MINT709 CAPSTONE Project Report

Performance comparisons of Internetwork Protocols

Prepared by: Haroun Yussuf

Prepared for Dr. M.H. Macgregor

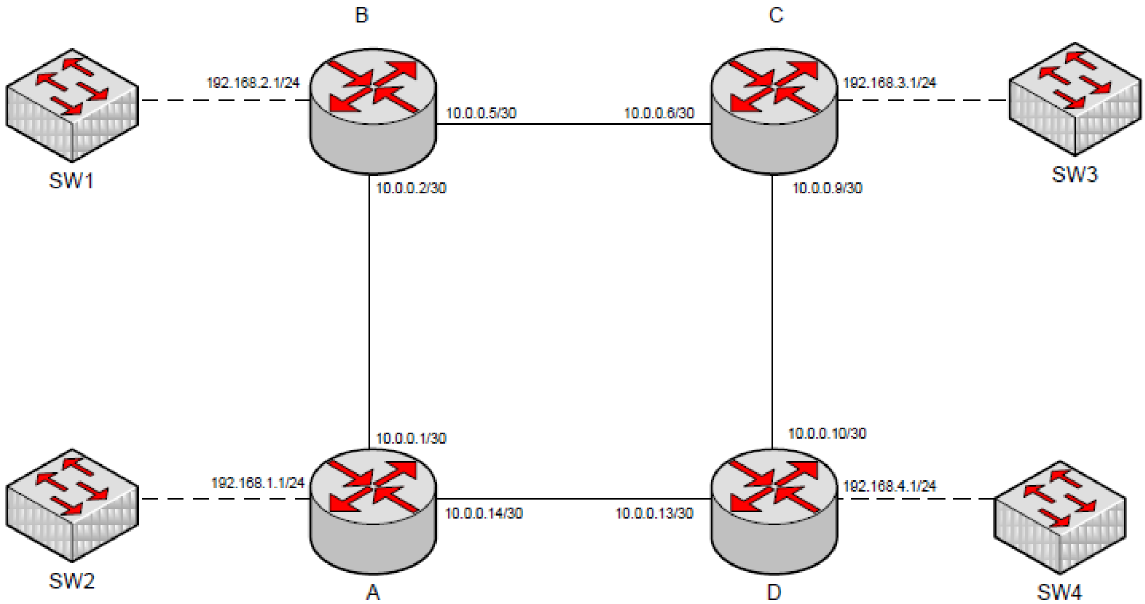
August 2011

Brief Introduction

Introduction

In the simplest of terms, network performance management is considered the second function of network management; the first being fault management. Managing the performance of anything implies the existence of a set of base criteria which represent the optimal way of operation that will be compared to the current operational attributes of the system in question. These are referred to as the performance metrics of the system.

The performance metrics of Internetworking protocols depend on the layer at which the subject protocol operates. In this study, the goal was to select a small number of these protocols – a few from each layer – as the subjects. Although the subtitles which have been used to group the protocols may not accurately classify the protocols under each section, the idea was and still remains to relate them in functionality. The data and resources for the study have been collected from secondary sources including papers written on the subject and observations from the traces of lab runs based on the lab setup shown in the network diagram below.



The importance of network performance measurement can never be emphasized enough; and is a subject area that hardware manufacturers and network administrators should give the attention it deserves. The process of network performance measurement, as any other performance measurement exercise, starts with capturing data that would be analyzed later to identify trends. There are many tools that facilitate this; and the tool of choice for this study was Wire shark (previously Ethereal) which provides an easy way to capture traffic on network interfaces and visualize it, while having the flexibility to dig deeper into the frame content and details at each layer of the Internetworking stack.

A study on network performance metrics and their composition by DANTE presented at TERENA (a regional research and education network) networking conference in 2006 discusses a Network Metric Composition Framework in which the performance metrics have been categorized into Layer 3 metrics and additional metrics. The first included the usual delay, loss, bandwidth and availability metrics; while the second included device specific data, netflow data and most importantly routing. The approach taken to this CAPSTONE project study takes a slightly different approach, and is based on the premise and assumption that because of the layered architecture of the network protocols imposed by the OSI and TCP/IP models, the protocols and technologies operating at each of the layers impact the overall performance of the networking system. As a result, the study looks at a sample of the protocols at each of these layers and how they interact while eventually making the communication happen.

The goal behind performance measurement and specifying the metrics used in doing so is to outline traffic trends, identify anomalies and tune the performance by eliminating all bottlenecks. The report looks into the following protocols, tools and technologies to achieve this:

- Ping, traceroute, ICMP and DHCP
- TCP and UDP
- HTTP, SSH, Telnet and FTP
- ARP and RARP
- Unicast, Multicast and Ethernet
- RIP, OSPF, IS-IS, and BGP

The approach taken in each section is to provide an overview of the protocol in question discussing its functionality, issues and network performance metrics which vary depending on the protocol’s functionality and role in the communication process. Next, the traces captured in the test runs are presented and commented on; and finally the whole exercise is summarized.

Network Protocols

This section comprises of Ping and Traceroute - which both depend on the Internet Control Message Protocol, and the Dynamic Host Configuration Protocol (DHCP) whose performance characteristics and issues will be studied.

The analysis work is based on the configuration presented in the introduction of this report – the same lab configuration which will be used for the rest of this study. The real world performance factors belong to either one of three categories; the normal network overheard which accounts for 20 percent of the traffic under most circumstances, the external performance limiters such as the processing capabilities and memory capacity of the nodes, and finally – and most importantly – the network configuration problems which is the part we have more control over. The first two factors are mainly a matter of budget and availability of the suitable resources. The network configuration problems may include poor design issues and device misconfigurations.

Tools such as Ping and Traceroute have historically been used to pinpoint and troubleshoot network problems. There are various implementations of both tools by different vendors and it is not our goal to compare them; however, the goal is to look into their utility as performance measuring tools; and also look into the performance issues these tools themselves might create.

Overview of Ping and Traceroute

Ping is the most commonly used network diagnostics tool and performance evaluation in TCP/IP networks. It can be very useful in identifying network protocol problems that inhibit smooth communication between nodes, and the measure packet delay which is a great performance metric and indicator of faults in the network.

For security issues mostly involving target reconnaissance and more dangerously Denial of Service Attacks (especially DDoS), many of the ISPs filtered out the ICMP echo packets (message type 8) which is the foundation of Ping and Traceroute, rendering them less effective since 2003.

ICMP packet

	Bit 0 - 7	Bit 8 - 15	Bit 16 - 31
IP Header (20 bytes)	Version/IHL	Type of service	Length
	Identification		*flags* and *offset*
	Time To Live (TTL)	Protocol	Checksum
	Source IP address		
	Destination IP address		
ICMP Payload (8+ bytes)	Type of message	Code	Checksum
	Quench		
	Data (*optional*)		

Figure 1 ICMP Packet

Ping works by sending an echo request message to the destination node; the destination node returns an echo reply. The packet loss is recorded, and the time between the transmission and reception of the ICMP packets is measured to be presented as the Round-Trip Time. The uses of the ping tool include:

- Testing the availability and reach-ability of a node

- Delay and round-trip times of packets

- Packet losses and high input queues drops by comparing the input queue drops and the actual output drops 0000

The ‘debug ip packet’ feature of Cisco routers helps provide even more detailed information about the ping results. It will give the details of the nature of the unreachable message returned by ICMP, for example.

Traceroute, like ping, has different implementations depending on the platform and operating system; this is exemplified by the variations in the command name where it is tracert in windows and traceroute in Unix and Unix-like operating systems.

Traceroute sends 3 UDP datagrams with their Time-to-live (TTL) field set to one; when it reaches its destination, it responds with an ICMP Time Exceeded Message (TEM) - which is message type 11 - indicating the expiration of the packet. The process continues in a recursive manner until the final destination is reached. The purpose is to give a trace of the path the packet took to reach that final destination.

Issues with Ping and Traceroute in Performance Measuring

The Round-Trip-Time actually only gives a rough idea of the delay in the network link as it considers the general picture of the time required to send an echo packet and get an answer. The problem is that this metric is not precise enough for performance evaluation; and the reason is that the node (a PC or a router) caries out some process-switching which most of the time considers the ping packet to have less priority. If the router, for example, is busy processing other tasks (process-oriented services), it will take longer to receive the ICMP echo reply.

Traffic Generation and Capture Process

Pinging across the internetwork (global)

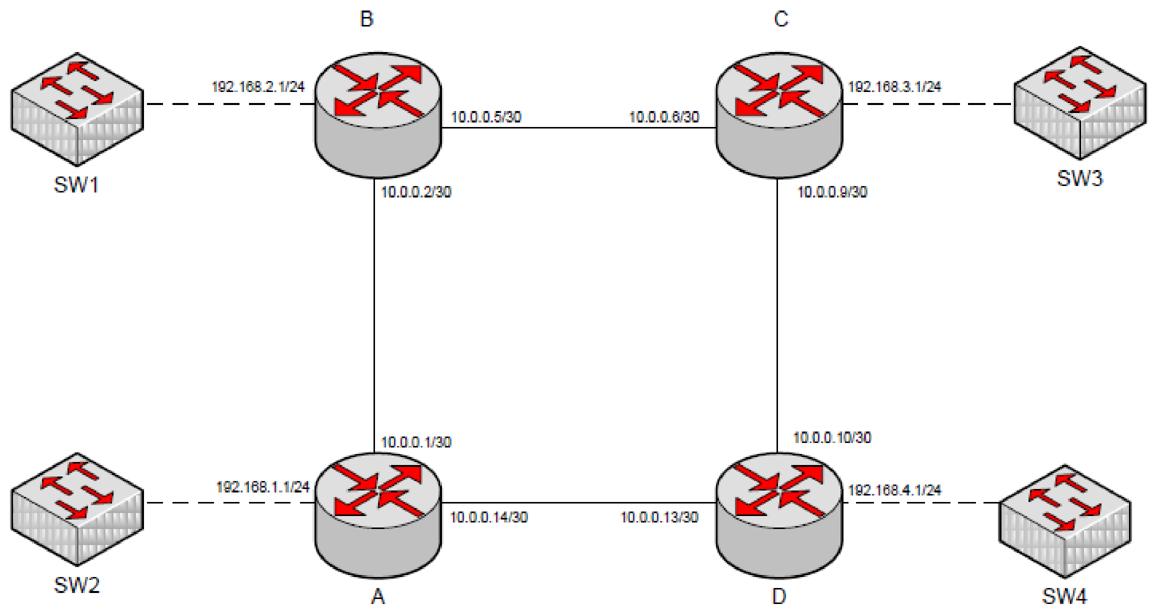


Figure 2 Test network

A node (laptop) attached to SW3 with IP address 192.168.3.3 attempts to ping another device connected to SW2 with IP address 192.168.1.2 in this scenario. This provides an example of pinging through the routers to ensure that the delay and related round-trip-time metric are not more than approximations since the intermediary routing devices will not consider the ping traffic as being of less priority. Below is the detail of the captured packets:

86	17.441660	192.168.3.3	192.168.3.255	NBNS	Name query NB ISATAP<00>
87	17.596952	192.168.3.3	192.168.1.2	ICMP	Echo (ping) request (id=0x0001, seq(be/le)=48/12288, ttl=128)
88	17.637471	192.168.1.2	192.168.3.3	ICMP	Echo (ping) reply (id=0x0001, seq(be/le)=48/12288, ttl=61)
89	18.205310	192.168.3.3	192.168.3.255	NBNS	Name query NB ISATAP<00>
90	18.595256	192.168.3.3	192.168.1.2	ICMP	Echo (ping) request (id=0x0001, seq(be/le)=49/12544, ttl=128)
91	18.635740	192.168.1.2	192.168.3.3	ICMP	Echo (ping) reply (id=0x0001, seq(be/le)=49/12544, ttl=61)
92	18.674203	192.168.3.3	208.84.198.145	UDP	Source port: 37539 Destination port: 42784

Detailed ICMP message

```
⊟ Frame 87: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
    Arrival Time: Jul  5, 2010 01:28:22.398339000 E. Africa Standard Time
    Epoch Time: 1278282502.398339000 seconds
    [Time delta from previous captured frame: 0.155292000 seconds]
    [Time delta from previous displayed frame: 0.155292000 seconds]
    [Time since reference or first frame: 17.596952000 seconds]
    Frame Number: 87
    Frame Length: 74 bytes (592 bits)
    Capture Length: 74 bytes (592 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
⊟ Ethernet II, Src: CompalCo_b3:c4:ab (00:16:d4:b3:c4:ab), Dst: Cisco_bf:83:20 (00:08:21:bf:83:20)
  ⊞ Destination: Cisco_bf:83:20 (00:08:21:bf:83:20)
  ⊞ Source: CompalCo_b3:c4:ab (00:16:d4:b3:c4:ab)
    Type: IP (0x0800)
⊟ Internet Protocol, Src: 192.168.3.3 (192.168.3.3), Dst: 192.168.1.2 (192.168.1.2)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 60
    Identification: 0x0527 (1319)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: ICMP (1)
  ⊞ Header checksum: 0xb044 [correct]
    Source: 192.168.3.3 (192.168.3.3)
    Destination: 192.168.1.2 (192.168.1.2)
⊟ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4d2b [correct]
    Identifier: 0x0001
    Sequence number: 48 (0x0030)
    Sequence number (LE): 12288 (0x3000)
  ⊞ Data (32 bytes)
```

Traceroute Scenario

```
C:\Users\haroun>tracert 192.168.1.2

Tracing route to TEAM4 [192.168.1.2]
over a maximum of 30 hops:

  1     1 ms     1 ms     1 ms  192.168.3.1
  2    25 ms    25 ms    25 ms  10.0.0.5
  3    49 ms    49 ms    49 ms  10.0.0.1
  4    59 ms    59 ms    59 ms  TEAM4 [192.168.1.2]

Trace complete.
```

As seen in the sample outputs for the ping program, the round-trip response time values for each ping packet sent are shown in the ping packet statistics:

64 bytes from 192.168.1.100: icmp_seq=0 ttl=255 time=0.712 ms

The response time is shown in milliseconds. For internal LAN connections, the response times should be well within 1 or 2 milliseconds. For WAN connections, the response times can often be over 200 or 300 milliseconds, depending on WAN connectivity speeds. For VSAT connections it is approximately 1000 – 1400 ms round trip time according to Wikipedia entry on Satellite Internet access.

The tracert command is executed from router C, the interface with IP address 192.168.3.1, and the results of its execution is shown above. The captured packet trace is shown below:

47	23.709160	192.168.1.3	192.168.1.2	ICMP	Echo (ping) request (id=0x0001, seq(be/le)=36/9216, ttl=1)
48	23.709523	192.168.1.2	192.168.1.3	ICMP	Echo (ping) reply (id=0x0001, seq(be/le)=36/9216, ttl=64)
49	23.710641	192.168.1.3	192.168.1.2	ICMP	Echo (ping) request (id=0x0001, seq(be/le)=37/9472, ttl=1)
50	23.710832	192.168.1.2	192.168.1.3	ICMP	Echo (ping) reply (id=0x0001, seq(be/le)=37/9472, ttl=64)
51	23.711752	192.168.1.3	192.168.1.2	ICMP	Echo (ping) request (id=0x0001, seq(be/le)=38/9728, ttl=1)
52	23.711952	192.168.1.2	192.168.1.3	ICMP	Echo (ping) reply (id=0x0001, seq(be/le)=38/9728, ttl=64)

Performance Analysis

In a Cisco network, as in any other networks based on other vendor's products, performance of the network is limited by the medium itself. In addition to the standard overheard that comes with TCP/IP protocols, turning on diagnostic and debugging tools will have a significant performance reduction on the network.

To ensure the accuracy of measuring performance attributes of a network, especially delay and throughput, either of two things are required:

- If the diagnostics are being done on a node that is not an intermediary router, to make sure that the same node is not involved in any process-intensive tasks. The suggestion here is to execute ping and traceroute from a standard computer

- If these commands are being executed on a router, most process intensive tasks need to be turned off, including debug and related diagnostic commands.

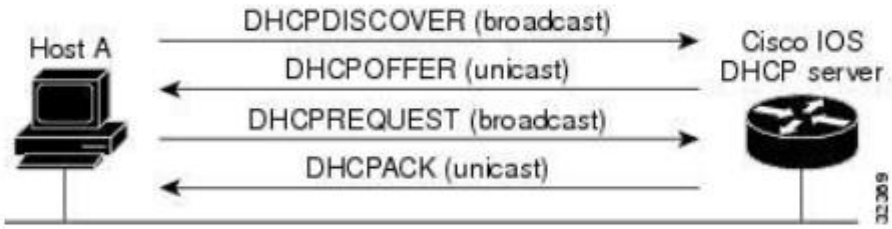
Some resources also suggest that Access-Control Lists may be used to control and filter the traffic that needs to be debugged if it is necessary to keep the debug commands on. Buffering debug messages to be viewed later using ‘show log’ command is also another option.

Dynamic Host Configuration Protocol

Protocol Overview

Dynamic Host Configuration Protocol (DHCP) is a client-server architecture protocol for automatically providing configuration parameters such as IP addresses, default gateways and subnet mask information to hosts on a network.

DHCP supports three mechanisms for IP address allocation. In automatic allocation, DHCP assigns a permanent IP address to a client. In dynamic allocation, DHCP assigns an IP address to a client for a limited period of time (or until the client explicitly relinquishes the address). In manual allocation, a client's IP address is assigned by the network administrator, and DHCP is used simply to convey the assigned address to the client. A particular network will use one or more of these mechanisms, depending on the policies of the network administrator.



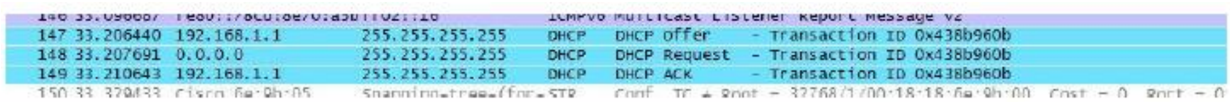
By using DHCP, dynamically configuring the host to the network is done by a simple handshake. In history, there are also many dynamic automatic configuration protocols. Other protocols that can also provide the mechanism of automatic configuration include RARP and BOOTP. These protocols use simple interaction; the client requests and the server replies. RARP (Reverse Address Resolution Protocol) is executed on Ethernet, and converts the Ethernet address to an IP address. RARP handshake is mainly used in the diskless workstations. RARP uses an Ethernet frame directly, meanwhile BOOTP uses UDP. BOOTP returns IP addresses with the subnet mask of a network, IP addresses of routers, etc. RARP and BOOTP have two defects. First, these protocols only support static allocation (conversion) of an IP address. RARP and BOOTP protocol do not solve the requirement of dynamic allocation. Secondly, these protocols can provide only few parameters.

Traffic Generation and Capture Process

Using the same lab configuration as above, the following packets have been captured during the automatic IP assignment process that started once a laptop was connected to the network.

The first step is to locate a DHCP server through broadcast to the segment.

Following the transaction ID, it was possible to follow the rest of the process, as shown below.



Once the DHCP server is located, the client sends the configuration information request directly to the server node, in which case the client will be assigned an IP address and other configuration details, with a specific lease time.

A detailed view of the parameter assignment is shown here as well.

```
⊞ Frame 147: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
⊞ Ethernet II, Src: Cisco_96:f2:41 (00:08:21:96:f2:41), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol, Src: 192.168.1.1 (192.168.1.1), Dst: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
⊟ Bootstrap Protocol
    Message type: Boot Reply (2)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x438b960b
    Seconds elapsed: 0
  ⊞ Bootp flags: 0x8000 (Broadcast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 192.168.1.3 (192.168.1.3)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: CompalCo_b3:c4:ab (00:16:d4:b3:c4:ab)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ⊞ Option: (t=53,l=1) DHCP Message Type = DHCP Offer
  ⊞ Option: (t=54,l=4) DHCP Server Identifier = 192.168.1.1
  ⊞ Option: (t=51,l=4) IP Address Lease Time = 7 days
  ⊞ Option: (t=58,l=4) Renewal Time Value = 3 days, 12 hours
  ⊞ Option: (t=59,l=4) Rebinding Time Value = 6 days, 3 hours
  ⊞ Option: (t=1,l=4) Subnet Mask = 255.255.255.0
  ⊞ Option: (t=3,l=4) Router = 192.168.1.1
    End Option
    Padding
```

Performance Analysis of DHCP

Bahlmann (2002) suggests an approach to testing carrier class DHCP and puts forth the following performance parameters:

- Average transaction time: The current average of all completed DHCP transactions between server and client. This number is helpful and will gradually increase as the server and the network becomes increasingly taxed.

- Average overall cycle time: The current average of all completed DHCP cycles with the server (DISCOVER to ACK).

- Percentage of completed DHCP transactions: The percentage of the number of transactions with the server that have been successfully completed by the DHCP client generator (completed as opposed to timed out or dropped).
- Current transaction rate: The number of transactions currently being sent to the server per second.

As each client transaction is about to begin, it is helpful to obtain a snap shot of these average times, the last completed individual transaction, and the overall cycle time and then store these along with the record assigned to the impending transaction. The purpose of obtaining this snap shot is to be able to determine the overall performance of the DHCP server upon the last good transactions before it begins dropping packets (as finding this point should be the goal of any quality DHCP testing). When stress testing, you want to find the spot at which the server fails, begins to drop packets, and/or does not complete requested DHCP transactions with clients. Note that each of these spots may take place at different times (if at all) as load is increased (failure of the server may or may not occur unless the incoming packets somehow overload the application, available resources [disk, connection, memory, etc.], or the operating system [swap/virtual memory, memory, disk, etc.]). If the server does not fail, it may just drop some packets while completing others – it all depends on the capability of the server to prioritize its processing capability and complete the work it has started. It is the duty of the client generator to determine this point as well as the performance of the server leading up to that point. The sweet spot of the server (how many DHCP clients it can effectively maintain during any given time) may well be the spot at which the server can no longer keep up with any additional load or potentially just beyond this point depending on what the server does upon reaching saturation as well as its ability to overcome these instances and catch back up with the incoming requests.

Transport Protocols

Overview of TCP and UDP

The Transmission Control Protocol and the User Datagram Protocol are the most commonly used transport layer protocols of today. Their performances, although affected by that of the other lower layer protocols on top of which they run, defines the overall performance of the communication link. This section attempts to look deeper into the two transport protocols and identify their performance metrics, while exploring the performance tuning approaches for the two protocols.

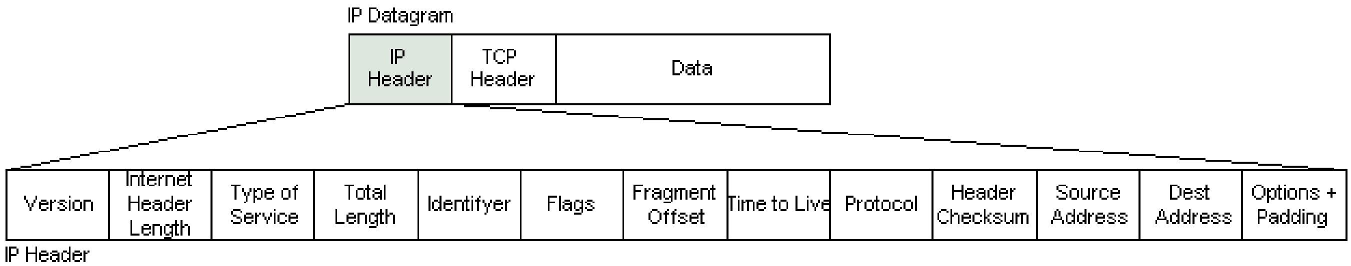
A key element in the performance of any communication link is the physical layer through which the actual transmission takes place. Several papers and literature have studied the performance of TCP and UDP on the various mediums commonly in use today including but not limited to wireless and optical networks. The focus of this section is to propose a holistic, more generic approach to the key performance metrics pertaining to the two protocols.

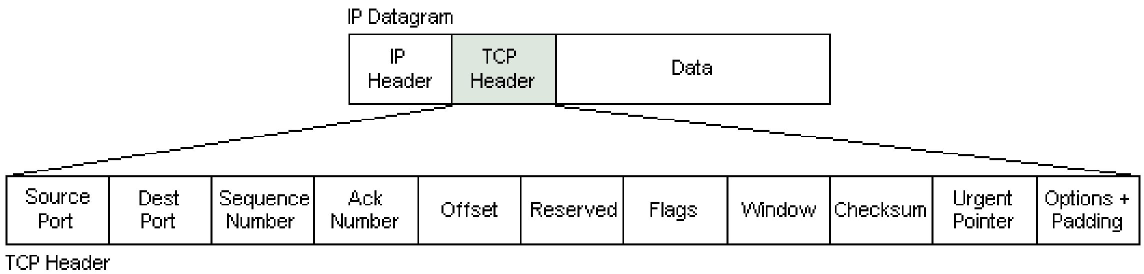
While the two protocols have been designed to tackle the end to end transmission of the packets, the purposes of their design and hence their uses vary. Understanding the differences in their behaviors and their respective applications is crucial as they drive the communications and data transmission across the Internet. This will eventually contribute to understanding their performance characteristics.

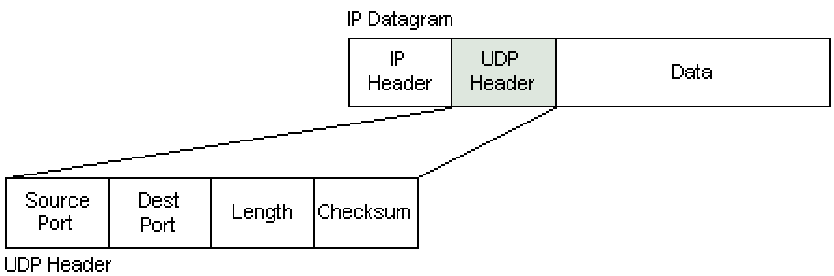
The Transmission Control Protocol (TCP) is used to provide reliable transmission between two nodes, which is facilitated by mechanisms built into the protocol that ensure the establishment of a virtual connection (session) before transmission, and acknowledgment of the packets sent among other techniques. TCP also provides congestion control, meaning it reduces its frame sending rate if it detects that the network is overloaded. Most typical applications need the reliability and other services provided by TCP, and don't care about loss of a small amount of performance to overhead. For example, most applications that transfer files or important data between machines use TCP, because loss of any portion of the file renders the entire operation useless. Examples include such well-known applications as the Hypertext Transfer Protocol (HTTP) used by the World Wide Web (WWW), the File Transfer Protocol (FTP) and the Simple Mail Transfer Protocol (SMTP).

On the other hand, the User Datagram Protocol (UDP) is a connection-less transport protocol that gives no guarantees on the success rate of the transmissions. That is to say that applications using UDP as the transport layer protocol do not require the guarantee that the data sent was received successfully. Although this might seem bad, it is important to note that no one protocol is better than the other, it is only that one is more suitable for certain situations than the other. The overhead that is typical of TCP might not be required for certain applications such as VOIP, while it is important that an FTP session has successfully completed despite the overhead/cost involved in establishing and maintaining a session over TCP.

IP, TCP and UDP Header Formats







The differences in frame and/or header structures indicates the differences in behavior, and hence in performance requirements

Reliability:

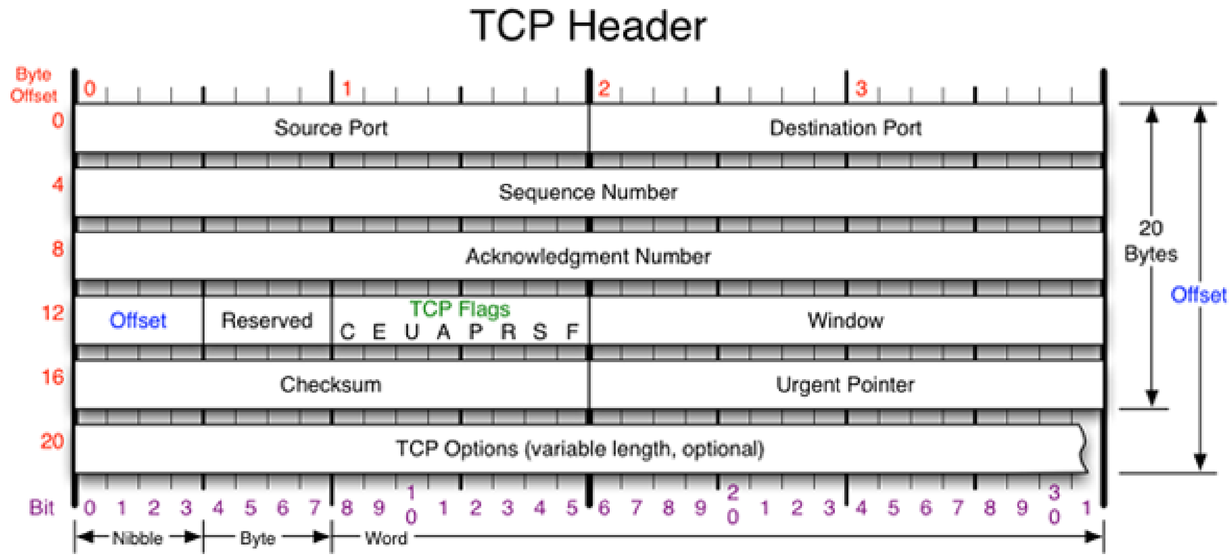
TCP: connection-oriented

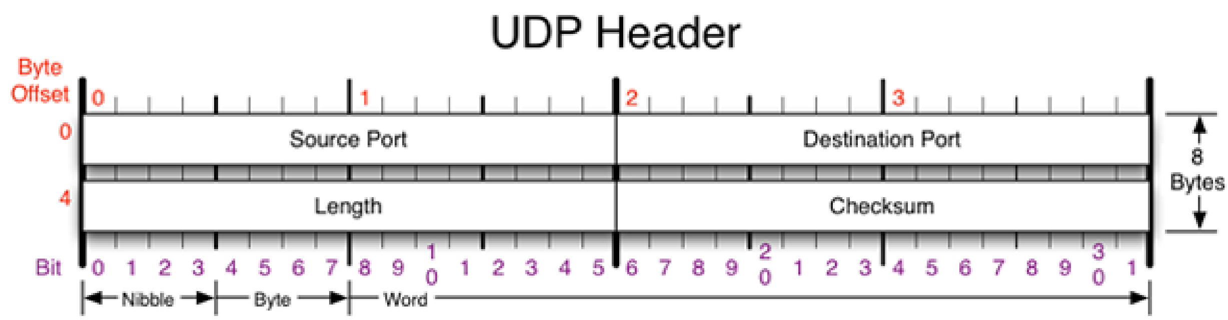
UDP: connectionless

Ordered:
TCP: order of message receipt is guaranteed
UDP: order is not guaranteed
Protocol weight:
TCP: heavyweight, because of the connection/ordering overhead
UDP: lightweight, very few overhead
Packets:
TCP: streaming, data is read as a stream, with nothing distinguishing where one packet ends and

another begins. There may be multiple packets per read call.
UDP: datagrams, one packet per one read call.

More detailed header formats are presented below:



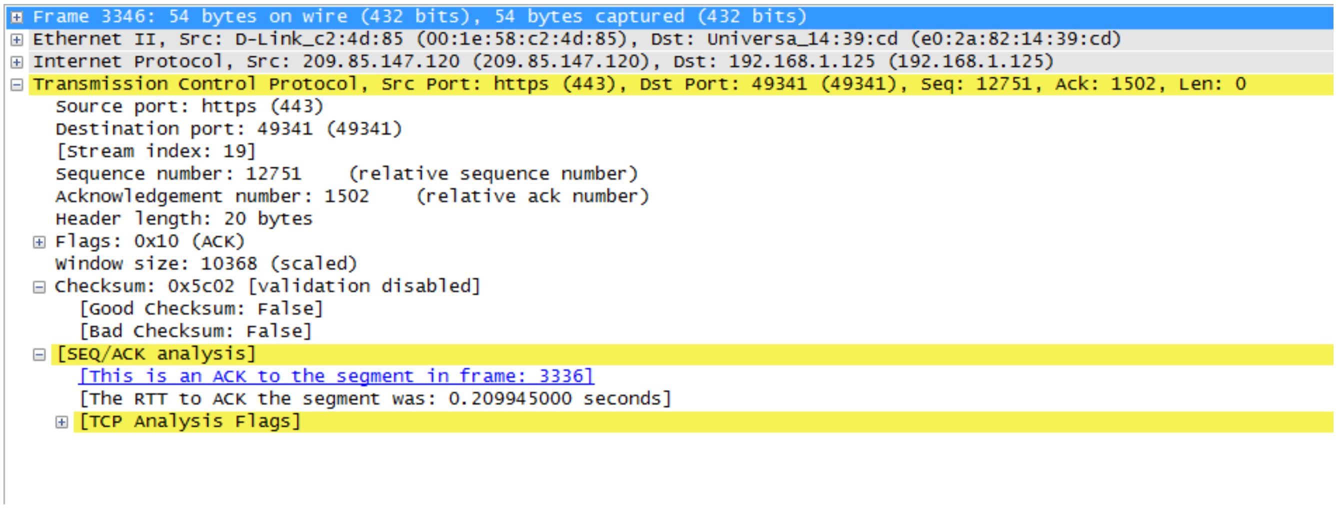


Traffic Capture Process/Methodology

In the process of analyzing the performance of the two transport layer protocols, data from the previous captures has been perused. According to the test network design and the traffic captured during the test runs, TCP and UDP packets did not require any additional or specific traces.

The ideal approach could have been the modeling of UDP and TCP packets using queuing theory to pinpoint opportunities for improved optimization and tuning.

Performance Metrics



```
⊞ Frame 3346: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
⊞ Ethernet II, Src: D-Link_c2:4d:85 (00:1e:58:c2:4d:85), Dst: Universa_14:39:cd (e0:2a:82:14:39:cd)
⊞ Internet Protocol, Src: 209.85.147.120 (209.85.147.120), Dst: 192.168.1.125 (192.168.1.125)
⊟ Transmission Control Protocol, Src Port: https (443), Dst Port: 49341 (49341), Seq: 12751, Ack: 1502, Len: 0
    Source port: https (443)
    Destination port: 49341 (49341)
    [Stream index: 19]
    Sequence number: 12751    (relative sequence number)
    Acknowledgement number: 1502    (relative ack number)
    Header length: 20 bytes
  ⊞ Flags: 0x10 (ACK)
    Window size: 10368 (scaled)
  ⊟ Checksum: 0x5c02 [validation disabled]
      [Good Checksum: False]
      [Bad Checksum: False]
  ⊟ [SEQ/ACK analysis]
      [This is an ACK to the segment in frame: 3336]
      [The RTT to ACK the segment was: 0.209945000 seconds]
    ⊞ [TCP Analysis Flags]
```

The Wireshark packet traces above show the TCP header structure of live traffic. The various sections in the header represent some key performance metrics that can be manipulated to enhance the performance of TCP including most importantly the Window Size. However, there are other elements that contribute to the performance of a TCP or a UDP transmission, some of those elements which have also been addressed by other literature include:

1. The physical layer aspects of the transmission link (the hardware and the medium)
2. The throughput including the maximum number of transactions per second, the MTU size
3. The adapter receive and transmit queues
4. Device specific buffers which again involves the hardware I/O characteristics alluded to in number 1

Performance Analysis of the Transport Layer Protocols

In general, it is known that TCP provides a reliable connection through its three-way handshake process, whereas UDP does not. In addition, the acknowledgement and retransmit features, TCP facilitates a more reliable link and is more suited to applications requiring the transmission of large amounts of data.

This study and others before it show that TCP also offers higher throughput than UDP; however, when using UDP the end-to-end delay performance improves which makes it more suitable for delay sensitive applications such as VOIP, and other applications that require the transmission of information in small bursts such as those used for Telemetry and tele-operations.

Application Protocols

Introduction and Overview

This section of the CAPSTONE report deals with the performance metrics and issues of higher layer protocols with the goal of pinpointing the possible performance bottleneck areas. In this context, higher layer refers to the application layer protocols that are most commonly used in the TCP/IP protocol suite out of which four essential, very popular protocols have been selected to understand the performance issues surrounding the higher layer protocols.

These four protocols are:

- Hyper Text Transfer Protocol (HTTP): the ubiquitous protocol that made the World Wide Web and other Internet services possible.
- File Transfer Protocol (FTP): the protocol that makes it possible to transfer files between two nodes across networks.
- Telnet: is a protocol that facilitates bidirectional text based communication between two nodes using virtual terminals
- SSH: Secure Shell is Telnet's more secure cousin mostly used for out-of-band system administration

To understand the potential performance bottlenecks in the higher layer protocols, we need to first identify the underlying protocols in the stack that deal with the transport layer and data link layers. Looking down the OSI layer stack, higher layer protocols are susceptible to the weaknesses and performance issues of the lower layer protocols on which they depend for moving their data from one node to another destination node. By dissecting the structure of a higher layer data frame and identifying the various elements it contains that are critical for the performance of the protocols and their operations, it should be apparent as to what metrics are involved and how we can tweak that to squeeze the maximum performance out of the connections or sessions.

Performance Metrics and Measurement Issues

HyperText Transfer Protocol

HTTP protocol is what makes the World Wide Web possible. HTTP is a generic stateless object-oriented protocol, which may be used for many similar tasks such as name servers, and distributed object-oriented systems, by extending the commands, or methods, used. A feature of HTTP is the negotiation of data representation, allowing systems to be built independently of the development of new advanced representations.

On the internet, the communication takes place over a TCP/IP connection. This does not preclude this protocol being implemented over any other protocol on the internet or other networks. In these cases, the mapping of the HTTP request and response structures onto the transport data units of the protocol in question is outside the scope of the specification of the protocol. However, it should not be that complicated specially considering the layered architecture of the networking models commonly used.

The protocol is basically stateless, a transaction consisting of:

Connection: The establishment of a connection by the client to the server - when using TCP/IP port 80 is the well-known port, but other non-reserved ports may be specified in the URL;

Request: The sending, by the client, of a request message to the server;

Response: The sending, by the server, of a response to the client;

Close: The closing of the connection by either both parties.

The format of the request and response parts is defined in RFC 2068 and related specifications.

Analysis of the HTTP Traces

	Time	Source	Destination	Protocol	Info
1	0.000000	142.244.164.24	74.125.127.105	HTTP	GET /firefox?client=firefox-a&rls=org.mozilla:en-US:official HTTP/1.1
2	0.031603	74.125.127.105	142.244.164.24	HTTP	HTTP/1.1 302 Found (text/html)
3	0.098126	142.244.164.24	74.125.127.99	HTTP	GET /firefox?client=firefox-a&rls=org.mozilla:en-US:official HTTP/1.1
4	0.172179	142.244.164.24	74.125.127.99	HTTP	GET /images/firefox/redpandahead.png HTTP/1.1
5	0.203936	74.125.127.99	142.244.164.24	HTTP	HTTP/1.1 200 OK (PNG)
6	0.261444	74.125.127.105	142.244.164.24	HTTP	[TCP Retransmission] HTTP/1.1 302 Found (text/html)
7	0.436956	74.125.127.99	142.244.164.24	HTTP	[TCP Retransmission] HTTP/1.1 200 OK (PNG)
8	4.023482	142.244.164.24	74.125.127.99	HTTP	GET /imghp?client=firefox-a&rls=org.mozilla:en-US:official&hl=en&tab=wi HTTP/1.1
9	4.088369	74.125.127.99	142.244.164.24	HTTP	[TCP Previous segment lost] Continuation or non-HTTP traffic
10	4.092684	142.244.164.24	74.125.127.99	HTTP	[TCP ACKed lost segment] GET /intl/en_ALL/images/logos/images_logo_lg.gif HTTP/1.1

The above screen capture of the trace in Wireshark demonstrates the request-response mechanism employed by the HTTP protocol. The request is sent by the browser using the GET message and indicating the browser and the version of supported HTTP; the first response is a confirmation that the request resource has been found; and it then goes about iteratively downloading the elements of the requested page including the images and then an OK acknowledgement in the form of the 200 code is sent back by the web server.

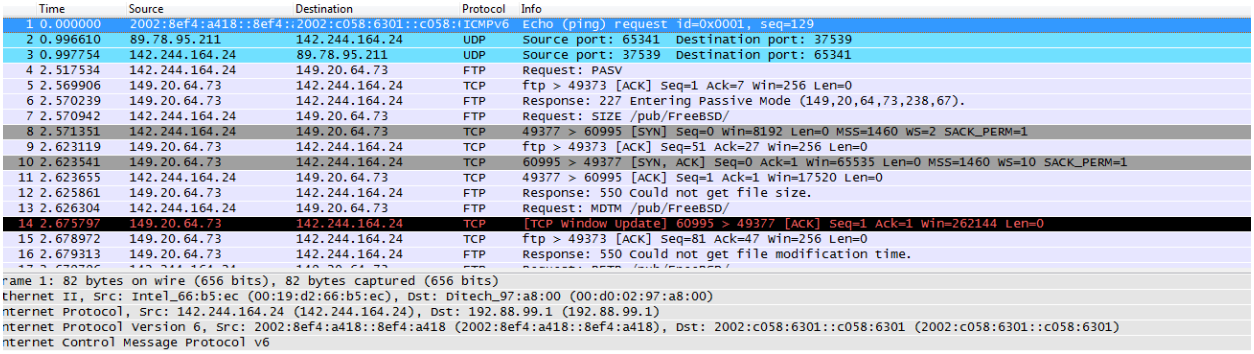
This simple structure of the request response mechanism is what makes it easy and straight forward to code browsers and web servers. A major issue in HTTP performance is the compression mechanism employed; this can affect the throughput and speed of the http protocol in any given scenario.

Most often, HTTP compression is implemented on the server side as a filter or module which applies the gzip algorithm to responses as the server sends them out. Any text based content can be compressed. In the case of purely static content, such as markup, style sheets, and JavaScript, it is usually possible to cache the compressed representation, sparing the CPU of the burden of repeatedly compressing the same file. When truly dynamic content is compressed, it usually must be recompressed each time it is requested (though there can be exceptions for quasi dynamic content, given a smart enough compression engine). This means that there is trade off to be considered between processor utilization and payload reduction. A highly configurable compression tool enables an administrator to adjust the tradeoff point between processor utilization and compressed resource size by setting the compression level for all resources on the web site, thereby not wasting CPU cycles on over compressing objects which might compress just as tightly with a lower level setting as with a higher one. This also allows for the exclusion of binary image files from HTTP compression, as most images are already optimized when they are created in an editor such as Illustrator. Avoid the needless recompression of images as this may actually increase their file size or introduce distortion.

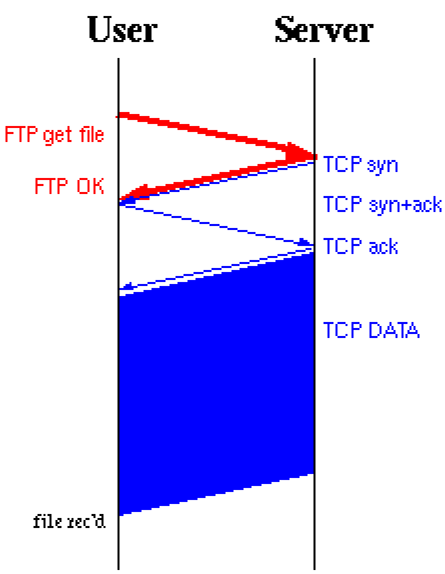
All in all, HTTP performance comes down to the implementation and configuration of the web server. Focusing on tweaking the web server’s performance by modifying its operating parameters would result in great returns in performance.

HTTP compression, otherwise known as content encoding, is a publicly defined way to compress textual content transferred from web servers to browsers. HTTP compression uses public domain compression algorithms, like gzip and compress, to compress XHTML, JavaScript, CSS, and other text files at the server. This standards-based method of delivering compressed content is built into HTTP 1.1, and most modern browsers that support HTTP 1.1 support ZLIB inflation of deflated documents. In other words, they can decompress compressed files automatically, which saves time and bandwidth.

File Transfer Protocol



The FTP protocol follows the same request and response mechanism as the HTTP protocol show in the following diagram. The diagram below explains the above traces in more stark terms.

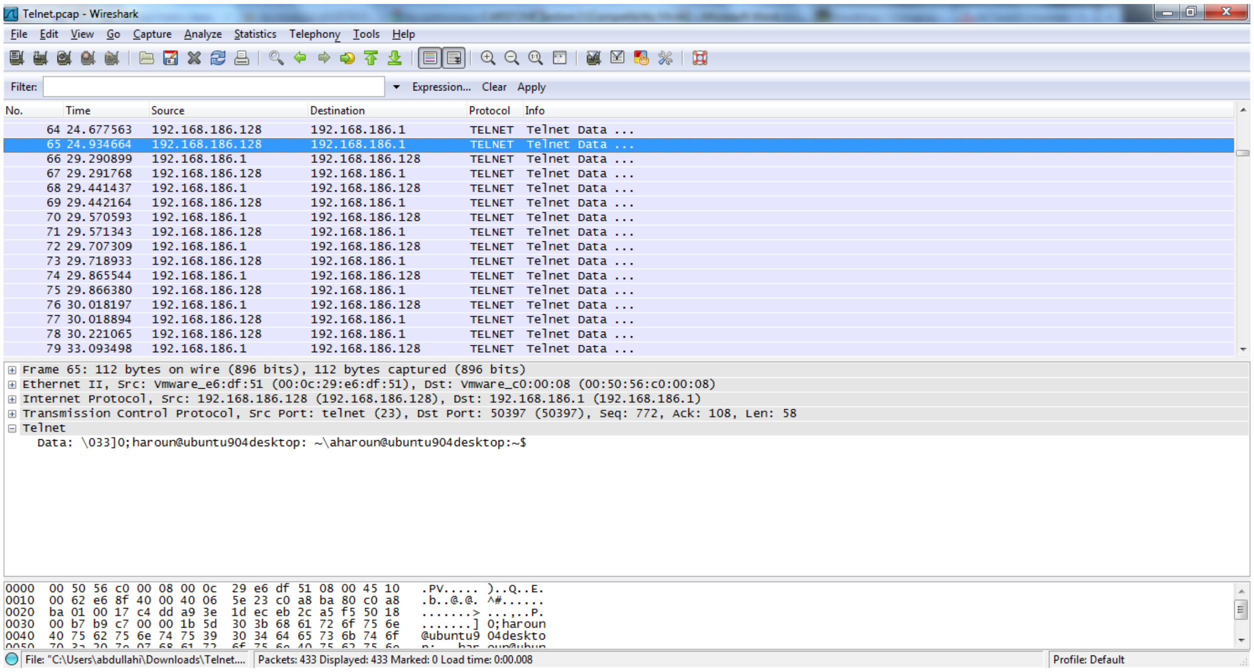


The factors that affect the performance metrics of an FTP server include:

- Mechanical elements such as the type of disks on the server and the nature of the IO operations needed to reply to the requests
- The file system type
- Any FTP caches in place
- Lower level protocols underlying the FTP protocol operations

The usual performance metrics apply to the FTP protocol as well as the HTTP protocol; most important of these is the throughput and the data transfer rate.

Telnet

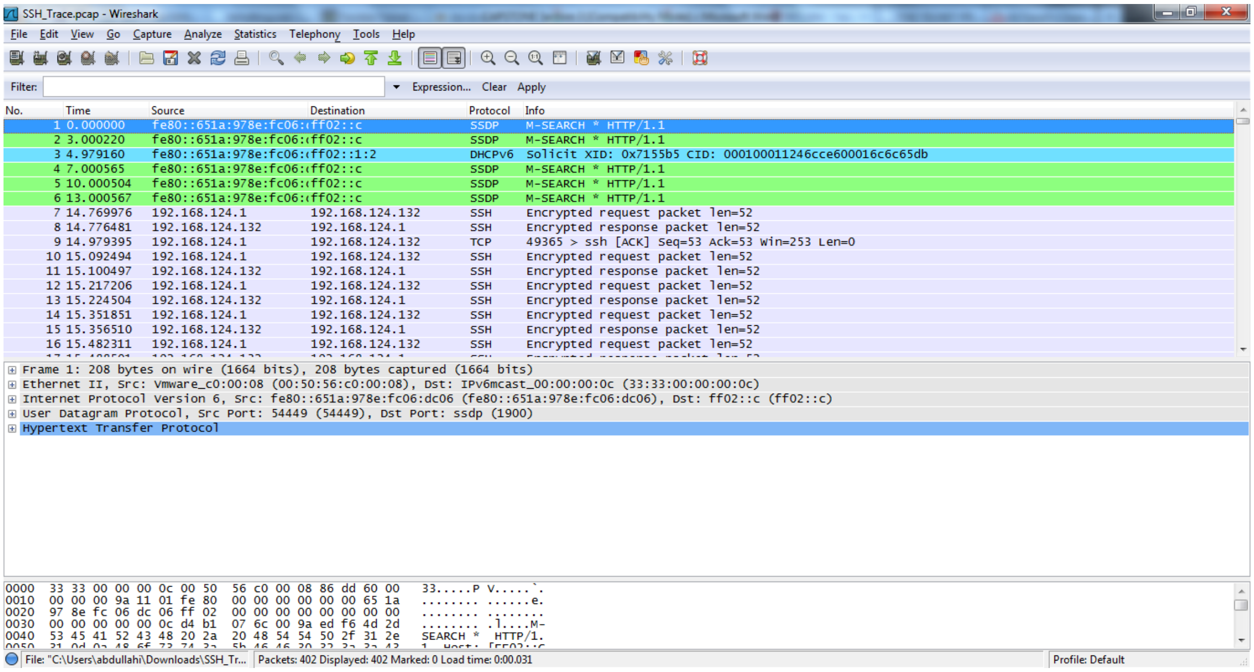


TELNET is a general protocol, meant to support logging in from almost any type of terminal to almost any type of computer. Its use and functionality, however, seems to be left with console connections to routers and computers in secure environments. It allows a user at one site to establish a TCP connection to a login server or terminal server at another site. A TELNET server generally listens on TCP Port 23.

The protocol is insecure by design as can be seen from the above figure where the submitted password is shown in the traces in plaintext. The protocol uses the concept of Network Virtual Terminals, and the connection between the two nodes is full duplex although it does not seem to be like that as the nature of the communication it is used for does not take full advantage of this capability.

The key performance metrics for the TELNET protocol are affected by the underlying TCP protocol’s own performance.

Secure Shell (SSH)



SSH or Secure Shell is the most common remote login protocol and application in use today, as it offers the security protocols such as rlogin and telnet lack. The protocol is based on TCP; and in its simplest mode of operation, it connects to a server, negotiates a shared secret key using Diffie- Hellman, then begins encrypting the session (typically using the Blowfish cipher). A username and password are passed over the encrypted session and, if authenticated, the server starts a command shell over the encrypted session.

Using TCP at the Transport Layer poses some performance issues. A number of network applications make use of multiplexed channels inside of a single TCP connection to handle data transfer and/or control information. Because these channels cannot make use of the TCP windows for flow control they must implement their own. This means that a second window can be imposed on top of the existing TCP window. The result of this is that even if the TCP window is correctly sized for the current to produce exceptional FTP performance a user may still encounter dismal throughput under one of these applications. This is because the application window, which is often statically defined, is too small for many typical paths. This forces the connection to slow down to the limit of the smaller of the two windows.

The best current example of this is the SSH2 protocol. It is not uncommon for a user to be sitting on a connection they can utilize less than 1% of because of this double window problem. While a user might not experience any issues in interactive sessions it’s a very noticeable problem in bulk data transfers (e.g. SCP, rsync -essh, sftp, etc) and is common source of frustration – especially for users with access to high performance network connections.

Data Link Protocols

Overview

This section looks into the second layer among the OSI networking model, and the performance issues related to this layer. Although this study might not look into the low level mechanisms that define this layer's functions, it will focus more into a couple of protocols and concepts that are part of the layer's operations. The function of the data link layer is to take requests from the network layer and send requests to the physical layer below it. To be more specific, the data link layer has the following functions:

Logical Link Control (LLC): Logical link control refers to the functions required for the establishment and control of logical links between local devices on a network. As mentioned above, this is usually considered a DLL sub layer; it provides services to the network layer above it and hides the rest of the details of the data link layer to allow different technologies to work seamlessly with the higher layers. Most local area networking technologies use the IEEE 802.2 LLC protocol.

Media Access Control (MAC): This refers to the procedures used by devices to control access to the network medium. Since many networks use a shared medium (such as a single network cable, or a series of cables that are electrically connected into a single virtual medium) it is necessary to have rules for managing the medium to avoid conflicts. For example. Ethernet uses the CSMA/CD method of media access control, while Token Ring uses token passing.

Data Framing: The data link layer is responsible for the final encapsulation of higher-level messages into *frames* that are sent over the network at the physical layer.

Addressing: The data link layer is the lowest layer in the OSI model that is concerned with addressing: labeling information with a particular destination location. Each device on a network has a unique number, usually called a *hardware address* or *MAC address*, which is used by the data link layer protocol to ensure that data intended for a specific machine gets to it properly.

Error Detection and Handling: The data link layer handles errors that occur at the lower levels of the network stack. For example, a cyclic redundancy check (CRC) field is often employed to allow the station receiving data to detect if it was received correctly.

Commonly used examples of Data Link Layer protocols are Ethernet, PPP and Token Ring. This layer is usually in the form of a software device driver for the network interface card (NIC

The layer itself is divided into two parts as mentioned above- MAC and LLC, which communicate with the layers above and below the data link layer. MAC (media access control) determines how data on a network meant for a specific computer reaches it and how a computer can transmit data. Every physical card has a unique MAC address and every frame sent on the network has both source and destination MAC addresses in the header. So the receiving DLL knows which frames on the network are meant for itself, and which computer sent the frame.
In this category, the focus is on ARP, RARP, multicast, unicast and Ethernet. The objective is to point out the key performance issues and related factors, and will follow the format of the previous sections.

ARP and RARP

Although included in this section, the ARP and RARP protocols are categorized as IP layer protocols. Address Resolution Protocol (ARP) is a required TCP/IP standard defined in RFC 826, Address Resolution Protocol (ARP). ARP resolves IP addresses used by TCP/IP-based software to media access control addresses used by LAN hardware. The Reverse ARP protocol is defined in RFC 903.

ARP provides the following protocol services to hosts located on the same physical network:

- Media access control addresses are obtained by using a network broadcast request in the form of the question what is the media access control address for a device that is configured with the enclosed IP address?
- When an ARP request is answered, both the sender of the ARP reply and the original ARP requester record each other's IP address and media access control address as an entry in a local table called the ARP cache for future reference.

RARP is described in Internet Engineering Task Force (IETF) publication RFC 903. [1] It has been rendered obsolete by the Bootstrap Protocol (BOOTP) and the modern Dynamic Host Configuration Protocol (DHCP), which both support a much greater feature set than RARP. This is most commonly used for products that use mass deployment of software (OSes).

RARP requires one or more server hosts to maintain a database of mappings of Link Layer addresses to their respective protocol addresses. Media Access Control (MAC) addresses needed to be individually configured on the servers by an administrator. RARP was limited to serving only IP addresses.

The Wireshark traces below show the operation of the Address Resolution Protocol in the test network.

	Time	Source	Destination	Protocol	Info
6	22.156909	Intel_66:b5:ec	Broadcast	ARP	Who has 142.244.164.26? Tell 142.244.164.24

In the above diagram, a broadcast message is sent across the network segment as part of the ARP protocol operations. The goal is to get the MAC address of the machine interface with the IP address 142.244.164.26. The response to this will be a unicast message back to the node that initiated the ARP request.

Unicast and Multicast

Unicast, Multicast and Broadcast are basically communication modes used in data transmission; the classification is based on the nature of what is on the receiving end of the transmission; and has many applications in the operations of the internets as will be seen in the upcoming sections.

Unicast packets are sent from host to host. The communication is from a single host to another single host. Broadcast (not included in this discussion) is used when a single device is transmitting a message to all other devices in a given address range. This broadcast could reach all hosts on the subnet, all subnets, or all hosts on all subnets. Broadcast packets have the host (and/or subnet) portion of the address set to all ones. By design, most modern routers block IP broadcast traffic and restrict it to the local subnet.

Multicast is a special protocol for use with IP. Multicast enables a single device to communicate with a specific set of hosts, not defined by any standard IP address and mask combination. This allows for communication that resembles a conference call. Anyone from anywhere can join the conference, and everyone at the conference hears what the speaker has to say. The speaker's message isn't broadcasted everywhere, but only to those in the conference call itself. A special set of addresses is used for multicast communication. In the previous classification of IPv4 addresses, class D was reserved for multicast operations. If the operations of routing protocols are studied carefully, it would appear that protocols such as OSPF and EIGRP use multicast to share their routing tables and updates.

15	13.070093	169.254.178.12	224.0.0.252	IGMP	V2 Membership Report / Join group 224.0.0.252
16	13.161736	169.254.178.12	224.0.0.2	IGMP	V2 Leave Group 239.255.255.250
17	13.163414	169.254.178.12	239.255.255.250	IGMP	V2 Membership Report / Join group 239.255.255.250
18	13.292841	169.254.178.12	224.0.0.2	IGMP	V2 Leave Group 224.0.0.252
19	13.293151	169.254.178.12	224.0.0.252	IGMP	V2 Membership Report / Join group 224.0.0.252
20	13.570121	169.254.178.12	239.255.255.250	IGMP	V2 Membership Report / Join group 239.255.255.250
21	13.570273	169.254.178.12	224.0.0.252	IGMP	V2 Membership Report / Join group 224.0.0.252
22	16.156339	169.254.178.12	224.0.0.2	IGMP	V2 Leave Group 224.0.0.252

In the above trace capture, the highlighted line (17) indicates a multicast message to the address 239.255.255.250 (class D according to the now obsolete and irrelevant IP address classification system).

	Time	Source	Destination	Protocol	Info
1	0.000000	85.73.34.205	142.244.164.24	UDP	Source port: 33203 Destination port: 37539
2	0.000441	142.244.164.24	85.73.34.205	UDP	Source port: 37539 Destination port: 33203
3	1.062953	24.196.201.203	142.244.164.24	UDP	Source port: 55005 Destination port: 37539
4	1.063389	142.244.164.24	24.196.201.203	UDP	Source port: 37539 Destination port: 55005
5	1.348945	190.203.161.103	142.244.164.24	UDP	Source port: 30610 Destination port: 37539
6	1.349368	142.244.164.24	190.203.161.103	UDP	Source port: 37539 Destination port: 30610
7	1.660453	186.45.85.185	142.244.164.24	UDP	Source port: 23825 Destination port: 37539
8	1.660877	142.244.164.24	186.45.85.185	UDP	Source port: 37539 Destination port: 23825
9	1.725722	2002:8ef4:a418::8ef4:	2002:c058:6301::c058:	ICMPv6	Echo (ping) request id=0x0001, seq=265
10	1.750210	142.244.164.24	129.128.5.233	DNS	Standard query A data2.wowzio.com
11	1.750581	142.244.164.24	129.128.76.233	DNS	Standard query A data2.wowzio.com
12	1.751373	129.128.5.233	142.244.164.24	DNS	Standard query response A 67.207.149.136
13	1.751483	129.128.76.233	142.244.164.24	DNS	Standard query response A 67.207.149.136
14	1.752206	142.244.164.24	129.128.5.233	DNS	Standard query AAAA data2.wowzio.com
15	1.753389	129.128.5.233	142.244.164.24	DNS	Standard query response
16	1.754598	142.244.164.24	67.207.149.136	TCP	49745 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2 SACK_PERM=1

Ethernet

Like unicast, multicast and ARP; Ethernet is not restricted to the data link layer of the communication network model. Ethernet is defined in IEEE 802.3 standard, and has largely superseded other LAN networking protocols and technologies as of 1980. Systems communicating over Ethernet divide a stream of data into individual packets called frames. Each frame contains source and destination addresses and error-checking data so that damaged data can be detected and re-transmitted.

Ethernet is a family of protocols and techniques that operate at the physical and data link layers of the OSI reference model. Ethernet uses a protocol called CSMACD. This stands for Carrier

Sense, Multiple Access, and Collision Detection. The Multiple Access part means that every station is connected to a single copper wire (or a set of wires that are connected together to form a single data path). The Carrier Sense part says that before transmitting data, a station checks the wire to see if any other station is already sending something. If the LAN appears to be idle, then the station can begin to send data.

Common Performance Metrics

The following are common metrics to most network and systems administrators.

Latency: It can take a long time for a packet to be delivered across intervening networks. In reliable protocols where a receiver acknowledges delivery of each chunk of data, it is possible to measure this as round-trip time.

Packet loss: In some cases, intermediate devices in a network will lose packets. This may be due to errors, to overloading of the intermediate network, or to intentional discarding of traffic in order to enforce a particular service level.

Retransmission: When packets are lost in a reliable network, they are retransmitted. This incurs two delays: First, the delay from re-sending the data; and second, the delay resulting from waiting until the data is received in the correct order before forwarding it up the protocol stack.

Throughput: The amount of traffic a network can carry is measured as throughput, usually in terms such as kilobits per second. Throughput is analogous to the number of lanes on a highway, whereas latency is analogous to its speed limit.

These factors, and others (such as the performance of the network signaling on the end nodes, compression, encryption, concurrency, and so on) all affect the effective performance of a network. In some cases, the network may not work at all; in others, it may be slow or unusable. And because applications run over these networks, application performance suffers. Various intelligent solutions are available to ensure that traffic over the network is effectively managed to optimize performance for all users.

In summary, the OSI layered model of communication imposes a systematic approach to data transmission and receipt. This dictates that any single transmission’s performance is influenced by the internal algorithms implemented at layer.

Routing Protocols

Routing protocols are concerned with how the routers communicate with each other, and share routing information and approaches to select routes. This category of protocols are considered layer management protocols for the network layer, and may run over a variety of routed protocols such as TCP and UDP, and other non-transport layer protocols such as IS-IS which runs on CLNS.

This section explores the performance metrics and analysis of routing protocols; specifically, RIP, OSPF, IS IS and BGP will be addressed. Since the routing protocols differ in their philosophy, goals and applications than the routed protocols discusses in the previous sections, their performance metrics and complexities are more concerned with issues such as convergence times and overheads introduced by the control traffic.

To analyze the performance of the four protocols, the test lab setup introduced in the first section of this report has been used. The section begins with an overview of each of the four protocols to provide a clearer picture on their respective operations.

A dynamic routing protocol is responsible for path determination, routing updates and choosing the best path in a network (host node to destination node). Performance analysis of different routing protocols has been done based on different performance metrics like network convergence, router convergence, queuing delay and throughput and network bandwidth utilization, CPU utilization and routing traffic.

RIP

Overview

RIP (Routing Information Protocol) is categorized as an interior, distance vector routing protocol, and uses the Bellman-Ford single-source shortest path algorithm. One of the oldest routing protocols to be used, RIP has been in use for over 20 years; and currently has three versions: RIPv1, RIPv2 and RIPng.

It utilizes a mechanism known as routing by rumor in which each router broadcasts the whole of its routing table out of its active ports, and the receiving routers adopt that information and also pass it on to the others. One of the key performance issues with RIP is this periodic (30 second interval) update that includes the whole routing table; which, if it grows very large, can have dire consequences on bandwidth. Because of this and other reasons including the allowed 15 hops maximum, RIP is not a preferable routing protocol in today's network environments.

Performance metrics

The performance of the routing protocols is mainly based on the underlying algorithm it uses to select the best path. In this regard, RIP uses Bellman-Ford single-source shortest path algorithm, which has an algorithmic performance worst case scenario of **O(|V|*|E|)**, where V is akin to the number of routers and E the links between the routers. The inherent limitation of lack of scalability stems from the use of this algorithm.

In terms of bandwidth usage, the protocol sends routing updates every 30 seconds using UDP among other messages. Although the use of UDP eases the burden on the bandwidth (no acknowledgements and other overhead required as in TCP), the 30 second update intervals put an extra overhead on the available bandwidth. The severity of this is much felt on small bandwidth links such as serial connections. RIP is al so known for its slow convergence times and reliability issues due to the possibility of creating routing loops.

Traces

	Time	Source	Destination	Protocol	Info
1	0.000000	CompalCo_b3:c4:ab	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.2
2	0.421563	fe80::78cd:8e70:a5b0:	ff02::1:2	DHCPv6	Solicit XID: 0xfb2c1e CID: 0001000112810aa80016d4b3c4ab
3	1.578935	Cisco_96:f2:41	Cisco_96:f2:41	LOOP	Reply
4	11.579014	Cisco_96:f2:41	Cisco_96:f2:41	LOOP	Reply
5	21.579301	Cisco_96:f2:41	Cisco_96:f2:41	LOOP	Reply
6	21.587774	Cisco_96:f2:41	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Router_A Port ID: FastEthernet0/1
7	24.900174	192.168.1.1	255.255.255.255	RIPv1	Response
8	31.579725	Cisco_96:f2:41	Cisco_96:f2:41	LOOP	Reply
9	32.432690	fe80::78cd:8e70:a5b0:	ff02::1:2	DHCPv6	Solicit XID: 0xfb2c1e CID: 0001000112810aa80016d4b3c4ab
10	41.579647	Cisco_96:f2:41	Cisco_96:f2:41	LOOP	Reply
11	51.579897	Cisco_96:f2:41	Cisco_96:f2:41	LOOP	Reply
12	53.216619	CompalCo_b3:c4:ab	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.2
13	53.736620	192.168.1.1	255.255.255.255	RIPv1	Response
14	53.991627	CompalCo_b3:c4:ab	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.2
15	54.990047	CompalCo_b3:c4:ab	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.2
16	56.238436	CompalCo_b3:c4:ab	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.2
17	56.986904	CompalCo_b3:c4:ab	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.2
18	58.000859	CompalCo_b3:c4:ab	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.2
19	59.264862	CompalCo_b3:c4:ab	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.2
20	59.997702	CompalCo_b3:c4:ab	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.2
21	60.996065	CompalCo_b3:c4:ab	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.2
22	61.580014	Cisco_96:f2:41	Cisco_96:f2:41	LOOP	Reply
23	71.580270	Cisco_96:f2:41	Cisco_96:f2:41	LOOP	Reply
24	81.580438	Cisco_96:f2:41	Cisco_96:f2:41	LOOP	Reply
25	81.588886	Cisco_96:f2:41	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Router_A Port ID: FastEthernet0/1
26	83.301303	192.168.1.1	255.255.255.255	RIPv1	Response
27	86.932724	Cisco_96:f2:41	DEC-MOP-Remote-Consol	0x6002	DEC DNA Remote Console
28	91.580644	Cisco_96:f2:41	Cisco_96:f2:41	LOOP	Reply
29	101.580826	Cisco_96:f2:41	Cisco_96:f2:41	LOOP	Reply

On startup, RIP broadcasts a packet carrying a Request message out each RIP-enabled interface. The RIP process then enters a loop, listening for RIP Request or Response messages from other routers. Neighbors receiving the Request send a Response containing their route table.

Line number 7 shows an interface with IP address 192.168.1.1 sending a response. The request broadcast capture was missed in this diagram, but we can at least see the loop. This broadcast mechanism (destination of 255.255.255.255) is what makes RIP inefficient and not suitable for scalable internetworks.

Analysis

The above figure represents the RIP traces captures during the lab tests. It shows the routing loop issues the RIP protocol faces, which also results in unnecessary overhead. Just by looking at the entries in this trace file, it is apparent that almost have of them refer to routing loops.

OSPF

Overview

OSPF is an interior routing protocol categorized as a link-state protocol as it uses a link state routing algorithm (shortest path first). OSPF routes IP packets within a single Autonomous System (AS) by gathering link state information from routers and building a topology map out of it. Based on the topology map, the routing table to be provided to the Internet Protocol is determined.

Since OSPF is aware of the network topology, it detects any changes in the topology, and converges quickly (in seconds) – a feature that makes it more stable and reliable than RIP. OSPF uses Dijkstra's algorithm for shortest paths to find the shortest path tree for each route.

The OSPD packet header has a structure that shows the use of the Area concept in simplifying the management of the resources and traffic. Every OSPF packet starts with a common 24 byte header. This header contains all the necessary information to determine whether the packet should be accepted for further processing. This determination is described in Section 8.2 of the specification.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version #   |     Type      |         Packet length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Router ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Area ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |             AuType            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Explanations of the header elements:

Version #

The OSPF version number. This specification documents version 2 of the protocol.

Type

The OSPF packet types are as follows. The format of each of these packet types is described in a later section.

Type Description

1 Hello

2 Database Description

3 Link State Request

4 Link State Update

5 Link State Acknowledgment

Packet length

The length of the protocol packet in bytes. This length includes the standard OSPF header.

Router ID

The Router ID of the packet's source. In OSPF, the source and destination of a routing protocol packet are the two ends of an (potential) adjacency.

Area ID

A 32 bit number identifying the area that this packet belongs to. All OSPF packets are associated with a single area. Most travel a single hop only. Packets travelling over a virtual link are labeled with the backbone Area ID of 0.0.0.0.

Checksum

The standard IP checksum of the entire contents of the packet, starting with the OSPF packet header but excluding the 64-bit authentication field. This checksum is calculated as the 16-bit one's complement of the one's complement sum of all the 16-bit words in the packet, excepting the authentication field. If the packet's length is not an integral number of 16-bit words, the packet is padded with a byte of zero before check-summing.

AuType

Identifies the authentication scheme to be used for the packet. Authentication is discussed in Appendix D of the specification. Consult Appendix D for a list of the currently defined authentication types.

Authentication

A 64-bit field for use by the authentication scheme.

The protocol uses a data structure called the link-state database. A router has a separate link state database for every area to which it belongs. The link state database has been referred to elsewhere in the text as the topological database. All routers belonging to the same area have identical topological databases for the area.

Performance metrics and other features

OSPF has the following characteristics:

(1) Fast detection of changes in the topology and very fast reestablishment of routes without Loops, which translates to fast convergence times.

(2) Low overload, use updates that inform about changes on routes.

(3) Division of traffic by several equivalent routes.

(4) Routing according type of service.

(5) Use of multi-send in local area networks.

(6) Subnet and Super-net mask.

(7) Authentication

Traces

Exchange of the Hello packets for neighbor discovery.

	Time	Source	Destination	Protocol	Info
1	0.000000	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
2	9.999964	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
3	19.999834	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
4	29.999780	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
5	33.623236	10.0.0.14	224.0.0.5	OSPF	Hello Packet
6	37.951955	Cisco_96:f2:c0	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Router_A Port ID: FastEthernet0/0
7	39.999612	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
8	43.619636	10.0.0.14	224.0.0.5	OSPF	Hello Packet
9	49.999605	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
10	53.619566	10.0.0.14	224.0.0.5	OSPF	Hello Packet
11	59.999387	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
12	63.619473	10.0.0.14	224.0.0.5	OSPF	Hello Packet
13	69.999398	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
14	73.619558	10.0.0.14	224.0.0.5	OSPF	Hello Packet
15	79.999205	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
16	83.619254	10.0.0.14	224.0.0.5	OSPF	Hello Packet
17	90.009494	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
18	93.619150	10.0.0.14	224.0.0.5	OSPF	Hello Packet
19	97.951332	Cisco_96:f2:c0	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Router_A Port ID: FastEthernet0/0
20	99.999012	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
21	103.619042	10.0.0.14	224.0.0.5	OSPF	Hello Packet
22	109.999031	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
23	113.618945	10.0.0.14	224.0.0.5	OSPF	Hello Packet
24	119.998865	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
25	123.618829	10.0.0.14	224.0.0.5	OSPF	Hello Packet
26	129.998768	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
27	133.618743	10.0.0.14	224.0.0.5	OSPF	Hello Packet
28	139.998617	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
29	142.163256	10.0.0.13	224.0.0.5	OSPF	Hello Packet
30	142.164661	10.0.0.14	10.0.0.13	OSPF	Hello Packet

	Time	Source	Destination	Protocol	Info
28	139.998617	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
29	142.163256	10.0.0.13	224.0.0.5	OSPF	Hello Packet
30	142.164661	10.0.0.14	10.0.0.13	OSPF	Hello Packet
31	142.165311	10.0.0.13	10.0.0.14	OSPF	DB Description
32	142.165407	10.0.0.13	10.0.0.14	OSPF	Hello Packet
33	142.167120	10.0.0.14	10.0.0.13	OSPF	DB Description
34	142.167538	10.0.0.13	10.0.0.14	OSPF	DB Description
35	142.169317	10.0.0.14	10.0.0.13	OSPF	DB Description
36	142.169647	10.0.0.13	10.0.0.14	OSPF	DB Description
37	142.171171	10.0.0.14	10.0.0.13	OSPF	DB Description
38	142.171411	10.0.0.13	10.0.0.14	OSPF	LS Request
39	142.171555	10.0.0.13	10.0.0.14	OSPF	DB Description
40	142.173555	10.0.0.14	10.0.0.13	OSPF	LS Update
41	142.662337	10.0.0.13	224.0.0.5	OSPF	LS Update
42	142.699378	10.0.0.14	224.0.0.5	OSPF	LS Update
43	142.806583	10.0.0.13	224.0.0.5	OSPF	LS Update
44	143.618677	10.0.0.14	224.0.0.5	OSPF	Hello Packet
45	144.670158	10.0.0.13	224.0.0.5	OSPF	LS Acknowledge
46	145.162703	10.0.0.14	224.0.0.5	OSPF	LS Acknowledge
47	149.998575	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
48	152.162103	10.0.0.13	224.0.0.5	OSPF	Hello Packet
49	153.618577	10.0.0.14	224.0.0.5	OSPF	Hello Packet
50	157.950776	Cisco_96:f2:c0	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Router_A Port ID: FastEthernet0/0
51	159.998414	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
52	162.162133	10.0.0.13	224.0.0.5	OSPF	Hello Packet
53	163.618502	10.0.0.14	224.0.0.5	OSPF	Hello Packet
54	169.998414	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
55	172.162244	10.0.0.13	224.0.0.5	OSPF	Hello Packet
56	173.106252	192.168.3.2	224.0.0.22	IGMP	V3 Membership Report / Leave group 224.0.0.252

LSA updates

No.	Time	Source	Destination	Protocol	Info
79	184.399848	CompalCo_b3:c4:ab	Broadcast	ARP	Who has 192.168.3.1? Tell 192.168.3.2
80	189.998190	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
81	192.167681	10.0.0.13	224.0.0.5	OSPF	Hello Packet
82	193.618204	10.0.0.14	224.0.0.5	OSPF	Hello Packet
83	200.006868	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
84	202.162274	10.0.0.13	224.0.0.5	OSPF	Hello Packet
85	203.618070	10.0.0.14	224.0.0.5	OSPF	Hello Packet
86	209.998050	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
87	212.162307	10.0.0.13	224.0.0.5	OSPF	Hello Packet
88	213.617949	10.0.0.14	224.0.0.5	OSPF	Hello Packet
89	215.836150	10.0.0.13	224.0.0.5	OSPF	LS Update
90	217.802841	10.0.0.14	224.0.0.5	OSPF	LS Update
91	217.950196	Cisco_96:f2:c0	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Router_A Port ID: FastEthernet0/0
92	218.337956	10.0.0.14	224.0.0.5	OSPF	LS Acknowledge
93	219.997872	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
94	220.302384	10.0.0.13	224.0.0.5	OSPF	LS Acknowledge
95	222.162458	10.0.0.13	224.0.0.5	OSPF	Hello Packet
96	223.617927	10.0.0.14	224.0.0.5	OSPF	Hello Packet
97	229.997832	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
98	232.162491	10.0.0.13	224.0.0.5	OSPF	Hello Packet
99	233.617797	10.0.0.14	224.0.0.5	OSPF	Hello Packet
100	234.522857	10.0.0.14	224.0.0.5	OSPF	LS Update
101	234.557925	10.0.0.14	224.0.0.5	OSPF	LS Update
102	237.022351	10.0.0.13	224.0.0.5	OSPF	LS Acknowledge
103	239.997707	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
104	242.162512	10.0.0.13	224.0.0.5	OSPF	Hello Packet
105	243.618076	10.0.0.14	224.0.0.5	OSPF	Hello Packet
106	249.997596	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
107	252.162541	10.0.0.13	224.0.0.5	OSPF	Hello Packet
108	253.617589	10.0.0.14	224.0.0.5	OSPF	Hello Packet

Analysis

The first figure in this group shows the exchange of the Hello packets. Routers periodically send hello packets on all interfaces, including virtual links, to establish and maintain neighbor relationships. Hello packets are multicast on physical networks that have a multicast or broadcast capability, which enables dynamic discovery of neighboring routers. (On non-broadcast networks, dynamic neighbor discovery is not possible, so you must configure all neighbors statically using the neighbor statement.)

Hello packets consist of the OSPF header plus the following fields:

- Network mask—Network mask associated with the interface.
- Hello interval—how often the router sends hello packets. All routers on a shared network must use the same hello interval. You configure this interval with the hello-interval statement.
- Options—Optional capabilities of the router.
- Router priority—the router's priority to become the designated router. You can configure this value with the priority statement.
- Router dead interval—how long the router waits without receiving any OSPF packets from a router before declaring that router to be down. All routers on a shared network

must use the same router dead interval. You can configure this value with the dead-interval statement.

- Designated router—IP address of the designated router.
- Backup designated router—IP address of the backup designated router.
- Neighbor—IP addresses of the routers from which valid hello packets have been received within the time specified by the router dead interval.

The next two figures indicate the process of building the link state database and the exchange of LSAs. Database Description Packets

When initializing an adjacency, OSPF exchanges database description packets, which describe the contents of the topological database. These packets consist of the OSPF header, packet sequence number, and the link-state advertisement's header.

Link-State Request Packets

When a router detects that portions of its topological database are out of date, it sends a link-state request packet to a neighbor requesting a precise instance of the database. These packets consist of the OSPF header plus fields that uniquely identify the database information that the router is seeking.

Link-State Update Packets

Link-state update packets carry one or more link-state advertisements one hop farther from their origin. The router multicasts (floods) these packets on physical networks that support multicast or broadcast mode. The router acknowledges all link-state update packets and, if retransmission is necessary, sends the retransmitted advertisements unicast.

Link-state update packets consist of the OSPF header plus the following fields:

- Number of advertisements—Number of link-state advertisements included in this packet.
- Link-state advertisements—the link-state advertisements themselves.

Link-State Acknowledgment Packets

The router sends link-state acknowledgment packets in response to link-state update packets to verify that the update packets have been received successfully. A single acknowledgment packet can include responses to multiple update packets.

Link-state acknowledgment packets consist of the OSPF header plus the link-state advertisement header.

Link-State Advertisement Packet Types

Link-state request, link-state update, and link-state acknowledgment packets are used to reliably flood link-state advertisement packets. OSPF sends the following types of link-state advertisements:

- Router link advertisements—are sent by all routers to describe the state and cost of the router's links to the area. These link-state advertisements are flooded throughout a single area only.
- Network link advertisements—are sent by designated routers to describe all the routers attached to the network. These link-state advertisements are flooded throughout a single area only.
- Summary link advertisements—are sent by area border routers to describe the routes that they know about in other areas. There are two types of summary link advertisements: those used when the destination is an IP network, and those used when the destination is an AS boundary router. Summary link advertisements describe inter-area routes; that is, routes to destinations outside the area but within the AS. These link-state advertisements are flooded throughout the advertisement's associated areas.
- AS external link advertisement—are sent by AS boundary routers to describe external routes that they know about. These link-state advertisements are flooded throughout the AS (except for stub areas).

Each link-state advertisement type describes a portion of the OSPF routing domain. All link-state advertisements are flooded throughout the AS.

Each link-state advertisement packet begins with a common 20-byte header.

OSPF Metrics

The primary OSPF metric is *cost*, which Cisco and other manufacturers configure to be inversely proportional to the bandwidth of that interface. Lower cost means a faster interface and shorter end-to-end transmission times and thus the shortest path. The bandwidth of an interface is indirectly passed on with the OSPF route in the form of an additive 'cost' metric to indicate the speed of the entire path to the destination via the local interface link. Because OSPF is a link state protocol, higher speed links have a lower cost than low speed links.

IS-IS

An Overview

IS IS is the defacto standard for large service provider networks, and is defined in RFC 1142. It is an interior gateway protocol, which means it is designed for use within an administrative domain or network. IS-IS is a link-state routing protocol, operating by reliably flooding link state information throughout a network of routers. Each IS-IS router independently builds a database of the network's topology, aggregating the flooded network information. Like the OSPF protocol, IS-IS uses Dijkstra's algorithm for computing the best path through the network. Packets (datagrams) are then forwarded, based on the computed ideal path, through the network to the destination.

While OSPF is natively built to route IP and is itself a Layer 3 protocol that runs on top of IP, IS-IS is natively an OSI network layer protocol (it is at the same layer as CLNS). The widespread adoption of IP worldwide may have contributed to OSPF's popularity. IS-IS does not use IP to carry routing information messages. IS-IS is neutral regarding the type of network addresses for which it can route. OSPF, on the other hand, was designed for IPv4. This allowed IS-IS to be easily used to support IPv6. To operate with IPv6 networks, the OSPF protocol was rewritten in OSPF v3 (as specified in RFC 2740).

IS-IS routers build a topological representation of the network. This map indicates the subnets which each IS-IS router can reach, and the lowest-cost (shortest) path to a subnet is used to forward traffic.

OSPF has a larger set of extensions and optional features. However IS-IS is less chatty and can scale to support larger networks. Given the same set of resources, IS-IS can support more routers in an area than OSPF. This has contributed to IS-IS as an ISP-scale protocol.

Performance metrics

The IS-IS routing protocol is a link-state protocol, as opposed to distance-vector protocols such as Interior Gateway Routing Protocol (IGRP) and Routing Information Protocol (RIP). Link-state offers several advantages over distance-vector protocols. It is faster converging, supports much larger internetworks, and is less susceptible to routing loops. Features of IS-IS include:

- Hierarchical routing
- Classless behavior
- Rapid flooding of new information
- Fast Convergence
- Very scalable
- Flexible timer tuning

IS IS uses the following metrics for its routing operations:

- Default metric (required): cost—No automatic calculation of the metric for IS-IS takes place, compared to some routing protocols that calculate the link metric automatically based on bandwidth (OSPF) or bandwidth/delay (EIGRP). Using narrow metrics (the default), an interface cost is between 1 and 63 (a 6-bit metric value). All links use the metric of 10 by default. The total cost to a destination is the sum of the costs on all outgoing interfaces along a particular path from the source to the destination, and the least-cost paths are preferred.
- Delay, expense, and error (optional)—these metrics are intended for use in type of service (ToS) routing. These could be used to calculate alternative routes referring to the DTR (delay, throughput, and reliability) bits in the IP ToS field.

Traces

	Time	Source	Destination	Protocol	Info
1	0.000000	Cisco_96:f2:c0	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Router_A Port ID: FastEthernet0/0
2	1.264186	Cisco_96:f2:c0	ISIS-all-level-1-IS's	ISIS	L1 HEL[illegible].0000.0001
3	2.047603	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
4	4.652176	Cisco_96:f2:c0	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0001
5	10.112093	Cisco_96:f2:c0	ISIS-all-level-1-IS's	ISIS	L1 HELLO, System-ID: 0000.0000.0001
6	12.047561	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
7	13.380026	Cisco_96:f2:c0	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0001
8	19.915987	Cisco_96:f2:c0	ISIS-all-level-1-IS's	ISIS	L1 HELLO, System-ID: 0000.0000.0001
9	21.431962	Cisco_96:f2:c0	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0001
10	22.047397	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
11	29.155904	Cisco_96:f2:c0	ISIS-all-level-1-IS's	ISIS	L1 HELLO, System-ID: 0000.0000.0001
12	29.940050	Cisco_96:f2:c0	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0001
13	32.047341	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
14	37.491824	Cisco_96:f2:c0	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0001
15	38.715801	Cisco_96:f2:c0	ISIS-all-level-1-IS's	ISIS	L1 HELLO, System-ID: 0000.0000.0001
16	42.047224	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
17	46.379746	Cisco_96:f2:c0	ISIS-all-level-1-IS's	ISIS	L1 HELLO, System-ID: 0000.0000.0001
18	47.179762	Cisco_96:f2:c0	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0001
19	52.047181	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
20	54.863642	Cisco_96:f2:c0	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0001
21	56.203630	Cisco_96:f2:c0	ISIS-all-level-1-IS's	ISIS	L1 HELLO, System-ID: 0000.0000.0001
22	59.999406	Cisco_96:f2:c0	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Router_A Port ID: FastEthernet0/0
23	62.046993	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
24	64.643560	Cisco_96:f2:c0	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0001
25	65.939670	Cisco_96:f2:c0	ISIS-all-level-1-IS's	ISIS	L1 HELLO, System-ID: 0000.0000.0001
26	72.046979	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
27	72.795462	Cisco_96:f2:c0	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0001
28	74.395416	Cisco_96:f2:c0	ISIS-all-level-1-IS's	ISIS	L1 HELLO, System-ID: 0000.0000.0001
29	82.046788	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
30	82.099371	Cisco_96:f2:c0	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0001

	Time	Source	Destination	Protocol	Info
90	180.192842	Cisco_c0:16:78	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0004
91	180.346392	Cisco_96:f2:c0	ISIS-all-level-2-IS's	ISIS	L2 HEL[illegible].0000.0001
92	180.888643	Cisco_c0:16:78	ISIS-all-level-2-IS's	ISIS	L2 LSP, LSP-ID: 0000.0000.0004.00-00, Sequence: 0x00000005, Lifetime: 1199s
93	180.905943	Cisco_96:f2:c0	ISIS-all-level-2-IS's	ISIS	L2 LSP, LSP-ID: 0000.0000.0001.00-00, Sequence: 0x00000006, Lifetime: 1199s
94	181.849292	Cisco_c0:16:78	ISIS-all-level-1-IS's	ISIS	L1 HELLO, System-ID: 0000.0000.0004
95	182.045811	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
96	183.092857	Cisco_c0:16:78	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0004
97	185.064904	Cisco_c0:16:78	ISIS-all-level-1-IS's	ISIS	L1 HELLO, System-ID: 0000.0000.0004
98	185.868696	Cisco_c0:16:78	ISIS-all-level-2-IS's	ISIS	L2 CSNP, Source-ID: 0000.0000.0004.00, Start LSP-ID: 0000.0000.0000.00-00, End LSP-ID: ffff.ffff.
99	186.020978	Cisco_c0:16:78	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0004
100	186.568723	Cisco_c0:16:78	ISIS-all-level-1-IS's	ISIS	L1 CSNP, Source-ID: 0000.0000.0004.00, Start LSP-ID: 0000.0000.0000.00-00, End LSP-ID: ffff.ffff.
101	188.332926	Cisco_c0:16:78	ISIS-all-level-1-IS's	ISIS	L1 HELLO, System-ID: 0000.0000.0004
102	188.590322	Cisco_96:f2:c0	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0001
103	188.726292	Cisco_96:f2:c0	ISIS-all-level-1-IS's	ISIS	L1 HELLO, System-ID: 0000.0000.0001
104	189.224917	Cisco_c0:16:78	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0004
105	191.468980	Cisco_c0:16:78	ISIS-all-level-1-IS's	ISIS	L1 HELLO, System-ID: 0000.0000.0004
106	192.045773	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
107	192.544903	Cisco_c0:16:78	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0004
108	194.484929	Cisco_c0:16:78	ISIS-all-level-1-IS's	ISIS	L1 HELLO, System-ID: 0000.0000.0004
109	194.761192	Cisco_c0:16:78	ISIS-all-level-2-IS's	ISIS	L2 CSNP, Source-ID: 0000.0000.0004.00, Start LSP-ID: 0000.0000.0000.00-00, End LSP-ID: ffff.ffff.
110	195.120917	Cisco_c0:16:78	ISIS-all-level-2-IS's	ISIS	L2 HELLO, System-ID: 0000.0000.0004
111	195.355100	192.168.3.2	224.0.0.22	IGMP	V3 Membership Report / Leave group 224.0.0.252
112	195.393543	CompalCo_b3:c4:ab	Broadcast	ARP	Who has 192.168.3.1? Tell 192.168.3.2
113	195.397190	192.168.3.2	224.0.0.22	IGMP	V3 Membership Report / Join group 224.0.0.252 for any sources
114	195.397570	192.168.3.2	224.0.0.22	IGMP	V3 Membership Report / Leave group 224.0.0.252
115	195.429978	192.168.3.2	224.0.0.22	IGMP	V3 Membership Report / Join group 224.0.0.252 for any sources
116	195.432474	192.168.3.2	224.0.0.252	LLMNR	Standard query ANY PHM780
117	195.465378	192.168.3.2	224.0.0.22	IGMP	V3 Membership Report / Join group 224.0.0.252 for any sources
118	195.532376	192.168.3.2	224.0.0.252	LLMNR	Standard query ANY PHM780
119	195.725979	CompalCo_b3:c4:ab	Broadcast	ARP	Who has 192.168.3.1? Tell 192.168.3.2

Analysis

The screen capture of the IS IS traces shows the exchange of the Hello packets in the process of forming adjacencies. The overhead involved in establishing the adjacencies and sharing the updates is part of the design of IS IS. The performance of the protocol may be improved by adopting the IS IS configuration best practices, and setting the update intervals in a way that does not compromise the integrity of the routing information.

BGP

Overview

The Border Gateway Protocol (BGP) is an exterior gateway protocol classified as a path vector routing protocol. It is the protocol used in the Internet backbone/core, and uses different set of criteria than interior gateway protocols to select routes including but not limited to path, routing policies and rules. Most Internet service providers must use BGP to establish routing between one another (especially if they are multi-homed). Therefore, even though most Internet users do not use it directly, BGP is one of the most important protocols of the Internet.

BGP neighbors, peers are established by manual configuration between routers to create a TCP session on port 179. A BGP speaker will periodically send 19-byte keep-alive messages to maintain the connection (every 60 seconds by default). Among routing protocols, BGP is unique in using TCP as its transport protocol. A BGP-enabled router uses a simple finite state machine to make its peering decisions with its neighbors.

Performance metrics and issues

Internal BGP scalability

An autonomous system with internal BGP (IBGP) must have all of its IBGP peers connect to each other in a full mesh (where everyone speaks to everyone directly). This full-mesh configuration requires that each router maintain a session to every other router. In large networks, this number of sessions may degrade performance of routers, due either to a lack of memory, or too much CPU process requirements.

Route reflectors and confederations both reduce the number of IBGP peers to each router and thus reduce processing overhead. Route reflectors are a pure performance-enhancing technique, while confederations also can be used to implement more fine-grained policy.

Route reflectors reduce the number of connections required in an AS. A single router (or two for redundancy) can be made a route reflector: other routers in the AS need only be configured as peers to them.

Confederations are sets of autonomous systems. In common practice, only one of the confederation AS numbers is seen by the Internet as a whole. Confederations are used in very large networks where a large AS can be configured to encompass smaller more manageable internal ASs.

Confederations can be used in conjunction with route reflectors. Both confederations and route reflectors can be subject to persistent oscillation unless specific design rules, affecting both BGP and the interior routing protocol, are followed.

However, these alternatives can introduce problems of their own, including the following:

- route oscillation
- sub-optimal routing
- increase of BGP convergence time

Additionally, route reflectors and BGP confederations were not designed to ease BGP router configuration. Nevertheless, these are common tools for experienced BGP network architects. These tools may be combined, for example, as a hierarchy of route reflectors.

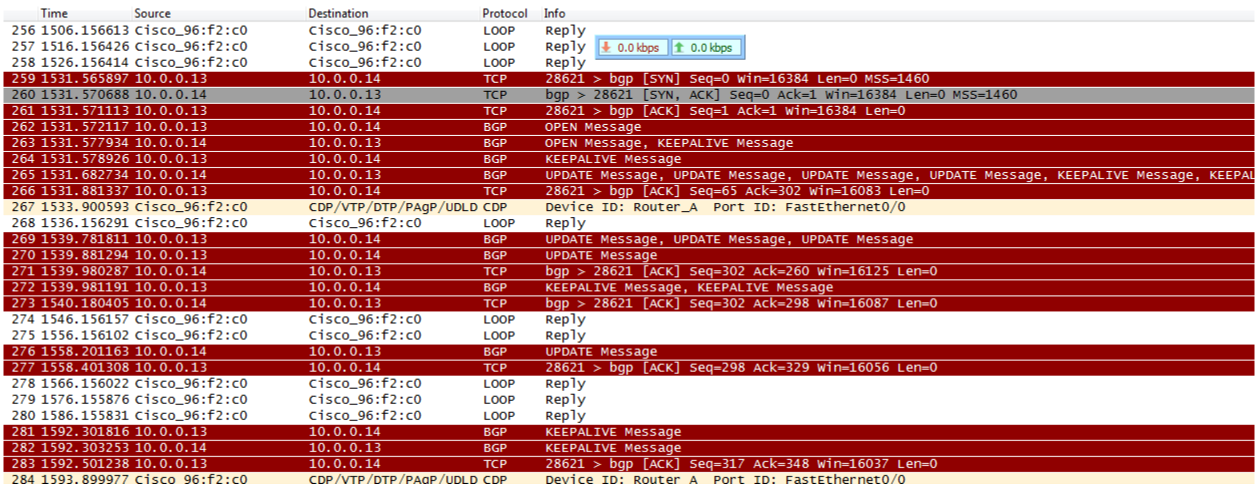
Instability

The routing tables managed by a BGP implementation are adjusted continually to reflect actual changes in the network, such as links breaking and being restored or routers going down and coming back up. In the network as a whole it is normal for these changes to happen almost continuously, but for any particular router or link changes are supposed to be relatively infrequent. If a router is misconfigured or mismanaged then it may get into a rapid cycle between down and up states. This pattern of repeated withdrawal and re-announcement, known as route flapping, can cause excessive activity in all the other routers that know about the broken link, as the same route is continuously injected and withdrawn from the routing tables. The BGP design is such that delivery of traffic may not function while routes are being updated. On the Internet, a BGP routing change may cause outages for several minutes.

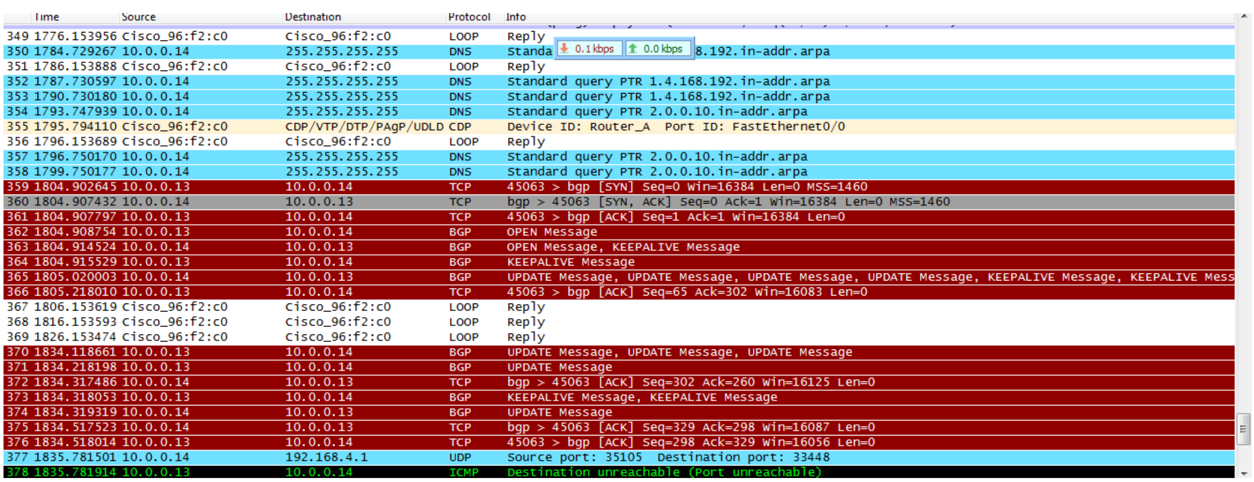
- Complexity (as compared to other protocols), overhead, convergence times, security.

- path attributes describe the characteristics of paths, and are used in the process of route selection
- Metrics to calculate shortest path
- Message types, Transport protocol (TCP), port number
- Fast Convergence time, timers
- Reliability (Routing loops)

Traces



	Time	Source	Destination	Protocol	Info
256	1506.156613	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
257	1516.156426	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
258	1526.156414	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
259	1531.565897	10.0.0.13	10.0.0.14	TCP	28621 > bgp [SYN] Seq=0 Win=16384 Len=0 MSS=1460
260	1531.570688	10.0.0.14	10.0.0.13	TCP	bgp > 28621 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460
261	1531.571113	10.0.0.13	10.0.0.14	TCP	28621 > bgp [ACK] Seq=1 Ack=1 Win=16384 Len=0
262	1531.572117	10.0.0.13	10.0.0.14	BGP	OPEN Message
263	1531.577934	10.0.0.14	10.0.0.13	BGP	OPEN Message, KEEPALIVE Message
264	1531.578926	10.0.0.13	10.0.0.14	BGP	KEEPALIVE Message
265	1531.682734	10.0.0.14	10.0.0.13	BGP	UPDATE Message, UPDATE Message, UPDATE Message, UPDATE Message, KEEPALIVE Message, KEEPAL
266	1531.881337	10.0.0.13	10.0.0.14	TCP	28621 > bgp [ACK] Seq=65 Ack=302 Win=16083 Len=0
267	1533.900593	Cisco_96:f2:c0	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Router_A Port ID: FastEthernet0/0
268	1536.156291	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
269	1539.781811	10.0.0.13	10.0.0.14	BGP	UPDATE Message, UPDATE Message, UPDATE Message
270	1539.881294	10.0.0.13	10.0.0.14	BGP	UPDATE Message
271	1539.980287	10.0.0.14	10.0.0.13	TCP	bgp > 28621 [ACK] Seq=302 Ack=260 Win=16125 Len=0
272	1539.981191	10.0.0.13	10.0.0.14	BGP	KEEPALIVE Message, KEEPALIVE Message
273	1540.180405	10.0.0.14	10.0.0.13	TCP	bgp > 28621 [ACK] Seq=302 Ack=298 Win=16087 Len=0
274	1546.156157	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
275	1556.156102	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
276	1558.201163	10.0.0.14	10.0.0.13	BGP	UPDATE Message
277	1558.401308	10.0.0.13	10.0.0.14	TCP	28621 > bgp [ACK] Seq=298 Ack=329 Win=16056 Len=0
278	1566.156022	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
279	1576.155876	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
280	1586.155831	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
281	1592.301816	10.0.0.13	10.0.0.14	BGP	KEEPALIVE Message
282	1592.303253	10.0.0.14	10.0.0.13	BGP	KEEPALIVE Message
283	1592.501238	10.0.0.13	10.0.0.14	TCP	28621 > bgp [ACK] Seq=317 Ack=348 Win=16037 Len=0
284	1593.899977	Cisco_96:f2:c0	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Router_A Port ID: FastEthernet0/0



	Time	Source	Destination	Protocol	Info
349	1776.153956	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
350	1784.729267	10.0.0.14	255.255.255.255	DNS	Standa[illegible]8.192.in-addr.arpa
351	1786.153888	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
352	1787.730597	10.0.0.14	255.255.255.255	DNS	Standard query PTR 1.4.168.192.in-addr.arpa
353	1790.730180	10.0.0.14	255.255.255.255	DNS	Standard query PTR 1.4.168.192.in-addr.arpa
354	1793.747939	10.0.0.14	255.255.255.255	DNS	Standard query PTR 2.0.0.10.in-addr.arpa
355	1795.794110	Cisco_96:f2:c0	CDP/VTP/DTP/PAgP/UDLD	CDP	Device ID: Router_A Port ID: FastEthernet0/0
356	1796.153689	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
357	1796.750170	10.0.0.14	255.255.255.255	DNS	Standard query PTR 2.0.0.10.in-addr.arpa
358	1799.750177	10.0.0.14	255.255.255.255	DNS	Standard query PTR 2.0.0.10.in-addr.arpa
359	1804.902645	10.0.0.13	10.0.0.14	TCP	45063 > bgp [SYN] Seq=0 Win=16384 Len=0 MSS=1460
360	1804.907432	10.0.0.14	10.0.0.13	TCP	bgp > 45063 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460
361	1804.907797	10.0.0.13	10.0.0.14	TCP	45063 > bgp [ACK] Seq=1 Ack=1 Win=16384 Len=0
362	1804.908754	10.0.0.13	10.0.0.14	BGP	OPEN Message
363	1804.914524	10.0.0.14	10.0.0.13	BGP	OPEN Message, KEEPALIVE Message
364	1804.915529	10.0.0.13	10.0.0.14	BGP	KEEPALIVE Message
365	1805.020003	10.0.0.14	10.0.0.13	BGP	UPDATE Message, UPDATE Message, UPDATE Message, UPDATE Message, KEEPALIVE Message, KEEPALIVE Mess
366	1805.218010	10.0.0.13	10.0.0.14	TCP	45063 > bgp [ACK] Seq=65 Ack=302 Win=16083 Len=0
367	1806.153619	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
368	1816.153593	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
369	1826.153474	Cisco_96:f2:c0	Cisco_96:f2:c0	LOOP	Reply
370	1834.118661	10.0.0.13	10.0.0.14	BGP	UPDATE Message, UPDATE Message, UPDATE Message
371	1834.218198	10.0.0.13	10.0.0.14	BGP	UPDATE Message
372	1834.317486	10.0.0.14	10.0.0.13	TCP	bgp > 45063 [ACK] Seq=302 Ack=260 Win=16125 Len=0
373	1834.318053	10.0.0.13	10.0.0.14	BGP	KEEPALIVE Message, KEEPALIVE Message
374	1834.319319	10.0.0.14	10.0.0.13	BGP	UPDATE Message
375	1834.517523	10.0.0.14	10.0.0.13	TCP	bgp > 45063 [ACK] Seq=329 Ack=298 Win=16087 Len=0
376	1834.518014	10.0.0.13	10.0.0.14	TCP	45063 > bgp [ACK] Seq=298 Ack=329 Win=16056 Len=0
377	1835.781501	10.0.0.14	192.168.4.1	UDP	Source port: 35105 Destination port: 33448
378	1835.781914	10.0.0.13	10.0.0.14	ICMP	Destination unreachable (Port unreachable)

Analysis

The above traces demonstrate the operations of the BGP protocol. When a BGP router first comes up on the Internet, either for the first time or after being turned off, it establishes connections with the other BGP routers with which it directly communicates. The first thing it does is download the entire routing table of each neighbouring router. After that it only exchanges much shorter update messages with other routers.

BGP routers send and receive update messages to indicate a change in the preferred path to reach a computer with a given IP address. If the router decides to update its own routing tables because this new path is better, then it will subsequently propagate this information to all of the other neighbouring BGP routers to which it is connected, and they will in turn decide whether to update their own tables and propagate the information further.

BGP uses the TCP/IP protocol on port 179 to establish connections. It has strong security features, including the incorporation of a digital signature in all communications between BGP routers.

Each BGP router contains a Routing Information Base (RIB) that contains the routing information maintained by that router. The RIB contains three types of information:

- Ad-RIBs-In. The unedited routing information sent by neighbouring routers.
- Loc-RIB. The actual routing information the router uses, developed from Adj-RIBs-In.
- Adj-RIBs-Out. The information the router chooses to send to neighbouring routers.
- BGP routers exchange information using four types of messages:
- Open. Used to open an initial connection with a neighbouring router.
- Update. These messages do most of the work, exchanging routing information between neighbouring routers, and contain one of the following pieces of information.
- Withdrawn routes. The IP addresses of computers that the router no longer can route messages to.
- Paths. A new preferred route for an IP address. This path consists of two pieces of information -- the IP address, and the address of the next router in the path that is used to route messages destined for that address.
- Notification. Used to indicate errors, such as an incorrect or unreadable message received, and are followed by an immediate close of the connection with the neighbouring router.
- Keepalive. Each BGP router sends a 19 byte Keepalive message to each neighboring router to let them know that it is still operational about every 30 seconds, and no more often than every three seconds. If any router does not receive a Keepalive message from a neighboring router within a set amount of time, it closes its connection with that router,

and removes it from its Routing Information Base, repairing what it perceives as damage to the network.

- Routing messages are the highest precedence traffic on the Internet, and each BGP router gives them first priority over all other traffic. This makes sense -- if routing information can't make it through, then nothing else will.

Summary

The layered models of network architecture (both OSI and TCP/IP) make not only the communications process, which is a complicated processes when seen as one monolithic task, a simple, manageable task but also the process of troubleshooting the points of hiccups within the communication system. Looking at the whole issue of encapsulation and decapsulation, the hand-over of data from one layer to the next and back, and the overheads involved, it is perfectly reasonable to question the downside of the layered approach to data communications and networks in terms of the impact of this separation of tasks among the various layers on the performance of internetworking protocols and the overall system.

As a first step to look into the performance issues of the internetworking protocols, it is necessary to identify the metrics used, and evaluate the network performance against these metrics as criteria. This is difficult task in itself because:

- The layered approach to communications system design and internetworking models dictates that the function of each specific protocols span not all the layers of the model
- As a result, each protocol has a specific function and role in the communication process and/or system
- And because each protocol has a specific function and role, which will might call for a different set of algorithms and data structures, each protocol will probably have a different set of metrics that capture its performance criteria

To overcome this challenge, it was seen crucial that the sample set of protocols used in the study should be selected on the basis of their functions in the data communication process; and that each of the protocols be scrutinized separately. A lab environment, with the network diagram shown in the introduction, has been setup to try out and inspect the functions of and mechanisms used by each of the protocols that have been in this study, which included the following (grouped according to their functions and the layers of the OSI reference model they affect:

- Ping, traceroute, ICMP and DHCP
- TCP and UDP

- HTTP, SSH, Telnet and FTP
- ARP and RARP
- Unicast, Multicast and Ethernet
- RIP, OSPF, IS-IS, and BGP

Each of the protocols has been configured in the test-bed network, and the traces of the test runs have been captured using the Open Source Wireshark.

A revision of the generic findings of the study

The general performance metrics and their underlying assumptions

The purpose of this study was mainly to identify the various performance metrics of internetworking protocols using a select set of the most commonly used internetworking protocols as sample. If the internetworking model and the communication process is taken as one monolithic task, the performance metrics for the internetwork and data communication system would include the common metrics that have now become more or less marketing buzzwords such as throughput, delay and round-trip times. However, taken separately, and studied as individual autonomous systems, the internetwork protocols will have their own performance criteria and metrics depending on the underlying technologies and algorithms. An example of this would be the difference between the metrics of the different routing protocols (i.e. RIP and OSPF); although they both have the same functionality in the communications network, they have their own unique metrics.

The study also emphasizes that the overall performance of the internetwork communication is the 'sum' of the communications protocols that participate in the communication session. Since some protocols at a certain layer might use the services of another protocol in another layer, depending on the direction of this communication (read encapsulating or otherwise), the performance of the using protocol is also impacted by that of the protocols it depends to do its task or play its role in the communication process.

Next steps on how to improve this work: if there is anything this indicates, it is the need to build performance metrics identification and testing framework that encompasses all internetwork protocols. Most of the RFCs and papers that have been studied throughout this experiment either focused on service performance, and other general metrics such as throughput and delay; however, it appears that throughput and delay and the availability of services is not sufficient to measure the performance of all the protocols. There should be other criteria that can also equally apply to all the internetworking protocols either in use today or will be invented in the future. This is an area I would like to explore more, as it is, in my opinion, an interesting research area.

Bibliography

1. Ethernet Network Performance Differentiation , Ethernet Technology Summit, Session 202 on Broadband Networks, February 24, 2011 http://www.ethernetsummit.com/English/Collaterals/Proceedings/2011/20110224_S202_Gum.pdf.
2. Address Resolution Protocol - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Address_Resolution_Protocol.
3. ARP and RARP Address Translation, Comptechdocs Repository, http://www.comptechdoc.org/independent/networking/guide/netarp.html.
4. Bandwidth, Packets Per Second, and Other Network Performance Metrics - Cisco Systems, http://www.cisco.com/web/about/security/intelligence/network_performance_metrics.html.
5. BGP, Border Gateway Protocol, http://www.livinginternet.com/i/iw_route_egp_bgp.htm.
6. Birds-Eye.Net Carrier-Class DHCP Testing Setup, http://www.birds-eye.net/technical_archive/carrier_class_dhcp_testing.htm.
7. Border Gateway Protocol - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Border_Gateway_Protocol.
8. Border Gateway Protocol - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Border_Gateway_Protocol.
9. Border Gateway Protocol: Conformance and Performance Testing :: White Papers :: Ixia - Enabling a Converged World, http://www.ixiacom.com/library/white_papers/display?skey=bgp.
10. CLNS - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/CLNS.
11. DHCP_Overview.pdf, http://www.cisco.com/en/US/docs/ios-xml/ios/ipaddr_dhcp/configuration/15-1mt/DHCP_Overview.pdf.
12. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.3.540&rep=rep1&type=pdf.
13. Ethernet - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Ethernet.
14. High Performance Enabled SSH/SCP [PSC], http://www.psc.edu/networking/projects/hpn-ssh/.
15. HTTP Compression, http://www.http-compression.com/.
16. HTTP Performance Overview, http://www.w3.org/Protocols/HTTP/Performance/.
17. IEEE 802.3 ETHERNET, http://www.ieee802.org/3/.
18. Intermediate System-to-Intermediate System Protocol [IP Routing] - Cisco Systems, http://www.cisco.com/en/US/products/ps6599/products_white_paper09186a00800a3e6f.shtml.
19. Internet Control Message Protocol - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Internet_Control_Message_Protocol.
20. Investigations and Performance Evaluation of Dynamic Routing Protocol With New Proposed Protocol for WAN, from MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY - White Papers, Webcasts and Case Studies - ZDNet, http://whitepapers.zdnet.com/abstract.aspx?docid=3240867.
21. IS-IS - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/IS-IS.

22. Key features and algorithms of the BGP-4 protocol., http://freesoft.org/CIE/RFC/1774/2.htm.
23. Measuring IP Network Performance - The Internet Protocol Journal - Volume 6, Number 1 - Cisco Systems, http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_6-1/measuring_ip.html.
24. OSPF Design Guide - Cisco Systems, http://www.cisco.com/en/US/tech/tk365/technologies_white_paper09186a0080094e9e.shtml.
25. OSPF Packet Details, http://cisco.iphelp.ru/faq/5/ch08lev1sec1.html.
26. OSPF Packets, http://www.juniper.net/techpubs/software/junos/junos74/swconfig74-routing/html/ospf-overview7.html.
27. Reverse Address Resolution Protocol - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Reverse_Address_Resolution_Protocol.
28. RFC 2131 - Dynamic Host Configuration Protocol, http://tools.ietf.org/html/rfc2131.
29. RFC 4251, http://www.ietf.org/rfc/rfc4251.txt.
30. RFC 792 - Internet Control Message Protocol (RFC792), http://www.faqs.org/rfcs/rfc792.html.
31. RFC 903 - A Reverse Address Resolution Protocol (RFC903), http://www.faqs.org/rfcs/rfc903.html.
32. Routing protocol - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Routing_protocol.
33. Satellite Internet access - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Satellite_Internet_access.
34. SSh - A Telnet Replacement, http://web.science.mq.edu.au/it/doc/services/ssh/.
35. SSH Protocol Overview, http://www.freesoft.org/CIE/Topics/139.htm.
36. TCP and UDP performance tuning, http://publib.boulder.ibm.com/infocenter/aix/v6r1/topic/com.ibm.aix.prftungd/doc/prftungd/tcp_udp_perf_tuning.htm.
37. TCP/IP Architecture, http://technet.microsoft.com/en-us/library/cc751234.aspx.
38. Telnet - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Telnet.
39. The Industrial Ethernet Book | Knowledge | Technical Articles | Performance metrics for Industrial Ethernet, http://www.iebmedia.com/index.php?id=5430&parentid=63&themeid=255&hft=38&showdetail=true&bb=1.
40. The TCP/IP Guide - Data Link Layer (Layer 2), http://www.tcpipguide.com/free/t_DataLinkLayerLayer2.htm.
41. The TCP/IP Guide - TCP and UDP Overview and Role In TCP/IP, http://www.tcpipguide.com/free/t_TCPandUDPOverviewandRoleInTCPIP.htm.
42. Understanding OSPF Routing - Webopedia.com, http://www.webopedia.com/DidYouKnow/Computer_Science/2006/OSPF_Routing.asp.
43. Understanding the Ping and Traceroute Commands - Cisco Systems, http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_tech_note09186a00800a6057.shtml.