Communication Protocols and Internet Architectures

Harvard University

Lecture #7

Instructor: Len Evenchik
cs40@evenchik.com or evenchik@fas.harvard.edu

ALIGHLSOD1701

Lecture Agenda

- Course Logistics
- Q&A and Topics from Last Week
- SP3 (Review)
- Transport Layer Protocols
- UDP
- TCP
- Connection Management
- One Minute Wrap-Up

Course Logistics

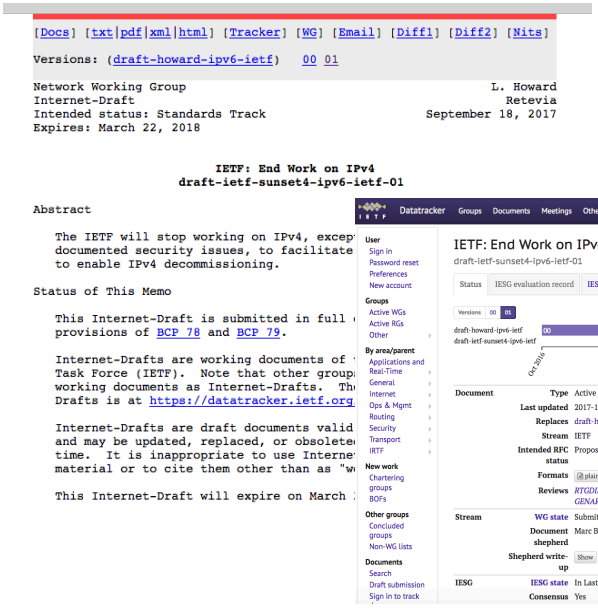
Course Logistics

- What are the Optional Readings that are posted each week?
- Homework update
- There will be an online midterm exam and an on-campus or proctored final exam. Students in New England must take the final exam on campus while distance students must arrange to have it proctored.
- Please see the syllabus for the dates of the midterm and the final exam.
- **Please submit a one minute wrap-up each week. Thank You!**

Q&A
Topics from Last Week

IPv6 Fundamentals

ID on IPv4 End of Work



[Docs] [txt|pdf|xml|html] [Tracker] [WG] [Email] [Diff1] [Diff2] [Nits]

Versions: (draft-howard-ipv6-ietf) 00 01

Network Working Group — L. Howard
Internet-Draft — Retevia
Intended status: Standards Track — September 18, 2017
Expires: March 22, 2018

IETF: End Work on IPv4
draft-ietf-sunset4-ipv6-ietf-01

Abstract

The IETF will stop working on IPv4, excep
documented security issues, to facilitate
to enable IPv4 decommissioning.

Status of This Memo

This Internet-Draft is submitted in full
provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of
Task Force (IETF). Note that other group
working documents as Internet-Drafts. Th
Drafts is at https://datatracker.ietf.org

Internet-Drafts are draft documents valid
and may be updated, replaced, or obsolete
time. It is inappropriate to use Interne
material or to cite them other than as "w

This Internet-Draft will expire on March

Datatracker Groups Documents Meetings Other User

IETF: End Work on IPv4
draft-ietf-sunset4-ipv6-ietf-01

Status | IESG evaluation record | IESG writeups | Email expansions | History

Document — Type: Active Internet-Draft (sunset4 WG)
Last updated: 2017-10-02 (latest revision 2017-09-18)
Replaces: draft-howard-ipv6-ietf
Stream: IETF
Intended RFC status: Proposed Standard
Formats: plain text | xml | pdf | html | bibtex
Reviews: RTGDIR Last Call Review - due: 2017-10-12
GENART Last Call Review - due: 2017-10-12

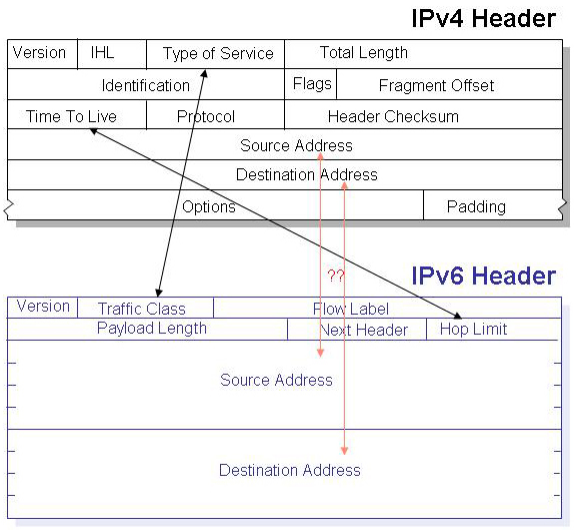
Stream — WG state: Submitted to IESG for Publication (wg milestone: Jul 2017 - IPv6 Support Within ...)
Document shepherd: Marc Blanchet
Shepherd write-up: Show (last changed 2017-09-19)

IESG — IESG state: In Last Call (ends 2017-10-12)
Consensus: Yes

Primary IPv6 Changes from IPv4
(as described in RFC 2460)

- Expanded Addressing capabilities: addresses are 128 bits long, improved auto-configuration, anycast addresses, etc.
- Simplified Header format
- Better support for Options and Extensions
- Capability for Flow Labeling is added
- Added Authentication and Privacy Capabilities

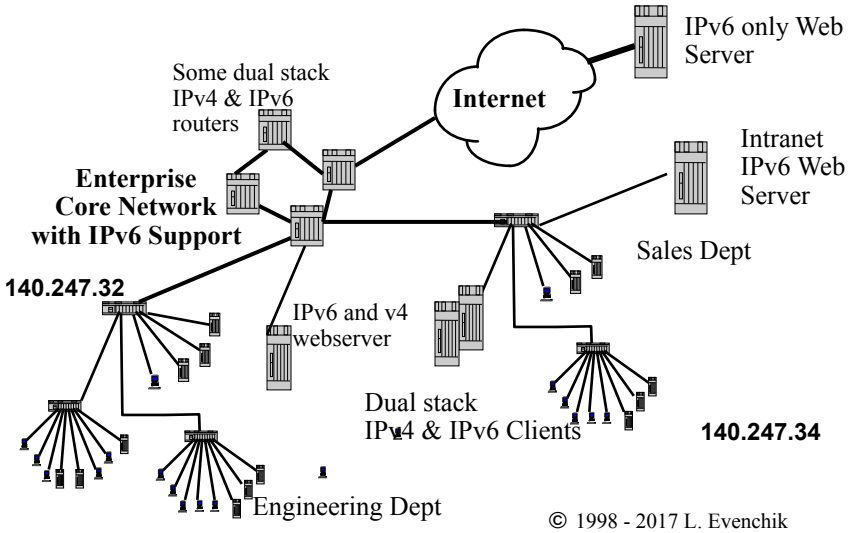
IPv4 and IPv6



Source: http://ispcolumn.isoc.org/2007-08/fig1.jpg

Are You Using IPv6 Today?

Supporting IPv6 and IPv6 Transition



IPv6 Ping

The command on Unix is ping6 and on Win7 it is ping -6

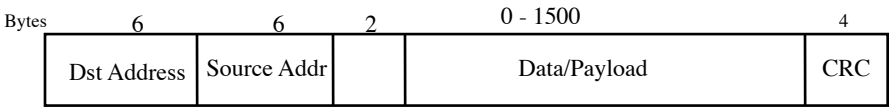
```
fas% ping6 -I eth0 ff02::1
PING ff02::1 from fe80::20b:cdff:fe82:57e7 eth0: 56 data bytes

64 bytes from fe80::20b:cdff:fe82:57e7: icmp_seq=1 time=0.060 ms
64 bytes from fe80::21e:4fff:fe32:494b: icmp_seq=1 time=0.478 ms
64 bytes from fe80::20b:cdff:fe83:a9a:  icmp_seq=1 time=0.901 ms
64 bytes from fe80::20b:cdff:fe82:4673: icmp_seq=1 time=1.82 ms
64 bytes from fe80::224:e8ff:fe64:e434: icmp_seq=1 time=2.19 ms
64 bytes from fe80::224:e8ff:fe64….  Etc


fas% ping6 -I eth0 SomeIPv6Host
```

IEEE EtherTypes Available via IANA

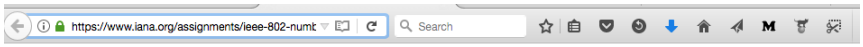
https://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml



EtherType (Hex)
IPv4 is 0800
IPv6 is 86DD

IEEE EtherTypes Available via IANA

https://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml



IEEE 802 Numbers

Ethertype (decimal)	Ethertype (hex)	Exp. Ethernet (decimal)	Exp. Ethernet (octal)	Description	References
0000	0000-05DC	-	-	IEEE802.3 Length Field	[Neil_Sembower]
0257	0101-01FF	-	-	Experimental	[Neil_Sembower]
0512	0200	512	1000	XEROX PUP (see 0A00)	[Boggs, D., J. Shoch, E.
2048	0800	513	1001	Internet Protocol version 4 (IPv4)	[RFC7042]
2049	0801	-	-	X.75 Internet	[Neil_Sembower]
2050	0802	-	-	NBS Internet	[Neil_Sembower]
2051	0803	-	-	ECMA Internet	[Neil_Sembower]
2052	0804	-	-	Chaosnet	[Neil_Sembower]
	86DE			Delta Controls	[Neil_Sembower]
	86DD			Internet Protocol version 6 (IPv6)	[RFC7042]
34527	86DF	-	-	ATOMIC	[JBP]

Address Allocation

IANA Address Resources

iana
Internet Assigned Numbers Authority

DOMAINS NUMBERS PROTOCOLS ABOUT IANA

Number Resources

Overview
Abuse Issues
Overview
Questions and Answers

Number Resources

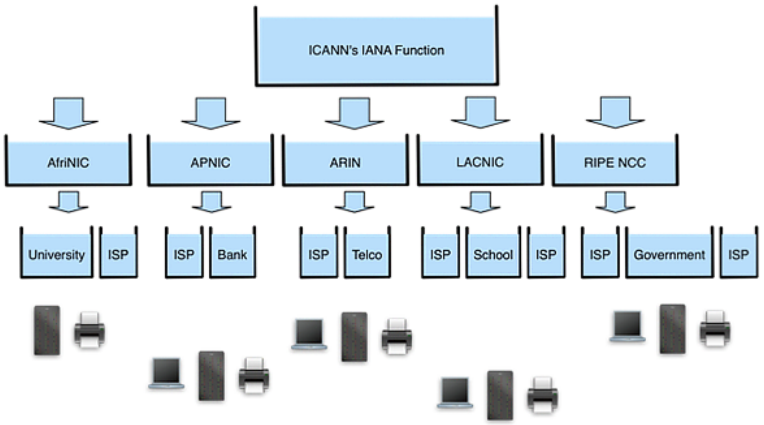
IANA is responsible for global coordination of the Internet Protocol addressing systems, as well as the Autonomous System Numbers used for routing Internet traffic.

Currently there are two types of Internet Protocol (IP) addresses in active use: IP version 4 (IPv4) and IP version 6 (IPv6). IPv4 was initially deployed on 1 January 1983 and is still the most commonly used version. IPv4 addresses are 32-bit numbers often expressed as 4 octets in "dotted decimal" notation (for example, *192.0.2.53*). Deployment of the IPv6 protocol began in 1999. IPv6 addresses are 128-bit numbers and are conventionally expressed using hexadecimal strings (for example, *2001:0db8:582:ae33::29*).

Both IPv4 and IPv6 addresses are generally assigned in a hierarchical manner. Users are assigned IP addresses by Internet service providers (ISPs). ISPs obtain allocations of IP addresses from a local Internet registry (LIR) or National Internet Registry (NIR), or from their appropriate Regional Internet Registry (RIR):

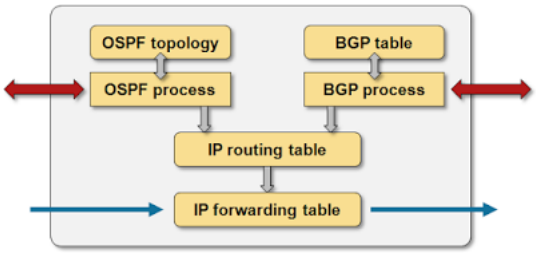
Registry	Area Covered
AFRINIC	Africa Region
APNIC	Asia/Pacific Region
ARIN	Canada, USA, and some Caribbean Islands
LACNIC	Latin America and some Caribbean Islands
RIPE NCC	Europe, the Middle East, and Central Asia

IPv4 Address Distribution



Source and copyright:
http://www.icann.org/en/announcements/announcement-29jan10-en.htm

BGP Routing

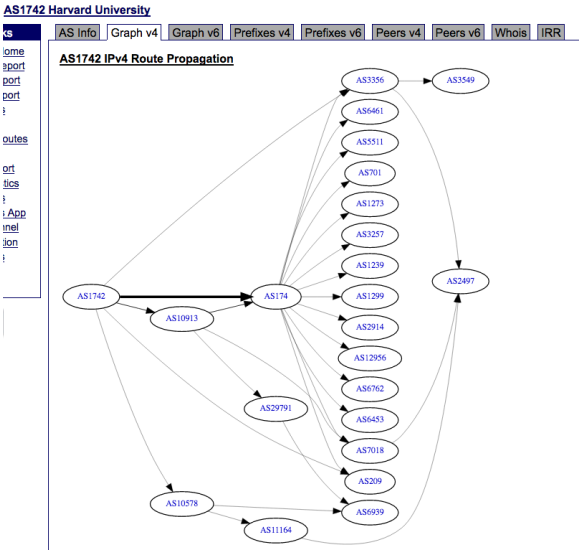


Source unknown

ASN - Autonomous System Nuumber

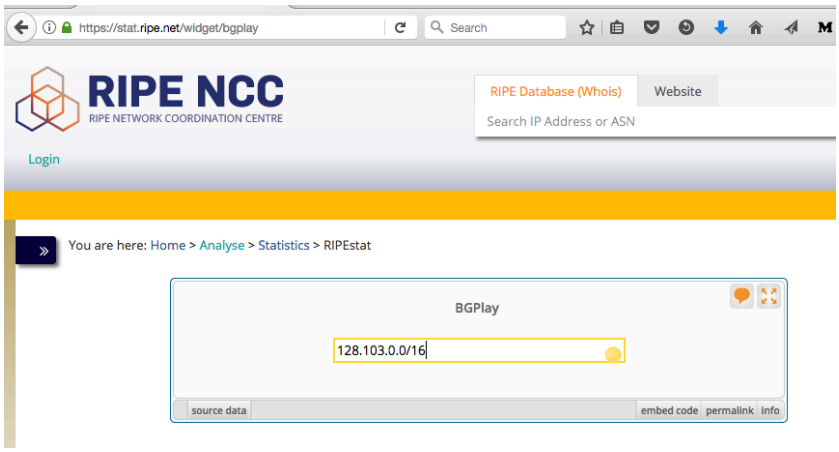
- In simple terms, an Autonomous System is a group of routers that are managed by a single organization. An AS can be a large user (corporation or university), an ISP, or another type of network provider.
- An AS is identified by an ASN
- The concept of an Autonomous System provides a way to mange the complexity of the Internet and Internet routing.
- The number of autonomous systems in the Internet is significantly less than the number of networks that comprise the Internet.
- One AS originates and announces multiple network prefixes. It originates fewer than it announces.

Harvard AS 1742 Route Propagation (IPv4)



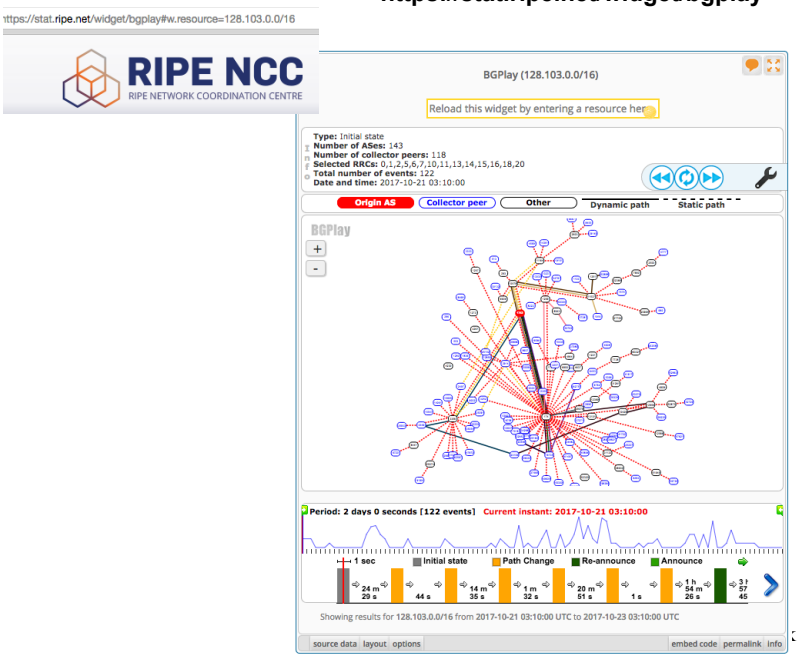
Source: http://bgp.he.net// (October 2017)

BGP Analysis Tool called BGPLAY
https://stat.ripe.net/widget/bgplay

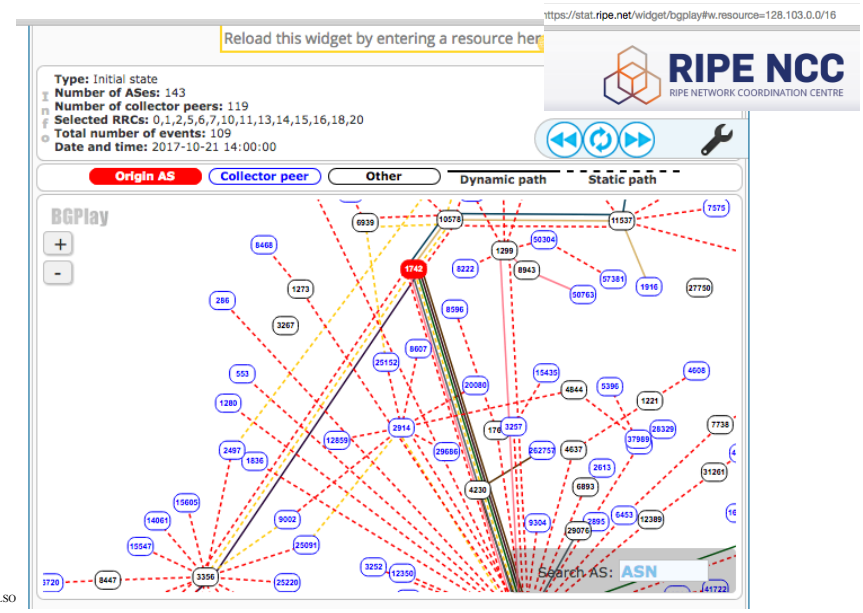


ALIGHLSOD1701

https://stat.ripe.net/widget/bgplay

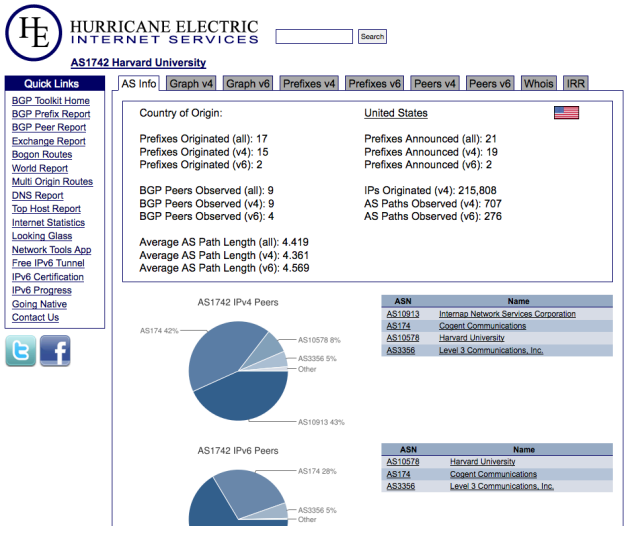


BGP Analysis Tool called BGPLAY
https://stat.ripe.net/widget/bgplay



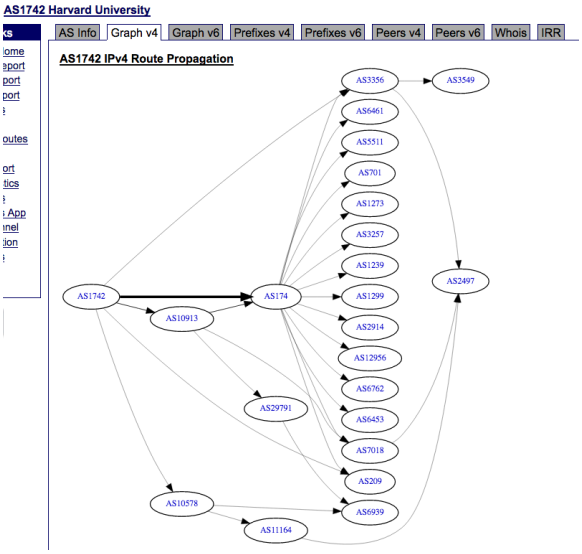
ALIGHLSO

Harvard AS 1742



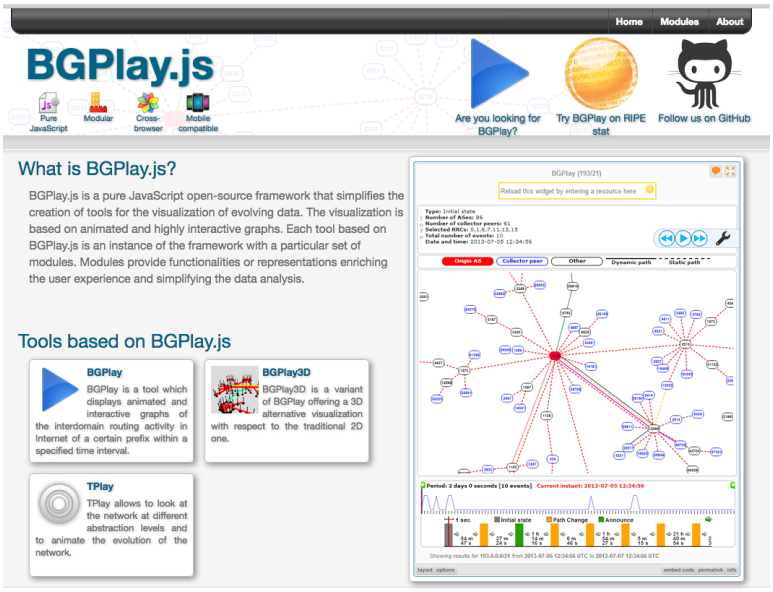
Note that Harvard has a more than one AS.
Source: http://bgp.he.net//

Harvard AS 1742 Route Propagation (IPv4)

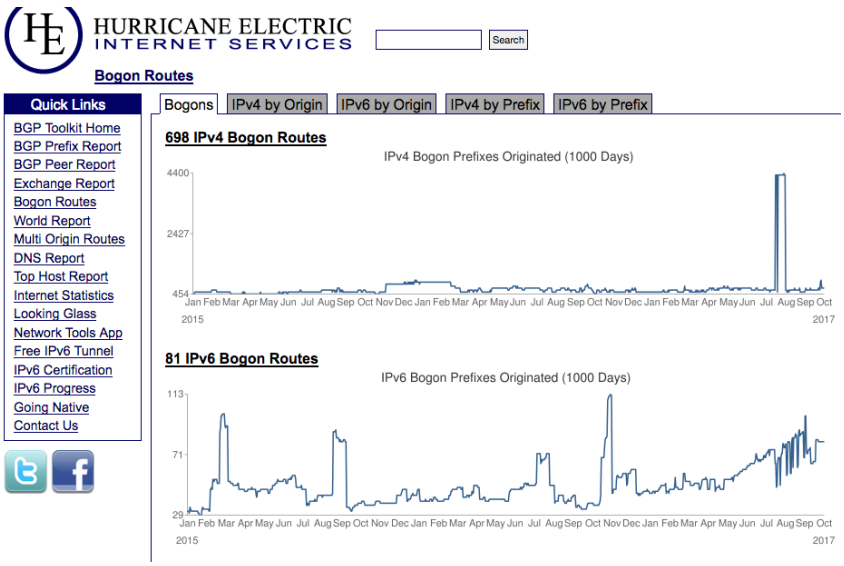


Source: http://bgp.he.net// (October 2017)

https://bgplayjs.com/

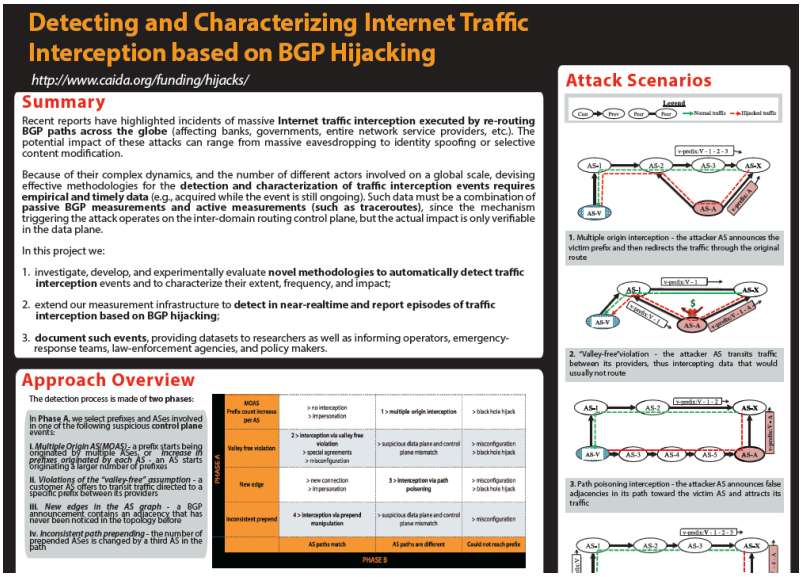


BOGON Route Announcements

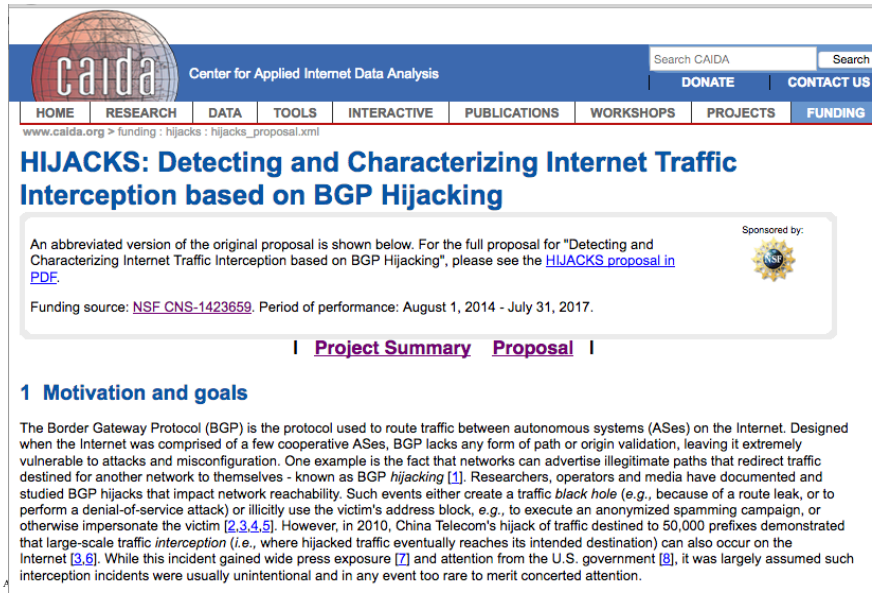


Source: http://bgp.he.net// (October 2017) © 1998 - 2017 L. Evenchik

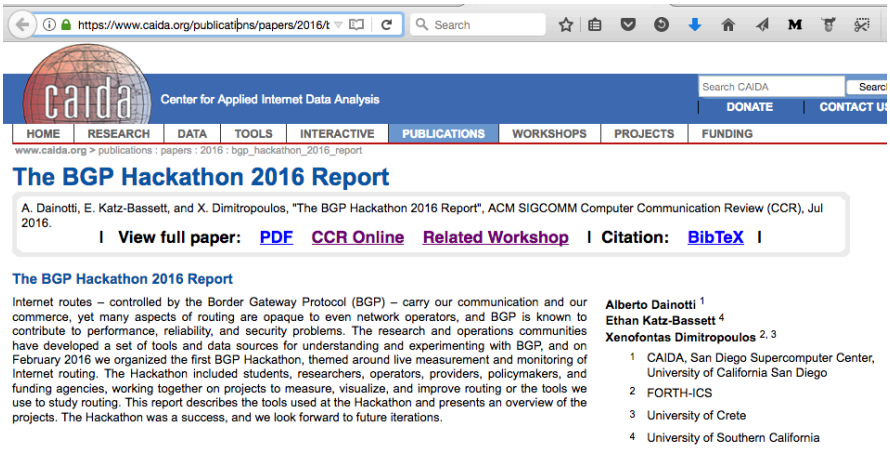
AS Research and BGP Hijacking



Research on BGP and AS Hijacking
https://www.caida.org/funding/hijacks/hijacks_proposal.xml

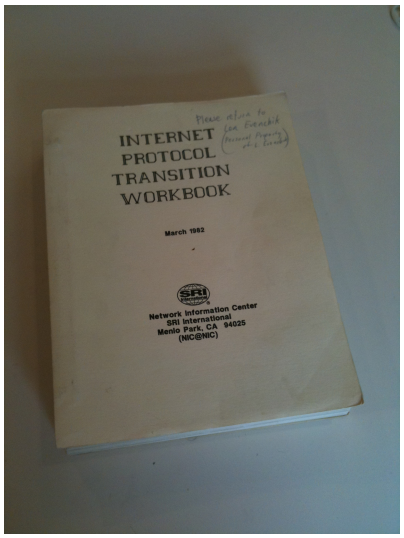


CAIDA Sponsored BGP Hackathon
https://www.caida.org/publications/papers/2016/bgp_hackathon_2016_report/



ALIGHLSOD1701

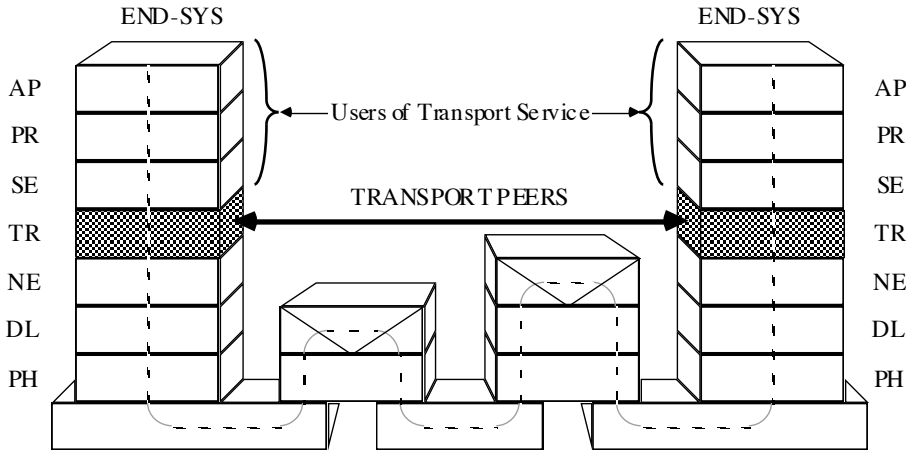
Transport Layer



TCP Transition Workbook
March 1982

Transport Service

(Source: unknown, but this comes from the days of the 7-layer model)



(Source: unknown, circa 1986)

SP3 Protocol Framework

- Service
  - The Service is a description of what the protocol does, not how it is done. This should be a few sentences long.
- Purpose
  - The Purpose describes the specific functionality that the protocol provides and how it is accomplished. Examples are flow control, error detection, error correction, etc.
- Packets
  - The Packet layout determines how the various bits and fields within the packet are defined, assembled and used.
- Procedures
  - The Procedures describe the various packet exchanges and the reason for each exchange.

SP3 - Service

What type of Service does the protocol provide?

The Service is a short description of what the protocol does, not how it is done. For example, a link layer protocol could provide any of the following services

- Reliable service, including sequenced delivery. This is commonly known as connection-oriented service.
- Reliable service, but not with sequenced delivery.
- Unreliable service. This is known as connectionless or datagram service. (IP, UDP)
- Unreliable service, but with the sequenced delivery of messages. (RTP is an example of this.)
- Are there more?

SP3 – Purpose

What specific functionality does the protocol provide and how does it do it?

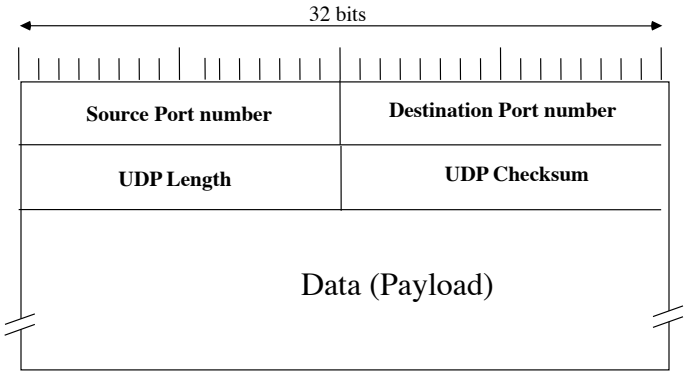
- Addressing
- Multiplexing
- Sequencing
- Error control - two parts to this, detection and correction
- Flow control
- Option negotiation
- Encryption
- Fragmentation and reassembly
- *plus many others...*

User Datagram Protocol

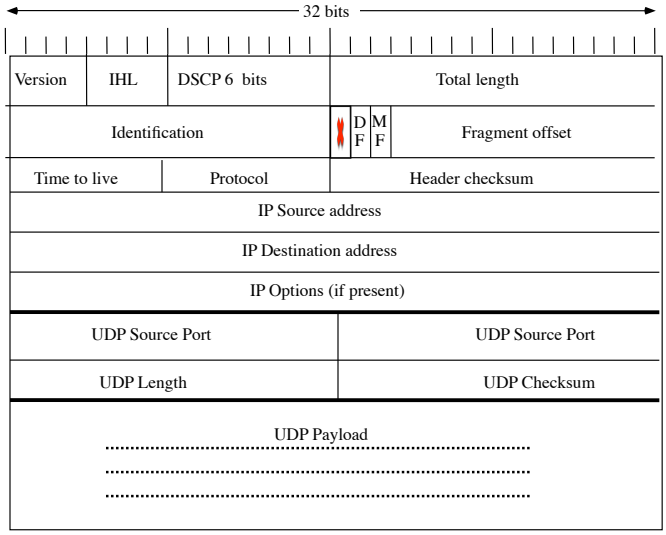
UDP Protocol

- User Datagram Protocol is a very simple transport layer protocol
- UDP provides a Datagram delivery service
- Multiplexing feature is provided via abstract destination points known as protocol Ports
- Port assignment mechanisms include the use of well-known ports as well as dynamic binding
- Where would you look to find the listing of port number assignments?
- UDP checksum provides end-to-end error detection and it includes a pseudo header that includes IP header fields. The checksum is very important, but it is also important to understand that its calculation violates the concept of strict protocol layering.

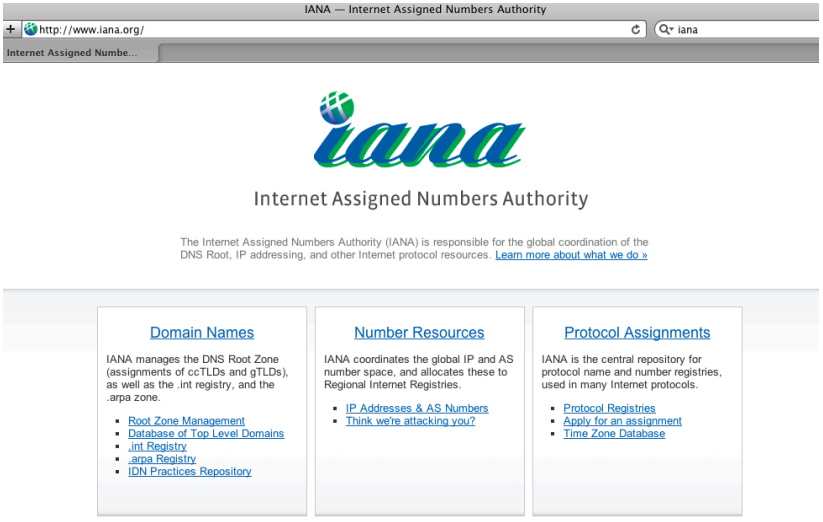
UDP Packet Format



Combined IP/UDP Packet Layout



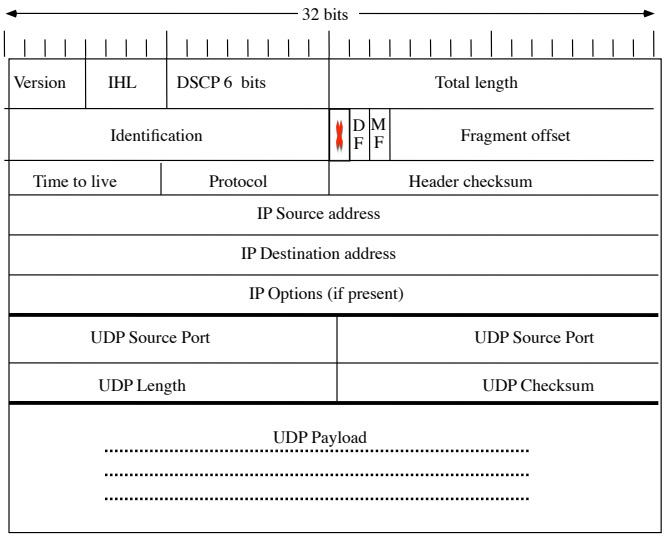
WWW.IANA.ORG



UDP Pseudo Header

- UDP checksum provides end-to-end error detection, but not correction.
- UDP checksum includes a pseudo header which is conceptually prefixed to the UDP header
- Pseudo header includes the IP source and destination addresses, the IP protocol field, and the UDP length
- The checksum is very important, but it is also important to understand that its calculation violates the concept of strict protocol layering.

Combined IP/UDP Packet Layout



Transmission Control Protocol (TCP)

Description of TCP (from RFC 793)

TCP provides a … reliable securable logical circuit or connection service between pairs of processes…..
To do this using less reliable communication systems requires...

- Basic Data Transfer
- Connections
- Reliability
- Flow Control
- Multiplexing
- Precedence and Security

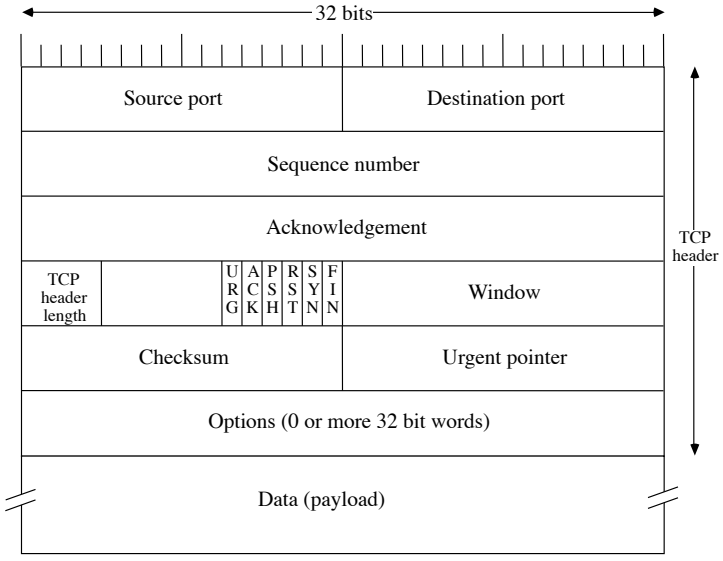
Remember, TCP is a communication protocol, not a specific piece of software

Source, RFC 793

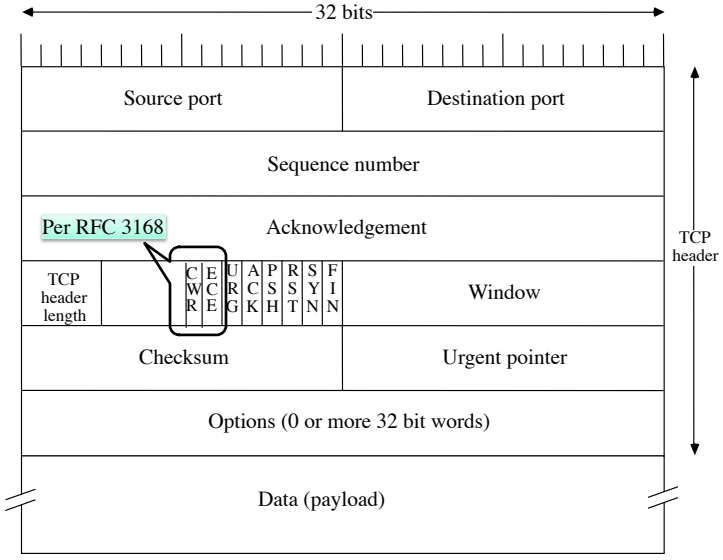
RFC 4614 (2006)
TCP is Complicated and Implementation Specific

- RFC 4614 was written to provide a roadmap to TCP. It describes the most important RFCs and just as important, it explains what prior work is no longer relevant.
- It is important to note that it was written in 2006. See RFC 6247.
- This roadmap is divided into four main section:
  - Basic (core) Functionality
  - Recommended Enhancements
  - Experimental Extensions
  - Historic Extensions

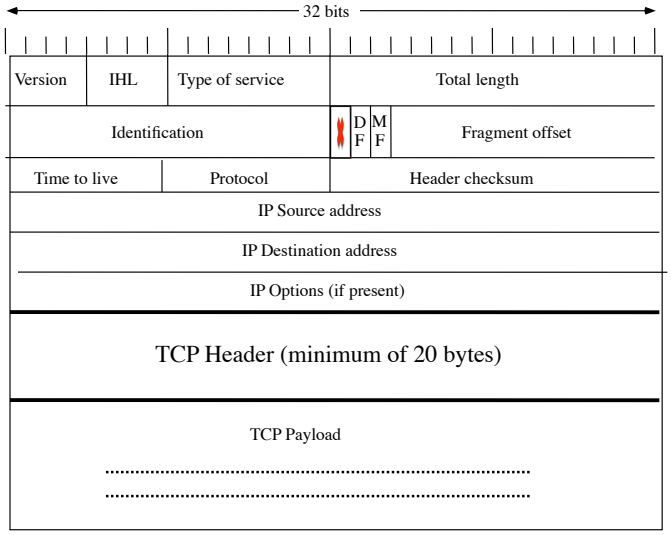
TCP Packet Header (RFC 793, 1981)



TCP Packet Header, 2011



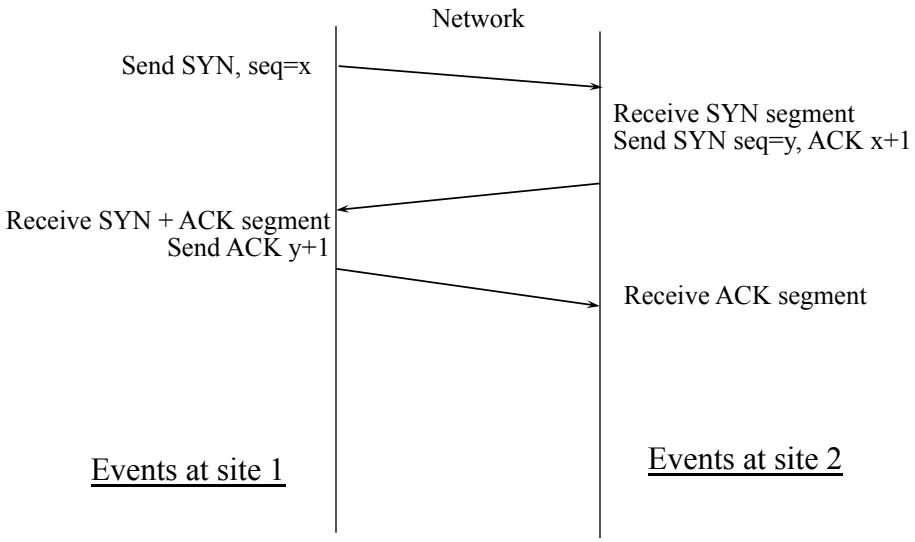
Combined IP/TCP Packet Layout



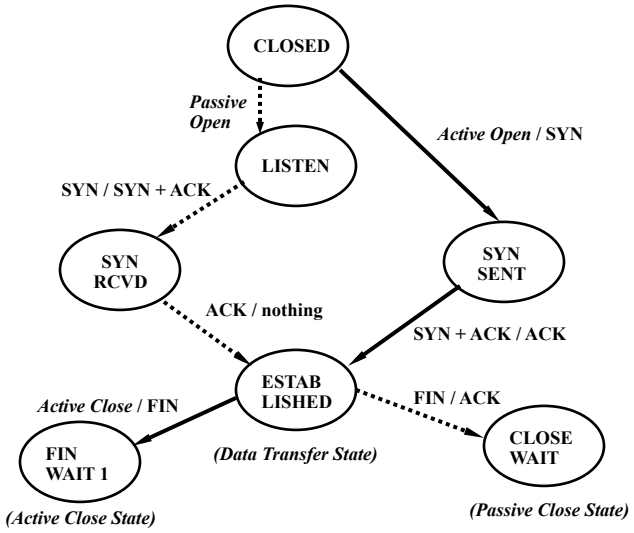
Meaning of Flag Bits in TCP Header

- URG - urgent pointer field is valid
- ACK - acknowledgement field is valid
- PSH - this segment requests a push
- RST - reset the connection
- SYN - synchronize sequence numbers
- FIN - sender has reached end of its byte stream
- CWR and ECE - provides info on congestion

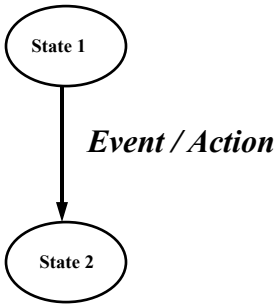
TCP Connection Establishment



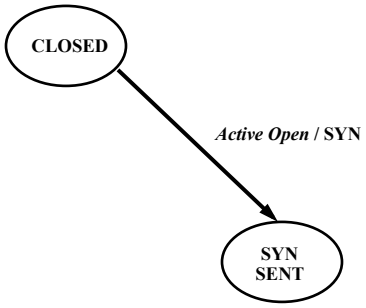
TCP State Transition Diagram, Client/Server - abridged



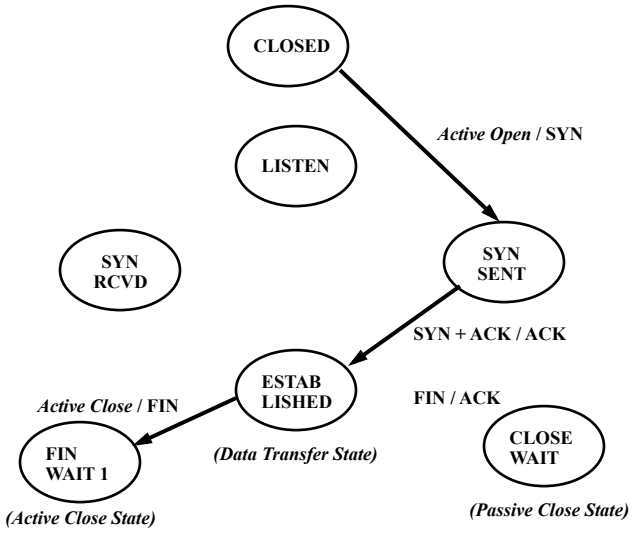
State Transition Diagram



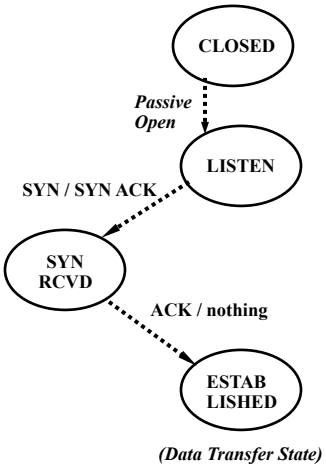
TCP State Transition Diagram, Client Side - abridged



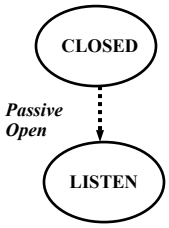
TCP State Transition Diagram, Client Side - abridged



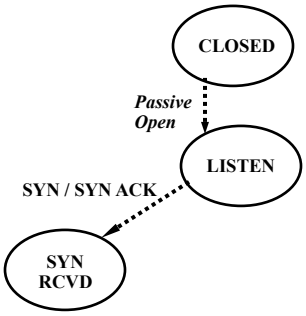
TCP State Transition Diagram- Server Side (abridged)



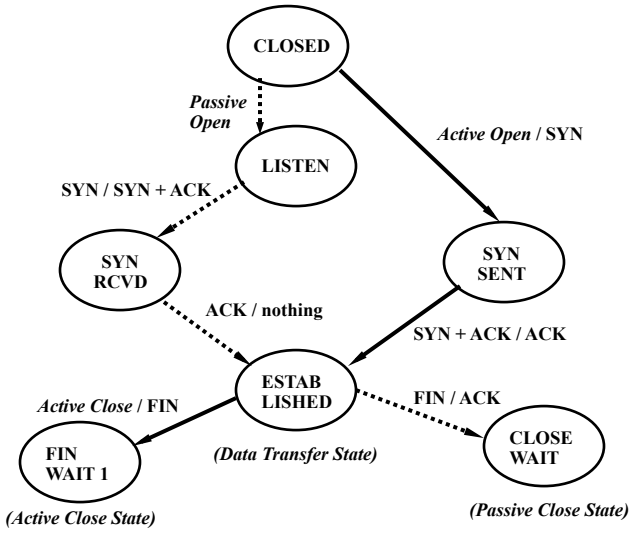
TCP State Transition Diagram- Server Side (abridged)



TCP State Transition Diagram- Server Side (abridged)



TCP State Transition Diagram, Client/Server - abridged



TCP Sequencing, Flow Control and Segmentation Some General Characteristics

- A TCP data stream is a sequence of octets (bytes). This is different than the link layer protocols we have studied which keep track of entire frames (not bytes.)
- TCP uses a sliding window and this window identifies bytes not packets
- TCP does not know anything about the bytes it is sending. (As you would expect with transport protocols.)
- The size of the Flow Control window changes dynamically over time
- Retransmission timeout also changes dynamically (based on RTT) over time

TCP Byte Stream Functionality

- A TCP data stream is a sequence of octets (bytes.) Data is acknowledged at the byte level, not the segment, packet or frame level.
- TCP connections are full duplex, point-to-point. Data flows simultaneously but independently in each direction.
- When data is actually sent on the wire is at the discretion of the sending TCP software.
- When data is given to the application is at the discretion of the receiving TCP.
- The intent of the PUSH bit is to modify when TCP sends or processes the received data (as we will see.)
- The receiving application can be told about important data located in the byte stream via the use of the URGENT bit. Note this is a "hint" not a command.

TCP Segmentation

- TCP sends data in segments. The default segment size is 536 bytes of data (for IPv4.)
- TCP maximum segment size (MSS) is negotiated during connection setup. TCP decides how big the segments can be, not the application software.
- The maximum segment size (MSS) is dependent upon the size of MTU (Maximum Transmission Unit.) What does this violate?
- TCP segments can arrive out of order. Lets think about this for a minute.

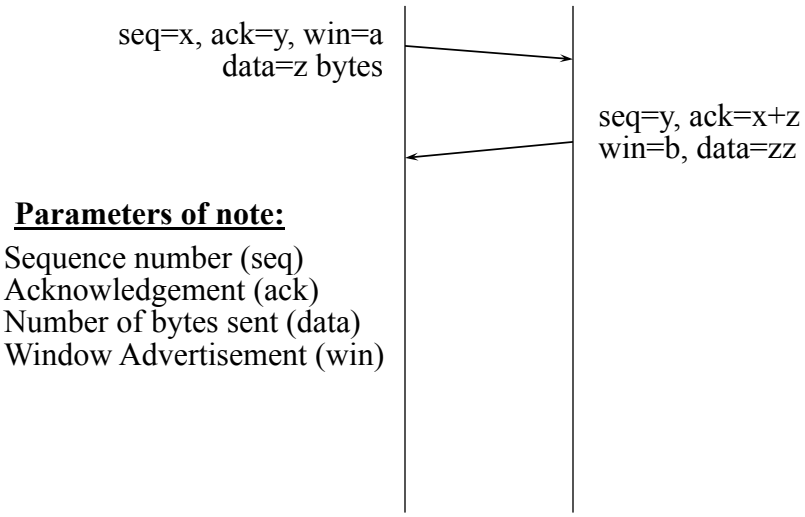
TCP Flow Control

- TCP uses a byte oriented, variable size, sliding window.
- Window size changes dynamically during the connection. How is this done?
- Window Advertisements and Acknowledgements are independent.
- Sending an ACK = X means that X-1 has been received correctly.
- A window advertisement of zero is legitimate (but a 1 byte segment can always be sent.)
- Remember, TCP is full duplex so flow control must be done in both directions simultaneously.
- In TCP, congestion control is not the same a flow control. More on this in a minute.

Meaning of Flag Bits in TCP Header

- URG - urgent pointer field is valid
- ACK - acknowledgement field is valid
- PSH - this segment requests a push
- RST - reset the connection
- SYN - synchronize sequence numbers
- FIN - sender has reached end of its byte stream
- CWR and ECE - provides info on congestion

An Example of TCP Sequencing, Flow Control and Segmentation



Parameters of note:

Sequence number (seq)
Acknowledgement (ack)
Number of bytes sent (data)
Window Advertisement (win)

Wrap Around for 32 bit Sequence Number

Bandwidth	Time until Wraparound
T1 (1.5 Mbps)	6.4 hours
Ethernet (10 Mbps)	57 minutes
T3 (45 Mbps)	13 minutes
FDDI (100 Mbps)	6 minutes
STS-3 (155 Mbps)	4 minutes
STS-12 (622 Mbps)	55 seconds
STS-24 (1.2 Gbps)	28 seconds

Table 5.1 Time until 32-bit sequence number space wraps around.

- This table shows for example that the 32-bit sequence number wraps around in 6 minutes for a 100BaseT link
- A TCP option can "extend" the sequence number

Source and Copyright of table is
Computer Networks by Davie and Peterson

Keeping the Pipe Full

Bandwidth	Delay × Bandwidth Product
T1 (1.5 Mbps)	18 KB
Ethernet (10 Mbps)	122 KB
T3 (45 Mbps)	549 KB
FDDI (100 Mbps)	1.2 MB
STS-3 (155 Mbps)	1.8 MB
STS-12 (622 Mbps)	7.4 MB
STS-24 (1.2 Gbps)	14.8 MB

Table 5.2 Required window size for 100-ms RTT.

- This table assumes a 100 msec RTT time
- The Delay X Bandwidth product dictates the necessary size for the Advertised Window
- A TCP option allows TCP to increase the advertised window size

Source and Copyright of table is
Computer Networks by Davie and Peterson

TCP Congestion Contol

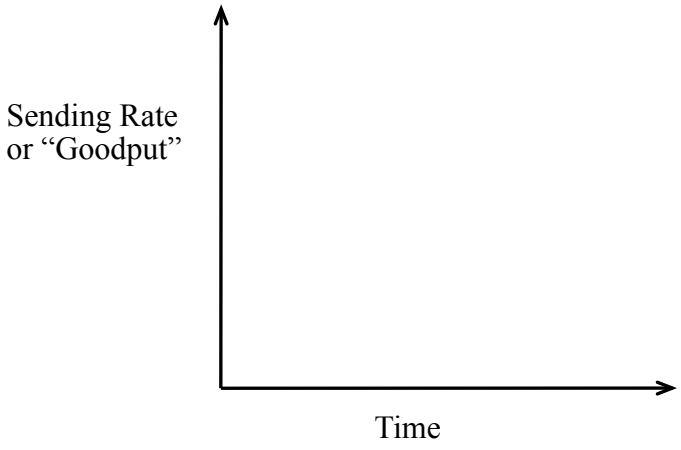
TCP Congestion Control (1)

- The assumption today is that networks are reliable (but of course, not perfect) and therefore packets are dropped due to network congestion.
- A second assumption is that network congestion occurs at routers (or other network devices) due to bursts of traffic.
- Congestion control and flow control are very different. (A classic drawing represents this with pipes and overflowing buckets of water.)
- Congestion control and Slow Start were not part of RFC 793. Van Jacobson designed them in 1988.
- TCP "Slow Start" addresses congestion control, not flow control.

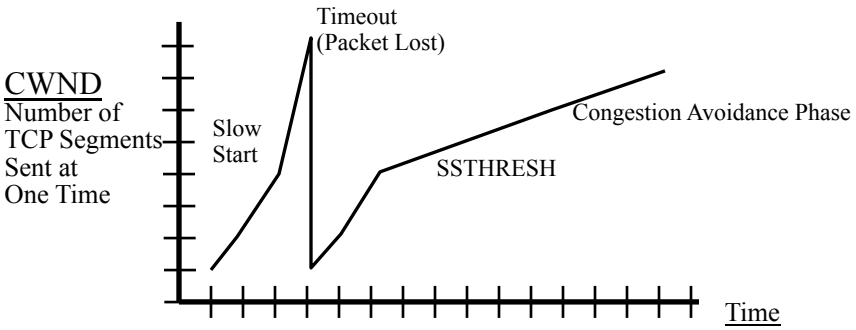
TCP Congestion Control (2)

- Note that "Slow Start" is a misnomer. It is actually exponential growth.
- TCP uses four intertwined Congestion Control algorithms and mechanisms (RFC 5681)
  - slow start
  - congestion avoidance
  - fast retransmit
  - fast recovery.
- TCP Congestion Avoidance is additive-increase, multiplicative-decrease (AIMD)
- Finally, the first assumption we make for congestion control is that networks are reliable, and this does not apply to wireless networks. What does this mean for real world implementations?

TCP Congestion Contol



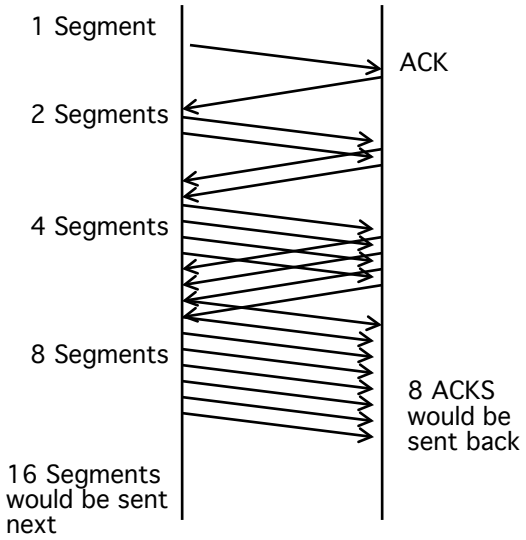
Simplified TCP Slow Start (1)



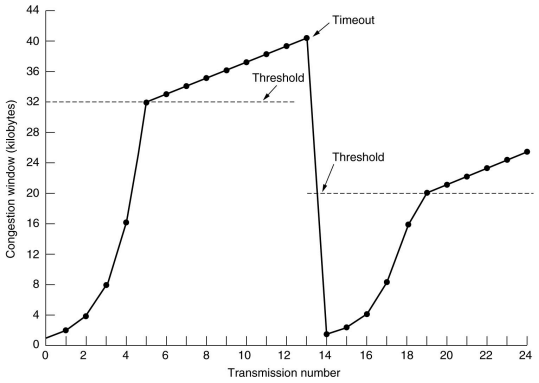
Parameters of note:

Receiver's Advertised Window Size for Flow Control
Initial Window (IW)
Congestion Window Size (CWND)
Slow Start Threshold (SSTHRESH)

Packet Flow in Slow Start



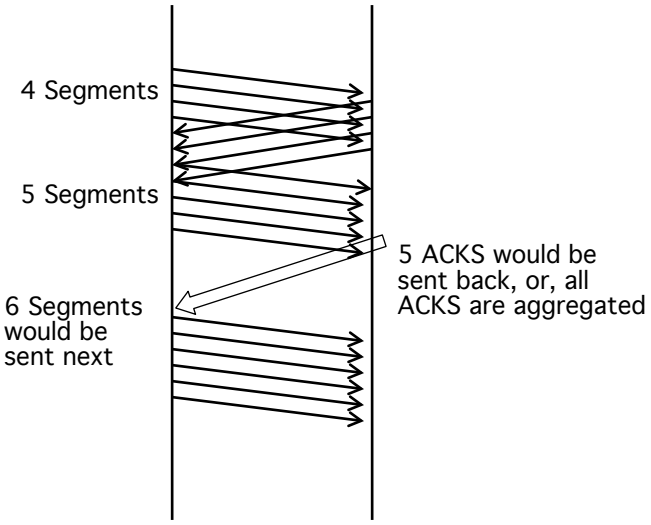
TCP Slow Start (2)



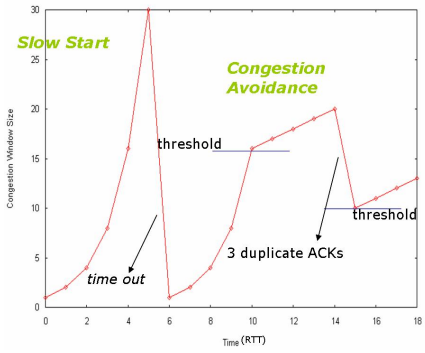
- The Slow Start Threshold is decreased after a loss
- This graph shows one segment sent at start of Slow Start phase

Source and Copyright of graph is
Computer Networks by Tanenbaum

Additive Increase in Congestion Avoidance



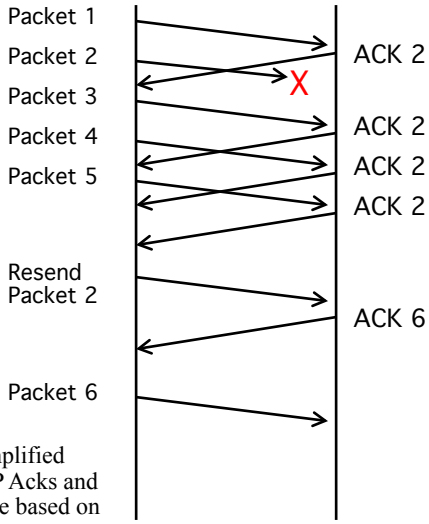
TCP Slow Start (3)
Fast Recovery and Fast Retransmit Algorithm



- Fast Retransmit occurs after three (3) duplicate Acks are received
- Fast Recovery means that CWND is not reduced to IW

Source and Copyright of graph is unknown

Fast Retransmit via Duplicate ACK



Note that this is a simplified diagram and that TCP Acks and Sequence numbers are based on the byte, not the segment or packet

Additional TCP Features

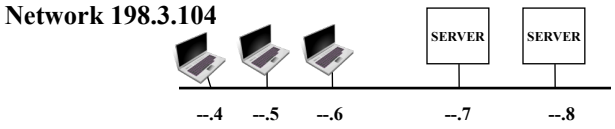
- TCP has Selective Acknowledgements (SACK)
- TCP connections can be aborted immediately by RST bit
- TCP uses a modified three way handshake to close connections
- TCP ports identify applications at each end of the connection via Port numbers
- TCP uses both static and dynamic port binding
- TCP checksum uses the same pseudo header as UDP

Some TCP Implementation Details

- Implementations are known as Tahoe, Reno, Vegas, illinois, westwood, etc.
- RTT Calculation (Jacobson 1988)
- Karn's Algorithm: don't update RTT on retransmitted segments.
- Nagle's algorithm (1984): addresses problem of sending one byte at a time
- Silly Window Syndrome (1982)
- The congestion control RFCs we have talked about
- The number of segments that are initially sent during Slow Start continues to change (RFC 3390 and recent IDs)

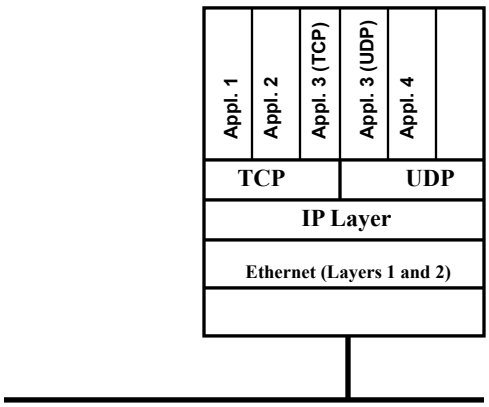
Connection Management

Application Layer Connection Management

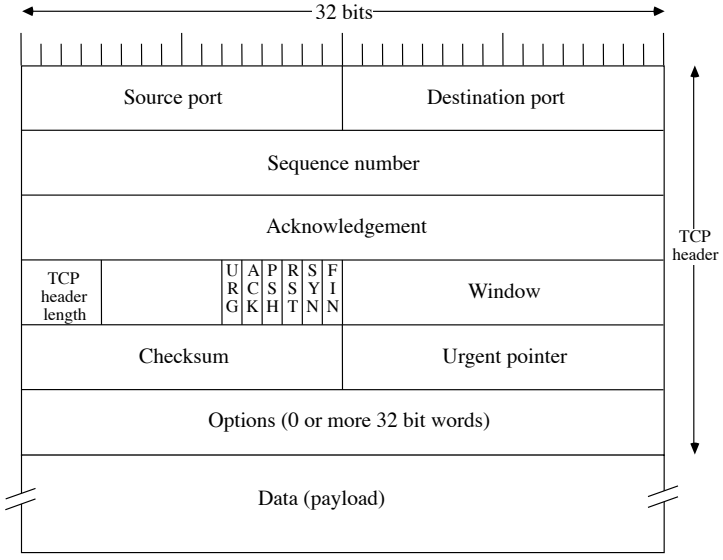


- **How does a system keep track of all of its application layer connections?**
- **Can we see the details of these connections?**

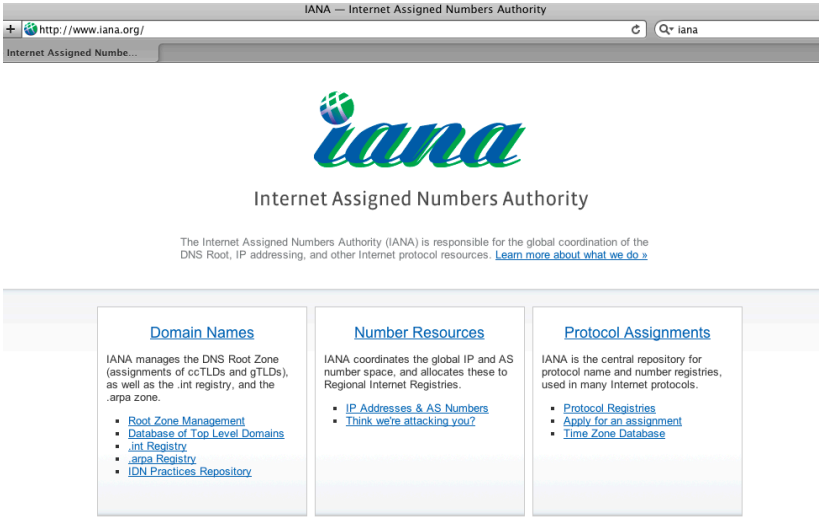
Application Layer Software Schematic



TCP Packet Header



WWW.IANA.ORG

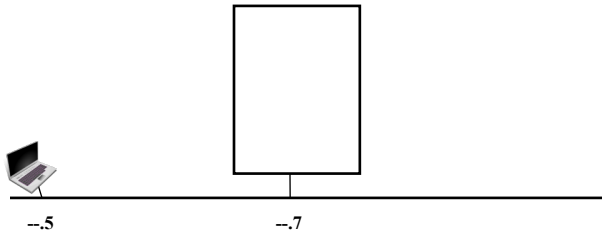


Some Well Known TCP Port Numbers

20,21	FTP	File transfer
22	SSH	Secure Shell
23	Telnet	Remote login, not encrypted
25	SMTP	Email
80	HTTP	world wide web
110	POP3	Remote email access
443	HTTPS	Encrypted web traffic
1720	H.323	Video conferencing
5060	SIP	Session Initiation Protocol

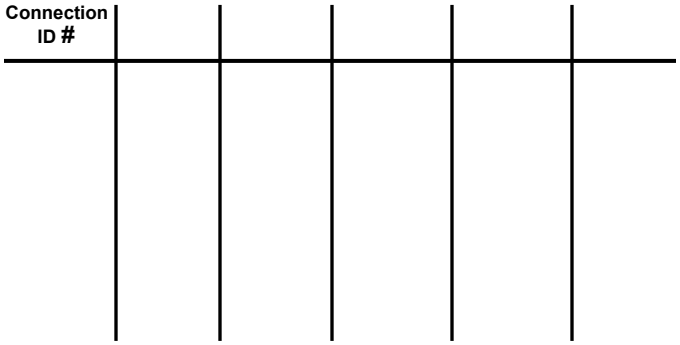
(SIP for VoIP also uses multiple dynamic ports)

Application Layer Connection Management (2)

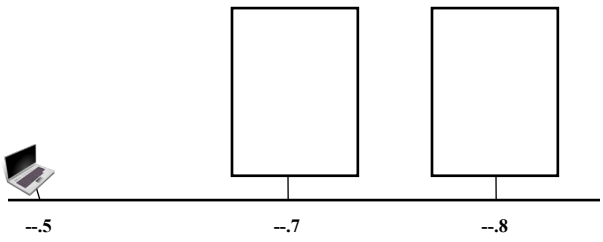


Network 198.3.104

Connection Management Table



Application Layer Connection Management (2)

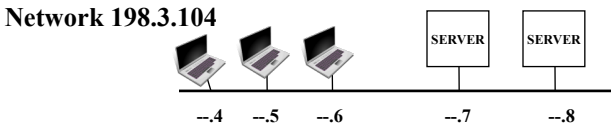


Network 198.3.104

Connection Management Table

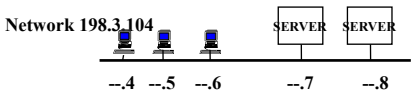
Connection ID #	Protocol (TCP/UDP)	Local IP	Remote IP	Local Port	Remote Port

Application Layer Connection Management



- **How does a system keep track of all of its application layer connections?**
- **Can we see the details of these connections?**

Connection Management Table



Connection ID #	Protocol (TCP/UDP)	Local IP	Remote IP	Local Port	Remote Port

netstat -an

```
tcp   140.247.30.107.80    24.60.123.123.1518      ESTABLISHED
tcp   140.247.30.107.23    24.60.234.234.2055      ESTABLISHED
tcp   140.247.30.107.25    24.60.222.221.2006      ESTABLISHED
tcp   140.247.30.107.110   134.174.111.222.1186    FIN_WAIT_2
tcp   140.247.30.107.143   134.174.123.213.1682    ESTABLISHED
tcp   140.247.30.107.80    134.174.212.121.1683    ESTABLISHED
tcp   140.247.30.107.22    24.60.33.22.1516        TIME_WAIT
tcp   *.80                 *.*                     LISTEN
tcp   *.443                *.*                     LISTEN
tcp   *.22                 *.*                     LISTEN
tcp…..
```


One Minute Wrap-Up

- Please do this Wrap-Up at the end of each lecture.
- Please fill out the form on the website.
- The form is anonymous (but you can include your name if you want.)
- Please answer three questions:
  - What is your grand "Aha" for today's class?
  - What concept did you find most confusing in today's class?
  - What questions should I address next time
- **Thank you!**