NETWORK COMPUTING ONLINE The Technology Solutions Ce

Interactive Network Design Manual

Designing-And Redesigning-Today's Local Area Network

by Art Wittmann

Introduction

You've built a network and it runs reliably! Congratulations; you're doing better than many. But you know - deep in the pit of your stomach - that sooner or later it's going to be time to rebuild your network.

Reasons to Rebuild Your Network

There are three big reasons to build a new network and a couple of lesser ones. The three biggies are:

- A substantial increase in the number of users on your network.

- A substantial change in the power of the average user's workstation

- New applications emerge that demand more or different network services.

Lesser reasons include:

- Making the network more manageable for changes, moves and adds.

- Adding redundancy and improving reliability of the network

- Updating out-of-date equipment

Note that the big reasons are very much user driven, while the lesser reasons are mostly driven by the needs of the network manager - that's you. It isn't that your needs are inconsequential, however, satisfied users are what turns the budget wheels, so they get first consideration.

Design Philosophy: Switch When You Can, Route When You Must

Until as recently as three years ago, routers where the only game in town for adding bandwidth to networks. Some larger networks have been built with transparent bridges, however, these networks usually proved to be difficult to scale and manage.

Switching hubs have become much more popular over the last few years and now offer the features necessary to build a large, reliable high-performance network. Switching hubs initially where nothing more than multiport bridges, offering little more than bandwidth. Now, with virtual LANs and some layer-three protocol processing, switching hubs can be used to safely build economical high-performance networks.

In this article we will take a fairly progressive view of routers and switches. Our philosophy throughout will be to switch where you can and route where you must. Some vendor has probably already coined the phrase, but it is a good catch-all for the advice that we will provide throughout this document.

Our reasoning here is simple:

1. Routers are software-driven devices that excel in flexibility and feature sets. Generally, much, if not all of the routing decisions are determined by algorithms run on general purpose RISC CPUs. Because of this, routers are:

a. Expensive on a per-port basis. CPUs and memory cost a lot of money and router vendors extract heavy margins to support their ongoing software development efforts.

b. Routers are not particularly fast. CPU algorithms take time to run and, given the chance, you and I usually load up routers with all kinds of rules and control lists that must be checked on a per-packet basis, slowing the router even more.

c. Routers are a great way to get from a trusted network segment to an untrusted segment. Generally traffic between such segments (say, between an engineering department and a marketing department) is orders of magnitude less than within a department. Also, for all of the reasons we gave for routers being slow above, they also make great firewalls.

Switches, on the other hand, are firmware-driven devices. Virtually all of what they do has been committed to silicon in the form of Application Specific Integrated Circuits (ASICs). Custom ASICs can provide lightening fast algorithmic processing, but they allow for fairly little flexibility in the algorithm run. As a result switches are:

a. Simple. They take in packets, find a path for them and spit them back out another port. Network managers can't set up a large number of parameters on a per-switch basis,

b. Cheap. Particularly Ethernet switches have been reduced down to a few chips and usually only one major chip per port. They are beginning to rival the price of non-switched intelligent hubs.

c. Effective. Because they are simple devices, they can deliver exactly what you want - bandwidth to users. New technology like virtual LANs (VLANs) and the ability to deliver VLAN traffic outside of the box make the manageable on most all networks.

Another assumption we'll make is that you need more performance and flexibility out of your network. We'll also look closely at some management issues. These are three key areas that matter to you as a network admin. and that vendors use to distinguish themselves.

Table of Contents

Current Common Architectures

While there are probably hundreds of network topologies that have been used over recent years, we'll boil them down to three. By clicking on the type of network closest to what you have now or plan to build, you can get to the appropriate discussion of network design. It is probably wise, however, to read the discussions for all three, because some of the concepts developed in detail for the smaller networks also apply to the larger ones.

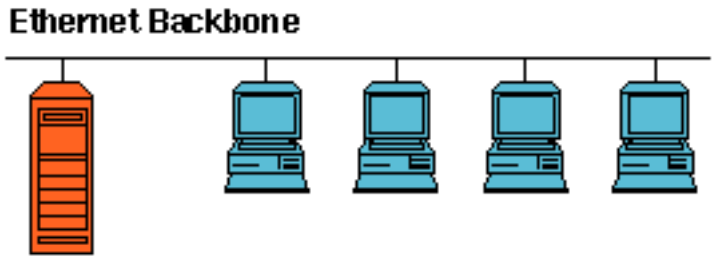
Small, single segment networks. Usually 100 or fewer users. The common topologies are Ethernet and Token-Ring. The network usually has no more than two or three servers.

Medium-sized, collapsed backbone networks. Usually 1,000 or fewer users. Common topologies are Ethernet and Token-Ring. A single router, or just a couple routers, sit as the backbone of the network and provide connectivity for the network. The network usually has ten or so servers.

Large, high-speed connected networks. Usually more than 1,000 users and often involving more than one building or a number of floors of a large office building. Desktop connectivity is still with Ethernet and Token-Ring, backbone is most often FDDI. Routers sit on the high-speed backbone and provide connectivity to attached Ethernet and Token-Ring segments.

The Small LAN

Figure One:



A Small Ethernet LAN

The first step to consider is whether most of the traffic on your LAN is directed at only a few machines on the network. In other words, is your network's traffic mostly client/server? If so, it makes sense to get the servers onto a bigger pipe than the workstations. If most of your traffic is peer-to-peer, then high-speed server links won't help that much.

So how do you tell? If you don't have a network analyzer, figuring out where your traffic is going can be a trick. Almost all small networks today are client/server networks. If your servers store user files, applications or both, then it is likely that most of the traffic on your network is from your servers. However, if you are using Windows for Workgroups, or some other peer-to-peer NOS for file sharing, then the traffic on you network will be much more peer to peer (some other operating systems might be: Unix, MacOS 7 or higher, LANtastic or Windows95).

The Peer-to-Peer Network

In the peer-to-peer network, you need to make sure that your network really is the bottleneck. It takes some kind of analyzer to tell, but it's probably worth either getting a basic software-only analyzer or renting a hardware analyzer for a day or two. On peer-to-peer networks, it is usually the case that most files of interest reside locally on each machine on the network. The result is that network traffic tends to be light. If, on the other hand, you've got some bandwidth intensive peer-to-peer application, like video conferencing, you may indeed need more bandwidth in your network.

Peer-to-peer Ethernet and Token-Ring switches can be found, however, they are becoming more rare. The upside to these switches is that they are fairly cheap and tend to operate with very low latencies. They often don't offer sophisticated management options and don't support enhancements like virtual LANs. These switches are usually fixed-configuration devices and support anywhere from six to 25 ports per switch.

Buffer Architecture

Depending upon the network architecture that you adopt, the internal architecture of the switch may make a

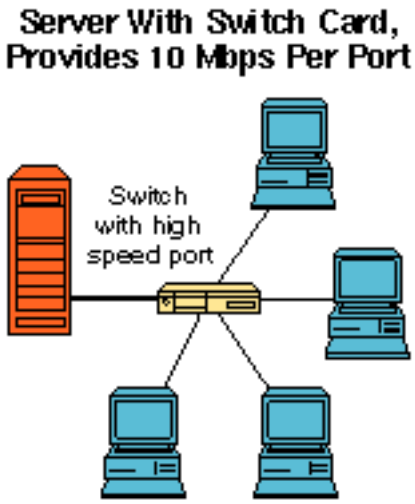
difference to you. If you plan to connect just one station per switch port, the internal buffer architecture of the switching isn't all that critical. You'll want to make certain that each port can buffer a good number of packets. Something on the order of 64 KB per port is in order.

However, if you will connect a number of stations per port, the buffer architecture itself can make a difference. A shared memory architecture, at least on switching hubs with 25 or fewer ports, can allow the switch to allocate more buffers to busy ports. Some hubs also employ mechanisms for throttling back stations that are sending too much data. On an Ethernet switch, collisions are simulated. On Token-Ring switches, the token may held for as long as possible.

The goal is not to drop packets. Dropped packets will be very noticeable to end users as some protocols can take as long as two seconds to time out before retransmitting a packet. Significantly less than one percent of packets should be dropped, otherwise you'll get complaints from end users.

Small Client-Server Networks

Figure Two:



A small, switched, client/server LAN

Possibly some of the best products on the market are intended for small client/server-oriented networks or work groups. These switches have one or two high-speed ports - typically FDDI, ATM or Fast Ethernet and anywhere from eight to 100 ports. To get to the 100-port level, you'll have to go with a slotted hub. Fixed-configuration units usually top out at about 25 ports. Prices here can be extremely competitive and prices well below $500 per port should be expected. In fact, some switching hubs may be priced lower than $200 per port making them competitive with high end non-switching hubs.

The choice between FDDI, ATM and Fast Ethernet can be a tough one. FDDI is the oldest and most proven technology. It is also expensive and somewhat tricky to implement in a switching hub. The problem is that FDDI and Ethernet have a different maximum transmission unit (MTU). In fact, FDDI's MTU is 4,500 bytes or three times that of Ethernet at 1,514 bytes. There are various methods for negotiating the proper packet size to use between stations and just the fact that this negotiation must take place makes FDDI's use more complicated.

TCP/IP is particularly tricky since the protocol relies on the device that connects an FDDI and Ethernet network to fragment packets down to their proper size. IP fragmentation is complicated and it implies that the switch must know something about TCP/IP. Building knowledge of higher level protocols into the switch just makes

it more expensive. On the other hand, it is well known that FDDI is an extremely efficient transport, easily delivering up to 98% of its rated bandwidth.

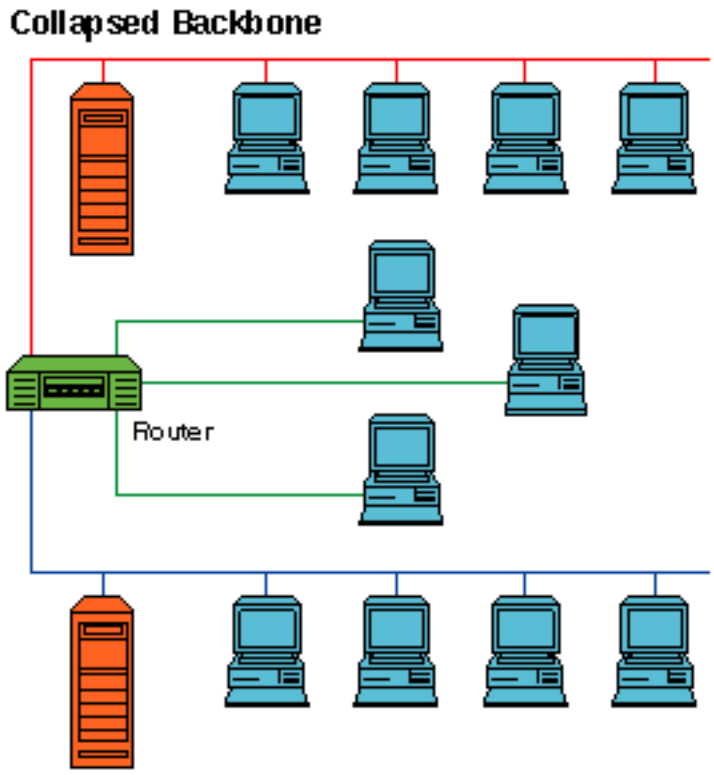
Fast Ethernet is the new kid on the block, but the case for its use is compelling. The technology has become cheap to implement and it doesn't suffer from the MTU problems that make FDDI expensive to implement. Also compelling is the fact that Fast Ethernet can fairly easily be implemented as a full duplex technology. Don't expect to see 200 Mbps flowing across your fast Ethernet pipe, that just isn't the way client/server works. However, it is reasonable to expect a total throughput of more than 100 Mbps.

Fast Ethernet can deliver efficient bandwidth, and by using fiber optics, it can deliver it to distances up to 1.24 miles [two kilometers], just like FDDI. Fast Ethernet does not have FDDI's built in fault tolerance or link management features, however, these are probably not that critical for smaller networks. So, in terms of performance and price, Fast Ethernet is hard to beat, and Fast Ethernet to Ethernet switches are most often the best way to introduce additional bandwidth into a small network.

ATM is the up-and-coming technology, but the question is whether it is appropriate for small networks. In general, ATM is probably not a good choice for a small network. It is expensive and really excels as a backbone technology for a large internetwork. It is more costly than FDDI and far most costly than Fast Ethernet. It is also a work in progress. Part of the reason that the cost of ATM remains high is that all of the standards needed to fully realize ATM's potential aren't yet finished. Companies can not yet dedicate silicon to ATM and those that do risk producing chips and hardware that may not conform with all standards.

Medium-Sized Networks and The Collapsed Backbone

Figure Three:



A Collapsed-Backbone Design

If you're like most in this segment of the market (100 to 1,000 users), you bought a couple routers five years

ago or so and have been adding to them and upgrading them all along. They work and your network is fairly reliable because of them. So the question is, should you continue or should you rearchitect?

The Case For The Status Quo

As it turns out, even if you need more ports and more performance, you can get it out of today's routers. Router vendors have felt the heat from switch vendors and while they aren't selling their hardware at bargain basement prices, they are offering substantially better hardware at about the same price as the previous generation of hardware. Of course you still have to fork over that substantial chunk of change for a new router - but at least it will not be substantially more than you've been paying.

Depending on the vendor, the hardware you buy will be third or fourth generation of the product. That means a couple of things - first, that the router will probably be a stable product and will probably deliver two to five times the performance and port density of the previous generation. These products tend to be evolutionary rather than revolutionary.

While two to five times the performance is nothing to be taken lightly, it may or may not get you by. It really depends on what is planned for your users at the desktop. If you are anticipating lots of high-end systems at the desktop and the possibility of multimedia applications, a simple router upgrade may not do the trick.

On the other hand, if your users are unlikely to be running multimedia applications or commonly making huge file transfers a simple upgrade may be just the ticket. After all, you already know how to manage it and you probably have found ways to allocate addresses and balance usage. There's a lot to be said for sticking with the system you know.

When Revolution Is Required

Many networks in this category are ready for a little revolution. Powerful desktop computers and servers that can deliver new applications change the performance requirements of networks in a way that make routers inappropriate. Further, a simple router upgrade does nothing to improve the manageability or flexibility of your network.

Routers, by their nature, associate network segments with individual ports on the router. Switches offer the ability to assign ports to virtual LANs (VLANs) under software control. Just fire up your switch management software and move ports to whichever virtual LAN is appropriate. Some vendors have taken this a step further, associating a user's physical network address with a virtual LAN. This allows notebook computer users to plug in to the network anywhere and automatically be configured on their virtual LAN. It's a pretty powerful idea, the only drawback being that each switch port can only accommodate one station at a time. With switch port prices coming down as low as $200 per port, this may not be out of the question.

What Happens In a Virtual LAN

The concept of a virtual LAN is straight forward enough, but what really happens to network traffic? After all, in a switched environment, traffic should only be seen by the source and destination nodes (or segments) anyway.

The exception here is for broadcasts and multicasts. They must be seen by all stations on the virtual LAN. In fact, a virtual LAN might just as well be called a broadcast zone. On some networks, broadcasts may be a big enough problem that just breaking the network into broadcast zones would be enough.

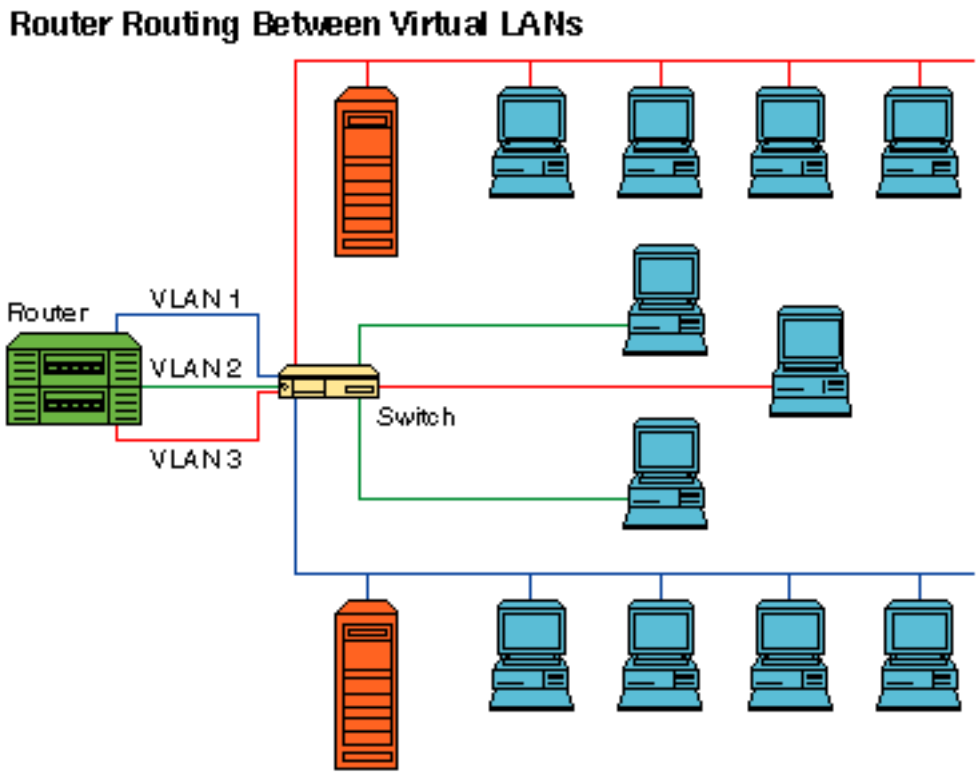
Virtual LANs usually go one step further though. It is usually not possible to move traffic from a station on one VLAN to a station on a different VLAN. In this way, the VLAN is like a fire wall requiring some sort of routing technology to move traffic between the VLANs.

It may seem silly to force traffic between stations that could be on the same switch to go through a router

somewhere at the back of the network. That fact hasn't escaped the bright folks that write ATM specifications in particular. If you've heard of MultiProtocol over ATM or MPOA, it addresses exactly this problem.

Switching hubs with VLAN support can go anywhere in the network. The approach we favor is to use a switching hub at the back of your network in conjunction with a router to route between VLANs. (Shown in figure below) This approach will improve network performance and give you the flexibility to assign VLANs within software. No more going to the wiring closet to move users between networks.

Figure Four:



A Router with Virtual LANs

The down side to this approach is that you will probably have to reassign network addresses to properly build the new network. This generally isn't that big of a problem with AppleTalk and NetWare/IPX, however, it can be quite an undertaking if you need to do it with TCP/IP. If you have to reassign IP numbers, do yourself a favor and start assigning them from a central server either by reverse ARP, BOOTP or DHCP. If you need to move users between VLANs, changing their IP number will then be as easy as editing an address table. That's a lot better than having to visit the users' office.

In order to get the benefit of VLANs, you really need to reduce the number of segments within your network. A new VLAN should only be needed if there is a good administrative reason to segregate traffic. For example, you may want a VLAN for divisions within your corporation (for example, one VLAN each for Engineering, Marketing, Finance, HR and so on). Again, the biggest impact is likely to be on your IP number scheme as you may have to change the subnet mask that you use in order to accomplish this.

Changing the IP subnet mask isn't a problem either unless you have control over only part of your IP network. Then it can cause real problems as getting differing IP subnet addresses to work can be a problem. In this case, it may be necessary to a VLAN to support a long subnet mask.

Servers On Their Own Ports

Regardless of how you reconstruct your network, give serious thought to putting the server on its own port and preferably its own high-speed port. It just makes sense that if you have many clients on their own ports that the servers should be off on their own higher-speed ports. Fast Ethernet is a great, low cost way to connect servers into the network. Many switches have FDDI ports, and there is nothing wrong with FDDI (see FDDI discussion in Small Networks segment). However, any advantage to FDDI is negated by its price tag. Individual switched FDDI ports generally run in the $4,000 to $10,000 range.

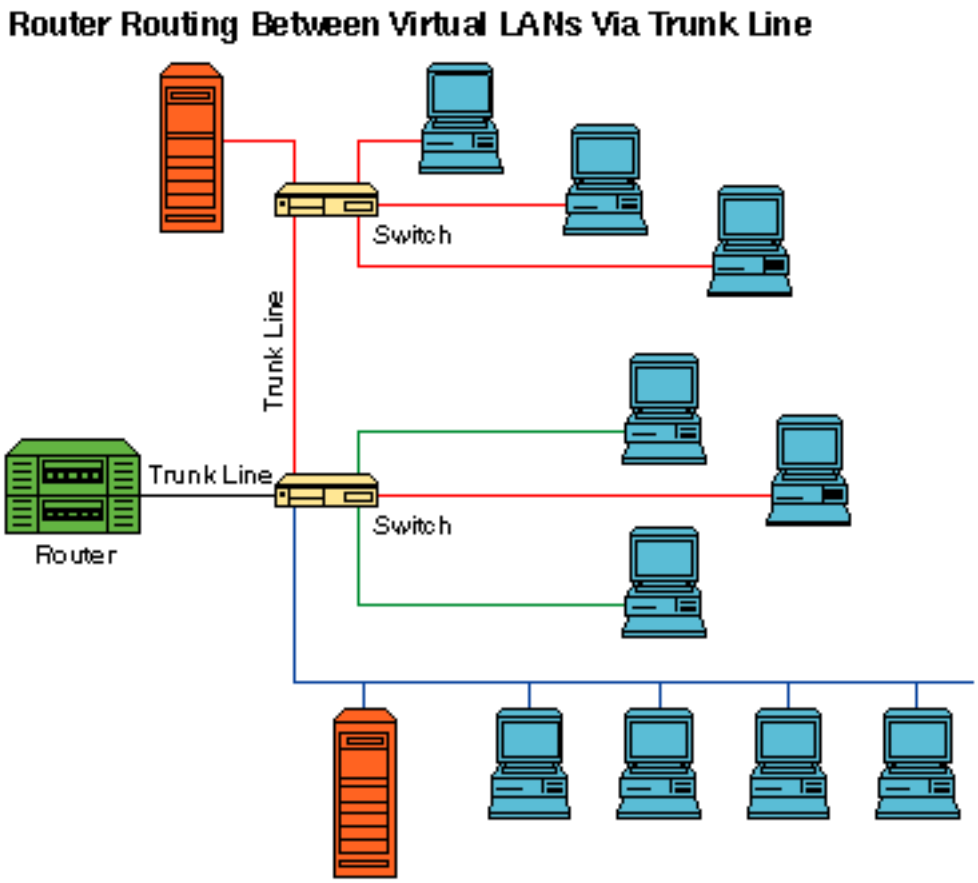
The exception to the FDDI rule, of course, is for networks that already use FDDI. This is usually pretty rare or fairly limited in collapsed backbone networks, so it may be justified to replace existing FDDI or buy one FDDI port to which existing FDDI gear can be attached.

Trunk Lines

The diagram above shows a router connected to each VLAN by standard Ethernet or Token-Ring connections. Your existing router can be used for this and if you create just a few large VLANs, your existing router should be able to accommodate the traffic between VLANs. This is a great way to leverage your existing investment in your router and if you have WAN connections through it, you'll maintain that connectivity.

It may be difficult to connect all of your VLANs back to your router, particularly if you decide to use more than one switch in your network. Although methods are currently proprietary, some vendors are beginning to offer VLAN trunking (running many VLANs over one connection). The diagram below shows this.

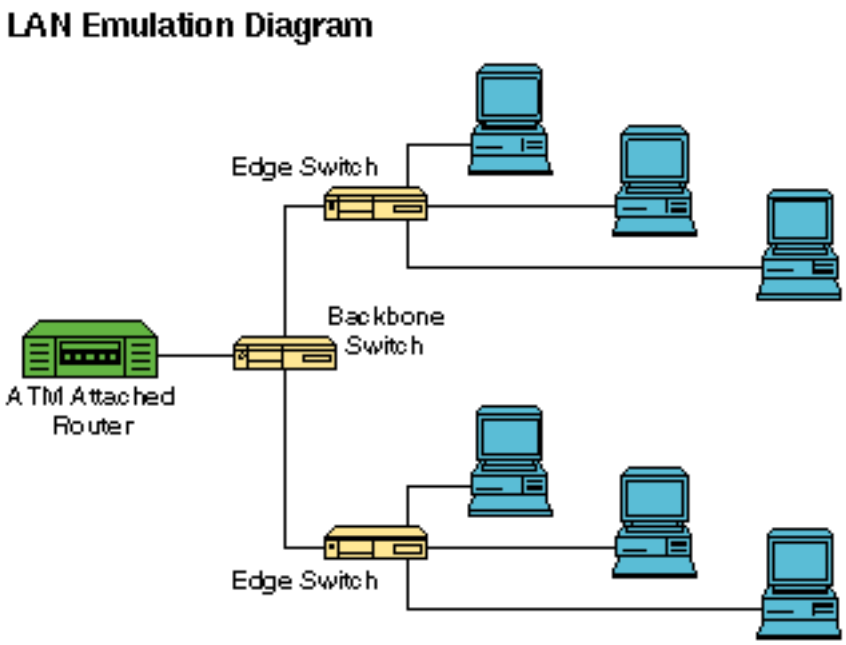
Figure Five:



Virtual LANs Routed Via Trunk Lines

ATM and LAN Emulation

Figure Six:



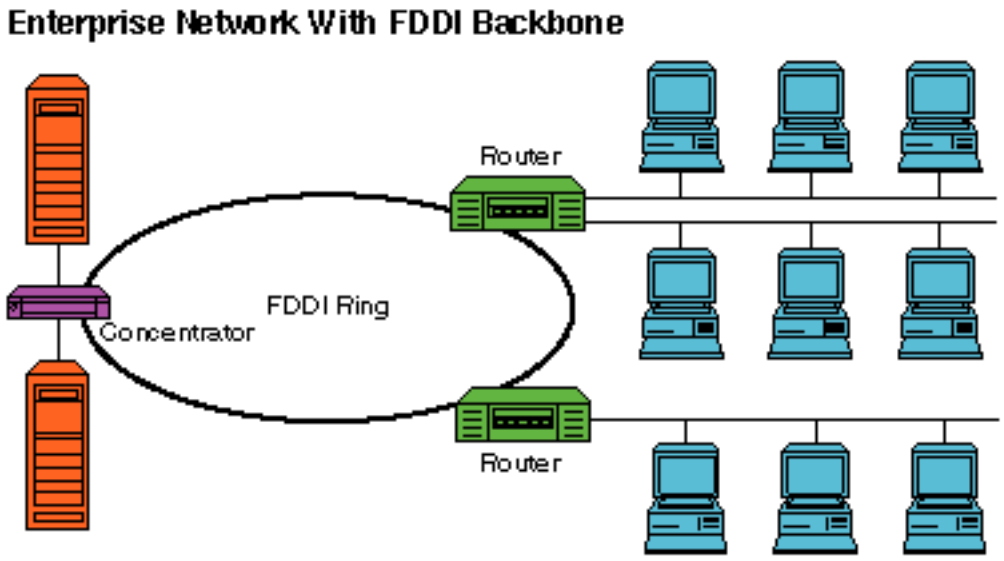
ATM and LAN Emulation

About the only nonproprietary way to get trunking is to go to ATM and use LAN Emulation to develop your virtual LANs. Using LAN Emulation and Ethernet to ATM edge switches an existing network can handle trunking and all of the VLAN concepts that we have talked about so far in a nonproprietary way. A warning is in order here: LAN Emulation is only now being tested for interoperability and it is almost assured that a single vendor solution will still work better than a mixed solution, but at least there is hope in the short term. It will be a while before non-ATM methods for carrying VLAN information are standardized.

The other significant advantage to use ATM and LAN Emulation is that servers can easily reside on more than one LAN. The virtual circuit mechanism that ATM uses to set up data paths can easily be used to get a server onto a number of LANs. At least today, there is no way to get servers onto multiple proprietary VLANs short of using multiple interface cards.

Big Networks

Figure Seven:



Large LAN with FDDI Backbone

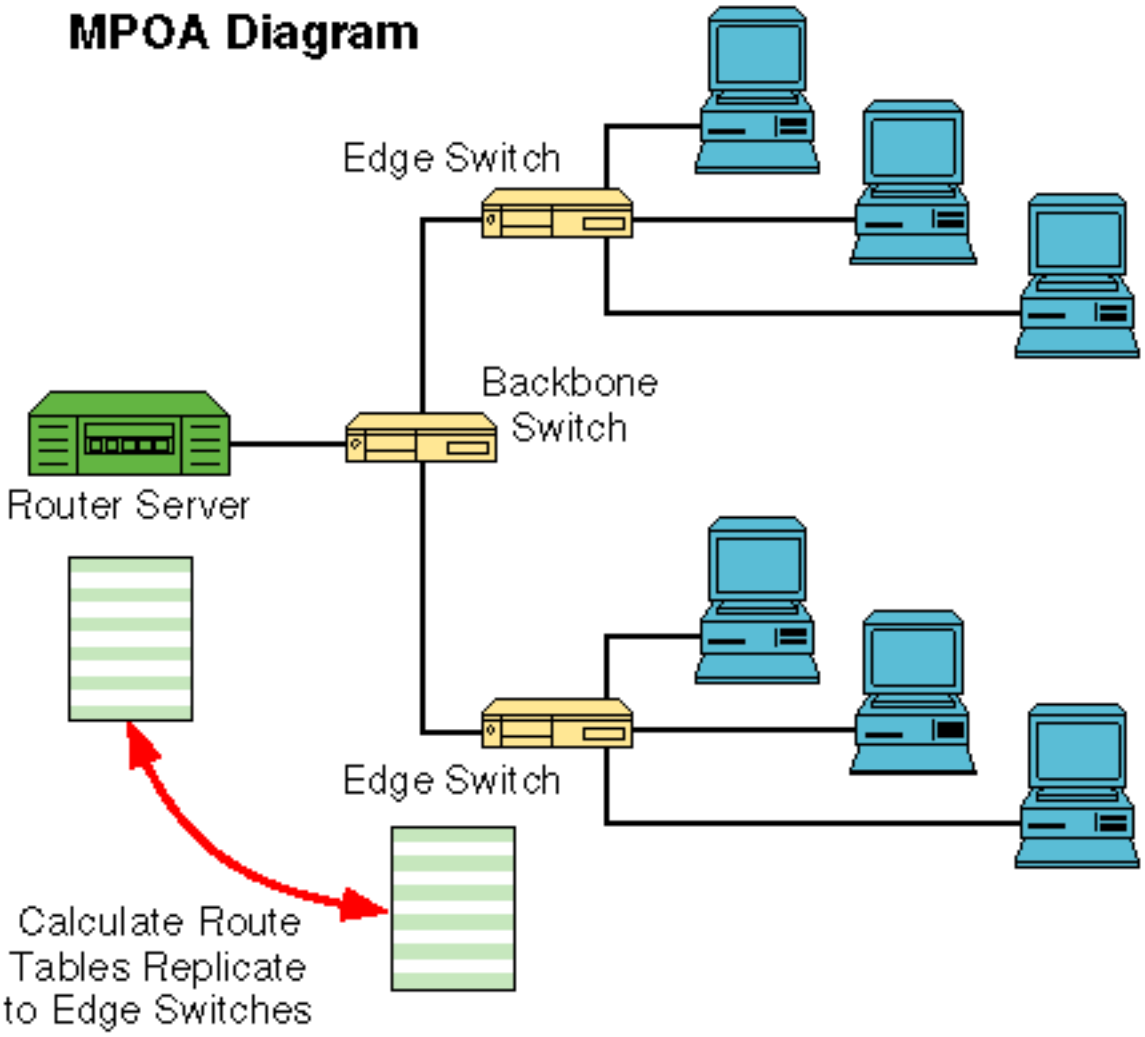
The first thing to do on large networks is to find the bottlenecks. There are two common bottlenecks on large internetworks. The backbone itself may be a bottleneck. It takes a pretty busy network to max out an FDDI ring, but it does happen and with ever-increasing regularity. The other common bottlenecks are older routers on the network. Especially if you have added a lot of rules and filter to your network.

If you've already invested in some sort of high-speed backbone, it's probably FDDI and you've probably sunk a lot of money into it. Dumping that investment and going to another architecture is a tough call. A number of vendors have come out with FDDI switching solutions that allow you to maintain your current investment in FDDI. The problem is that these solutions are expensive ($4,000 to $10,000 per port) and they typically don't offer you any more flexibility in managing and changing your network.

As a quick fix, there is nothing wrong with using FDDI switching, however ATM is now becoming a reasonable solution for adding flexibility and bandwidth into a network. The discussion above with regard to LAN Emulation and ATM applies equally well to large internetworks. The market is now seeing a number of switches boasting total bandwidth of 10 Gbps and more.

These switches are made specifically for the large internetwork. The concern, of course, is the scalability of the LAN Emulation. The pipe between a backbone switch and a router is typically only 155 Mbps. This may not be sufficient for very large networks. Further, if it takes a number of hops to get to the router at the back of the network, overall latency may become a problem. The industry realizes this and has proposed MPOA (MultiProtocol Over ATM).

While we don't propose that you attempt to buy and implement MPOA today, it is an interesting concept. The idea is to get routing tables to the edge switches in the network and to let them make routing decisions based on



Multiprotocol Over ATM

The route server at the back of the network would use link state protocols to develop a map of the network and then apply any rules and access controls to build the routing tables that the switches would use to make routing decisions.

Don't look for usable MPOA solutions before 1997, but keep the model in mind. It is the most flexible and expandable model to be envisioned for ATM that still uses current connectionless protocols.

Table of Contents

Appendix: Concepts and Definitions

Definitions: Network Components

the extended segment should be seen by all stations on an extended segment.

Network - The term itself has come to be rather ambiguous, referring to a segment, extended segment or internetwork. We often call any of these "The Network."

Internetwork - A set of segments or extended segments joined together by a router.

Unicast Packet - A data packet addressed to a single station. An example might be data from a client to a server.

Multicast Packet - A data packet addressed to a group of stations. The destination address is formed in such a way that stations realize that the packet may be destined for many other stations.

Broadcast Packet - A data packet addressed to any and all stations on the local segment. Broadcasts are often used by stations who have just joined the network - broadcasts are made to find out information about the segment that has just been joined.

Repeater - A device that facilitates connecting stations onto the segment. It does not understand network addresses - it merely copies data bit by bit from and to the physical media to which it is attached. On Token-Ring segments, this device is often called a Media Access Unit or MAU. A repeater is not considered an intelligent device.

Bridge - A bridge is used to connect two or more similar segments together (for example, Token-Ring to Token-Ring or Ethernet to Ethernet). A bridge has two purposes. The first is to extend the length and number of stations that a segment can support. Secondly, the bridge reduces overall traffic flow by only passing data packets that are not destined for a hardware address on a local segment. All broadcast and multicast traffic must cross a bridge - since no true destination can be known. In recent years, bridging technology has been used between dissimilar media (for example, Ethernet to FDDI), this sometimes may cause problems as we will see later. A bridge is considered an intelligent device. (See also bridging.)

Router - Sometimes called a gateway, it is used to connect two or more (potentially extended) segments. The segments may be similar or dissimilar. Routing information beyond the hardware address must be contained within the data packet. Virtually no broadcasts or multicasts are ever propagated across a router since no exact destination information is typically contained within these packets. Hardware addresses have only local significance to a router - higher level routing information is globally significant. (See also routing.)

Concepts: Shared-medium Networks

Sharing data on a network means multiplexing it on the basis of either frequency or time, and arranging for some sharing, or contention, scheme.

Multiplexing

Frequency Division Multiplexing (broadband)

Some of the earliest data networks devised used Frequency Division Multiplexing or FDM. These networks are also known as broadband networks. Just as we divide up the radio spectrum into channels, so can we divide up the spectrum over a cable. A possible example would be to use, say, 200 MHz of bandwidth and divide it into 20 channels of 10 MHz each. Each 10-MHz channel could, theoretically, be used to transmit up to 10 Mbps of data.

Apart from the obvious advantage of providing more bandwidth, the analog techniques used can also permit signals to be transmitted over fairly long distances on standard cabling - sometimes up to ten times further or more than straight digital signals on the same cable. So broadband systems can be used to provide very high data rates over fairly long distances.

The advantages of broadband seem significant. However, it is seldom used for local area networks today. In fact, other than digital systems that piggy-backed on private CATV systems, broadband is almost strictly the domain of telephone carriers. There are good reasons for this. First, analog broadband networks are expensive to build and maintain. They must be tuned and retuned as the network is extended. The equipment required to do this is expensive and the expertise is rare. Equipment that is attached to a broadband network is also expensive as it must have a digital-to-analog modem and transmitter - much like any radio system would have.

The real nail in the coffin for broadband systems were the time of their introduction. Indeed, a good broadband network could accommodate 500 Mbps or more data traffic and do it over a few square miles. The problem was that in the mid-70's to early 80's no one had computers that use 500 Mbps of bandwidth and we had barely begun to build local area networks. Few had even thought about extending the network over areas like a few square miles. About the only commercial example of broadband being used for LANs was IBM's PC-LAN products. It used only two fairly narrow channels of a broadband network and hence only could transmit and receive data at about 1 Mbps.

Time Division Multiplexing (baseband)

If sharing a network by dividing up the frequency spectrum doesn't fly - then sharing by dividing up time is the only other alternative. The idea here is to use baseband signaling - essentially putting digital signals right on the wire - and sharing it by devising mechanisms for computers to take turns accessing the bandwidth. Through the 70's, a few companies devised ways to build baseband networks. The four most popular systems where IBM's Token-Ring, Xerox's Ethernet, Advanced Interlink's ARCnet and Apple's LocalTalk. These all use a few basic techniques for arbitrating bandwidth.

Packet Contention

CSMA/CD Networks

When Xerox began building high-end printers that would produce copy directly from workstations, they needed a mechanism to get images from the moderately priced workstations to the fairly expensive printers. In the 70's at Xerox's Palo Alto research center, work was under way to develop a shared data network that could do the job.

The goal was to find a simple algorithm - one that could be implemented in the fairly basic silicon available at the time. Ethernet was the result of the efforts and the method for sharing the wire was called Carrier Sense Multiple Access with Collision Detection or CSMA/CD. The idea was a fairly simple one. Listen before you talk - that's the multiple access part and stop talking if you hear some one else, that's the collision detection part.

Essentially, if a station found no traffic on the wire, it could start putting data onto the network. If, while it was putting data on the network, it sensed a collision, the station would stop immediately and wait a random amount of time before transmitting again. The specifics of Ethernet were designed so that once a station got a few bytes into its transmission, all stations on the segment should be able to detect the signal and remain silent until the transmitting station finished. So, on a properly implemented Ethernet, the collision rate should only be a few percent of the packets even when the network was 60 to 70 percent busy or more.

With its 10-Mbps data rate, Ethernet was easily able to handle the transmissions of many PDP-11 or VAX 750 machines (the prominent candidates for networking at the time) without difficulty. Networking a computer at the introduction of Ethernet was not a trivial decision. While today it is rare to find an Ethernet adapter with a price tag above a few hundred dollars, a network interface from the late 70's was likely to cost $5,000 or so.

Today, however, Ethernet has been rendered a single-chip solution and provided by many vendors right on the motherboard of $1,200 PCs. In these configurations, Ethernet now adds no more than $25 to the price of a PC.

Token Passing Rings

Token-Ring, and later FDDI, which employs token-ring techniques, is a newer and more complex method for sharing network bandwidth. Before we get into the specifics of Token-Ring, lets talk a bit about why anyone would find it necessary to build a more complex technology than Ethernet.

In the early 80's, shortly after the introduction of the IBM PC, it was observed that on a network with fast minicomputers and comparatively slow PCs, the PCs could be starved on the network. Further, due to the variety and quality of Ethernet implementations available, many found that Ethernet was unusable when network utilization reached only into the 20 or 30 percent range or so.

Because Ethernet employed random back-offs and was subject to network hogs, it was thought to be unsuitable for mission critical networking. The term bandied about was non-deterministic. In other words, there was no way to mathematically assure that a given station could transmit a given amount of data within any particular time frame. In fact, the folks who thought up Ethernet could show that they could make assurances within high probabilities - but that wasn't good enough.

Token Ring's approach was to arrange stations into a logical ring. Once the ring was formed, a token was generated and passed between the stations on the ring. If a station had data to transmit, it removed the token from the network, transmitted its data and then passed the token along to the next station. Each station could transmit data up to some maximum time or until it was out of data to send - whichever was shorter. In this way, every station is assured access to the network regardless of the station's speed or network interface design.

This sounds really good, doesn't it? There, of course, are trade-offs. First and foremost, the algorithm behind Token-Ring is an order of magnitude more complex than Ethernet. When the ring is running normally, Token-Ring seems pretty simple. However, consider the added complexity of loosing a token, finding two tokens on the network, unexpectedly loosing a station from the ring or bringing a new station into the ring.

Each of these complex cases must be handled correctly. It's just a lot harder to implement than Ethernet's listen-then-talk model. Add to this the fact that IBM picked 4 Mbps as the base data rate for Token-Ring and you can see how simpler, faster Ethernet might be more attractive. Token-Ring also had the problem that if it became deterministic (that is, every station held the token as long as it could), it, too, became painfully slow to use.

If ASIC technology were as advanced then as it is today, Token-Ring might have Token-Ring networks everywhere. It wasn't very advanced, Token-Ring cards were and are expensive when compared to Ethernet and it is hard to make an argument for using IBM's style of Token-Ring.

Concepts: Switched Networks

In either case, whether your technology is Token-Ring, Ethernet or some other shared medium technology, the success of the shared network depends on each station having comparatively little data to transmit or on having relatively few stations on each segment of the network. The more a station can saturate a network, the better it is to have fewer stations on the network.

Switching is basically a technology that is meant to facilitate reducing the number of stations per segment. The term switching is taken from the telecommunications industry where the devices that routed telephone calls were originally called mechanical switches. Switching has come to imply an architecture where any inbound traffic can be redirected to any outbound port with relatively little concern for traffic loss or congestion.

The way in which a switch decides how direct traffic could be by almost any mechanism. It could using bridge or routing techniques, or it could use some other mechanism to pre-determine the path that subsequent data transmissions will take.

Packet-based

To realize the goal of very few stations per switched port, the switch market aims to provide the bandwidth advantages of bridges and routers at a price closer to that of a repeater. The only way to achieve this sort of price point is to rely heavily on ASICs and other custom silicon which has only recently become reasonably cheap to produce.

These chips rip into packets and determine just enough to decide how to direct the packet. Virtually every major networking vendor has either developed such chips or is working on them. As new generations of chips emerge, the chip count on switches goes down and so do the prices. In fact, in the current generations of Ethernet switches in particular, the high-speed uplink ports are the most expensive pieces of the switch. In some cases, an FDDI or ATM port can cost as much as 10 or 12 Ethernet ports.

Cell-based

As good as packet-based switching is, there are certain types of traffic for which they are not ideally suited. Further, the complexity associated with handling variable length packets, each containing their own detailed addressing makes packet-based switching still a fairly expensive proposition.

Cell-based switching is a solution aimed at handling non-data traffic (for example, voice and video) along with data. One problem with router and bridge-based systems is the latency that they introduce to the network. Routers and Bridges almost always fully capture and then forward a data packet. If the router or bridge can do this instantly, up to 1.4 milliseconds of delay is introduced to Ethernet packets traveling through them. Routers, in particular, are likely to introduce more delay because they often must process the packet with a single central CPU.

For data networks, these delays are usually not serious - in fact they usually go unnoticed. However, these delays are significant for video and audio traffic. Cell-based networks in general, and ATM in particular, are architected to handle general digitized data including voice, video and computer-originated data.

The idea behind cell-based networks is to chop standard data packets into much smaller fixed-length cells. In ATM's case, these cells are 48 bytes long with another five bytes for addressing and control.

One fact that should immediately become obvious is that a five-byte address field is too small to hold even a single six-byte physical address. Obviously, there must be something else going on. Indeed, ATM requires that a route be determined before data starts flowing.

The five-byte addresses are only relevant from an end station to a switch or between switches. Each switch then builds a table that includes the translation of incoming addresses with out-going addresses.

By predetermining the flow of data using that predetermined path throughout a data exchange, ATM assures that cells will arrive in the proper order at the end station. In fact, ATM includes no mechanism for retransmission of cells. Higher order protocols must take care of any data lost in the ATM network. However, when cells do arrive in the correct order and an in a timely fashion, it can be a simple matter (that is - cost effective) to retrieve the data, voice or video information contained with in the cells.

We will touch on ATM only lightly here. A much more in-depth discussion will be included later.

Concepts: Network Addressing

Since all traffic on a network must be seen by each and every station on the network, there must be some way to designate which data packets are destined for which stations. In other words, each station must have an address that is unique to its hardware.

It seems clear that if two stations are on two completely separate networks, they really don't need to have

different hardware addresses. After all, they'll never see each other's traffic. Up to this point, we haven't talked about defines a network, so we must more closely define some terms. These terms are used to describe the pieces of hardware that tie a network together.

It should be obvious that hardware addresses and their uniqueness is most important on segments and extended segments. To that extent, the hardware address of any station could be set by the local administrator of any particular segment.

This indeed how ARCnet works. Up to 255 addresses may be configured for ARCnet stations. These addresses are usually set by configuring jumpers on the network card itself. Apple's LocalTalk takes a slightly different tact. Rather than worry about setting addresses, each station just picks one and then broadcasts to see if any other station is using it. For small segments, both of these techniques work well. However, the bigger the network, the less comfortable this scheme becomes.

Ethernet, Token-Ring and FDDI have employed a different technique. Their hardware addresses are considerably longer - six bytes long rather than just one or two. The upper three bytes are assigned to hardware manufacturers who then assign the lower three bytes themselves. The scheme allows for 16 million manufacturers, each of whom can then assign 16 million addresses to their products.

This scheme was originally administered by Xerox for Ethernet until the standards for Ethernet were turned over the IEEE, which now administers hardware address assignment as well.

Each packet that is sent on a network must contain a source and destination hardware address. Some topologies have allowed for different length addresses as well as local assignment of addresses. However, in almost all cases, the globally administered six-byte addresses are used.

Concepts: Bridging

How it works

We've described the basic functioning of bridges. They essentially build a list of known physical addresses and note which port those addresses reside. These addresses are valid only for a certain length of time, after which, if no traffic has been seen from the address, it is removed from the table. Any packet that has a destination address unknown on the originating is retransmitted on all ports of the bridge except the originating port.

If the bridge is a little smarter, it will determine if the address is known on a different port and only transmit the packet on that port which contains the known address. This is essentially the functioning of a very basic switch.

Remember that broadcasts have no known destination and therefore must be sent on all ports of the bridge. This can lead to problems on large networks.

Other problems can occur when media are mixed on a bridge. The most significant problem here occurs when one media has a different maximum allowable packet size than the other (known as MTU or Maximum Transmission Unit).

Some protocols provide for a mechanism called MTU discovery. This is fine as long as the stations are using some connection-oriented protocol and it makes sense to store the discovered MTU. However, if they are using a connectionless protocol, it makes no sense to rediscover the MTU with each transmission.

When to use it

In general, the solution to the MTU problem, is to make bridges that are at least smart enough about higher-layer protocols to participate in MTU discovery if it is used. In the case of TCP/IP specifically, the

bridge must be capable of fragmenting packets.

Packet fragmentation is normally performed by routers and it can be a fairly taxing task for some routers. IP fragmentation is a process of taking large packets and breaking them down into packets as small or smaller than the MTU of the destination media. Bridges (or switches for that matter) that can perform IP fragmentation are generally able to handle any protocol that might be thrown at them.

The bridge's requirement to pass on all broadcasts can cause problems, too. On large networks, usually ones with tens of bridges and hundreds of stations, the propagation of broadcasts through the network can result in other stations creating broadcasts as well. This is known as a broadcast storm. They can last a while and consume as much network bandwidth as is available.

A more common problem occurs when a significant number of broadcasts occur on a fast backbone and have to be propagated to slower media. If broadcasts consume 5 percent of the bandwidth on 100-Mbps media, it probably isn't a problem. However, those same broadcasts would saturate a 4-Mbps Token-Ring segment or take 50 percent of the available bandwidth on an Ethernet segment. That is a significant problem.

Most bridges provide mechanisms for filtering broadcasts and in some cases, this may provide an adequate solution. However, on larger networks at least some routers should be used.

Concepts: Routing

How it works

TCP/IP, IPX/SPX, AppleTalk and a bunch of other protocols all operate at the network layer. That is, they employ at least a two levels of addressing where bridged systems have a flat, universal addressing scheme. Bridging's technique of forwarding packets with unknown destination addresses doesn't scale to global proportions, indeed, it doesn't scale well past a few hundred nodes.

By dividing addresses into a network field and a node field, it is possible to more accurately direct packets. In fact, just this two-level hierarchy is enough to build a global network.

If a router's job were just to steer packets around an internetwork, we'd probably have much cheaper routers than we do. The fact of the matter is that routers usually do much more. They also store and rebroadcast information about the internetwork, keep protocol dependent tables, enforce administrative rules on network traffic and provide redirection for special purpose broadcasts. All of this is fairly CPU intensive, and routers, as a result, tend to be bottlenecks in networks.

When to use it

For all that, routers do have their uses and should in no way be avoided. There is no better way to erect a wall between two different parts of an organization (say, marketing and engineering). Routers are also the only game in town when it comes to connecting your private network to a public network (like the Internet).

Further, routers are the thing to use when connecting networks via comparatively slow wide area networks. If you're paying for wide area bandwidth, you'll want all the control possible over the data that flows across the network.

These are the instances where there is no substitute for routing. However, in the local area network, routing is not the best way to increase the overall bandwidth within your network. That is best done with switches that have some routing smarts.

Concepts: Switching

What's the difference?

Switching has matured beyond simple multiport bridging. There are a number of important features that not only make switching the most economical way to get more bandwidth in your network, they also make a switched network much easier to administer.

In terms of bandwidth, switches provide high speed, low latency bandwidth. Latency is usually much lower than for routers as there is usually less processing going on in a switch as well as many processors (most often ASICs). In instances where traffic is flowing between like media (say Token-Ring to Token-Ring), switches can begin retransmitting the packet before they have completely received the packet. This is called cut-through bridging (as opposed to store and forward) and can reduce latency even more.

On the administrative side, virtual LAN (VLAN) is now a feature commonly found on switches. VLAN technology addresses some of the flaws in bridging without necessarily introducing the complexities of routing.

The idea of VLANs is to take some group of ports on the switch and treat them together as a LAN segment. The net effect of this is to create broadcast domains since all other traffic is still directed only at the port for which it is destined.

Traffic flowing between VLANs must be routed. However, VLANs can usually encompass many more segments than a regular bridged network might have. This reduces the number of router ports needed and often results low levels of traffic between VLANs (often just mail).

Some switch vendors have built routing functions into their switches and others have chosen to not. While some route IPX, IP, AppleTalk and DECnet, most only handle IPX and IP - bridging all other protocols. Depending on the configuration of your network and the ease with which you can reconfigure your network addresses, routing may be worth its additional cost.

When to use it

Switching, particularly as a means to accessing an ATM backbone, will likely be the preferred mechanism for building high bandwidth networks over the next three to five years. Virtually any network that has outgrown a single segment design can benefit from switching. Probably the bigger issue is converting networks that currently employ routers. Reworking network addresses can be a challenge and in some environments it can be almost impossible. (See also switched networks.)

Today's networks

Ethernet

Physical characteristics

Ethernet has been essentially described in four specifications from the IEEE. These build upon the work done initially by Xerox and later by Xerox, Intel and Digital Equipment Corp. together. These specifications involve various types of cables, connection rules and other hardware considerations. However, they all employ the general CSMA/CD algorithms discussed earlier.

Note that in order for CSMA/CD to work properly, there must be a minimum packet size on the network. That minimum size has been set at 64 Bytes and the length of the various network segments where more than two transceivers can exist has been determined based upon the propagation speed of data over the media.

Topologies

10BASE-5 or Thick Ethernet

10BASE-5 is the original Ethernet system. It employs a quarter of an inch diameter, 50 ohm coax cable (with a minimum bend radius of 10 inches). 10BASE-5 segments can run in length up to 500 meters with as many as 100 transceiver connections spaced at least 2.75 yards apart.

10BASE-5 transceivers access the media by piercing the thick coaxial cable. These transceiver taps are known as vampire taps. Since they don't actually require breaking the physical cable, the electrical signals over the cable are typically fairly clean.

10BASE-5 systems were originally envisioned to be cheap and fairly easy to build. The large cable needed simply to be run by rooms where computing equipment would be located. Taps would be made into the cable by using external transceivers. As it turned out, the requirement of an external transceiver and the thick cable, which was expensive and difficult to work with, limited to use of 10BASE-5.

10BASE-2 (A.K.A Thin Ethernet and CheaperNet)

Thin Ethernet was a fairly popular specification and is still used in many environments today. With a maximum segment length of 203.5 yards, it requires that the 50 ohm cable be only .2 inches thick (a bend radius of two inches). It also uses standard BNC connectors and "T's" to provide access to the media. Typically, T's are connected directly to the back of network interface cards, thus eliminating the need for an external transceiver .

Only 30 transceivers can be inserted onto a Thin Ethernet segment and they must be spaced at least 19.69 inches apart. 3Com was heavily involved in developing Thin Ethernet hardware, much as they are today. Their hardware was able to handle slightly longer segments, up to 220 yards in length. Unfortunately, mixing other vendors equipment into an environment where cable runs exceed 203.5 yards can cause problems. For this reason, keeping total lengths to 203.5 yards is a good idea.

10BASE-T

Neither of the coax-based Ethernet specifications lent themselves well to the structured wiring plants that telco workers had been building for decades. Using telco-style wiring was seen as necessary if networked computers were to populate most every desk in the corporate world.

Various vendors realized this and began making Ethernet implementations that could run over standard category 3 twisted pair wiring. The same wiring that drives most every telephone in the world.

The standard eventually came down to supporting 110-yard segments of category-3 cable with a maximum of two transceivers per cable (the end station being one and the hub being the other). Standard RJ-45 phone jacks are used for host connections and transceivers are almost always built onto the network interface card, making the connecting hardware and card very economical.

10BASE-F

10BASE-F is an Ethernet over fiber-optics specification. Its main purpose is to provide long Ethernet runs and electrical isolation either up building risers or between buildings.

Like most other multimode fiber specifications, 10BASE-F segments can go as long as 1.24 miles and accommodate only two transceivers.

Token Ring

Physical characteristics

Token-Ring is heavily used in IBM mainframe environments. It's standardization has taken place in the IEEE

802.5 committee. Token Passing need not be a ring topology, IEEE 802.4 defines Token Bus. However, the ring topology is good since a station that put data on the ring can also take it off, therefore knowing whether the data made it all the way around uncorrupted.

Transmission speeds of 4 and 16 Mbps have been standardized. Data units are always at least 22 bytes long and their maximum length is determined by the Token Holding Time (THT), which usually allows for packets up to approximately 4,500 bytes.

One station on each Token-Ring segment will act as the monitor. This is usually the first station to enter the network, but each station must be capable of acting as the monitor. The monitor has a few very important responsibilities. It must create the original token, compensate for ring jitters, be able to store one whole token so that the token is occasionally fully removed the ring, remove unowned or mangled packets from the ring and finally establish the order of stations on the ring.

Each station must receive and retransmit each packet on the Token-Ring network, so the major concern in Token-Ring is the aggregate differences between the clocks on all of the Token Ring cards. This difference in clock rates - and the potential data loss - is known as 'jitter.' Almost all of the difficulties associated with multivendor Token-Ring networks center around jitter problems.

Topologies

Token-Ring can make use of a wide variety of topologies. The most common today is through active hubs with end-station runs using telephone grade wire. However different limitations exist for four different Token Ring topologies. For each 4-Mbps and 16-Mbps Token-Ring, there rules governing their use over both unshielded twisted pair (UTP) wiring as well as shielded twisted pair (STP).

Until as recently as late 1991, IBM was unwilling to admit that 16-Mbps Token-Ring could or should be run over UTP wiring, preferring STP wiring. Indeed, the Manchester II encoding used to put Token-Ring data onto a wire (it's the same encoding mechanism as used for Ethernet), requires a physical signaling rate of 32 MHz, and the FCC is quite careful about systems that run at these rates as they can interfere with a number of broadcast technologies. Companies such as Proteon and Synoptics (now Bay Networks) had shown an UTP 16-Mbps system commonly and IBM has since agreed to standardize the technology.

For any ring, 4 Mbps or 16 Mbps, the maximum number of stations has been set at 260 stations. The limit on the number of stations is due to total jitter present throughout the ring. Each station has a small buffer that can be used to compensate for differences in clock rates around the network. Only the monitor station, however, is responsible for correcting the ring's apparent jitter. This allows for the one-bit delay between stations. If more than 260 stations are present on a ring, the monitor may not have enough "room" in its latency buffer to account for all the jitter present in the ring.

In reality, it is difficult to build a ring of 260 stations. Somewhat less than 100 is probably a more realistic number. Each adapter must maintain its own clock and each one on the network may meet the requirements for a 260-station ring. Few stations is more prudent.

Passive token ring MAUs or Multistation Access Units where the original means for building a ring as envisioned by IBM. MAU's were unpowered devices that simply allowed for star-shaped rings, thus permitting a structured wiring plant (like the telephone network.) Passive MAUs have given way to active devices that have a number of advantages. Active devices can act as repeaters, and thus elevate concern for signal degradation due to overall ring length.

Whether active or passive, each MAU has a Ring-IN port and a Ring-OUT port. These ports are used to extend the ring beyond the MAU. As with Ethernet, Fiber can be used on these ports for distances of up to 1.24 miles, and up to 550 miles of IBM type 1 STP cable. Conversely, NICs are supposed to be able to drive signals on up to 770 miles of type 1 STP cable. Lobes from MAU to end station and back may not exceed 110 miles when using type 1 cable (since 110 + 550 + 110 would total to the 770 miles a single station can drive.)

If UTP is used for end station runs, then no more than 72 stations are permitted on the ring. Essentially all type 1 cable measurements can be used, however a conversion of 1.1 miles of type 1 cable to .495 miles of type 3 cable must be made.

FDDI

Physical Characteristics

Fiber Distributed Data Interface (FDDI) was the first standardized 100-Mbps technology. From day one, it was and is envisioned to be a backbone technology. Station management, redundant links and fairly flexible architecture give FDDI its backbone flavor. They also make it a fairly expensive technology - especially compared with other 100-Mbps technologies like Fast Ethernet and 100VG-anyLAN.

As the name implies, FDDI was intended to run on fiber. Standards have been written for data grade UTP (category 5) as well as STP wiring. FDDI raw baud rate is actually 125 Mbps as FDDI's minimum data units are expanded from four bits to five bits. The additional bit allows bit patterns to be chosen so that series of '0's are not permitted. FDDI uses a non-return-to-zero, invert on 1's encoding technique. By not allowing consecutive 0's, FDDI can maintain its clocking at a frequency of 125 MHz rather than the doubled frequencies required by Ethernet (although not Fast Ethernet) and Token-Ring.

FDDI uses token passing for its media access arbitration just as Token Ring does. However, rather than specify a flat Token Holding Time, FDDI uses a Target Token Rotation time. The Token Holding Time is then calculated by dividing the target rotation time by the number of active stations on the ring. This addresses one of the faults of Token Ring - that crowded rings get quite slow, and stations may not get a chance to send data for 50 milliseconds or more. On most FDDI rings, the TRT is usually 5 to 10 milliseconds, assuring that each station will have regular access to the ring.

The proper setting for the TRT is some matter for debate. Those who want to see lots of data on the ring and very little token time encourage a high TRT. Those who want to see very regular access and are less concerned about ring utilization push for low TRTs. The 5 milliseconds mentioned above should be viewed as a fairly low TRT, 10 milliseconds is moderate and anything above it is a high TRT. This number is usually configurable on a station by station basis. When new stations enter the ring, a new TRT will be determined, it will be the lowest TRT requested by all stations on the ring.

Jitter is less of a concern on FDDI rings than on Token-Rings. Each station on the FDDI ring has a buffer that is used to compensate for differences in clock rates - as opposed to Token-Ring where only one station responsible for managing jitter compensation. As a result, FDDI has maximum station count of 500 nodes. Each ring can be up to 62 miles in length and the distance between stations can be up to 2 km using multimode fiber. Single-mode fiber can be used for distances of 12.4 miles, but get your wallet out as single-mode transceivers and fiber are extremely expensive.

All stations directly on the ring must be dual-attached stations. That is, four fibers will be used to build two distinct rings and each station must be attached to both rings. The secondary ring is normally not used for data transmission. It is there only to fix faults that may occur. Packets on the two rings flow in opposite directions and should there be a fault (broken cable or down station), stations adjacent to the fault wrap their transmit and receive lines from the two rings essentially forming one big ring.

This bigger ring may actually be up to 220 yards in length. The stations that have wrapped to form the new ring will constantly probe for a fix in the fault and will return the ring to its normal operation when the fault is no longer detected.

Single-attached stations including UTP-attached stations must go through concentrators to attach to the main ring. Concentrators are active devices that manage the insertion of station into the ring as well as provide for link integrity tests and other connection management functions. They also make the architecture flexible in that an end station with a dual-attach card can be "dual homed" to two different concentrators. If the primary

connection should fail for any reason, the secondary connection can still be used to access the ring - a good fault-tolerant option.

ATM

We have described ATM briefly here as a cell-based technology. The technology will be discussed further in a future chapter devoted just to ATM. It is worth pointing out some of the differences between ATM and the technologies that we have talked about up to this point.

ATM is a point-to-point technology. There is no concept of sharing ATM's media. While this seems like a fairly odd choice, considering how expensive it can be to dedicate media and bandwidth to each and every station on the network, it is in fact a fairly logical choice. ATM was originally conceived as a wide area transport for use by telcos. In the telco world, the idea of many stations attaching to the same wire is as outdated as party lines.

This is not to say that two stations' data will never travel over the same wire, indeed this happens all the time, and must happen for the whole notion of a network to be reasonable. However, only two devices share each wire - one on each end. For that reason, the arbitration mechanisms that we've fairly carefully described for shared media networks are not appropriate for ATM. Rather, mechanisms need to be developed to arbitrate the bandwidth that will be available to two stations through the life of their data exchange.

The ATM specification for traffic flow control is called ABR or Available Bit Rate. It is a complex specification that must be implemented in silicon at the same level where packets are segmented into cells and cells are reassembled into packets. In the short term, this makes the economical promise of simple SAR chips (Segmentation and Re-assembly) a little hard to realize.

Another problem that faces ATM networks is their point to point nature. By definition, point-to-point, connection-oriented networks cannot support broadcasts and multicasts. However, we know that upper-layer protocols like TCP/IP and IPX require broadcasts to operate properly.

Finding a way to map the existing networking protocols onto ATM is a complex task. There are two approaches to attacking the problem. One is called LAN Emulation and it basically follows the same model as bridging. The other is called MPOA or MultiProtocol Over ATM.

LAN Emulation (LANE) provides mapping of six-byte LAN addresses into 20-byte ATM addresses as well as providing mechanisms for setting up virtual circuits between stations wishing to communicate, providing broadcast resolution mechanisms and handling for unknown packets. In this way, the ATM network looks like a bridge with various components exploded throughout the network. The devices that provide access from a shared LAN technology like Ethernet into the ATM network are called edge switches. Under LANE, the edge switches need to send broadcast packets to the device on the ATM network that can handle them. However, once a virtual circuit is set up, the Edge switch sends to the intended end station without outside intervention.

The problem with LANE is that when stations need to communicate with other stations not on the same emulated LAN, they must go through a router. That router is a potential bottle neck. A better solution might be to have the ATM network emulate a router rather than a bridge. That is, provide mechanisms for resolving addresses at the network layer and making the edge switches smart enough to determine the route without a router.

That is essentially what MPOA does. It includes the mechanisms of LANE and adds a route server that builds routing tables and pushes them out to the edge switches. The edge switches then need only consult the table to determine the proper path to configure for the destination packet.

While this sounds simple, it isn't. Each protocol needs particular handling and may require more processing than simple routing. For example, if a packet originates on a Token-Ring node and is destined for an Ethernet node, the original data packet may be as large as 4,500 bytes - three times as is allowed by the Ethernet node.

Each routed protocol handles problems like this in different ways and each must be accommodated.

Of course, the ideal way for a station to operate on an ATM network is to set up its own virtual circuits after consulting some central registry for an address (think of this as directory assistance or the White Pages). However, we have a couple decades of development and applications invested in our present applications and we can't just throw all that away - so LANE and MPOA are important to the success of ATM.

Topologies

A number of speeds have been suggested for ATM. Perhaps the first commonly implemented ATM system was the so-called ATM-TAXI system. TAXI is a chipset intended to implement FDDI's physical layer and therefore gave us 100-Mbps ATM. While this technology was instructional, it will not be commonly used in the end.

The telco industry has settled on 155 Mbps (also known as OC-3) as basic rate for ATM service. The next step up will be OC-12 or 622 Mbps. The step down from 155 Mbps is still a bit unclear. However, right now IBM's 25.6-Mbps ATM specification is winning favor as it uses many of the physical interface elements of 16-Mbps Token-Ring. In fact, the first switch port and end-station card combination to come to market with a price less than $1,000 was from IBM.

The problem for ATM25, as it is known, is that it may not represent enough of an advantage over switched Ethernet used in conjunction with LANE or MPOA. On the other hand, virtually any voice or video technology likely to run to personal computers during this century will likely work just fine over a 25-Mbps full duplex system. They may not all work so well over Ethernet. These three rates, therefore, are likely to be the ones to see significant volume over coming the months and probably years.

As might be guessed, ATM622 will require fiber. ATM155 will run over fiber or category 5 UTP cabling. ATM25 will work over category 3 or category 5 UTP wiring.
