Realtime
publishers
"Leading the Conversation"

The Shortcut Guide™ To

Optimized WAN Application Delivery

sponsored by

Blue Coat®

Ed Tittel

Copyright Statement

Chapter 2: Managing Scale and Scope in a Distributed WAN Environment

At the very core of every network infrastructure is the routing process. Network cabling and media, in all its various forms, creates the veins and arteries for the network, where routing is a dynamic internal process through which data travels around a system composed of intermediary devices and connection endpoints. Routing manages the communications path selection process in computer and telecommunications networks—and is a component function in all such networks—and determines when and where to deliver data-bearing traffic. Routing involves the direct forwarding of data packets in packet-switched networks to designated endpoint addresses through intermediary devices known as routers, bridges, switches, firewalls, and gateways.

✎ This chapter uses the term *routing* in a broadly defined, generally applicable way. This usage is entirely different from the more specific term *router*, which is effectively a TCP/IP Layer 3 device. Request for Comment (RFC) 1983 defines routing as "The process of selecting the correct interface and next hop for a packet being forwarded." That's really what this guide is all about—finding and using the next best hop to ensure secure, timely, and/or qualitative delivery of network data, and optimizing traffic across hops that involve wide area network (WAN) links.

The science of routing is the process of identifying connective pathways along which to deliver data between subnets or external network sources, using a variety of logical and algorithmic techniques. It is the directional flow of datagram or packet traffic from source to destination according to some defined passageway that is typically specified through administratively managed memory-resident routing tables. A router selects the correct interface from its available routing table and determines the next hop along which to forward a packet. Similar network address structures (closely related numeric values) imply proximity within a network, even for WAN-spanning connections. The process of accessing the Internet through a WAN connection is depicted in Figure 2.1.

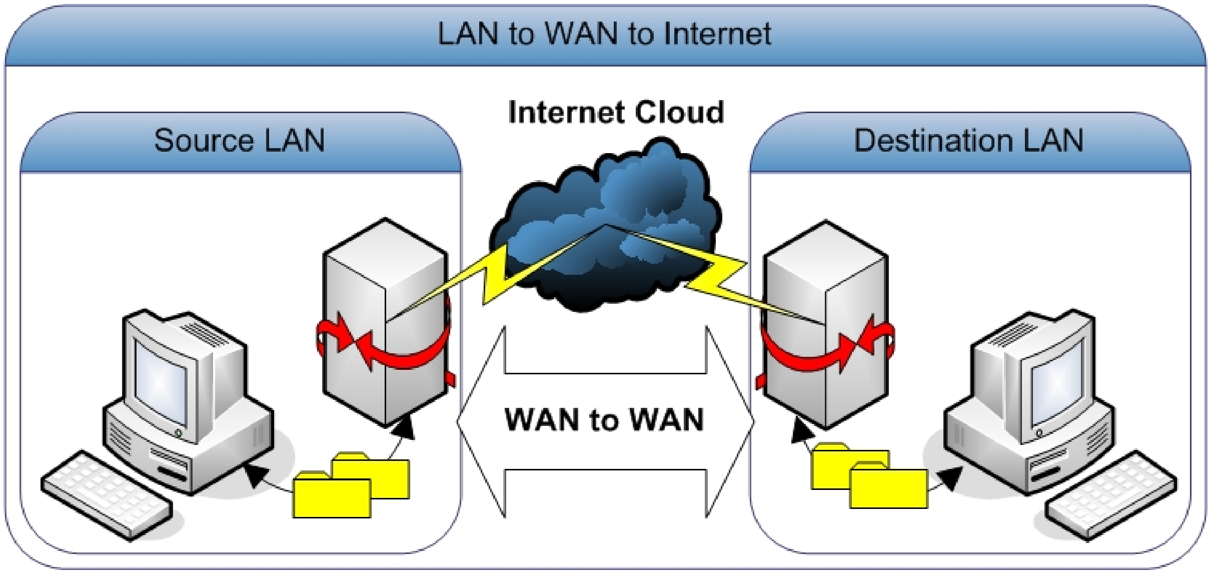

Figure 2.1: When moving a packet across a WAN link, the router picks it up from some internal interface, then forwards it out an external interface, which typically delivers the packet into an "Internet cloud." At the destination side, the same packet eventually arrives at the router's external interface for forwarding into an internal LAN.

Lessons from the Routing World

Routing doesn't make the world go round, but it does help telecommuters get around. Much of its operation is transparent and unknown to the majority in the networked working world, the very nature of which is lost on the non-technically minded. However, routing is still universally understood to be a necessary and vital process to establish a connection in the computing world, and to stay that way. There are many valuable lessons to be learned and worthwhile observations to be made from past experiences and present encounters with routing applications, services, and technologies, even where WAN optimization is concerned.

Routers themselves create broadcast domains that enable neighboring equipment to identify unknown hosts within the boundaries of the network perimeter. Broadcast traffic reaches all connected hosts, and by virtue of this fact, bandwidth utilization can get out of hand quickly within large-scale networks. Overly chatty protocols such as the Address Resolution Protocol (ARP) and Reverse ARP (RARP) are present in TCP/IP network environments wherever IP addresses are mapped to Media Access Control (MAC) addresses. ARP and RARP broadcast "who has" queries to elicit "is at" responses to identify these IP and MAC pairings (or lack thereof) for the purposes of bootstrapping network-loading operating systems (OSs) to initialize appliances and devices that are network-aware and to manage routing and special handling for low-level, locally connected client interfaces. This explains why such traffic is usually restricted to specific access domains and is almost always a purely local (rather than wide-area) form of interaction.

From a processing resource perspective, routing grows exponentially complex particularly in large networks owing to the number of potential intermediate destinations a packet may traverse before reaching its final destination. Routers manage information about paths that enable packet-based messages to reach their intended recipients, forwarding units of digital information along a particular path designated in headers and defined by parameters (protocol fields) contained within the message. In a much broader sense of the term, routing may also include the translation of such messages between Local Area Network (LAN) segments that utilize different Link Layer Control (LLC) protocols.

✎ A packet is the basic unit on any TCP/IP-based packet-switched pathway. It is a formatted block of information that includes protocol fields, headers, trailers, and optional payloads. Protocol properties parameterize how a packet is to be handled and delivered. They also include putative identities (in the form of IP addresses) for both sender and recipient stations, error-control information, message payload, and optional routing characteristics, all of which we discuss in more detail later in this chapter.

A packet can be a complete unit in itself or part of some larger ongoing communication between endpoints. Computer communications links that do not support packets, such as traditional point-to-point (PPP) telecommunications links, simply transmit data as a series or stream of bytes, characters, or bits. TCP/IP networks handle such links with relative ease by providing reversible encodings to enable them to be transited using native formats, then retransformed back into packet-based traffic on the other side of such links. Also, TCP/IP networks chop up large data sequences into smaller packets for transmission and logically group data according to the DoD network reference model, which creates four layers populated with various protocol definitions.

Imagine a router is the mail room for a busy postal clerk who’s constantly rushing deliverables between senders and recipients. Envision each packet as an envelope full of mail circulating the globe, and for many fleeting moments throughout his day, this busy mail clerk processes such items. Now consider that some mail has higher priority than others and is marked accordingly to reflect its status. That mail will be processed with more attention to delivery timeframes than other pieces of mail, so it may very well “jump the line” or receive other special handling along its way.

Also consider that certain pieces of mail are too big (in size, shape, or weight) to fit into a single envelope or reasonably large box, so its contents are broken into a larger number of smaller, simpler packages and elements, and sent in multiple bit and pieces. Perhaps some of these items are marked “fragile” or “one of many,” indicating other special handling or delivery considerations. In essence, these parcels specify some special handling characteristics that are dealt with by other post office personnel who may handle them at some point during their trip from sender to receiver. From a simplified perspective, this model is analogous to packet routing.

Alas, this is where the router-as-a-mailman analogy ends and a more accurate definition of routing prevails. The analogy breaks down because network routing is far more complex than what mail courier services encounter and endure. Packets possess a vast variety of protocol properties and parameters that influence their handling and delivery throughout the routing process, enough to swamp mere human minds but well within the primitive (but more calculating) capabilities of the kinds of computer “brains” present in modern high-speed networking gear (switches, routers, and so forth).

A router can itself be a computer or some functional equivalent that is used to interconnect two or more network segments. It operates at Layer 3 of the OSI reference model, routing traffic through network segments so as to move it toward the final destination to which it is addressed. A router accomplishes this task by interpreting the network (Layer 3) address of every packet it receives to make an algorithm-based decision about the next interface to which that packet must be delivered.

The pathways along which packets travel may be static or dynamic. Static routes use pathways that must be explored, negotiated, and then established before traffic can proceed across them, whereas dynamic routes are made and used as needed, in keeping with parameters presented within packets in motion or based on data included in connection requests that last only as long as they're needed. Either way, a router must keep up with changes to network topology, route availability, traffic conditions, and other factors that can influence if, when, and how quickly traffic moves across pathways accessible through specific interfaces. Figure 2.2 shows a simplified routing grid, with cost factors applied for paths to networks A through E.

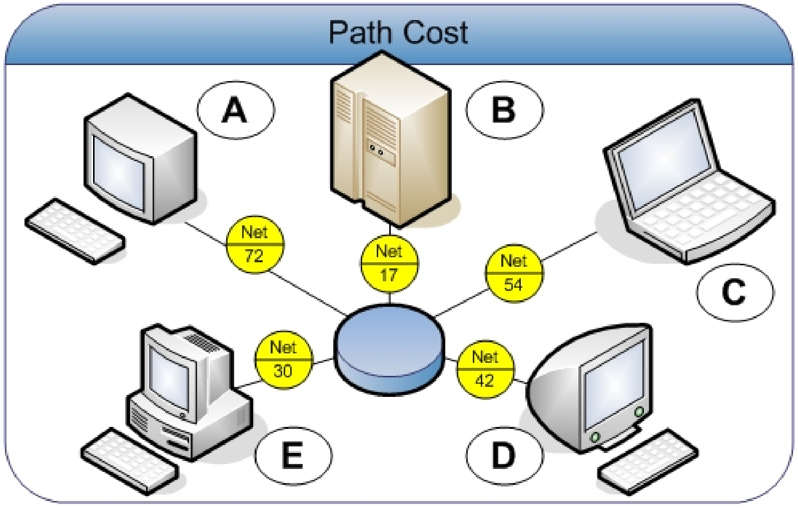

Figure 2.2: Routers must track and keep up with path cost factors to understand how to forward specific types of packets for transmission across the WAN, symbolized by the light blue cylinder at the picture's center.

Introduction of Local and Border Distinctions

Local and border distinctions define the edges of a given network perimeter, from the perspective that traffic is essentially bound or confined to some virtual circuit pathway or transmission space on its way across perimeter access points, whether inbound (from the WAN to the LAN) or outbound (from the LAN to the WAN). On local or private networks, packet transmission between recipients is tightly bound to a LAN topology. A LAN topology may span one or more LAN segments, which themselves reside within a network border that outlines the perimeter or outermost edge of a network's reach.

Network boundaries or borders are where a great deal of interesting activity typically occurs. This is as much a boundary that distinguishes between levels of knowledge (absolute knowledge inside the border, limited knowledge outside the border) as it does between levels of control (absolute control inside the border, limited or no control outside the border), cost (lower costs inside the border, higher outside), and speed (higher speeds inside the border, lower speeds outside—with differences sometimes as high as three or more orders of decimal magnitude). By no surprising coincidence, all these conditions also hold for WAN optimization, and likewise play a significant role in designing and implementing effective WAN optimization techniques. It's also the case that communications across the boundary, and over the WAN, is more costly and time-consuming than communications that stay within local LAN boundaries. This also explains why compression is so commonly applied to all WAN traffic, as a way of helping to limit communications costs and associated bandwidth consumption.

Within the border, network routers can behave more or less as they want to, and impose all kinds of control and management schemes on traffic as they see fit, including classes or quality of service controls to prioritize and manage traffic of different kinds and importance in different and appropriate ways. Within the border, enterprises can establish their own routing systems to establish and maintain whatever routing regimes they might want to use and change them over time as they see fit. They can even choose whatever kinds of routing protocols they like, and configure them any way they like.

Outside the network border, however, freedom of choice disappears, and flexibility is considerably diminished. Border routers must use whatever protocols are required in the exterior (outside the border) routing environment, and must adhere to whatever controls, tags, addresses, and so forth that the exterior environment decrees. Irrespective of whatever kinds of quality or class of service mechanisms may be in use inside the border, outbound traffic must map into whatever kinds of labels or tags are known to the exterior environment, and often means that quality and class of service information either becomes irrelevant or is reduced to two or perhaps three levels (fast and slow, or perhaps slow, middling, and faster, where three different levels are available).

Outside the border is also where big delays kick in (WAN links are invariably far slower than LAN links, and public pathways likewise slower than private ones, if only because of higher utilization and traffic volumes) and where traffic gets more expensive to move. This phenomenon helps to explain much of the appeal inherent to WAN optimization, and stems from reductions in traffic achieved through all sorts of clever techniques that include protocol proxies, caching, shared symbol and data dictionaries, and more.

📖 Throughout the remainder of this chapter, several references will be made to the concept of an *autonomous system* (AS)—a collection or collective group of IP networks and routers under control of a common administration with common routing policies. An official definition can be found in RFC 1930 at http://tools.ietf.org/html/rfc1930. An AS may reside inside the network boundary and operate within its borders. Anything outside the border is usually under somebody else's control, though that routing domain is probably also an AS. But exterior routing requires consensus to operate and adherence to common rules and requirements to use.

Introducing the Routing Information Protocol

Routing Information Protocol (RIP) was once commonly used on internal networks as an Interior Gateway Protocol (IGP) so that routers could dynamically adapt to changes to network connections via route advertisements. These advertisements communicate information regarding reachable networks and metric distances to those networks. Although RIP is still actively used to lesser extents in small-scale modern environments (fewer than a dozen routers, fewer than a thousand nodes), it has been rendered obsolete and surpassed by far superior routing protocols designed to behave well in large, complex networking environments (hundreds to thousands of routers and many, many thousands of nodes).

✎ IGP is a routing protocol used within an AS to determine reachability between endpoints within that system. In its distance-vector form, IGP identifies available pathways through advertisement of routing locations (in relation to other locations that likewise advertise themselves). When IGP uses a link-state-oriented protocol, each node possesses complete network topology information for all available pathways. Both *distance-vector* and *link-state* concepts are described shortly.

However, RIP quickly shows is crippling limitations within any sizable network environment. Chiefly among its inadequacies is a non-negotiable 15-hop limitation, which severely restricts the operational capacity and logical expanse of WAN topologies. RIP also cannot handle variable-length subnet masks (VLSM), a problem for an ever-shrinking IP address space. RIP routers also periodically advertise full routing tables that are a major unnecessary consumer of available bandwidth—another major blemish for WAN topologies. Convergence on RIP networks occurs slowly, with routers enduring a period of holding formation and garbage collection before expiring information that has not been recently received—also inappropriate for large-scale networks, particularly slow links and WAN clouds.

From a network management perspective, RIP possesses no concept of network delays and link costs, and therefore provides no resolution for these issues. Routing decisions are entirely hop count-based, even despite better aggregate link bandwidth or lower latency. Also problematic is the fact that RIP network topologies are uncharacteristically flat, with no concept of containment boundaries or logically divided areas. RIP networks fall drastically behind without Classless Inter-Domain Routing (CIDR) capability, the use of link aggregation or route summarization.

A second version, RIPv2, seeks to address several shortcomings and glaring omissions from its predecessor but still possesses the 15-hop limitation and slow convergence. Both of these properties are essential to support modern large-scale network environments. As is usually the case, technological innovation designed by human inspiration has a way of besting the most difficult of challenges. RIP also describes the most basic kind of operation that involves WAN optimization, in that it is most often applied between pairs of devices across a single, specific link, where the parties on each side of a WAN connection have thorough or exhaustive knowledge of everything they need to know about what’s on the “other side” of that WAN link. This might be viewed as a paragon of static routing, in that much of what WAN optimization devices can do depends on knowing the ins and outs of operations and characteristics on both sides of the WAN link, and of taking steps based on that knowledge to limit the use of that WAN link as much as such knowledge will permit.

Open Shortest Path First Algorithms

The Open Shortest Path First (OSPF) protocol is one of several hierarchical interior gateway protocols (IGPs) used for routing in IP between small to large networks, utilizing link-state data in the individual areas or routing domains that define the hierarchy. An algorithm-based computation calculates the shortest path tree inside each area and is perhaps the most widely used IGP in large enterprise networks. OSPF dynamically determines the best routing path for IP traffic over a TCP/IP network and is designed to generate less router update traffic than is required by the RIP format it replaces. By design, OSPF also incorporates least-cost, equal-cost, and load-balancing capabilities. Unlike RIP, OSPF incorporates cost or preference information when selecting the routes it will use from the routes it knows about. This kind of selective behavior is also typical for WAN optimization, which at some level is all about managing costs and limiting WAN access as much as possible, without impeding communication.

✎ What is meant by *link-state*? Consider a link as any interface on the WAN router. The state of that link describes the interface and its relationship to nearby routers; this state description includes its IP address, subnet mask, network connection type, interconnected routers, and so forth. Collectively, this information forms a link-state database, described later.

RIP is a distance-vector protocol, which means that it uses hop count to select the shortest route to a destination network. RIP always uses the lowest hop count despite the speed or reliability properties of its supplied network link. OSPF is a link-state protocol, meaning it can algorithmically consider a variety of link-related conditions when determining the best path to a network destination, including speed and reliability properties. Furthermore, OSPF has no hop limitation and routers can be added to the network as necessary making it highly suitable for highly scalable enterprise WAN environments.

OSPF also provides several other enhancements still outstanding from its predecessors, RIP versions 1 and 2. OSPF has unlimited hop count, VLSM capability, and uses IP multicast to send link-state updates as they occur to reduce network noise. Routing changes are propagated instantaneously, so OSPF has better convergence than RIP. OSPF allows for better load-balancing, enables the logical definition of networks (with routers divided into areas), and limits the delivery of link-state updates network-wide. Password-secured route authentication, external route tagging for AS, and aggregate routing are also advantages OSPF has over RIP.

✎ What is meant by *convergence*? From a network routing perspective, convergence is essentially the combination and merging of advertised routes and route updates from all available sources of such information (other routers). When we say that RIP converges more slowly than OSPF, that means it takes longer to propagate updates through a collection of RIP routers because each update goes through hold-off and garbage collection periods that timeout and delete stale information more slowly than is the case in OSPF.

A link-state database (LSDB) is constructed as a tree-image of the network topology and identical copies are periodically updated on all routers in each OSPF-aware area. Assigned areas in an OSPF model are designated numeric values that correspond to regions of an enterprise network, with each additional OSPF area directly or virtually connected to the backbone area.

OSPF is hierarchical, propagates changes quickly, and supports overlapping variable subnet masks (VLSMs) to enable multicasting within distinct network areas. This, too, makes OSPF more efficient than RIP and helps keep update traffic volumes down. After initialization, OSPF routers only advertise updates as routes change and never include an entire routing table in a single update (as RIP is wont to do). Also, OSPF areas may be logically segmented and summarized to reduce routing table sizes; the OSPF protocol remains an open standard unregulated by any single vendor. As such, OSPF is well-suited to WAN deployment where combinatorial network topologies may have no clear hierarchy, contain large numbers of routers, lack an efficient means for route update propagation, or utilize potentially conflicting subnetworks and masks. Overall, WAN overhead is lowered through more efficient delivery of routing tables, and traffic reduction strategies built-in to OSPF further help eliminate potential line noise and subsequent bandwidth waste.

That said, OSPF taxes processing resources heavily and maintains multiple copies of routing information. OSPF uses very little bandwidth where no network changes exist but will flood network devices after recovering from a power outage. It isn't perfect, but it's a perfectly functional model. The analogies to WAN optimization are also fairly strong, in that initial setup and cache population will tax such systems most heavily, just as resynchronization activities will impose the greatest overhead on communications between pairs of WAN optimization devices.

📖 The OSPF protocol format is specified in RFC 2328, which you can find by pointing your favorite online browser to http://www.ietf.org/rfc/rfc2328.txt.

Interior Gateway Routing Protocol

Interior Gateway Routing Protocol (IGRP) is a routing protocol developed at Cisco Systems in the mid-1980s, as the company sought to implement a robust and efficient routing protocol within autonomous systems. As a first attempt to improve upon and optimize overhead for RIP, and to add support for protocols other than TCP/IP (especially OSI connectionless network protocol or CLNP networks), IGRP combines a distance vector algorithm with a number of wide-ranging metrics for internetwork delay, available bandwidth, reliability, and network load to offer a more flexible and accommodating routing regime than RIP could deliver.

Metric values for reliability and load are eight-bit values that can accommodate numbers from 0 to 255. Bandwidth metrics can represent speeds from 1200 bits per second (bps) to 10 Gbps. Delay values are 24-bit numbers than can accommodate any value from 0 to 2^{24} – 1. IGRP also lets network administrators define constants they can use to influence route selection, where such values are hashed against these metrics, and each other, using a special IGRP routing algorithm to produce a single composite metric value. This lets administrators give high or lower weighting to specific metrics, and thereby, to fine-tune IGRP's route selection capabilities.

IGRP also supports multi-path routing, where bandwidth can be load balanced across multiple links, or where failure of a primary link automatically brings a backup link into use (a situation commonly described as failover). To keep update traffic and activity under control, IGRP also makes use of numerous timers and convergence features to limit update frequency, to decide how long to keep quiet routes for which no updates have been received active, and to determine how long to maintain routing table entries as they age over time. All of these characteristics provided significant improvements over RIP without switching to a link-state view of the networks that IGRP serves.

Over time, IGRP has been one of the most widely used and successful routing protocols. Cisco took great pains to preserve useful functions from RIP but greatly expanded IGRP's reach and capabilities. IGRP does lack support for VLSMs, which led to the introduction of Enhanced IGRP (and many switchovers to OSPF), in the early 1990s as pressure on IP address space and a desire to compress routing tables drove service providers, communications companies, and large organizations and corporations to abandon IGRP in favor of newer, still more capable routing protocols.

IGRP's numerous metrics and their wide representational capabilities provide a strong indication of what determined, well-designed programming can do to better represent network traffic needs and conditions. In much the same vein, WAN optimization devices also employ numerous complex metrics to characterize and manage WAN traffic and to reshape such traffic to better map LAN characteristics for WAN transmission and transport.

Enhancements Introduced via the Border Gateway Protocol

Border Gateway Protocol (BGP) is the core routing protocol for the Internet—thus, it is an exterior routing protocol used for informational exchanges between ASs. As an inter-autonomous path vector protocol, BGP operates by maintaining tables of IP networks or prefixes that designate network reachability (much like a telecommunications exchange) among ASs. BGP does not use traditional interior gateway metrics but rather bases its routing decisions according to pathway, network policies, and administrative rule sets. Currently, BGP exists in versions 2, 3, and 4.

BGP was intended to replace the now defunct Exterior Gateway Protocol (EGP). It now far outclasses and outnumbers original EGP installations. BGP provides a fully decentralized routing scheme, thereby enabling the Internet itself to operate as a truly decentralized system. It supports internal sessions—routes between routers in the same AS—and external sessions between routers from differing ASs. BGP can be used alongside OSPF where an autonomous system boundary router uses BGP as its exterior (Internet-facing) gateway protocol and OSPF as the IGP.

📖 BGP and OSPF interaction is all spelled out in RFC 1403—BGP OSPF Interaction. You can read up on this subject at http://www.ietf.org/rfc/rfc1403.txt.

BGP exchanges routing information for the Internet and acts as the adhesive protocol between Internet Service Providers (ISPs). Customer and client networks (such as university or corporate enterprise networks) will usually employ an IGP (such as RIP or OSPF, where the former suffices for small, simple networks and the latter becomes necessary for larger, more complex ones) for internal routing exchanges. These customers and client networks then connect to ISPs that use BGP to exchange customer/client and ISP routes. When BGP is utilized between Ass, the protocol is referred to as an External BGP (EBGP). When an ISP uses BGP to exchange routes within a single AS, it's called an Interior BGP (IBGP).

BGP is a robust, reliable, and scalable routing protocol capable of handling tens of thousands of routes via numerous route parameters called *attributes* that define routing policies and maintain a stable routing environment. Classless Inter-Domain Routing (CIDR) and route aggregation (to reduce routing table size) are two prominent features of BGP version 4, as widely used on the Internet. Route aggregation is a technique used to conserve address space and limit the amount of routing information that must be advertised to other routers. From a conceptual viewpoint, CIDR takes a block of contiguous class C addresses and represents them in an abbreviated and concatenated numerical form.

BGP offers capabilities and scale that goes well beyond current WAN optimization technology, which seldom scales to embrace systems by the thousands, let alone in larger numbers. Nevertheless, BGPs facilities at aggregating traffic, managing complex routes, and reducing addressing and traffic complexity have provided important models for WAN optimization techniques, albeit at a smaller scale.

Quality or Class of Service Labels and Related Queuing Disciplines

One common denominator for many of these WAN-worthy enterprise network protocols is traffic congestion control. For routers to route effectively, their pathways must be unobstructed and their presence must maintain a complete view of their operational domains. Surprisingly for some, routers don’t actually exercise much control over what network traffic they route because they do not ultimately originate such traffic. A router is merely a conduit through which traffic is conducted, and can only serve to influence the nature of its traffic flow. WAN optimization works in much the same way, where both routers and WAN optimization devices can impose controls on the traffic they see, according to predetermined rules or filters.

A router can drop traffic altogether, selectively accept or reject traffic, or place certain frames ahead of others as they queue up to access some particular interface. Between a cooperative set of hosts with similar specialized components, a router even can accept reservations and retain bandwidth for resource-intensive applications. However, all but the last method provide only primitive forms of traffic control. That last entry is in a league of its own that involves qualitative delivery of network service capacity with an absolute guarantee of minimum performance values.

Class of Service (COS) and Quality of Service (QoS) are two such strategies and include an ad hoc collection of technologies and techniques designed to designate priority values for certain types of traffic so as to do their best to ensure that minimal performance levels are achieved for higher priority types. Typically, the elements of QoS schemes are somewhat ad hoc in nature and often reflect only momentary or perhaps even idiosyncratic notions of priority and relative weight or importance. Because of this, QoS deployment can erect barriers for creating a true end-to-end strategy because applications, platforms, and vendors frequently differ in the techniques and technologies they use to define and assign QoS or CoS designations.

Furthermore, this creates great difficulty for IT groups seeking to deploy a consistent, seamless QoS solution across an enterprise WAN, which invariably spans multiple departments and sites and may even involve multiple sets of common carriers for long-haul links. Each department might rely on diverse sets of architectures, platforms, and software. Bringing all these elements into agreement through a common QoS approach can be as much of a chore as it is a challenge, and always involves lots of time, negotiation, and many drafts en route to a consensual and workable solution. Some of the different types of related queuing mechanisms used to prioritize traffic include the following (and remember that these mechanisms only have an impact when and as traffic starts piling up at some interface queue, waiting for its turn to be transmitted; only if multiple packets line up together for transmission can any kind of priority be imposed):

- Priority Queuing (PQ)—Traffic is prioritized according to a priority list and sorted into one of four (high, medium, normal, and low) priority queues.
- Custom Queuing (CQ)—Traffic is divided into several queues, one of which is serviced first (keep-alives, critical traffic) and the remaining traffic is serviced in a round-robin fashion
- Weighted Fair Queuing (WFQ)—Automatically sorts among traffic, capable of managing two-way data streams, with packets sorted in weighted order of arrival of the last bit

- Weighted Random Early Detection (WRED)—A congestion avoidance mechanism that uses TCP congestion control to drop packets randomly but does so based on IP precedence (this provides the weighting mechanism where higher-precedence packets are less likely to be dropped than lower-precedence ones), prior to periods of high congestion; such packet losses require their transmitters to decrease their transmission rates, normally to the point where all packets reach their destination, thereby indicating that congestion has cleared
- Weighted Round Robin (WRR)—A congestion avoidance mechanism that segregates traffic into various classes of queues and then grants access to the interface to each queue for a duration determined by the priority associated with its class; this avoids starvation problems that strict round-robin queuing can experience, where lower-ranked queues may be completely blocked from access to the interface as soon as higher-ranked queues begin to experience congestion (even though a low-ranked queue may enjoy only small, infrequent time slices for access, it will still get some access to an associated interface when WRR is used).

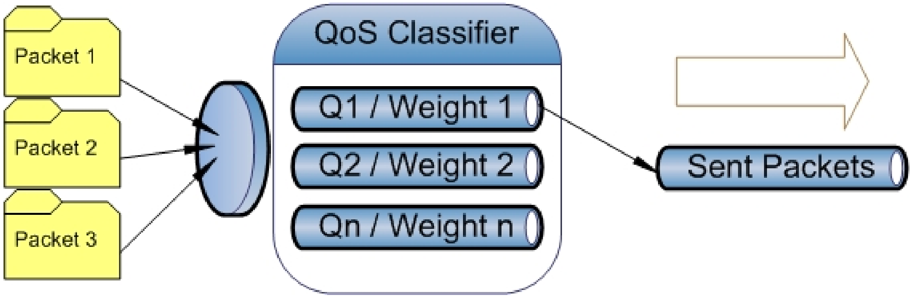

Figure 2.3: In general, queuing priority works by inspecting incoming packets for internal CoS or QoS identifiers, then depositing those packets into any of a number of priority queues. The ways in which queues are serviced, and how long each queue gains exclusive access to the attached network interface, determines how each of the preceding queuing disciplines is implemented.

Data packets are scheduled on the network through a series of queue service disciplines used to determine service priority, delay bounds, jitter bounds, and bandwidth allocation. Each queue is assigned a certain weight indicative of the amount of its guaranteed capacity. Among these choices, the Weighted Round Robin (WRR) technique may be mathematically proven to provide the most reasonable performance both in guaranteeing bandwidth and achieving fairness requirements. WRR, however, fails to accommodate some end-to-end delay requirements and jitter bounds, and thus may not be suitable for time-sensitive streaming traffic such as video or voice.

When discussing QoS, the terms *service priority*, *delay bounds*, *jitter bounds,* and *bandwidth allocation* all describe properties of queue service disciplines. *Service priority* is the precedence value given to specific application, service, or protocol traffic. *Delay bounds* specify predetermined operational latency values, whereas *jitter bounds* specify a predefined range of transmission signal variance. *Bandwidth allocation* is the amount of traffic (or range of signal frequency) provisioned on a given transmission medium.

QoS does confer an ability to apply priority levels for various applications, users, or data flows, and to guarantee a certain level of performance for a specific data flow. Individual requirements can be guaranteed for bit rates, delay, jitter, packet drop probability, and error rate. Such guarantees may be necessary where network capacity is insufficient to accommodate any and all traffic using best-effort delivery (no QoS, no priority) particularly for real-time streaming multimedia applications such as VoIP, IP-TV, or other fixed bit-rate, time-sensitive protocols. QoS mechanisms can be instrumental to improving performance anywhere network capacity is limited and multiple protocols are in use, particularly when some of that traffic takes precedence over the rest or where exceeding certain delay thresholds may make such traffic unusable or the user experience unacceptable.

Many branch office routers support various forms of QoS and will allow network administrators to apply traffic-shaping policies to network flows both inbound and outbound. This can help to ensure that business-critical applications perform acceptably as long as sufficient bandwidth is available to them.

Available tools to establish QoS between a service provider and a subscriber may include a contractual Service Level Agreement (SLA) that specifies guarantees for network or protocol performance, throughput, or latency values. These guarantees are typically based on mutually agreed upon measures and enforced through traffic prioritization.

📖 SLAs are discussed briefly in Chapter 1. For more information about SLAs in general, please visit the SLA Information Zone at http://www.sla-zone.co.uk/.

At this point, we've coursed through the evolution of network topology-aware protocols that work within defined parameters (or perimeters, if you prefer) of network boundaries. These protocols use their existing knowledge of network topology to make instantaneous decisions about how to handle packet transmissions, including when and where to make delivery. Such protocols can be encapsulated one within another, in fact, wrapped in layers of enveloping protocol data much like a set of nested Russian Matrioshka dolls. Ultimately, however, some outer layer tag, label, or value helps to set priority and instructs routers how to handle the contents whenever it encounters a non-empty queue for some network interface.

WAN Optimization techniques often prove surprisingly helpful as organizations seek to implement class or quality of service mechanisms for their network traffic. Of course, because such priorities weigh most heavily on traffic that crosses WAN links, there's a definite and beneficial synergy between QoS mechanisms and WAN optimization. On the one hand, QoS seeks to make sure that the most important and deserving traffic gets an appropriate share of WAN bandwidth and is subject to the lowest possible latencies. On the other hand, WAN optimization seeks to compress, compact, and reduce the amount of data that actually has to traverse WAN links between specific pairs of senders and receivers. Thus, WAN optimization often helps to impose and enforce all kinds of traffic policy, including class or quality of service, as well as providing the means whereby companies and organizations can make the most and best use out of WAN bandwidth made available to them.

We've covered the many traditional and time-honored protocols introduced to enhance routing performance using a number of techniques, tactics, and technological approaches. Let's transition into more modernized big-league protocols that significantly up the ante for routing gambles.

Historical Attempts at WAN Optimization

Before the advent of explicit WAN optimization technologies and tools, and devices designed specifically to support them, numerous technologies appeared that sought to deliver some of the same kinds of benefits as explicit WAN optimization, in many cases using tools or techniques that would also be employed in WAN optimization to good effect. We take a look at some of these in the sections that follow.

Compression on the Wire

Even early in the Internet era, hardware vendors realized that WAN bandwidth was more precious than LAN bandwidth, and took aggressive steps to compress traffic before shipping it across such links (where it could be decompressed at the other end before entering another LAN). Computer modem technology is a testament to benefits of compression, in that newer generations (V.34, V.44, V.90, V.92) use the same phone lines and underlying communications infrastructure to achieve ever-improving analog bandwidth of up to 56 Kbps thanks mostly to ever-better and more capable compression algorithms. The same holds true for the hardware used to establish most types of WAN links, where compression is likewise an integral part of data communications using such gear.

WAN optimization seeks to take compression several steps further. For one thing, it can take a more logical view of data, and perform symbol substitutions before compression is applied for sending, and use them after a received message is decompressed. The potential savings in bandwidth can be enormous, as when multi-megabit or even multi-gigabit objects or files can be replaced with symbol references that are at most 64 kilobits in size. Also, WAN optimization devices can make use of whatever the state of the art for compression hardware happens to be at the time they're built, and can also apply encryption/decryption at the same time providing proper keys are available to make such transforms.

Protocol Tuning, Tunneling, or Replacement

The Transmission Control Protocol, or TCP, is the workhorse at the TCP/IP transport layer. It offers reliable, robust delivery but also requires acknowledgements for transmissions received, and includes numerous mechanisms (windowing, congestion management, slow start, and so forth) to manage its own performance and latency characteristics. Much has been said about, and much can be made of TCP's tuning mechanisms. But one enduring and valuable way for WAN acceleration and optimization to occur is for TCP traffic to be repackaged inside UDP packets for WAN transport, thereby foregoing window management, acknowledgements, and other reliability and robustness mechanisms altogether, often by "spoofing" necessary TCP behavior on either end of a WAN connection, and sometimes by rearchitecting applications to replace TCP with UDP (usually along with built-in reliability and robustness mechanisms higher in the protocol stack, or in the application itself, to compensate for the loss of functionality that TCP delivers). These techniques have been used for some time to improve WAN performance and reduce latency, and remain just as applicable to WAN optimization as ever.

Caching for Web Access

Web caching, as this technique is sometimes known, seeks to reduce the distance between the users who request Web documents and the servers that supply them, based on the reduction in latency that proximity delivers. In fact, the closer the server, the lower the latency. Essentially, a web cache sits at some location between its users and a primary server, where the closer such a cache server sits to the users, the lower the latency they will experience when requests for documents may be satisfied from its cache.

The cache maintains a set of copies of documents that users request through it, so that any subsequent request for a document in the cache may be satisfied directly and more immediately. Users will still experience the latency associated with accessing the primary server each time they ask for something that's not in the cache server's stores, and maintaining cache currency will impose some overhead on access. But for reducing latency on frequently-accessed information, a cache server can deliver impressive improvements.

Modern-day WAN optimization devices make heavy use of caching to achieve often impressive latency reductions. Because site-to-site traffic often involves repeated access to a common set of documents, in fact, WAN optimization devices often deliver better latency improvements by using bigger, faster caches and by using cache dictionaries to exchange symbol references across WAN Links. Because cache symbols can point to cache entries at object and file levels, data reductions for WAN traffic of many orders of magnitude become possible.

Data Mirroring Boosts Web Access Speeds

Other historical approaches for speeding up Web access relied on mirroring servers at multiple locations, where again the guiding principle was to reduce end-user latency by minimizing the distance between the server and any user who wishes to access its contents. Mirroring is like a more proactive form of caching, in that instead of waiting for access requests to move data or documents from the primary server to a local cache, every time anything changes on the primary server that change is pushed to the mirrored servers, wherever they happen to be (and vice-versa, in a world where Web pages grow increasingly interactive).

The mirroring approach proves particularly useful for applications or services that use bandwidth-intensive content, such as multimedia (videos, movies, games, music, and other forms of streaming media). The best mirroring approaches also perform geographic analysis of network traffic patterns, in the never-ending quest to situate servers so as to keep end-user latency as low as cost factors will allow.

This approach plays into WAN optimization in that use of proactive and aggressive caching and copying techniques is a hallmark of both approaches, and where intense scrutiny of user behavior and traffic patterns helps to maintain local copies of information most likely to be requested in the near future.

Wide Area File Services (WAFS)

Although WAFS was introduced in 2004, and thus follows the preceding two items by five years or more, it has already become a historical rather than a current approach to WAN optimization. Although virtually all of the early WAN optimization vendors offered some form of Wide Area File Services in their earliest implementations, the current state of WAN optimization has progressed beyond WAFS to include the more generic notions of Wide Areas Application Services (WAAS) or Wide Area Data Management (WADM) which supports storage area network (SAN) consolidation by providing users with real-time, read-write access to data centers where corporate applications and data repositories typically reside.

WAFS seeks to reduce latency and overcome bandwidth limitations by caching data center content at remote offices or sites, and incurring bandwidth and access use only when synchronizing caches across WAN links. Other typical optimization techniques used for WAFS include protocol substitution (where LAN-centric file system protocols such as CIFS or NFS are replaced behind the scenes by more WAN-friendly alternatives), hardware compression/decompression of traffic entering and exiting WAN links, and data deduplication techniques that make heavy use of symbol dictionaries at both file and object levels.

Modern WAN optimization techniques tend to do more of the same things that WAFS did, and to do them better. They often involve more aggressive and proactive caching, bigger caches with more RAM and high-speed RAID arrays for disk storage, and faster, more capable symbol dictionary creation and management capabilities.

Middle Mile Acceleration

In traditional networking or telecommunication terms, the middle mile is the WAN link that spans between a service provider and the Internet backbone, or that span between the servers that deliver broadband applications and the network core. The bandwidth required to service such sites is often called “backhaul” and refers to the number and type of network connections required to deliver the necessary aggregated bandwidth, usually measured in some number of T-1 connections per site. For mobile phone companies, for example, backhaul often represents 30% of their overall expenses, and provides a strong impetus for acceleration and optimization techniques.

Techniques to make the most of middle mile acceleration often involve situating servers at the edge of the Internet, thereby bringing them close to users whose locations force them through some particular edge point to obtain Internet access. Situating a server in the same place brings all of the benefits of caching and mirroring already discussed in earlier items in this section into play, and is particularly well-suited for streaming media of all kinds (video, TV, movies on demand, music, and so on).

WAN optimization technology uses these same principles to bring needed data or multimedia closer to its users, thereby limiting causes for spikes in utilization that can swamp WAN connections, and cause severe latency for all users who must share some particular WAN link. It’s entirely appropriate to view this kind of technology as a “load-leveling” approach to WAN bandwidth utilization because it uses careful analysis of demand to anticipate requests for specific materials, and makes sure they’ve been pushed across WAN links before they’re needed (and can then be accessed at the edge of the Internet or some equivalent communications infrastructure, rather than requiring its constant traversal).

What Conventional Routing Techniques Suggest for WAN Optimization

It's certainly a truism that WAN latencies and communications behaviors vary greatly from their LAN counterparts and primarily in scope and extent. Simply put, anything that happens quickly or involves low latency on a LAN is likely to happen more slowly and involve longer latency (sometimes, by two or more orders of magnitude) on a WAN. This explains why one of the primary characteristics of WAN optimization involves techniques designed to reduce the number of packets that must flow across a WAN link as well as techniques to reduce the frequency of communication and the volume of data that must actually be transported from one end of a WAN link to the other (and vice versa, when response follows request as is invariably the case).

To that end, WAN optimization makes use of a variety of techniques to cut down on WAN traffic:

- Protocol substitution—WAN optimization devices often elect to manage "chatty" protocol sessions locally on the LAN, then encapsulate and manage related communications across the WAN using different, less chatty protocols instead (where they can also consolidate multiple short messages into one or more longer ones). This offers the benefits of reducing the number and frequency of messages that must traverse a WAN link and enables designers to choose and use WAN-friendly message structures and information transfers instead.
- Data substitution and caching—WAN optimization can inspect packet payloads at the application layer, looking for recurring strings or data elements. Once these are copied across the WAN link, as long as they don't change thereafter, pairs of WAN optimization devices can exchange index values or pointers to arbitrarily long strings or data elements, and reduce the amount of data that must flow across the WAN link by many orders of magnitude. In general, this technique type lowers data volumes by 30 to 60% across the board.
- Data compression (and encryption)— To outgoing traffic already subjected to protocol substitution as well as data substitution and caching, WAN optimization can apply a final "squeeze play" prior to transmission and usually employs special-purpose high-speed hardware to perform this task. This process may also include an encryption step to render packets in transit as opaque to unauthorized third parties between sending and receiving ends as modern technology will allow.

In general, WAN optimization devices can be considered part of the equipment at the network boundary because they generally sit between the LAN on one side and the boundary router that connects to the WAN on the other. This lets them apply their techniques—including those just mentioned in the preceding paragraph—to traffic on its way out one end of the WAN connection, where another device on the other end of the connection reverses those changes (or carries on its side of a local conversation, where protocol substitution may be at work). This relationship is depicted in Figure 2.4, which shows that WAN optimization devices sit between the LAN side of a boundary router and the LAN itself.

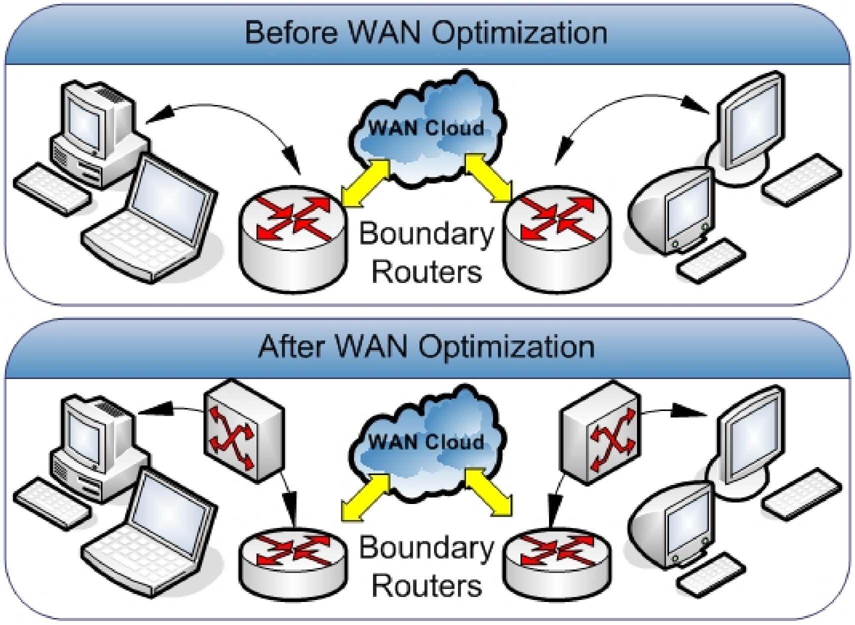

Figure 2.4: When installed, WAN optimization devices typically sit between the LAN and the boundary router (or most properly, on the stream of traffic destined for any WAN links inside the boundary router).

MPLS Introduces the Network Cloud

Then along came the Multi-Protocol Label Switching (MPLS) data-carrying mechanism for packet-switched networks. It straddles Layers 2 (data link) and 3 (network) of the OSI reference model and is designed to provide a unified data-carrying service for both circuit-switched and packet-switched clients that require datagram-driven service. MPLS can handle many types of network traffic including IP packets, along with frame formats native to Asynchronous Transfer Mode (ATM), Synchronous Optical Network (SONET), and Ethernet.

✎ *Circuit switching* is an early communications technology designed for analog-based phone networks modified to use digital circuit switching technology for dedicated connections between sender and receiver. *Packet switching* is a follow-up communications system that utilizes digital packets to transmit all forms of communications signals and serves as the primary method of communications for the Internet and other digital communications. A *datagram-driven* service is one where individual packets that comprise entire messages are sent individually across the transmission medium.

The original motivation for MPLS was to support construction of simple but extremely fast network switches so that IP packets could be forwarded as quickly as available high-speed technologies will permit. This approach keeps traffic continually on the move and requires little or no intermediate storage in slower queues where traffic must pause and wait for its allotted service interval. MPLS also supports multiple service models and can perform traffic management on the fly.

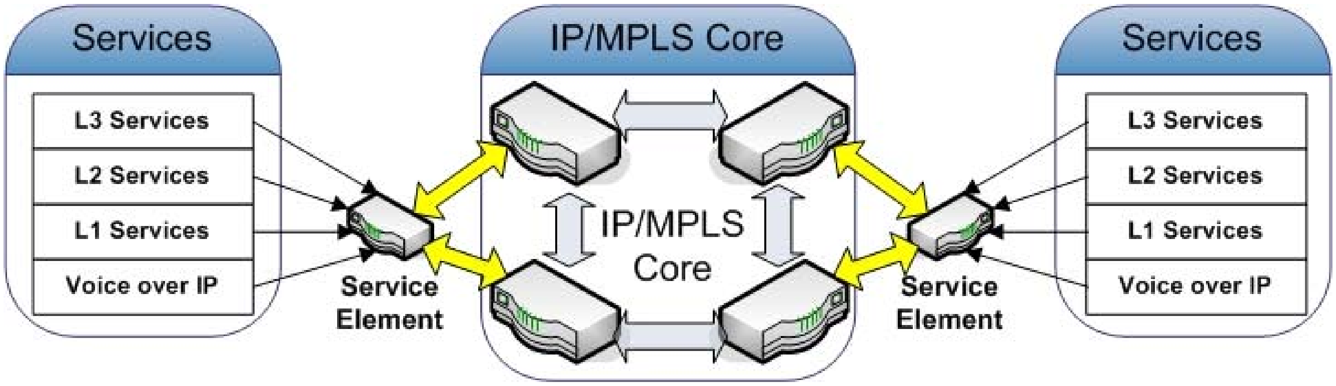

Figure 2.5: WAN optimization devices provide ingress and egress for services for Layer 1 to Layer 3 protocols.

Figure 2.5 shows how service elements (which might include boundary routers and WAN optimization devices) can provide ingress and egress for services at Layers 1 through 3 of the ISO/OSI model, along with access for streaming or time-sensitive services such as voice, video, and so forth. In an MPLS environment, traffic essentially flows from an ingress service element to some corresponding egress service element through an IP/MPLS core architecture where only MPLS labels need to be inspected and managed as traffic flows through a core network cloud. Here, the cloud analogy is a good one because IT professionals lose substantial visibility into and access to what is going on in the IP/MPLS core, but in exchange obtain better traffic management and much higher transit times through that core.

MPLS prefixes packets that enter the cloud at any egress point with an MPLS header, which contains one or more 32-bit MPLS label fields (because multiple labels may be affixed, this data structure is called a label stack). Each label is constructed as follows:

- 20-bit label value
- 3-bit QoS field (actually this is better described as a prioritized CoS scheme, though this flag is still called QoS)
- 1-bit bottom of stack flag (if set, indicates the current label is the bottom of the stack)
- 8-bit time to live (TTL) field

MPLS-labeled packets can be switched from an incoming to an outgoing port based on a simple label lookup rather than requiring a lookup into a routing table for IP addresses (a more complex, compute-intensive operation). Such lookups can be performed while the packet is moving through a switch fabric rather than requiring the attention of a separate CPU. Entry and exist points for MPLS networks are called Label Edge Routers (LERs). These devices push MPLS labels onto packets as they enter the cloud, then strip them off when they leave the cloud. In the core, routers that forward traffic purely on the basis of the MPLS label are called Label Switch Routers (LSRs), though an LSR may push a second (or additional) label onto a packet with an MPLS label from an LER already affixed.

Labels are distributed among LERs and LSRs using a special Label Distribution Protocol (LDP). LSRs in MPLS networks periodically exchange label and reachability data according to standard algorithms to permit them to manage a complete map of the network paths they may use to forward packets according to their labels. When a labeled MPLS packet hops from one MPLS router to another, it is said to be traversing an MPLS tunnel. Label Switch Paths (LSPs) may also be configured in an MPLS network to support network-based IP virtual private networks (IP VPNs) or to move traffic across specific paths in the network. In many ways, LSPs resemble permanent virtual circuits (PVCs) in Frame Relay or ATM networks, although they do not require specific Layer 2 technologies to be at their disposal.

When an unlabeled packet enters an LER to transit the MPLS cloud, the LER determines that packet's forwarding equivalence class (FEC) and pushes one or more labels onto the packet's freshly created label stack. This is also where QoS/CoS regimes may be applied so as to expedite high-priority traffic. Once the label stack is complete, the LER passes the packet onto the next hop router. When an MPLS router receives a labeled packet, the topmost label in the stack is examined. Depending on its contents, one of the following operations will be performed:

- Swap—The topmost label is switched out for a new label, and the packet gets forwarded along the associated path for that label.
- Push—A new label is pushed on top of the stack, on top of the existing label, thereby encapsulating that packet inside another layer of MPLS information. This technique supports hierarchical routing for MPLS packets, and explains how MPLS VPNs operate within the MPLS cloud (the core sees only relevant path information, and only VPN service routers deal with private traffic data).
- Pop—The topmost label is removed from the label stack, which may reveal another label beneath it (when this occurs, it is called decapsulation). If it is the bottom label in the stack, the packet will no longer be traversing an MPLS tunnel on its next hop and is leaving the MPLS cloud behind. Perforce this step is usually handled at an egress router (LER). Sometimes when an LER handles many MPLS tunnels, the MPLS router one hop ahead of the LER may pop the final label(s) to relieve the processing involved in cleaning up the label stack.

Although MPLS traffic remains in the cloud, the contents of such packets is completely ignored, except for the contents of the MPLS label stack. Even then, transit routers (LSRs) typically only work with the label at the top of the stack, and forwarding occurs based on label content only. This explains how MPLS operates independently of other routing protocols and the routing tables they require as well as the well-known IP longest prefix match performed at each hop in a conventional IP router.

Successful implementation of QoS/CoS for MPLS depends on its ability to handle multiple services and to manage traffic priority and flow thanks to extremely quick label inspection and label stack operations. MPLS can be especially helpful when service providers or enterprises want to impose service level requirements for specific classes of service so that low-latency applications such as voice over IP (VoIP) or video teleconferencing can count on acceptable levels of latency and jitter for traffic on the move.

That said, MPLS carriers differ in the number of classes of service they offer (up to a maximum of 8 different classes, as dictated by the QoS field size). Specific features, service guarantees, and pricing for classes of service also differ from carrier to carrier.

Lessons Learned for Optimizing WAN Application Access

WAN optimization works very well with MPLS environments where CoS labels may be invoked to speed priority traffic on its way. The MPLS label method also makes an excellent metaphor for the kinds of speedups that protocol substitution and reduced message frequency techniques provide for WAN optimization—that is, they permit traffic to be expedited and moved without requiring intermediate devices to dig too deeply into packet structures or to invoke intelligence about how higher-layer protocols work and behave. Once traffic is properly packaged, be it either in the MPLS or the WAN optimization context, all that is then needed is to speed it on its way from sender to receiver (where egress devices at the other end restore that traffic to its pristine original state).

Mechanisms for Speeding/Prioritizing Delivery

CoS or QoS data permits time-sensitive traffic to transit the WAN (which may apply to the MPLS cloud or at other steps along the way) more quickly than it otherwise might. But WAN optimization devices can also inspect and rearrange outgoing traffic to push higher-priority applications and services to the front of the line simply by recognizing protocols, services, or source/destination addresses involved. This only adds to the overall capabilities to provide (and honor) service guarantees or to meet SLAs.

Methods to Keep Continued Communications Moving

In the MPLS environment, a tunnel can be maintained as long as data flow persists across some particular path (or between specific ingress and egress ports at the edge of the MPLS cloud). Once the tunnel is set up and corresponding label and path information has been specified, it becomes extremely quick and efficient to keep related data moving through it. In a similar fashion, WAN optimization devices can maintain and use state information about existing (and persisting) connections between senders and receivers to keep application data moving smoothly between them. Session establishment and tear-down may involve certain delays, but as long as an ongoing session remains active, it’s very easy to move data between the parties to that session.

Managing Application Flow to Maximize WAN Throughput (NetFlow Models)

Properly equipped routers can also monitor and report on data flows between communication partners or peers. Network flows, which often fall within the Cisco-defined NetFlow rubric, are conceptualized as a unidirectional sequence of packets, all of which share at least five common values:

- Source IP address
- Destination IP address
- Source UDP or TCP port number
- Destination UDP or TCP port number
- IP protocol

Routers maintain information about duration and volume of data flows and emit flow records when inactivity or explicit flow termination occurs (TCP makes this easy with explicit connection establishment and closing behaviors, but timers must be set and watched for those connections that do not close normally or explicitly). Analysis of network flow data permits construction of a comprehensive view of traffic by type, volume, duration, and even security implication or events. In much the same way, monitoring and management of traffic through a WAN optimization device supports ready characterization and volume analysis but also supports application of acceptable use and security policies. Not only does WAN optimization compress and reduce desirable network traffic, it also provides a ready foil for unwanted or unauthorized network traffic, protocols, and services.

Proxies Enable Inspection, Analysis, Control, and Short-Circuiting

In computing terms, a proxy server is any computer system or application instance (as in virtualized contexts) placed anywhere on the network (at endpoints or intermediary points) that services the requests of its clients by directing requests to other servers. Conceptually, it is an intermediary device—physical or virtual—that sits between two communicating parties and provides some level of middleman service roles.

A client connects to the proxy agent, requests some server or service (that is, a remote server connection, a file, or resource) and the proxy handles the request from the target server on behalf of the client. The proxy server may—at the administrator's discretion—provide file- or page-caching, content filtration, secure transmission (for example, SSL, TLS, S/MIME) or policy enforcement among many other things. A proxy that passes all requests back to the client unmodified is a *gateway* or *tunneling proxy*.

Having an intermediary device acting in direct control over client-server traffic affords administrators several unique advantages. Proxies enable inspection of traffic flow, analysis of traffic patterns according to payload or protocol, explicit traffic control, and short-circuiting of unapproved traffic. In many cases, a proxy server is a protocol accelerator or response-time enhancer, such as SSL/TLS and Web caching proxy. Content-filtering proxies concentrate Internet traffic into a chokepoint-like sieve and apply administratively-controlled content delivery as defined by an Acceptable Use Policy (AUP).

Proxies Insert Themselves Directly into Traffic Flow

Proxies are deployed between internal end points and external destinations where they can directly influence network traffic flow, giving them a distinct vantage point on and control over client-server transmissions. Thus, the proxy can identify the types of traffic moving through it and can prioritize some traffic over other traffic. In addition, the proxy can police unwanted or unauthorized traffic.

> ☞ Where such traffic isn't blocked completely, it may prove useful to limit bandwidth to some ridiculously small value—for example, 1 Kbps. Doing so will keep connections active long enough for administrators to document them and, if necessary, drop in on offenders to remind them about acceptable use policy requirements and possible repercussions for its violation.

A transparent or intercepting proxy is used in combination with a gateway within an enterprise network setting to coerce client browsers through the proxy chokepoint, often invisibly to the client. Client connections are redirected from the intended gateway to the proxy without client-side configuration or knowledge and are used to prevent flouting or avoidance of administratively defined acceptable use policy. The transparency value originates from the manner in which the proxy operates with the client completely unaware of its presence.

Proxies Are Application-Sensitive and Offer Extensive Controls

Unlike firewalls, proxies are application-sensitive and offer extensive controls over application workloads across the network—providing complementary functionality to most firewalls. Although a firewall understands basic protocols and rules regarding their usage, a proxy is more application-aware in that it can actually distinguish and differentiate in the data beyond the IP or protocol header. A proxy can, in fact, dig deeply into the payloads enclosed within those protocols.

This ability to employ application- or service-specific intelligence gives the proxy the ability to apply rule bases to all kinds of application activities. Authorized applications or services may be associated with authorized source and destination addresses, data types, and so forth, so that they remain useful for regular workaday activities but useless for other outside activities. Thus, a secure file transfer application could proceed to copy or obtain financial and personnel data from specific servers, but users who attempt to use the same tools to download videos or images from other, perhaps less savory, servers will find themselves prevented from doing so.

Proxies Can Mitigate or Fix Protocols and Manage Transactions

Certain protocols have resource implications that aren't always readily apparent or immediately significant. Security protocols that utilize encryption routines tend to consume resources in a very dynamic and sometimes edacious manner. Largely, this remains imperceptible and goes undetected until other applications occupying the same general-purpose resource space usurp it to near-full capacity. In this moment, all resource-intensive heavy-hitters become illuminated and systems become unresponsive and at worst irrecoverable.

Particular proxy platforms accelerate encryption-related network protocols such as IP Security (IPSec) and SSL/TLS by offloading the processing burden to dedicated hardware. Though not considerable as a proxy in this context, such offload engines are present in PCI-based network cards that even provide a fully functional TCP/IP network stack. In addition to relieving general-purpose resources of network-oriented encryption processing tasks, proxies that facilitate encryption offloading provide security where it doesn't already exist—in the applications or platform—in a full-coverage manner.

From an administrative viewpoint, intermediary proxy devices also make the full range of network activity visible to network support personnel. This data can be approached statistically to yield all kinds of information about general employee activities, productivity, and time allocation. But it can also be approached on an event-triggered basis to signal possible security breaches, violations of acceptable use policy, or unwanted or unauthorized access to files and services to provide information about (and, if necessary, legally admissible evidence) such things as they occur and through logging or auditing after the fact.

WAN Optimization Extends Many Opportunities

By filtering and prioritizing traffic before it hits the WAN, then compressing, encrypting, and reducing all traffic that actually transits the WAN, WAN optimization supports careful conservation of a scarce and often expensive resource. Though the optimization applied may obscure the traffic that actually traverses the WAN, careful inspection and characterization of what goes into a WAN optimization device on the sending end, and equal attention to what comes out of such a device on the receiving end, can provide a wealth of information about how networks are used (and how well they're used) within most businesses and organizations. Certainly, the ability to limit bandwidth consumption and control costs helps explain the value that WAN optimization adds to the bottom line. But the visibility into activity and the opportunity for consistent, centralized management and control over network use and activity also explains why WAN optimization offers more than financial incentives to its users.

This concludes our discussion of managing scale and scope in a WAN environment. In the next chapter, you will learn more about the details involved in WAN optimization tools and techniques, as we explore traditional, legacy approaches to this issue, as well as modern, state-of-the-art WAN acceleration techniques.