Filer Deployment Strategies for Evolving LAN Topologies

Andy Watson | Network Appliance | TR 3009

TECHNICAL REPORT

Table of Contents

Section 1 is not a tutorial in the rudiments of TCP/IP networking. There are many textbooks available to elucidate that topic for the interested student or reader. Instead, the goal here is to construct an evolutionary perspective, providing context for the discussions in subsequent sections. (A network-savvy reader might prefer to skip directly to Section 2.)

Section 2 examines the general options available for high-performance, dedicated file server deployment.

Section 3 explores the continuum of "hierarchical" versus "localized" placement of file servers in modern networks, and illustrates the use of NetApp filers (file servers) in both paradigms.

[TR3009]

Preface to the Reader

Network Appliance continues to evolve its technology and products at a fast pace, with significant new features and performance enhancements introduced every nine months or less over the past three years. This paper reflects the nomenclature and product characteristics at the time of publication. In particular, current NetApp filers supersede the model numbers referenced in this paper.

Abstract

The design of a Local Area Network has significant implications for high-speed, high-volume data access. Even the fastest file server cannot be effectively utilized unless it is implemented appropriately for its LAN context. For example, a dedicated high-performance file server might be configured with;

- a local connection to each of many client subnetworks; or

  a small number of high-bandwidth connections at or near the top of the network hierarchy.

Either approach could provide consolidated access to a large aggregate client population.

Furthermore, the scale of the file server itself has ramifications for the LAN architecture. In some cases, a very large server can be effectively implemented in a central location. Other environments may use a number of smaller file servers distributed at the workgroup and department level. Each approach will make different demands on the LAN; these differences must be considered when designing a network or configuring a file server.

This paper reviews the evolution of LAN technologies and topologies, and explores the issues surrounding file server deployment for the most extensible delivery of bandwidth to users and applications.

Introduction

Managing the growth of a high-performance network computing environment is a challenging, ongoing process. Three interdependent components must be kept in balance:

- Desktop & Compute Server performance;
- Network health and bandwidth; and
- Data storage capacity, accessibility, and performance.

Network Appliance satisfies the third dimension of the problem (data) with *filers* (file server appliances) that are fast, simple, and reliable. However, unless the other two aspects of network computing are also addressed, overall performance will be bottlenecked.

In a static network environment, balancing these three variables would be mostly straightforward, but in the growing, dynamically evolving real world, the experience is a little like playing the "Whack-A-Mole" arcade game. You score points by hammering down the mole in one location, only to have it pop up elsewhere. Laughter issues from the bowels of the machine as you battle it with an awkward three-pound sledge and a growing sense of *deja vu.* If your responsiveness approaches perfection, you'll set a new world's record. If not, the mole gleefully runs amok.

In the arcade game, you need only thrash away at one mole at a time. Unfortunately, in a network computing environment there are those three perpetual targets (Figure 1) for your attention: compute resources (especially desktop workstations and PCs); file servers; and the network itself, which is all-pervasive.

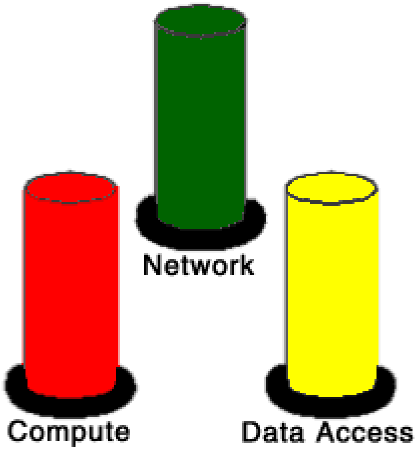


Figure 1: The Three "Moles" of Network Computing

Though more precise than an unwieldy wooden mallet, the resources at your disposal are rarely sufficient to the task in hand. Better than lightning-fast reaction is a clever, proactive strategy: the only way to win is to stay ahead of the problems.

Perhaps you've recently upgraded your network, and last year's bandwidth limitations have been banished from the foreseeable future. Or so you thought. Those new, high-performance workstations installed on the desktops of the Engineering Department are (apparently) evidencing symptoms of data starvation. Is the new network less than the sum of its parts? Or, maybe the servers cannot handle the load, now that the previous bottlenecks of constrained network and sluggish workstations have been eliminated? And beyond *this* current obstacle, which network computing "mole" will laugh at you next?

Network Appliance's high-performance filers can be used to best advantage if the other two elements of network computing are appropriately implemented. This paper addresses the issues that have historically faced network managers and system administrators as they evolve the network and server infrastructure required to realize the full potential of ever-faster computational resources and increasingly ambitious application software.

Section 1: A Chronology of LAN Technologies

Local Area Networks (LANs) have evolved to feed the growing I/O appetite of ever-faster computer systems. It would have been all but impossible a dozen years ago to predict the network technology of 1996, or to foresee its widespread implementation. In the early days of TCP/IP networking, the limited power of computer systems was easily matched by the networking media and devices. Since then, computational horsepower and networking bandwidth and connectivity have increased significantly. Collaborative network computing demands more shared data access than ever before.

File servers have likewise come to play a critical role in the management and delivery of data. System Administrators (SAs) and Network Managers (NMs) have developed LAN topologies designed to facilitate optimal data access. The deployment options for file servers have expanded with the evolution of LAN topologies. This Section reviews the history of LAN technologies, with special attention to the placement of file servers.

Simple Ethernet-based LANs

In the early '80s, Ethernet's 10-Mbps (Megabits per second) bandwidth appeared inexhaustible. Hundreds of modestly powered computers, running mostly standalone applications, could be served by a single Ethernet LAN. At that time, single coax cables were typical of such LANs: "Thick Net" (10Base5) was soon largely replaced by the more popular "Thin Net" (10Base2) because of its simpler connectors and lighter, more pliable construction. Every computer system on the LAN was part of the "daisy chain" of devices connected directly to a single shared coax.

As computer systems became faster, application software emerged which exercised networks in new ways. Physical and bandwidth limitations drove the creation of multiple LANs, and a new set of internetworking goals emerged (Table 1).

Goal (1)	Provide Optimal Connectivity
Goal (2)	Maximize Effective Bandwidth
Goal (3)	Facilitate Interaction/Isolation Between Disparate LANs

Table 1: Internetworking Goals

The simplest interpretation of Goal (1) might be the extension of an individual LAN segment (usually to support more workstations or PCs). More interesting is the creation of paths *between* multiple segments, subnetworks, or distinct LANs. The fundamental role of any network is to provide connectivity, of course, but Goal (1) implies connectivity beyond the local physical and/or logical network.

Goal (2) may seem obvious, but there are trade-offs between maximizing the utilization of a network and optimizing the effective performance experienced by end users. The *effective* bandwidth is a (hopefully large) fraction of the *potential* bandwidth. One common problem is broadcast traffic, which if frequently transmitted by a large number of nodes can collectively consume significant bandwidth. Similarly, externally-generated traffic, passing through on its way to another external destination, can "pollute" a local segment or subnet, leaving less bandwidth available for local users' own traffic. In designing a large LAN, it makes sense to avoid congesting multiple segments or subnets with each other's traffic.

Goal (3) reflects the need to accommodate multiple protocols and diverse applications. Sometimes these interact, with appropriate translation, and in other cases they simply coexist separately. For example, protocols or applications with a reliance on broadcast traffic (e.g., Appletalk) might need to be isolated to avoid congesting non-Appletalk environments with irrelevant traffic. A counter-example would be electronic mail, where messages may be exchanged between mail servers using IP and others using SNA or IPX.

At first, LANs were extended with *repeaters*, devices which passively amplified signals from one cable segment to the next, without any kind of filtering of traffic. This addressed Goal (1), by increasing connectivity, but often *detracted* from Goal (2) -- adding nodes to a single Ethernet-based LAN increased contention for the same fixed 10-Mbps bandwidth. The contention itself reduced available bandwidth due to the inherent nature of Ethernet.

If multiple devices simultaneously attempt to transmit over the same Ethernet, a *collision* occurs. Both packets involved in a collision are lost and must be retransmitted, after a waiting period. Ethernet interfaces employ collision-avoidance methods to detect other traffic on the network and await opportunities for safe transmission. (This is also known as *Ethernet Deference*.)

The presence of larger populations of devices on the same LAN increases the statistical likelihood that two nodes will both act upon the same perceived transmission opportunity, resulting in a collision. As collision rates rise, effective bandwidth availability for that Ethernet is reduced.

In the mid '80s, *bridges* were introduced. Repeaters were replaced by bridges at many existing sites, and have seen significantly reduced use ever since.

Like repeaters, bridges also extend LAN connectivity, advancing Goal (1). But whereas a repeater is strictly passive, blindly regenerating signals from one segment to the next, a bridge actively *recreates* the packets it forwards. This means that a bridge can be used to extend a LAN over arbitrary distances -- or for use in MANs (Metropolitan Area Networks) and WANs (Wide Area Networks). Repeater-implemented networks are constrained in overall length by the maximum propagation time of a signal on the resulting single physical Ethernet.

Bridges are also better than repeaters with respect to Goal (2).

Each segment in a bridged network is an isolated collision domain. Signals do not propagate from one segment to another without being interpreted by the bridge. A bridge uses the Data-Link Layer(1) information in each packet's headers to maintain tables which map physical addresses to specific bridge *ports*. An incoming packet is forwarded to its exit port based on its destination address. Bridge ports use collision avoidance methods to attempt safe packet transmission, just like any other Ethernet interface. Therefore, even if two devices on two *bridged* segments transmit simultaneously, no collision will result. Both packets will first be forwarded by the bridge before the other device detects the other's traffic, and the bridge's transmitting ports will employ collision avoidance.

Bridges are not perfect, however. Forwarding takes time -- the latencies incurred while transiting the bridge might comprise a significant percentage of response time(2). Also, some bridges (especially older models) may not be able to sustain the throughput required by high-performance systems, causing some packets to be "dropped." When using such bridges, it is often necessary to throttle throughput by reading and writing smaller blocks of data per file service operation (e.g., Read or Write 1-KB blocks instead of 8-KB blocks).

The Impact of NFS

With Sun's introduction of NFS (Network File System, a protocol for file sharing) in 1985, and the availability of affordable -- and often diskless -- Sun workstations (with built-in Ethernet interfaces), Ethernet utilization on LANs rose dramatically. Increased use of NFS brought with it the conflicting phenomena of greater bandwidth requirements *and* increased likelihood of high collision rates per Ethernet. By the late '80s, LANs were noticeably evolving to accommodate increasing NFS traffic.

To reduce collisions -- and thereby increase effective bandwidth, as per Goal (2) -- populations of systems *per Ethernet segment* were reduced, and additional segments were added as needed. Bridges were employed to provide connectivity between segments.

SAs at NFS-intensive sites deployed one or more file servers per bridged segment (Figure 2a). But for traffic to reach one segment from another, it might have to pass through one or more intermediary segments, conflicting with Goal (2). For example, if a client workstation on Segment #1 accessed files exported by a server on Segment #3, then all of that traffic would traverse Segment #2, reducing the effective bandwidth available to systems residing on Segment #2.

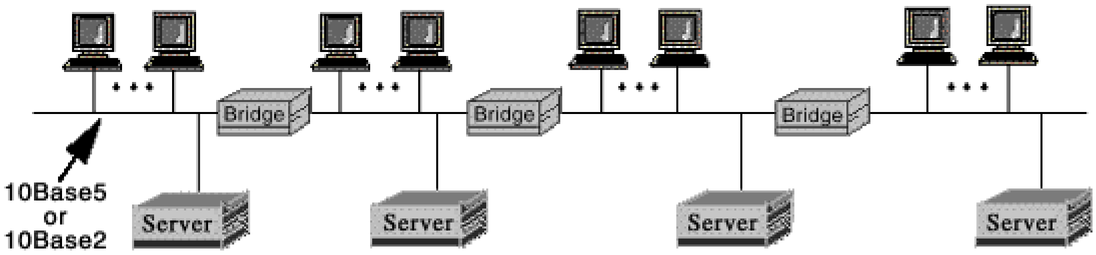


Figure 2a

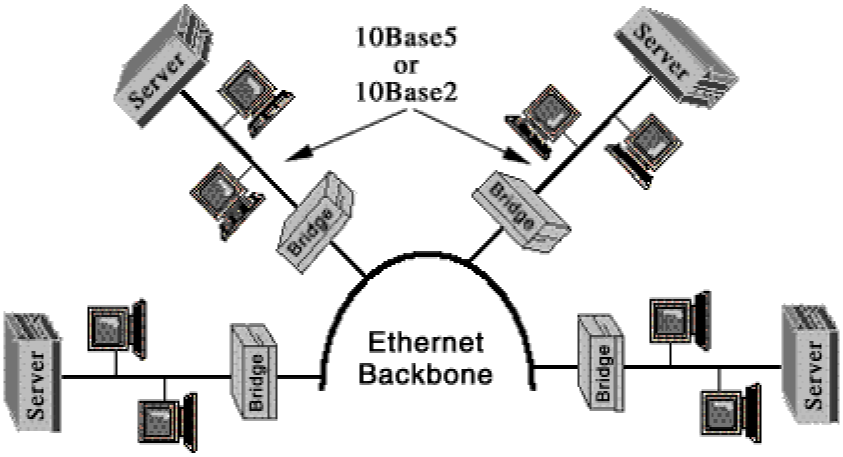


Figure 2b

Implementing a *backbone* for intersegment traffic (Figure 2b) eliminated this problem. Bridges connected segments to a backbone instead of directly to each other.

Routed Internetworks

Unlike bridges, which extend a single network, routers connect logically distinct networks. Routers operate on information in the Network Layer, allowing them to use multiple Center-to-Edge™ routing paths. This enables complex network topologies to be implemented with redundancy, load-balancing, and dynamic adaptation to changes in the network.

The introduction of routers fostered the creation of subnetworks ("subnets") with their own distinct logical address spaces -- essentially separate LANs, often based around different protocols. Broadcast traffic can be straightforwardly localized on subnets by the use of properly constructed subnet masks. Furthermore, the logical address space of very large networks would be impossible to manage were it not for routers and subnet partitioning.

Routers provide connectivity to other subnets, isolate broadcasts and collisions, and -- because they operate on the Network Layer -- transcend protocol differences to facilitate interaction where desired or minimize cross-protocol interaction where it would be counter-productive. In other words, routers address all three goals of internetworking as shown on Table 1.

The availability of high-performance routers facilitated a move toward hierarchical topologies. The simplest hierarchy has only a single layer in its structure, where the router is the backbone (Figure 3).

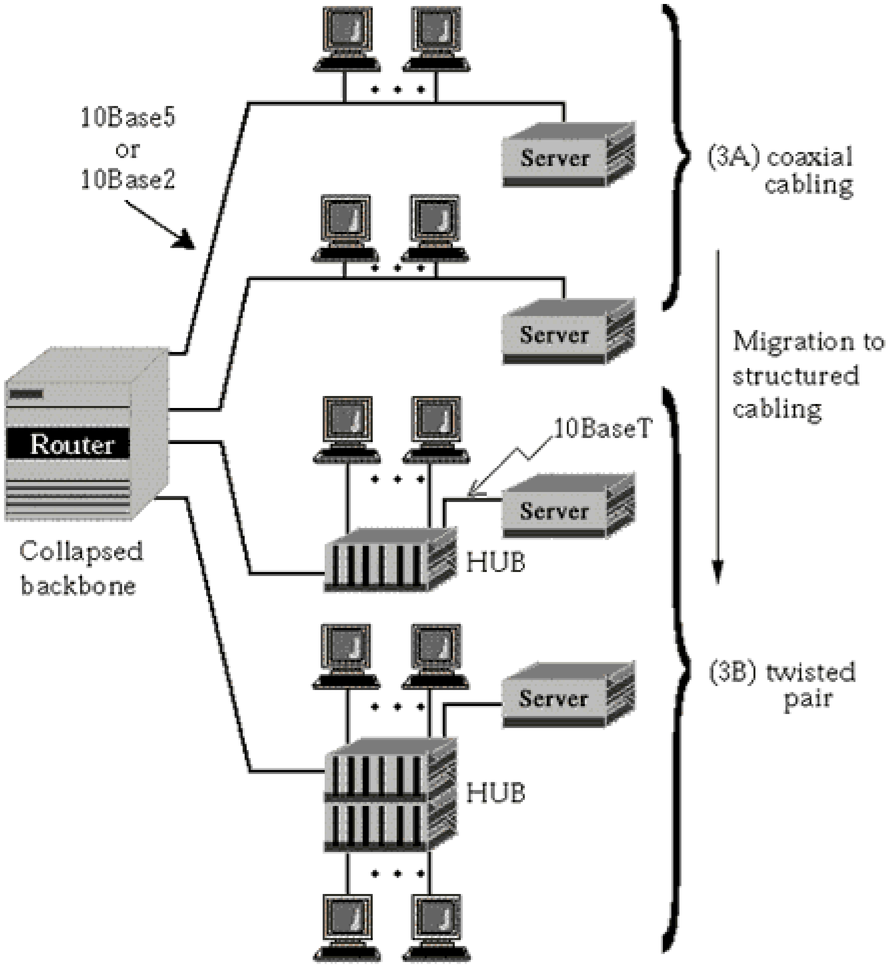


Figure 3

Structured Cabling: UTP and Hubs

By the time "collapsed backbone" topologies became feasible (early '90s), most large sites had implemented twisted-pair (UTP(3)) wiring for Ethernet (Figure 3b). UTP is less expensive and more reconfigurable than Thick- and Thin-net.

By simply moving a UTP cable from a port on one *hub* (see below) to another port on another hub, a computer system's "network drop" can be quickly relocated to a new subnet. Many larger networks consolidate their hubs for all subnets in telephone closets, and "patch" between them as needed (Figure 4).

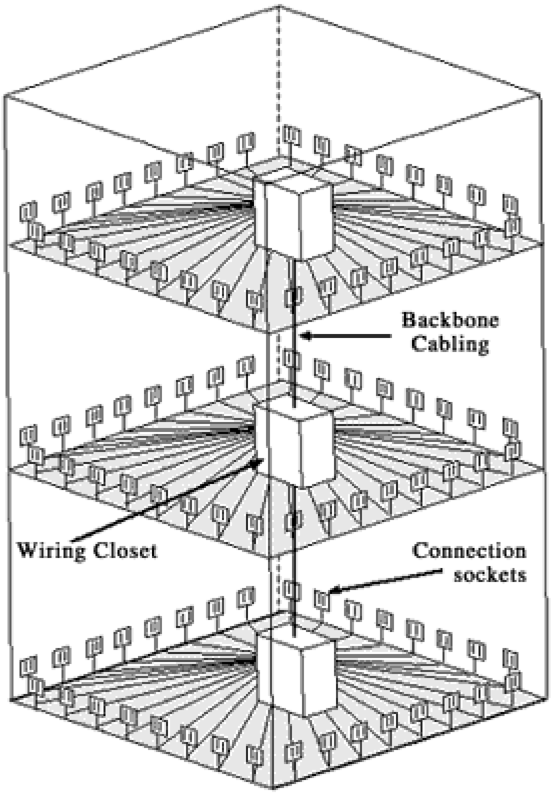


Figure 4

A hub is essentially a "network in a box," eliminating the daisy-chaining of a coax-based network. Some types of hubs have enhanced capabilities. An *intelligent hub* might have a management interface, be able to respond to remote network monitoring tools, and/or isolate ports where problems have been detected. UTP cabling is virtually always implemented with hubs.

In the case of FDDI (fiber-optic or copper (4)), hubs are sometimes called *concentrators* (functionally, they are equivalent). Figure 3b illustrates the one-node-per-port approach of UTP and hubs as compared to the shared-media, coax-based topology of Figure 3a.

Switches

More recently, *switches* have become popular as replacements for hubs, providing the same advantages versus hubs that bridges offered over repeaters. This derives from the fact that a switch is essentially a multi-port bridge. Generally, switches are only modestly more expensive than hubs and offer opportunities for greater aggregate bandwidth.

Some switches have enhanced capabilities that allow them to act as routers. Conversely, some routers can be configured to act like bridges. The resulting overlap category has spawned the term *brouter* to describe such "routing bridge" hybrids.

Some switch designs use a matrix of connections between every possible combination of ports. Others move traffic through a central memory repository. Most switches employ a very fast backplane. In all cases, most switches use traditional store-and-forward bridging methods, with similar issues of latency(5).

To improve performance, some vendors employ "cut-through" switching, wherein packet forwarding commences immediately upon arrival of the destination address portion of the packet header, even while the remainder of the packet is still being received on the incoming port. However, cut-through switching introduces some risks (e.g., the remainder of the packet might turn out to be corrupted), and so might not necessarily see wider use in the future.

Switches also contribute to overall throughput by simultaneously passing packets between pairs of segments (Figure 5).

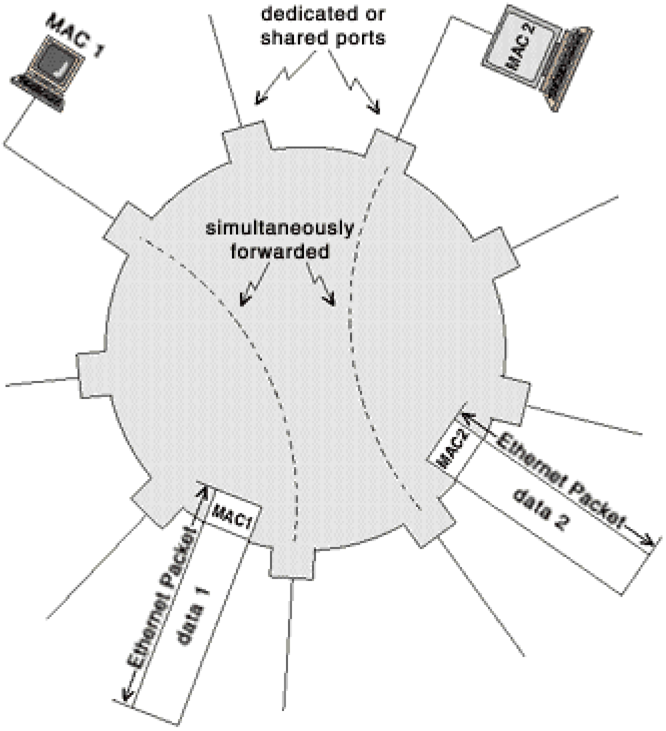


Figure 5

To maximize performance using a switch, each attached device may have its own dedicated port. Not all systems warrant a dedicated switch port. Multiple users' desktop systems, for example, might share a switch port through a hub (Figure 6), if none of them are running I/O-intensive applications on high-performance systems.

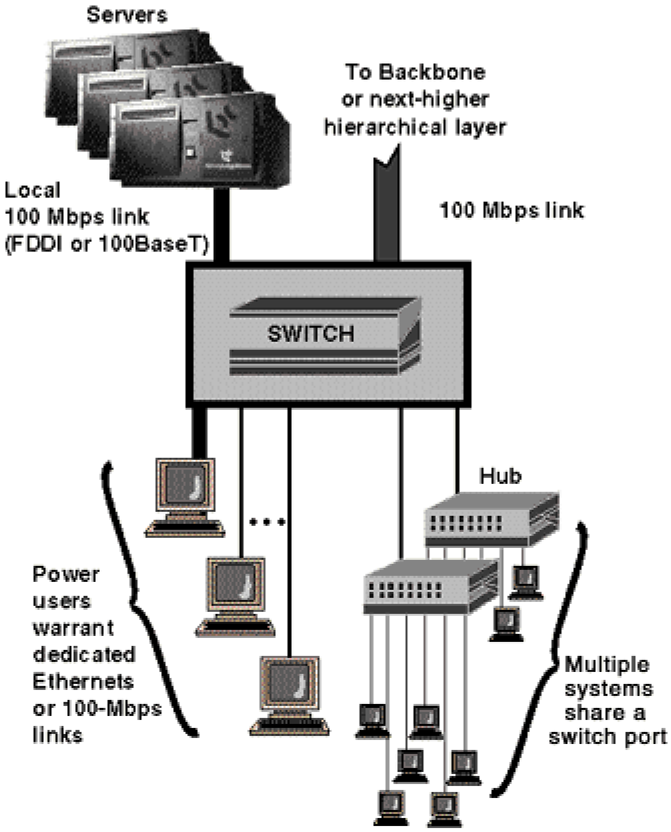
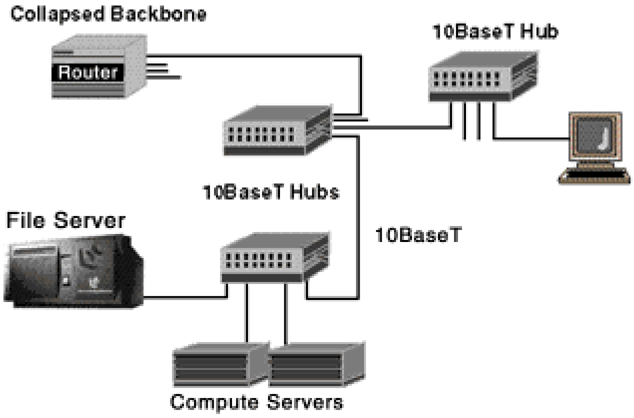

Servers
To Backbone or next-higher hierarchical layer
Local 100 Mbps link (FDDI or 100BaseT)
100 Mbps link
SWITCH
Hub
Power users warrant dedicated Ethernets or 100-Mbps links
Multiple systems share a switch port

Figure 6

In addition to providing numerous 10BaseT interfaces, many switches are modular, and may be configured with one or more 100-Mbps connections. Such ports are especially useful for file servers, where the aggregate traffic from large client populations requires higher bandwidth. (In fact, it is not unusual for a NetApp filer to use *multiple* 100-Mbps ports on a switch. With conventional file servers, this would be difficult if not impossible, because of complications with routing tables on the server. But NetApp filers always reply using the same interface on which an incoming request was received.)

Switches are straightforwardly installed into existing structured cabling environments. Figure 7 (below) depicts a popular upgrade path that is repeating itself at many customer locations.



Before

After

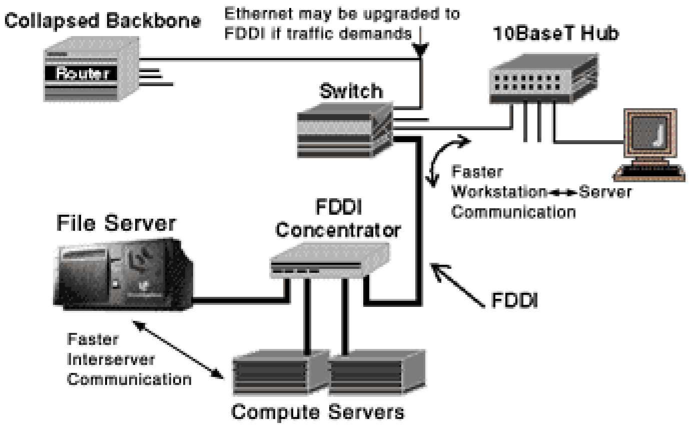

Collapsed Backbone
Router
Ethernet may be upgraded to FDDI if traffic demands
10BaseT Hub
Switch
Faster Workstation↔Server Communication
File Server
FDDI Concentrator
FDDI
Faster Interserver Communication
Compute Servers

Figure 7

With no perturbations at the periphery of the network (where the workstations reside), selected hubs are replaced by switches and 100-Mbps concentrators. The concentrator gives each of its servers a wider data path by which to deliver service to workstations and communicate with one another. The switch conveys those services more efficiently to downstream workstations. Thus, a switch can breathe new performance life into 10-Mbps Ethernet, forestalling desktop upgrades, and limiting more expensive 100-Mbps technologies to the wiring closet and low numbers of closet-to-closet runs.

100-Mbps Options

Whereas FDDI is the dominant technology in older backbones, at the workgroup level UTP-based FDDI-TP ("copper FDDI") and 100BaseT (Fast Ethernet) are prevailing cost-effective alternatives. FDDI-TP had a head start in the market and still has a performance edge (owing to its collisionless, token ring approach), but Fast Ethernet is gaining ground(6).

There are many factors contributing to Fast Ethernet's increasing popularity.

- It is less costly, at the hub, switch, and workstation levels, than other 100-Mbps alternatives (e.g., ATM or FDDI).
- Most network equipment vendors offer switches that support Fast Ethernet.
- Fast Ethernet enjoys the support of more than 75 network and system vendors, including industry leaders such as Cisco, Bay Networks, 3Com, Cabletron, Intel, DEC, and Sun. This extensive multivendor support ensures the development of a wide range of interoperable products at very competitive prices.
- Workstation market leader Sun includes Fast Ethernet interfaces standard on its newest UltraSPARC-based systems.
- Existing structured cabling infrastructure can often be used with Fast Ethernet.
- SAs and NMs have many years of practical experience with 10-Mbps Ethernet, doing problem resolution and traffic capacity planning; this corporate asset is a motive to stick with a kindred and more familiar technology. (3Com reports there are 60 million Ethernet users, worldwide.)

ATM (Asynchronous Transfer Mode)

ATM is a cell-switching and multiplexing technology that combines the benefits of dedicated circuits (invariant transmission delay and guaranteed capacity) with those of packet switching (flexibility and efficiency for intermittent traffic). The fixed length of ATM's cells (53 bytes -- 48 bytes for the "payload" and 5 bytes for headers) facilitate high-speed implementations that can support isochronous (time critical) applications such as video and telephony with constant flow rates, in addition to more conventional data communications between computers where fluctuation in packet arrival rates is typically not problematic(7). ATM standards define a broad range of bandwidths -- from 1.5 Mbps (via T1 or DS1) to 622 Mbps (OC-12) and above -- but most commercially available ATM products currently provide 155.52 Mbps (OC-3) or 100 Mbps (TAXI). ATM is currently implemented over fiber connections and various twisted-pair wiring alternatives.

All devices in an ATM network attach directly to an ATM switch. Multiple ATM switches can be combined in a fabric sometimes called an "ATM cloud" and *virtual circuits* can be dynamically created between any two nodes on one or more ATM switches. So long as the switch can handle the aggregate cell transfer rate, additional connections to (and through) the switch can be made.

ATM continues to evolve as the various standards groups(8) finalize specifications for interoperability. Particularly thorny is the question of how best to implement connection-less IP traffic via connection-oriented ATM. The ATM Forum and IETF (Internet Engineering Task Force) are attempting to develop specifications for this, but it hasn't been easy. Three approaches are under consideration.

- (A) The "Classical IP" over ATM specification (RFC 1597) uses an ATM cloud to emulate an *IP subnet*.
- (B) LANE (LAN Emulation) uses ATM to emulate properties of a shared-media LAN *segment* at the OSI Data-Link layer.
- (C) The forthcoming Multiprotocol Over ATM (MPOA) specification hopes to address routing deficiencies of LANE.

Detractors of (A) and (B) criticize scaling problems, single points of failure in supporting services (e.g., address resolution), and complicated network management. Critics of (C) complain that MPOA is overly complex.

All things considered, pundits predict that in most LAN environments, ATM equipment costs, installation complexities, and interoperability challenges are likely to limit initial implementation to campus backbones and MANs, with use in WANs increasing over the next few years. Another contributing factor in ATM's relatively slow pace of adoption (compared to Fast Ethernet, for example) is the dearth of video and audio applications which could exploit ATM's isochronous data delivery features. Some performance-sensitive sites will nonetheless implement ATM as soon as possible, wherever the fastest (lowest latency) commercially-available networking technology is required.

Section 2: Mapping Data Topology onto Network Topology

In Section 1, three fundamental approaches to file server deployment were shown to be possible:

- (1) One or more file servers per subnet;
- (2) Multiple network interfaces on each server - one per subnet;
- (3) One (or more) high-bandwidth interface(s) per server, accessed from multiple subnets via intermediaries like switches.

In fact, there is a fourth category:

- (4) A combination of methods (2) and (3).

This Section discusses the general case for (1), (2), and (3), above. Section 3 will analyze real-world implementations which, unsurprisingly, reflect the eclectic pragmatism of (4).

Server-per-Subnet

The server-per-subnet approach is trickier to implement and maintain when applied to large sites. It doesn't scale well; the pitfalls are many.

Maintaining a great many small servers imposes expensive and exhausting administrative overhead -- fixing problems, upgrading hardware and software, tuning and load-balancing for better performance, performing daily backups, archiving project milestones to tape, and so on. Keeping replicated files in synch across multiple servers is arduous and fraught with risk. For example, if a library or executable used by engineers accessing multiple servers is updated but not properly propagated, a whole day's work might have to be redone. And good performance is hard to achieve, and harder still to sustain, mostly due to inevitable cross-traffic between subnets.

A complicated assemblage of intrinsically complex (and probably overloaded) components is extremely unlikely to run smoothly or reliably. Except for the life-saving strategy of file server consolidation (see below), there is little else that can be done to constrain the problem to manageable dimensions. And no one could have greater appreciation for the benefits of server consolidation than someone who has lived through a heroic, months- or years-long, large-scale server-per-subnet attempt. (Figures 2b, 3a, and 3b, in Section 1, depict the server-per-subnet approach.)

Over time, the number of NFS clients per subnet continually decreases as workstation performance -- and related appetite for file service -- increases. For example, installing faster machines on the desktops of 30 users sharing a subnet usually necessitates splitting the group onto two or three subnets. Spreading a workgroup over multiple subnets means that a larger fraction of server accesses will be one hop(9) away. Additional routing (more hops) adds delay, and decreases user productivity.

Relatively slow(10) file servers aggravate this problem. The typical general-purpose computer system used as a server in a server-per-subnet approach is a workstation retrofitted for use as a server, with few (if any) optimizations for file service. To compensate for lack of file server power, SAs (System Administrators) must install more of them, further complicating the situation.

These pseudo-servers are also, as a rule, NFS *clients* mounting from one another, creating interdependencies which bring entire networks to a standstill when any one server goes down. This "cross-mounted" condition is difficult to avoid for two reasons:

- General-purpose servers are inevitably used for tasks other than dedicated file service, with other applications and services requiring access to files on other servers; and
- SAs use cross-mounting to create a shared logical name space for directory hierarchies.

However, over-stressed retrofitted-workstation "servers" are prone to instability, and the interdependence of cross-mounting translates into widespread downtime. While one server hangs unresponsively, awaiting human intervention to reboot, others are also down or sluggish. Once rebooted, the conventional file server (unlike a NetApp filer) must then verify all its file systems' integrity (using the Unix *fsck* utility), perhaps taking several hours to return to full operation.

When not responding to down servers, SAs in these environments are kept busy load-balancing Ä redistributing utilization across multiple subnets (attempting to maintain as much locality of reference as possible), multiple file servers (spreading around an aggregate demand that no single conventional server can handle), and multiple file systems (redistributing the I/O over steadily growing numbers of disks). Striving to perform reliable (restorable) file system backups in this environment adds further complexity to an SA's unenviable burden.

The near-futility of the server-per-subnet method of file server deployment created the market opportunity that inspired the development of high-performance, dedicated file servers with higher storage capacities, more networking bandwidth, and greater reliability.

Multi-homed Servers

Consolidating file service is an important strategy in data deployment. Multi-homed(11) servers can be directly attached to all subnets (attached in parallel with a router, as shown in Figure 8), improving performance by eliminating hops. This also improves reliability by eliminating cross-mounts, and by being better able to handle heavy file service loads. Replication of data, as required in the previous server-per-subnet approach, is eliminated.

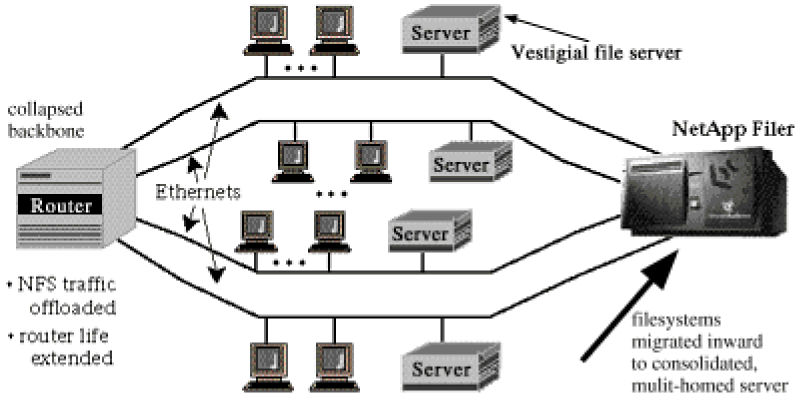


Figure 8

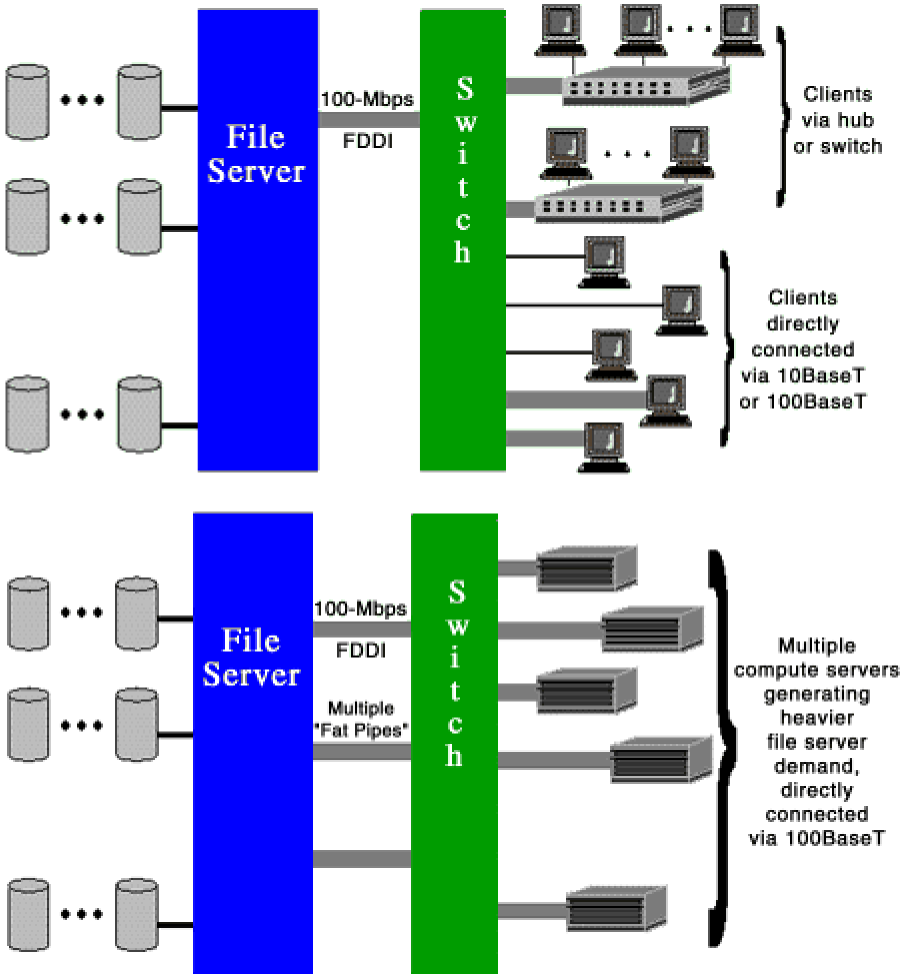
A single, consolidated file server has other advantages. The cost of administering a single, reliable file server is clearly less than for a multitude of cross-mounted and load-stressed retrofitted-workstation servers. No data need be carefully replicated (and stored redundantly, making for additional savings in reduced disk storage capacity requirements). All backup and restore procedures can be simplified. (With NetApp filers in particular, many user requests for restoration of accidentally-deleted or overwritten files are eliminated entirely -- thanks to the Snapshot feature of NetApp's WAFL file system, which allows users to restore most such files on their own without burdening the SA. Furthermore, full-integrity backups of live file systems can be made from a Snapshot, such that there are no scheduled backup events during which users cannot access and update their files, as with other conventional servers.)

Load-balancing exercises are vastly reduced by server consolidation. Network load balancing is dramatically reduced. All file-service-related subnet traffic can remain isolated, with no situations where a client (which might have been another server) need pollute any other subnet with its file access traffic. File space allocation among multiple servers is replaced by the straightforward administrative practice of capacity planning on one easily monitored system. (With NetApp filers in particular, file space allocation is rendered almost trivial by the single large file system, spread over all disks, and the "soft partition" mechanism of *tree quotas*, which allows re-allocation of disk space on the fly simply by editing the */etc/quotas* file. NetApp filers also allow live, dynamic growth in the capacity of the file system, with a procedure that takes about 30 seconds.)

With lowered costs of acquisition and operation, and increased reliability and availability, it seems almost too good to be true that a single, consolidated file server could also offer *better* *performance* than the server-per-subnet approach. However, a dedicated server, designed for only that purpose, has many optimizations over general-purpose machines. And the elimination of both router hops and collision-inducing (bandwidth-robbing) traffic on remote subnets can have a significant positive effect on a network's health and performance.

Fat Pipes

An alternative to outfitting a consolidated file server with many 10-Mbps Ethernet interfaces (one per subnet) is shown in Figure 9. A single "fat pipe" (high-bandwidth network interface) can accommodate the aggregate file access traffic of many NFS clients with excellent, high-performance results.



File Server
100-Mbps
FDDI
Switch
Clients via hub or switch
Clients directly connected via 10BaseT or 100BaseT
File Server
100-Mbps
FDDI
Multiple "Fat Pipes"
Switch
Multiple compute servers generating heavier file server demand, directly connected via 100BaseT

Figure 9

A consolidated high-bandwidth interface for a file server works well in most cases because, typically, not all subnets are saturated with file-access loads at the same time. And even when they *are* all moderately busy, the demand may not be sustained and the peaks will probably not occur together. Therefore, their collective bandwidth can usually be "collated" onto a single 100-Mbps file server interface (Figure 9a). This is especially true if the single "fat pipe" is FDDI, where over-90% utilization is possible without significantly degrading performance (as is the case with collision-sensitive Fast Ethernet, where utilization of over 70% is difficult to achieve).

Only in a somewhat unusual network computing environment would the activity on all subnets simultaneously create sustained, heavy file service demands. Typically, such conditions are only seen when the NFS clients are headless compute engines executing continuous analytical tasks (e.g., simulations). In that case, multiple 100-Mbps interfaces should be provided (Figure 9b).

Not all dedicated file servers can operate effectively in the configuration shown in Figure 9b. NetApp filers, unlike most conventional servers, do not select from multiple available network interfaces based on internal routing tables, for purposes of sending replies to NFS client requests. Instead, a NetApp filer always replies on the same interface which received the incoming request. This allows multiple clients connected to the same switch to explicitly mount from different addresses (corresponding to different interfaces) on the same filer, and have the server's replies distributed over those same multiple interfaces. A conventional server would insist on using the same interface for all replies to all clients on the same subnet (even though they are attached to different ports on the switch).

Furthermore, high-performance switches usually have robust throughput capabilities, and add very little latency, so there is no reason not to take advantage of the ease with which the network can be grown by adding switch ports as needed. And, if the switch can be operated in a matrix with other switches (as is the case with ATM), then it becomes less important where in the switching fabric the file server's fat pipe is physically connected with respect to the clients.

Another reason to consider using the "fat pipe" approach to file server deployment is that it facilitates faster NFS response times, for several reasons:

- 100-Mbps networks are simply faster than 10-Mbps (Ethernet) networks, incurring less latency in transit;
- The elimination of Ethernet deference delays can bring a substantial performance boost at higher levels of network utilization(12) ; and
- With FDDI, the larger MTU (Maximum Transmission Unit) reduces the overhead of fragmentation and reassembly(13) .

Finally, by configuring servers with a smaller number of "fat pipes" (versus a multitude of standard Ethernet interfaces), file server acquisition costs are lower. This savings derives from both the lowered cost of the server (with fewer network interface cards in its configuration) and because fewer ports on the switch (or hubs) are required to deploy the server. In the Big Picture, however, this cost savings is small, and should not be a dominant factor in any data deployment decisions.

Section 3: Hierarchical vs Localized File Server Deployment

In applying its filers to hundreds of networks, Network Appliance has observed a full spectrum of data deployment, ranging from localized to hierarchical. Some customers have exploited the many-Ethernet configurability of a filer (up to 17 10-Mbps Ethernet interfaces are supported on an F330) in a fully localized manner. Others have employed a smaller number of affordable 100-Mbps FDDI and Fast Ethernet connections to provide optimal data access. In some cases, this might represent localized deployment, with the filer directly connected to an FDDI network populated by high-performance NFS clients. In other cases, it might be purely hierarchical, with filer placement one or more hops removed from its clients. In actual practice, it is common to see elements of both strategies used simultaneously: filers are often outfitted for localized access by one population of primary clients, and hierarchically accessible to the rest of the LAN.

Hierarchical Data Deployment

This approach collects all data at the root of the network, usually together with other shared compute-server resources for analysis/simulation, database, web, mail, news, and so on. There is usually a high-speed network directly connecting these servers to each other (in itself a localized strategy for data deployment if any of the other servers are clients to the filer), with one or more router(s) providing paths between the servers and the user community's desktops (Figure 10)(14).

Broadly, there are three factors why some sites have implemented a hierarchical data topology.

- Collecting together all servers and routers in a central computer room environment is useful for administering the equipment. In security-conscious companies (or at environmentally challenging installations) all servers and routers may be placed in a single, locked, clean room served with uninterruptible power and heavy-duty air-conditioning. This *physical* consolidation of everything but the hubs, which usually remain dispersed in telephone closets, encourages a hierarchical perspective in the minds of SAs and NMs.
- In very large populations, aggregate NFS load may be high, but individual users may simply not need the highest possible performance. The latency cost (maybe 5-10 milliseconds) of being one or two hops away from their data may be acceptable, especially if the network is healthy, such that unacceptable network-induced latencies are rare.
- By restricting filers (and related higher speed intra-server traffic) to the computer room, some sites have been able to extend the useful lifetime of their wiring, which might include older Thick- or Thin-net based cables snaking through ceilings across an extensive campus. For example, depending on network traffic patterns, perhaps only the computer room's servers truly need 100-Mbps bandwidth amongst themselves.

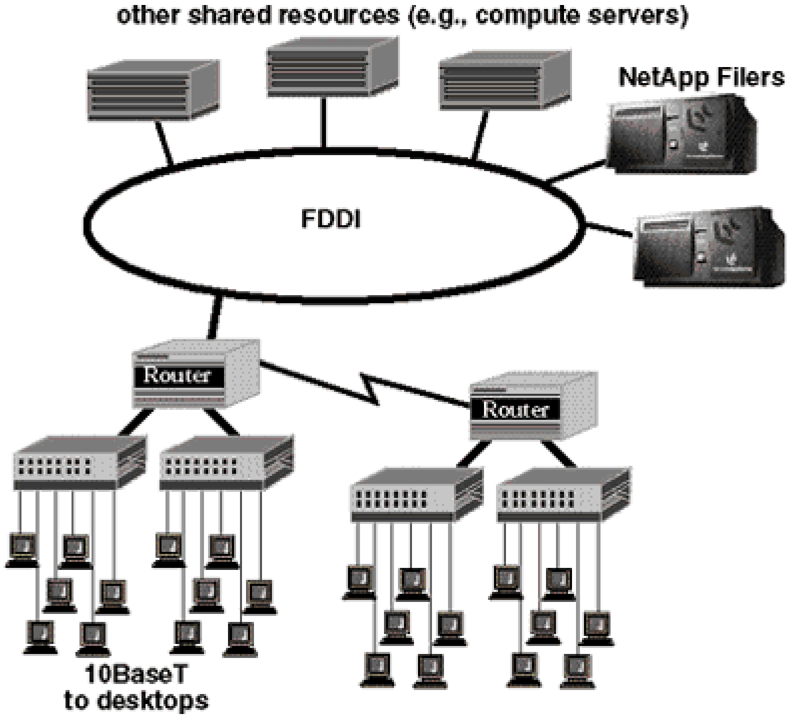


Figure 10

In the purest form of "client/server" computing, the desktops execute relatively trivial applications which generate back-end work for high-performance servers.
A hierarchical strategy for file server deployment is consistent with that model, especially in that the heavy and steady communications traffic between the compute engines and the filers is kept

off the lower levels of the network, allowing larger numbers of desktop systems to coexist happily in higher numbers per segment or subnet.

Localized Data Deployment

Localized data deployment is an increasingly common strategy. File service can be dispersed campus-wide, and kept as topologically close as possible to its point of use (although the servers might still be physically installed in a remote computer room). Network congestion is avoided, especially in the case where all servers and clients have their own dedicated ports on a switch or concentrator, for example.

Figure 11 illustrates a typical configuration. All data and computational resources enjoy direct, local (zero hop), 100-Mbps access to the file server. The total assemblage of equipment often fits in the same wiring closet, sharing the same standard 19-inch racks. This structural unit -- switch, concentrator, servers -- can be repeated on a per-workgroup or per-wiring-closet basis.

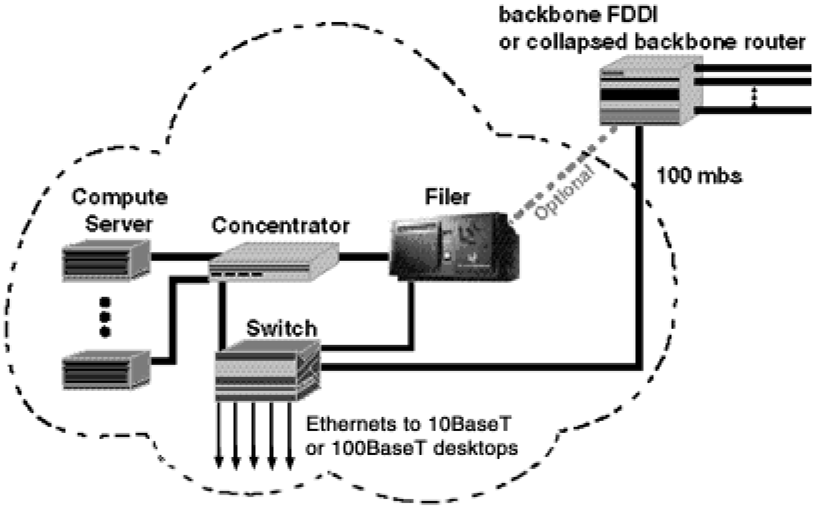


Figure 11

Localized data deployment might be considered under circumstances where a hierarchical approach might be less effective or unnecessarily costly.

- In general, the best possible performance can *only* be achieved by direct connectivity. Intermediary network hops add latency, and fluctuating network conditions introduce variability in file service response times which are notoriously counter-productive.
- Some types of applications, such as MCAD (Computer Aided Design for Mechanical engineering design and analysis), demand very high performance for reading and writing large files. Whereas a hierarchical approach works best when the data access is bursty(15), with MCAD applications the sustained large-file accesses would flood the shared, intermediate hierarchical layers, negatively impacting available bandwidth for other uses (including other MCAD-related I/O). Localizing file service in MCAD environments is almost always a practical necessity.
- When work units (workgroups or departments) within a given campus operate autonomously, a hierarchical strategy would provide for sharing of consolidated resources where no functional sharing requirements exist.
- Work units want local control over the resources essential to meeting their deadlines. Usually chief among essential resources is the data stored in files on the server. By storing it on a locally-controlled file server -- and *not* storing it centrally (one or more layers higher in the network hierarchy) on a shared file server -- the work unit can ensure

that their data will not be taken off-line "for the good of all" when someone in another department makes the global decision to upgrade that central server to the next software revision level, or otherwise make it temporarily unavailable. Local connectivity to the file server also implies independence from external network conditions, which otherwise might interfere with data accessibility or performance.

- Organizational budgets and politics can be such that the localized approach is favored. Individual projects often must independently justify and fund purchase of file server resources. It's only natural for buyers to want to control the equipment for which they've paid.
- Some companies belief that services for a group should be as local as possible sometimes derives from bad experiences with insufficiently fast paths between the central servers and network-remote workstations. Talk of fast networking fabric may not always reverse this belief; thus, localized service can be a long-lived bias.
- When groups are dispersed over large distances, since campus-to-campus WANs are not fast enough, localized data deployment (with some operational autonomy) is effectively mandatory.

In summary, localized data deployment can be used to improve performance, isolate non-bursty network traffic, simplify networks, and address political or financial issues in large organizations.

Hybrid approaches

The capacity and communications configurability of NetApp filers is such that it may be used at both extremes of the localized/hierarchical file server deployment spectrum. Furthermore, it is rare for a network's data deployment model to reflect a purely hierarchical or fully localized strategy. Figures 12 and 13 show the hybrid approaches taken at Western Digital(16) and Cirrus Logic(17), respectively.

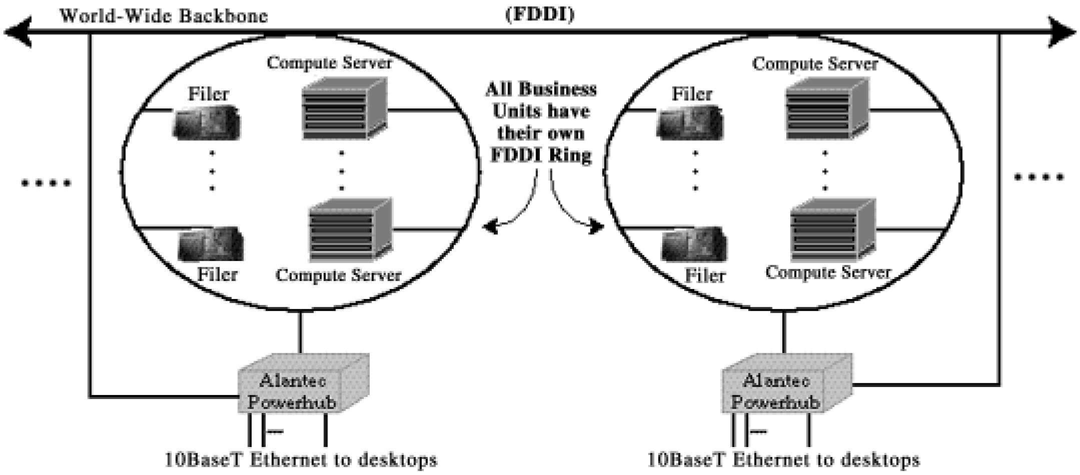


Figure 12: Western Digital

Western Digital implemented an essentially hierarchical approach, wherein central FDDI rings, each with multiple filers and compute servers, drive desktops indirectly through Alantec Powerhubs. One might call theirs a "localized hierarchical" approach in that each group's resources are localized within a one-layer hierarchical topology.

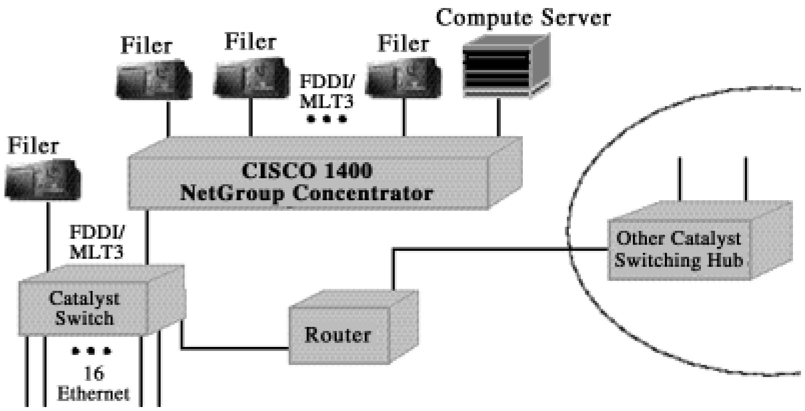


Figure 13: Cirrus Logic

Cirrus Logic uses a combination of switched-Ethernet and FDDI workgroup- and project-affiliated "clumps" of logical subnets. Data resides both at the workgroup (switch level) and also at the departmental (concentrator) level. Note how a router is used to link workgroup switches as peers. We'd call this hierarchical, with some localization.

In general, almost any LAN of significant size will exhibit some aspects of complexity. File server deployment decisions must be consistent with the objectives of the organization and the capabilities of the network. Hybrid strategies with localization for optimized performance and hierarchical placement for accessibility are currently more the rule than the exception.

Conclusions

Network computing in the '90s depends heavily on the data infrastructure. Scalability is the watchword. As large sites evolve their networks, two trends in file server deployment have emerged:

- consolidation of data storage for widely-shared *common* data placed near the top of the network hierarchy; and
- greater distribution of *local* data storage at the department or workgroup level.

In both cases, the functional goals(18) are the same -- performance and scalability. NetApp filers offer both the high-end performance and the configurability necessary to place data wherever it can be most effectively accessed and managed.

Data deployment methods range across a spectrum bracketed by the purely hierarchical approach at one extreme to the fully localized at the other. In practice, most networks borrow from both philosophies. NetApp filers can be sized to support both deployment paradigms and any blend of the two. For example, relatively small filers (e.g., the F220) might be distributed at the department and workgroup levels, while larger campus- or enterprise-scale NetApp filers (e.g., the F330 or F540) can be used at the core of the network.

Filers work together with high-end networking products like switches to feed data-hungry workstations and compute servers at the lowest possible latencies, and with the greatest opportunity for scalability. The synergy with switching technology is especially strong because NetApp filers can easily utilize multiple interfaces per switch (something most conventional file servers cannot easily do, if at all). Filers can be paired individually with switches, or configured with high-bandwidth connections into multiple switches, for extremely cost-effective and scalable distributed deployment.

Accommodating growth while providing optimal data access and facilitating responsible data management can overrun budgets and stress support personnel. NetApp filers, by virtue of their fast, simple, and reliable appliance nature, address these challenges and help contain runaway systems administrative overhead costs. The comprehensive range of performance, capacity, and connectivity options offered by Network Appliance provide solutions at all levels in the network hierarchy, and directly address the performance and scalability objectives of a modern data infrastructure.

Footnotes

1. The ISO (International Standards Organization) developed the seven-layer OSI (Open Systems Interconnect) Reference Model to describe communication between network devices. The model is explained in virtually every networking reference text, and so is not covered in this paper.
2. "Response time" refers to the round-trip elapsed time between a client's request and a server's reply.
3. UTP (unshielded twisted pair) cabling terminates in modular RJ-11 or RJ-45 connectors. "Category 5" UTP cables are extensible beyond 10BaseT (standard 10-Mbps Ethernet over UTP), and can be used for 100BaseT (100-Mbps "Fast Ethernet"), FDDI-TP (also known as "CDDI" -- copper-based 100-Mbps FDDI), or ATM at OC-3 data rates (155 Mbps).
4. All references to "FDDI" in this paper imply *either* FDDI (fiber) or FDDI-TP (copper), unless indicated otherwise.
5. Switches, being newer-generation technology, are of course faster than their ancestors, the bridges of yesteryear.
6. This is reflected in Network Appliance's own shipments. 50% of all NetApp filers are shipped with FDDI added on, whereas 20-25% of total units have one or more optional 100BaseT interface cards beyond the standard Fast Ethernet connection provided on the motherboard.
7. For voice and video applications, it is difficult or impossible to gracefully compensate for sporadic pauses or delays in data transmissions. The first 100 ms of transmitted content cannot be presented to the user together with the second 100 ms of content, and so if it fails to arrive on time it is essentially lost, creating a gap which causes the image or sound to "stutter" or "jitter" or otherwise produce undesirable visual or audible artifacts.
8. Most notably, the ATM Forum (jointly founded in 1991 by Cisco Systems, NET/ADAPTIVE, Northern Telecom, and Sprint) and the International Telecom Union.
9. Whenever packets must be forwarded between subnets (usually by a router) the latency incurred counts as one "hop". The propagation time between any two points on a network is, in large part, a function of the number of hops. Optimal routes usually have fewer hops.
10. In 1990, if an NFS file server's overall average response time was under 100 ms, it was considered acceptable. This threshold soon dropped to 70 ms by 1992, 50 ms by 1993, and in 1996 many sites expect (and routinely achieve) faster-than-10 ms average response times from their NetApp filers.
11. A "multi-homed" server has multiple network interfaces (typically for multiple IP addresses).
12. Fast Ethernet still has this constraint, of course. However, *full duplex* 100BaseT, a variant supported by some Fast Ethernet devices, completely eliminates Ethernet deference by using separate 100-Mbps pathways for incoming and outbound traffic.
13. An FDDI packet carries a data payload of ~4500 bytes, compared to only ~1500 bytes for Ethernet. This means that an 8-KB NFS Read or Write block can be sent over only two FDDI packets instead of six for Ethernet. Of course, this advantage only applies if the

client is also connected to the switch with FDDI. Otherwise, the larger packets from the server will need to be fragmented when forwarded to the client's smaller-MTU pipe.

14. NetApp Customer Profile 4003, Defense Industrial Supply Center (Defense Logistics Agency), describes a hierarchical topology wherein about 1800 clients access two NetApp filers.
15. Many nodes out on branches and twigs of the network tree can share the larger branches and trunks if data access patterns are bursty -- infrequent and relatively brief events, providing sufficient opportunity to accommodate other users' similarly-sporadic network traffic.
16. See also NetApp Customer Profile 4001, "Western Digital Corporation."
17. See also NetApp Customer Profile 4002, "Cirrus Logic, Inc."
18. The other aspects of network data access (e.g., reliability, data integrity, etc.) are still critically important, of course, but these are *intrinsic* to the file servers themselves and not directly related to the *extrinsic* issues of network topology.

Network Appliance, Inc.
495 East Java Drive
Sunnyvale, CA 94089
www.netapp.com