ECE423/544: Computer and Communication Networks Spring 2020

D. Raychaudhuri
Final Review

Includes teaching materials from L. Peterson, J. Kurose, K. Almeroth
Definition of a Network

A network is an inter-connected set of technology components designed to provide data transfer between attached end points.
Network Topologies

- Ring
- Mesh
- Star
- Fully Connected
- Line
- Tree
- Bus
Protocols

• On top of a packet switched network, need
• Set of rules governing communication between network elements (applications, hosts, routers)
• Protocols define:
  – format and order of messages
  – actions taken on receipt of a message
Layering

TCP/IP - model

- Application
- Transport
- Internet
- Network interface
Packet Headers

Packet Headers can contain:
- addresses, flow ID, pkt type, service type, error checks, QoS, …
Encapsulation
Socket Basics

- **Client**
  - Create Socket: `socket`
  - Bind socket to port: `bind`
  - Send the connect request to server and waiting server to respond: `Connect`
  - Process inside the bracket will happen when request of client is accepted:
    - `recvfrom/sendto`
    - `close`

- **Server**
  - Create Socket: `socket`
  - Bind socket to port: `bind`
  - Listen and Accept:
    - `Listen`
    - `Accept`
    - `Client traffic is lining up here and waiting to be accepted`
    - `Client accepted will be ready for send and receive data between server`
  - Send and receive data:
    - `sendto/recvfrom`
    - `close`
High Level Design

Technology choice (e.g. MPLS optical)

Technology choice (e.g. IP router)

Technology choice (e.g. Ethernet SW)

Technology choice (e.g. 802.11n)

Access Net

Mbpl needed?

Physical Span?

Users
(#, density, mobility)

bps
Pkt size
Burst statistics
Stream parameters

bps/sq-m for wireless access
Network Design

- Network design = Architecture + topology + protocol specification
- Protocol specification =
  - Packet structure (control & data) & layering
  - Syntax
  - Semantics
  - Algorithm (routing, mcast, etc.)
CSMA with Collision Detection

CSMA/CD (Carrier Sense Multiple Access with Collision Detection) protocol further improves ALOHA by aborting transmissions as soon as a collision is detected.

Operation:
• To send data, a node first listens to the channel to see if anyone else is transmitting.
• If not, it transmits a frame
• If channel busy, deferral as in CSMA
  – the node wait a random period of time and repeats the algorithm (non-persistent), or waits until the end of the transmission (1-persistent)
• The node will detect the collision, if collision detected, abort its transmission (reducing channel wastage), waits a random amount of time, and starts all over again.
Ethernet Overview

• **History**
  - developed by Xerox PARC in mid-1970s
  - roots in Aloha packet-radio network
  - standardized by Xerox, DEC, and Intel in 1978
  - similar to IEEE 802.3 standard

• **CSMA/CD**

• **Evolution: Bus topology (90’s) ➔ Star topology (now)**

• **Most successful access network technology**
CSMA/CA

- Wireless LANs
- How can a node detect collision if it cannot listen while talking?
- Collision Avoidance
  - Random Backoff (instead of 1-persistent)
  - Request-to-send (RTS)/clear-to-send (CTS)
- CS no longer works well
  - Rules:
    - carrier  ==>  do not transmit
    - no carrier  ==>  OK to transmit
  - But the above rules do not always apply to wireless.
802.11 - MAC layer

- **Priorities**
  - defined through different inter frame spaces
  - no guaranteed, hard priorities
  - SIFS (Short Inter Frame Spacing)
    - highest priority, for ACK, CTS, polling response
  - PIFS (PCF IFS)
    - medium priority, for time-bounded service using PCF
  - DIFS (DCF, Distributed Coordination Function IFS)
    - lowest priority, for asynchronous data service

Source: P. Bhagwat
802.11 MAC protocol: CSMA/CA

- Use CSMA with collision Avoidance
  - Based on carrier sense function in PHY called Clear Channel Assessment (CCA)
- Reduce collision probability where mostly needed
- Efficient backoff algorithm stable at high loads
- Possible to implement different fixed priority levels

P. Bhagwat
Bridges/LAN switches

• Link layer device
  – stores and forwards frames
  – examines frame header and **selectively** forwards frame based on MAC dest address
  – when frame is to be forwarded on segment, uses the corresponding MAC to access segment (e.g. CSMA/CD for Ethernet)
A Switched Enterprise Network
### Forwarding

- Which port to forward a frame?
  - Use forwarding database/table
  
  ![MAC address, port, Time-to-Life (TTL)]

- How to build the forwarding table???
  - A routing problem

#### Forwarding Table

<table>
<thead>
<tr>
<th>Destination</th>
<th>Port</th>
<th>Time-to-Live (TTL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>H</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

---

![Network Diagram]
Self Learning (Learning Bridges)

- Forwarding tables entries are set automatically with a simple heuristic:
  - The source address field of a frame that arrives on a port tells which host is reachable from this port.
  - When a frame received, switch “learns” location of sender
  - records sender/location pair in forwarding table with TTL = MAX_TTL
    - TTL reset to MAX_TTL every time a frame with the same source addr is received to refresh the existing table entry
    - Entry removed when TTL counts down to 0
Spanning Tree Protocol Example

1. B3 receives (B2,0,B2)
2. Since 2<3, B3 accepts B2 as root
3. B3 adds 1 to distance advertised by B2 and sends (B2,1,B3) to B5
4. Meanwhile B2 accepts B1 as root because it has lower ID and sends (B1,1,B2) to B3
5. B5 accepts B1 as root and sends (B1,1,B5) towards B3
6. B3 accepts B1 as root and it notes that both B2 and B5 are closer to the root than it is.
7. B3 stops forwarding messages on both its interfaces
Restructured Network using SDN

1. Open interface to packet forwarding

2. At least one Network OS probably many. Open- and closed-source

3. Well-defined open API
Alice’s code:
- Simple learning switch
- Per Flow switching
- Network access control/firewall
- Static “VLANs”
- Her own new routing protocol: unicast, multicast, multipath
- Home network manager
- Packet processor (in controller)
- IPvAlice
IP Internet

- Concatenation of Networks

- Protocol Stack
Service Model

- Connectionless (datagram-based)
- Best-effort delivery (unreliable service)
  - packets are lost
  - packets are delivered out of order
  - duplicate copies of a packet are delivered
  - packets can be delayed for a long time
- Datagram format

```
<table>
<thead>
<tr>
<th>0</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>19</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>HLen</td>
<td>TOS</td>
<td>Length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ident</td>
<td>Flags</td>
<td>Offset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTL</td>
<td>Protocol</td>
<td>Checksum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SourceAddr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DestinationAddr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options (variable)</td>
<td>Pad (variable)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Global Addresses

• Properties
  – globally unique
  – hierarchical: network + host

• Dot Notation
  – 10.3.2.4
  – 128.96.33.81
  – 192.12.69.77
Routing Problem

- Network as a Graph

Problem: Find lowest cost path between two nodes

- Factors
  - static: topology
  - dynamic: load
Distributed Bellman-Ford

Start Conditions:
Each router starts with a vector of (zero) distances to all directly attached networks

Send step:
Each router advertises its current vector to all neighboring routers.

Receive step:
Upon receiving vectors from each of its neighbors, router computes its own distance to each neighbor. Then, for every network X, router finds that neighbor who is closer to X than to any other neighbor. Router updates its cost to X. After doing this for all X, router goes to send step.
Link State Algorithm

Flooding:
1) Periodically distribute link-state advertisement (LSA) to neighbors
   - LSA contains delays to each neighbor
2) Install received LSA in LS database
3) Re-distribute LSA to all neighbors

Path Computation
1) Use Dijkstra’s shortest path algorithm to compute distances to all destinations
2) Install <destination, nexthop> pair in forwarding table
Forward Search Algorithm (Dijkstra)

1. Initialize **Confirmed** list with self entry 0
2. For node just added to **confirmed** list, call it “next” and select is LSP
3. Calculate distance to each neighbor of next
4. If neighbor is not on either confirmed or tentative list, add (neighbor, cost, nexthop) to tentative list
5. If neighbor is current on tentative list and cost is less than currently cost, then replace current entry with (neighbor, cost, nexthop)
6. If tentative list is empty stop, otherwise pick entry from tentative list with lowest cost and move to confirmed list. Return to step 2
IP Address

“class-full” addressing:

class

A
0 network host 1.0.0.0 to 127.255.255.255
 B 10 network host 128.0.0.0 to 191.255.255.255
 C 110 network host 192.0.0.0 to 223.255.255.255
 D 1110 multicast address 224.0.0.0 to 239.255.255.255

32 bits
IP addressing (Summary)

• **Classful addressing:**
  – inefficient use of address space, address space exhaustion
    • e.g., class B net allocated enough addresses for 65K hosts, even if only 2K hosts in that network

• **CIDR: Classless InterDomain Routing**
  – network portion of address of arbitrary length
  – address format: `a.b.c.d/x`, where `x` is # bits in network portion of address

```
11001000  00010111 0001000  0  00000000
```

```
network part
11001000 00010111 00010000
```

```
host part
00000000
```

```
200.23.16.0/23
```
Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1

- **Organization 0**
  - 200.23.16.0/23

- **Organization 2**
  - 200.23.20.0/23

- **Organization 7**
  - 200.23.30.0/23

- **Organization 1**
  - 200.23.18.0/23

Fly-By-Night-ISP

- "Send me anything with addresses beginning 200.23.16.0/20"

ISPs-R-Us

- "Send me anything with addresses beginning 199.31.0.0/16 or 200.23.18.0/23"

Internet
Routing in the Internet

- The Global Internet consists of Autonomous Systems (AS) interconnected with each other:
  - **Stub AS**: small corporation: one connection to other AS’s
  - **Multihomed AS**: large corporation (no transit): multiple connections to other AS’s
  - **Transit AS**: provider, hooking many AS’s together

- Two-level routing:
  - **Intra-AS**: administrator responsible for choice of routing algorithm within network
  - **Inter-AS**: unique standard for inter-AS routing: BGP
Internet AS Hierarchy

Inter-AS border (exterior gateway) routers

Intra-AS (interior gateway) routers
BGP Operations (Simplified)

Establish session on TCP port 179

Exchange all active routes

Exchange incremental updates

While connection is ALIVE exchange route UPDATE messages
Route Selection Summary

- Highest Local Preference
- Shortest AS_PATH
- Lowest MED
- i-BGP < e-BGP
- Lowest IGP cost to BGP egress
- Lowest router ID

Enforce relationships
Traffic engineering
Throw up hands and break ties
Internet Multicast Service Model

multicast group concept:
- Each group has its own IP multicast address
- A host can join or leave freely
- Routers forward multicast datagrams (with destination address of the group’s multicast address) to hosts that have “joined” that multicast group
IGMP: Internet Group Management Protocol

- **host**: sends IGMP report when application joins mcast group
  - IP_ADD_MEMBERSHIP socket option
  - host need not explicitly “unjoin” group when leaving
- **router**: sends IGMP query at regular intervals
  - host belonging to a mcast group must reply to query
Approaches for building mcast trees

Approaches:

- **source-based tree**: one tree per source
  - shortest path trees
  - reverse path forwarding
- **group-shared tree**: group uses one tree
  - minimal spanning (Steiner)
  - center-based trees

...we first look at basic approaches, then specific protocols adopting these approaches
# Network Protocols Summary

<table>
<thead>
<tr>
<th>Network</th>
<th>Architecture</th>
<th>Algorithm</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.3 Ethernet</td>
<td>Shared link</td>
<td>CSMA/CD</td>
<td>Ethernet packet</td>
<td>6B S/D addresses</td>
</tr>
<tr>
<td>802.11 WiFi</td>
<td>Shared radio link</td>
<td>CSMA/CA</td>
<td>Ethernet+ packet</td>
<td>S/D and intermediate</td>
</tr>
<tr>
<td>SW Ethernet</td>
<td>Campus net</td>
<td>Self-learning bridge, Spanning Tree</td>
<td>Ethernet packet, control msg</td>
<td>6B S/D, control, etc.</td>
</tr>
<tr>
<td>Basic IP</td>
<td>Wide area network</td>
<td>Bellman Ford/DV, Djikstra</td>
<td>IPv4 packet ICMP</td>
<td>Class A,B,C addr TTL etc.</td>
</tr>
<tr>
<td>Advanced IP</td>
<td>Global Network</td>
<td>DV with full path + local policy</td>
<td>BGP packets: update etc.</td>
<td>CIDR S/D addresses; AS #, path vectors, ...</td>
</tr>
<tr>
<td>IP Multicast</td>
<td>Wide area net</td>
<td>DV or Djikstra RPF;</td>
<td>Mcast addr; IGMP</td>
<td>Join, prune, etc.</td>
</tr>
</tbody>
</table>
**QoS: Network Congestion**

- All networks have saturating throughput
  - Reduction in performance beyond max capacity
  - Need to keep input load below $G_0$
  - Also must avoid unstable equilibrium point in overload region

![Graph showing QoS and network congestion]

- **Capacity Limit**
- **Unstable network load**
- **Stable network load lines with congestion control**
- **Normal operating Point ($G_0$)**
- **Traffic margin**
- **Overload region**
- **Congestion control policies**
QoS: FQ illustration

Variation: Weighted Fair Queuing (WFQ)
Random Early Detection (RED)

• Motivation:
  – High bw-delay flows have large queues to accommodate transient congestion
  – TCP detects congestion from loss - after queues have built up and increase delay

• Aim:
  – Keep throughput high and delay low
  – Accommodate bursts
**RED Operation**

- If `AvgLen <= MinThreshold`
  - Queue the packet
- If `MinThreshold < AvgLen < MaxThreshold`
  - Calculate probability `P`
  - Drop the arriving packets with probability `P`
- If `AvgLen >= MaxThreshold`
  - Drop the arriving packet

\[
\text{TempP} = \text{MaxP} \times \frac{(\text{AvgLen} - \text{MinThreshold})}{(\text{MaxThreshold} - \text{MinThreshold})}
\]

\[
P = \frac{\text{TempP}}{1 - \text{count} \times \text{TempP}}
\]

- Count: # of newly arriving packets that have been queued (not dropped) while `AvgLen` has been between the two thresholds
  - Count increases, `P` increases
  - Make drop more evenly distributed (Avoid bias against bursty traffic)
Flowspecs

• Tspec: describes the flow’s traffic characteristics
• Rspec: describes the service requested from the network
Admission control

• When new flow request arrives, look at Rspec and Tspec and decide whether to admit or reject
  – Can it provide the desired service requested by the flow, given the currently available resources without causing any previously admitted flow to receive worse service that agreed?

• Not policing
Reservation protocol: RSVP

Upper layer protocols and applications

IP service interface

IP (ICMP, IGMP, RSVP)

Link layer service interface

Link layer modules
PATH and RESV messages

Sender 1

Sender 2

PATH

RESV (merged)

receiver 1

receiver 2
IP DiffServ

- IP packets carry 6-bit service code points (DSCP)
  - Potentially support 64-different classes of services
  - In implementation, the number of DSCPs used in a network is much smaller, e.g. simple two-class network

- ToS field => now used for DiffServ and ECN
  - Bits 0-5: Differentiated Services Code Point (DSCP)
  - Bit 6: ECN-capable
  - Bit 7: ECN
Premium traffic flow

Company A

- Packets in premium flows have bit set
- Premium packet flow restricted to R bytes/sec

ISP

- Unmarked packet flow

Host

- first hop router
- internal router
- border router

Border router
Red with In or Out (RIO)

- Similar to RED, but with two separate probability curves
- Has two classes, “In” and “Out” (of profile)
- “Out” class has lower Minthresh, so packets are dropped from this class first
- As avg queue length increases, “in” packets are dropped
**IP Protocol Stack: Key Abstractions**

- **Application layer:**
  - Applications

- **Transport layer:**
  - Reliable streams
  - Messages

- **Network layer:**
  - Best-effort *global* packet delivery

- **Link layer:**
  - Best-effort *local* packet delivery

- **Transport layer:**
  - Provide applications with good abstractions
  - Without support or feedback from the network
  - Is the lowest layer in the network stack that is an end-to-end protocol
Transport Protocols

• Logical communication between processes
  • Sender divides a message into segments
  • Receiver reassembles segments into message

• Transport services
  • (De)multiplexing packets
  • Detecting corrupted data
  • Optionally: reliable delivery, flow control, ...
User Datagram Protocol (UDP)

- **Service:** Support for multiple processes on each host to communicate
  - Issue: IP only provides communication between hosts (IP addresses)
- **Solution**
  - Add port number and associate a process with a port number
  - *4-Tuple Unique Connection Identifier:* [SrcPort, SrcIPAddr, DestPort, DestIPAddr]

- **Lightweight communication between processes**
  - Send and receive messages
  - Avoid overhead of ordered, reliable delivery
    - No connection setup delay, in-kernel connection state

- **Used by popular apps**
  - Query/response for DNS
  - Real-time data in VoIP
### TCP Header

<table>
<thead>
<tr>
<th>Source port</th>
<th>Destination port</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sequence number</th>
<th>Acknowledgment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HdrLe</th>
<th>Flags</th>
<th>Advertised window</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Checksum</th>
<th>Urgent pointer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Options (variable)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Flags:** SYN, FIN, RST, PSH, URG, ACK
Host A

Sequence number = 1st byte

Host B

Sequence Number

ISN (initial sequence number)

TCP Data

TCP Data
TCP Acknowledgments

Host A

ISN (initial sequence number)

Sequence number = 1\textsuperscript{st} byte

TCP Data

Host B

TCP Data

ACK sequence number = next expected byte
Sliding Window

• Allow a larger amount of data “in flight”
  – Allow sender to get ahead of the receiver
    though not too far ahead.

Sending process

Last byte written

TCP

Last byte ACKed

Last byte sent

Receiving process

TCP

Last byte read

Next byte expected

Last byte received
Reasons for Retransmission

- Packet lost
- ACK lost
- DUPLICATE PACKET
- Early timeout
- DUPLICATE PACKETS
• When the *network* cannot support the sender’s rate
  – Queues at the network elements overflow
TCP “Slow Start”: To start quickly!

- Maintain another variable slow start threshold (ssthresh)
  - Last known stable rate
  - If (cwnd > ssthresh)
    - State = congestion avoidance
  - Else
    - State = slow start

- In Slow start
  - Increase the congestion window exponentially every RTT

Key: How is ssthresh calculated?

Every ACK reception
- $w = w + 1$
- $w = \text{cwnd in segments}$

Every ACK reception
- $\text{cwnd} = \text{cwnd} + \text{MSS}$
- $\text{cwnd in bytes}$
TCP Slow Start Example
Mobile Host on the Internet

• Solutions?
  – Change IP address → Connectivity is lost... what is host’s IP?
  – Keep IP address → Routing will fail... host not found.

How can S send to the moved mobile host?
Mobile IP Solution

Home Address: MH’s permanent IP address, network ID of this address identifies the mobile’s home network.

Foreign Network: a network, other than MA’s home network, that MH is currently attached to.

Corresponding Host: a host or router communicating with a mobile node.

Home Agent: a router attached to the MH’s home network maintains current location information for the MH and is responsible for forwarding packets destined for the MH when MH is away from home.

Home Network: the network identified with a mobile node.

Route Optimization

Foreign Agent (FA): a router in the foreign network that the MH is visiting provides routing services to the MH; may serve as default router for outgoing packet from MH.

Mobile Host: a host or router capable of changing its point of attachment to the Internet.
Ad-hoc Networks

- Each mobile device (node) can act as a router
- Links form and break based on mobility and environmental factors
- Connectivity (e.g., high probability of instantaneous end-to-end paths existing) is assumed
Dynamic Source Routing (DSR)
From Shweta Jain’s Slides

- When node S wants to send a packet to node D, but does not know a route to D, node S initiates a route discovery.
- Source node S *floods* the network with route request (RREQ) packets (also called query packets).
- Each node *appends its own address* in the packet header when forwarding RREQ.
Ad Hoc On-Demand Distance Vector Routing (AODV)

From Shweta Jain’s Slides

• AODV retains the desirable feature of DSR that routes are maintained only between nodes which need to communicate.

• AODV attempts to improve on DSR by maintaining **routing tables** at the nodes, so that data packets do not have to contain routes.

• No caches are used.
  – Only one route per destination in the routing table.
  – Only maintain the freshest route, if multiple possibilities.
OLSR: Optimized Link-State Routing

- Only **multipoint relays** (MPR) participate in flooding.
- Multipoint relays of node X are its neighbors such that each two-hop neighbor of X is a one-hop neighbor of at least one multipoint relay of X.
  - Each node transmits its neighbor list in hello messages, so that all nodes know their 2-hop neighbors, in order to choose the multipoint relays.
  - Select as few multipoint relays as possible.
- Only multipoint relays are used for flooding.
Disruption Tolerant Networks

• DTN Characteristics
  – Intermittent connectivity
  – Partitioning
  – No guarantee of end-to-end paths

• Goals:
  – High message delivery ratio
  – Acceptable delay

• Routing Approach
  – Store-carry-and-forward routing
  – Replication
Routing Categories

Replication Scale

- Hard limit on number of replicates per message
- No limit on number of replicas

- Spray and Wait
- Spray and Focus
- Encounter-based Routing
- MaxProp
- Prophet
- RAPID