications components in other sectors such as railways, power, tourism, banking, and rural development.. Computer networks run over telephone infrastructure at relatively small marginal cost, providing increased economic benefit. Consider the success of the Relcom (RELiable COMmunications) network in the ex- Soviet Union. Relcom was established in April, 1989, using a Microvax in Moscow and PC compatible (286 and 386) computers connected with dial-up lines and 2400 and 9600 bps modems. On August 22, 1990, they started international Internet connectivity with hourly phone calls from Helsinki to Moscow for batch transfer of e-mail and Usenet news. By September, 1993, Relcom served nearly 7,000 organizations and an estimated 200,000 users connecting 162 regions and cities.

In spite of having begun under a communist regime, Relcom carried commercial traffic from the start, and was heavily commercial a year after it began. The network was used for markets and business communication within the nation, and for international transactions and coordination. In a 1994 Usenet News posting, Relcom co-founder Vadim Antonov said he believed the social and business impact of Relcom had been greater than that of the Internet in the US.

Relcom is unusual since networks in developing nations have usually started in the university and research community. However it is not the only example. One of the newer networks, in Ghana, accepted commercial traffic from its inception, and 36 of its 89 customers were businesses 8 months after it began.

Networks enable international communication with suppliers, customers, and other stakeholders. Much of the economic success of the US is attributable to our lead in establishing a mass, tariff-free market supported by good communication and transportation. Networks can help open mass, global markets to developing nations.

This of course raises the specter of reduced wages and environmental destruction in developed nations. The entrance of new competition will surely hurt certain industries and workers, but can this globalization be stopped? I do not think so. If that is the case, we in developed nations should console ourselves by noting the positive implications—increased investment opportunity and trade, increased efficiency, lower-cost goods, service and distribution jobs, a more just, peaceful, humane world, and so forth. Economists since Ricardo have insisted that the economic pie is largest when every person (and nation) does what he or she does best. At the same time, we should work against the negative side effects, for example, by seeking international environmental and labour standards.

Education

In 1992, Pedro Hepp and his colleagues at the Catholic University in Chile began a five-year project to develop and evaluate an elementary school network called Enlaces (links). Their goals were to enhance efficiency, quality and equity in education and to "integrate the children into the culture." They began with a pilot in six locations, and today there are 144 networked schools. Initially, each school had two computers and a 2400 bps modem. Today, there are between 3 and 10 computers and an Ethernet in each school. They began with batch transfer, making interactive access impossible, but a dozen schools are now getting IP connectivity to the Internet on a pilot basis.

Enlaces provides a variety of services—student and teacher newsletters, educational software, curriculum notes, computer conferences, e-mail, and database access. It has been formally evaluated, for example, they have shown a statistically significant effect on student creativity, and the government has decided to expand nationwide. With World Bank funding, the goal is to reach 100% of the secondary schools and 50% of the primary schools by 2000.

The support structure has been decentralized through the (long) country with 15 universities participating. One of their strong beliefs is that the teachers are at the centre of the network, and their training and support budget is 25% of the total project. From the beginning, Hepp

understood the importance of supporting low income, rural and outlying areas. A project like this not only benefits Chile, we can all learn from it. (What proportion of the schools in your community have Ethernets)?

One might argue that Chile is a prosperous developing nation, and therefore atypical. That is so, but even poor nations can make progress. The Cuban economy was dealt a debilitating blow by the fall of Communism in Eastern Europe, but has continued allocating resources to a small education network. Cuba's networking project is based in the Cuban Youth Computing Clubs (YCCs). Begun in 1987, the YCCs are typically Cuban in their stress on grass roots participation. The centres are reminiscent of Bob Albrecht's People's Computer Company (PCC) and similar experiments dating back to the 1960s in the United States. Like the PCC, they have computers running games, drawing programmes, and other software, which the children may use in a relatively unstructured manner. Additionally, the YCCs offer classes on using application packages and programming. There are now 150 YCCs spread throughout the nation. Eighty of these have 2,400 bps modems which are used to dial into shell accounts on a PC running Unix in Havana. That computer makes UUCP transfers to Canada twice daily, connecting rural Cuba to the world.

It is not coincidental that Cuba and Chile provide examples of education networks in developing nations. Both have

strong records of investment in human capital—education and health care. (Cuban and Chilean adult literacy and infant mortality are both above the averages for the UNDP "highly-developed" nations).

These examples have stressed primary and secondary schools, but networks in developing nations generally begin in the university and research community. The advantages of networks to academia are obvious—databases are shared, conferences organized, papers circulated and discussed, collaborative research and writing undertaken, and so forth. It should be noted that this is not a one- way street. Scientists in developing nations, for example, Cuban biotechnologists have much to contribute to the rest of the world.

Universities and research in developing nations will be strengthened, and the "brain drain" diminished as the Net reduces pressure on professionals to move abroad. Early in the century, physics research was concentrated in a few centres. Increased international communication—journals and conferences—led to worldwide dispersion of physics research. International meetings and journals grew, but not as rapidly as domestic activity. The Net will accelerate the spread of excellence.

Health Care

We are also seeing early application of networks to health care in developing nations. For example, HealthNet links health care workers in 16 African nations and 4 Asian nations

with each other and with colleagues and databases in developed nations using a variety of communication protocols over leased and switched land lines and terrestrial and satellite packet radio. It provides e-mail, a listserver, electronic publications, database access, distance learning, Internet consulting and support, and facilitates cooperation between libraries.

As an example application, consider the use of the ProMED (Programme for Monitoring Emerging Diseases) mailing list during the recent Ebola virus outbreak in Zaire. ProMED was established by 60 researchers in September 1993, and now has over 1,600 members in 80 countries. The list first heard of the outbreak from member Dr. Karl Johnson, the man who discovered and named the Ebola virus in Zaire in 1976. They circulated information from the U.S. Centers for Disease Control & Prevention, The World Health Organization, The Canadian Health Department, Health Canada, The Swiss Tropical Medicine Institute, The South African National Institute of Virology, and other organizations and Web sites. Subscribers provided sources and bibliographies for information about the Ebola virus and disease and reports of local reaction in Cameroon, Uganda and other countries neighbouring Zaire. They got information to and from effected nations, helping control the spread of the virus and treat the disease, and they provided objective news to the general public.

Today, most HealthNet communication is international, since intranational connectivity is still very sparse in developing nations. But, one can imagine many networking applications in healthcare in a nation like Cuba or China where "barefoot doctors" and other paramedical people serve poor communities and rural areas.

Note that HealthNet uses satellite technology, which may have great promise for developing nations. They do not use the heavy, geostationary satellites that carry television or long distance telephony, but small, low-earth-orbit (LEO) satellites. The current HealthNet satellite is capable of full-duplex, 9,600 bps communication.

Several users may request messages at the same time, but only two can be sending. The satellite is in polar orbit, so it covers the globe, with locations near the equator getting 4 daily passes of about 13 minutes. The ground stations are PC- compatibles with a controller, radio and antenna. Messages may go to any HealthNet user, satellite station, or the Internet, with Internet routing through a gateway at the Memorial University of Newfoundland. The system as typical off-line e-mail with file attachments.

Thirteen-minute uplinks at 9,600 bps will not solve the world's health communication problems, but this is an experiment which hopefully scales up. Several consortia are raising capital and beginning work on LEO satellite networks. Most ambitious is Teledesic, a venture financed by Bill Gates

and cellular entrepreneur Craig McCaw. They plan a network of 840 LEO satellites in 21, 435-mile high orbital planes, optimized for digital communication—routers in space. They are targeting two million simultaneous connections and T1 speeds, which would enable connectivity in rural clinics and villages. While many feel this is overly ambitious, they are counting on mass-produced components and technological progress for success.

HealthNet also uses terrestrial radio links, running IP over paths up to 1,000 km. Again, they are operating at a very small scale, but entrepreneurs are investing in terrestrial wireless infrastructure in developing nations. For example, the International Telecommunication Union has established WorldTel, an ambitious organization that is raising capital and beginning pilot installations for wireless telephone links to rural communities in developing nations.

Democracy and Human Rights

One might expect networks to encourage democracy by providing people living under dictatorship with outside information and ideas, and by enabling them to share ideas and coordinate political activity within their nations. For example, the Net was used for both inter and intranational communication during the failed Soviet Coup attempt, and it carried news and discussion of events in Tian An Men Square, Chiapas, and so forth. Indeed, we have a dictator's dilemma - - the Net is good for economic development, but

may undermine control. Going beyond anecdote, Kedzie presents multivariate statistical analysis showing that interconnectivity is a better predictor of democracy than schooling, GDP, life expectancy, ethnic homogeneity, or population, particularly in regions of newly emerging democracy. He also analysed the data looking for causality, finding stronger evidence for networks leading to democracy than democracy leading to networks or a spurious correlation of the two with development. Still, he concludes that "the most plausible relationship between democracy and networked communication (and perhaps economic development) may be a virtuous circle with positive feedbacks in both directions".

Many organizations supporting human rights and democracy in developing nations use the Net. The Association for Progressive Communication (APC) has been a leader in this effort since 1989, coordinating the operation and development of networks devoted to peace, ecology, human rights, and other "progressive" causes. By August 1995, there were 18 member networks, serving over 31,000 activists, educators, nonprofits and non-governmental organizations (NGOs) in over 133 countries. APC also exchanges e-mail and selected conferences with 40 partner networks. In September, 1995, APC was granted Consultative Status, Category 1, with the Economic and Social Council of the United Nations. This means they can have a permanent

representative at the UN, and are entitled to submit written statements to the Council, to be granted hearings, and to propose agenda items for consideration by the Council and its subsidiary bodies.

Quality of Life

The environment is under stress everywhere. We have pollution, and energy and other resources are limited. To the extent that networks enable us to substitute communication for transportation, they will have helped. We normally think of this effect with telecommuters in developed nations, but it may also save a rural farmer, labourer or craftsman a trip to town. More important, rural people may not move. The humanity is flocking to cities in search of better education, health care, and employment. They joke that the national bird of China is now the "crane" because so many high rise buildings are under construction, but can the environment stand the strain of 100 new high-rise Hong Kong's? And, what of congestion, traffic, crime and other side-effects of urbanization? This is not a simple defence of the noble rural life. When allowed, rural people move to the city because it provides a better life. If rural or town life can be improved, fewer may feel compelled to move.

It emphasized Hanoi and Ho Chi Minh City, as opposed to, say, networking regional capitals and fanning out from there. Implicitly or explicitly, infrastructure planning is social planning. Could a Vietnamese Net help curb urbanization

while providing urban advantages in towns and rural areas? This issue is tied to productivity—prosperous nations involve a high percentage of the population in intellectual and economic life. Perhaps breadth of choice is at the heart of quality of life. A simpler, rural life may be desirable if it is freely chosen, rather than imposed by necessity. Choice implies awareness, and communication technology expands horizons, making us aware of vocational, political, and value issues and alternatives.

ANALYSTS OF TRADITIONAL PHYSICAL SECURITY SYSTEMS

Analysts of traditional physical security systems have suggested two further design principles which, unfortunately, apply only imperfectly to computer systems.

Work Factor

Compare the cost of circumventing the mechanism with the resources of a potential attacker. The cost of circumventing, commonly known as the "work factor," in some cases can be easily calculated. For example, the number of experiments needed to try all possible four letter alphabetic passwords is $26^4 = 456\ 976$. If the potential attacker must enter each experimental password at a terminal, one might consider a four-letter password to be adequate. On the other hand, if the attacker could use a large computer capable of trying a million passwords per second, as might be the case where industrial

espionage or military security is being considered, a four-letter password would be a minor barrier for a potential intruder. The trouble with the work factor principle is that many computer protection mechanisms are *not* susceptible to direct work factor calculation, since defeating them by systematic attack may be logically impossible. Defeat can be accomplished only by indirect strategies, such as waiting for an accidental hardware failure or searching for an error in implementation. Reliable estimates of the length of such a wait or search are very difficult to make.

Compromise Recording

It is sometimes suggested that mechanisms that reliably record that a compromise of information has occurred can be used in place of more elaborate mechanisms that completely prevent loss. For example, if a tactical plan is known to have been compromised, it may be possible to construct a different one, rendering the compromised version worthless. An unbreakable padlock on a flimsy file cabinet is an example of such a mechanism.

Although the information stored inside may be easy to obtain, the cabinet will inevitably be damaged in the process and the next legitimate user will detect the loss. For another example, many computer systems record the date and time of the most recent use of each file. If this record is tamperproof and reported to the owner, it may help discover unauthorized use.

In computer systems, this approach is used rarely, since it is difficult to guarantee discovery once security is broken. Physical damage usually is not involved, and logical damage (and internally stored records of tampering) can be undone by a clever attacker. As is apparent, these principles do not represent absolute rules—they serve best as warnings. If some part of a design violates a principle, the violation is a symptom of potential trouble, and the design should be carefully reviewed to be sure that the trouble has been accounted for or is unimportant.

Considerations Surrounding Protection

Briefly, then, we may outline our discussion to this point. The application of computers to information handling problems produces a need for a variety of security mechanisms. We are focusing on one aspect, computer protection mechanisms—the mechanisms that control access to information by executing programmes. At least four levels of functional goals for a protection system can be identified: all-or-nothing systems, controlled sharing, user-programmed sharing controls, and putting strings on information. But at all levels, the provisions for dynamic changes to authorization for access are a severe complication.

Since no one knows how to build a system without flaws, the alternative is to rely on eight design principles, which tend to reduce both the number and the seriousness of any flaws: Economy of mechanism, fail-safe defaults, complete

mediation, open design, separation of privilege, least privilege, least common mechanism, and psychological acceptability. Finally, some protection designs can be evaluated by comparing the resources of a potential attacker with the work factor required to defeat the system, and compromise recording may be a useful strategy.

TECHNICAL UNDERPINNINGS

The Development Plan

At this point we begin a development of the technical basis of information protection in modern computer systems. There are two ways to approach the subject: from the top down, emphasizing the abstract concepts involved, or from the bottom up, identifying insights by, studying example systems. We shall follow the bottom-up approach, introducing a series of models of systems as they are, (or could be) built in real life. The reader should understand that on this point the authors' judgment differs from that of some of their colleagues. The top-down approach can be very satisfactory when a subject is coherent and self-contained, but for a topic still containing *ad hoc* strategies and competing world views, the bottom-up approach seems safer.

Our first model is of a multiuser system that completely isolates its users from one another. We shall then see how the logically perfect walls of that system can be lowered in a controlled way to allow limited sharing of information between users. The mechanics of sharing using two different models:

the capability system and the access control list system. It then extends these two models to handle the dynamic situation in which authorizations can change under control of the programmes running inside the system. Further extensions to the models control the dynamics. The final model (only superficially explored) is of protected objects and protected subsystems, which allow arbitrary modes of sharing that are unanticipated by the system designer. These models are not intended so much to explain the particular systems as they are to explain the underlying concepts of information protection.

Our emphasis throughout the development is on direct access to information (for example, using LOAD and STORE instructions) rather than acquiring information indirectly (as when calling a data base management system to request the average value of a set of numbers supposedly not directly accessible).

Control of such access is the function of the protected subsystems developed near the end of the paper. Herein lies perhaps the chief defect of the bottom-up approach, since conceptually there seems to be no reason to distinguish direct and indirect access, yet the detailed mechanics are typically quite different. The beginnings of a top-down approach based on a message model that avoids distinguishing between direct and indirect information access may be found in a paper by Lampson.

The Essentials of Information Protection

For purposes of discussing protection, the information stored in a computer system is not a single object. When one is considering direct access, the information is divided into mutually exclusive partitions, as specified by its various creators. Each partition contains a collection of information, all of which is intended to be protected uniformly. The uniformity of protection is the same kind of uniformity that applies to all of the diamonds stored in the same vault: any person who has a copy of the combination can obtain any of the diamonds. Thus the collections of information in the partitions are the fundamental objects to be protected.

Conceptually, then, it is necessary to build an impenetrable wall around each distinct object that warrants separate protection, construct a door in the wall through which access can be obtained, and post a guard at the door to control its use. Control of use, however, requires that the guard have some way of knowing which users are authorized to have access, and that each user have some reliable way of identifying himself to the guard. This authority check is usually implemented by having the guard demand a match between something he knows and something the prospective user possesses. Both protection and authentication mechanisms can be viewed in terms of this general model.

Before extending this model, we pause to consider two concrete examples, the multiplexing of a single computer

system among several users and the authentication of a user's claimed identity. These initial examples are complete isolation systems—no sharing of information can happen. Later we will extend our model of guards and walls in the discussion of shared information.

An Isolated Virtual Machine

A typical computer consists of a processor, a linearly addressed memory system, and some collection of input/output devices associated with the processor. It is relatively easy to use a single computer to simulate several, each of which is completely unaware of the existence of the others, except that each runs more slowly than usual.

Such a simulation is of interest, since during the intervals when one of the simulated (commonly called *virtual*) processors is waiting for an input or output operation to finish, another virtual processor may be able to progress at its normal rate. Thus a single processor may be able to take the place of several. Such a scheme is the essence of a multiprogramming system.

To allow each virtual processor to be unaware of the existence of the others, it is essential that some isolation mechanism be provided.

One such mechanism is a special hardware register called a *descriptor register*. In this figure, all memory references by the processor are checked by an extra piece of hardware that is interposed in the path to the memory. The descriptor

register controls exactly which part of memory is accessible. The descriptor register contains two components: a *base* value and a *bound* value.

The base is the lowest numbered address the programme may use, and the bound is the number of locations beyond the base that may be used. We will call the value in the descriptor register a *descriptor*, as it describes an object (in this case, one programme) stored in memory. The programme controlling the processor has full access to everything in the base-bound range, by virtue of possession of its one descriptor. As we go on, we shall embellish the concept of a descriptor: it is central to most implementations of protection and of sharing of information.

So far, we have not provided for the dynamics of a complete protection scheme: we have not discussed who loads the descriptor register. If any running programme could load it with any arbitrary value, there would be no protection. The instruction that loads the descriptor register with a new descriptor must have some special controls—either on the values it will load or on who may use it. It is easier to control who may use the descriptor, and a common scheme is to introduce an additional bit in the processor state. This bit is called the *privileged state* bit. All attempts to load the descriptor register are checked against the value of the privileged state bit; the privileged state bit must be ON for the register to be changed. One programme runs with the

privileged state bit ON, and controls the simulation of the virtual processors for the other programmes. All that is needed to make the scheme complete is to ensure that the privileged state bit cannot be changed by the user programmes except, perhaps, by an instruction that simultaneously transfers control to the supervisor programme at a planned entry location. (In most implementations, the descriptor register is not used in the privileged state.)

One might expect the supervisor programme to maintain a table of values of descriptors, one for each virtual processor. When the privileged state bit is OFF, the index in this table of the programme currently in control identifies exactly which programme—and thus which virtual processor—is accountable for the activity of the real processor. For protection to be complete, a virtual processor must not be able to change arbitrarily the values in the table of descriptors. If we suppose the table to be stored inside the supervisor programme, it will be inaccessible to the virtual processors. We have here an example of a common strategy and sometime cause of confusion: the protection mechanisms not only protect one user from another, *they may also protect their own implementation*. We shall encounter this strategy again.

So far, this virtual processor implementation contains three protection mechanisms that we can associate with our abstractions. For the first, the information being protected

is the distinct programmes. The guard is represented by the extra piece of hardware that enforces the descriptor restriction. The impenetrable wall with a door is the hardware that forces all references to memory through the descriptor mechanism. The authority check on a request to access memory is very simple. The requesting virtual processor is identified by the base and bound values in the descriptor register, and the guard checks that the memory location to which access is requested lies within the indicated area of memory.

The second mechanism protects the contents of the descriptor register. The wall, door, and guard are implemented in hardware, as with the first mechanism. An executing programme requesting to load the descriptor register is identified by the privileged state bit. If this bit is OFF, indicating that the requester is a user programme, then the guard does not allow the register to be loaded. If this bit is ON, indicating that the requester is the supervisor programme, then the guard does allow it.

The third mechanism protects the privileged state bit. It allows an executing programme identified by the privileged state bit being OFF (a user programme) to perform the single operation "turn privileged state bit ON and transfer to the supervisor programme." An executing programme identified by the privileged state bit being ON is allowed to turn the bit OFF. This third mechanism is an embryonic form of the

sophisticated protection mechanisms required to implement protected subsystems. The supervisor programme is an example of a protected subsystem, of which more will be said later.

The supervisor programme is part of all three protection mechanisms, for it is responsible for maintaining the integrity of the identifications manifest in the descriptor register and the privileged state bit. If the supervisor does not do its job correctly, virtual processors could become labelled with the wrong base and bound values, or user programmes could become labelled with a privileged state bit that is ON, The supervisor protects itself from the user programmes with the same isolation hardware that separates users, an example of the "economy of mechanism" design principle.

With an appropriately sophisticated and careful supervisor programme, we now have an example of a system that completely isolates its users from one another. Similarly isolated permanent storage can be added to such a system by attaching some longterm storage device (*e.g.*, magnetic disk) and developing a similar descriptor scheme for its use. Since long-term storage is accessed less frequently than primary memory, it is common to implement its descriptor scheme with the supervisor programmes rather than hardware, but the principle is the same. Data streams to input or output devices can be controlled similarly. The combination of a virtual processor, a memory area, some

data streams, and an isolated region of long-term storage is known as a virtual machine.

Long-term storage does, however, force us to face one further issue. Suppose that the virtual machine communicates with its user through a typewriter terminal. If a new user approaches a previously unused terminal and requests to use a virtual machine, which virtual machine (and, therefore, which set of long-term stored information) should he be allowed to use? We may solve this problem outside the system, by having the supervisor permanently associate a single virtual machine and its long-term storage area with a single terminal. Then, for example, padlocks can control access to the terminal. If, on the other hand, a more flexible system is desired, the supervisor programme must be prepared to associate any terminal with any virtual machine and, as a result, must be able to verify the identity of the user at a terminal. Schemes for performing this authentication are the subject of our next example.

Authentication Mechanisms

Our second example is of an authentication mechanism: a system that verifies a user's claimed identity. The mechanics of this authentication mechanism differ from those of the protection mechanisms for implementing virtual machines mainly because not all of the components of the system are under uniform physical control. In particular, the user himself and the communication system connecting his terminal to

the computer are components to be viewed with suspicion. Conversely, the user needs to verify that he is in communication with the expected computer system and the intended virtual machine. Such systems follow our abstract model of a guard who demands a match between something he knows and something the requester possesses. The objects being protected by the authentication mechanism are the virtual machines. In this case, however, the requester is a computer system user rather than an executing programme, and because of the lack of physical control over the user and the communication system, the security of the computer system must depend on either the secrecy or the unforgeability of the user's identification. In time-sharing systems, the most common scheme depends on secrecy. The user begins by typing the name of the person he claims to be, and then the system demands that the user type a password, presumably known only to that person.

There are, of course, many possible elaborations and embellishments of this basic strategy. In cases where the typing of the password may be observed, passwords may be good for only one use, and the user carries a list of passwords, crossing each one off the list as he uses it. Passwords may have an expiration date, or usage count, to limit the length of usefulness of a compromised one. The list of acceptable passwords is a piece of information that must be carefully guarded by the system. In some systems, all passwords are

passed through a hard-to-invert transformation before being stored, an idea suggested by R. Needham. When the user types his password, the system transforms it also and compares the transformed versions. Since the transform is supposed to be hard to invert (even if the transform itself is well known), if the stored version of a password is compromised, it may be very difficult to determine what original password is involved. It should be noted, however, that "hardness of inversion" is difficult to measure. The attacker of such a system does not need to discern the general inversion, only the particular one applying to some transformed password he has available.

Passwords as a general technique have some notorious defects. The most often mentioned defect lies in choice of password—if a person chooses his own password, he may choose something easily guessed by someone else who knows his habits. In one recent study of some 300 self-chosen passwords on a typical time-sharing system, more than 50 percent were found to be short enough to guess by exhaustion, derived from the owner's name, or something closely associated with the owner, such as his telephone number or birth date. For this reason, some systems have programmes that generate random sequences of letters for use as passwords. They may even require that all passwords be system-generated and changed frequently. On the other hand, frequently changed random sequences of letters are hard to memorize, so such

systems tend to cause users to make written copies of their passwords, inviting compromise. One solution to this problem is to provide a generator of "pronounceable" random passwords based on digraph or higher order frequency statistics to make memorization easier.

A second significant defect is that the password must be exposed to be used. In systems where the terminal is distant from the computer, the password must be sent through some communication system, during which passage a wiretapper may be able to intercept it. An alternative approach to secrecy is unforgeability.

The user is given a key, or magnetically striped plastic card, or some other unique and relatively difficult-to-fabricate object. The terminal has an input device that examines the object and transmits its unique identifying code to the computer system, which treats the code as a password that need not be kept secret. Proposals have been made for fingerprint readers and dynamic signature readers in order to increase the effort required for forgery.

The primary weakness of such schemes is that the hard-to-fabricate object, after being examined by the specialized input device, is reduced to a stream of bits to be transmitted to the computer. Unless the terminal, its object reader, and its communication lines to the computer are physically secured against tampering, it is relatively easy for an intruder to modify the terminal to transmit any sequence of bits he chooses. It

may be necessary to make the acceptable bit sequences a secret after all. On the other hand, the scheme is convenient, resists casual misuse, and provides a conventional form of accountability through the physical objects used as keys.

A problem common to both the password and the unforgeable object approach is that they are "one-way" authentication schemes. They authenticate the user to the computer system, but not *vice-versa*. An easy way for an intruder to penetrate a password system, for example, is to intercept all communications to and from the terminal and direct them to another computer—one that is under the interceptor's control. This computer can be programmed to "masquerade," that is, to act just like the system the caller intended to use, up to the point of requesting him to type his password. After receiving the password, the masquerader gracefully terminates the communication with some unsurprising error message, and the caller may be unaware that his password has been stolen. The same attack can be used on the unforgeable object system as well.

A more powerful authentication technique is sometimes used to protect against masquerading. Suppose that a remote terminal is equipped with enciphering circuitry, such as the LUCIFER system, that scrambles all signals from that terminal. Such devices normally are designed so that the exact encipherment is determined by the value of a key, known as the *encryption* or*transformation* key.

For example, the transformation key may consist of a sequence of 1000 binary digits read from a magnetically striped plastic card. In order that a recipient of such an enciphered signal may comprehend it, he must have a deciphering circuit primed with an exact copy of the transformation key, or else he must cryptanalyse the scrambled stream to try to discover the key.

The strategy of encipherment/decipherment is usually invoked for the purpose of providing communications security on an otherwise unprotected communications system. However, it can simultaneously be used for authentication, using the following technique, first published in the unclassified literature by Feistel.

The user, at a terminal, begins bypassing the enciphering equipment. He types his name. This name passes, unenciphered, through the communication system to the computer. The computer looks up the name, just as with the password system. Associated with each name, instead of a secret password, is a secret transformation key. The computer loads this transformation key into its enciphering mechanism, turns it on, and attempts to communicate with the user. Meanwhile, the user has loaded his copy of the transformation key into his enciphering mechanism and turned it on. Now, if the keys are identical, exchange of some standard hand-shaking sequence will succeed. If they are not identical, the exchange will fail, and both the user and

the computer system will encounter unintelligible streams of bits. If the exchange succeeds, the computer system is certain of the identity of the user, and the user is certain of the identity of the computer.

The secret used for authentication—the transformation key—has not been transmitted through the communication system. If communication fails (because the user is unauthorized, the system has been replaced by a masquerader, or an error occurred), each party to the transaction has immediate warning of a problem.

Relatively complex elaborations of these various strategies have been implemented, differing both in economics and in assumptions about the psychology of the prospective user. For example, Branstad explored in detail strategies of authentication in multinode computer networks. Such elaborations, though fascinating to study and analyse, are diversionary to our main topic of protection mechanisms.

Shared Information

The virtual machines are totally independent, as far as information accessibility was concerned. Each user might just as well have his own private computer system. With the steadily declining costs of computer manufacture there are few technical reasons not to use a private computer. On the other hand, for many applications some sharing of information among users is useful, or even essential. For example, there may be a library of commonly used, reliable

programmes. Some users may create new programmes that other users would like to use. Users may wish to be able to update a common data base, such as a file of airline seat reservations or a collection of programmes that implement a biomedical statistics system. In all these cases, virtual machines are inadequate, because of the total isolation of their users from one another. Before extending the virtual machine example any further, let us return to our abstract discussion of guards and walls.

Implementations of protection mechanisms that permit sharing fall into the two general categories described by Wilkes:

- "List-oriented" implementations, in which the guard holds a list of identifiers of authorized users, and the user carries a unique unforgeable identifier that must appear on the guard's list for access to be permitted. A store clerk checking a list of credit customers is an example of a list-oriented implementation in practice. The individual might use his driver's license as a unique unforgeable identifier.
- "Ticket-oriented" implementations, in which the guard holds the description of a single identifier, and each user has a collection of unforgeable identifiers, or tickets, corresponding to the objects to which he has been authorized access. A locked door that opens with a key is probably the most common example of a ticket-oriented mechanism; the guard is implemented

as the hardware of the lock, and the matching key is the (presumably) unforgeable authorizing identifier.

Authorization, defined as giving a user access to some object, is different in these two schemes. In a list-oriented system, a user is authorized to use an object by having his name placed on the guard's list for that object. In a ticket-oriented system, a user is authorized by giving him a ticket for the object.

We can also note a crucial mechanical difference between the two kinds of implementations. The list-oriented mechanism requires that the guard examine his list at the time access is requested, which means that some kind of associative search must accompany the access. On the other hand, the ticket-oriented mechanism places on the user the burden of choosing which ticket to present, a task he can combine with deciding which information to access. The guard only need compare the presented ticket with his own expectation before allowing the physical memory access. Because associative matching tends to be either slower or more costly than simple comparison, list-oriented mechanisms are not often used in applications where traffic is high. On the other hand, ticket-oriented mechanisms typically require considerable technology to control forgery of tickets and to control passing tickets around from one user to another. As a rule, most real systems contain both kinds of sharing implementations—a list-oriented system at

the human interface and a ticket-oriented system in the underlying hardware implementation. This kind of arrangement is accomplished by providing, at the higher level, a list-oriented guard whose only purpose is to hand out temporary tickets which the lower level (ticket-oriented) guards will honor. Some added complexity arises from the need to keep authorizations, as represented in the two systems, synchronized with each other. Computer protection systems differ mostly in the extent to which the architecture of the underlying ticket-oriented system is visible to the user.

Finally, let us consider the degenerate cases of list- and ticket-oriented systems. In a list-oriented system, if each guard's list of authorized users can contain only one entry, we have a "complete isolation" kind of protection system, in which no sharing of information among users can take place. Similarly, in a ticket-oriented system, if there can be only one ticket for each object in the system, we again have a "complete isolation" kind of protection system. Thus the "complete isolation" protection system turns out to be a particular degenerate case of both the list-oriented and the ticket-oriented protection implementations. These observations are important in examining real systems, which usually consist of interacting protection mechanisms, some of which are list-oriented, some of which are ticket-oriented, and some of which provide complete isolation and therefore may happen to be implemented as degenerate examples of

either of the other two, depending on local circumstances.

We should understand the relationship of a user to these transactions. We are concerned with protection of information from programmes that are executing. The user is the individual who assumes accountability for the actions of an executing programme. Inside the computer system, a programme is executed by a virtual processor, so one or more virtual processors can be identified with the activities directed by the user.

In a list-oriented system it is the guard's business to know whose virtual processor is attempting to make an access. The virtual processor has been marked with an unforgeable label identifying the user accountable for its actions, and the guard inspects this label when making access decisions. In a ticket-oriented system, however, the guard cares only that a virtual processor present the appropriate unforgeable ticket when attempting an access. The connection to an accountable user is more diffuse, since the guard does not know or care how the virtual processor acquired the tickets. In either case, we conclude that in addition to the information inside the impenetrable wall, there are two other things that must be protected: the guard's authorization information, and the association between a user and the unforgeable label or set of tickets associated with his virtual processors.

Since an association with some user is essential for establishing accountability for the actions of a virtual

processor, it is useful to introduce an abstraction for that accountability—the*principal*. A principal is, by definition, the entity accountable for the activities of a virtual processor. In the situations discussed so far, the principal corresponds to the user outside the system. However, there are situations in which a one-to-one correspondence of individuals with principals is not adequate. For example, a user may be accountable for some very valuable information and authorized to use it. On the other hand, on some occasion he may wish to use the computer for some purpose unrelated to the valuable information. To prevent accidents, he may wish to identify himself with a different principal, one that does not have access to the valuable information—following the principle of least privilege. In this case there is a need for two different principals corresponding to the same user.

Similarly, one can envision a data base that is to be modified only if a committee agrees. Thus there might be an authorized principal that cannot be used by any single individual; all of the committee members must agree upon its use simultaneously.

Because the principal represents accountability, that authorizing access is done in terms of principals. That is, if one wishes a friend to have access to some file, the authorization is done by naming a principal only that friend can use.

For each principal we may identify all the objects in the system which the principal has been authorized to use. We

will name that set of objects the *domain* of that principal. Summarizing, then, a principal is the unforgeable identifier attached to a virtual processor in a list-oriented system.

When a user first approaches the computer system, that user must identify the principal to be used. Some authentication mechanism, such as a request for a secret password, establishes the user's right to use that principal. The authentication mechanism itself may be either list- or ticket-oriented or of the complete isolation type.

Then a computation is begun in which all the virtual processors of the computation are labelled with the identifier of that principal, which is considered accountable for all further actions of these virtual processors. The authentication mechanism has allowed the virtual processor to enter the domain of that principal. That description makes apparent the importance of the authentication mechanism. Clearly, one must carefully control the conditions under which a virtual processor enters a domain.

Finally, we should note that in a ticket-oriented system there is no mechanical need to associate an unforgeable identifier with a virtual processor, since the tickets themselves are presumed unforgeable. Nevertheless, a collection of tickets can be considered to be a domain, and therefore correspond to some principal, even though there may be no obvious identifier for that principal. Thus accountability in ticket-oriented systems can be difficult to pinpoint.

Now we shall return to our example system and extend it to include sharing. Consider for a moment the problem of sharing a library programme—say, a mathematical function subroutine. We could place a copy of the math routine in the long-term storage area of each virtual machine that had a use for it.

This scheme, although workable, has several defects. Most obvious, the multiple copies require multiple storage spaces. More subtly, the scheme does not respond well to changes. If a newer, better math routine is written, upgrading the multiple copies requires effort proportional to the number of users. These two observations suggest that one would like to have some scheme to allow different users access to a single *master copy* of the programme.

The storage space will be smaller and the communication of updated versions will be easier. In terms of the virtual machine model of our earlier example, we can share a single copy of the math routine by adding to the real processor a second descriptor register, placing the math routine somewhere in memory by itself and placing a descriptor for it in the second descriptor register.

Following the previous strategy, we assume that the privileged state bit assures that the supervisor programme is the only one permitted to load either descriptor register.

In addition, some scheme must be provided in the architecture of the processor to permit a choice of which

descriptor register is to be used for each address generated by the processor.

A simple scheme would be to let the high-order address bit select the descriptor register. Thus, all addresses in the lower half of the address range would be interpreted relative to descriptor register 1, and addresses in the upper half of the address range would be relative to descriptor register 2.

An alternate scheme, suggested by Dennis, is to add explicitly to the format of instruction words a field that selects the descriptor register intended to be used with the address in that instruction.

The use of descriptors for sharing information is intimately related to the addressing architecture of the processor, a relation that can cause considerable confusion. The reason why descriptors are of interest for sharing becomes apparent by comparing parts a and b. When programme A is in control, it can have access only to itself and the math routine; similarly, when programme B is in control, it can have access only to itself and the math routine.

Since neither programme has the power to change the descriptor register, sharing of the math routine has been accomplished while maintaining isolation of programme A from programme B.

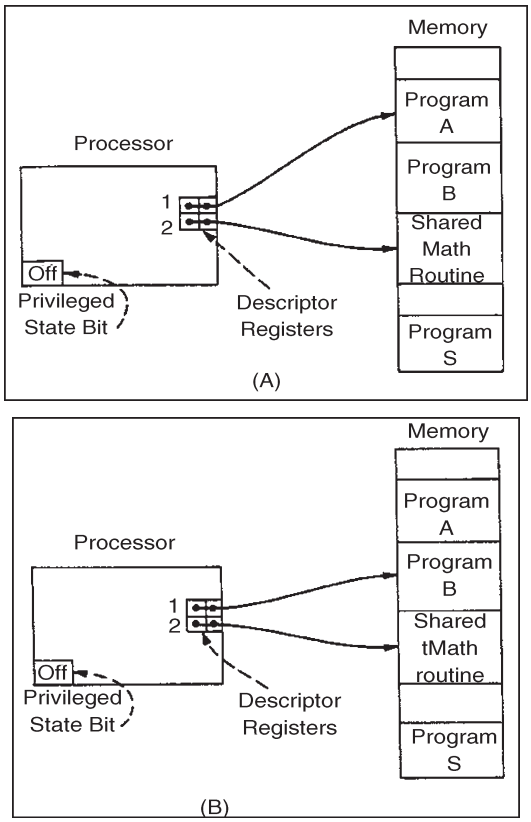


Fig. Sharing of a Math Routine by use of Two Descriptor Registors. (a) Programme A in Control of Processor. (b) Programme B in Control of Processor.

The effect of sharing is shown even more graphically redrawn with two virtual processors, one executing programme A and the other executing programme B.

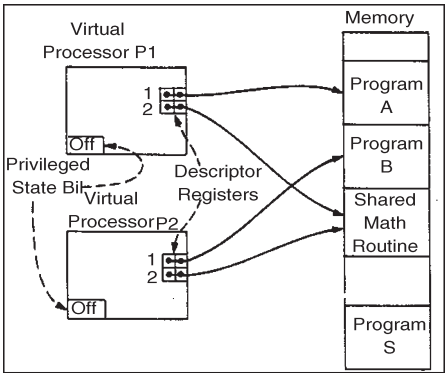


Fig. Redrawn to Show Sharing of a Math Routine by Two Virtual Processors Simultaneously.

Whether or not there are actually two processors is less important than the existence of the conceptually parallel access paths. Every virtual processor of the system may be viewed as having its own real processor, capable of access to the memory in parallel with that of every other virtual processor.

There may be an underlying processor multiplexing facility that distributes a few real processors among the many virtual processors, but such a multiplexing facility is essentially unrelated to protection. Recall that a virtual processor is not permitted to load its own protection descriptor registers. Instead, it must call or trap to the supervisor programme S which call or trap causes the privileged state bit to go ON and thereby permits the supervisor programme to control the extent of sharing among virtual processors. The processor multiplexing facility must be prepared to switch the entire state of the real processor from one virtual processor to another, including the values of the protection descriptor registers.

Although the basic mechanism to permit information sharing is now in place, a remarkable variety of implications that follow from its introduction require further mechanisms.

These implications include the following.

- If virtual processor P_1 can overwrite the shared math routine, then it could disrupt the work of virtual processor P_2.

- The shared math routine must be careful about making modifications to itself and about where in memory it writes temporary results, since it is to be used by independent computations, perhaps simultaneously.
- The scheme needs to be expanded and generalized to cover the possibility that more than one programme or data base is to be shared.
- The supervisor needs to be informed about which principals are authorized to use the shared math routine (unless it happens to be completely public with no restrictions).

Let us consider these four implications in order. If the shared area of memory is a procedure, then to avoid the possibility that virtual processor P_1 will maliciously overwrite it, we can restrict the methods of access. Virtual processor P_1 needs to retrieve instructions from the area of the shared procedure, and may need to read out the values of constants embedded in the programme, but it has no need to write into any part of the shared procedure. We may accomplish this restriction by extending the descriptor registers and the descriptors themselves to include*accessing permission*, an idea introduced for different reasons in the original Burroughs B5000 design. For example, we may add two bits, one controlling permission to read and the other permission to write in the storage area defined by each descriptor. In

virtual processor P_1, descriptor 1 would have both permissions granted, while descriptor 2 would permit only reading of data and execution of instructions. An alternative scheme would be to attach the permission bits directly to the storage areas containing the shared programme or data. Such a scheme is less satisfactory because, unlike the descriptors so far outlined, permission bits attached to the data would provide identical access to all processors that had a descriptor. Although identical access for all users of the shared math routine might be acceptable, a data base could not be set up with several users having permission to read but a few also having permission to write.

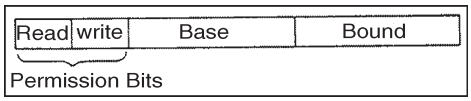


Fig. A Descriptor Containing READ and WRITE Permission Bits

The second implication of a shared procedure, mentioned before, is that the shared procedure must be careful about where it stores temporary results, since it may be used simultaneously by several virtual processors. In particular, it should avoid modifying itself. The enforcement of access permission by descriptor bits further constrains the situation. To prevent programme A from writing into the shared math routine, we have also prohibited the shared math routine from writing into itself, since the descriptors do not change when, for example, programme A transfers control to the math routine. The math routine will find that it can read but not write into itself, but that it can both read and write into

the area of programme A. Thus programme A might allocate an area of its own address range for the math routine to use as temporary storage.

As for the third implication, the need for expansion, we could generalize our example to permit several distinct shared items merely by increasing the number of descriptor registers and informing the supervisor which shared objects should be addressable by each virtual processor. However, there are two substantially different forms of this generalization—*capability systems* and *access control list systems*. The capability systems are ticket-oriented, while access control list systems are list-oriented. Most real systems use a combination of these two forms, the capability system for speed and an access control list system for the human interface. Before we can pursue these generalizations, and the fourth implication, authorization, more groundwork must be laid.

The development of protection continues with a series of successively more sophisticated models. The initial model, of a capability system, explores the use of encapsulated but copyable descriptors as tickets to provide a flexible authorization scheme. In this context we establish the general rule that communication external to the computer must precede dynamic authorization of sharing. The limitations of copyable descriptors—primarily lack of accountability for their use—lead to analysis of revocation and the observation

that revocation requires indirection. That observation in turn leads to the model of access control lists embedded in indirect objects so as to provide detailed control of authorization.

The use of access control lists leads to a discussion of controlling changes to authorizations, there being at least two models of control methods which differ in their susceptibility to abuse. Additional control of authorization changes is needed when releasing sensitive data to a borrowed programme, and this additional control implies a nonintuitive constraint on where data may be written by the borrowed programme. The concept of implementing arbitrary abstractions, such as extended types of objects, as programmes in separate domains.

COLLECTION OF NETWORK ELEMENTS

A CDN is a collection of network elements arranged for more effective delivery of content to end-users. Collaboration among distributed CDN components can occur over nodes in both homogeneous and heterogeneous environments. CDNs can take various forms and structures. They can be centralized, hierarchical infrastructure under certain administrative control, or completely decentralized systems. There can also be various forms of internetworking and control sharing among different CDN entities. General considerations on designing a CDN can be found.

The typical functionality of a CDN includes:

- Request redirection and content delivery services to direct a request to the closest suitable surrogate server using mechanisms to bypass congestion, thus overcoming flash crowds or SlashDot effects.
- Content outsourcing and distribution services to replicate and/or cache content to distributed surrogate servers on behalf of the origin server.
- Content negotiation services to meet specific needs of each individual user (or group of users).
- Management services to manage the network components, to handle accounting, and to monitor and report on content usage.

A CDN provides better performance through caching or replicating content over some mirrored Web servers (*i.e.* surrogate servers) strategically placed at various locations in order to deal with the sudden spike in Web content requests, which is often termed as flash crowd or SlashDot effect. The users are redirected to the surrogate server nearest to them. This approach helps to reduce network impact on the response time of user requests. In the context of CDNs, content refers to any digital data resources and it consists of two main parts: the encoded media and metadata. The encoded media includes static, dynamic and continuous media data (*e.g.* audio, video, documents, images and Web pages). Metadata is the content description that allows

identification, discovery, and management of multimedia data, and also facilitates the interpretation of multimedia data. Content can be pre-recorded or retrieved from live sources; it can be persistent or transient data within the system.

CDNs can be seen as a new virtual overlay to the Open Systems Interconnection (OSI) basic reference model. This layer provides overlay network services relying on application layer protocols such as HTTP or RTSP for transport. The three key components of a CDN architecture are – content provider, CDN provider and end-users. A content provider or customer is one who delegates the URI name space of the Web objects to be distributed. The origin server of the content provider holds those objects. A CDN provider is a proprietary organization or company that provides infrastructure facilities to content providers in order to deliver content in a timely and reliable manner. End-users or clients are the entities who access content from the content provider's website.

CDN providers use caching and/or replica servers located in different geographical locations to replicate content. CDN cache servers are also called edge servers or surrogates. In this chapter, we will use these terms interchangeably. The surrogates of a CDN are called Web cluster as a whole. CDNs distribute content to the surrogates in such a way that all cache servers share the same content and URL. Client requests are redirected to the nearby surrogate, and a

selected surrogate server delivers requested content to the end-users. Thus, transparency for users is achieved. Additionally, surrogates send accounting information for the delivered content to the accounting system of the CDN provider.

The Evolution of CDNs

Over the last decades, users have witnessed the growth and maturity of the Internet. As a consequence, there has been an enormous growth in network traffic, driven by rapid acceptance of broadband access, along with increases in system complexity and content richness. The over-evolving nature of the Internet brings new challenges in managing and delivering content to users. As an example, popular Web services often suffer congestion and bottleneck due to the large demands made on their services. A sudden spike in Web content requests may cause heavy workload on particular Web server(s), and as a result a hotspot can be generated. Coping with such unexpected demand causes significant strain on a Web server. Eventually the Web servers are totally overwhelmed with the sudden increase in traffic, and the Web site holding the content becomes temporarily unavailable.

Content providers view the Web as a vehicle to bring rich content to their users. A decrease in service quality, along with high access delays mainly caused by long download times, leaves the users in frustration. Companies earn

significant financial incentives from Web-based e-business. Hence, they are concerned to improve the service quality experienced by the users while accessing their Web sites. As such, the past few years have seen an evolution of technologies that aim to improve content delivery and service provisioning over the Web. When used together, the infrastructures supporting these technologies form a new type of network, which is often referred to as content network.

Several content networks attempt to address the performance problem through using different mechanisms to improve the Quality of Service (QoS). One approach is to modify the traditional Web architecture by improving the Web server hardware adding a high-speed processor, more memory and disk space, or maybe even a multi-processor system. This approach is not flexible. Moreover, small enhancements are not possible and at some point, the complete server system might have to be replaced. Caching proxy deployment by an ISP can be beneficial for the narrow bandwidth users accessing the Internet.

In order to improve performance and reduce bandwidth utilization, caching proxies are deployed close to the users. Caching proxies may also be equipped with technologies to detect a server failure and maximize efficient use of caching proxy resources. Users often configure their browsers to send their Web request through these caches rather than sending directly to origin servers. When this configuration is properly

done, the user's entire browsing session goes through a specific caching proxy. Thus, the caches contain most popular content viewed by all the users of the caching proxies.

A provider may also deploy different levels of local, regional, international caches at geographically distributed locations. Such arrangement is referred to as hierarchical caching. This may provide additional performance improvements and bandwidth savings.

A more scalable solution is the establishment of server farms. It is a type of content network that has been in widespread use for several years. A server farm is comprised of multiple Web servers, each of them sharing the burden of answering requests for the same Web site. It also makes use of a Layer 4-7 switch, Web switch or content switch that examines content request and dispatches them among the group of servers. A server farm can also be constructed with surrogates instead of a switch.

This approach is more flexible and shows better scalability. Moreover, it provides the inherent benefit of fault tolerance. Deployment and growth of server farms progresses with the upgrade of network links that connects the Web sites to the Internet. Although server farms and hierarchical caching through caching proxies are useful techniques to address the Internet Web performance problem, they have limitations. In the first case, since servers are deployed near the origin server, they do little to improve the network performance

due to network congestion. Caching proxies may be beneficial in this case. But they cache objects based on client demands. This may force the content providers with a popular content source to invest in large server farms, load balancing, and high bandwidth connections to keep up with the demand. To address these limitations, another type of content network has been deployed in late 1990s. This is termed as Content Distribution Network or Content Delivery Network, which is a system of computers networked together across the Internet to cooperate transparently for delivering content to end-users.

With the introduction of CDN, content providers started putting their Web sites on a CDN. Soon they realised its usefulness through receiving increased reliability and scalability without the need to maintain expensive infrastructure. Hence, several initiatives kicked off for developing infrastructure for CDNs. As a consequence, Akamai Technologies evolved out of an MIT research effort aimed at solving the flash crowd problem. Within a couple of years, several companies became specialists in providing fast and reliable delivery of content, and CDNs became a huge market for generating large revenues. The flash crowd events like the 9/11 incident in USA, resulted in serious caching problems for some site. This influenced the CDN providers to invest more in CDN infrastructure development, since CDNs provide desired level of protection to Web sites against flash crowds.

First generation CDNs mostly focused on static or Dynamic Web documents. On the other hand, for second generation of CDNs the focus has shifted to Video-on-Demand (VoD), audio and video streaming. But they are still in research phase and have not reached to the market yet. With the booming of the CDN business, several standardization activities also emerged since vendors started organizing themselves. The Internet Engineering Task Force (IETF) as a official body took several initiatives through releasing RFCs (Request For Comments). Other than IETF, several other organizations such as Broadband Services Forum (BSF), ICAP forum, Internet Streaming Media Alliance took initiatives to develop standards for delivering broadband content, streaming rich media content – video, audio, and associated data – over the Internet.

In the same breath, by 2002, large-scale ISPs started building their own CDN functionality, providing customized services. In 2004, more than 3000 companies were found to use CDNs, spending more than $20 million monthly. A market analysis shows that CDN providers have doubled their earnings from streaming media delivery in 2004 compared to 2003. In 2005, CDN revenue for both streaming video and Internet radio was estimated to grow at 40%.

A recent marketing research shows that combined commercial market value for streaming audio, video, streaming audio and video advertising, download media and

entertainment was estimated at between $385 million to $452 million in 2005. Considering this trend, the market was forecasted to reach $2 billion in four-year (2002-2006) total revenue in 2006, with music, sports, and entertainment subscription and download revenue for the leading content categories.

However, the latest report from AccuStream iMedia Research reveals that since 2002, the CDN market has invested $1.65 billion to deliver streaming media (excluding storage, hosting, applications layering), and the commercial market value in 2006 would make up 36% of the $1.65 billion four-year total in media and entertainment, including content, streaming advertising, movie and music downloads and User Generated Video (UGV) distribution.

Insight into CDNs

A typical content delivery environment where the replicated Web server clusters are located at the edge of the network to which the end-users are connected. A content provider (*i.e.* customer) can sign up with a CDN provider for service and have its content placed on the content servers.

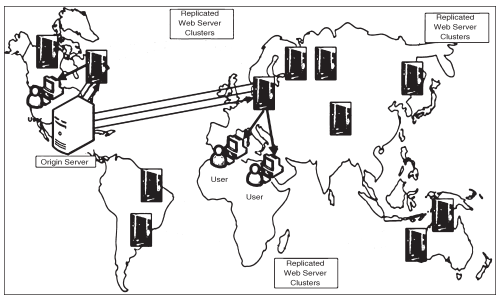


Fig. Abstract Architecture of a Content Delivery Network (CDN)

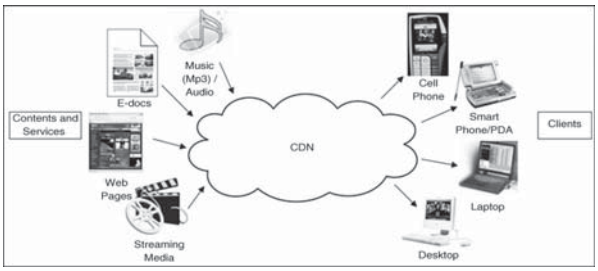


Fig. Content/services Provided by a CDN

The content is replicated either on-demand when users request for it, or it can be replicated beforehand, by pushing the content to the surrogate servers. A user is served with the content from the nearby replicated Web server. Thus, the user ends up unknowingly communicating with a replicated CDN server close to it and retrieves files from that server. CDN providers ensure the fast delivery of any digital content. They host third-party content including static content (*e.g.* static HTML pages, images, documents, software patches), streaming media (*e.g.* audio, real time video), User Generated Videos (UGV), and varying content services (*e.g.* directory service, e-commerce service, file transfer service). The sources of content include large enterprises, Web service providers, media companies and news broadcasters. The end-users can interact with the CDN by specifying the content/service request through cell phone, smart phone/PDA, laptop and desktop. The different content/services served by a CDN provider to end-users.

CDN providers charge their customers according to the content delivered (*i.e.* traffic) to the end-users by their surrogate servers. CDNs support an accounting mechanism

that collects and tracks client usage information related to request-routing, distribution and delivery. This mechanism gathers information in real time and collects it for each CDN component. This information can be used in CDNs for accounting, billing and maintenance purposes. The average cost of charging of CDN services is quite high, often out of reach for many small to medium enterprises (SME) or not-for-profit organizations.

The most influencing factors affecting the price of CDN services include:

- Bandwidth cost
- Variation of traffic distribution
- Size of content replicated over surrogate servers
- Number of surrogate servers
- Reliability and stability of the whole system and security issues of outsourcing content delivery A CDN is essentially aimed at content providers or customers who want to ensure QoS to the end-users while accessing their Web content.

The analysis of present day CDNs reveals that, at the minimum, a CDN focuses on the following business goals: scalability, security, reliability, responsiveness and performance.

Scalability: The main business goal of a CDN is to achieve scalability. Scalability refers to the ability of the system to expand in order to handle new and large amounts of data,

users and transactions without any significant decline in performance. To expand in a global scale, CDNs need to invest time and costs in provisioning additional network connections and infrastructures. It includes provisioning resources dynamically to address flash crowds and varying traffic. A CDN should act as a shock absorber for traffic by automatically providing capacity-on-demand to meet the requirements of flash crowds. This capability allows a CDN to avoid costly over-provisioning of resources and to provide high performance to every user.

Security: One of the major concerns of a CDN is to provide potential security solutions for confidential and high-value content. Security is the protection of content against unauthorized access and modification. Without proper security control, a CDN platform is subject to cyber fraud, distributed denial-of-service (DDoS) attacks, viruses, and other unwanted intrusions that can cripple business. A CDN aims at meeting the stringent requirements of physical, network, software, data and procedural security. Once the security requirements are met, a CDN can eliminate the need for costly hardware and dedicated component to protect content and transactions. In accordance to the security issues, a CDN combat against any other potential risk concerns including denial-of-service attacks or other malicious activity that may interrupt business.

Reliability, Responsiveness and Performance: Reliability refers to when a service is available and what are the bounds on service outages that may be expected. A CDN provider can improve client access to specialized content through delivering it from multiple locations. For this a fault-tolerant network with appropriate load balancing mechanism is to be implemented. Responsiveness implies, while in the face of possible outages, how soon a service would start performing the normal course of operation. Performance of a CDN is typically characterized by the response time (*i.e.* latency) perceived by the end-users. Slow response time is the single greatest contributor to customers' abandoning Web sites and processes. The reliability and performance of a CDN is affected by the distributed content location and routing mechanism, as well as by data replication and caching strategies.

Hence, a CDN employs caching and streaming to enhance performance especially for delivery of media content. A CDN hosting a Web site also focuses on providing fast and reliable service since it reinforces the message that the company is reliable and customer-focused.

4

Security in Networks

Networks their design, development, and usage are critical to our style of computing. We interact with networks daily, when we perform banking transactions, make telephone calls, or ride trains and planes. The utility companies use networks to track electricity or water usage and bill for it. When we pay for groceries or gasoline, networks enable our credit or debit card transactions and billing. Life without networks would be considerably less convenient, and many activities would be impossible. Not surprisingly, then, computing networks are attackers' targets of choice. Because of their actual and potential impact, network attacks attract the attention of journalists, managers, auditors, and the general public. For example, when you read the daily newspapers, you are likely to find a story about a network-based attack

at least every month. The coverage itself evokes a sense of evil, using terms such as hijacking, distributed denial of service, and our familiar friends viruses, worms, and Trojan horses. Because any large-scale attack is likely to put thousands of computing systems at risk, with potential losses well into the millions of dollars, network attacks make good copy.

The media coverage is more than hype; network attacks are critical problems. Fortunately, your bank, your utility company, and even your Internet service provider take network security very seriously. Because they do, they are vigilant about applying the most current and most effective controls to their systems. Of equal importance, these organizations continually assess their risks and learn about the latest attack types and defence mechanisms so that they can maintain the protection of their networks.

In this chapter we describe what makes a network similar to and different from an application programme or an operating system. In investigating networks, you will learn how the concepts of confidentiality, integrity, and availability apply in networked settings. At the same time, you will see that the basic notions of identification and authentication, access control, accountability, and assurance are the basis for network security, just as they have been in other settings.

Networking is growing and changing perhaps even faster than other computing disciplines. Consequently, this chapter

is unlikely to present you with the most current technology, the latest attack, or the newest defence mechanism; you can read about those in daily newspapers and at web sites. But the novelty and change build on what we know today: the fundamental concepts, threats, and controls for networks. By developing an understanding of the basics, you can absorb the most current news quickly and easily. More importantly, your understanding can assist you in building, protecting, and using networks.

NETWORK CONCEPTS

To study network threats and controls, we first must review some of the relevant networking terms and concepts. This review does not attempt to provide the depth of a classic networking reference. Our study of security focused on the individual pieces of a computing system, such as a single application, an operating system, or a database. Networks involve not only the pieces but also importantly the connections among them.

Networks are both fragile and strong. To see why, think about the power, cable television, telephone, or water network that serves your home. If a falling tree branch breaks the power line to your home, you are without electricity until that line is repaired; you are vulnerable to what is called a single point of failure , because one cut to the network destroys electrical functionality for your entire home. Similarly, there may be one telephone trunk line or water

main that serves your home and those nearby; a failure can leave your building, street, or neighbourhood without service. But we have ways to keep the entire network from failing. If we trace back through the network from your home to the source of what flows through it, we are likely to see that several main distribution lines support an entire city or campus. That is, there is more than one way to get from the source to your neighbourhood, enabling engineers to redirect the flow along alternative paths. Redundancy makes it uncommon for an entire city to lose service from a single failure. For this reason, we say that such a network has resilience or fault tolerance.

Complex routing algorithms reroute the flow not just around failures but also around overloaded segments. The routing is usually done automatically; the control programme is often supplemented by human supervision or intervention. Many types of networks have very high reliability by design, not by accident. But because there often is less redundancy near a network's endpoints than elsewhere, we say that the network has great strength in the middle and fragility at the perimeter.

From the user's perspective, a network is sometimes designed so that it looks like two endpoints with a single connection in the middle. For example, the municipal water supply may appear to be little more than a reservoir (the source), the pipes (the transmission or communication

medium), and your water faucet (the destination). Although this simplistic view is functionally correct, it ignores the complex design, implementation, and management of the "pipes." In a similar way, we describe computer networks in this chapter in ways that focus on the security concepts but present the networks themselves in a simplistic way, to highlight the role of security and prevent the complexity of the networks from distracting our attention. Please keep in mind that our network descriptions are often abstractions of a more complex actuality.

The Network

A network in its simplest form, as two devices connected across some medium by hardware and software that enable the communication. In some cases, one device is a computer (sometimes called a "server") and the other is a simpler device (sometimes called a "client") enabled only with some means of input (such as a keyboard) and some means of output (such as a screen). For example, a powerful computer can be a server, but a handheld personal digital assistant (PDA) or a cell phone might be a network client. In fact, because more consumer devices are becoming network-enabled, network security issues will continue to grow.

Fig. Simple View of Network.

Although this model defines a basic network, the actual situation is frequently significantly more complicated.

- The simpler client device, employed for user-to-computer communication, is often a PC or workstation, so the client has considerable storage and processing capability.
- A network can be configured as just a single client connected to a single server. But more typically, many clients interact with many servers.
- The network's services are often provided by many computers. As a single user's communication travels back and forth from client to server, it may merely pass through some computers but pause at others for significant interactions.
- The end user is usually unaware of many of the communications and computations taking place in the network on the user's behalf.

The user at one of the lettered client machines may send a message to System 3, unaware that communication is actually passing through the active Systems 1 and 2. In fact, the user may be unaware that System 3 sometimes passes work to System 4.

A single computing system in a network is often called a node, and its processor (computer) is called a host . A connection between two hosts is known as a link . Network computing consists of users, communications media, visible

hosts, and systems not generally visible to end users. Systems 1 through 4 are nodes. In our figure the users are at the lettered client machines, perhaps interacting with Server F.

Users communicate with networked systems by interacting directly with terminals, workstations, and computers. A workstation is an end-user computing device, usually designed for a single user at a time. Workstations often have powerful processors and good- sized memory and storage so that they can do sophisticated data manipulation (such as converting coded data to a graphical format and displaying the picture). A system is a collection of processors, perhaps including a mixture of workstations and independent processors, typically with more processing power and more storage capacity than a workstation.

Environment of Use

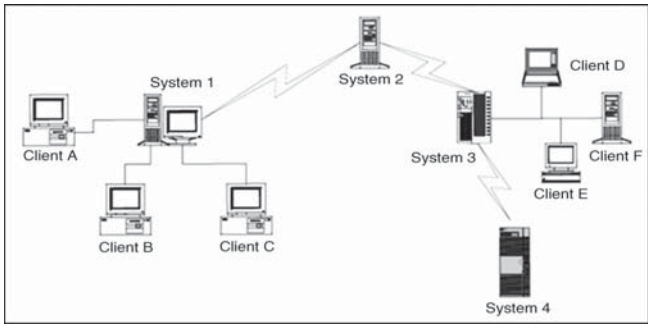


Fig. More Complex but More Typical View of Networks.

The biggest difference between a network and a stand-alone device is the environment in which each operates. Although some networks are located in protected spaces (for example, a local area network in a single laboratory or office),

at least some portion of most networks is exposed, often to total strangers. The relatively simple network is a good example. Systems 2, 3, and 4 are remote from System 1, and they may be under different ownership or control.

Networks can be described by several typical characteristics:

- *Anonymity.* You may have seen the cartoon image that shows a dog typing at a workstation, and saying to another dog, "On the Internet, nobody knows you're a dog." A network removes most of the clues, such as appearance, voice, or context, by which we recognize acquaintances.
- *Automation.* In some networks, one or both endpoints, as well as all intermediate points, involved in a given communication may be machines with only minimal human supervision.
- *Distance.* Many networks connect endpoints that are physically far apart. Although not all network connections involve distance, the speed of communication is fast enough that humans usually cannot tell whether a remote site is near or far.
- *Opaqueness.* Because the dimension of distance is hidden, users cannot tell whether a remote host is in the room next door or in a different country. In the same way, users cannot distinguish whether they are connected to a node in an office, school, home, or warehouse, or whether the node's computing system

is large or small, modest or powerful. In fact, users cannot tell if the current communication involves the same host with which they communicated the last time.

- *Routing diversity*. To maintain or improve reliability and performance, routings between two endpoints are usually dynamic. That is, the same interaction may follow one path through the network the first time and a very different path the second time. In fact, a query may take a different path from the response that follows a few seconds later.

Shape and Size

The way a network is configured, in terms of nodes and connections, is called the network topology. You can think of the topology as the shape of the network.

The topology ranges from very simple, such as two hosts connected by one path, to very complex, such as the Internet.

These two extremes highlight three dimensions of networks that have particular bearing on a network's security.

- *Boundary.* The boundary distinguishes an element of the network from an element outside it. For a simple network, we can easily list all the components and draw an imaginary line around it to separate what is in the network from what is outside. But listing all the hosts connected to the Internet is practically

impossible. For example, a line surrounding the Internet would have to surround the entire globe today, and Internet connections also pass through satellites in orbit around the earth. Moreover, as people and organizations choose to be connected or not, the number and type of hosts change almost second by second, with the number generally increasing over time.

- *Ownership.* It is often difficult to know who owns each host in a network. The network administrator's organization may own the network infrastructure, including the cable and network devices. However, certain hosts may be connected to a network for convenience, not necessarily implying ownership.
- *Control.* Finally, if ownership is uncertain, control must be, too. To see how, pick an arbitrary host. Is it part of network A? If yes, is it under the control of network A's administrator? Does that administrator establish access control policies for the network, or determine when its software must be upgraded and to what version? Indeed, does the administrator even know what version of software that host runs?

The truth is that, for many networks, it is difficult and at times impossible to tell which hosts are part of that network, who owns the hosts, and who controls them. Even for networks significantly smaller than the Internet, major

corporate, university, or government networks are hard to understand and are not even well known by their system administrators. Although it seems contrary to common sense, many corporations today have no accurate picture of how their networks are configured. To understand why, consider a network of automated teller machines for a multinational bank. The bank may have agreements with other banks to enable customers to withdraw money anywhere in the world. The multinational bank may understand its own bank's network, but it may have no conception of how the connecting banks' networks are configured; no "big picture" shows how the combined networks look or operate. Similarly, a given host may be part of more than one network. In such a situation, suppose a host has two network interfaces. Whose rules does that host (and that host's administrator) have to follow? Depicting, configuring, and administering networks are not easy tasks.

Mode of Communication

A computer network implements communication between two endpoints. Data are communicated either in digital format (in which data items are expressed as discrete binary values) or analog (in which data items are expressed as points in a continuous range, using a medium like sound or electrical voltage). Computers typically store and process digital data, but some telephone and similar cable communications are in analog form (because telephones were

originally designed to transmit voice). When the transmission medium expects to transfer analog data, the digital signals must be converted to analog for transmission and then back to digital for computation at the receiving end. Some mostly analog networks may even have some digital segments, so the analog signals are digitized more than once. These conversions are performed by a modem, which converts a digital data stream to tones and back again.

Media

Communication is enabled by several kinds of media. We can choose among several types, such as along copper wires or optical fibre or through the air, as with cellular phones. Let us look at each type in turn.

Cable

Because much of our computer communication has historically been done over telephone lines, the most common network communication medium today is wire. Inside our homes and offices, we use a pair of insulated copper wires, called a twisted pair or unshielded twisted pair (UTP). Copper has good transmission properties at a relatively low cost. The bandwidth of UTP is limited to under 10 megabits per second (Mbps), so engineers cannot transmit a large number of communications simultaneously on a single line. Moreover, the signal strength degrades as it travels through the copper wire, and it cannot travel long distances without a boost.

Thus, for many networks, line lengths are limited to approximately 300 feet. Single twisted pair service is most often used locally, within a building or up to a local communications drop (that is, the point where the home or office service is connected to the larger network, such as the commercial telephone system). Although regular copper wire can transmit signals, the twisting reduces crossover (interference and signal transfer) between adjacent wires.

However, as speeds or capacities change, the basic ranking of two technologies tends to remain the same. Another choice for network communication is coaxial (coax) cable, the kind used for cable television. Coax cable is constructed with a single wire surrounded by an insulation jacket. The jacket is itself surrounded by a braided or spiral-wound wire. The inner wire carries the signal, and the outer braid acts as a ground. The most widely used computer communication coax cable is Ethernet, carrying up to 100 Mbps over distances of up to 1500 feet.

Coax cable also suffers from degradation of signal quality over distance. Repeaters (for digital signals) or amplifiers (for analog signals) can be spaced periodically along the cable to pick up the signal, amplify it, remove spurious signals called "noise," and retransmit it.

Optical Fibre

A newer form of cable is made of very thin strands of glass. Instead of carrying electrical energy, these fibres carry pulses

of light. The bandwidth of optical fibre is up to 1000 Mbps, and the signal degrades less over fibre than over wire or coax; the fibre is good for a run of approximately 2.5 miles. Optical fibre involves less interference, less crossover between adjacent media, lower cost, and less weight than copper. Thus, optical fibre is generally a much better transmission medium than copper. Consequently, as copper ages, it is being replaced by optical fibre in most communication systems. In particular, most long distance communication lines are now fibre.

Wireless

Radio signals can also carry communications. Similar to pagers, wireless microphones, garage door openers, and portable telephones, wireless radio can be used in networks, following a protocol developed for short-range telecommunications, designated the 802.11 family of standards. The wireless medium is used for short distances; it is especially useful for networks in which the nodes are physically close together, such as in an office building or at home. Many 802.11 devices are becoming available for home and office wireless networks.

Microwave

Microwave is a form of radio transmission especially well suited for outdoor communication. Microwave has a channel capacity similar to coax cable; that is, it carries similar

amounts of data. Its principal advantage is that the signal is strong from point of transmission to point of receipt. Therefore, microwave signals do not need to be regenerated with repeaters, as do signals on cable. However, a microwave signal travels in a straight line, presenting a problem because the earth curves. Microwave signals travel by line of sight: The transmitter and receiver must be in a straight line with one another, with no intervening obstacles, such as mountains. A straight microwave signal transmitted between towers of reasonable height can travel a distance of only about 30 miles because of the earth's curvature. Thus, microwave signals are "bounced" from receiver to receiver, spaced less than 30 miles apart, to cover a longer distance.

Infrared

Infrared communication carries signals for short distances (up to 9 miles) and also requires a clear line of sight. Because it does not require cabling, it is convenient for portable objects, such as laptop computers and connections to peripherals.

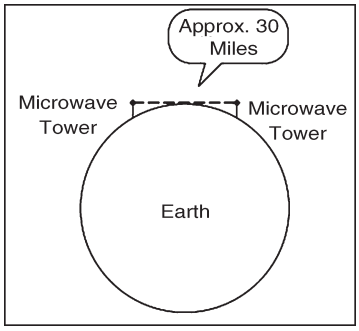


Fig. Microwave Transmission.

An infrared signal is difficult to intercept because it is a point-to-point signal. However, it is subject to "in the middle" attacks in which the interceptor functions like a repeater, receiving the signal, extracting any desired data, and retransmitting to the original destination the original signal or a modified version. Because of line-of-sight requirements and limited distance, infrared is typically used in a protected space, such as an office, in which in-the-middle attacks would be difficult to conceal.

Satellite

Many communications, such as international telephone calls, must travel around the earth. In the early days of telephone technology, telephone companies ran huge cables along the ocean's bottom, enabling calls to travel from one continent to another. Today, we have other alternatives. The communication companies place satellites in orbits that are synchronized with the rotation of the earth (called geosynchronous orbits), so the satellite appears to hover in a fixed position 22,300 miles above the earth. Although the satellite can be expensive to launch, once in space it is essentially maintenance free. Furthermore, the quality of a satellite communication link is often better than an earthbound wire cable.

Satellites act as nave transponders : Whatever they receive they broadcast out again. Thus, satellites are really sophisticated receivers, in that their sole function is to receive

and repeat signals. From the user's point of view, the signal essentially "bounces" off the satellite and back to earth. For example, a signal from North America travels 22,300 miles into the sky and the same distance back to a point in Europe.

We can project a signal to a satellite with reasonable accuracy, but the satellite is not expected to have the same level of accuracy when it sends the signal back to earth. To reduce complexity and eliminate beam focusing, satellites typically spread their transmissions over a very wide area. A rather narrow angle of dispersion from the satellite's transmitter produces a fairly broad pattern (called the footprint) on the surface of the earth because of the 22,300-mile distance from the satellite to earth. Thus, a typical satellite transmission can be received over a path several hundred miles wide; some cover the width of the entire continental United States in a single transmission. For some applications, such as satellite television, a broad footprint is desirable. But for secure communications, the smaller the footprint, the less the risk of interception.

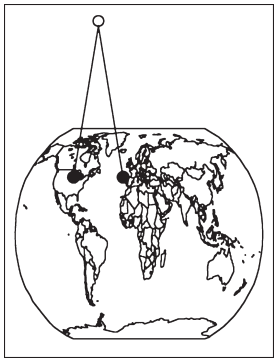


Fig. Satellite Communication.

Protocols

When we use a network, the communication media are usually transparent to us. That is, most of us do not know whether our communication is carried over copper wire, optical fibre, satellite, microwave, or some combination. In fact, the communication medium may change from one transmission to the next. This ambiguity is actually a positive feature of a network: its independence. That is, the communication is separated from the actual medium of communication. Independence is possible because we have defined protocols that allow a user to view the network at a high, abstract level of communication (viewing it in terms of user and data); the details of how the communication is accomplished are hidden within software and hardware at both ends. The software and hardware enable us to implement a network according to a protocol stack, a layered architecture for communications. Each layer in the stack is much like a language for communicating information relevant at that layer. Two popular protocol stacks are used frequently for implementing networks: the Open Systems Interconnection (OSI) and the Transmission Control Protocol and Internet Protocol (TCP/IP) architecture. We examine each one in turn.

ISO/OSI Reference Model

The International Standards Organization (ISO)/ Open Systems Interconnection model consists of layers by which a network communication occurs.

How communication works across the different layers. We can think of the layers as creating an assembly line, in which each layer adds its own service to the communication. In concert, the layers represent the different activities that must be performed for actual transmission of a message. Separately, each layer serves a purpose; equivalent layers perform similar functions for the sender and receiver. For example, the sender's layer four affixes a header to a message, designating the sender, the receiver, and relevant sequence information. On the receiving end, layer four reads the header to verify that the message is for the intended recipient, and then removes this header.

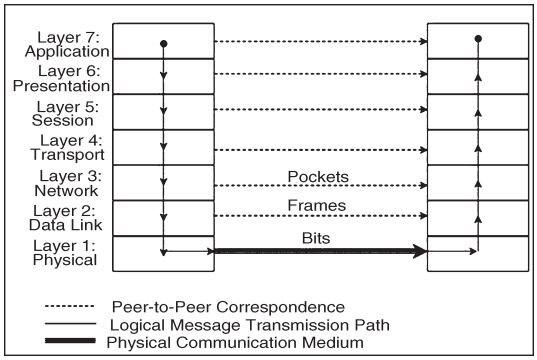


Fig. ISO/ OSI Network Model.

Each layer passes data in three directions: above with a layer communicating more abstractly, parallel or across to the same layer in another host, and below with a layer handling less abstract (that is, more fundamental) data items. The communications above and below are actual interactions, while the parallel one is a virtual communication path. Parallel layers are called "peers."

Let us look at a simple example of protocol transmission. Suppose that, to send e-mail to a friend, you run an application such as Eudora, Outlook, or Unix mail. You type a message, using the application's editor, and the application formats the message into two parts: a header that shows to whom the message is intended (as well as other things, such as sender and time sent), and a body that contains the text of your message. The application reformats your message into a standard format so that even if you and your friend use different mail applications, you can still exchange e-mail. This transformation is shown in Figure below.

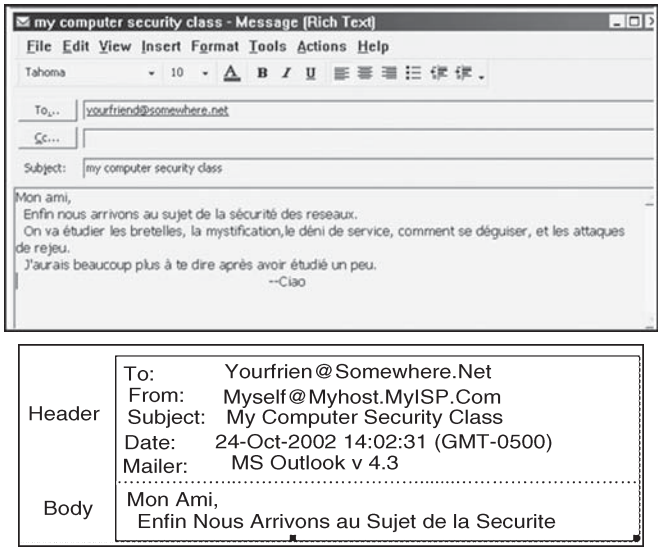


Fig. Transformation.

However, the message is not transmitted exactly as you typed it, as raw text. Raw text is a very inefficient coding, because an alphabet uses relatively few of the 255 possible characters for an 8-bit byte. Instead, the presentation layer is likely to change the raw text into something else. It may

do compression, character conversions, and even some cryptography. An e-mail message is a one-way transfer (from sender to receiver), so it is not initiating a session in which data fly back and forth between the two endpoints. Because the notion of a communication session is not directly relevant in this scenario, we ignore the session layer for now. Occasionally, spurious signals intrude in a communication channel, as when static rustles a telephone line or interference intrudes on a radio or television signal. To address this, the transport layer adds error detection and correction coding to filter out these spurious signals.

Addressing

Suppose your message is addressed to yourfriend@somewhere.net. This notation means that "somewhere.net" is the name of a destination host (or more accurately, a destination network). At the network layer, a hardware device called a router actually sends the message from your network to a router on the network somewhere.net. The network layer adds two headers to show your computer's address as the source and somewhere.net's address as the destination. Logically, your message is prepared to move from your machine to your router to your friend's router to your friend's computer. (In fact, between the two routers there may be many other routers in a path through the networks from you to your friend.) Together, the network layer structured with destination address, source address, and

data is called a packet. The basic network layer protocol transformation is shown in Figure below.

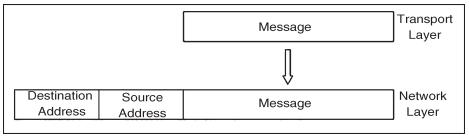


Fig. Network Layer Transformation.

The message must travel from your computer to your router. Every computer connected to a network has a network interface card (NIC) with a unique physical address, called a MAC address (for Media Access Control). At the data link level, two more headers are added, one for your computer's NIC address (the source MAC) and one for your router's NIC address. A data link layer structure with destination MAC, source MAC, and data is called a frame. Every NIC selects from the network those frames with its own address as a destination address. The data link layer adds the structure necessary for data to get from your computer to another computer (a router is just a dedicated computer) on your network.

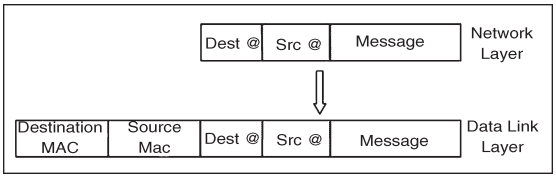


Fig. Data Link Layer Transformation.

Finally, the message is ready to be sent out as a string of bits. We noted earlier that analog transmissions communicate bits by using voltage or tone changes, and digital transmissions communicate them as discrete pulses. The

physics and electronics of how bits are actually sent are handled at the physical layer.

On the receiving (destination) side, this process is exercised in reverse: Analog or digital signals are converted to digital data. The NIC card receives frames destined for it. The recipient network layer checks that the packet is really addressed to it. Packets may not arrive in the order in which they were sent (because of network delays or differences in paths through the network), so the session layer may have to reorder packets. The presentation layer removes compression and sets the appearance appropriate for the destination computer. Finally, the application layer formats and delivers the data as an e-mail message to your friend.

The layering and coordinating are a lot of work, and each protocol layer does its own part. But the work is worth the effort because the different layers are what enable Outlook running on an IBM PC on an Ethernet network in Washington D.C. to communicate with a user running Eudora on an Apple computer via a dial-up connection in Prague. Moreover, the separation by layers helps the network staff troubleshoot when something goes awry.

Layering

Each layer reformats the transmissions and exchanges information with its peer layer. Let us summarize what each layer contributes. A typical message that has been acted upon by the seven layers in preparation for transmission. Layer 6

breaks the original message data into blocks. At the session layer (5), a session header is added to show the sender, the receiver, and some sequencing information. Layer 4 adds information concerning the logical connection between the sender and receiver.

The network layer (3) adds routing information and divides the message into units called packets, the standard units of communication in a network. The data link layer (2) adds both a header and a trailer to ensure correct sequencing of the message blocks and to detect and correct transmission errors. The individual bits of the message and the control information are transmitted on the physical medium by level 1. All additions to the message are checked and removed by the corresponding layer on the receiving side.

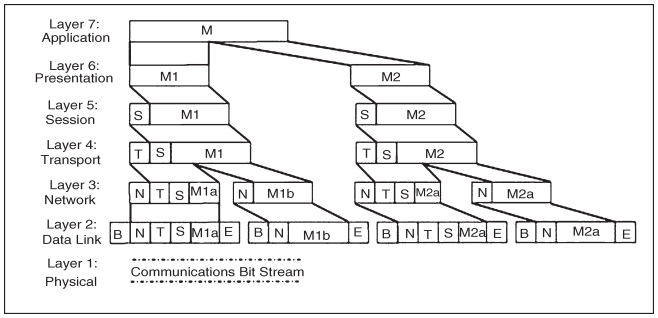


Fig. Message Prepared for Transmission.

The OSI model is one of several transmission models. Different network designers implement network activities in slightly different combinations, although there is always a clear delineation of responsibility. Some designers argue that the OSI model is overly complexit has too many levels and so other models are typically shorter.

TCP/IP

The OSI model is a conceptual one; it shows the different activities required for sending a communication. However, full implementation of a seven-layer transmission carries too much overhead for megabit-per-second communications; the OSI protocol slows things down to unacceptable levels.

For this reason, TCP/IP (Transmission Control Protocol/ Internet Protocol) is the protocol stack used for most wide area network communications. TCP/IP was invented for what became the Internet. TCP/IP is defined by protocols, not layers, but we can think of it in terms of four layers: application, host-to-host (end-to-end) transport, Internet, and physical. In particular, an application programme deals only with abstract data items meaningful to the application user. Although TCP/IP is often used as a single acronym, it really denotes two different protocols: TCP implements a connected communications session on top of the more basic IP transport protocol. In fact, a third protocol, UDP (user datagram protocol) is also an essential part of the suite.

The transport layer receives variable-length messages from the application layer; the transport layer breaks them down into units of manageable size, transferred in packets. The Internet layer transmits application layer packets in datagrams, passing them to different physical connections based on the data's destination (provided in an address accompanying the data). The physical layer consists of device

drivers to perform the actual bit-by-bit data communication. How each layer contributes to the complete interaction.

The TCP protocol must ensure the correct sequencing of packets as well as the integrity (correct transmission) of data within packets. The protocol will put out-of-sequence packets in proper order, call for retransmitting a missing packet, and obtain a fresh copy of a damaged packet. In this way, TCP hands a stream of correct data in proper order to the invoking application. But this service comes at a price. Recording and checking sequence numbers, verifying integrity checks, and requesting and waiting for retransmissions of faulty or missing packets take time and induce overhead. Most applications expect a flawless stream of bits, but some applications can tolerate a less accurate stream of data if speed or efficiency is critical.

A TCP packet is a data structure that includes a sequence number, an acknowledgment number for connecting the packets of a communication session, flags, and source and destination portnumbers. A port is a number designating a particular application running on a computer. For example, if Jose and Walter begin a communication, they establish a unique channel number by which their computers can route their respective packets to each of them. The channel number is called a port. Each service uses a well-known port, such as port 80 for HTTP (web pages), 23 for Telnet (remote terminal connection), 25 for SMTP (e-mail), or 161 for SNMP(network

management). More precisely, each of these services has a waiting process that monitors the specified port number and tries to perform its service on any data passed to the port.

The UDP protocol does not provide the error-checking and correcting features of TCP, but it is a much smaller, faster protocol. For instance, a UDP datagram adds 8 bytes for control information, whereas the more complex TCP packet adds at least 24 bytes.

Addressing

Scheme for communication to occur, the bits have to be directed to somewhere. All networks use an addressing scheme so that data can be directed to the expected recipient. Because it is the most common, we use the Internet addressing scheme known as IP addresses in our examples, since it is the addressing handled by the IP protocol.

All network models implement an addressing scheme. An address is a unique identifier for a single point in the network. For obvious reasons, addressing in shared, wide area networks follows established rules, while addressing in local area networks is less constrained.

Starting at the local area network, each node has a unique address, defined in hardware on the network connector device (such as a network interface card) or its software driver. A network administrator may choose network addresses to be easy to work with, such as 1001, 1002, 1003 for nodes on one LAN, and 2001, 2002, and so forth on another.

A host on a TCP/IP wide area network has a 32-bit address, called an IP address. An IP address is expressed as four 8-bit groups in decimal notation, separated by periods, such as 100.24.48.6. People prefer speaking in words or pseudowords, so network addresses are also known by domain names, such as ATT.COM or CAM.AC.UK. Addressing tables convert domain names to IP addresses.

The world's networks are running out of unique addresses. This 32-bit standard address is being increased to 128 bits in a scheme called IPv6. But because 32-bit addresses will remain for some time, we focus on the older version.

A domain name is parsed from right to left. The rightmost portion, such as .COM, .EDU, .NET, .ORG, or .GOV, or one of the two-letter country specific codes, such as .UK, .FR, .JP, or .DE, is called a top-level domain. A small set of organizations called the Internet Registrars controls these top-level domains; the registrars also control the registration of second-level domains, such as ATT in ATT.COM. Essentially, the registrars publish addresses of hosts that maintain tables of the second-level domains contained in the top-level domain. A host connected to the Internet queries one of these tables to find the numeric IP address of ATT in the .COM domain. AT&T, the company owning the ATT Internet site, must maintain its own host to resolve addresses within its own domain, such as MAIL.ATT.COM. You may find that the first time you try to resolve a fully qualified

domain name to its IP address, your system performs a lookup starting at the top; for subsequent attempts, your system maintains a cache of domain name records that lets it resolve addresses locally. Finally, a domain name is translated into a 32-bit, four-octet address, and that address is included in the IP packets destined for that address. (We return to name resolution later in this chapter because it can be used in network attacks.)

Routing Concepts

A host needs to know how to direct a packet from its own IP address. Each host knows to what other hosts it is directly connected, and hosts communicate their connections to their neighbours. For the example network of Figure above, System 1 would inform System 2 that it was one hop away from Clients A, B, and C. In turn, System 2 would inform its other neighbour, System 3, that it (System 2) was two hops away from Clients A, B, and C. From System 3, System 2 would learn that System 3 was one hop away from Clients D and E, Server F, and System 4, which System 2 would then pass to System 1 as being a distance of two hops. The routing protocols are actually more complex than this description, but the concepts are the same; hosts advertise to their neighbours to describe to which hosts (addresses) they can route traffic and at what cost (number of hops). Each host routes traffic to a neighbour that offers a path at the cheapest cost.

5

Network Applications

- Access to remote programmes.
- Access to remote databases.
- Value-added communication facilities.

Calling up a distant computer via a network is cheaper than calling it directly. The lower rate is possible because in a normal telephone call ties up an expensive, dedicated circuit for the duration of the call, whereas access via a network ties up long-distance lines only while data are actually being transmitted.

MOTIVATION

Sophisticated multi-party applications will use many traffic streams with very different characteristics and will be network-aware so they can perform well on a variety of networks. At the same time, we see the emergence of an electronic service industry that is eager to deliver a wide

variety of services to end-users. Services will range from low-level "bearer" services that transport bit streams over the network infrastructure to value-added services such as video conferencing, computing services, and data mining. Complex applications will support cooperation among multiple parties by combining video conferencing with access to large amounts of archived data, real time data streams, and distributed computing tasks. Supporting this service model and this emerging class of complex services requires innovation in a number of areas.

- First, the requirements on how the network should handle traffic streams will be very diverse, both in terms of the ability to share resources between cooperating traffic streams and the quality of service for individual streams.
- Second, conditions in the network and at the endpoints will change continuously, and mechanisms are needed that allow the network, services and application to adjust quickly.
- Third, in many cases, applications and services have advance knowledge of changes in resource requirements, and mechanisms are needed to make use of this information to optimize performance.
- Fourth, we have to develop systematic methods for balancing the constraints and priorities of services competing for network resources.

NETWORK MODEL

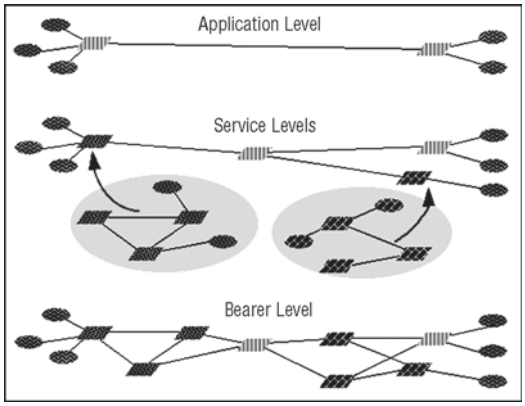
We view network entities as playing one of three roles, as is illustrated in the figure above:

- A bearer provides links and switching points that move bits among endpoints according to certain simple agreements about resource allocation.
- A service provider packages bearer functions and its own computing resources to provide services ranging from simple services such as CBR ATM connections to complex value added services such as multimedia conferencing services that adapt media rates and formats among heterogeneous endpoints, or distributed computing and storage capabilities. Services can be hierarchical, *i.e.* a complex service might be built on top of simpler services. We will distinguish between services that require handling the data and those that don't. The latter can be handled directly in the switching points, while the former might be implemented on service nodes. Service nodes are outside of the core network and are represented in our model by "virtual endpoints".
- An application runs on a set of computational endpoints and operates by invoking services in the network. Applications can be very simple, (*e.g.* dialing a number on a phone to get directory assistance) or

more complex (*e.g.* a distributed computing application that uses a collective communication service).

Let us illustrate the model using a video conferencing scenario. The application provides information on participants, connectivity, limits on simultaneous activity on the connections, etc. If the conference is dynamic, the information has to include a characterization of the dynamic behaviour. The video conferencing service will combine the application information with a requirements specification specific to video conferencing.

This results in a session specification that is used to contract for bearer resources and computation capability to satisfy the session requirements. The service mesh thus constructed provides an abstracted view of the underlying bearer resources, in which only those links and switching points at the periphery of the bearers' meshes are visible; this is illustrated in the figure below.



The service layer will also provide the bearer layer with information on how the traffic belonging to the new

application should be handled. These instructions will be translated into specific switch schedules, buffer allocations, etc. by the bearer.

Throughout the video conference resource allocation will be adjusted. These adjustments can be triggered by the application (*e.g.* a receiver changes the video source it wants to see), by the service layer (*e.g.* reallocating resources in the service mesh) or by the bearer layer (*e.g.* changes in available bandwidth). The adjustments will be very different: the simplest changes will require only local changes on one or a small number of nodes, while others, like rerouting, will require a global view. Ideally, many changes can be accommodated through local adjustments, since this will result in more responsive and efficient service.

A useful view of this functionality is provided by the "active network" concept under discussion among a group of ARPA-supported researchers. In this view, the network conveys objects that may carry methods to be invoked at different points, and may "programme" switching and processing entities in application-specific ways. Our research fits well into this model. The resource allocation capabilities we envision are needed to allow the network to host the customised computations envisioned for active networks. At the same time, active network mechanisms will be used in support of the customization of resource management to meet application needs. We call this type of network "application-

aware" since it directly cooperates with applications and services to maximize network responsiveness to application requirements. In the following section, we elucidate the research issues that we plan to pursue in developing the underpinnings of such an "application-aware" network.

ALLOCATING A VIRTUAL APPLICATION MESH

A virtual application mesh represents the resources allocated for the application, *e.g.* bandwidth on links, "capacity" on switch nodes, and virtual end-point resources. Virtual meshes differ from traditional routes in two fundamental ways. First, resources are associated with an application session, and not with individual connections. Second, while switches traditionally make resource allocation decisions fairly independently, switch nodes coordinate resource allocation both at startup and runtime. These features make it possible to optimize resource allocation using global goals and constraints (*e.g.* limit total resource utilization) instead of local ones (*e.g.* fairness on a per-link and per-connection basis).

The virtual application mesh also supports application awareness: the application has considerable flexibility over how resource inside the virtual mesh are allocated initially and reallocated, for example in response to changes in the conditions in the network or in the application requirements.

These changes must be handled quickly and efficiently. How the virtual application mesh is laid out will impact the cost of these adjustments, so the initial mesh allocation problem has a planning component. However, allocating a virtual application mesh is considerably more complex than to traditional point-to-point or point-to- multipoint routing. It requires allocating more diverse resources, and also the simultaneous allocation of resources for a large number of application streams.

SWITCH POINT MANAGEMENT

Service-related activities in networks take place on a wide range of time scales. However, resource allocation decisions are typically limited, *e.g.* per-packet scheduling inside the network combined with a transport protocol executing on the endpoint. Responsive application-aware networking will require coordinated resource allocation at many time scales.

Coordination can take the form of exchanging information regarding status and present and future requirements, coordination of resource allocation activities, and the specification (using a language or programme) of actions to be taken under certain conditions. These interactions will pay off in several areas. Examples include application- and service-specific dynamic sharing of resources between streams under changing network conditions; the implementation of services models that sit in between best effort and guaranteed services, for example using application

provided hints; application- and service-specific flow control; and the extension of the time horizon of schedulers, for example by considering application frame boundaries during scheduling.

Critical to the implementation of this network model is the ability to safely inject application and service state and code into the network. This enables application and service specific actions on switching points without the need for expensive and time consuming interactions with endpoints. Interactions between switching points and the application running on the endpoints will of course still be required, but they can be done in a service-specific instead of a generic fashion. The application and service "presence" in the network can range from simple parametres, through specifications using more complex languages, to actual code. The full spectrum of mechanisms will be needed in a fully application-oriented network. What mechanism will be used for a specific action will not only depend on what degree of flexibility is needed, but also on practical considerations. For example, including service code for cell-level switching is impractical.

RESOURCE MANAGEMENT FRAMEWORK

While traditional networks allocate resources on a per packet or per connection basis, resource allocation in application-aware networks is subject to constraints and goals of a variety of entities including link owners, service providers, and applications. We briefly describe a hierarchical

resource management framework that allows the systematic integration of these goals and constraints.

The resource allocation policy of a communication link is represented by a directed acyclic graph with a single root representing the link and leaf nodes representing individual traffic streams. Intermediate nodes represent organizational entities. Each node gets resources from its parents and specifies how its resources are distributed to its children. Examples of policies include fair-sharing at different granularities, reservation, and strict priority.

This graph is a language that can be used by different entities to specify how traffic streams or collections of traffic streams should share bandwidth. By combining subgraphs, the resource management policies set by different entities (link, service providers, applications) can be represented simultaneously. Tools are used to translate a graph into a schedule that can be used by switch nodes, and incremental changes in the graph translate into incremental changes in the schedule.

ROBUSTNESS AND SECURITY

Our approach lends itself naturally to dealing with failure recovery in an integrated fashion throughout the resource allocation process. One of the "changes" that can be considered during the creation of the virtual mesh is the failure of nodes and links, and both the topology of the virtual application mesh and the instructions to switching points

can implicitly and explicitly prepare for a quick response to failures. The degree of robustness can be application and service specific. The issue of security shows up in a number of areas. First, the network has to verify that application input (parametres, specification and programmes) can be acted upon safely since incorrect input can endanger the operation of the network. This requires a combination of language, compilation and runtime techniques, and we plan to use mostly existing technology for this security aspect. Second, the network has to guarantee that only authorized entities can modify the network state. We plan to address this using existing authentication methods. Note that the virtual application mesh can form the basis for providing security. As part of the creation of the mesh, relationships of trust can be established between the nodes in the mesh. This can speed up security checking during execution.

EXPERIMENTAL APPROACH

Carnegie Mellon University is developing a comprehensive suite of resource management mechanisms in support of such "application-aware" networks. We will support resource allocation along three dimensions: resource allocation in the "space" consisting of the physical network infrastructure and attached processing and storage resources; decision making on different time scales, ranging from application startup to packet and cell scheduling; and resource allocation by different organizational entities sharing the infrastructure.

In all three dimensions, the mechanisms we develop will provide for extensive tailoring to application requirements. The resource management techniques will be evaluated in a testbed driven by increasingly more aggressive applications and services. Tesbed development will take place in three steps. Initially, we will use the existing Credit Net ATM network for quick experiments that will guide the design of interfaces and protocols. A second step will be a local area version of our network architecture and resource management software. Finally, we plan to perform a wide area evaluation, hopefully in cooperation with groups working on related research topics.

6

The Developing Trend of Computer Network Management System

Computer network management system is now starting to enter the application layer. Traditionally, computer network management system mainly concerned about the various network equipment, which was in the network layer. Centered around equipment or equipment assemblage, it used SNMP to control and manage equipment. Web users have higher demand on network as well as network bandwidth. Some demands concern the transmission of time sensitive data, such as real-time audio and video, but some data are not time sensitive. Therefore, considering the limit current network bandwidth, it is imperative to change the previous practice of not differentiating service content, but to provide high quality service to all individual users according to the

service contents, so as to better utilize the resources of bandwidth. This is called QOS (Quality of Services). With this idea, network management starts to move the controlling force from network layer to application layer. R1MON2 has tried this way, which was an important change to network management system. In spite of all the versatile technologies used in network management system, as a result of standardization activities and the need for system interconnection.

Distributed Network Management

The key of distributed object is to solve the problem of cross-platform connection and interaction, and to realise distributed application system. The CORBA presented by OMG is quite an ideal platform. Distributed network management is to set up multi domain management processes. Domain management process takes charges of the objects in the domain, and at the same time, different domains coordinates and interacts with each other, so as to perform the management of global area network. Thus, not only is the load of central network management reduced, but time lag for transmission of information on network management is decreased as well, which makes management more effective. Distributed technologies mainly have two aspects: one utilizes CORBA, and the other mobile agent technology. In the near future, centralized distributed network management model can be used to realise the functions of centralization and data acquisition distribution.

Integrated Network Management

Integrated network management requires network management system provide the multitier management support. It can keep all the sub networks in perspective through one operating platform, understanding its operating businesses, identifying and eliminating failures. Thus, the multi interlinked networks management is fulfilled. With network management having become more and more important, many different network management systems have emerged, including those that manage SDH networks and IP networks. The networks managed by these systems interlinks with and interdependent on with each other. There are multi network management systems at the same time. They are independent, in charge of different parts of the network. There can even be a few network management systems with same contents existing concurrently. They come from different manufacturers, and manage their equipment respectively, which has greatly made network management more complex.

Businesses Monitoring

Traditionally, network management aims at network equipment, and could not directly reflect the impact of equipment failure on businesses. Up until now, some network products have realised the monitoring of processes. However, for some services, even though the services end, but the processes still exist. The monitoring of services cannot be

clearly shown. For customers, they are concerned more about the services they get, such as quantity and quality of programs. Therefore, the monitoring of services and businesses is the further goal of management.

Intelligent Management

It supports strategic management and network management system self diagnosis and self adjustment. Network management is the method of managing the tools that belong to a network and maintaining, administering all the systems that are connected in the network. For one to be able to efficiently manage a network that person should be a qualified network administrator and should have in depth knowledge of the functionalities of the network and different topologies of network. There are two aspects in any network, one is the logical aspect and the other is the physical level. The network administrator should be good at both the logical and the physical aspects to be able to troubleshoot efficiently.

Network Administration Functions

The main task of network administration is to keep the network running smoothly 24 hours a day. The main function is to monitor the network constantly and look for possible trouble, detect and rectify the trouble before the network gets affected.

The network administration part of the job involves the resources available on the network and their functioning.

The administrator is required to keep a track of all the resources available in the network and ensure they are functioning properly.

Network maintenance part of the job involves installing updates frequently, updating the service packs for the network software's and also applying patches for the routers when needed. The software and the hardware for the network must be constantly monitored for updates and maintained in order for the network to function properly.

The provisioning part of the job is accommodating extra resources on the network or upgrading the devices on the network. Suddenly if an entirely new device is introduced to the network, the network may stop functioning, so in order to avoid this the administrator needs to make extra provisions for possible devices at the beginning itself with precision.

Network Architecture

The network administrator needs to have a thorough understanding of the network architecture in order to be able to troubleshoot when there are problems. Most of the networks follow a similar architecture and function in the same way. The network architecture is designed in such a way that if the normal functioning is disrupted in any way then the network will send out an e-mail to the administrator, of any unwanted event, or shutdowns. Since the administrator is notified of the problem it gets taken care of immediately. The disaster recovery is also apart of network

planning and management. The network architecture itself plays a pro-active role in the network by aiding in the functioning of the network. The other crucial part of the network is the network protocol. Most networks use the Simple Network Management Protocol and some others use the Common Management Information Protocol.

Network Management History

By the time computers gained popularity, the demand for networks has also been increasing. Companies and people started needing systems which would work in an environment and still required the controls to be in one place. As and when more devices were introduced it became difficult to add one individual device for each computer. Companies needed an easier way of managing the resources. A Network was the perfect answer; however networks were not easy and needed advanced working technology.

When the network initially began it was difficult, but today networks have advanced through software and topologies and there are many efficient methods to handle them.

There are many kinds of networks today like cable networks, wireless networks, digital networks, Satellite connections and all these networks work on similar network topologies. Companies use a combination of these networks for their functionalities. Based on these networks internet and intra company networks have promoted business to a large extent.

FAULT MANAGEMENT-STATE OF THE ART

The terms *fault, error* and *failure* have often been confused. Incorrect interpretation of these terms may lead to their misuses. Definitions distinguishing them can be found in Wang's paper. A *fault* is a software or hardware defect in a system that disrupts communication or degrades performance. An *error* is the incorrect output of a system component. If a component presents an error, we say the component fails. This is a *component failure.*

Failure or component failure corresponds to the production of an error by this component. It is essential that we distinguish the terms. The *fault* is the direct or indirect cause of the errors. The *errors* are manifestations of the *fault.* The *failure* is the overall result of the *fault.* However, if a component produces errors, we cannot conclude that there is a fault within the component.

Network faults can be characterised by several aspects such as their symptoms, propagation (transmission of an error from a component to another), duration in the network and severity. Though network faults can be distinguished through their characteristics, it is worth noting that it is difficult to measure these properties accurately since they are subjected to the manner in which they are controlled and managed. The occurrence of a fault may be detected by users through symptoms that may be produced by some network components as a result of error. Fault symptoms

can be associated to four types of error-timing errors, timely errors, commission errors and omission errors.

These symptoms may take one of the following forms:

- *An output with an expected value comes either too early or too late*: This situation is due to a timing error. It is usually seen by users as a slow response or a time-out, when their applications are indirectly influenced by the effects of the faults.
- *An output with an unexpected value within the specified time interval*: This situation is due to a timely error which usually indicates a minor fault in the applications or underlying software and hardware.
- *An output with an unexpected value outside the specified time interval*: This is due to a commission error. If no response is produced, it is associated with an omission error. An omission error can be regarded as a special case of commission error. A commission error usually implies a severe fault has occurred in the network.

If these symptoms are observed, there is a possibility that a fault has occurred somewhere in the network resulting in the components to produce erroneous outputs (errors). A fault in one component may have consequences on other associated components. Besides failing its own system, it may produce errors which could be transmitted to other components and degrade other systems as well. In this way,

a fault in a single component may have global effects on the network. This phenomenon is referred to as fault propagation. A fault which occurs in an isolated systems may not affect other systems because there is no interaction between them. However, when the isolated systems are connected together by communications, errors produced by a fault may travel in a packet to other systems. A component can fail as a result of faults within it and the erroneous input produced by faults in other components.

This is one of the major characteristics of network faults. Media for fault propagation include parameter, data and traffic. The duration of a network fault, though recognised as one of its important criteria is somewhat difficult to measure.

This is due to three reasons. Firstly, a fault will not be perceived until it produces errors. Secondly, it may take a long time for a particular fault to be isolated. Finally, the effects of network faults may not be eliminated automatically when the faults are removed. They will remain for some time until the operation is completely restored. Therefore, network faults can only be generally divided according to their duration into three groups: permanent, intermittent and transient. A permanent fault will exist in the network until a repair action has been taken. This results in permanent maximum degradation of the service. An intermittent fault occurs in a discontinuous and a periodic manner. Its outcome

will be failures in current processes. This implies maximum degradation of the service level for a short period of time. A transient fault will momentarily cause a minor degradation of the service. Faults of the first type will cause an event report to be sent out and changes made in the network configuration to prohibit further utilisation of this resource.

For a fault of the second type, the severity of the fault may transfer from being intermittent to being permanent if an excessive occurrence of this kind of fault becomes significant. Finally, a transient fault will usually be masked by the error recovery procedures of network protocols and therefore may not be observed by the users. It is fundamental for a designer of a fault management system to have knowledge of fault characteristics. This is because not all faults will have the same priority. The fault management system designer will have to decide which faults must be managed.

FAULT MANAGEMENT PROCESS

Recent literature suggests that a comprehensive fault management system is composed of monitoring, reporting, logging, trouble ticketing, filtering, correlating, diagnosis and recovery activities. The domain in which the fault management system will operate. For the purpose, we have chosen to divide the activities associated with fault management into four major categories, namely detection, isolation, correction and administration. Error detection provides the capability to recognise faults. It consists of

monitoring and reporting activities. Information provided by monitoring devices must be current, timely, accurate, relevant and complete. Reporting activity include investigation of critical criteria which require notification and a mechanism for report generation. It also involves determining appropriate destination for sending notifications.

Its purpose is to isolate the actual fault, given a number of possible hypotheses of faults. Testing may be the most appropriate way of isolating the fault at this stage. Isolation comprises four activities: filtration, interpretation, correlation and diagnosis. Filtration involves analysing management information in order to identify new faults or if the fault has occurred before, to update its count.

Filtration discards management information notifications that are of no significance and routes applicable notifications to their appropriate destinations within the system. Important information contained in the event reports must be extracted. Here the nature, structure and significance of the event report are examined.

Important information such as the name of the event report which normally represents the predefined condition that was met and triggers the generation of the event report itself. Other useful information is the time when the event report was generated. This information is important when performing correlation activity so that we may distinguish which events are related, and which are not. Correlation

proves to be helpful, when two or more notifications received are actually due to a single fault. Through correlation, some faults may be indirectly detected.

Hypotheses can be drawn from alarm correlation giving possible causes of fault. The objective of the diagnosis process is to isolate the cause of a fault down to a network resource. Given a set of probable causes, the diagnosis process is carried out. It involves identification and analysis of problems by gathering, examining and testing the symptoms, information and facts.

Once isolated, the effect of the fault must be minimized through bypass and recovery, and permanent repair instituted. Where applicable, steps must be taken to ensure that problems do not recur. This procedure consists of three activities: reconfiguration, recovery and restoration.

Reconfiguration or bypassing involves activating redundant resources specifically assigned to backup critical entities, suspending services, or re-allocating resources to more important uses. The objective is to reduce the immediate impact of the failure. This function may be in a mixture of manual, semi-auto and automatic procedures. It may be possible for a fault to recover before any reconfiguration attempt is made. This depends on the nature of the failure, the criticality of the service and the expected time required to recover/reconfigure. Once a fault has been rectified, the repair needs to be tested and the entity returned to service.

This needs to be scheduled at an appropriate time and depends on the expected service disruption in doing so.

Fault administration service ensures that faults are not lost or neglected, but they are solved in a timeous fashion. This involves monitoring fault records, maintaining an archive of fault information, analysing trends, tracking costs, educating personnel and enforcing company policy with regards to problem resolution. It consists of three activities: logging, tracking and trend analysis. Logging maintains a log of event reports on faults that have occurred in the network. This will be used for trend analysis, reporting and future diagnosis of the same type of or similar failures. Tracking keeps track of existing problems and persons responsible for and/or working on each one, facilitates communication between problem solving entities and prevents duplicate problem solving efforts. This includes prioritisation of open faults due to their severity, and escalation of fault isolation or correction processes based on duration and severity of the faults. The current open fault records need to be ordered according to a priority scheme such that the most costly, or potentially costly failures are timeously resolved. The whole activities may be accomplished using a trouble ticket system. Important information includes the frequency of occurrence of a particular failure and how much down time the various users are experiencing. Trends may indicate a need to redesign areas of the communication

environment, replace inferior equipment, enhance problem solving expertise, acquire new problem solving tools, improve problem solving procedures, improve education, renegotiate service level agreements. In this project, all activities in fault detection and isolation procedures and some aspects in recovery and administration procedures are implemented.

Typical Problems

One of the most critical problem associated with fault monitoring as given by Dupuy and Stallings is u*nobservable faults*. In this situation, certain faults are inherently unobservable through local observation. For example, the existence of a deadlock between co-operating distributed processes may not be observable locally. Other faults may not be observable because the vendor equipment is not instrumented to record the occurrence of a fault.

Other problems are defined by Fried and Tjong as follows. *Too many related observation:* A single failure can affect many active communication paths. The failure of a WAN back-bone will affect all active communication between the token-ring stations and stations on the Ethernet LANs, as well as voice communication between the PBXs. Furthermore, a failure in one layer of the communications architecture can cause degradation or failures in all the dependent higher layers. This kind of failure is an example of propagation of failures. Because a single failure may generate many secondary failures, they may occur around the same time and may often

obscure the single underlying problem. *Absence of automated testing tools:* Testing to isolate faults is difficult, expensive and time consuming. It requires significant expertise in device behaviours and tools to pursue testing. Even such a simple task as tracing the progressions of packets along a virtual circuit is typically impossible to accomplish. This leads to empirical rules of operation as "the only way to test if a virtual circuit is up, is to take it down".

The process of recovery typically involves a combination of automatic local resetting combined with manual activation of recovery procedures. Recovery presents a number of interesting technical challenges. Which include a*utomatic recovery as a source of fault:* Since most network devices or processes are designed to recover automatically from local failures, this can also be a source of faults. The problem in fault administration is the maintenance of fault reports log which has been made difficult due to the lack of functionalities for creating or deleting records. The logging facility is usually performed in a static manner, where logging characteristics are stagnant. Therefore, some important fault occurrences are unable to be recorded. If log attribute values can be changed, the logging behaviour may also be altered.

FMS: A FAULT MANAGEMENT SYSTEM BASED ON THE OSI STANDARDS

In order to overcome the problems, we have designed and implemented FMS. It offers three types of fault management

applications as depicted namely the Fault Maintenance Application (FMA), the Log Maintenance Application (LMA), and the Diagnostic Test

Application (DTA). These fault management applications work together with the OSI Agent to perform fault management tasks on the network resources. The OSI Agent serves as a fault-monitoring agent. It has the capability to independently report errors to FMA. It can also issue a report when a monitored variable crosses a threshold. This allows the FMS to anticipate faults.

In addition, it maintains a log of events. These logs can be accessed and manipulated by LMA. DTA provides a set of diagnostic tests which may be invoked by the user. This facility is beneficial to the network administrator whenever there is any suspicion that some of the resources are not functioning as desired.

The paragraphs that follow explain how these facilities help in increasing fault management efficiency. The FMA solves the problems of unobservable fault due to ill-equipped vendor equipment to record the occurrence of a fault. This is done by monitoring the real resource critical properties and when significant events involving these properties occur, these events are reported to FMA so that further investigation is initiated. FMA acts upon the receipt of these events by performing other fault management processes on them. These processes include event interpretation and filtration,

event correlation, invocation of predefined diagnostic tests and initiating recovery process. Event or alarm correlation is used by the FMA to solve the problems of too many related observation. In addition, the reporting criteria is designed to be reasonably tight to reduce the volume of alarms received by the FMA. In accomplishing this objective, only events that require attention are reported. Thus, in the FMS implementation, event reports are equal to alarms. Nevertheless, the objective to anticipate failure is neither neglected nor sacrificed. Hence, a number of managed objects are designed to issue event reports when the monitored attributes cross thresholds. Automated testing is provided in FMA by scheduling the execution of predefined diagnostic tests for every event report that is received, so that the source of failure becomes apparent.

Subsequently, recovery action pertaining to the diagnosis is carried out automatically. On the other hand, the DTA provides a function that allows diagnostic tests to be invoked by the user. This facility proves to be beneficial to the experienced user who wants to skip trivial tests and choose only specific ones.

This results in lower consumption of the network resources for the purpose of management. The lack of adequate tools for systematic auditing is overcome by the supports provided by the LMA. The LMA has the capability to initiate error condition (events) logging. Furthermore, the LMA can access

these logs and control logging behaviour by setting their log attribute values. Other supports include facilities for deleting logs and log records, and reviewing events (by reviewing log records) for diagnostic purpose or trend analysis.