Review of Internet Architecture and Protocols

Professor Guevara Noubir
Northeastern University
noubir@ccs.neu.edu

Reference Textbooks:

Computer Networks: A Systems Approach, L. Peterson, B. Davie, Morgan Kaufmann

Learning Objectives

- Describe how the key Internet protocols operate and interface with each other:
  - Internet Protocol, addressing, IP over LAN/WLAN
  - Routing (RIP, OSPF, BGP)
  - End-to-end protocols (e.g., TCP, UDP)
  - Domain Name System
- Use socket programming APIs for network applications

Outline

Lesson 1: Internet Protocol

Lesson 2: IP Addressing

Lesson 3: IP over LAN

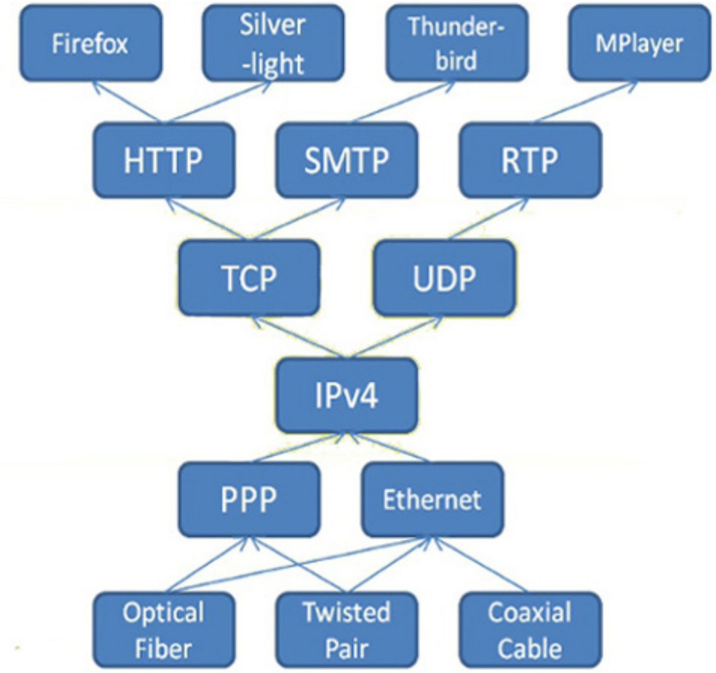
Lesson 4: Routing

Lesson 5: End-to-End protocols

Lesson 6: Naming

Lesson 1: IP – The Internet Protocol

- Goal: scalability
  - Interconnect a large number of heterogeneous networks
  - Support diverse application



- How: concatenation of networks

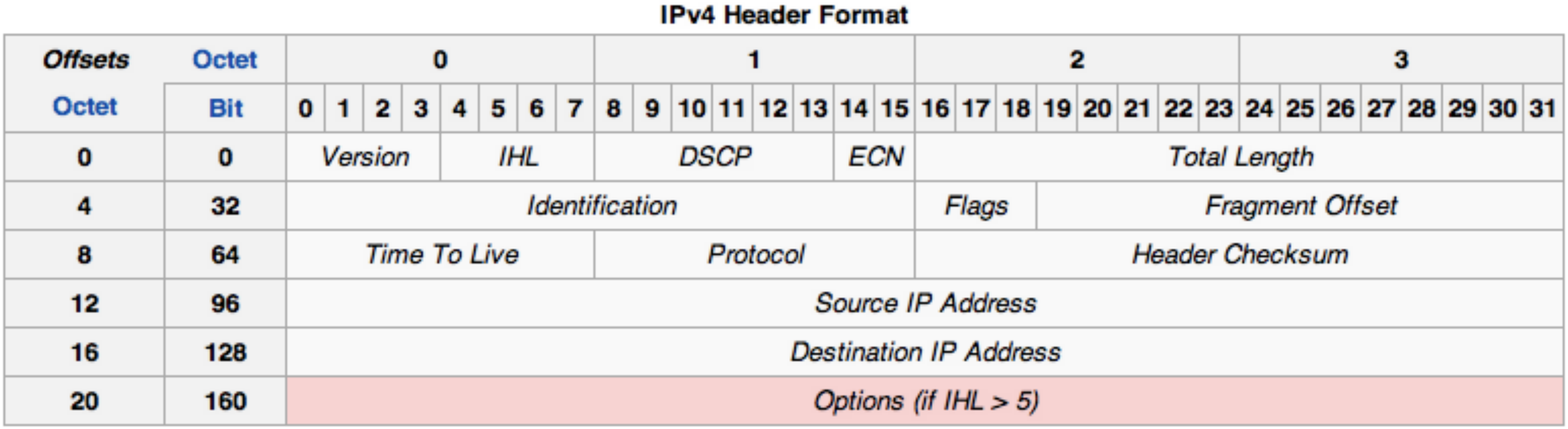
- Protocol Stack with the Internet Protocol (IP) as the focal point

IP Service Model

To keep routers simple and scalable IP choose:

- Connectionless (datagram-based)
- Best-effort delivery (unreliable service)
  - Packets can be lost, delayed, received out of order, or duplicate

IP packet format

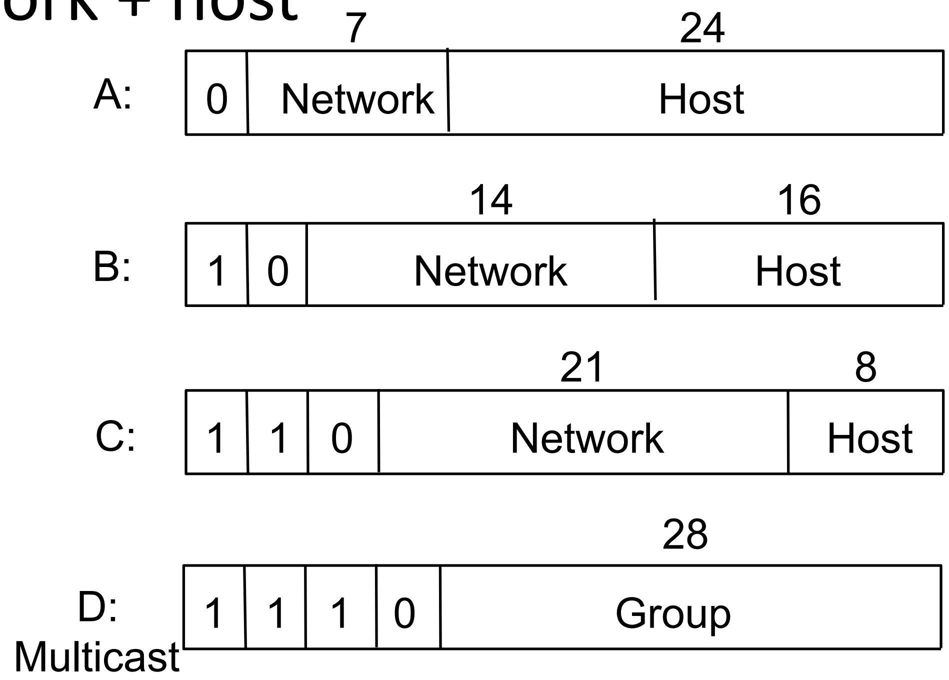


IPv4 Header Format

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	*Version*				*IHL*				*DSCP*						*ECN*		*Total Length*															
4	32	*Identification*																*Flags*			*Fragment Offset*												
8	64	*Time To Live*								*Protocol*								*Header Checksum*															
12	96	*Source IP Address*																															
16	128	*Destination IP Address*																															
20	160	*Options (if IHL > 5)*																															

Lesson 2: IP Addressing

- Properties of IP addresses
  - Globally unique (with some exceptions)
  - Hierarchical: network + host



A: 0 | Network (7) | Host (24)

B: 1 | 0 | Network (14) | Host (16)

C: 1 | 1 | 0 | Network (21) | Host (8)

D: Multicast: 1 | 1 | 1 | 0 | Group (28)

- Dot Notation
  - 10.3.2.4
  - 128.96.33.81
  - 192.168.69.77

Scaling IP Addresses

Assignment of IP addresses according to classes is inefficient:

- Inefficient use of Hierarchical Address Space
  - Class C with 2 hosts (2/256 = 0.78% efficient)
  - Class B with 255 hosts (255/65536 = 0.39% efficient)
- Still Too Many Networks
  - Routing tables do not scale
  - Route propagation protocols do not scale

Two solutions:

- Subnetting
  - Class B network 128.96.34.0 can be subdivided into two subnets
  - Subnet number: 128.96.34.0 with mask 255.255.255.128 and
  - Subnet number: 128.96.34.128 with mask 255.255.255.128
- Supernetting also called Classless Inter Domain Routing (CIDR)
  - Assign block of contiguous network numbers to nearby networks
  - Represent blocks with a single pair **`(first_network_address, count)`**
  - Restrict block sizes to powers of 2
  - E.g., 192.4.16 – 192.4.31: /20

Lesson 3: IP over LAN

Packet forwarding strategy:

- Every packet contains destination's address
- If directly connected to destination network, then forward to host (e.g., using appropriate MAC address)
- If not directly connected to destination network, then forward to some router (using MAC address of router)
- Forwarding table maps network number into next hop
- Each host has a default router
- Each router maintains a forwarding table

Forwarding an IP packet on an ethernet link requires the knowledge of the MAC address of the next hop.

- Question: how?

Address Translation

To forward a packet, nodes need to map IP addresses into a link layer addresses. The link layer address could be the address of:

- Destination host
- Next hop router

Possible techniques:

- Encoding the link layer address in the host part of IP address is not practical
- Maintain a table

Address Resolution Protocol (ARP) maintains a table of IP to physical (link-layer) address mapping by

- broadcasting request if IP address not in table
- target machine responds with its physical address
- table entries are discarded if not refreshed

ARP Details

Request Format:

- HardwareType: type of physical network (e.g., Ethernet)
- ProtocolType: type of higher layer protocol (e.g., IP)
- HLEN & PLEN: length of physical and protocol addresses
- Operation: request or response
- Source/Target-Physical/Protocol addresses

ARP Rules:

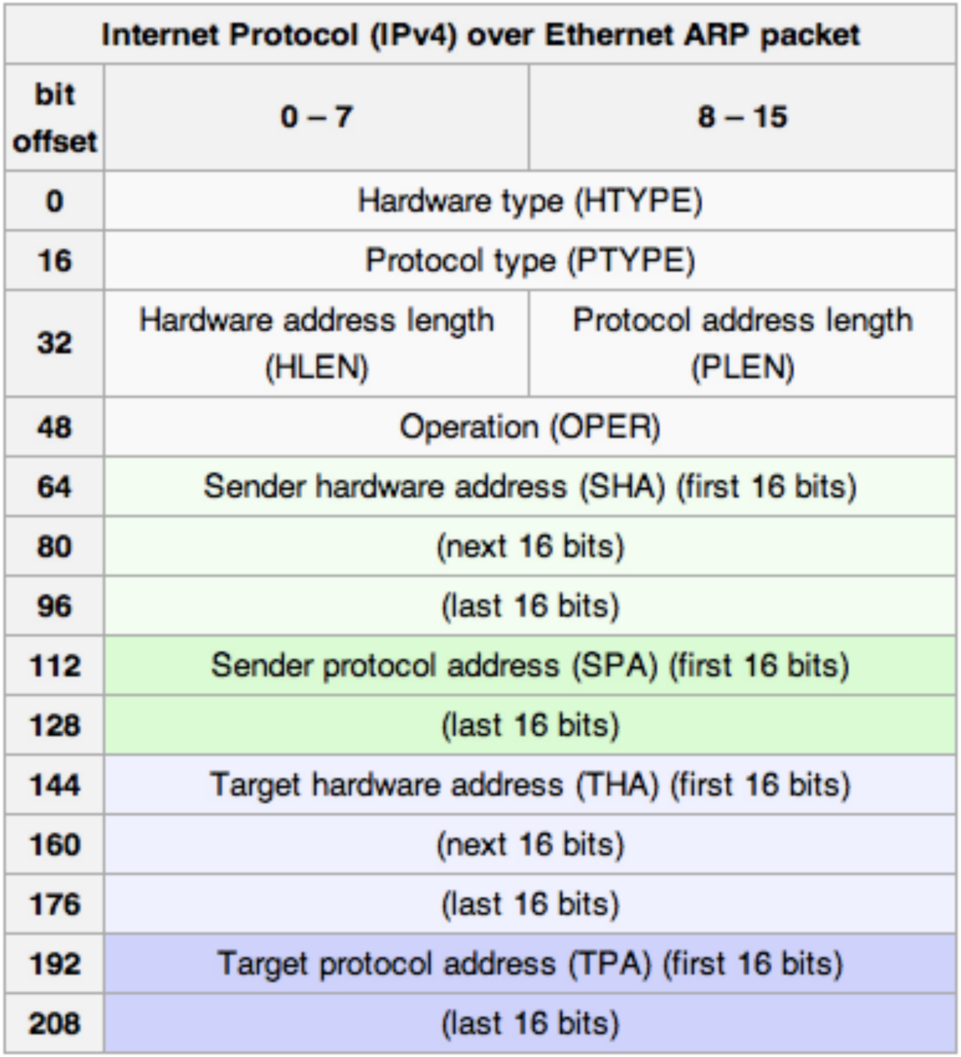
- Table entries typically timeout in 15 minutes
- Update table with source when you are the target
- Update table if already have an entry
- Do not refresh table entries upon reference

Example of table:

```
firenze:~ noubir$ arp -a
babel-115.ccs.neu.edu (129.10.115.1) at 0:e:d6:5:b4:0 on en0 [ethernet]
arora.ccs.neu.edu (129.10.115.132) at 0:50:56:be:64:c0 on en0 [ethernet]
crew-netmon-0.ccs.neu.edu (129.10.115.195) at 0:50:56:ad:0:9 on en0 [ethernet]
```

ARP has security vulnerabilities called ARP Poisoning to be practiced in the man-in-the-middle attacks laboratory

ARP Packet Format



Internet Protocol (IPv4) over Ethernet ARP packet		
bit offset	**0 – 7**	**8 – 15**
0	Hardware type (HTYPE)	
16	Protocol type (PTYPE)	
32	Hardware address length (HLEN)	Protocol address length (PLEN)
48	Operation (OPER)	
64	Sender hardware address (SHA) (first 16 bits)	
80	(next 16 bits)	
96	(last 16 bits)	
112	Sender protocol address (SPA) (first 16 bits)	
128	(last 16 bits)	
144	Target hardware address (THA) (first 16 bits)	
160	(next 16 bits)	
176	(last 16 bits)	
192	Target protocol address (TPA) (first 16 bits)	
208	(last 16 bits)	

Internet Control Message Protocol (ICMP) RFC 792

- Corresponds to ProtocolType = 1 in the IP packet header
- Important for network diagnosis
- Example of ICMP Codes:
  - Echo (ping)
  - Redirect (from router to inform source host of better route)
  - Destination unreachable (protocol, port, or host)
  - TTL exceeded (so datagrams don't cycle forever)
  - Checksum failed
  - Reassembly failed
- Discuss use in traceroute utility, MTU discovery

Dynamic Host Configuration Protocol (DHCP)

- IP addresses of interfaces cannot be configured at manufacturing phase (like for Ethernet) because they are location dependent
- Configuration is an error-prone process
- Solution: centralize the configuration information in a DHCP server:
    - DHCP server discovery: broadcast a DHCPDISCOVER request
    - Request are relayed (unicast) to the server by DHCP relays
    - DHCP server broadcast replies with <HWADDR, IPADDR, lease-info>
- Runs on top of UDP

Lesson 4: Routing Overview

Forwarding vs Routing processes

- Forwarding: to select an output port based on destination address and routing table
- Routing: process by which the routing table is built

Routing:

- Network can be modeled as a graph
- Problem: find a path between two nodes

Factors

- Cost: bandwidth, delay, reliability
- Policies between backbone providers

Two approaches to building routing tables

- Distance Vector and Link State protocols

Two classes of routing protocols

- Intra-domain routing (within an Autonomous System) e.g., RIP or OSPF
- Inter-domain routing (across AS) also Exterior Gateway Protocol e.g., BGP

Distance Vector Routing Protocols

- Each node maintains a set of triples
  - `(Destination, Cost, NextHop)`
- Exchange updates directly with neighboring routers
  - Periodically (on the order of several seconds)
  - Whenever table changes (called *triggered* update)
- Updates are a list of pairs that report the cost to reach destinations
  - `(Destination, Cost)`
- Routers update their local table if they receive a “better” route
  - Lower cost
  - Came from next-hop
- Updates result in refresh existing routes (delete routes on time out)
- Limitations: potential formation of loops when links break

Routing Information Protocol (RIP)

- Implements a distance vector approach (Bellman-Ford's algorithm)
- Protocol runs over UDP, port 520
- Protocol overview:
  - Init: send a request packet over all interfaces
  - On response reception: update the routing table
  - On request reception:
    - If request for complete table (*address family*=0) send the complete table
    - Else send reply for the specified address (infinity=16)
  - Regular routing updates:
    - Every 30 seconds part/entire routing table is sent (broadcast) to neighboring routers
  - Triggered updates: on metric change for a route
  - Has a simple authentication scheme

Link State Routing Protocols

- Strategy
  - Flood links information: send to all nodes (not just neighbors) information about directly connected links (not entire routing table)
  - Each node has a global view of the networks links and can locally compute paths
- Link State Packet (LSP)
  - Id of the node that created the LSP
  - Cost of link to each directly connected neighbor
  - Sequence number (SEQNO) to avoid loops
  - Time-to-live (TTL) for this packet

Link State (cont)

Reliable flooding

- Store most recent LSP from each node
- Forward LSP to all nodes but one that sent it
- Do not forward already received LSPs
- Generate new LSP periodically
  - Increment SEQNO
- Start SEQNO at 0 when reboot
- Decrement TTL of each stored LSP
  - Discard when TTL=0

Open Shortest Path First

- IP protocol (not over UDP), reliable (sequence numbers, acks)
- Protocol overview: link state protocol
  - The link status (cost) is sent/forwarded to all routers (LSP)
  - Each router knows the exact topology of the network
  - Each router can compute a route to any address
  - simple authentication scheme
- Advantages over RIP
  - Faster to converge
  - The router can compute multiple routes (e.g., depending on the type of services, load balancing)
  - Use of multicasting instead of broadcasting (concentrate on OSPF routers)

Popular Interior Gateway Protocols

- RIP: Route Information Protocol
  - Distributed with Unix
  - Distance-vector algorithm
  - Based on hop-count
- OSPF: Open Shortest Path First
  - Uses link-state algorithm
  - Supports load balancing
  - Supports basic integrity check http://www.faqs.org/rfcs/rfc2328.html

Exterior Gateway Protocols BGP-4: Border Gateway Protocol

The Internet is composed of Autonomous Systems (AS)

- Stub AS: has a single connection to another AS
- Multihomed AS: has connections to more than one AS
  - Does not carry transit traffic for other AS
- Transit AS: has connections to more than one AS
  - Carries both transit and local traffic

Each AS has:

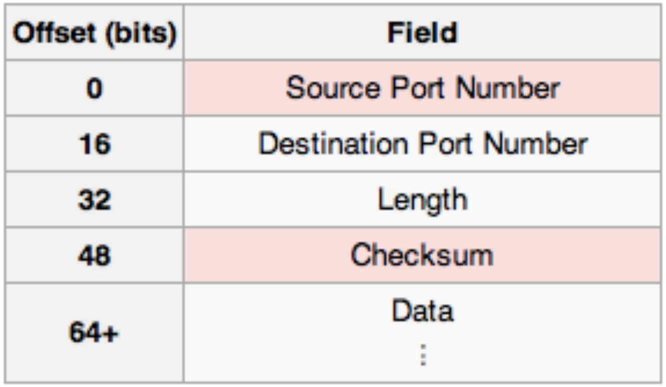
- One or more border routers
- At least one BGP *speaker* that advertises:
  - Local networks, and other reachable networks (transit AS only)
  - Advertises complete *path* of AS to reach destination
  - Possibility to withdraw path
- In the backbone BGP speakers inject learned information using IBGP + intradomain routing protocol to reach border routers

BGP-4 is vulnerable to attacks such as IP addresses hijacking and misconfigurations

Lesson 5: End-to-End Protocols

Simple Demultiplexor (UDP)

- Simplest possible protocol for application to application communication over the Internet
  - ○ Adds multiplexing to IP using SrcPort and DstPort
- Unreliable and unordered datagram service
- No flow control
- Endpoints identified by ports
  - ○ Servers have *well-known* ports (e.g., DNS: port 53, talk: 517, smtp: 25)
  - ○ see **`/etc/services`** on Unix
- Header format



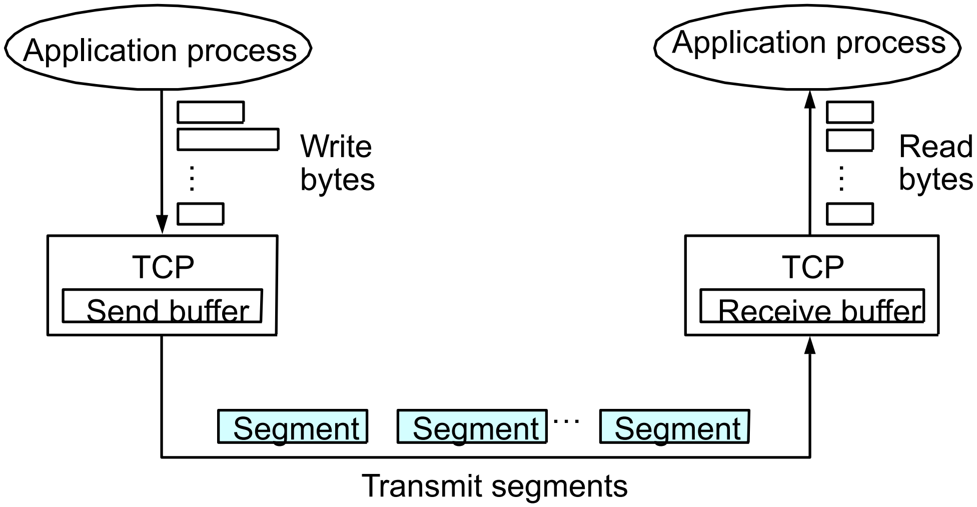
Offset (bits)	Field
0	Source Port Number
16	Destination Port Number
32	Length
48	Checksum
64+	Data ⋮

- Optional checksum
  - ○ Pseudo header + UDP header + data
  - ○ Pseudo header = protocol number, source IP addr, dest IP addr, UDP length

End-to-End Protocols
Transport Control Protocol (TCP)

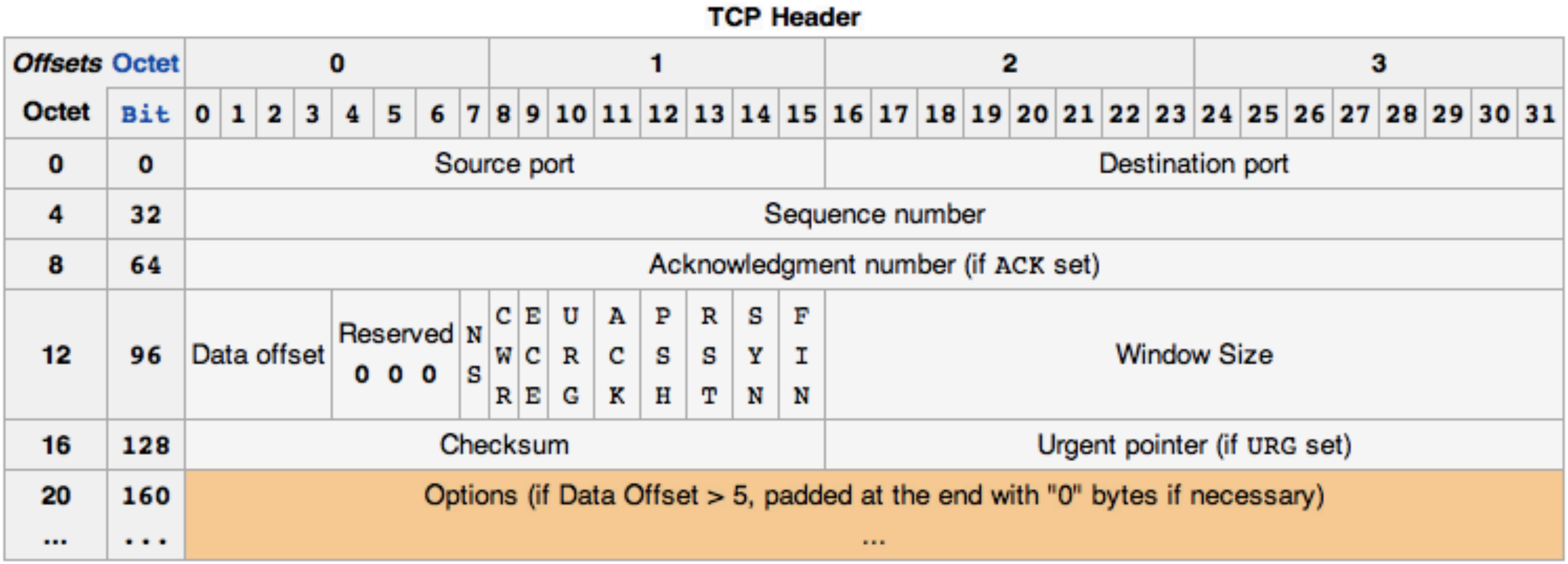
- Reliable
- Connection-oriented
- Byte-stream
  - Application writes bytes
  - TCP sends *segments*
  - Application reads bytes

- Key mechanisms
  - Connection establishment using a handshake protocol: SYN, ACK/SYN, ACK, FIN
  - Flow control prevents the sender from overrunning the receiver
  - Congestion control prevents the sender from overrunning network



Transport Control Protocol (TCP)

- Each connection is uniquely identified by the 4-tuple:
  - `(SrcPort, SrcIPAddr, DsrPort, DstIPAddr)`
- Sliding window and flow control
  - `acknowledgment, SequenceNum, AdvertisedWindow`



TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0			N S	C W R	E C E	U R G	A C K	P S H	R S T	S Y N	F I N	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
20 ...	160 ...	Options (if Data Offset > 5, padded at the end with "0" bytes if necessary) ...																															

Lesson 6: Naming: Domain Name System (DNS)

- Naming is a general paradigm for mapping abstract names to physical resources (e.g., name to IP address or name to email address)

- DNS is a fundamental application layer protocol, not visible but invoked every time a remote site is accessed

```
madrid.ccs.neu.edu -> 129.10.112.229
```

- The domain names are organized as hierarchical zones starting at the Top Level Domains (TLD) (.net, .com, .edu, etc.)
  - Each zone has at least two dns servers
- DNS runs on top of UDP port 53 (and also in a limited way on top of TCP port 53)
- The DNS resolver (client) hierarchically queries DNS servers until it obtains the mapping between a name/resource and an IP address

DNS Resource Records

- Each name server maintains a collection of *resource records*
  `(Name, Value, Type, Class, TTL)`
- Name/Value: not necessarily host names to IP addresses
- Type
  - A: Value is an IP address
  - NS: Value gives domain name for host running name server that knows how to resolve names within specified domain.
  - CNAME: Value gives canonical name for particle host; used to define aliases.
  - MX: Value gives domain name for host running mail server that accepts messages for specified domain.
- Class: allow other entities to define types
  - IN: Means Internet
- TTL: how long the resource record is valid

DNS Typical Query

```
cosmicboard:~ noubir$ dig @129.10.116.61 ccs.neu.edu -t any

; <<>> DiG 9.7.3-P3 <<>> @129.10.116.61 ccs.neu.edu -t any
;; QUESTION SECTION:
;ccs.neu.edu.                          IN            ANY

;; ANSWER SECTION:
ccs.neu.edu.              300          IN            SOA          amber.ccs.neu.edu. hostmaster.ccs.neu.edu.
     2012092400 10800 1800 604800 300
ccs.neu.edu.              300          IN            NS           amber.ccs.neu.edu.
ccs.neu.edu.              300          IN            NS           asgard.ccs.neu.edu.
ccs.neu.edu.              300          IN            NS           tigana.ccs.neu.edu.
ccs.neu.edu.              300          IN            NS           alderaan.ccs.neu.edu.
ccs.neu.edu.              300          IN            NS           rivendell.ccs.neu.edu.
ccs.neu.edu.              300          IN            NS           mcs.anl.gov.
ccs.neu.edu.              300          IN            NS           joppa.ccs.neu.edu.
ccs.neu.edu.              300          IN            A            129.10.116.51
ccs.neu.edu.              300          IN            MX           50 atlantis.ccs.neu.edu.
ccs.neu.edu.              300          IN            MX           10 amber.ccs.neu.edu.

;; ADDITIONAL SECTION:
amber.ccs.neu.edu.        300          IN            A            129.10.116.51
joppa.ccs.neu.edu.        300          IN            A            129.10.116.53
asgard.ccs.neu.edu.       300          IN            A            129.10.116.61
tigana.ccs.neu.edu.       300          IN            A            129.10.116.83
alderaan.ccs.neu.edu.     300          IN            A            129.10.116.80
rivendell.ccs.neu.edu.    300          IN            A            129.10.116.52
atlantis.ccs.neu.edu.     300          IN            A            129.10.116.41

;; Query time: 6 msec
;; SERVER: 129.10.116.61#53(129.10.116.61)
```

Demonstration

Important Protocols

- ARP: Address Resolution Protocol
- BGP: Border Gateway Protocol
- CIDR: Classless Inter Domain Routing
- DHCP: Dynamic Host Configuration Protocol
- DNS: Domain Name System
- ICMP: Internet Control Message Protocol
- IP: Internet Protocol
- LAN: Local Area Network
- OSPF: Open Shortest Path First
- RIP: Routing Information Protocol
- TCP: Transport Control Protocol
- UDP: User Datagram Protocol

Summary

- Multi-layer stack of protocols:
  - Link Layer: ethernet (IEEE802.3), FDDI, ATM, wlan (IEEE802.11)
  - Network Layer:
    - Internet Protocol (IP) is a focal point
    - Routing protocols: RIP, OSPF, BGP-4
  - Transport Layer: UDP, TCP
  - Naming: DNS
- How do these protocols fit with each other?
- What is the syntax and semantic of typical packets (e.g., TCP, IP, UDP)
- What are the important mechanisms (e.g., TCP handshake, DNS resolution)