L8-9: Application Layer

Possible structures: P2P, client-server

DNS – Domain Name System

* Link page name to IP address
* Distributed database in hierarchy of many *name servers*
* IP address: hierarchical
* NOT centralised – cuts traffic, one point of failure, easier maintenance
* Root DNS servers => com/org/edu/etc servers => amazon/uon/etc servers
  + Assume client wants amazon.com
    - Queries root server to find .com server
    - Queries .com server to find amazon.com server
    - Queries amazon.com server to get IP address for [www.amazon.com](http://www.amazon.com).
* Local DNS server stores location in cache for quick reaccess but refreshes every 2 days (in case location is updated)

DNS Records

* Distributed database storing resource records (RRs)
* RR format: (name, value, type, ttl)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **type** | A | NS | CNAME | MX |
| **desc** | Address record | Name server record | Canonical name record | Mail exchange record |
| **name** | hostname | domain | alias name, i.e. [www.ibm.com](http://www.ibm.com) |  |
| **value** | IP | hostname of authoritative DNS name server | canonical name, e.g. servereast.backup2.ibm.com | mailserver associated with **name** |

L10: Application Layer 2

File Distribution Time – Client-Server

* server transmission: must sequentially send *N* file copies
  + *F/us* to send one copy (*F* = file size, *us* = server upload capacity)
  + *NF/us* to send *N* copies.
* client side: each client must download file copy
  + *dmin* = min client download rate
  + *F/dmin* = min client download time

*Dc-s* ≥ *max*{*NF/us*, *F/dmin*} = time to distribute *F* to *N* clients

File Distribution Time – P2P

* server transmission: must sequentially send *N* file copies
  + *F/us* to send one copy (*F* = file size, *us* = server upload capacity)
* client side: each client must download file copy
  + *dmin* = min client download rate
  + *F/dmin* = min client download time
  + must download *NF* bits
    - max upload rate = *us + ∑ui*
  + *Dc-s* ≥ *max*{*F/us*, *F/dmin*, *NF/*(*us + ∑ui*)} = time to distribute *F* to *N* clients

Multimedia Video

* coding: check redundance between frames to minimise bits used to encode image.
  + spatial: instead of all values, send 2: the value (e.g. pixel colour) and number of repeated values (e.g. 30 frames)
  + temporal: send differences between frames *i* and *i*+1.

L11: Transport Layer 1

UDP

* Source port #, dest port #, length of segment (including header), checksum (all 16 bits each), application data.
* No connection establishment/state at sender, receiver.
* No congestion control.
* Tiny header (64 bits)

UDP Checksum

* Check for error in transmitted segment
* sender:
  + calculate checksum (addition of segment contents)
  + add checksum value to field
* receiver:
  + compute and confirm checksum of received segment

Internet Checksum

* Wraparound carryover
* Compliment sum to get checksum
* Check with all values EXCEPT (obviously) checksum

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| int 1 |  | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| int 2 |  | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| wraparound | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| sum |  | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| checksum |  | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |

Reliable Data Transfer COVER MORE

* Recover from bit errors with ACKs (all good) or NAKs (negative acknowledgements)
* rdt2.1 – ACKs and NAKs
* rdt2.2 – only ACKs, use ACK for last packet received instead. MUST include seq # of packet. For duplicate ACK, retransmit packet
* rdt3.0 – assume channel loses packets as well
  + approach: sender wait reasonable amount of time for ACK, retransmits if not received in time
  + if just delayed, rdt already handles duplicates
  + 4 scenarios: all clear, packet loss, ACK loss, premature timeout
    - all clear
  + poor performance

Pipelining for Increased Utilisation

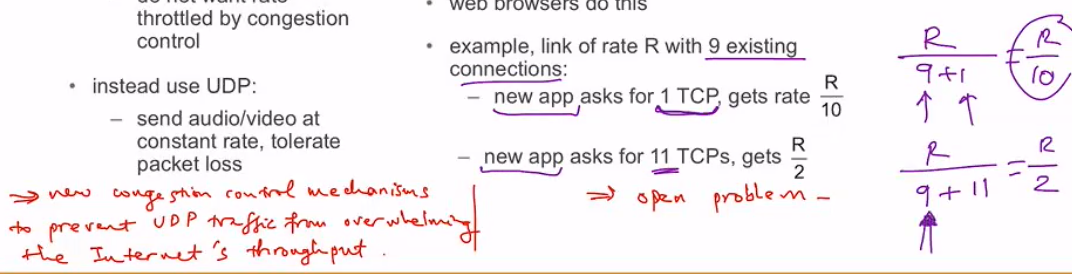
* Transmit multiple packets at a time
* 3-packet pipelining increases utilisation by a factor of 3.

L13:

TCP RTT; timeout

* timeout interval = Estimated RTT + 4\*DevRTT

L14: s



L15: Network Layer Overview and Routers

Destination-Based Forwarding – MORE

Switching Fabrics

* memory: packet copied to system’s memory, speed limited by memory bandwidth
* bus: packet from input to output port memory via shared bus, speed limited by bus bandwidth
* interconnection network (crossbar, banyan, etc): interconnection nets to handle multiprocessing, overcomes bus bandwidth limitations

Input Port Queueing

* Queueing occurs where fabric slower than combined input ports.
* Head-of-the-Line (HOL) blocking: queueing datagram in front prevents others from moving forward.

Output Ports

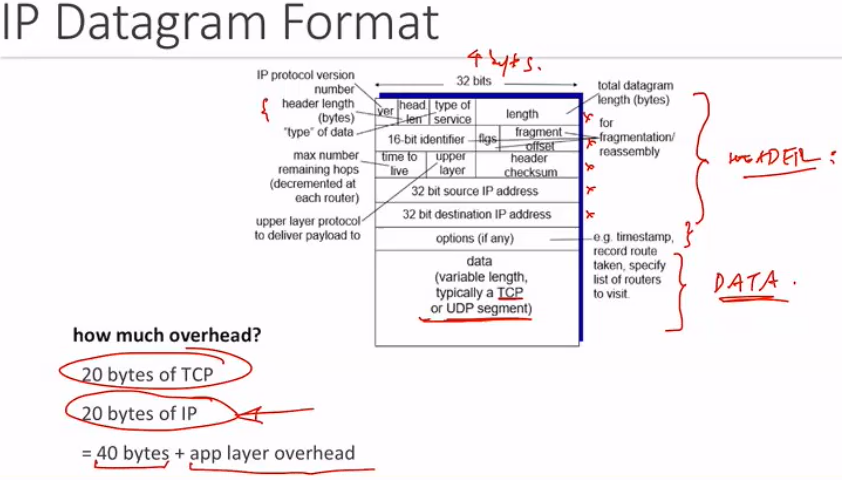
* buffering: datagrams (packets) arrive faster than the transmission rate.
  + packets can be lost due to lack of buffer.
  + With *N* flows and link capacity *C*, buffering =
* scheduling discipline: datagram selected for transmission first.

Scheduling Mechanisms/Policies

* Priority: send highest priority queued packet.
  + priority based on classes, classes defined by header info, marking, etc.
* Round Robin: scan class queues, sending one packet from each class
* Weighted Fair Queueing (WFQ): form of Round Robin.
  + each class has a weight, higher priority queues have more packets taken before moving on

L16: Network Layer – Internet

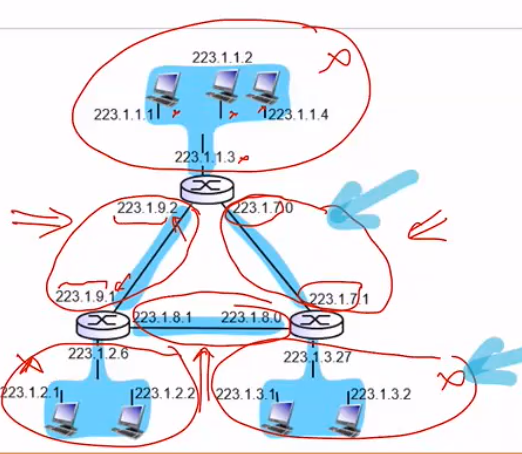
|  |
| --- |
| Transport Layer – TCP, UDP (20 bytes) |
| * routing protocols ⬄ forwarding table * path selection * RIP, OSPF, BGP * IP protocol * addressing convention * datagram format * packet handling conventions * ICMP protocol * error reporting * router signaling |
| Data (Link) Layer |
| Physical Layer |



IP Addressing

* 32-bit identifier for host, router interface
* interface: connection between host/router and physical link
  + routers have multiple interfaces
  + host has one or two interfaces (e.g. wired Ethernet, wireless 802.11).
* CIDR: Classless Inter Domain Routing
  + subnet address portion of arbitrary length
  + address format: a.b.c.d./x, where x is no. of bits in subnet

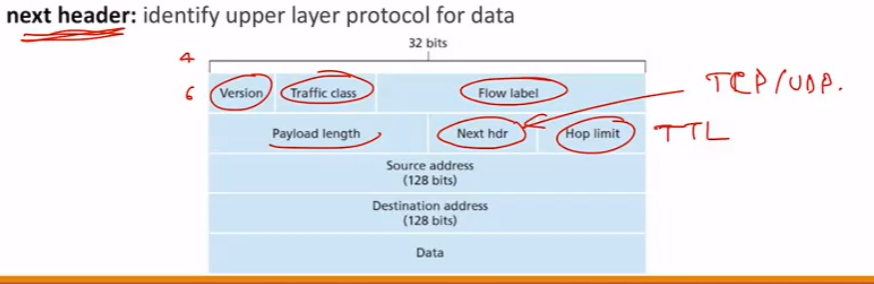
Subnets



IP Addresses: getting one

* Dynamic Host Configuration Protocol (DHCP)
  + Dynamically create IP when host joins network
  + Only hold address while connected, reassign when not in use
  + overview
    - host broadcasts *DHCP discover* to search for DHCP server.
    - DHCP server responds with *DHCP offer.*
    - host sends *DHCP request*.
    - DHCP server sends *DHCP ack*.
* Network: gets subnet part from ISP’s address space.

L17: IPv6



Changes from IPv4

* no checksum to improve processing time
* options allow, but outside of header
* ICMPv6
  + additional packet types. e.g “Packet Too Big”

Transition for IPv4 to IPv6

* Not all routers can be simultaneously changed – how will mixed routers work?
* tunnelling: IPv6 datagrams carried as payload to IPv4 routers.
  + requires source and dest addresses that IPv4 routers are *tunnelling between.*

L18: Network Layer Routing Protocols

Routing Algorithm Classification

* Global
  + routers have complete topology, cost info
  + “link state” algorithms
* Decentralised
  + routers know neighbours and link costs to neighbours
  + iterative process of info exchange
  + “distance vector” algorithms

L19:

Scalable Routing

* Aggregate routers into regions – “Autonomous Systems” (AS) or domains.
* Intra-AS Routing: routing among hosts in the same network
  + All routers in same AS must run same protocol
  + Routers in different AS can run different protocol
  + Gateway router has links to other AS’s.
* Inter-AS Routing: routing among different AS’s
  + Gateway router can perform intra- or inter-domain routing
* Forwarding table configurated by both systems.

Intra-AS Routing

* Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP) & Open Shortest Path First (OSPF) – focus on OSPF
* OSPF: Open, link-state
* router flood link-state adverts to all other routes in the AS
  + Messages carried over IP, not TCP/UDP.

Internet inter-AS Routing: BGP

* Border Gateway Protocol is one of the most important inter-domain protocols “glue that holds the internet together”.
* eBGP: obtain subnet reachability information from neighbouring AS’s
* iBGP:

Review shorthand

Source Switch Router Destination

|  |
| --- |
| Application |
| Transport |
| Network |
| Data |
| Physical |

|  |
| --- |
| Network |
| Data |
| Physical |



|  |  |
| --- | --- |
| Message | Application |
| Segment | Transport |
| Datagram | Network |
| Frame | Data |
|  | Physical |

|  |
| --- |
| Data (MAC) |
| Physical |

Inter- vs Intra-AS protocols

Policy: There is a risk of incompatible policies clashing. With a more localised intra-AS, standardising policies is made easier.

Scale: Problems of scale are more easily controlled in intra-AS. If an AS gets overly large, it can split into two AS’s connected by inter-AS routing rather than become bloated.

Performance: Since inter-AS routing is policy-oriented, this allows performance issues to be addressed within the AS.

congestion control vs flow control

FRAMES

ARP

Switches

MAC protocols

ALOHA normalised throughput = 0.184  
Slotted ALOHA normalised throughput = 0.368

Your computer first uses DHCP to obtain an IP address. Your computer first creates a special IP datagram destined to 255.255.255.255 in the DHCP server discovery step, and puts it in an Ethernet frame and broadcast it in the Ethernet. Then following the steps in the DHCP protocol, your computer is able to get an IP address with a given lease time.

A DHCP server on the Ethernet also gives your computer a list of IP addresses of first-hop routers, the subnet mask of the subnet where your computer resides, and the addresses of local DNS servers (if they exist).

Since your computer’s ARP cache is initially empty, your computer will use ARP protocol to get the MAC addresses of the first-hop router and the local DNS server. Your computer first will get the IP address of the Web page you would like to download. If the local DNS server does not have the IP address, then your computer will use DNS protocol to find the IP address of the Web page.

Once your computer has the IP address of the Web page, then it will send out the HTTP request via the first-hop router if the Web page does not reside in a local Web server. The HTTP request message will be segmented and encapsulated into TCP packets, and then further encapsulated into IP packets, and finally encapsulated into Ethernet frames. Your comp uter sends the Ethernet frames destined to the first-hop router. Once the router receives the frames, it passes them up into IP layer, checks its routing table, and then sends the packets to the right interface out of all of its interfaces.

Then your IP packets will be routed through the Internet until they reach the Web server.

The server hosting the Web page will send back the Web page to your computer via HTTP response messages. Those messages will be encapsulated into TCP packets and then further into IP packets. Those IP packets follow IP routes and finally reach your first-hop router, and then the router will forward those IP packets to your computer by encapsulating them into Ethernet frames.