Chapter 5
Link Layer and LANs

A note on the use of these ppt slides:
We’re making these slides freely available to all (faculty, students, readers). They’re in PowerPoint form so you can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a lot of work on our part. In return for use, we only ask the following:

❖ If you use these slides (e.g., in a class) in substantially unaltered form, that you mention their source (after all, we’d like people to use our book!)
❖ If you post any slides in substantially unaltered form on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2010
J.F Kurose and K.W. Ross, All Rights Reserved

Computer Networking:
A Top Down Approach
5th edition.
Jim Kurose, Keith Ross
Addison-Wesley, April 2009.
Synthesis: a day in the life of a web request

- journey down protocol stack complete!
  - application, transport, network, link

- putting-it-all-together: synthesis!
  - **goal**: identify, review, understand protocols (at all layers) involved in seemingly simple scenario: requesting www page
  - **scenario**: student attaches laptop to campus network, requests/receives www.google.com
A day in the life: scenario

- Comcast network: 68.80.0.0/13
- Google’s network: 64.233.160.0/19
- School network: 68.80.2.0/24
- DNS server

Browser accessing a web page from Google's network, which is connected through a web server (64.233.169.105). The school network is also depicted with a web server (64.233.169.105) and a browser accessing a web page.
A day in the life… connecting to the Internet

- connecting laptop needs to get its own IP address, addr of first-hop router, addr of DNS server: use DHCP
- DHCP request \textit{encapsulated} in UDP, encapsulated in IP, encapsulated in 802.1 Ethernet
- Ethernet frame \textit{broadcast} (dest: FFFFFFFF) on LAN, received at router running DHCP server
- Ethernet \textit{demuxed} to IP demuxed, UDP demuxed to DHCP
A day in the life... connecting to the Internet

- DHCP server formulates *DHCP ACK* containing client’s IP address, IP address of first-hop router for client, name & IP address of DNS server
- encapsulation at DHCP server, frame forwarded (*switch learning*) through LAN, demultiplexing at client
- DHCP client receives DHCP ACK reply

*Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router*
A day in the life... ARP (before DNS, before HTTP)

- before sending **HTTP** request, need IP address of www.google.com: **DNS**
- DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Eth. In order to send frame to router, need MAC address of router interface: **ARP**
- **ARP query** broadcast, received by router, which replies with **ARP reply** giving MAC address of router interface
- client now knows MAC address of first hop router, so can now send frame containing DNS query
A day in the life... using DNS

- IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router

- IP datagram forwarded from campus network into comcast network, routed (tables created by RIP, OSPF, IS-IS and/or BGP routing protocols) to DNS server

- demuxed to DNS server

- DNS server replies to client with IP address of www.google.com
A day in the life... TCP connection carrying HTTP

- to send HTTP request, client first opens **TCP socket** to web server
- TCP **SYN segment** (step 1 in 3-way handshake) **inter-domain routed** to web server
- web server responds with **TCP SYNACK** (step 2 in 3-way handshake)
- TCP **connection established!**
A day in the life... HTTP request/reply

- web page finally (!!!)
displayed

- HTTP request sent into TCP socket
- IP datagram containing HTTP request routed to www.google.com
- web server responds with HTTP reply (containing web page)
- IP datagram containing HTTP reply routed back to client

web server
64.233.169.105
Chapter 5: Summary

- principles behind data link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing

- instantiation and implementation of various link layer technologies
  - Ethernet
  - switched LANS, VLANs
  - PPP
  - virtualized networks as a link layer: MPLS

- synthesis: a day in the life of a web request
Chapter 5: let’s take a breath

- journey down protocol stack complete (except PHY)
- solid understanding of networking principles, practice
- ..... could stop here .... but lots of interesting topics!
  - wireless
  - multimedia
  - security
  - network management
Queueing Theory Primer

1. What is the Poisson process

2. What is a single server queue?

3. Little’s formula & its applications

4. Poisson Queues
   - And a comparison of packet/circuit switching
Poisson process

- Allows to answer question like
  - “If I receive ten message per minute, what is the probability that I receive 3 or more in the next second?”
  - Assumes that events happen independently at any particular time, with a constant “rate”.
    - Does not capture dependence or synchronization

- How to construct a Poisson process?
  - Iteratively: next event after an exp. Interval
  - Then start a new exp. interval
Poisson Process (Reminder)

The process is then called the \( \{E_n\}_{n=0}^{\infty} \) and follows exponential distribution

\[
\Pr [E_n > x] = e^{-\lambda x} \quad \mathbb{E} [f(E_i)] = \int_0^\infty \lambda e^{-\lambda x} f(x)dx
\]

- The process is then called the \textbf{Poisson Process}(\( \lambda \))
Queueing Theory Primer

1. What is the Poisson process

2. What is a single server queue?

3. Little’s formula & its applications

4. Poisson Queues
   - And a comparison of packet/circuit switching
Single server queue

- Allows one to model
  - Post-office, assembly lines, packets in a router
  - Answer question like “what is the average delay seen by a customer?”
- Parameters:
  - Input rate (Process of arrivals)
  - Service rate (customer served / unit of time)
  - The queue (Buffer size, Discipline used)
Queueing Theory Primer

1. What is the Poisson process
2. What is a single server queue?
3. Little's formula & its applications
4. Poisson Queues
   - And a comparison of packet/circuit switching
Little’s formula

- Let us introduce notation
  - $\lambda$: is the arrival rate in the queue

- Let us consider two performance metric
  - $N$: average # of customers in {queue, server}
  - $D$: average time spent in the system

- Then we have $N = \lambda \times D$
  - Seems intuitive, and is true VERY generally
  - General arrival process, service rate
  - For any queue (buffer, discipline) and part of it
Applications of Little’s formula

- In an network interface card and links:
  - Transmission line between two nodes
    - Delay $D=D_{\text{prop}}$, $N=\lambda \cdot D=\lambda \cdot D_{\text{prop}}$
  - Transmitter (~ server):
    - Delay $D=D_{\text{trans}}$, $N=\lambda \cdot D_{\text{trans}}=\text{utilization}$
  - In the queueing buffer:
    - Delay $D=D_{\text{queueing}}$, $N=\lambda \cdot D_{\text{queueing}}=\text{queue length}$
Queueing Theory Primer

1. What is the Poisson process

2. What is a single server queue?

3. Little’s formula & its applications

4. Poisson Queues
   - And a comparison of packet/circuit switching
What are the types of Queue

- The most important parameters are the arrival of process and the service time.

- Kendall’s notation: X/Y/k
  - k denotes the number of server (typically k=1)
  - X denotes the process of arrivals
    - M is for “memoryless”, which means it’s Poisson
  - Y denotes the service time distribution
    - M is for “memoryless”, = exponential service time
    - There are also G for general, D for deterministic
The $M/M/1$ queue

- It is the simplest to analyze
  - Because the system evolves with no memory
  - Even simpler than constant service time!
  - It has only two parameters
    - Arrival rate (how many customers arrived per sec.)
    - Service rate (how many customers can be served per sec.)
The M/M/1 Queue

Its evolution follows a Markov Chain

- Each state $0,1,2,...,i$ denotes “there are $i$ customers in \{queue, server\}”
- During a small time slot $\delta$, there is a probability
  - $\lambda \delta$ to receive a new customer: $i \rightarrow i+1$
  - $\mu \delta$ to complete servicing an existing customer: $i \rightarrow i-1$
The $M/M/1$ queue steady state

- Since this is a Markov chain,
  - its long term evolution follows a stationary measure $P(n)$ that is invariant by the transitions
  - Assume that: $\lambda P(n) = \mu P(n+1)$
    - Then $P(n)$ is a stationary measure
    - This also implies: $P(n)=\rho^n P(0)$ where $\rho=\lambda/\mu$
  - Hence, if $\rho<1$ we can show that $P(n)=\rho^n (1-\rho)$ is a stationary measure
The M/M/1 queue steady state

\[ P(\# \text{ customer} = n) = \rho^n (1-\rho) \text{ implies } \]

- Expected # of customers:
  \[ N = \sum_{n=0}^{\infty} nP(n) = \sum_{n=0}^{\infty} n\rho^n (1 - \rho) = \frac{\rho}{1-\rho} \]
  - Which is also: \[ N = \frac{\rho}{1-\rho} = \frac{\lambda/\mu}{1-\lambda/\mu} = \frac{\lambda}{\mu-\lambda} \]

- Expected delay \( D \) in the system?
  - Obtained by Little’s law: \[ D = \frac{1}{\mu - \lambda} \]
The $M/M/1$ queue steady state

\[ P(\# \text{ customer} = n) = \rho^n (1-\rho) \] implies

- Expected delay $W$ spent waiting in queue:
  - $W = D - \text{Expected time of service}$
  - So
    \[ W = D - \frac{1}{\mu} = \frac{1}{\mu - \lambda} - \frac{1}{\mu} \]

- Expected # customers in queue $Q$:
  - By Little’s law:
    \[ Q = \lambda W = \frac{\lambda}{\mu - \lambda} - \frac{\lambda}{\mu} \]
  - Note that we do not have $Q = N - 1$, because when the queue is empty $\#\text{cust} = \#\text{cust in queue}$
A real example

Let’s say

- 100 hungry students in an hour (=λ? =μ?)
- 30 seconds to give them a falafel
  - What is the service rate?

Characterizing this

- μ=120 customers in an hour
- D=1/(μ-λ)=1/20 hour=3 minutes
- N=number of people arriving in 3min = 5
- What is chance of being served immediately?
Circuit vs. Packet Switching

1

$\lambda/M$

$\lambda/M$

$\lambda/M$

Packets generated at random times

1 2 3

$\lambda/M$

1 2 3

$\lambda/M$

TDM, Time Division Multiplexing
Each user can send $\mu/N$ packets/sec and has packet arriving at rate $\lambda/N$ packets/sec

$$D = M/\mu + \frac{M(\lambda/\mu)}{\mu - \lambda}$$

Statistical Mutliplexer

$\lambda$

Buffer

$\mu$ packets/sec

$$D = 1/\mu + \frac{(\lambda/\mu)}{\mu - \lambda}$$

Data Link Layer 5-28