the Availability Digest

www.availabilitydigest.com

TCP/IP Illustrated, Volume 1: The Protocols

November 2009

The TCP/IP Protocol Suite is the foundation for the replication networks and the user networks that bind the nodes of an active/active system together.

W. Richard Stevens has written what many consider to be the “bible” of TCP/IP. In his very readable and extensive book (30 chapters and over 500 pages) entitled *TCP/IP Illustrated, Volume 1: The Protocols*,[1] he takes the reader through the TCP/IP protocols from header formats to network management.

Throughout the book, Stevens illustrates the finer points of the protocols with examples captured from a real-life TCP/IP subnet. This subnet comprised a pair of bridged Ethernet LANs connectiing several servers running different operating systems. His traces of message activity for various scenarios bring the protocols to life.

TCP/IP Protocol Suite Layers

The author begins with a description of the four layers of the TCP/IP protocol suite – the link, network, transport, and application layers. He introduces the role of the IP and TCP/UDP protocols in internetworking, illustrating this with detailed descriptions of their header formats and the means of encapsulating the IP network packets into TCP/UDP messages.

Link Layer

Going into more detail, he describes the link-layer Ethernet protocol and its encapsulation of IP datagrams, as well as the SLIP (Serial Line IP) and PPP (Point-to-Point Protocol) protocols for serial links. The author introduces the concept of MTU, or Maximum Transmission Unit, which is the largest packet that a path can handle. If a packet is larger than the MTU, IP fragments it into smaller packets no larger than the MTU.

Network Layer

Stevens goes into great detail describing the network-layer Internet Protocol (IP), including packet header formats, routing, and subnets. In addition to the destination address, an important field in the packet header is the time-to-live counter. This counter limits the number of hops (router-to-router transfers) that a packet can make before it is discarded. This prevents a packet from endlessly traversing a loop in the network.

[1] W. Richard Stevens, *TCP/IP Illustrated: The Protocols*, Addison-Wesley; 1994.

A particular problem in IP networks is discovering the physical destination to which a packet should be routed. The Address Resolution Protocol (ARP) that is used to effect this discovery is explained, along with the related Proxy ARP and Gratuitous ARP protocols. The opposite problem of a newly-booted device discovering its IP address is handled with the Reverse ARP (RARP) protocol.

IP network management is handled in large part by the Internet Control Message Protocol (ICMP). This protocol is used to communicate error messages and other conditions that require attention, such as destination unreachable or unknown, packet life exceeded, or bad IP header. It is also used to request status information from network components.

The "ping" program is the basic connectivity test between two systems. Running at the network layer, "ping" uses ICMP to determine if a remote host is reachable. Traceroute is a more extensive connectivity tool. It sends a sequence of UDP packets with incrementing time-to-live counters in order to map the route from one host to another.

IP routing is one of the most important functions of the IP layer. It is the function that properly routes a packet to its intended destination. The routing tables used by routers are described. The use of ICMP messages to discover routes and to report errors is explained. The Routing Information Protocol (RIP) is covered in detail. RIP is one of the protocols used by IP routers to continuously discover the network topology. The routing protocols OSPF (Open Shortest Path First) and the all-important Border Gateway Protocol (BGP) used to communicate between autonomous systems on the network are also explained.

Transport Layer

The transport layer provides the control for the data flow between two hosts on behalf of the application layer above. The two primary transport layer protocols are UDP (User Datagram Protocol) and TCP (Transmission Control Protocol).

UDP

UDP is a connectionless protocol that provides no guarantee of delivery of a datagram to its destination. The application generating UDP datagrams must be willing to accept this limitation. However, a fragmented UDP datagram is reassembled by IP and delivered as a full datagram.

The author explains how UDP interacts with the ARP protocol and demonstrates how UDP can be used to determine the MTU of a path.

Being connectionless, UDP can be used to multicast or broadcast messages. A message can be broadcast to all hosts on a cable with routing inhibited or to all hosts on a subnet. A multicast message is similar to a broadcast message except that it is directed only to those hosts in the specified multicast group. The link-layer protocol IGMP (Internet Group Management Protocol) lets all systems on a physical network know which hosts belong to which multicast groups.

Domain Name System (DNS)

The DNS is a distributed database used by transport level protocols to map host names identified by their URLs, or Universal Resource Locators, to IP addresses and vice versa. Each site, such as a company or a campus, maintains its own DNS name server containing the URL/IP address pairs of its local hosts.

The DNS name space is hierarchical. If the requested URL is not found in the local name server, the query is passed along to the next name server in the tree. This continues until the query is resolved or rejected.

To improve the efficiency of the DNS, recent hits are cached. Changes to mappings made to a name server may take several minutes to propagate to cache. Furthermore, if a backup name server is used, it will typically query the primary server only every few hours for changes.

Bootstrap Protocol

The RARP protocol described earlier is used to boot diskless systems with no knowledge of their IP addresses. However, RARP requests are not forwarded by routers, and therefore a RARP server must exist on each physical network.

BOOTP, the Bootstrap Protocol, uses UDP to overcome this limitation.

TCP

Along with IP, the Transmission Control Protocol, TCP, is the workhorse of the Internet Protocol Suite. TCP is a connection-oriented protocol that provides reliable message delivery. Stevens devotes eight chapters (and 169 pages) explaining TCP in significant detail.

As with IP and other services, he begins with a detailed description of the TCP header. The header contains, among other fields, the source and destination addresses, a sequence number to guarantee proper sequencing of messages, an acknowledgement number to confirm the sequence number of the last received message, a window size for flow control, and a checksum.

The TCP connection establishment and termination protocols are described in great detail, supported by a state transition diagram. In addition to normal establishment and termination, topics include connection timeouts, the specification by the parties of a maximum segment size, simplex connections (one-way only), connection aborts, simultaneous opens and closes, and various TCP options. He explores issues in the design of TCP servers, including the handling of port numbers, restricting local and foreign IP addresses, and the connection request queue.

TCP can handle interactive data flows and bulk data transfers. Because interactive flows involve shorter messages, known as *tinygrams*, measurements have shown that interactive traffic accounts for about 10% of all TCP traffic. The Nagle algorithm can improve channel efficiency over WANs by accumulating tinygrams into a single message.

Bulk data transfers utilize sliding windows and delayed acknowledgements for flow control. If the window at the receiver fills up, transmission of further data is delayed until the receiver has processed some messages and freed up space in the window.

Typically, several messages will be received before the last one is acknowledged via the acknowledgement number in the TCP header. The acknowledgement of a message indicates that all prior messages have been properly received. There is no negative acknowledgement. If a message is missing (typically due to a communication error), further messages received properly will not be acknowledged. It is up to the transmitter to detect this condition and to initiate a retransmission.

The author shows how the “bandwidth-delay product” can be used to size a window. This factor is the product of the bandwidth of the channel and the channel round-trip time. He also discusses congestion when a “fat pipe” feeds a “skinny pipe” and the means to send urgent data over a congested connection.

Since TCP does not support a negative acknowledgement, it times out if an expected acknowledgement is not received and then resends the unacknowledged data. It determines the appropriate timeout by monitoring the time intervals from the transmissions of messages to the receipt of their acknowledgements, smoothing this time over large intervals.

Stevens describes flow control methods used by TCP to minimize or avoid congestion and the impact of ICMP errors on TCP connections.

In addition to the retransmission timer, TCP maintains a *persist timer* and *keepalive* timer. The persist timer is set by one end of a connection when it has data to send, but it has been stopped because the receiving end has advertized a zero window size. When the timer expires, the sending end will query the receiving end for its current window size.

During idle times, there is no transmission activity on a TCP connection. Therefore, if the receiving server should crash, there is no way for the sending server to know this. The keepalive timer can be used to monitor this condition. Periodically – typically every two hours – the keepalive timer will timeout and the sending end can query the receiving end to find out if it is still up. If it receives no response, it can close the connection.

Stevens wraps up his discussion of TCP by talking about TCP futures. However, since his book was published in 1994, this discussion is somewhat dated.

Application Layer

Sitting on top of TCP and UDP are several applications that help manage communications. These include:

SNMP, the Simple Network Management Protocol

SNMP provides the mechanism for network management stations to determine the status of network elements. Network management consists of a Management Information Base (MIB) that defines the attributes of the network elements and the SNMP protocol that details the format of the packets exchanged. SNMP network management facilities typically use UDP for communication between the management stations and the network elements. The MIB and packet formats are described in full detail.

Telnet and Rlogin

Rlogin is a simple application that lets a client log onto a remote server so that the client can run programs on the server. Telnet does the same, but has many more options. Though Rlogin sends only one character per packet, Telnet is usually used in a line-at-a-time mode to send commands and receive responses from the remote server. Furthermore, Rlogin typically works only between Unix hosts, whereas Telnet is not sensitive to the types of hosts.

FTP, the File Transfer Protocol

FTP is the Internet standard for file transfer. It is used to transfer files between hosts that may use different operating systems. It supports a limited number of file types such as ASCII or binary and a limited number of file structures such as byte stream or record oriented. FTP establishes two connections between the source and target systems – one for commands and the other for data.

SMTP, the Simple Mail Transfer Protocol

SMTP allows users to communicate via email. Users deal with any one of a number of available email agents. Message transfer agents transfer mail queued by the sending email agent to the mailbox of the receiving email agent, which delivers it to the receiving user. SMTP supports relay agents that act as mail hubs

NFS, the Network File System

NFS provides transparent file access for clients to files and file systems. The client is unaware of whether the file is local or is on a remote server. NFS is implemented via Sun's Remote Procedure Call (RPC). A client uses NFS calls to mount a file system and to read or write a sequence of bytes from or to a file starting at a specified byte offset. The client can also get or set file attributes; get the status of a file; create, rename, or remove files; and create, read, and delete directories.

Other Applications

Stevens briefly describes other applications sitting on top of the TCP/IP stack including:

- the Finger protocol that returns information on one or more users on a specified host.
- The Whois protocol that provides information about DNS domains and administrators.
- Archie, which provides a directory of FTP servers across the Internet.
- WAIS (Wide Area Information Servers), which searches databases for keywords (pre-Google).
- Gopher, a user interface to Internet resources such as Archie and WAIS.
- Veronica (Very Easy Rodent-Oriented Netwide Index to Computerized Archives), an index of titles of Gopher services.
- X Windows, which lets clients use bit-mapped displays managed by a server.

Summary

Each of the chapters in TCP/IP Illustrated ends with a set of questions, solutions to which are contained in an appendix. Other appendices deal with tracing TCP/IP packet flow, computer clocks used for timekeeping in Unix systems, and TCP/IP configuration options. An extensive glossary of acronyms is provided.

With its exercises and solutions and its extensive bibliography, *TCP/IP Illustrated* is indeed a valuable reference for the serious networking practitioner or student.