

# Computer Networking

Michaelmas/Lent Term

M/W/F 11:00-12:00

LT1 in Gates Building

Slide Set 1

Andrew W. Moore

[andrew.moore@cl.cam.ac.uk](mailto:andrew.moore@cl.cam.ac.uk)

2014-2015

## Topic 1 Foundation

- Administrivia
- Networks
- Channels
- Multiplexing
- Performance: loss, delay, throughput

2

## Course Administration

### Commonly Available Texts

- ❑ Computer Networking: A Top-Down Approach  
Kurose and Ross, 6<sup>th</sup> edition 2013, Addison-Wesley  
(5<sup>th</sup> edition is also commonly available)
- ❑ Computer Networks: A Systems Approach  
Peterson and Davie, 5<sup>th</sup> edition 2011, Morgan-Kaufman

### Other Selected Texts (non-representative)

- ❑ Internetworking with TCP/IP, vol. I + II  
Comer & Stevens, Prentice Hall
- ❑ UNIX Network Programming, Vol. I  
Stevens, Fenner & Rudoff, Prentice Hall



3

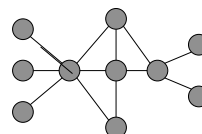
## Thanks

- Slides are a fusion of material from  
Ian Leslie, Richard Black, Jim Kurose, Keith Ross, Larry Peterson, Bruce Davie, Jen Rexford, Ion Stoica, Vern Paxson, Scott Shenker, Frank Kelly, Stefan Savage, Jon Crowcroft, Mark Handley, Sylvia Ratnasamy, and Adam Greenhalgh (and to those others I've forgotten, sorry.)
- Supervision material is drawn from  
Stephen Kell, Andy Rice, and the fantastic TA teams of 144 and 168
- Practical material will become available through this year  
But would be impossible without Georgina Kalogeridou, Nick McKeown, Bob Lantz, Te-Yuan Huang and Vimal Jeyakumar
- Finally thanks to the Part 1b students past and Andrew Rice for all the tremendous feedback.

4

## What is a network?

- A system of “links” that interconnect “nodes” in order to move “information” between nodes



- Yes, this is very vague

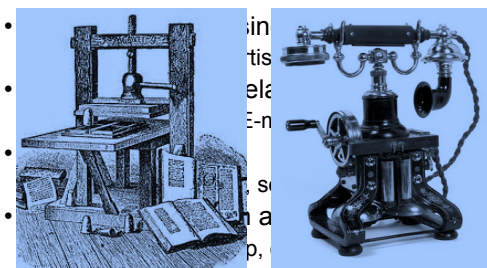
5

## There are *many* different types of networks

- Internet
  - Telephone network
  - Transportation networks
  - Cellular networks
  - Supervisory control and data acquisition networks
  - Optical networks
  - Sensor networks
- We will focus almost exclusively on the Internet

6

## The Internet is transforming everything



Took the dissemination of information to the next level

7

## The Internet is big business

- Many large and influential networking companies
  - Cisco, Broadcom, AT&T, Verizon, Akamai, Huawei, ...
  - \$120B+ industry (carrier and enterprise alone)
- Networking central to most technology companies
  - Google, Facebook, Intel, HP, Dell, VMware, ...

8

## Internet research has impact

- **The Internet started as a research experiment!**
- 4 of 10 most cited authors work in networking
- *Many* successful companies have emerged from networking research(ers)

9

## But why is the Internet *interesting*?

“What’s your formal model for the Internet?” -- *theorists*

“Aren’t you just writing software for networks” – *hackers*

“You don’t have performance benchmarks???” – *hardware folks*

“Isn’t it just another network?” – *old timers at AT&T*

“What’s with all these TLA protocols?” – *all*

“But the Internet seems to be working...” – *my mother*

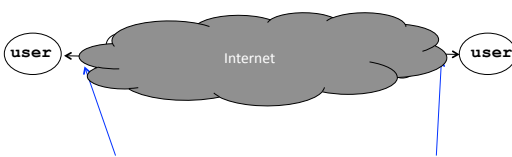
10

## A few defining characteristics of the Internet

11

## A federated system

- The Internet ties together different networks
  - >18,000 ISP networks



Tied together by IP -- the “Internet Protocol” : a single common interface between users and the network and between networks

12

## A federated system

- The Internet ties together different networks
  - >18,000 ISP networks
- A single, common interface is great for interoperability...
- ...but tricky for business
- Why does this matter?
  - ease of interoperability is the Internet’s most important goal
  - practical realities of incentives, economics and real-world trust drive topology, route selection and service evolution

13

## Tremendous scale

- 3 Billion users (43% of world population)
- 1+ Trillion unique URLs
- 194 Billion emails sent per day
- 1.75 Billion smartphones
- 1.23 Billion Facebook users
- 50 Billion WhatsApp messages per day
- 2 Billion YouTube videos watched per day
- Routers that switch 92Terabits/second
- Links that carry 400Gigabits/second

14

## Enormous diversity and dynamic range

- Communication latency: microseconds to seconds ( $10^6$ )
- Bandwidth: 1Kbits/second to 100 Gigabits/second ( $10^7$ )
- Packet loss: 0 – 90%
- Technology: optical, wireless, satellite, copper
- Endpoint devices: from sensors and cell phones to datacenters and supercomputers
- Applications: social networking, file transfer, skype, live TV, gaming, remote medicine, backup, IM
- Users: the governing, governed, operators, malicious, naïve, savvy, embarrassed, paranoid, addicted, cheap ...

15

## Constant Evolution

1970s:

- 56kilobits/second “backbone” links
- <100 computers, a handful of sites in the US (and one UK)
- Telnet and file transfer are the “killer” applications

Today

- 100+Gigabits/second backbone links
- 5B+ devices, all over the globe
- 20M Facebook apps installed per day

16

## Asynchronous Operation

- Fundamental constraint: **speed of light**
- Consider:
  - How many cycles does your 3GHz CPU in Cambridge execute before it can possibly get a response from a message it sends to a server in Palo Alto?
    - Cambridge to Palo Alto: 8,609 km
    - Traveling at 300,000 km/s: 28.70 milliseconds
    - Then back to Cambridge:  $2 \times 28.70 = 57.39$  milliseconds
    - $3,000,000,000 \text{ cycles/sec} \times 0.05739 = 172,179,999$  cycles!
- Thus, communication feedback is always *dated*

17

## Prone to Failure

- To send a message, **all** components along a path must function correctly
  - software, modem, wireless access point, firewall, links, network interface cards, switches,...
  - Including **human operators**
- Consider: 50 components, that work correctly 99% of time → 39.5% chance communication will fail
- Plus, recall
  - scale → lots of components
  - asynchrony → takes a long time to hear (bad) news
  - federation (**internet**) → hard to identify fault or assign blame

18

## An Engineered System

- Constrained by what technology is practical
  - Link bandwidths
  - Switch port counts
  - Bit error rates
  - **Cost**
  - ...

19



## Recap: The Internet is...

- A complex federation
- Of enormous scale
- Dynamic range
- Diversity
- Constantly evolving
- Asynchronous in operation
- Failure prone
- Constrained by what's practical to engineer
- Too complex for theoretical models
- "Working code" doesn't mean much
- Performance benchmarks are too narrow

20

## Performance – not just bits per second

Second order effects

- Image/Audio quality

Other metrics...

- Network efficiency (good-put *versus* throughput)

- User Experience? (World Wide Wait)



- Network connectivity expectation



- Others?

21

## Channels Concept

(This channel definition is very abstract)

- Peer entities communicate over channels
- Peer entities provide higher-layer peers with higher-layer channels

*A channel is that into which an entity puts symbols and which causes those symbols (or a reasonable approximation) to appear somewhere else at a later point in time.*



22

## Channel Characteristics

**Symbol type:** bits, packets, waveform

**Capacity:** bandwidth, data-rate, packet-rate

**Delay:** fixed or variable

**Fidelity:** signal-to-noise, bit error rate, packet error rate

**Cost:** per attachment, for use

**Reliability**

**Security:** privacy, unforgeability

**Order preserving:** always, almost, usually

**Connectivity:** point-to-point, to-many, many-to-many

Examples:

- Fibre Cable
- 1 Gb/s channel in a network
- Sequence of packets transmitted between hosts
- A telephone call (handset to handset)
- The audio channel in a room
- Conversation between two people

23

## Example Physical Channels

these example physical channels are also known as *Physical Media*

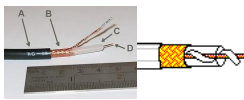
### Twisted Pair (TP)

- two insulated copper wires
- Category 3: traditional phone wires, 10 Mbps Ethernet
- Category 6: 1Gbps Ethernet
- Shielded (STP)
- Unshielded (UTP)



### Coaxial cable:

- two concentric copper conductors
- bidirectional
- baseband:
  - single channel on cable
  - legacy Ethernet
- broadband:
  - multiple channels on cable
  - HFC (Hybrid Fiber Coax)



### Fiber optic cable:

- high-speed operation
- point-to-point transmission
- (10' s-100' s Gps)
- low error rate
- immune to electromagnetic noise



24

## More Physical media: Radio

- Bidirectional and multiple access

- propagation environment effects:

- reflection
- obstruction by objects
- interference

### Radio link types:

- ❑ terrestrial microwave

- ❖ e.g. 45 Mbps channels

- ❑ LAN (e.g., Wifi)

- ❖ 11Mbps, 54 Mbps, 200 Mbps

- ❑ wide-area (e.g., cellular)

- ❖ 4G cellular: ~ 4 Mbps

- ❑ satellite

- ❖ Kbps to 45Mbps channel (or multiple smaller channels)

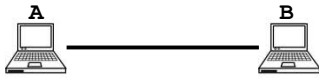
- ❖ 270 msec end-end delay

- ❖ geosynchronous versus low altitude



25

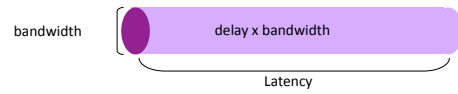
## Nodes and Links



Channels = Links  
Peer entities = Nodes

26

## Properties of Links (Channels)



- Bandwidth (capacity): “width” of the links
  - number of bits sent (or received) per unit time (bits/sec or bps)
- Latency (delay): “length” of the link
  - propagation time for data to travel along the link(seconds)
- Bandwidth-Delay Product (BDP): “volume” of the link
  - amount of data that can be “in flight” at any time
  - propagation delay × bits/time = total bits in link

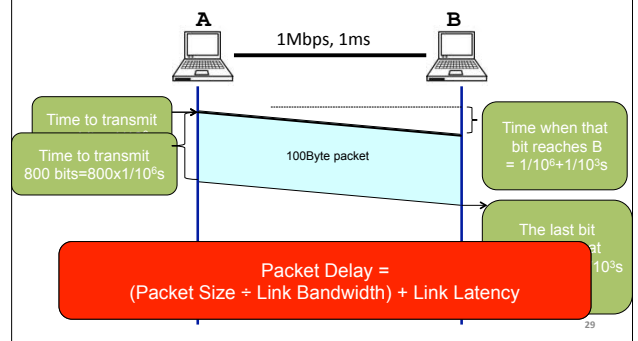
27

## Examples of Bandwidth-Delay

- Same city over a slow link:
  - BW~100Mbps
  - Latency~0.1msec
  - BDP ~ 10,000bits ~ 1.25KBytes
- Cross-country over fast link:
  - BW~10Gbps
  - Latency~10msec
  - BDP ~ 10<sup>8</sup>bits ~ 12.5GBytes

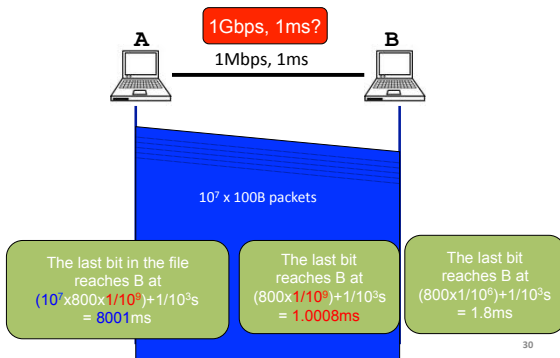
28

## Packet Delay Sending a 100B packet from A to B?



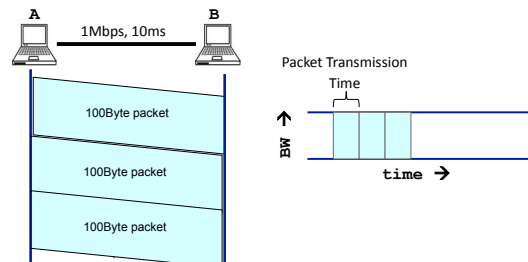
29

## 1GB file in 10<sup>7</sup> packets Sending a 100B packet from A to B?



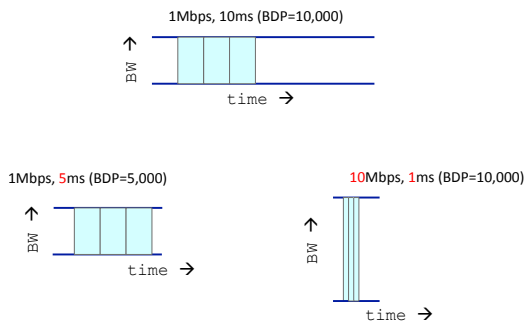
30

## Packet Delay: The “pipe” view Sending 100B packets from A to B?



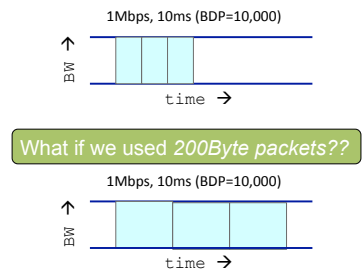
31

## Packet Delay: The "pipe" view Sending 100B packets from A to B?



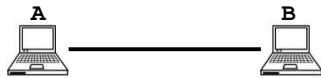
32

## Packet Delay: The "pipe" view Sending 100B packets from A to B?



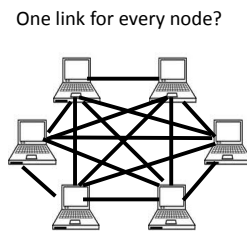
33

## Recall Nodes and Links



34

## What if we have more nodes?

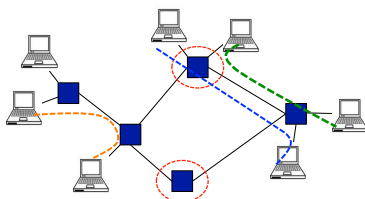


Need a scalable way to interconnect nodes

35

## Solution: A switched network

Nodes share network link resources



How is this sharing implemented?

36

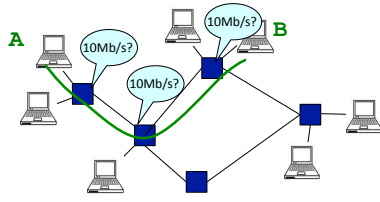
## Two forms of switched networks

- Circuit switching (used in the *POTS*: Plain Old Telephone system)
- Packet switching (used in the Internet)

37

## Circuit switching

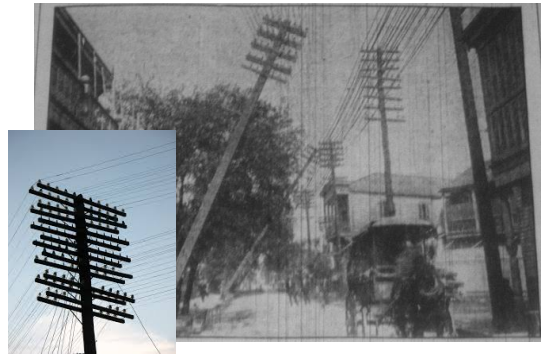
Idea: source **reserves** network capacity along a path



- (1) Node A sends a reservation request
- (2) Interior switches establish a connection -- i.e., "circuit"
- (3) A starts sending data
- (4) A sends a "teardown circuit" message

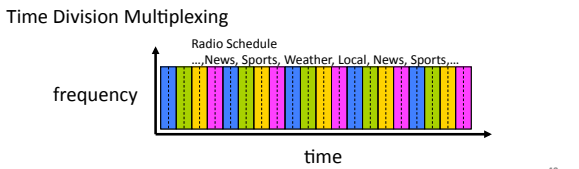
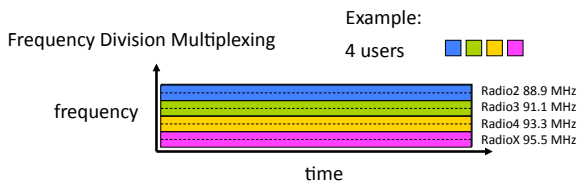
38

## Old Time Multiplexing



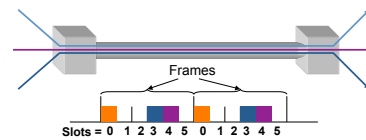
39

## Circuit Switching: FDM and TDM



40

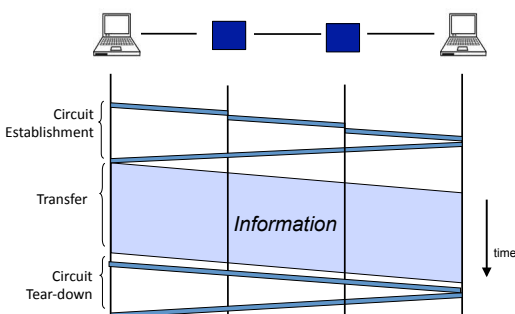
## Time-Division Multiplexing/Demultiplexing



- Time divided into frames; frames into slots
- Relative slot position inside a frame determines to which conversation data belongs
  - e.g., slot 0 belongs to orange conversation
- Slots are reserved (released) during circuit setup (teardown)
- If a conversation does not use its circuit **capacity is lost!**

41

## Timing in Circuit Switching



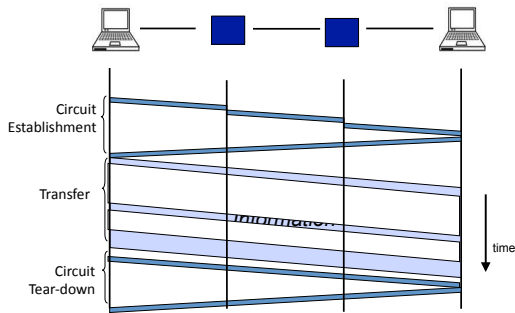
42

## Circuit switching: pros and cons

- Pros
  - guaranteed performance
  - fast transfer (once circuit is established)
- Cons

43

## Timing in Circuit Switching



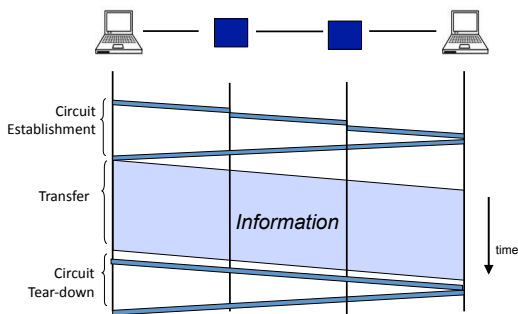
44

## Circuit switching: pros and cons

- Pros
  - guaranteed performance
  - fast transfer (once circuit is established)
- Cons
  - **wastes bandwidth if traffic is “bursty”**

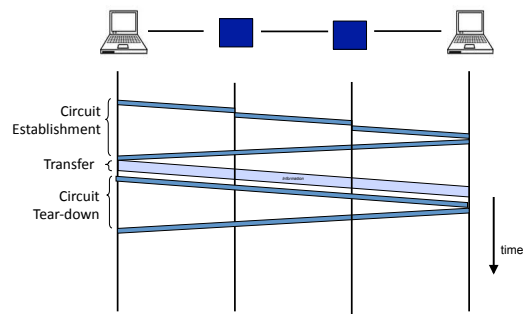
45

## Timing in Circuit Switching



46

## Timing in Circuit Switching



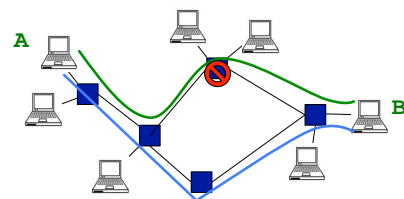
47

## Circuit switching: pros and cons

- Pros
  - guaranteed performance
  - fast transfers (once circuit is established)
- Cons
  - wastes bandwidth if traffic is “bursty”
  - **connection setup time is overhead**

48

## Circuit switching



Circuit switching doesn't "route around failure"

49

## Circuit switching: pros and cons

- Pros
  - guaranteed performance
  - fast transfers (once circuit is established)
- Cons
  - wastes bandwidth if traffic is “bursty”
  - connection setup time is overhead
  - **recovery from failure is slow**

50

## Numerical example

- How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  - All links are 1.536 Mbps
  - Each link uses TDM with 24 slots/sec
  - 500 msec to establish end-to-end circuit

Let's work it out!

51

## Two forms of switched networks

- Circuit switching (e.g., telephone network)
- Packet switching (e.g., Internet)

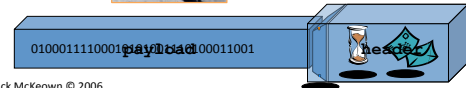
52

## Packet Switching

- Data is sent as chunks of formatted bits (**Packets**)
- Packets consist of a “**header**” and “**payload**”\*



1. Internet Address
2. Age (TTL)
3. Checksum to protect header



After Nick McKeown © 2006

53

## Packet Switching

- Data is sent as chunks of formatted bits (**Packets**)
- Packets consist of a “**header**” and “**payload**”\*
  - payload is the data being carried
  - header holds instructions to the network for how to handle packet (think of the header as an API)

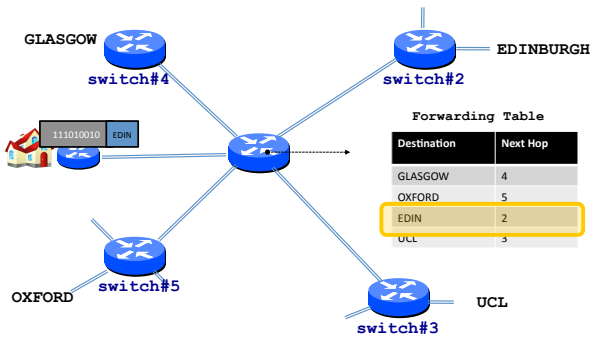
54

## Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a “header” and “payload”
- Switches “**forward**” packets based on their headers

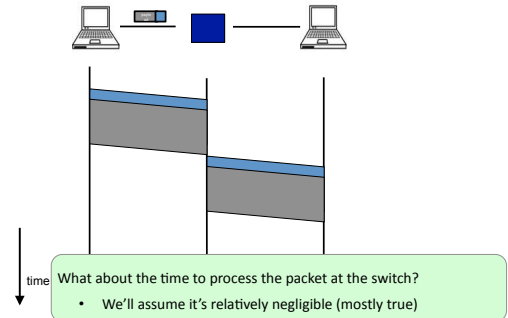
55

## Switches forward packets



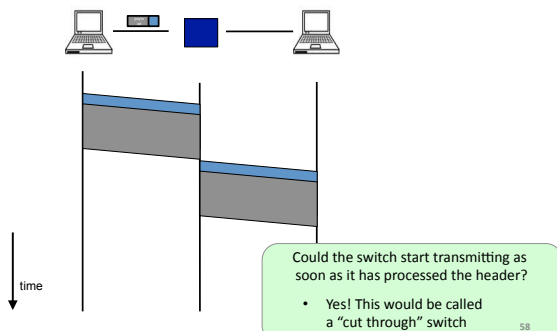
56

## Timing in Packet Switching



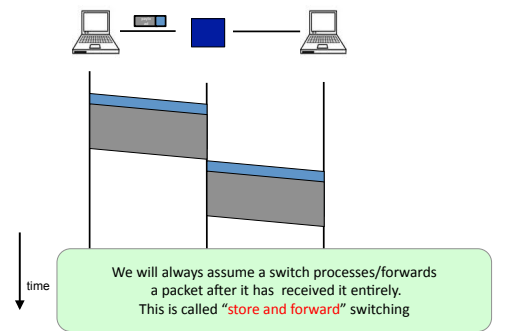
57

## Timing in Packet Switching



58

## Timing in Packet Switching



59

## Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a "header" and "payload"
- Switches "forward" packets based on their headers

60

## Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a "header" and "payload"
- Switches "forward" packets based on their headers
- Each packet travels independently
  - no notion of packets belonging to a "circuit"

61

## Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a “header” and “payload”
- Switches “forward” packets based on their headers
- Each packet travels independently
- No link resources are reserved in advance. Instead packet switching leverages **statistical multiplexing** (stat muxing)

62

## Multiplexing

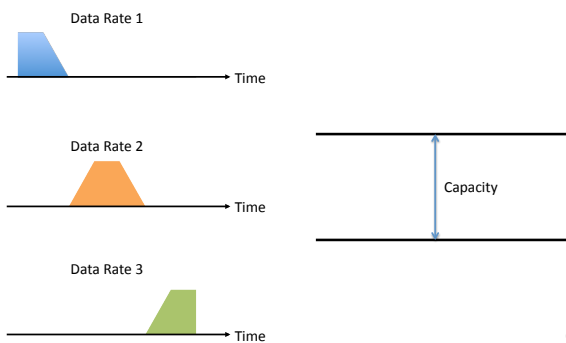


Sharing makes things efficient (cost less)

- One airplane/train for 100 people
- One telephone for many calls
- One lecture theatre for many classes
- One computer for many tasks
- One network for many computers
- One datacenter many applications

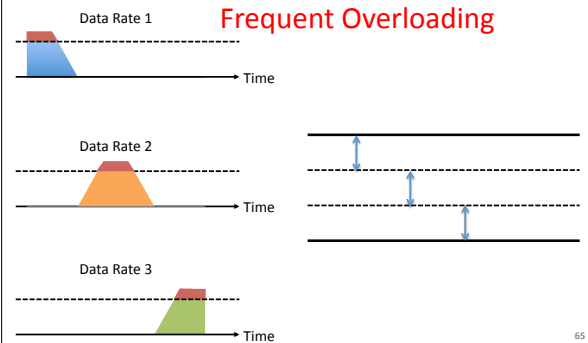
63

## Three Flows with Bursty Traffic



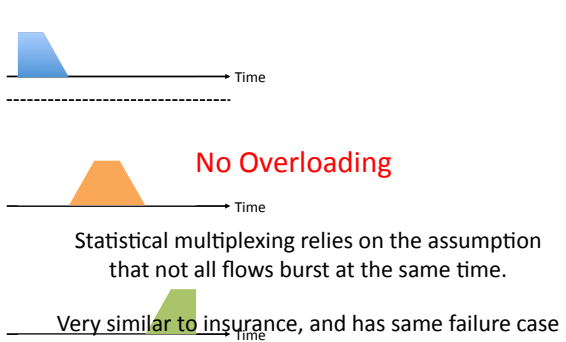
64

## When Each Flow Gets 1/3<sup>rd</sup> of Capacity



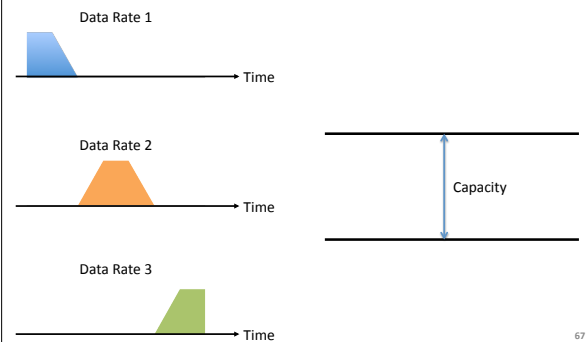
65

## When Flows Share Total Capacity



66

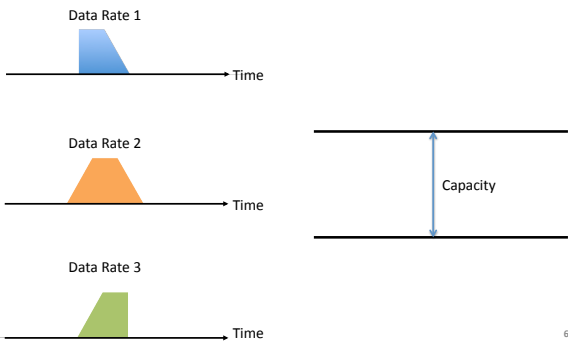
## Three Flows with Bursty Traffic



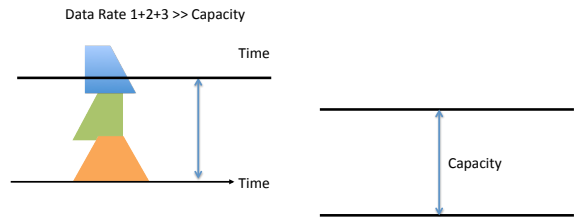
67



### Three Flows with Bursty Traffic

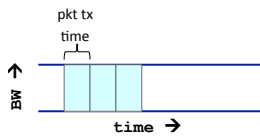


### Three Flows with Bursty Traffic

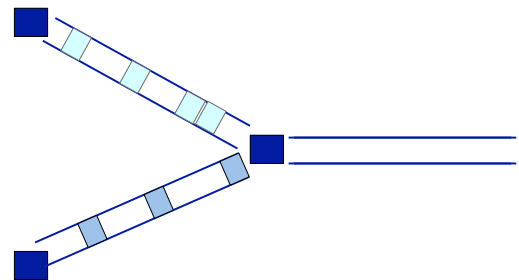


What do we do under overload?

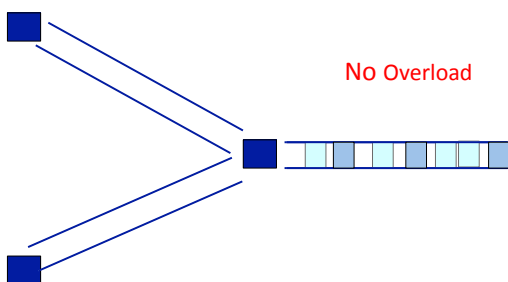
### Statistical multiplexing: pipe view



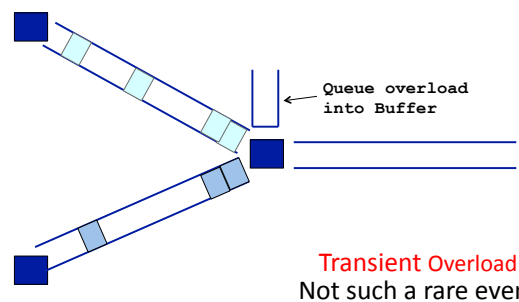
### Statistical multiplexing: pipe view



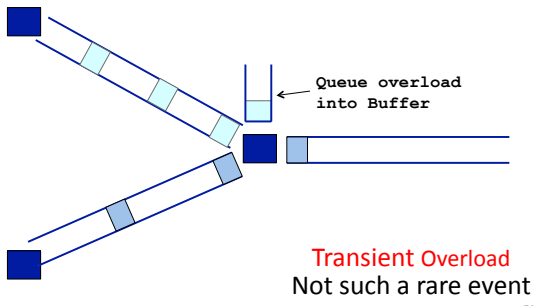
### Statistical multiplexing: pipe view



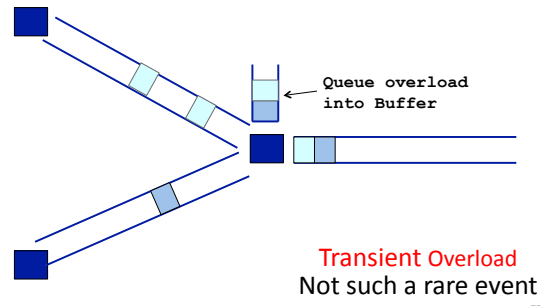
### Statistical multiplexing: pipe view



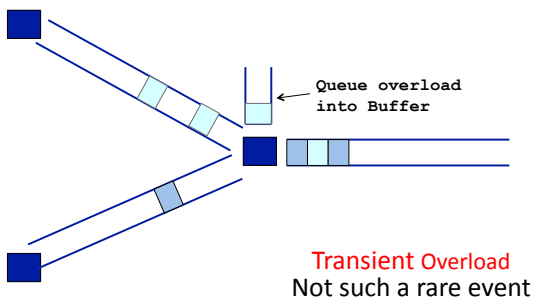
### Statistical multiplexing: pipe view



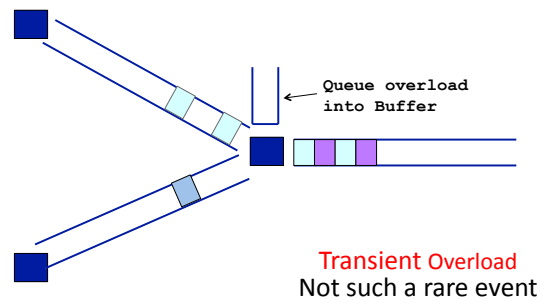
### Statistical multiplexing: pipe view



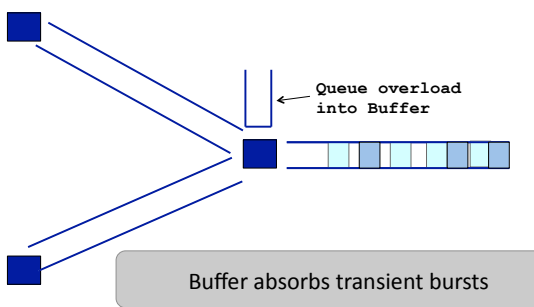
### Statistical multiplexing: pipe view



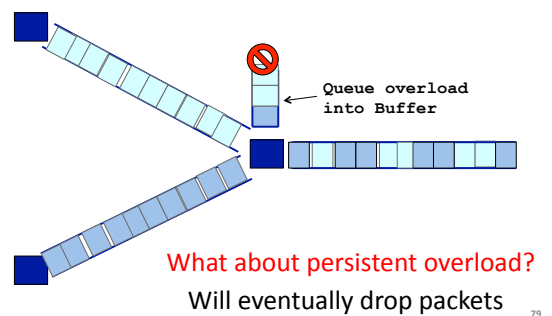
### Statistical multiplexing: pipe view



### Statistical multiplexing: pipe view



### Statistical multiplexing: pipe view



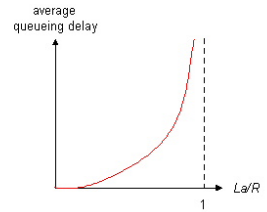
## Queues introduce queuing delays

- Recall,
    - packet delay = transmission delay + propagation delay (\*)
  - With queues (statistical muxing)
    - packet delay = transmission delay + propagation delay + queuing delay (\*)
  - Queuing delay caused by "packet interference"
  - Made worse at high load
    - less "idle time" to absorb bursts
    - think about traffic jams at rush hour or rail network failure
- (\* plus per-hop processing delay that we define as negligible)

80

## Queuing delay

- R=link bandwidth (bps)
- L=packet length (bits)
- a=average packet arrival rate



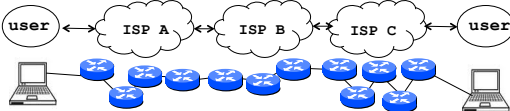
traffic intensity =  $La/R$

- $La/R \sim 0$ : average queuing delay small
- $La/R \rightarrow 1$ : delays become large
- $La/R > 1$ : more "work" arriving than can be serviced, average delay infinite – or data is lost (*dropped*).

81

## Recall the Internet *federation*

- The Internet ties together different networks
  - >18,000 ISP networks

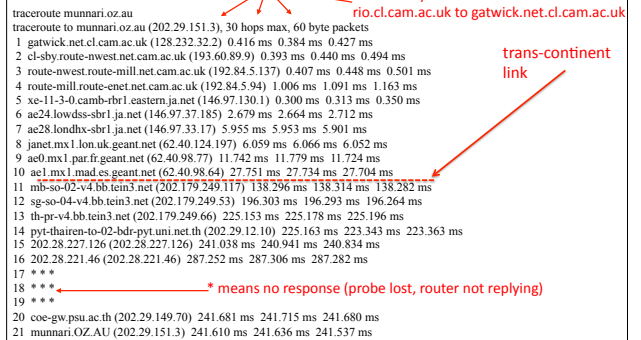


We can see (hints) of the nodes and links using traceroute...

82

## "Real" Internet delays and routes

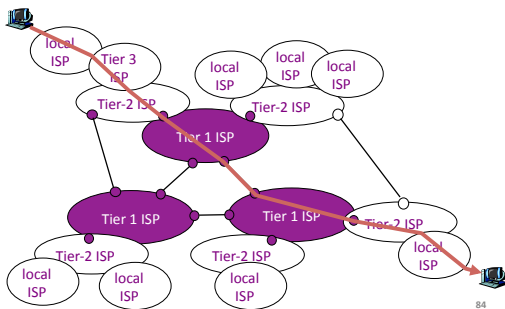
traceroute: rio.cl.cam.ac.uk to munnari.oz.au  
(tracepath on pwf is similar)



83

## Internet structure: network of networks

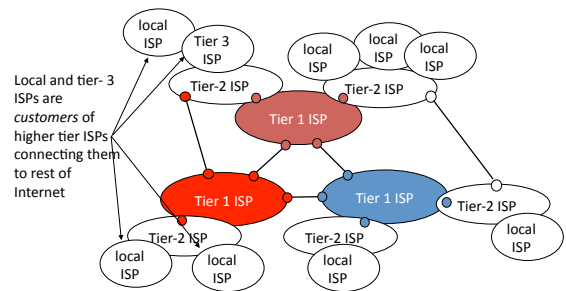
- a packet passes through many networks!



84

## Internet structure: network of networks

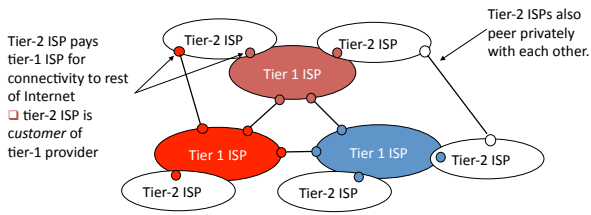
- "Tier-3" ISPs and local ISPs
  - last hop ("access") network (closest to end systems)



85

## Internet structure: network of networks

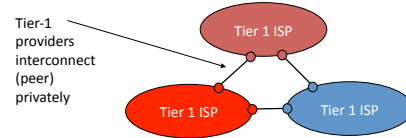
- “Tier-2” ISPs: smaller (often regional) ISPs
  - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs



86

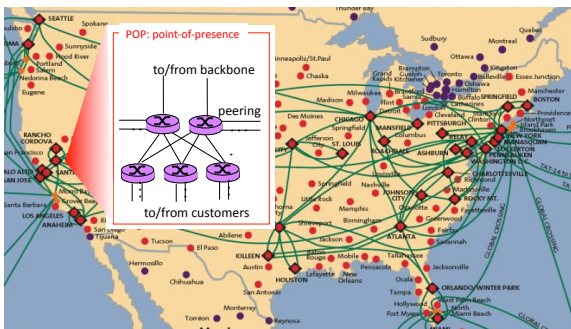
## Internet structure: network of networks

- roughly hierarchical
- at center: “tier-1” ISPs (e.g., Verizon, Sprint, AT&T, Cable and Wireless), national/international coverage
  - treat each other as equals



87

## Tier-1 ISP: e.g., Sprint



88

## Packet Switching

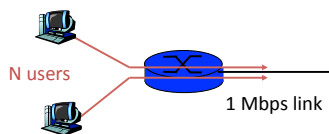
- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a “header” and “payload”
- Switches “forward” packets based on their headers
- Each packet travels independently
- No link resources are reserved in advance. Instead packet switching leverages **statistical multiplexing**
  - allows efficient use of resources
  - but introduces queues and queuing delays

89

## Packet switching versus circuit switching

*Packet switching may (does!) allow more users to use network*

- 1 Mb/s link
- each user:
  - 100 kb/s when “active”
  - active 10% of time
- **circuit-switching:**
  - 10 users
- **packet switching:**
  - with 35 users, probability > 10 active at same time is less than .0004



Q: how did we get value 0.0004?

90

## Packet switching versus circuit switching

Q: how did we get value 0.0004?

- 1 Mb/s link
- each user:
  - 100 kb/s when “active”
  - active 10% of time
- **circuit-switching:**
  - 10 users
- **packet switching:**
  - with 35 users, probability > 10 active at same time is less than .0004

HINT: Binomial Distribution

91

## Circuit switching: pros and cons

- Pros
  - guaranteed performance
  - fast transfers (once circuit is established)
- Cons
  - wastes bandwidth if traffic is “bursty”
  - connection setup adds delay
  - recovery from failure is slow

92

## Packet switching: pros and cons

- Cons
  - no guaranteed performance
  - header overhead per packet
  - queues and queuing delays
- Pros
  - efficient use of bandwidth (stat. muxing)
  - no overhead due to connection setup
  - resilient -- can ‘route around trouble’

93

## Summary

- A sense of how the basic ‘plumbing’ works
  - links and switches
  - packet delays= transmission + propagation + queuing + (negligible) per-switch processing
  - statistical multiplexing and queues
  - circuit vs. packet switching

94

## Topic 2 – Foundations and Architecture

- Abstraction
- Layering
- Layers and Communications
- Entities and Peers
- What is a protocol?
- Protocol Standardization
- The architects process
  - How to break system into modules
  - Where modules are implemented
  - Where is state stored
- Internet Philosophy and Tensions

2

## Abstraction Concept

A mechanism for breaking down a problem

*what not how*

- eg Specification *versus* implementation
- eg Modules in programs

Allows replacement of implementations without affecting system behavior

*Vertical versus Horizontal*

“Vertical” what happens in a box “How does it attach to the network?”

“Horizontal” the communications paths running through the system

**Hint:** paths are build on top of (“layered over”) other paths

3

## Computer System Modularity

Partition system into modules & abstractions:

- Well-defined interfaces give flexibility
  - **Hides** implementation - can be freely changed
  - Extend functionality of system by adding new modules
- E.g., libraries encapsulating set of functionality
- E.g., programming language + compiler abstracts away how the particular CPU works ...

4

## Computer System Modularity (cnt' d)

- Well-defined interfaces hide information
  - Isolate **assumptions**
  - Present high-level **abstractions**
- **But can impair performance!**
- Ease of implementation vs worse performance

5

## Network System Modularity

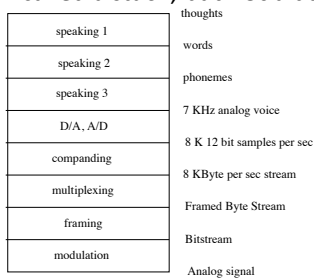
Like software modularity, but:

- Implementation is distributed across many machines (routers and hosts)
- Must decide:
  - How to break system into modules
    - **Layering**
  - Where modules are implemented
    - **End-to-End Principle**
  - Where state is stored
    - **Fate-sharing**

6

## Layering Concept

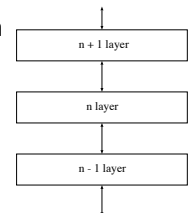
- A restricted form of abstraction: system functions are divided into layers, one built upon another
- Often called a *stack*; but **not** a data structure!



7

## Layers and Communications

- Interaction only between adjacent layers
- *layer n* uses services provided by *layer n-1*
- *layer n* provides service to *layer n+1*
- Bottom layer is physical media
- Top layer is application



8

## Entities and Peers

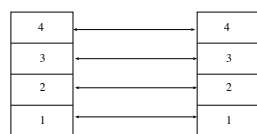
*Entity* – a *thing* (an independent existence)

Entities *interact* with the layers above and below

Entities *communicate* with *peer* entities

- same level but different place (eg different person, different box, different host)

Communications between peers is supported by entities at the lower layers



9

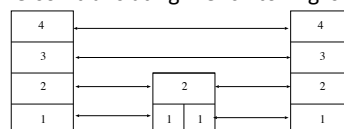
## Entities and Peers

Entities usually do something useful

- Encryption – Error correction – Reliable Delivery
- Nothing at all is also reasonable

Not all communications is end-to-end

- IP Router – Mobile Phone Cell Tower
- Person translating French to English



10

# Layering and Embedding

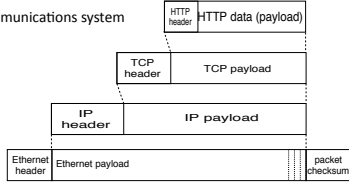
In Computer Networks we often see higher-layer information embedded within lower-layer information

- Such embedding can be considered a form of layering
- Higher layer information is generated by stripping off headers and trailers of the current layer
- eg an IP entity only looks at the IP headers

**BUT embedding is not the only form of layering**

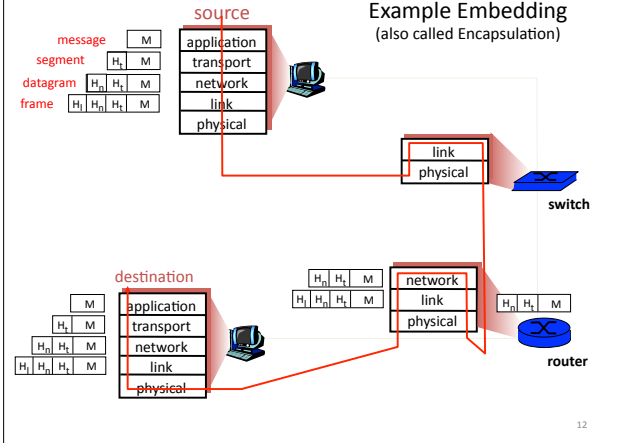
Layering is to help understand a communications system

**NOT** determine implementation strategy



11

## Example Embedding (also called Encapsulation)



12

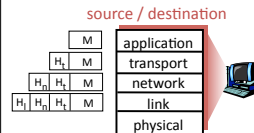
## Distributing Layers Across Network

- Layers are simple if only on a single machine
  - Just stack of modules interacting with those above/below
- But we need to implement layers across machines
  - Hosts
  - Routers (switches)
- What gets implemented where?

13

## What Gets Implemented on Host?

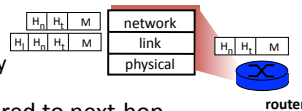
- Bits arrive on wire, must make it up to application
- Therefore, all layers must exist at the host



14

## What Gets Implemented on a Router?

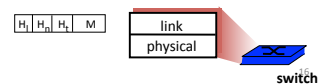
- Bits arrive on wire
  - Physical layer necessary
- Packets must be delivered to next-hop
  - Datalink layer necessary
- Routers participate in global delivery
  - Network layer necessary
- Routers don't support reliable delivery
  - Transport layer (and above) **not** supported



15

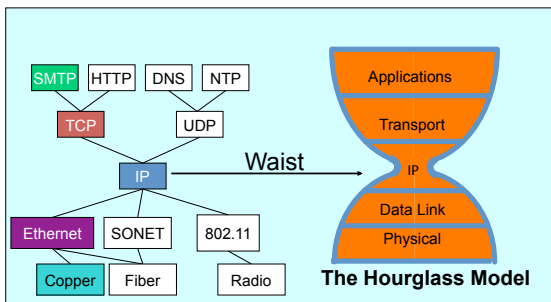
## What Gets Implemented on Switches?

- Switches do what routers do, except they don't participate in global delivery, just local delivery
- They only need to support Physical and Datalink
  - Don't need to support Network layer
- Won't focus on the router/switch distinction
  - When I say switch, I almost always mean router
  - Almost all boxes support network layer these days
  - Routers have switches but switches do not have routers



16

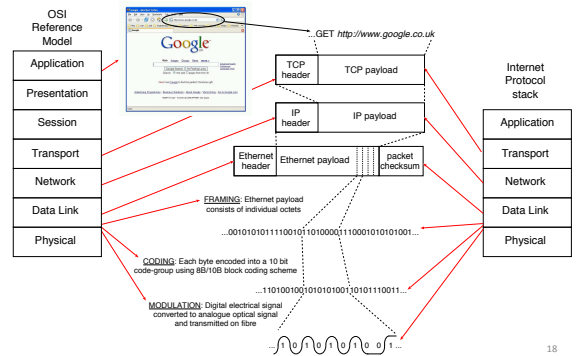
## The Internet Hourglass



There is just **one** network-layer protocol, **IP**.  
The “narrow waist” facilitates **interoperability**.

17

## Internet protocol stack versus OSI Reference Model



18

## ISO/OSI reference model

- **presentation**: allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- **session**: synchronization, checkpointing, recovery of data exchange
- Internet stack “missing” these layers!
  - these services, *if needed*, must be implemented in application
  - needed?

application
presentation
session
transport
network
link
physical

19

## What is a protocol?

### human protocols:

- “what’s the time?”
- “I have a question”
- introductions

... specific msgs sent

... specific actions taken when msgs received, or other events

### network protocols:

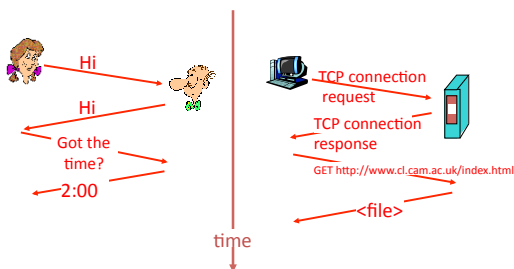
- machines rather than humans
- all communication activity in Internet governed by protocols

*protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt*

20

## What is a protocol?

a human protocol and a computer network protocol:



Q: Other human protocols?

21

## Protocol Standardization

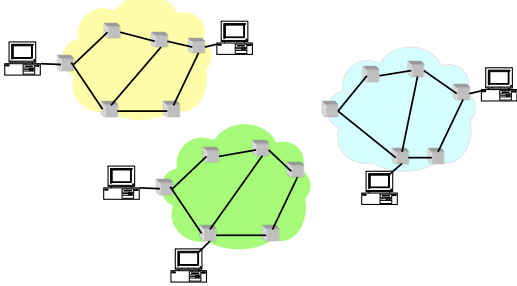
- All hosts must follow same protocol
  - Very small modifications can make a big difference
  - Or prevent it from working altogether
  - Cisco bug compatible!
- This is why we have standards
  - Can have multiple implementations of protocol
- Internet Engineering Task Force
  - Based on working groups that focus on specific issues
  - Produces “Request For Comments” (RFCs)
  - IETF Web site is <http://www.ietf.org>
  - RFCs archived at <http://www.rfc-editor.org>

22



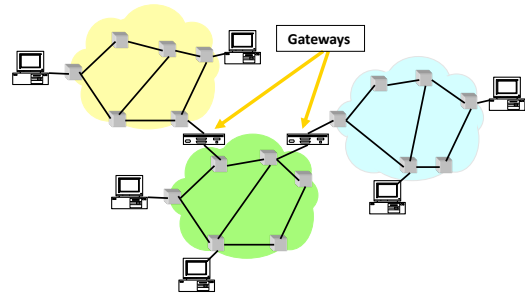
## So many Standards Problem

- Many different packet-switching networks
- Each with its own Protocol
- Only nodes on the same network could communicate



23

## INTERNet Solution



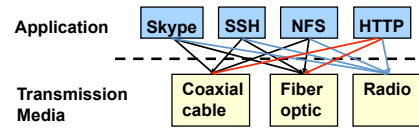
24

## Alternative to Standardization?

- Have one implementation used by everyone
- Open-source projects
  - Which has had more impact, Linux or POSIX?
- Or just sole-sourced implementation
  - Skype, many P2P implementations, etc.

25

## A Multitude of Apps Problem

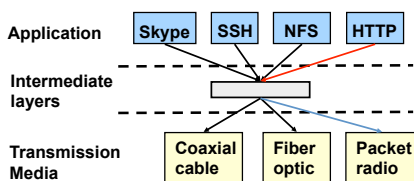


- Re-implement every application for every technology?
- No! But how does the Internet design avoid this?

26

## Solution: Intermediate Layers

- Introduce intermediate layers that provide [set of abstractions](#) for various network functionality and technologies
  - A new app/media implemented only once
  - Variation on "add another level of indirection"



27

## Remember that slide!

- The relationship between architectural principles and architectural decisions is crucial to understand

28

## Internet Design Goals (Clark '88)

- **Connect existing networks**
- Robust in face of failures
- Support multiple types of delivery services
- Accommodate a variety of networks
- Allow distributed management
- Easy host attachment
- Cost effective
- Allow resource accountability

29

## Real Goals

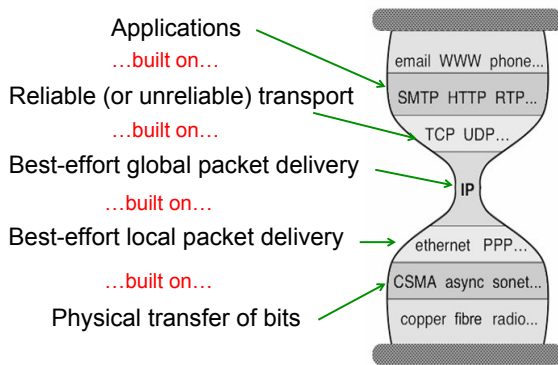
Internet Motto

*We reject kings, presidents, and voting. We believe in rough consensus and running code.* – David Clark

- **Build something that works!**
- Connect existing networks
- Robust in face of failures
- Support multiple types of delivery services
- Accommodate a variety of networks
- Allow distributed management
- Easy host attachment
- Cost effective
- ~~Allow resource accountability~~

30

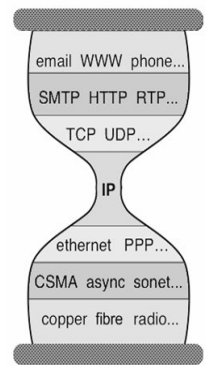
## In the context of the Internet



31

## Three Observations

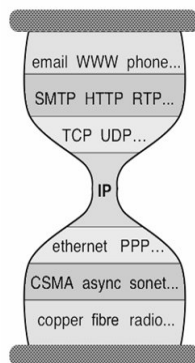
- Each layer:
  - Depends on layer below
  - Supports layer above
  - Independent of others
- Multiple versions in layer
  - Interfaces differ somewhat
  - Components pick which lower-level protocol to use
- But only one IP layer
  - Unifying protocol



32

## Layering Crucial to Internet's Success

- Reuse
- Hides underlying detail
- Innovation at each level can proceed in parallel
- Pursued by very different communities



33

What are some of the drawbacks of protocols and layering?

34

## Drawbacks of Layering

- Layer N may duplicate lower layer functionality
  - e.g., error recovery to retransmit lost data
- Information hiding may hurt performance
  - e.g., packet loss due to corruption vs. congestion
- Headers start to get really big
  - e.g., typical TCP+IP+Ethernet is 54 bytes
- Layer violations when the gains too great to resist
  - e.g., TCP-over-wireless
- Layer violations when network doesn't trust ends
  - e.g., firewalls

35

## Placing Network Functionality

- Hugely influential paper: "End-to-End Arguments in System Design" by Saltzer, Reed, and Clark ('84)
  - articulated as the "End-to-End Principle" (E2E)
- Endless debate over what it means
- Everyone cites it as supporting their position  
(regardless of the position!)

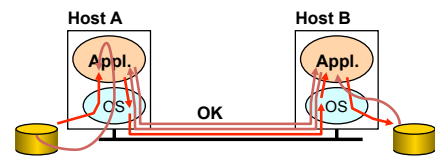
36

## Basic Observation

- Some application requirements can only be correctly implemented **end-to-end**
  - reliability, security, etc.
- Implementing these in the network is hard
  - every step along the way must be fail proof
- Hosts
  - **Can** satisfy the requirement without network's help
  - **Will/must** do so, since they can't rely on the network

37

## Example: Reliable File Transfer



- Solution 1: make each step reliable, and string them together to make reliable end-to-end process
- Solution 2: end-to-end **check** and retry

38

## Discussion

- Solution 1 is incomplete
  - What happens if any network element misbehaves?
  - Receiver has to do the check anyway!
- Solution 2 is complete
  - Full functionality can be entirely implemented at application layer with no need for reliability from lower layers
- Is there any need to implement reliability at lower layers?

39

## Summary of End-to-End Principle

- Implementing functionality (e.g., reliability) in the network
  - Doesn't reduce host implementation complexity
  - Does increase network complexity
  - Probably increases delay and overhead on all applications even if they don't need the functionality (e.g. VoIP)
- However, implementing in the network can improve performance in some cases
  - e.g., consider a very lossy link

40

## “Only-if-Sufficient” Interpretation

- Don't implement a function at the lower levels of the system unless it can be completely implemented at this level
- *Unless you can relieve the burden from hosts, don't bother*

41

## “Only-if-Necessary” Interpretation

- Don't implement *anything* in the network that can be implemented correctly by the hosts
- Make network layer absolutely minimal
  - This E2E interpretation trumps performance issues
  - Increases flexibility, since lower layers stay **simple**

42

## “Only-if-Useful” Interpretation

- If hosts can implement functionality correctly, implement it in a lower layer **only** as a performance enhancement
- But do so only if it **does not impose burden** on applications that do not require that functionality

43

## We have some tools:

- Abstraction
- Layering
- Layers and Communications
- Entities and Peers
- Protocol as motivation
- Examples of the architects process
- Internet Philosophy and Tensions

44

## Topic 3: The Data Link Layer

### Our goals:

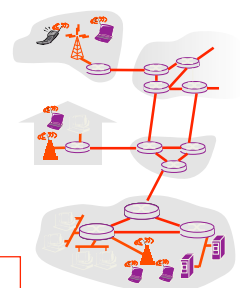
- understand principles behind data link layer services: (these are methods & mechanisms in your networking toolbox)
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - reliable data transfer, flow control:
- instantiation and implementation of various link layer technologies
  - Wired Ethernet (aka 802.3)
  - Wireless Ethernet (aka 802.11 WiFi)
- Algorithms
  - Binary Exponential Backoff
  - Spanning Tree

2

## Link Layer: Introduction

### Some terminology:

- hosts and routers are **nodes**
- communication channels that connect adjacent nodes along communication path are **links**
  - wired links
  - wireless links
  - LANs
- layer-2 packet is a **frame**, encapsulates datagram



**data-link layer** has responsibility of transferring datagram from one node to adjacent node over a link

3

## Link Layer (Channel) Services

- **framing, link access:**
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - “MAC” addresses used in frame headers to identify source, dest
    - different from IP address!
- **reliable delivery between adjacent nodes**
  - we see some of this again in the Transport Topic
  - seldom used on low bit-error link (fiber, some twisted pair)
  - wireless links: high error rates
    - Q: why both link-level and end-end reliability?

4

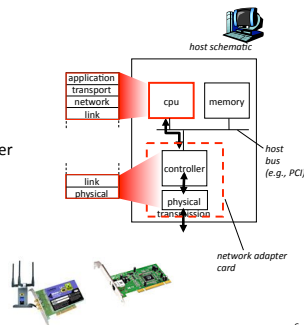
## Link Layer (Channel) Services - 2

- **flow control:**
  - pacing between adjacent sending and receiving nodes
- **error detection:**
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- **error correction:**
  - receiver identifies **and corrects** bit error(s) without resorting to retransmission
- **half-duplex and full-duplex**
  - with half duplex, nodes at both ends of link can transmit, but not at same time

5

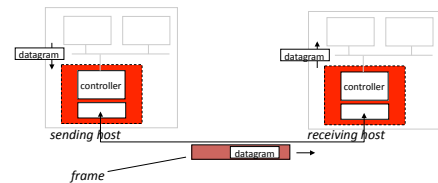
## Where is the link layer implemented?

- in each and every host
- link layer implemented in “adaptor” (aka **network interface card NIC**)
  - Ethernet card, PCMC card, 802.11 card
  - implements link, physical layer
- attaches into host’s system buses
- combination of hardware, software, firmware



6

## Adaptors Communicating

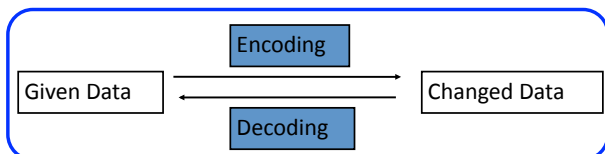


- **sending side:**
  - encapsulates datagram in frame
  - encodes data for the physical layer
  - adds error checking bits, provide reliability, flow control, etc.
- **receiving side**
  - decodes data from the physical layer
  - looks for errors, provide reliability, flow control, etc
  - extracts datagram, passes to upper layer at receiving side

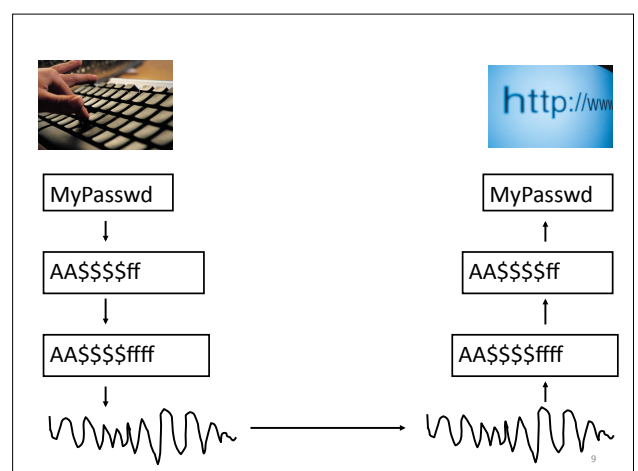
7

## Coding – a channel function

Change the representation of data.



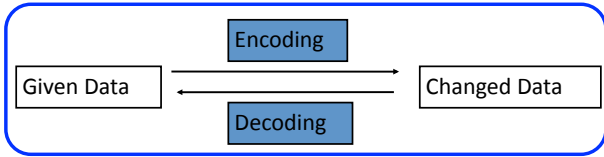
8



9

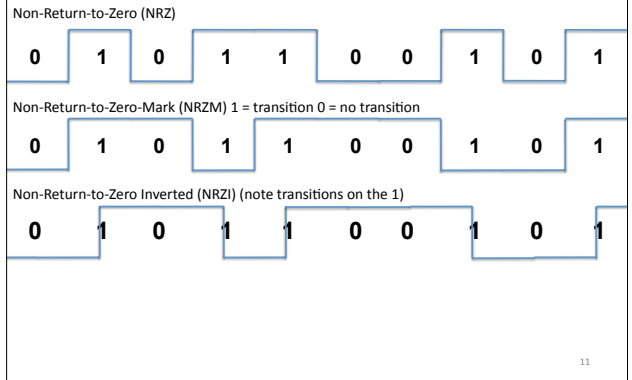
# Coding

Change the representation of data.

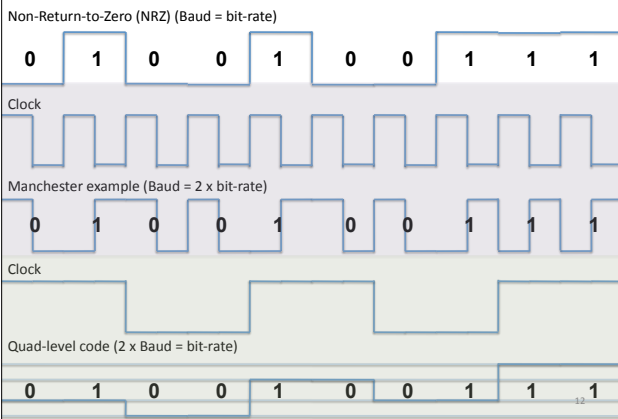


1. Encryption: MyPasswd <-> AA\$\$\$\$ff
2. Error Detection: AA\$\$\$\$ff <-> AA\$\$\$\$ffff
3. Compression: AA\$\$\$\$ffff <-> A2\$4f4
4. Analog: A2\$4f4 <->

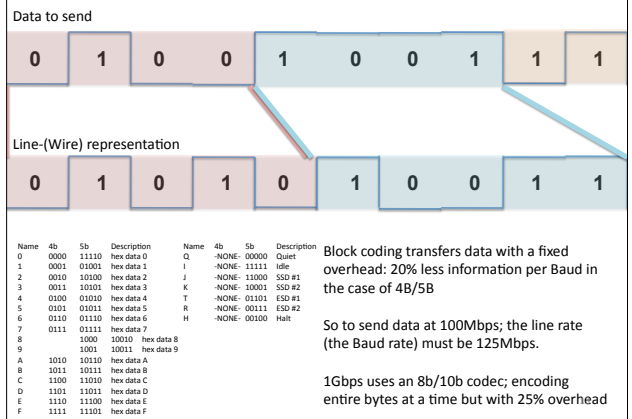
## Line Coding Examples where Baud=bit-rate



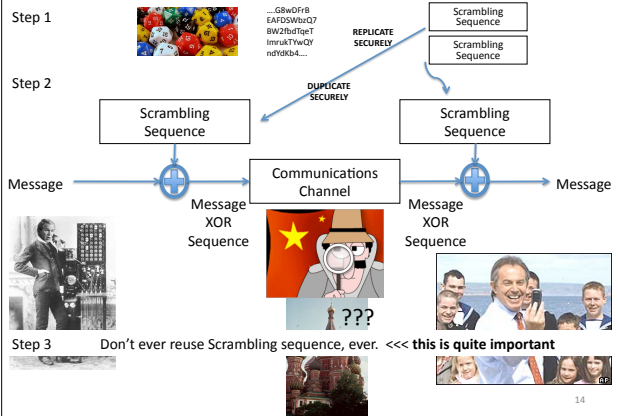
## Line Coding Examples



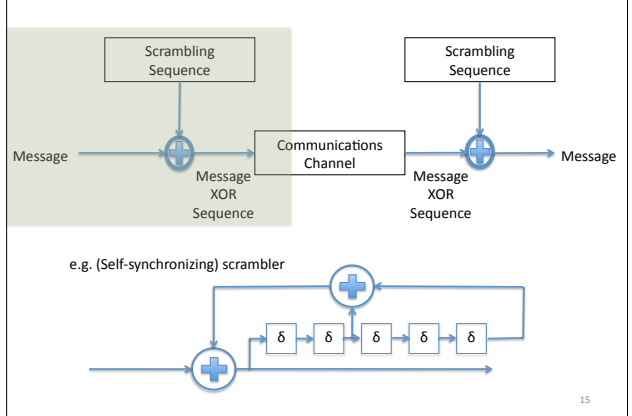
## Line Coding – Block Code example



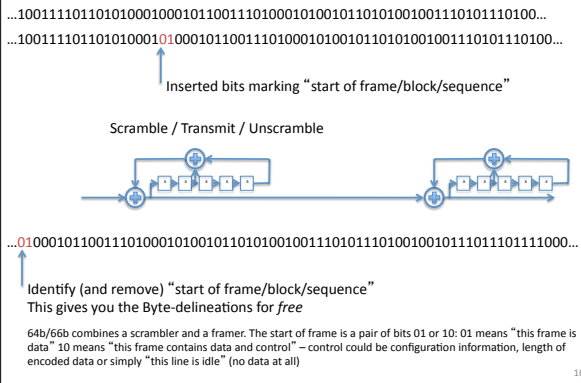
## Line Coding Scrambling – with secrecy



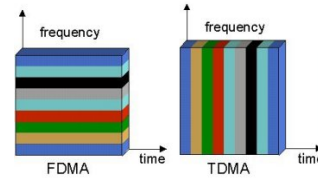
## Line Coding Scrambling– no secrecy



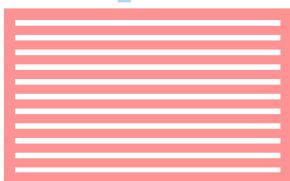
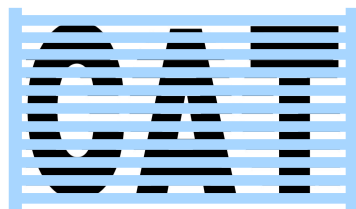
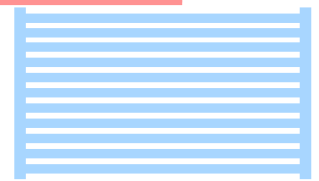
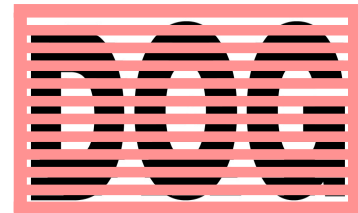
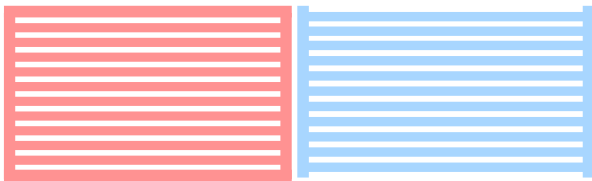
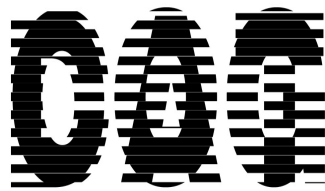
### Line Coding Examples (Hybrid)



### Multiple Access Mechanisms

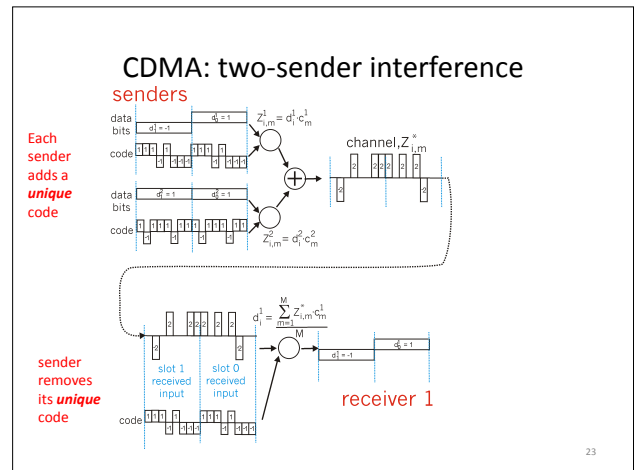
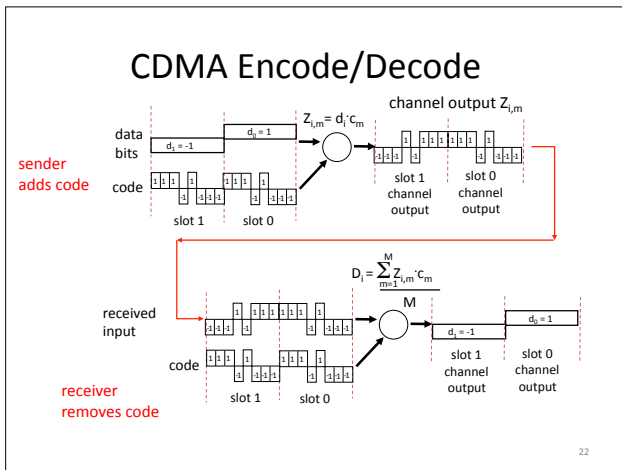


Each dimension is orthogonal (so may be trivially combined)  
 There are other dimensions too; can you think of them?



### Code Division Multiple Access (CDMA) (not to be confused with CSMA!)

- used in several wireless broadcast channels (cellular, satellite, etc) standards
- unique "code" assigned to each user; i.e., code set partitioning
- all users share same frequency, but each user has own "chipping" sequence (i.e., code) to encode data
- *encoded signal* = (original data) XOR (chipping sequence)
- *decoding*: inner-product of encoded signal and chipping sequence
- allows multiple users to "coexist" and transmit simultaneously with minimal interference (if codes are "orthogonal")



- ### Coding Examples summary
- Common Wired coding
    - Block codecs: table-lookups
      - fixed overhead, inline control signals
    - Scramblers: shift registers
      - overhead free
- Like earlier coding schemes and error correction/detection; you can combine these
- e.g. 10Gb/s Ethernet may use a hybrid
- CDMA (Code Division Multiple Access)
- coping intelligently with competing sources
  - Mobile phones
- 24

### Error Detection and Correction

How to use coding to deal with errors in data communication?

**Basic Idea :**

- Add additional information to a message.
- Detect an error and re-send a message.

Or, fix an error in the received message.

25

### Error Detection and Correction

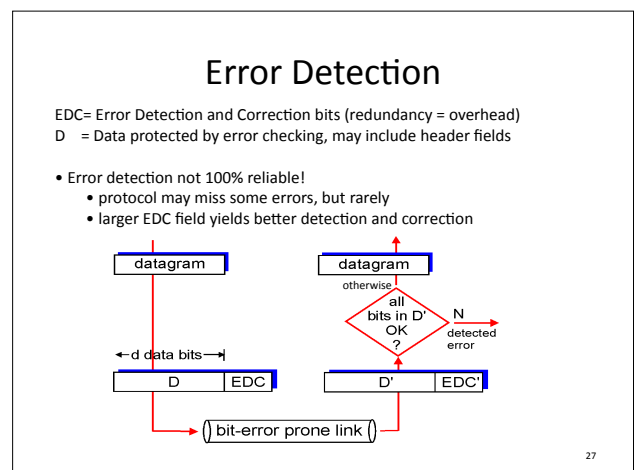
How to use coding to deal with errors in data communication?

**Basic Idea :**

- Add additional information to a message.
- Detect an error and re-send a message.

Or, fix an error in the received message.

26

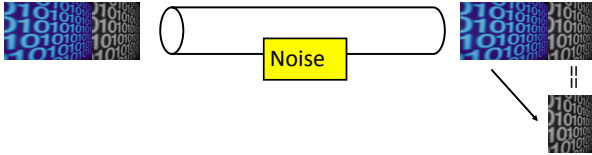




## Error Detection Code

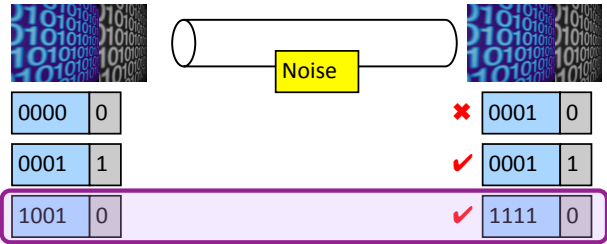
Sender:  
 $Y = \text{generateCheckBit}(X);$   
 $\text{send}(XY);$

Receiver:  
 $\text{receive}(X1Y1);$   
 $Y2 = \text{generateCheckBit}(X1);$   
 if  $(Y1 \neq Y2)$  ERROR;  
 else NOERROR



## Error Detection Code: Parity

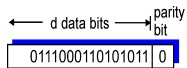
Add one bit, such that the number of 1's is even.



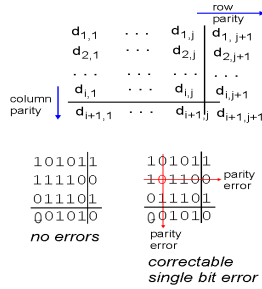
Problem: This simple parity cannot detect two-bit errors.

## Parity Checking

**Single Bit Parity:**  
 Detect single bit errors



**Two Dimensional Bit Parity:**  
 Detect and correct single bit errors



30

## Internet checksum

**Goal:** detect "errors" (e.g., flipped bits) in transmitted packet  
 (note: used at transport layer only)

**Sender:**

- treat segment contents as sequence of 1bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

**Receiver:**

- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected. *But maybe errors nonetheless?*

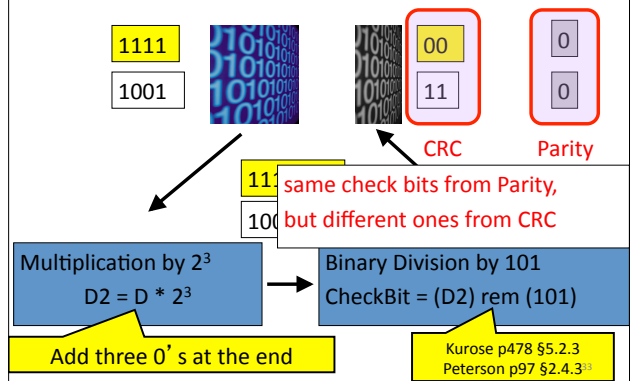
31

## Error Detection Code: CRC

- CRC means "Cyclic Redundancy Check".
- More powerful than parity.
  - It can detect various kinds of errors, including 2-bit errors.
- More complex: multiplication, binary division.
- Parameterized by n-bit divisor P.
  - Example: 3-bit divisor 101.
  - Choosing good P is crucial.

32

## CRC with 3-bit Divisor 101



## The divisor (G) – Secret sauce of CRC

- If the divisor were 100, instead of 101, data 1111 and 1001 would give the same check bit 00.
- Mathematical analysis about the divisor:
  - Last bit should be 1.
  - Should contain at least two 1's.
  - Should be divisible by 11.
- ATM, HDLC, Ethernet each use a CRC with well-chosen fixed divisors

Divisor analysis keeps mathematicians in jobs  
(a branch of *pure math*: combinatorial mathematics)

34

## Checksumming: Cyclic Redundancy Check recap

- view data bits,  $D$ , as a binary number
- choose  $r+1$  bit pattern (generator),  $G$
- goal: choose  $r$  CRC bits,  $R$ , such that
  - $\langle D, R \rangle$  exactly divisible by  $G$  (modulo 2)
  - receiver knows  $G$ , divides  $\langle D, R \rangle$  by  $G$ . If non-zero remainder: error detected!
  - can detect all burst errors less than  $r+1$  bits
- widely used in practice (Ethernet, 802.11 WiFi, ATM)



$$D * 2^r \text{ XOR } R \quad \text{mathematical formula}$$

35

## CRC Another Example – this time with long division

Want:

$$D \cdot 2^r \text{ XOR } R = nP$$

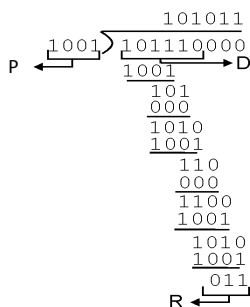
equivalently:

$$D \cdot 2^r = nP \text{ XOR } R$$

equivalently:

if we divide  $D \cdot 2^r$  by  $P$ ,  
want remainder  $R$

$$R = \text{remainder} \left[ \frac{D \cdot 2^r}{P} \right]$$



FYI: in K&R  $P$  is called the Generator:  $G$

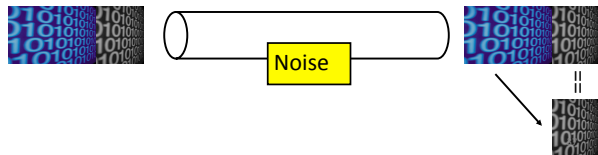
36

## Error Detection Code becomes....

Sender:  
Y = generateCheckBit(X);  
send(XY);

Receiver:

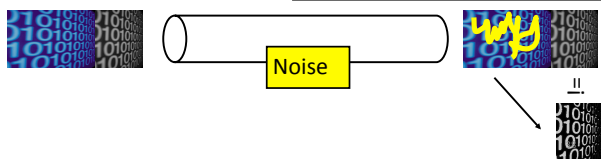
receive(X1Y1);  
Y2=generateCheckBit(X1);  
if (Y1 != Y2) ERROR;  
else NOERROR



## Forward Error Correction (FEC)

Sender:  
Y = generateCheckBit(X);  
send(XY);

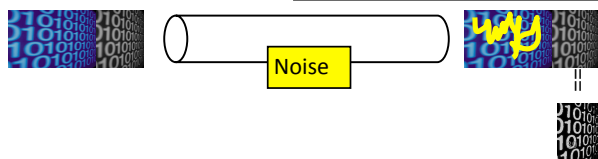
Receiver:  
receive(X1Y1);  
Y2=generateCheckBit(X1);  
if (Y1 != Y2) **FIXERROR(X1Y1)**;  
else NOERROR



## Forward Error Correction (FEC)

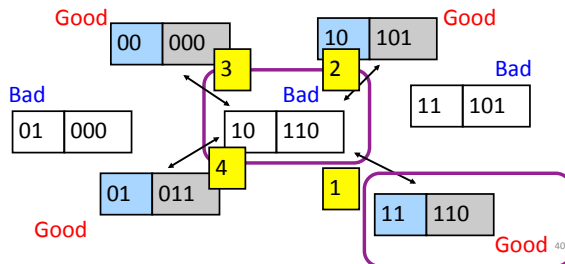
Sender:  
Y = generateCheckBit(X);  
send(XY);

Receiver:  
receive(X1Y1);  
Y2=generateCheckBit(X1);  
if (Y1 != Y2) **FIXERROR(X1Y1)**;  
else NOERROR



## Basic Idea of Forward Error Correction

Replace erroneous data by its "closest" error-free data.



## Error Detection vs Correction

Error Correction:

- Cons: More check bits. False recovery.
- Pros: No need to re-send.

Error Detection:

- Cons: Need to re-send.
- Pros: Less check bits.

Usage:

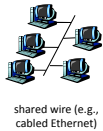
- Correction: A lot of noise. Expensive to re-send.
- Detection: Less noise. Easy to re-send.
- Can be used together.

41

## Multiple Access Links and Protocols

Two types of "links":

- point-to-point
  - point-to-point link between Ethernet switch and host
- **broadcast** (shared wire or medium)
  - old-fashioned wired Ethernet (*here be dinosaurs* – extinct)
  - upstream HFC (Hybrid Fiber-Coax – the Coax may be broadcast)
  - Home plug / Powerline networking
  - 802.11 wireless LAN



shared wire (e.g., cabled Ethernet)



shared RF (e.g., 802.11 WiFi)



shared RF (satellite)



humans at a cocktail party (shared air, acoustical)

42

## Multiple Access protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
  - **collision** if node receives two or more signals at the same time
- multiple access protocol
- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
  - no out-of-band channel for coordination

43

## Ideal Multiple Access Protocol

Broadcast channel of rate  $R$  bps

1. when one node wants to transmit, it can send at rate  $R$
2. when  $M$  nodes want to transmit, each can send at average rate  $R/M$
3. fully decentralized:
  - no special node to coordinate transmissions
  - no synchronization of clocks, slots
4. simple

44

## MAC Protocols: a taxonomy

Three broad classes:

- **Channel Partitioning**
  - divide channel into smaller "pieces" (time slots, frequency, code)
  - allocate piece to node for exclusive use
- **Random Access**
  - channel not divided, allow collisions
  - "recover" from collisions
- **"Taking turns"**
  - nodes take turns, but nodes with more to send can take longer turns

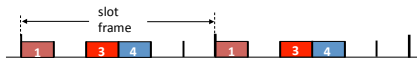
45



## Channel Partitioning MAC protocols: TDMA (time travel warning – we mentioned this earlier)

### TDMA: time division multiple access

- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: station LAN, 1,3,4 have pkt, slots 2,5,6 idle



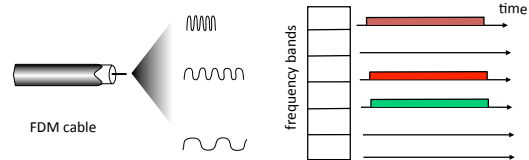
46



## Channel Partitioning MAC protocols: FDMA (time travel warning – we mentioned this earlier)

### FDMA: frequency division multiple access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: station LAN, 1,3,4 have pkt, frequency bands 2,5,6 idle



47

## “Taking Turns” MAC protocols

### channel partitioning MAC protocols:

- share channel *efficiently* and *fairly* at high load
- inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!

### Random access MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

### “taking turns” protocols

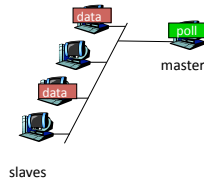
look for best of both worlds!

48

## “Taking Turns” MAC protocols

### Polling:

- master node “invites” slave nodes to transmit in turn
- typically used with “dumb” slave devices
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)

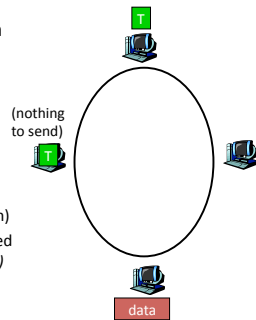


49

## “Taking Turns” MAC protocols

### Token passing:

- control **token** passed from one node to next sequentially.
- token message
- concerns:
  - token overhead
  - latency
  - single point of failure (token)
  - concerns fixed in part by a slotted ring (many simultaneous tokens)

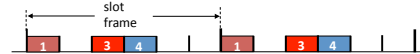


Cambridge students – this is YOUR heritage  
Cambridge RING, Cambridge Fast RING,  
Cambridge Backbone RING, these things gave us ATDM (and ATM)

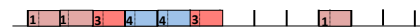
50

## ATM

In TDM a sender may only use a pre-allocated slot



In ATM a sender transmits labeled cells whenever necessary



ATM = Asynchronous Transfer Mode – an ugly expression  
think of it as ATDM – Asynchronous Time Division Multiplexing

That’s a variant of **PACKET SWITCHING** to the rest of us – just like Ethernet  
but using fixed length slots/packets/cells

Use the media when you need it, but

ATM had virtual circuits and these needed setup....

Worse ATM had an utterly irrational size

51

## Random Access MAC Protocols

- When node has packet to send
  - Transmit at full channel data rate
  - No *a priori* coordination among nodes
- Two or more transmitting nodes  $\Rightarrow$  collision
  - Data lost
- Random access MAC protocol specifies:
  - How to detect collisions
  - How to recover from collisions
- Examples
  - ALOHA and Slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA (wireless)

52

## Key Ideas of Random Access

- **Carrier sense**
  - Listen before speaking, and don't interrupt
  - Checking if someone else is already sending data
  - ... and waiting till the other node is done
- **Collision detection**
  - If someone else starts talking at the same time, stop
  - Realizing when two nodes are transmitting at once
  - ...by detecting that the data on the wire is garbled
- **Randomness**
  - Don't start talking again right away
  - Waiting for a random time before trying again

53

## CSMA (Carrier Sense Multiple Access)

- CSMA: listen before transmit
  - If channel sensed idle: transmit entire frame
  - If channel sensed busy, defer transmission
- Human analogy: don't interrupt others!
- Does this eliminate all collisions?
  - No, because of nonzero propagation delay

54

## CSMA Collisions

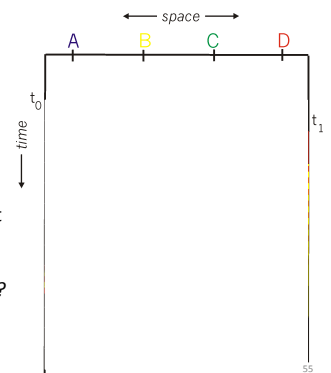
Propagation delay: two nodes may not hear each other's before sending.

Would slots hurt or help?

CSMA reduces but does not eliminate collisions

Biggest remaining problem?

Collisions still take full slot!  
How do you fix that?



55

## CSMA/CD (Collision Detection)

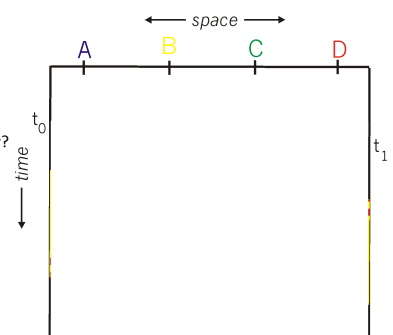
- CSMA/CD: carrier sensing, deferral as in CSMA
  - Collisions detected within short time
  - Colliding transmissions aborted, reducing wastage
- Collision detection easy in wired LANs:
  - Compare transmitted, received signals
- Collision detection difficult in wireless LANs:
  - Reception shut off while transmitting (well, perhaps not)
  - Not perfect broadcast (limited range) so collisions local
  - Leads to use of collision avoidance instead (later)

56

## CSMA/CD Collision Detection

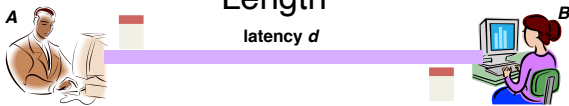
B and D can tell that collision occurred.

Note: for this to work, need restrictions on minimum frame size and maximum distance. Why?



57

## Limits on CSMA/CD Network Length



- Latency depends on physical length of link
  - Time to propagate a packet from one end to the other
- Suppose A sends a packet at time  $t$ 
  - And B sees an idle line at a time just before  $t+d$
  - ... so B happily starts transmitting a packet
- B detects a collision, and sends **jamming signal**
  - But A can't see collision until  $t+2d$

58

## Performance of CSMA/CD

- Time wasted in collisions
  - Proportional to distance  $d$
- Time spend transmitting a packet
  - Packet length  $p$  divided by bandwidth  $b$
- Rough estimate for efficiency ( $K$  some constant)

$$E \sim \frac{\frac{p}{b}}{\frac{p}{b} + Kd}$$

- Note:
  - For large packets, small distances,  $E \sim 1$
  - As bandwidth increases,  $E$  decreases
  - That is why high-speed LANs are all switched

59

## Benefits of Ethernet

- Easy to administer and maintain
- Inexpensive
- Increasingly higher speed
- Evolvable!

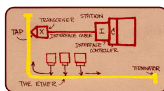
60

## Evolution of Ethernet

- Changed **everything** except the frame **format**
  - From single coaxial cable to hub-based star
  - From shared media to **switches**
  - From electrical signaling to optical
- **Lesson #1**
  - The right **interface** can accommodate many **changes**
  - Implementation is hidden behind interface
- **Lesson #2**
  - Really hard to displace the dominant technology
  - Slight performance improvements are not enough

61

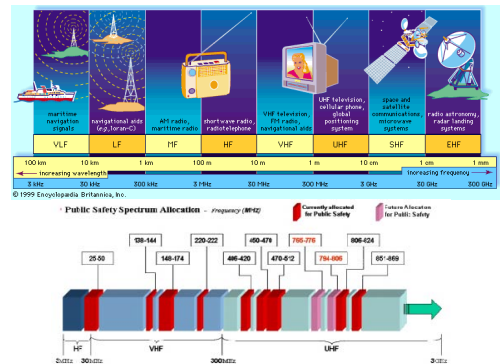
## Ethernet: CSMA/CD Protocol



- **Carrier sense:** wait for link to be idle
- **Collision detection:** listen while transmitting
  - No collision: transmission is complete
  - Collision: abort transmission & send **jam** signal
- **Random access:** **binary exponential back-off**
  - After collision, wait a random time before trying again
  - After  $m^{\text{th}}$  collision, choose  $K$  randomly from  $\{0, \dots, 2^m - 1\}$
  - ... and wait for  $K \cdot 512$  bit times before trying again
    - Using min packet size as "slot"
    - If transmission occurring when ready to send, wait until end of transmission (CSMA)

62

## The Wireless Spectrum



63

## Metrics for evaluation / comparison of wireless technologies

- Bitrate or Bandwidth
- Range - PAN, LAN, MAN, WAN
- Two-way / One-way
- Multi-Access / Point-to-Point
- Digital / Analog
- Applications and industries
- Frequency – Affects most physical properties:
  - Distance (free-space loss)
  - Penetration, Reflection, Absorption
  - Energy proportionality
  - Policy: Licensed / Deregulated
  - Line of Sight (Fresnel zone)
  - Size of antenna
- Determined by wavelength –  $\lambda = \frac{v}{f}$ ,

64

## Wireless Communication Standards

- Cellular (800/900/1700/1800/1900MHz):
  - 2G: GSM / CDMA / GPRS / EDGE
  - 3G: CDMA2000/UMTS/HSDPA/EVDO
  - 4G: LTE, WiMax
- IEEE 802.11 (aka WiFi):
  - b: 2.4Ghz band, 11Mbps (~4.5 Mbps operating rate)
  - g: 2.4Ghz, 54-108Mbps (~19 Mbps operating rate)
  - a: 5.0Ghz band, 54-108Mbps (~25 Mbps operating rate)
  - n: 2.4/5Ghz, 150-600Mbps (4x4 mimo).
- IEEE 802.15 – lower power wireless:
  - 802.15.1: 2.4Ghz, 2.1 Mbps (Bluetooth)
  - 802.15.4: 2.4Ghz, 250 Kbps (Sensor Networks)

65

## What Makes Wireless Different?

- Broadcast and multi-access medium...
  - err, so....
- BUT, Signals sent by sender don't always end up at receiver intact
  - Complicated physics involved, which we won't discuss
  - But what can go wrong?

66

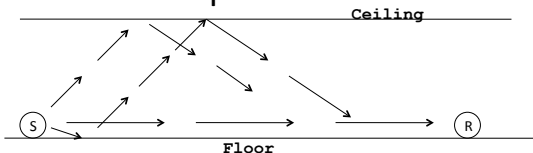
## Path Loss / Path Attenuation

- Free Space Path Loss: 
$$\text{FSPL} = \left(\frac{4\pi d}{\lambda}\right)^2 = \left(\frac{4\pi df}{c}\right)^2$$
  - d = distance
  - $\lambda$  = wave length
  - f = frequency
  - c = speed of light

- Reflection, Diffraction, Absorption
- Terrain contours (Urban, Rural, Vegetation).
- Humidity

67

## Multipath Effects



- Signals bounce off surface and interfere with one another
- Self-interference

68

## Interference from Other Sources

- External Interference
  - Microwave is turned on and blocks your signal
  - Would that affect the sender or the receiver?
- Internal Interference
  - Hosts within range of each other collide with one another's transmission
- We have to tolerate path loss, multipath, etc., but we can try to avoid internal interference

69

## Wireless Bit Errors

- The lower the SNR (Signal/Noise) the higher the Bit Error Rate (BER)
- We could make the signal stronger...
- Why is this not always a good idea?
  - Increased signal strength requires more power
  - Increases the interference range of the sender, so you interfere with more nodes around you
    - And then they increase their power.....
- Local link-layer Error Correction schemes can correct **some** problems

70

## Lets focus on 802.11

aka - WiFi ...  
What makes it special?

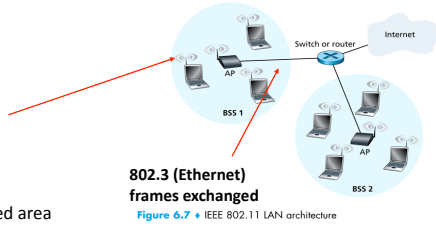
Deregulation > Innovation > Adoption > Lower cost = Ubiquitous technology

JUST LIKE ETHERNET – not lovely but sufficient

71

## 802.11 Architecture

802.11 frames exchanges



- Designed for limited area
- AP's (Access Points) set to specific channel
- Broadcast beacon messages with SSID (Service Set Identifier) and MAC Address periodically
- Hosts scan all the channels to discover the AP's
  - Host associates with AP

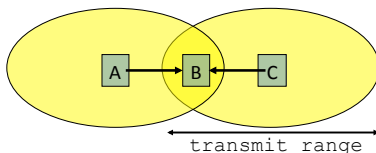
72

## Wireless Multiple Access Technique?

- Carrier Sense?
  - Sender can listen before sending
  - What does that tell the sender?
- Collision Detection?
  - Where do collisions occur?
  - How can you detect them?

73

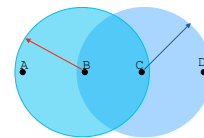
## Hidden Terminals



- A and C can both send to B but **can't hear each other**
  - A is a *hidden terminal* for C and vice versa
- Carrier Sense will be **ineffective**

74

## Exposed Terminals



- **Exposed node:** B sends a packet to A; C hears this and decides not to send a packet to D (despite the fact that this will not cause interference)!
- Carrier sense would prevent a successful transmission.

75



## Key Points

- No concept of a global collision
  - Different receivers hear different signals
  - Different senders reach different receivers
- Collisions are at receiver, not sender
  - Only care if receiver can hear the sender clearly
  - It does not matter if sender can hear someone else
  - As long as that signal does not interfere with receiver
- Goal of protocol:
  - Detect if receiver can hear sender
  - Tell senders who might interfere with receiver to shut up

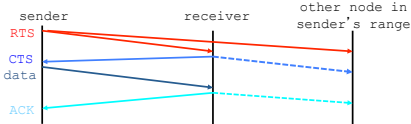
76

## Basic Collision Avoidance

- Since can't detect collisions, we try to *avoid* them
- Carrier sense:
  - When medium busy, choose random interval
  - Wait that many *idle* timeslots to pass before sending
- When a collision is inferred, retransmit with binary exponential backoff (like Ethernet)
  - Use **ACK** from receiver to infer “no collision”
  - Use exponential backoff to adapt contention window

77

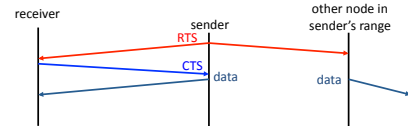
## CSMA/CA -MA with Collision Avoidance



- Before every data transmission
  - Sender sends a Request to Send (RTS) frame containing the length of the transmission
  - Receiver respond with a Clear to Send (CTS) frame
  - Sender sends data
  - Receiver sends an ACK; now another sender can send data
- When sender doesn't get a CTS back, it assumes collision

78

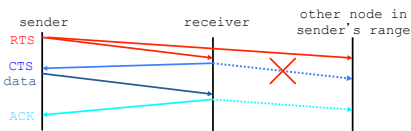
## CSMA/CA, con't



- If other nodes hear RTS, but not CTS: **send**
  - Presumably, destination for first sender is out of node's range ...

79

## CSMA/CA, con't

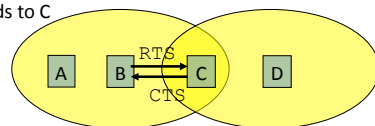


- If other nodes hear RTS, but not CTS: **send**
  - Presumably, destination for first sender is out of node's range ...
  - ... Can cause problems when a CTS is **lost**
- When you hear a CTS, you keep quiet until scheduled transmission is over (hear ACK)

80

## RTS / CTS Protocols (CSMA/CA)

B sends to C



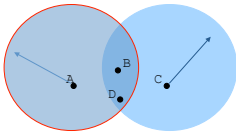
Overcome hidden terminal problems with contention-free protocol

1. B sends to C Request To Send (RTS)
2. A hears RTS and defers (to allow C to answer)
3. C replies to B with Clear To Send (CTS)
4. D hears CTS and defers to allow the data
5. B sends to C

81

## Preventing Collisions Altogether

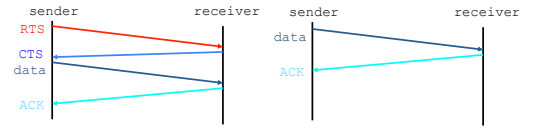
- Frequency Spectrum partitioned into several channels
  - Nodes within interference range can use separate channels



- Now A and C can send without any interference!
- Most cards have only 1 transceiver
  - Not Full Duplex: Cannot send and receive at the same time**
- Aggregate Network throughput doubles

82

## CSMA/CA and RTS/CTS



### RTS/CTS

- helps with hidden terminal
- good for high-traffic Access Points
- often turned on/off dynamically

### Without RTS/CTS

- lower latency -> faster!
- reduces wasted b/w if the  $Pr(\text{collision})$  is low
- good for when net is small and not weird
  - eg no hidden/exposed terminals

83

## CSMA/CD vs CSMA/CA (without RTS/CTS)

### CD Collision Detect

wired – listen and talk

- Listen for others
- Busy? goto 1.
- Send message (and listen)
- Collision?
  - JAM
  - increase your BEB
  - sleep
  - goto 1.

### CA Collision Avoidance

wireless – talk OR listen

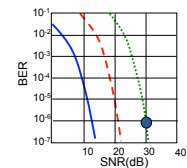
- Listen for others
- Busy?
  - increase your BEB
  - sleep
  - goto 1.
- Send message
- Wait for ACK (*MAC ACK*)
  - increase your BEB
  - sleep
  - goto 1.
- Got No ACK from MAC?
  - increase your BEB
  - sleep
  - goto 1.

84

## Changing the rules: an 802.11 feature

### Rate Adaptation

- base station, mobile dynamically change transmission rate (physical layer modulation technique) as mobile moves, SNR varies



- SNR decreases, BER increase as node moves away from base station
- When BER becomes too high, switch to lower transmission rate but with lower BER

..... QAM256 (8 Mbps)  
 - - - - QAM16 (4 Mbps)  
 ——— BPSK (1 Mbps)  
 ● operating point

85

## Summary of MAC protocols

- channel partitioning**, by time, frequency or code
  - Time Division, Frequency Division
- random access** (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - carrier sensing: easy in some technologies (wire), hard in others (wireless)
  - CSMA/CD used in Ethernet
  - CSMA/CA used in 802.11
- taking turns**
  - polling from central site, token passing
  - Bluetooth, FDDI, IBM Token Ring

86

## MAC Addresses

- MAC (or LAN or physical or Ethernet) address:
  - function: *get frame from one interface to another physically-connected interface (same network)*
  - 48 bit MAC address (for most LANs)
    - burned* in NIC ROM, nowadays usually software settable and set at boot time

```

oam22@rio:~$ ifconfig eth0
eth0  Link encap:Ethernet  HWaddr 00:30:48:fe:c0:64
      inet addr:128.232.33.4  Bcast:128.232.47.255  Mask:255.255.240.0
      inet6 addr: fe80::230:48ff:fe:c064/64  Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:215084512  errors:252  dropped:25  overruns:0  frame:123
      TX packets:146711866  errors:0  dropped:0  overruns:0  carrier:0
      collisions:0  txqueuelen:1000
      RX bytes:170815941033 (170.8 GB)  TX bytes:86755864270 (86.7 GB)
      Memory: f0000000-f0020000
    
```

87

## LAN Address (more)

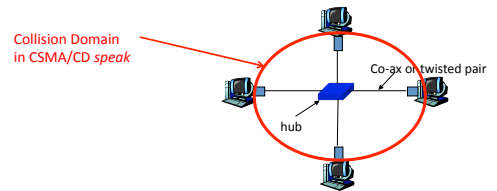
- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- analogy:
  - (a) MAC address: like Social Security Number
  - (b) IP address: like postal address
- MAC flat address → portability
  - can move LAN card from one LAN to another
- IP hierarchical address NOT portable
  - address depends on IP subnet to which node is attached

88

## Hubs

... physical-layer ("dumb") repeaters:

- bits coming in one link go out *all* other links at same rate
- all nodes connected to hub can collide with one another
- no frame buffering
- no CSMA/CD at hub: host NICs detect collisions

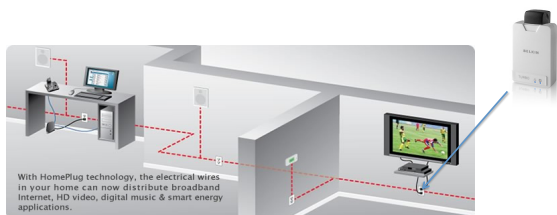


89

## CSMA/CD Lives....



### Home Plug and similar Powerline Networking....



90

## Switch

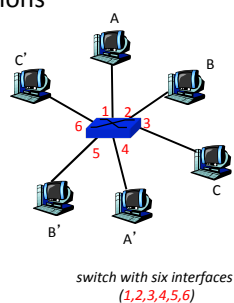
(like a Hub but smarter)

- **link-layer device: smarter than hubs, take active role**
  - store, forward Ethernet frames
  - examine incoming frame's MAC address, **selectively** forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
- **transparent**
  - hosts are unaware of presence of switches
- **plug-and-play, self-learning**
  - switches do not need to be configured

91

## Switch: allows *multiple* simultaneous transmissions

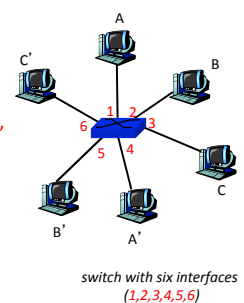
- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, but no collisions; full duplex
  - each link is its own collision domain
- **switching**: A-to-A' and B-to-B' simultaneously, without collisions
  - not possible with dumb hub



92

## Switch Table

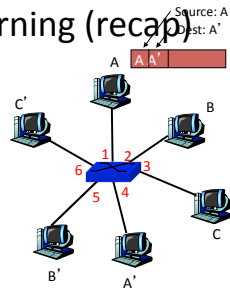
- **Q:** how does switch know that A' reachable via interface 4, B' reachable via interface 5?
- **A:** each switch has a **switch table**, each entry:
  - (MAC address of host, interface to reach host, time stamp)
- looks like a routing table!
- **Q:** how are entries created, maintained in switch table?
  - something like a routing protocol?



93

## Switch: self-learning (recap)

- switch **learns** which hosts can be reached through which interfaces
  - when frame received, switch "learns" location of sender: incoming LAN segment
  - records sender/location pair in switch table



MAC addr	interface	TTL
A	1	60

Switch table (initially empty)

94

## Switch: frame filtering/forwarding

### When frame received:

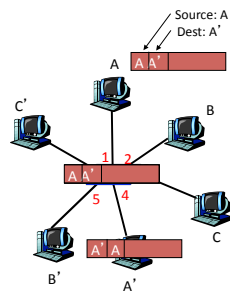
- record link associated with sending host
- index switch table using MAC dest address
- if entry found for destination
  - then {
    - if dest on segment from which frame arrived
      - then drop the frame
      - else forward the frame on interface indicated
    - }
  - else flood

forward on all but the interface on which the frame arrived

95

## Self-learning, forwarding: example

- frame destination unknown: **flood**
- destination A location known: **selective send**



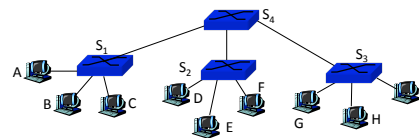
MAC addr	interface	TTL
A	1	60
A'	4	60

Switch table (initially empty)

96

## Interconnecting switches

- switches can be connected together

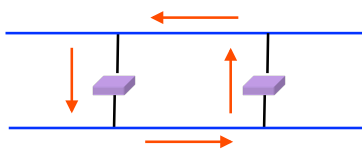


- Q:** sending from A to G - how does S<sub>1</sub> know to forward frame destined to F via S<sub>2</sub> and S<sub>3</sub>?
- A:** self learning! (works exactly the same as in single-switch case – flood/forward/drop)

97

## Flooding Can Lead to Loops

- Flooding can lead to **forwarding loops**
  - E.g., if the network contains a cycle of switches
  - "Broadcast storm"

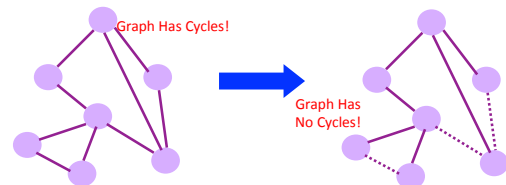


98



## Solution: Spanning Trees

- Ensure the forwarding **topology** has no loops
  - Avoid using some of the links when flooding
  - ... to prevent loop from forming
- Spanning tree**
  - **Sub-graph** that covers all vertices but *contains no cycles*
  - Links not in the spanning tree do not forward frames



99



## Topic 3: 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
  - WiFi
- algorithms
  - Binary Exponential Backoff
  - Spanning Tree