Module Five

CSG1105 Applied Communications

The Transport Layer

Functions of the Transport Layer

- Provides Application multiplexing (ports)
- Provides end-to-end delivery of data
- Segments messages (TCP)
- Ensures segments received in the correct sequence (TCP)

Application Multiplexing

- Any process requiring a TCP/IP connection identifies itself via a 16 bit number known as a **port address** or **port**
- Ports are one of two categories:
  - Well-known Ports: numbers 1-1023* allocated to servers e.g. 22 - SSH, 25 - SMTP, 80 - HTTP. These ports are allocated by the IANA (Internet Assigned number Authority)
  - Ephemeral Ports: numbers 1024 - 65535, generally used by client process and are assigned as needed
- There are many servers that use port numbers above 1023, some of which are mapped by the IANA, some of which are accepted by general usage
- The IP address Port number combination becomes a unique network address for an application
- The combination of each of the client and server IP Address/Port number pairs identifies a connection and is used in the most common TCP/IP API the **Socket**

Sockets

- A **Socket** is a programming API for TCP/IP introduced in 4.2BSD Unix (Berkeley Software Distribution) first introduced in 1983
- The socket is the *de facto* industry standard for accessing TCP/IP from an application
- Sockets use a **file handle** to access network services as the connection is treated as a **byte stream**
- A socket address consists of **protocol, IP address, port** e.g. `tcp,192.168.0.50,80`
- A connection is represented by an *association* which has the IP Address/Port of both client and server e.g.. `tcp,192.168.0.50,80,172.16.100.22,50111`
- Each connection on a host may be uniquely identified by this combination

Transport Layer Protocols

- The Transport Layer provides two protocols

1. UDP: User Datagram Protocol - single packet with Transport Layer header
2. TCP: Transmission Control Protocol - reliable byte stream

UDP: User Datagram Protocol

- UDP, the User Datagram Protocol, is an application interface to IP that provides port addresses for a single packet
- UDP uses *ports* as previously described for connection multiplexing
- UDP is *connectionless*: it does not provide any reliability, flow control or error recovery which must be handled by the application if required.
- UDP has far less overhead for transmission than TCP and is used in some streaming applications for this reason
- UDP is also used for services like DNS,TFTP,RPC & LDAP, that only require small amounts of data
- The maximum payload size of a UDP Datagram is 516 bytes

TCP: Transmission Control Protocol

- TCP is a *connection-oriented* protocol which provides a *full-duplex* byte stream between processes
  - TCP uses *ports* as previously described for connection multiplexing
  - The application writes and reads a *byte stream* to/from the TCP connection
  - TCP *Segments* the byte stream to be passed to the Internetwork layer (done with knowledge of the Internetwork layer MTU)
  - TCP provides *Reliability* using sequence numbers and acknowledgements (ACK)
  - TCP provides *Flow control* via the receiver ACK mechanism (*sliding window*)

Connection Establishment

- As previously mentioned, TCP is *connection oriented*. Think of it like a phone call...
  1. You enter a phone number you wish to call and the remote handset either rings or you get an engaged signal
  2. If the is engaged, the call attempt ends
  3. If the phone rings and someone answers, a brief dialogue occurs ensuring you're talking to the correct person
  4. Phone conversation continues until completion if no network errors
  5. Caller or called ends call
- Phone calls are connection oriented

Three Way Handshake 1

- TCP uses a process called the *Three Way Handshake* to establish a connection
  1. The server must have a process listening for incoming connection requests it creates a Socket with a *Passive Open*
  2. A client attempting to connect to the server creates an *Active Open* call
     a. The Client sends a SYN (synchronise) message with a random sequence number (SEQ), say 2044
     b. If not accepting requests, the server can reply with a NAK (negative acknowledge)
     c. If accepting requests, the server replies with a SYN ACK (acknowledge) with it's own random SEQ, say 42000, and acknowledges the request with the next expected SEQ, 2045
     d. The client completes the connection with an ACK with SEQ 2045 and ACK 42001

Three Way Handshake 2

3. The connection is now open and segments may be exchanged using the sequence numbers established
4. Data is exchanged and the sequence numbers are used for reliability and flow control
5. The connection is closed with one end sending a FIN segment , the other end will complete any operations in progress then sends a FIN segment.
6. When closed, all status fields etc are cleared and if appropriate an new passive open is performed on the server to again listen for incoming connections.

In practice, a busy server such as a web server, will have multiple listening processes that can be used to establish a connection.

Flow control

- Each TCP connection has a *sliding window* that assists in reliability and flow control
- The "window" is a range of unacknowledged segments that may be transmitted
- If the window is exceeded, the sender ceases sending until an appropriate ACK of sent segments is received
- If a receiving end is congested, it stops sending ACKs which will shut down the sender
- if a segment is lost, the receiving end will repeat sending ACKs for the last correctly received segment until the sender retransmits the lost segment
- The receiver, if it has segments after the lost segment still in its buffer, can acknowledge multiple segments when the lost segment is received.

Network Address Translation Overview

- Now that we have a basic under standing of TCP and IP addresses, we can explain a commonly used feature called **Network Address Translation**.
- There are two forms under this banner
  1. Network Address Translation (NAT)
  2. Port Address Translation (PAT - AKA Network Address Port Translation NAPT)
- PAT is now the most commonly used of the two, but understanding the basics of NAT will assist in the comprehension of PAT

Interlude: Private IP Addresses

- The issue of address exhaustion and the use of IP in non-Internet connected networks resulted in the publication of RFC1918 - Address Allocation for Private Internets
- This document outlined the available addresses and appropriate usage of three ranges of IP addresses
- It also mandated that these not be used on the wider Internet

Start	Finish	Mask	Net Size
10.0.0.0	10.255.255.255	/8	16777214
172.16.0.0	172.31.255.255	/12	1048574
192.168.0.0	192.168.255.255	/16	65534

- Internal use of a large space allows arbitrary codes to be embedded in addresses
- At ECU: **10.18.2.51/24** - Building 18, Level Two allowing 254 host per level

NAT

- Private IPs cannot be routed on the wider Internet
- NAT was originally meant to be a stopgap solution to IP address exhaustion while waiting for IPv6
- It also allowed organisations previously not connected to the Internet, but using IP internally to connect without having to renumber all networks
- Refer to RFC3022: Traditional IP Network Address Translator for details

NAT Operation

- A NAT gateway (usually a router), maintains a table of available external IP addresses
- When an internal host wants to establish a connection to an external host, the NAT gateway assigns an external IP to the connection and **rewrites** the IP header of the outgoing packet with the external IP
- A table is maintained with a tuple of the internal/external IP addresses
- When a packet is received for the internal host from the external host, the table is referenced and the IP header is again re-written
- When the connection is closed, the entry is removed from the table

Host	**Destination**	**Source**
Internal Outgoing	134.1.1.99:80	10.18.2.51:446
Gateway Outgoing	134.1.1.99:80	228.50.21.1:446
External Received	134.1.1.99:80	228.50.21.1:446
External Sent	228.50.21.1:446	134.1.1.99:80
Gateway Received	228.50.21.1:446	134.1.1.99:80
Internal Received	10.18.2.51:446	134.1.1.99:80

Traditional NAT

- Traditional NAT requires a **one-to-one** relationship between the internal IP address and the external IP address
- Allocating an individual external IP for each internal host's private IP would defeat the purpose of using NAT to preserve IP address space
- One solution was to provide a pool of available external IPs that could be used for NAT connections
- When no more IPs are available, no addition connections to IPs are possible
- With limited backbone IPs available, this would limit the number of IPs available for NAT

PAT/NAPT AKA NAT Overloading

- PAT uses the same concept as NAT, except that it rewrites the **port address** as well as the IP address on the external connection
- Port addresses are a 16 bit number, allowing 65354 connections **for a single external IP**
- Domestic ISP Routers utilise PAT to allow multiple concurrent external connections using a single ISP provided external IP

Internal IP	Internal Port	External IP	External Port
10.18.2.20	4673	134.6.101.20	1111
10.18.2.201	11673	134.6.101.20	1112
10.18.2.55	4483	134.6.101.20	1113
10.18.2.222	10987	134.6.101.20	1114

NAT Advantages

- Extend the life of IPv4 address space
- Provides additional security by hiding internal IP addresses
- Provides independence from ISP provided IPs. Less overhead if change ISP
- Provides flexibility for subnetting both in size and assigned addresses (eg. 10.18.2.51)

NAT Disadvantages

- NAT consumes RAM and processor cycles on device performing translation
- Processing time can cause delay in IP communication which may be an issue for time sensitive protocols
- May make it difficult to troubleshoot network issues as TAPs are needed on both sides of the NAT gateway
- Some protocols have issues transitioning across a NAT gateway

Multi-tier NAT

- It is possible to have more that one layer of NAT
- This can occur in small-scale ISPs where they have limited external IPs provide customers with an IP from the private IP set. The customer ISP router in turn will provide a NAT gateway for the customer internal network. This can result in significant latency in connections
- The server network may also implement NAT resulting in NAT at both ends of a connection