Computer Networking

Slide Set 2

Andrew W. Moore
Andrew.Moore@cl.cam.ac.uk

Topic 3.0: The Physical Layer

Our goals:

- Understand physical channel fundamentals
  - Physical channels can carry data in proportion to the signal and inversely in proportion to noise
  - Modulation represents Digital data in analog channels
  - Baseband vs. Broadband
  - Synchronous vs. Aynchronous

2

2

Physical Channels / The Physical Layer

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

Twisted Pair (TP)

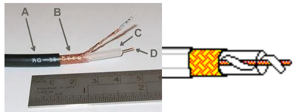
- two insulated copper wires
  - Category 3: traditional phone wires, 10 Mbps Ethernet
  - Category 8: 25Gbps Ethernet
- Shielded (STP)
- Unshielded (UTP)

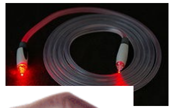
Coaxial cable:

- 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 Gbps)
- low error rate
- immune to electromagnetic noise





3

3

More Physical media: Radio

- Bidirectional and multiple access
- propagation environment effects:
  - reflection
  - obstruction by objects
  - interference

Radio link types:

- terrestrial microwave
  - e.g. 90 Mbps channels
- LAN (e.g., Wifi)
  - 11Mbps, 54 Mbps, 600 Mbps
- wide-area (e.g., cellular)
  - 5G cellular: ~ 40 Mbps - 10Gbps
- satellite
  - 27-50MHz typical bandwidth
  - geosynchronous versus low altitude
  - For geosync - 270 msec end-end delay to orbit

4

4

Physical Channel Characteristics - Fundamental Limits -

symbol type: generally, an analog waveform — voltage, current, photo intensity etc.

capacity: bandwidth

delay: speed of light in medium and distance travelled

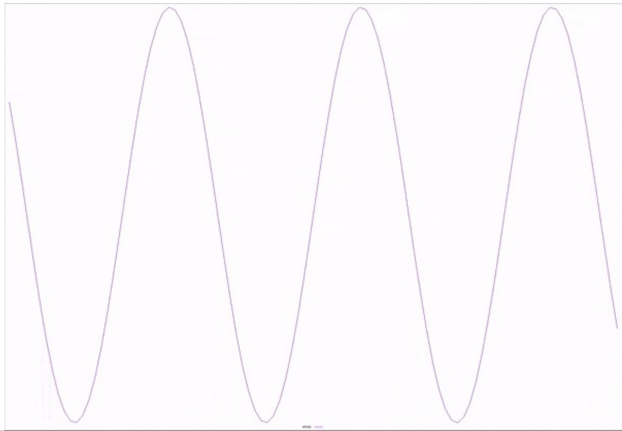
fidelity: signal to noise ratio

- measure of the range of frequencies of sinusoidal signal that channel supports
- E.g., a channel that supports sinusoids from 1 MHz to 1.1 MHz has a bandwidth of 100 KHz
- "supports" in this context means "comes out the other end of the channel"
- some frequencies supported better than others
- analysing what happens to an arbitrary waveform is done by examining what happens to its component sinusoids → Fourier analysis
- bandwidth is a resource

5

5

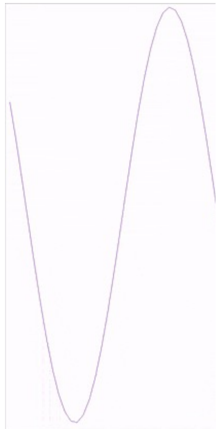
Analog meet Digital



6

6

Analog meet Digital



Square waves have high frequency components in them

Channels attenuate frequencies irregularly:
changing the shape of the signal

Receiver signal is related to the transmitted signal + noise

Noise may be systematic or random

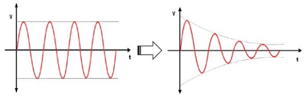
Systematic noise from interfering equipment
can in principle be eliminated (not always convenient)

Random noise caused by thermal vibration (thermal noise)

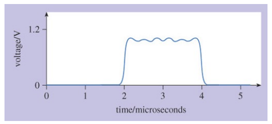
“White” noise is evenly distributed across frequencies
signal to noise ratio *S/N*
more distance more noise

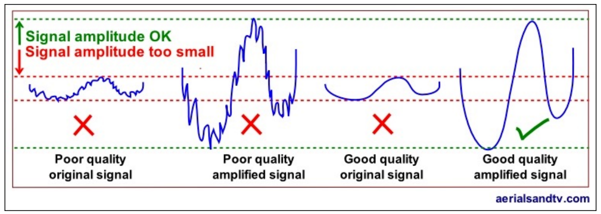
7

Noise: Enemy of Communications



Attenuation, External Noise,
Systematic, non-systematic,
digitization, interference, reflection, ...





8

8

Bandwidth vs Signal to Noise

what’s better: high bandwidth or low signal to noise?

- for channels with white noise have information capacity *C* measured in bits per second, of a channel

$$C = Blog_2(1 + S/N)$$

B is the bandwidth of the channel *S/N* is the ratio of received signal power to received noise power.

- channels with no noise have infinite information capacity
- channels with any signal have nonzero information capacity
- channels with signal to noise ratio of unity have an information capacity in bits per second equal to its bandwidth in hertz
- (This is actually NOT the definition of information capacity; it is derived from the definition)

9

9

(Digital) Channels

- Physical layer provides a channel
- Fixed rate for now
- Symbols are discrete values sent on the channel at fixed rate
- Symbols need not be binary
- Fidelity of the channel usually measured as a bit error rate — the probability that a bit sent as a 1 was interpreted as a 0 by the receiver or vice versa.

- Baud rate is the rate at which symbols can be transmitted
- Data rate (or bit rate) is the equivalent number of binary digits which can be sent
- E.g., if symbols represent with rate R then the data rate is 2 × R.

10

10

Modulation

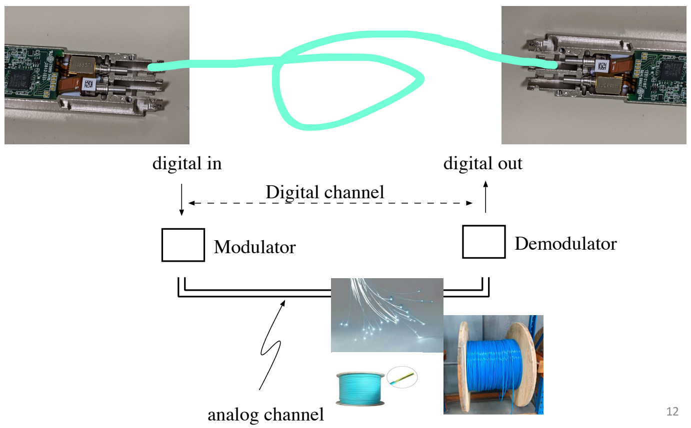
Two definitions:

- Transform an information signal into a signal more appropriate for transmission on a physical medium
- The systematic alteration of a carrier waveform by an information signal

In general, we mean the first here
(which encompasses the second).

11

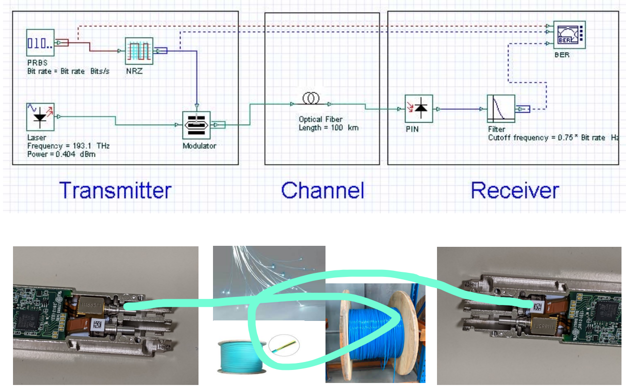
11



12

12

Communications

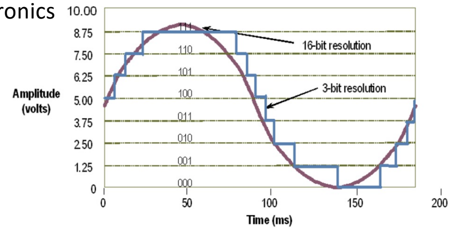


13

13

Analog/Digital Digital/Analog

Recall from Digital Electronics



Conversion errors can occur in both directions

e.g.

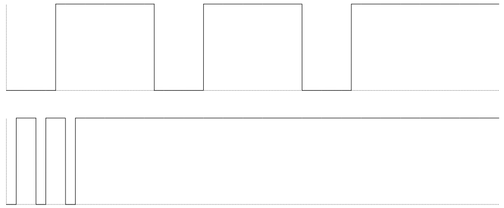
Noise leads to incorrect digitization

Insufficient digitization resolution leads to information loss

14

14

More Challenges



Where are the bits?

WHEN are the bits?

Bit boundaries can be asynchronous or synchronous

15

15

Asynchronous versus Synchronous

- Transmission is sporadic, divided into frames
- Receiver and transmitter have oscillators which are close in frequency producing tx clocks and rx clock
- Receiver synchronises the phase of the rx clock with the tx clock by looking at one or more bit transitions
- RX clock drifts with respect to the tx clock but stays within a fraction of a bit of tx clock throughout the duration of a frame
- Transmission time is limited by accuracy of oscillators

- Transmission is continuous
- Receiver continually adjusts its frequency to track clock from incoming signal
- Requires bit transitions to inform clock
- Phase locked loop: rx clock predicts when incoming clock will change and corrects slightly when wrong.

16

16

Asynchronous versus Synchronous

- Transmission is sporadic, divided into frames
- Receiver and transmitter have oscillators which are close in frequency producing tx clocks and rx clock
- Receiver synchronises the phase of the rx clock with the tx clock by looking at one or more bit transitions
- RX clock drifts with respect to the tx clock but stays within a fraction of a bit of tx clock throughout the duration of a frame
- Transmission time is limited by accuracy of oscillators

- Transmission is continuous
- Receiver continually adjusts its frequency to track clock from incoming signal
- Requires bit transitions to inform clock
- Phase locked loop: rx clock predicts when incoming clock will change and corrects slightly when wrong.

Bit transitions are critical

17

17

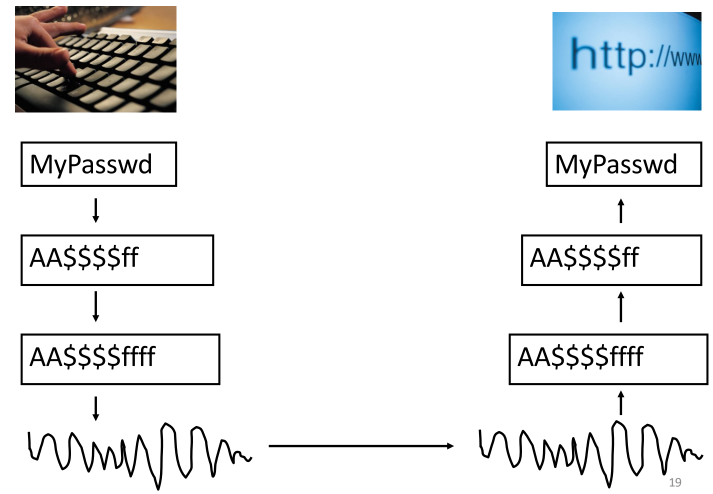
Coding – a channel function

Change the representation of data.

Given Data → Encoding → Changed Data

Changed Data → Decoding → Given Data

18



19

Coding

Change the representation of data.

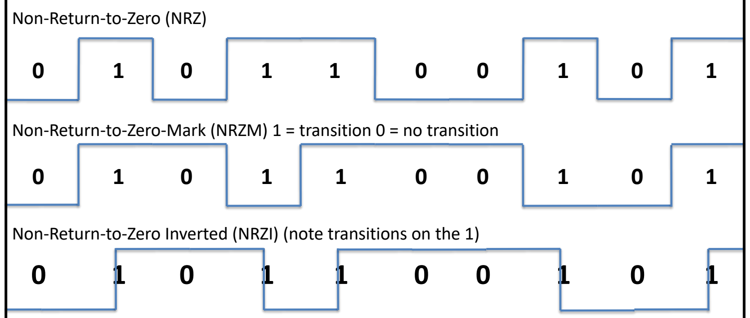
Given Data → Encoding → Changed Data

Changed Data → Decoding → Given Data

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

20

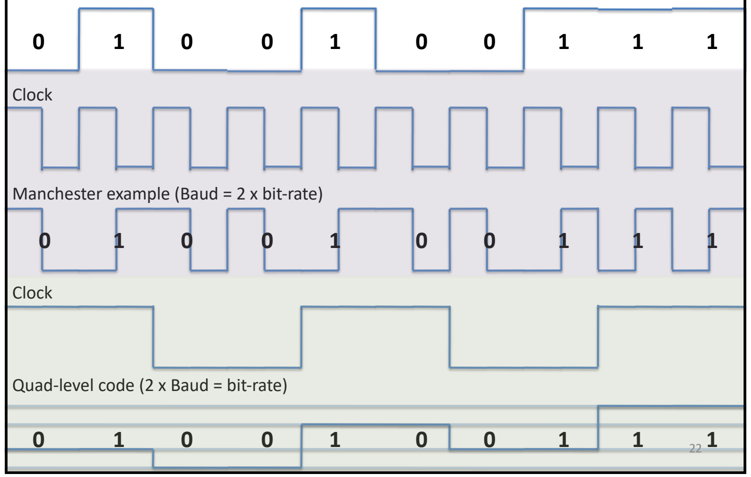
Line Coding Examples
where Baud=bit-rate



21

Line Coding Examples

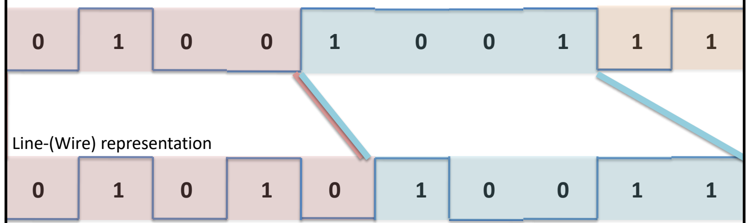
Non-Return-to-Zero (NRZ) (Baud = bit-rate)



22

Line Coding – Block Code example

Data to send



Name	4b	5b	Description	Name	4b	5b	Description
0	0000	11110	hex data 0	Q	-NONE-	00000	Quiet
1	0001	01001	hex data 1	I	-NONE-	11111	Idle
2	0010	10100	hex data 2	J	-NONE-	11000	SSD #1
3	0011	10101	hex data 3	K	-NONE-	10001	SSD #2
4	0100	01010	hex data 4	T	-NONE-	01101	ESD #1
5	0101	01011	hex data 5	R	-NONE-	00111	ESD #2
6	0110	01110	hex data 6	H	-NONE-	00100	Halt
7	0111	01111	hex data 7				
8	1000	10010	hex data 8				
9	1001	10011	hex data 9				
A	1010	10110	hex data A				
B	1011	10111	hex data B				
C	1100	11010	hex data C				
D	1101	11011	hex data D				
E	1110	11100	hex data E				
F	1111	11101	hex data F				

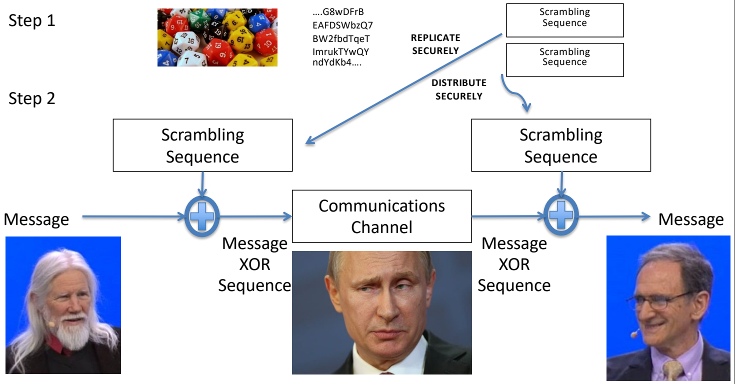
Block coding transfers data with a fixed overhead: 20% less information per Baud in the case of 4B/5B

So to send data at 100Mbps; the line rate (the Baud rate) must be 125Mbps.

1Gbps uses an 8b/10b codec; encoding entire bytes at a time but with 25% overhead

23

Line Coding Scrambling – with secrecy



Step 3 Don't ever reuse Scrambling sequence, ever. <<< **this is quite important**

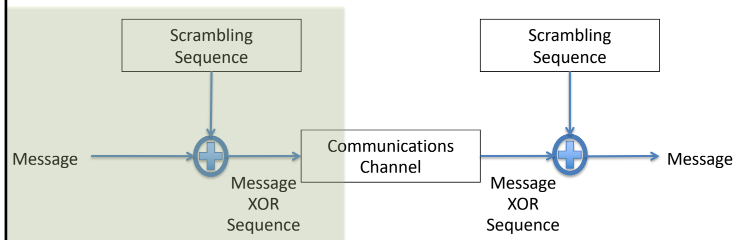
Whitfield Diffie

Martin Hellman

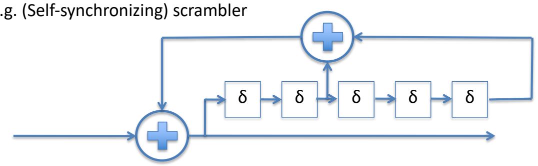
24

24

Line Coding Scrambling– no secrecy



e.g. (Self-synchronizing) scrambler



25

25

Line Coding Examples (Hybrid)

…1001111011010100010001011001110100010100101101010010011101011110100…

…1001111011010100010100010110011101000101001011010100100111010111101100…

Inserted bits marking "start of frame/block/sequence"

Scramble / Transmit / Unscramble

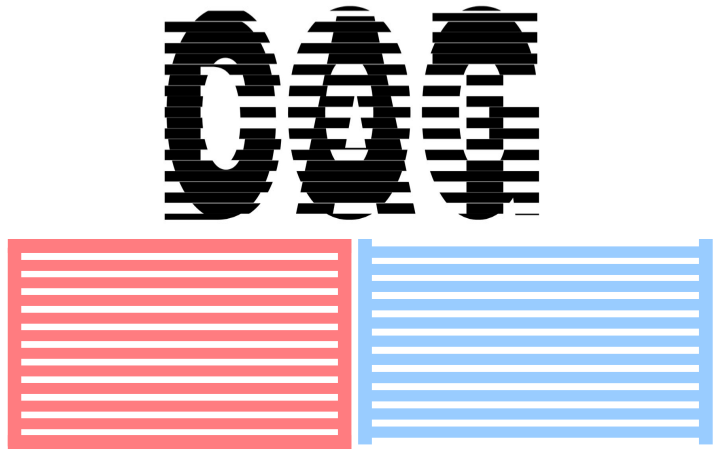
…0100010110011101000101001011010100100111010111010010010111011101111000…

Identify (and remove) "start of frame/block/sequence"
This gives you the Byte-delineations for *free*

64b/66b combines a scrambler and a framer. The start of frame is a pair of bits 01 or 10: 01 means "this frame is data" 10 means "this frame contains data and control" – control could be configuration information, length of encoded data or simply "this line is idle" (no data at all)

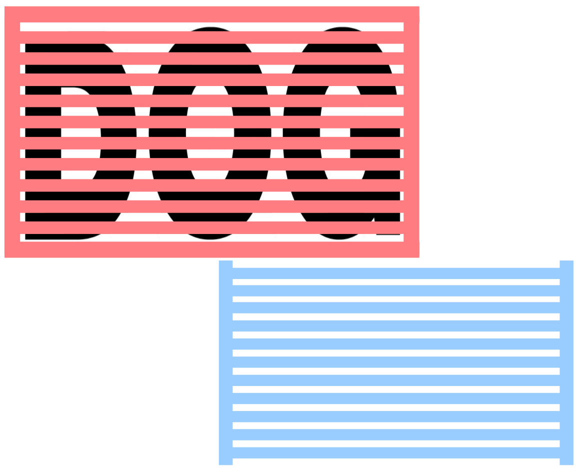
26

26



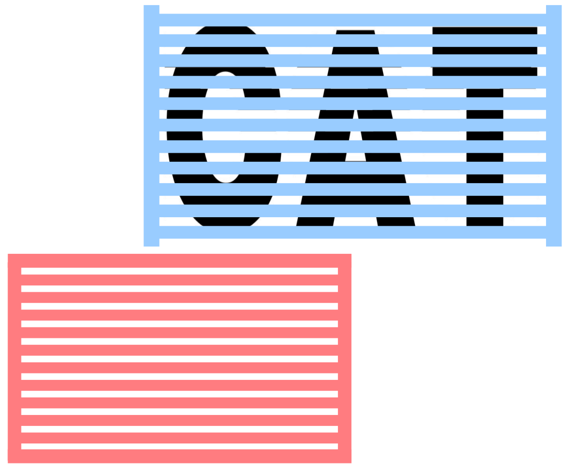
27

27



28

28



29

29

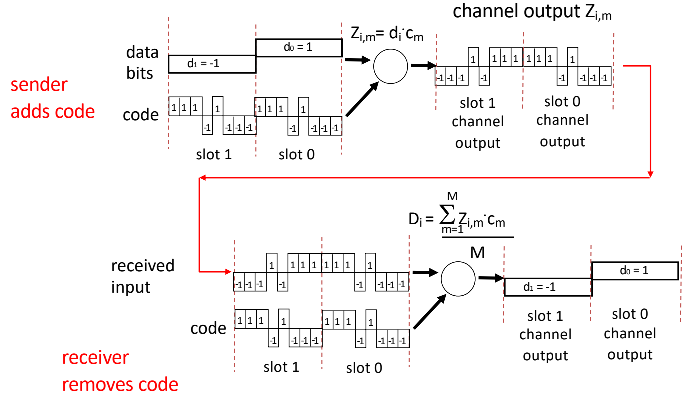
Code Division Multiple Access (CDMA) (not to be confused with CSMA!)

- 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")

30

30

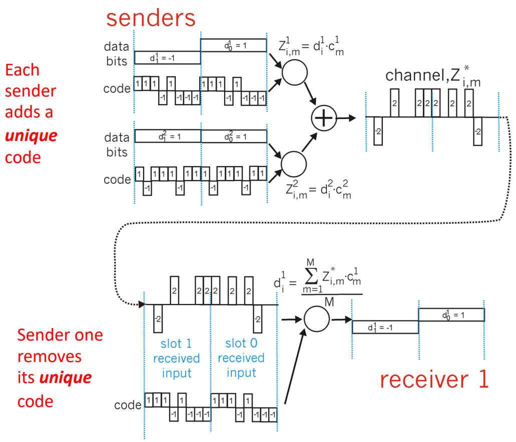
CDMA Encode/Decode



31

31

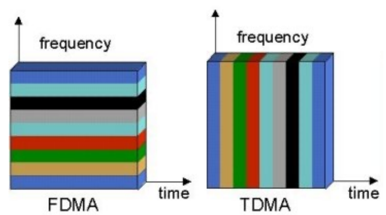
CDMA: two-sender interference



32

32

Multiple Access Mechanisms



Each dimension is orthogonal (so may be trivially combined)
Other dimensions are also available…

33

33

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

34

34

Error Detection and Correction

Transmission media are not perfect and cause signal impairments:

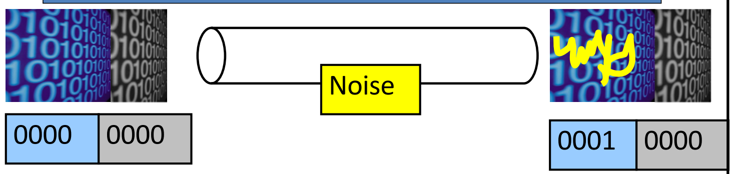
1. Attenuation
   - Loss of energy to overcome medium's resistance
2. Distortion
   - The signal changes its form or shape, caused in composite signals
3. Noise
   - Thermal noise, induced noise, crosstalk, impulse noise

Interference can change the shape or timing of a signal:
0 → 1 or 1 → 0

35

Error Detection and Correction

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



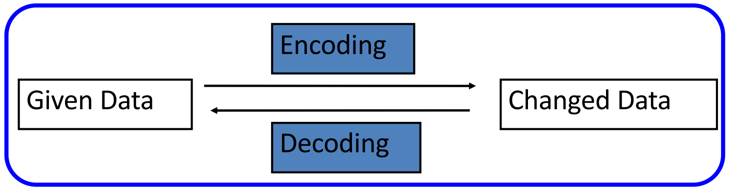
Basic Idea :
1. Add additional information (redundancy) to a message.
2. Detect an error and discard
   Or, fix an error in the received message.

36

36

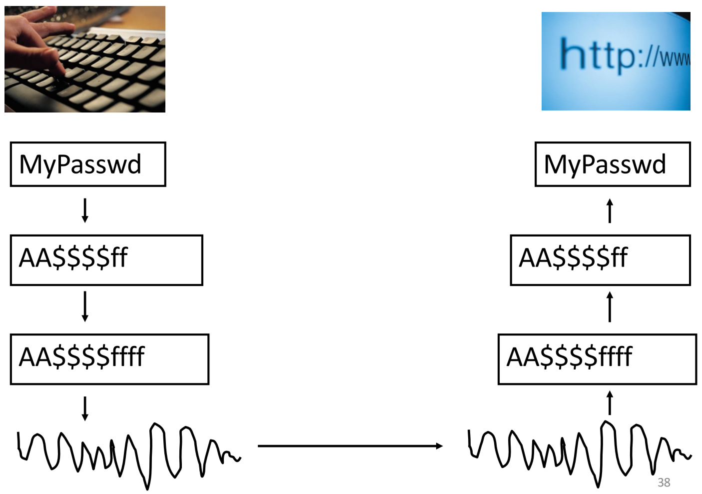
Coding – a channel function

Change the representation of data.



37

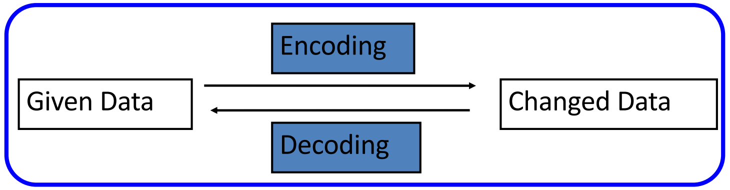
37

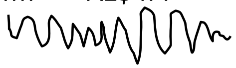


38

Coding Examples

Changig the representation of data.



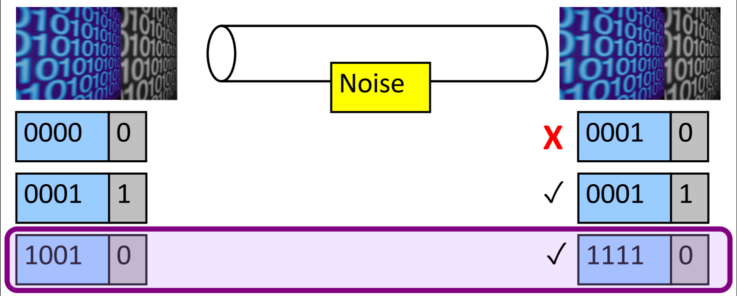
1. Encryption: MyPasswd <-> AA$$$$ff
2. Error Detection: AA$$$$ff <-> AA$$$$ffff
3. Compression: AA$$$$ffff <-> A2$4f4
4. Analog: A2$4f4 <-> 

39

39

Error Detection Code: Parity

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



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

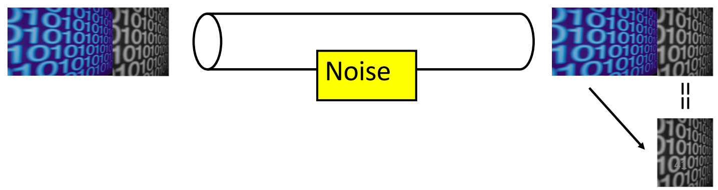
40

40

Error Detection Code

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

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



41

Error Detection Code: CRC

- CRC means "Cyclic Redundancy Check".
- *"A sequence of redundant bits, called CRC, is appended to the end of data so that the resulting data becomes exactly divisible by a second, predetermined binary number."*
- *CRC:= remainder (data ÷ predetermined divisor)*
- 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.

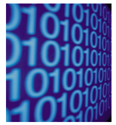
42

42

CRC with 3-bit Divisor 101

1111

1001



00

11

CRC

0

0

Parity

111

100

same check bits from Parity,

but different ones from CRC

Multiplication by 2^3

D2 = D * 2^3

Binary Division by 101

CheckBit = (D2) rem (101)

Add three 0's at the end

Kurose p478 §5.2.3

Peterson URL §2.4

43

Error Detection Code

Sender:

Y = generateCRC(X div P);

send(X);

send(Y);

Receiver:

receive(X1);

receive(Y1);

Y2=generateCRC(X1Y1 div P);

if (Y2 != 0s) ERROR;

else NOERROR

Noise

0s ==

44

Transforming Error Detection to…

Sender:

Y = generateCheckBit(X);

send(XY);

Receiver:

receive(X1Y1);

Y2=generateCheckBit(X1);

if (Y1 != Y2) ERROR;

else NOERROR

Noise

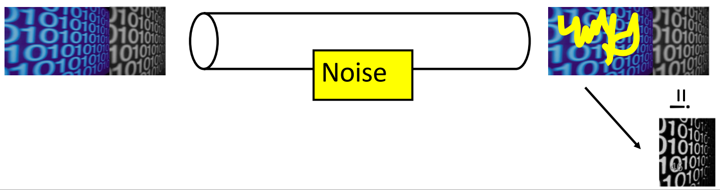
||

45

Forward Error Correction (FEC)

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

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

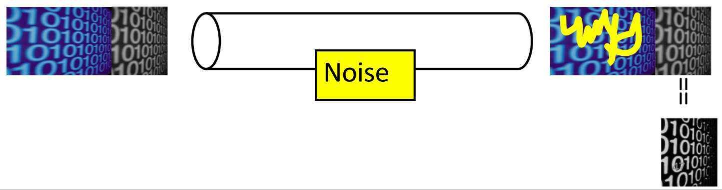


46

Forward Error Correction (FEC)

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

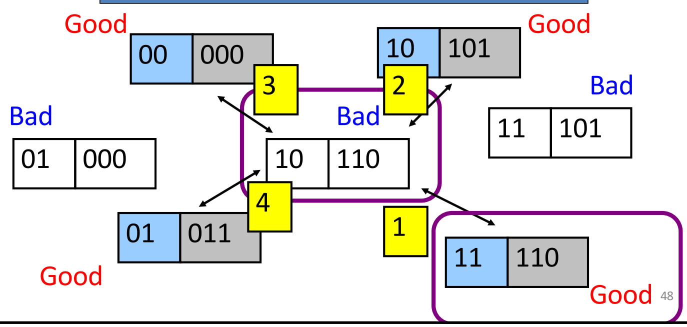
Receiver:
receive(X1Y1);
Y2=generateCheckBit(X1);
if (Y1 != Y2) FIXERROR(X1Y1);
else NOERROR



47

Basic Idea of Forward Error Correction

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



48

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.

FEC: Kurose&Ross P618 §7.3.3
No useful Peterson&Davie reference

49

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 Back-off
  - Spanning Tree (Dijkstra)
- General knowledge
  - Random numbers are important and hard

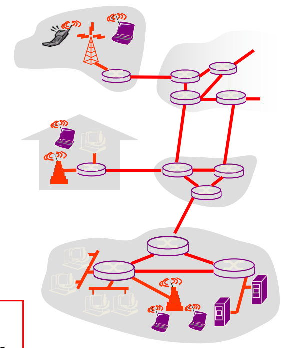
50

50

Link Layer: Introduction

Some reminder-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

51

51

Link Layer (Channel) Services - 1/2

- *framing, physical addressing:*
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, destination
    - This is **not** an IP address!
- *reliable delivery between adjacent nodes*
  - we revisit this again in the Transport Topic
  - seldom used on low bit-error link (fiber, some twisted pair)
  - wireless links: high error rates

52

52

Link Layer (Channel) Services – 2/2

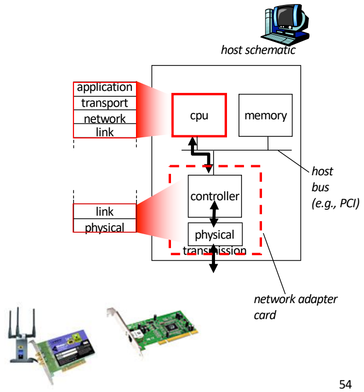
- *flow control:*
  - pacing between adjacent sending and receiving nodes
- *error control:*
  - *error detection*:
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
  - error correction:
  - receiver identifies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

53

53

Where is the link layer implemented?

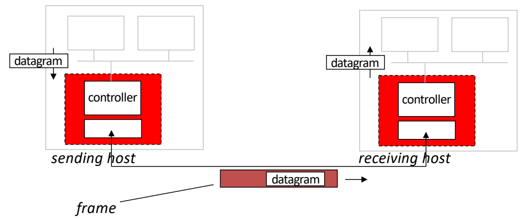
- in each and every host
- link layer implemented in "adaptor" (aka *network interface card* NIC)
  - Ethernet card, PCMCI card, 802.11 card
  - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



54

54

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

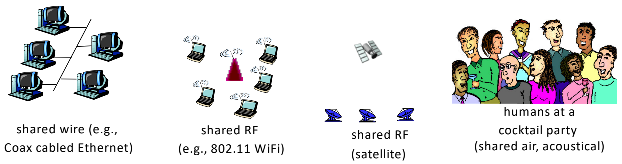
55

55

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



56

56

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

57

57

Ideal Multiple Access Protocol

<u>Broadcast channel of rate *R* bps</u>

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

58

58

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

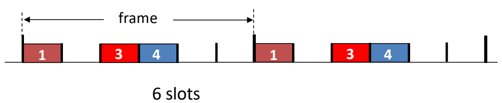
59

59

Channel Partitioning MAC protocols: TDMA
(we discussed 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



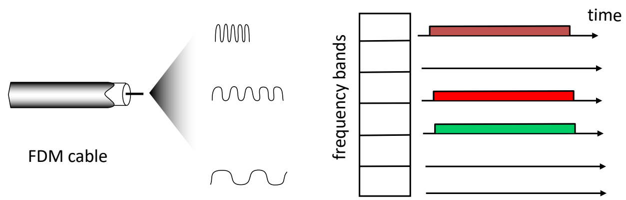
60

60

Channel Partitioning MAC protocols: FDMA
(we discussed 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



61

61

"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!

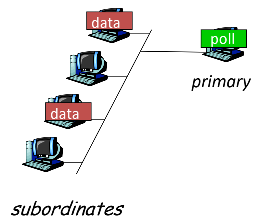
62

62

"Taking Turns" MAC protocols

Polling:

- Primary node "invites" subordinates nodes to transmit in turn
- typically used with simpler subordinate devices
- concerns:
  - polling overhead
  - latency
  - single point of failure (primary)



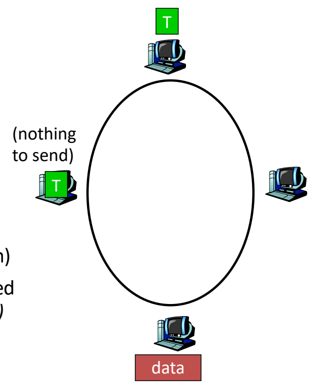
63

63

"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)*

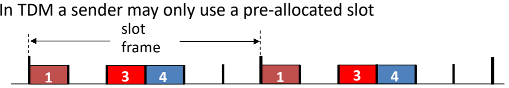


64

64

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….

65

65

"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:

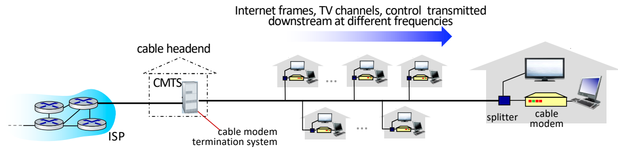
look for best of both worlds!

Recall.....

66

66

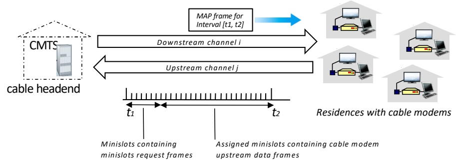
Cable access network: FDM, TDM *and* random access!



- multiple downstream (broadcast) FDM channels: up to 1.6 Gbps/channel
  - single CMTS transmits into channels
- multiple upstream channels (up to 1 Gbps/channel)
  - multiple access: all users contend (random access) for certain upstream channel time slots; others assigned TDM

67

Cable access network:



DOCSIS: data over cable service interface specification

- FDM over upstream, downstream frequency channels
- TDM upstream: some slots assigned, some have contention
  - downstream MAP frame: assigns upstream slots
  - request for upstream slots (and data) transmitted random access (binary backoff) in selected slots

68

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 ⇒ 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)

69

69

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

70

70

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

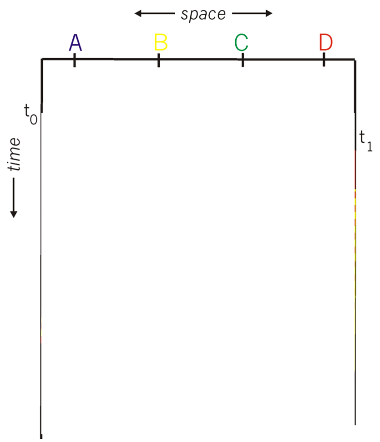
71

71

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?

72

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)

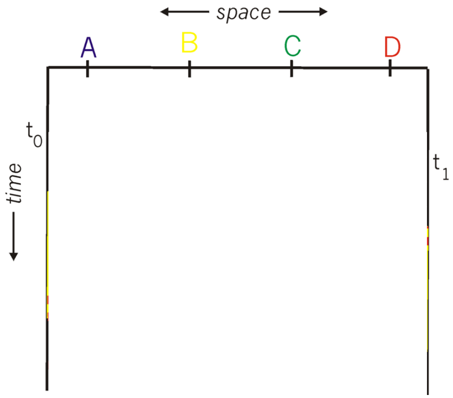
73

73

CSMA/CD Collision Detection

B and D can tell that collision occurred.

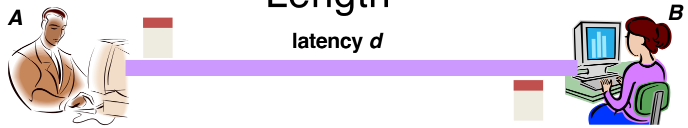
Note: for this to work, need restrictions on minimum frame size and maximum distance. Why?



74

74

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***

75

75

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)

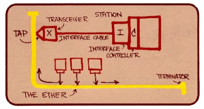
$$E \sim \frac{\frac{p}{b}}{\frac{p}{b} + Kd}$$

- Note:
  - For large packets, small distances, E ~ 1
  - As bandwidth increases, E decreases
  - That is why high-speed LANs are all switched aka packets are sent via a switch - (any d is bad)

76

76

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^{th} collision, choose K randomly from {0, …, 2^m-1}
  - … and wait for K*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)**

78

78

Benefits of Ethernet

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

79

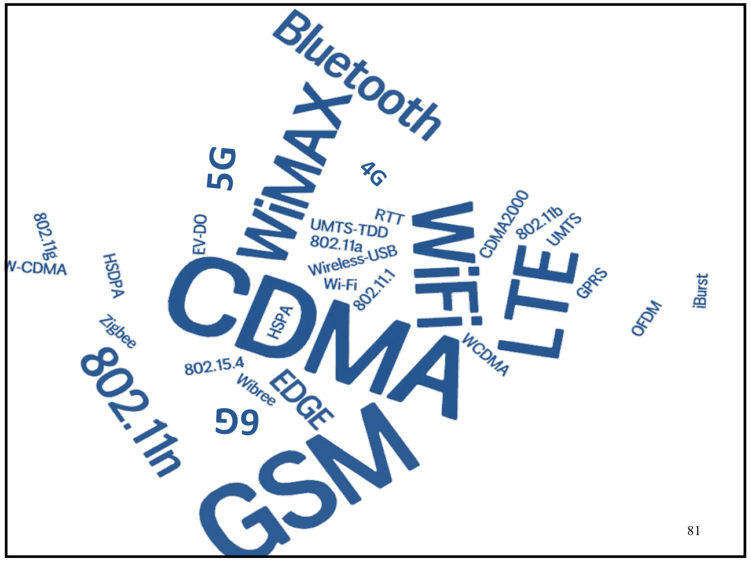
79

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

80

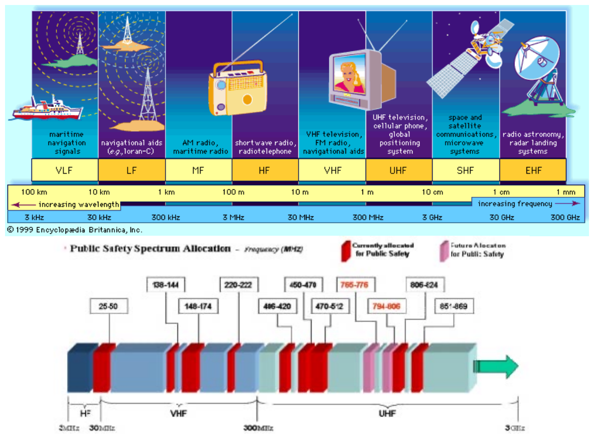
80



81

81

The Wireless Spectrum



82

82

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 – $\lambda = \frac{v}{f}$,)

83

83

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): (some examples)
  - 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)
  - ac: 2.4/5Ghz, 433-1300Mbps (improved coding 256-QAM)
  - ad: 60Ghz, 7Gbps
  - af: 54/790Mhz, 26-35Mbps (TV whitespace)
- IEEE 802.15 – lower power wireless:
  - 802.15.1: 2.4Ghz, 2.1 Mbps (Bluetooth)
  - 802.15.4: 2.4Ghz, 250 Kbps (Sensor Networks)

84

84

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?

85

85

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

86

86

IEEE 802.11 Wireless LAN

IEEE 802.11 standard	Year	Max data rate	Range	Frequency
802.11b	1999	11 Mbps	30 m	2.4 Ghz
802.11g	2003	54 Mbps	30m	2.4 Ghz
802.11n (WiFi 4)	2009	600	70m	2.4, 5 Ghz
802.11ac (WiFi 5)	2013	3.47Gpbs	70m	5 Ghz
802.11ax (WiFi 6)	2020 (exp.)	14 Gbps	70m	2.4, 5 Ghz
802.11af	2014	35 – 560 Mbps	1 Km	unused TV bands (54-790 MHz)
802.11ah	2017	347Mbps	1 Km	900 Mhz

- all use CSMA/CA for multiple access, and have base-station and ad-hoc network versions

87

802.11 Architecture

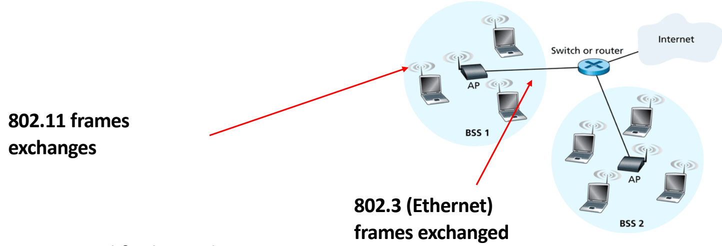


Figure 6.7 ♦ IEEE 802.11 LAN architecture

- 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

88

88

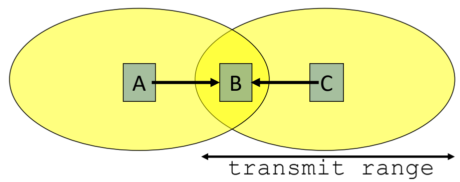
Wireless Multiple Access Technique?

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

89

89

Hidden Terminals

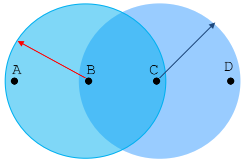


- A and C can both send to B but can't hear each other
  – A is a *hidden terminal* for C and vice versa
- Carrier Sense will be ineffective

90

90

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.

91

91

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

92

92

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

93

93

IEEE 802.11 MAC Protocol: CSMA/CA

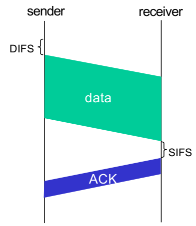
802.11 sender

1 if sense channel idle for **DIFS** then
   transmit entire frame (no CD)

2 if sense channel busy then
   start random backoff time
   timer counts down while channel idle
   transmit when timer expires
   if no ACK, increase random backoff interval, repeat 2

802.11 receiver

if frame received OK
   return ACK after **SIFS** (ACK needed due to hidden terminal problem)



94

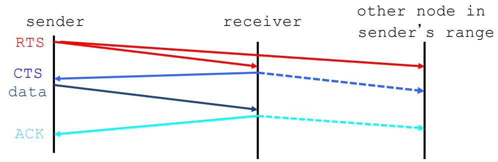
Avoiding collisions

idea: sender "reserves" channel use for data frames using small reservation packets

- sender first transmits *small* request-to-send (RTS) packet to BS using CSMA
  - RTSs may still collide with each other (but they're short)
- BS broadcasts clear-to-send CTS in response to RTS
- CTS heard by all nodes
  - sender transmits data frame
  - other stations defer transmissions

95

CSMA/CA – and in this case RTS/CTS

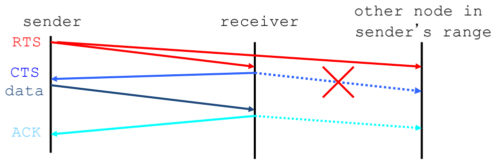


- Before every data transmission
  - Sender sends a Request to Send (RTS) frame containing the length of the transmission
  - Receiver respond with a Clear to Send (CTS) frame
  - Sender sends data
  - Receiver sends an ACK; now another sender can send data
- When sender doesn't get a CTS back, it assumes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

96

96

CSMA/CA, con't



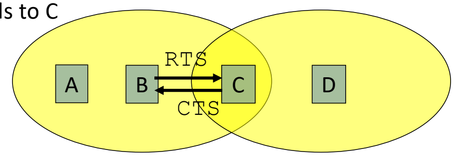
- If other nodes hear RTS, but not CTS: send
  - Presumably, destination for first sender is out of node's range …
  - … Can cause problems when a CTS is lost
- When you hear a CTS, you keep quiet until scheduled transmission is over (hear ACK)

97

97

RTS / CTS Protocols (CSMA/CA)

B sends to C



Overcome hidden terminal problems with contention-free protocol

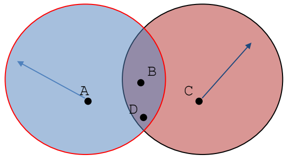
1. B sends to C Request To Send (RTS)
2. A hears RTS and defers (to allow C to answer)
3. C replies to B with Clear To Send (CTS)
4. D hears CTS and defers to allow the data
5. B sends to C

98

98

Preventing Collisions Altogether

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



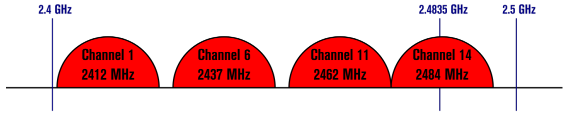
  - Now A and C can send without any interference!
- Most cards have only 1 transceiver
  - **Not Full Duplex: Cannot send and receive at the same time**
  - Aggregate Network throughput doubles

99

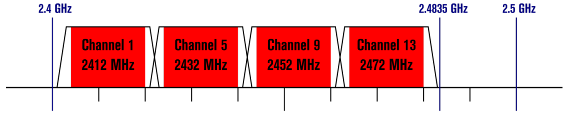
99

Non-Overlapping Channels for 2.4 GHz WLAN

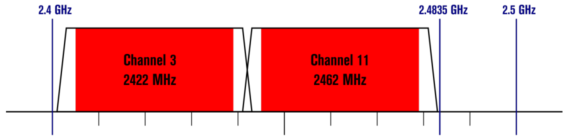
802.11b (DSSS) channel width 22 MHz



802.11g/n (OFDM) 20 MHz ch. width – 16.25 MHz used by sub-carriers

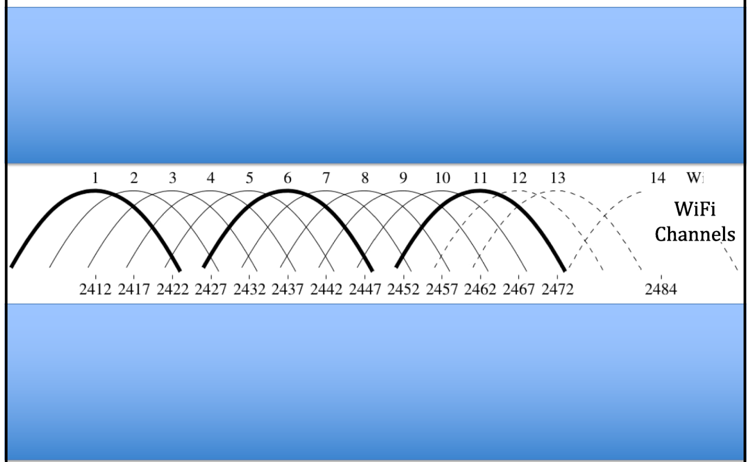


802.11n (OFDM) 40 MHz ch. width – 33.75 MHz used by sub-carriers



100

100



101

101

Wifi has been evolving!

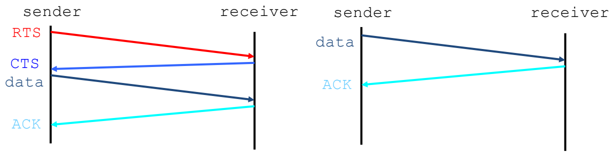
Using dual band (2.4GHz + 5GHz), multiple channels, MIMO, Meshing WiFi

Outside this introduction but the state of the art is very fast and very flexible

102

102

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(collision)* is low
- good for when net is small and not *weird*
  eg no hidden/exposed terminals

103

103

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

CD Collision Detect

wired – listen and talk

1. Listen for others
2. Busy? goto 1.
3. Send message (and listen)
4. Collision?
   a. JAM
   b. increase your BEB
   c. sleep
   d. goto 1.

CA Collision Avoidance

wireless – talk OR listen

1. Listen for others
2. Busy? goto 1.
3. Send message
4. Wait for ACK (*MAC ACK*)
5. Got No ACK from MAC?
   a. increase your BEB
   b. sleep
   c. goto 1.

104

104

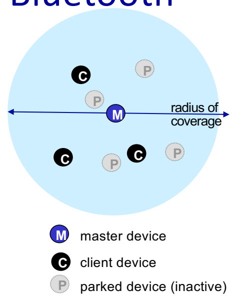
802.11: advanced capabilities

power management
- node-to-AP: "I am going to sleep until next beacon frame"
  - AP knows not to transmit frames to this node
  - node wakes up before next beacon frame
- beacon frame: contains list of mobiles with AP-to-mobile frames waiting to be sent
  - node will stay awake if AP-to-mobile frames to be sent; otherwise sleep again until next beacon frame

105

Personal area networks: Bluetooth

- TDM, 625 μsec sec. slot
- FDM: sender uses 79 frequency channels in known, pseudo-random order slot-to-slot (spread spectrum)
  - other devices/equipment not in piconet only interfere in some slots
- parked mode: clients can "go to sleep" (park) and later wakeup (to preserve battery)
- bootstrapping: nodes self-assemble (plug and play) into piconet



106

Summary of MAC protocols

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

107

107

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

```
awm22@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:fefe: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
```

108

108

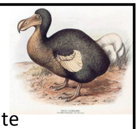
LAN Address (more)

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- analogy:
  - (a) MAC address: like a National Insurance Number
  - (b) IP address: like a 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

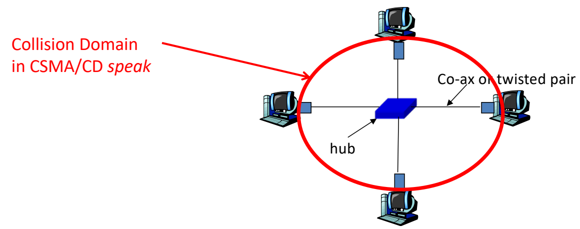
109

109

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



110

110

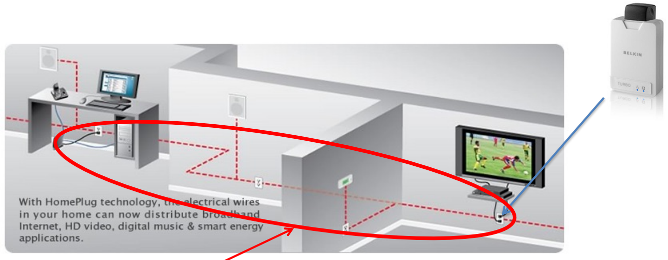
CSMA in our home

Home Plug Powerline Networking….

111

111

Home Plug and similar Powerline Networking….



Collision Domain
in CSMA *speak*

To secure network traffic on a specific HomePlug network, each set of adapters use an encryption key common to a specific HomePlug network

112

112

Switch (example: Ethernet Switch)

- 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

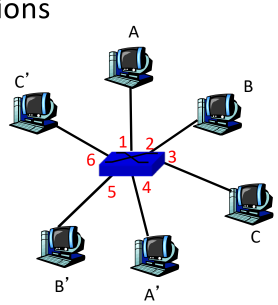
If you want to connect different physical media
(optical – copper – coax – wireless - ….)
you **NEED** a switch.
Why? (Because each link, each media access protocol is specialised)

113

113

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



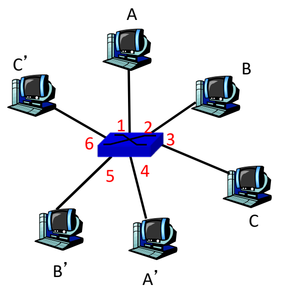
switch with six interfaces (1,2,3,4,5,6)

114

114

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?



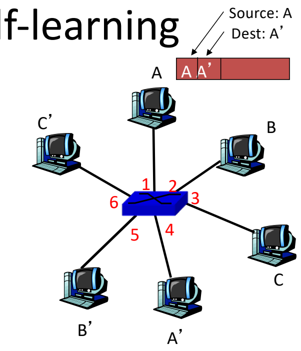
switch with six interfaces (1,2,3,4,5,6)

115

115

Switch: self-learning

- 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)

116

116

Switch: frame filtering/forwarding

When frame received:

1. record link associated with sending host
2. index switch table using MAC dest address
3. **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

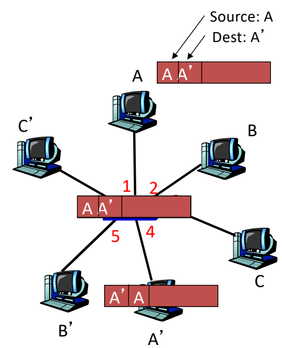
forward on all but the interface on which the frame arrived

117

117

Self-learning, forwarding: example

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



MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

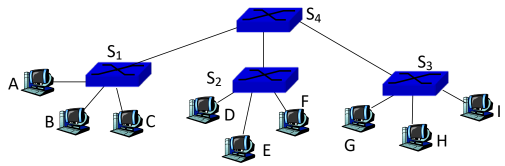
Switch table (initially empty)

118

118

Interconnecting switches

- switches can be connected together



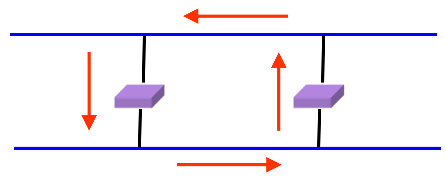
- Q: sending from A to G - how does S_1 know to forward frame destined to F via S_4 and S_3?
- A: self learning! (works exactly the same as in single-switch case – flood/forward/drop)

119

119

Flooding Can Lead to Loops

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

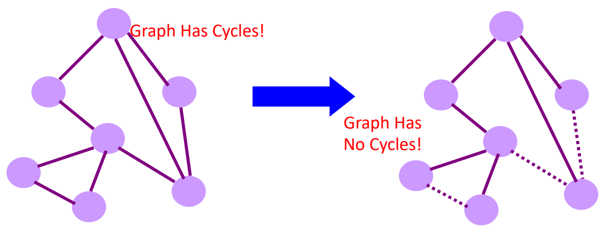


120

120

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



121

121

What Do We Know?

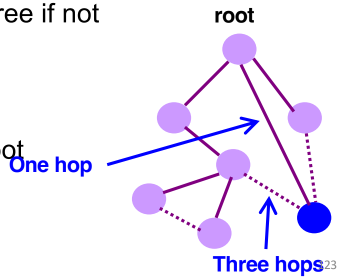
- *"Spanning tree algorithm is an algorithm to create a tree out of a graph that includes all nodes with a minimum number of edges connecting to vertices."*
- Shortest paths to (or from) a node form a tree
- So, algorithm has two aspects :
  - Pick a root
  - Compute shortest paths to it
- Only keep the links on shortest-path

122

122

Constructing a Spanning Tree

- Switches need to elect a root
  - The switch w/ smallest identifier (MAC addr)
- Each switch determines if each interface is on the shortest path from the root
  - Excludes it from the tree if not
- Messages (Y, d, X)
  - From node X
  - Proposing Y as the root
  - And the distance is d



123

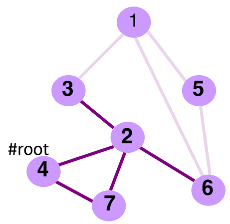
Steps in Spanning Tree Algorithm

- Initially, each switch proposes itself as the root
  - Switch sends a message out every interface
  - … proposing itself as the root with distance 0
  - Example: switch X announces (X, 0, X)
- Switches update their view of the root
  - Upon receiving message (Y, d, Z) from Z, check Y's id
  - If new id smaller, start viewing that switch as root
- Switches compute their distance from the root
  - Add 1 to the distance received from a neighbor
  - Identify interfaces not on shortest path to the root
  - … and exclude them from the spanning tree
- If root or shortest distance to it changed, "flood" updated message (Y, d+1, X)

124

Example From Switch #4's Viewpoint

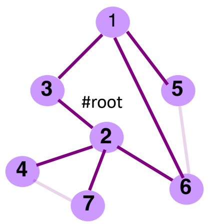
- Switch #4 thinks it is the root
  - Sends (4, 0, 4) message to 2 and 7
- Then, switch #4 hears from #2
  - Receives (2, 0, 2) message from 2
  - … and thinks that #2 is the root
  - And realizes it is just one hop away
- Then, switch #4 hears from #7
  - Receives (2, 1, 7) from 7
  - And realizes this is a longer path
  - So, prefers its own one-hop path
  - And removes 4-7 link from the tree



125

Example From Switch #4's Viewpoint

- Switch #2 hears about switch #1
  - Switch 2 hears (1, 1, 3) from 3
  - Switch 2 starts treating 1 as root
  - And sends (1, 2, 2) to neighbors
- Switch #4 hears from switch #2
  - Switch 4 starts treating 1 as root
  - And sends (1, 3, 4) to neighbors
- Switch #4 hears from switch #7
  - Switch 4 receives (1, 3, 7) from 7
  - And realizes this is a longer path
  - So, prefers its own three-hop path
  - And removes 4-7 link from the tree



126

Robust Spanning Tree Algorithm

- Algorithm must react to failures
  - Failure of the root node
    - Need to elect a new root, with the next lowest identifier
  - Failure of other switches and links
    - Need to recompute the spanning tree
- Root switch continues sending messages
  - Periodically reannouncing itself as the root (1, 0, 1)
  - Other switches continue forwarding messages
- Detecting failures through timeout (soft state)
  - If no word from root, times out and claims to be the root
  - Delay in reestablishing spanning tree is ***major problem***
  - Work on rapid spanning tree algorithms…

Given a switch-tree of a given size, link length, speed of computation, …

How long does a failure take to rectify?

127

127

Weirder "Data Link Layer" Networks

VLAN

Application
Transport
Network
Data Link (L2)
Data Link (L2)
Physical

VPN

Application
Transport
Network
Transport
Network
Data Link (L2)
Physical

Datacenter

"so you think your LAN has a lot of computers…."

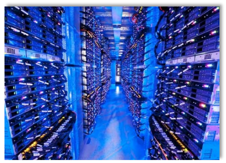
128

128

Datacenter networks

10's to 100's of thousands of hosts, often closely coupled, in close proximity:
- e-business (e.g. Amazon)
- content-servers (e.g., YouTube, Akamai, Apple, Microsoft)
- search engines, data mining (e.g., Google)

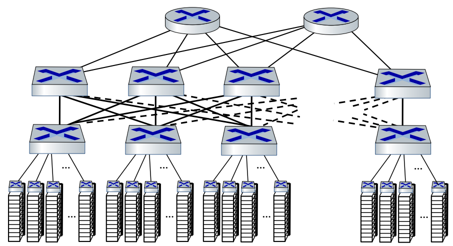
challenges:
- multiple applications, each serving massive numbers of clients
- reliability
- managing/balancing load, avoiding processing, networking, data bottlenecks



Inside a 40-ft Microsoft container, Chicago data center

129

Datacenter networks: network elements



Border routers
- connections outside datacenter

Tier-1 switches
- connecting to ~16 T-2s below

Tier-2 switches
- connecting to ~16 TORs below

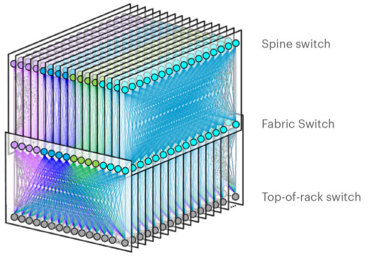
Top of Rack (TOR) switch
- one per rack
- 40-100Gbps Ethernet to blades

Server racks
- 20- 40 server blades: hosts

130

Datacenter networks: network elements

Facebook F16 data center network topology:

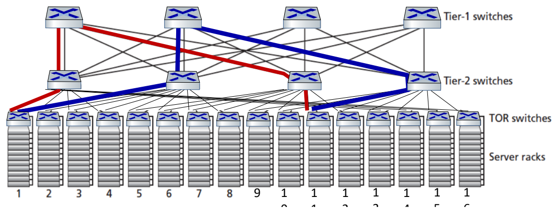


https://engineering.fb.com/data-center-engineering/f16-minipack/ (posted 3/2019)

131

Datacenter networks: multipath

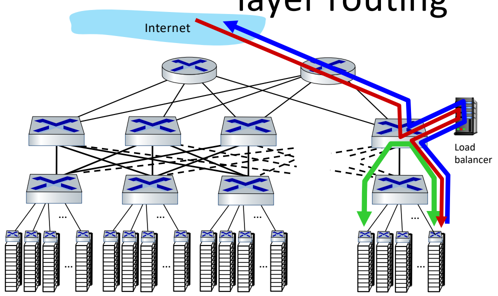
- rich interconnection among switches, racks:
  - increased throughput between racks (multiple routing paths possible)
  - increased reliability via redundancy



two disjoint paths highlighted between racks 1 and 11

132

Datacenter networks: application-layer routing



load balancer: application-layer routing

- receives external client requests
- directs workload within data center
- returns results to external client (hiding data center internals from client)

133

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

134

134

Topic 4: Network Layer

Our goals:

- understand principles behind network layer services:
  - network layer service models
  - forwarding versus routing (versus switching)
  - how a router works
  - routing (path selection)
  - IPv6

For the most part, the Internet is our example – again.

1

1

Recall: Network layer is responsible for *GLOBAL* delivery

Name: a *something*

Address: Where is a *something*

Routing: How do I get to the *something*

Forwarding: What path do I take next to get to the *something*

2

2

Addressing (at a conceptual level)

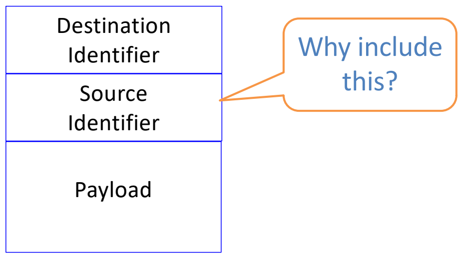
- Assume all hosts have unique IDs
- No particular structure to those IDs
- Later in topic I will talk about real IP addressing
- Do I route on location or identifier?
- If a host moves, should its address change?
  - If not, how can you build scalable Internet?
  - If so, then what good is an address for identification?

3 3

3

Packets (at a conceptual level)

- Assume packet headers contain:
  - Source ID, Destination ID, and perhaps other information

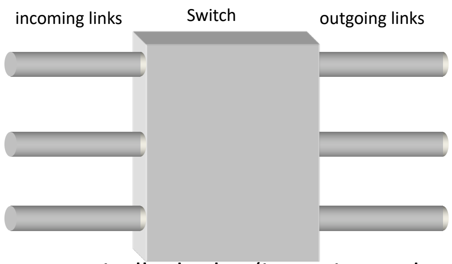


4

4

Switches/Routers

- Multiple ports (attached to other switches or hosts)



- Ports are typically duplex (incoming and outgoing)

5

5

A Variety of *(Internet Protocol-based)* Networks

- ISPs: carriers
  - Backbone
  - Edge
  - Border (to other ISPs)
- Enterprises: companies, universities
  - Core
  - Edge
  - Border (to outside)
- Datacenters: massive collections of machines
  - Top-of-Rack
  - Aggregation and Core
  - Border (to outside)

6

6

A Variety of *(Internet Protocol-based)* Routers

- ISPs: carriers
  - Backbone
  - Edge
  - Border (to other ISPs)

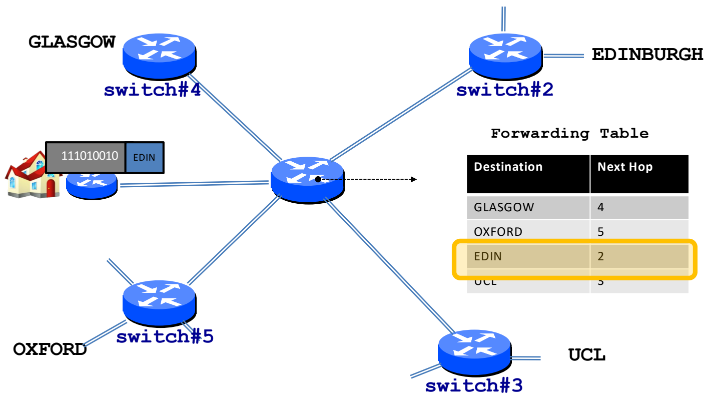
- Enterprises: companies, universities
  - Core
  - Edge
  - Border (to outside)

- Datacenters: massive collections of machines
  - Top-of-Rack
  - Aggregation and Core
  - Border (to outside)

7

7

Switches forward packets



8

8

Forwarding Decisions

- When packet arrives..
  - Must decide which outgoing port to use
  - In single transmission time
  - Forwarding decisions must be ***simple***
- Routing state dictates where to forward packets
  - Assume decisions are **deterministic**
- *Global routing state* is the collection of routing state in each of the routers
  - Will focus on where this routing state comes from
  - But first, a few preliminaries….

9

9

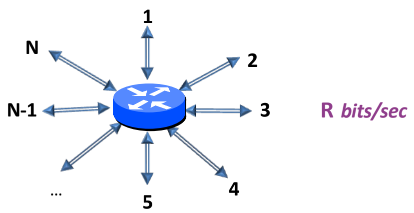
Forwarding vs Routing

- Forwarding: "data plane"
  - Directing a data packet to an outgoing link
  - Individual router using routing state
- Routing: "control plane"
  - Computing paths the packets will follow
  - Routers talking amongst themselves
  - Jointly creating the routing state
- Two very different timescales….

10

10

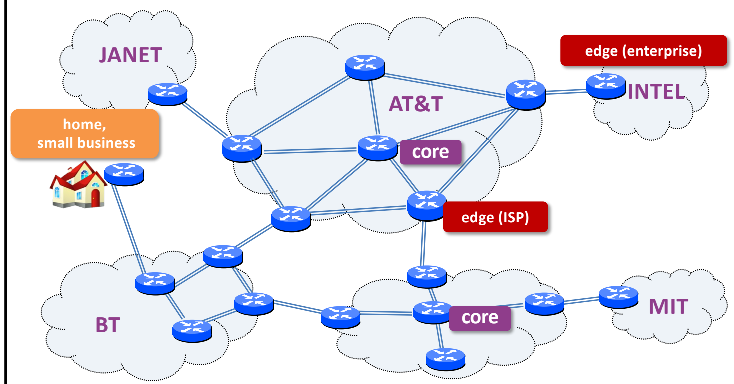
Router definitions



- **N = number of external router "ports"**
- **R = speed ("line rate") of a port**
- **Router capacity = N x R**

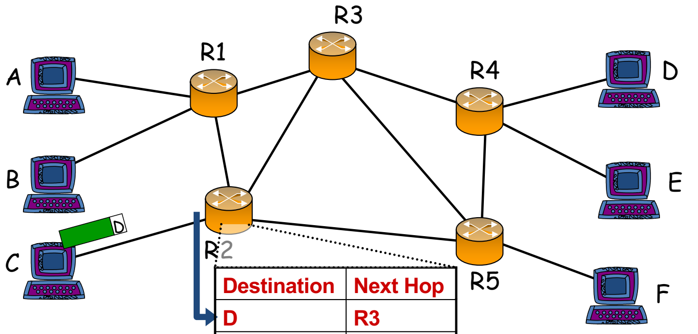
11

Networks and routers



12

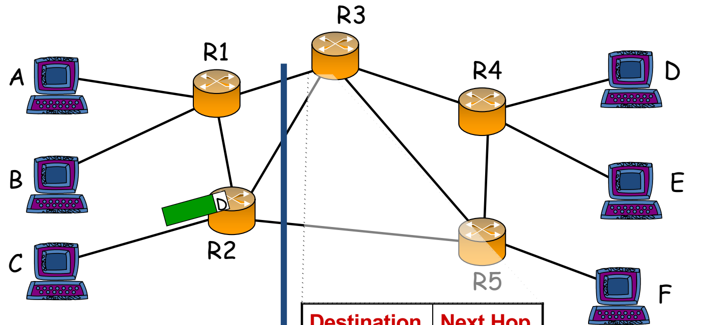
Basic Operation of Router



Destination	Next Hop
D	R3
E	R3
F	R5

13

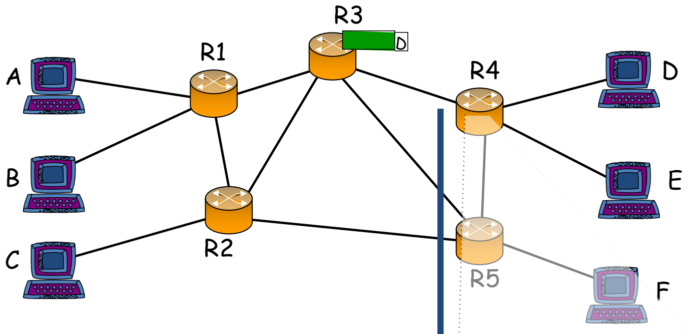
Basic Operation of Router



Destination	Next Hop
D	R4
E	R4
F	R5

14

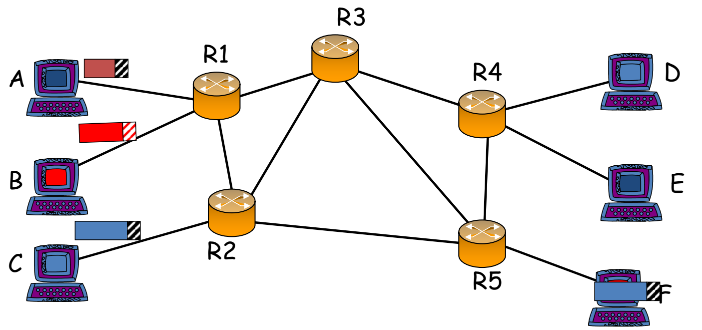
Basic Operation of Router



Destination	Next Hop
D	Port D
E	Port E
F	R5

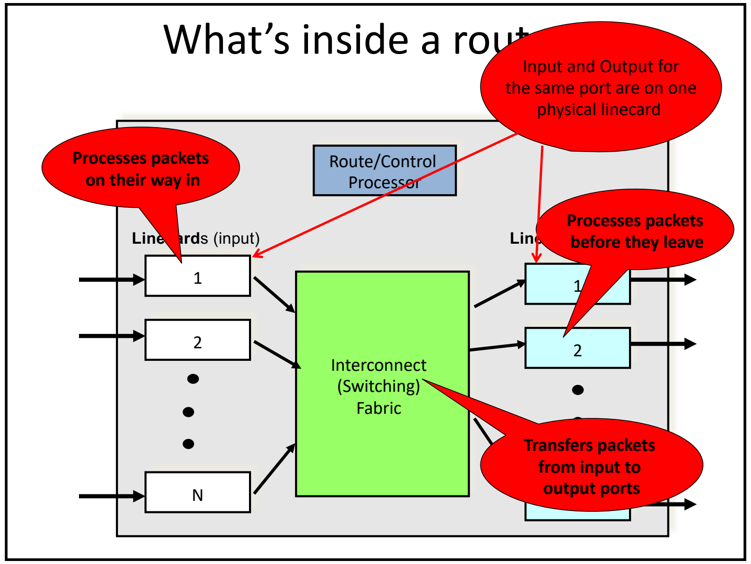
15

What does a router do?

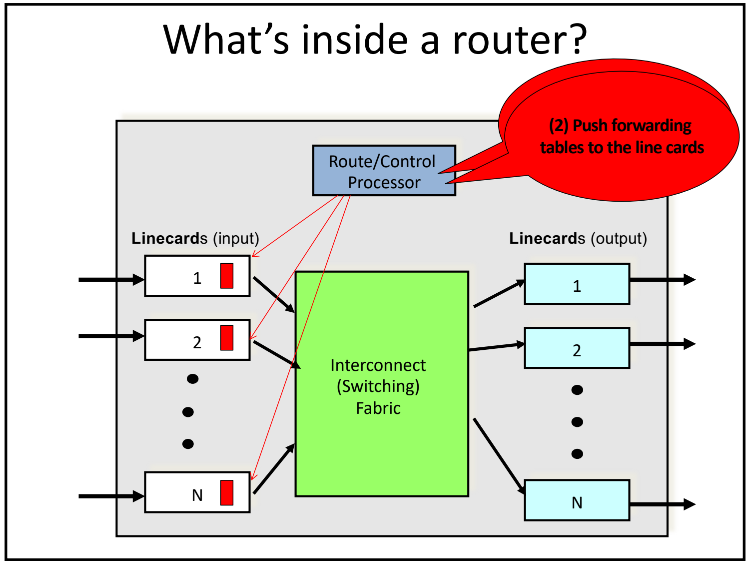


1. Every router performs a per-packet lookup for every packet
2. Each router performs a lookup in it's local lookup table
3. Each router performs lookups (ENTIRELY) independently of every other router

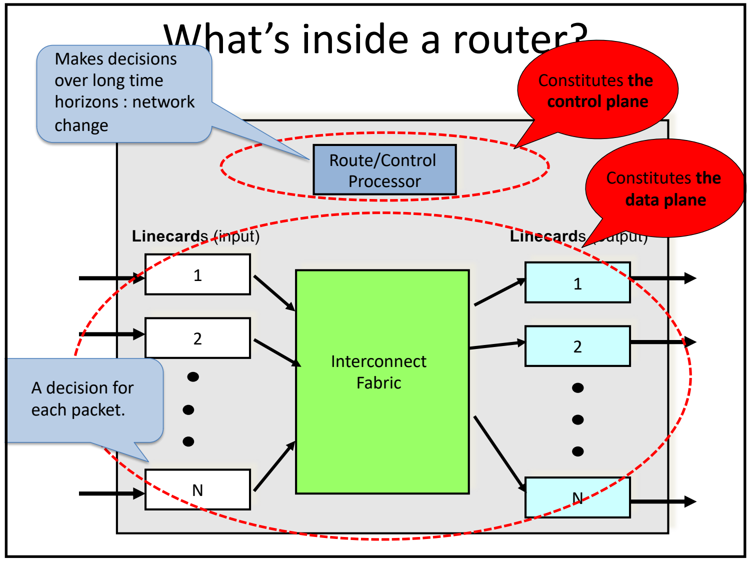
17


What's inside a rout
Input and Output for the same port are on one physical linecard
Processes packets on their way in
Route/Control Processor
Processes packets before they leave
ards (input)
Line
Lin
1
2
N
1
2
Interconnect (Switching) Fabric
Transfers packets from input to output ports

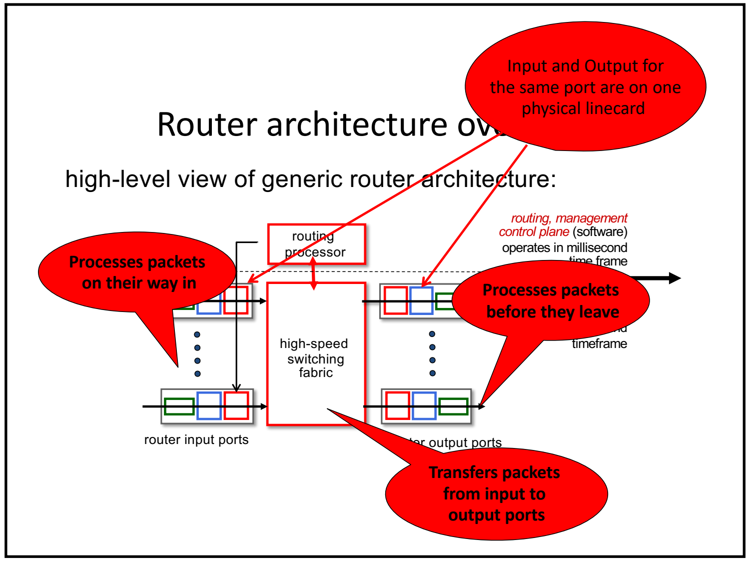
18


What's inside a router?
(2) Push forwarding tables to the line cards
Route/Control Processor
Linecards (input)
Linecards (output)
1
2
N
Interconnect (Switching) Fabric
1
2
N

19

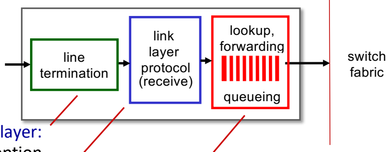

What's inside a router?
Makes decisions over long time horizons : network change
Constitutes the control plane
Route/Control Processor
Constitutes the data plane
Linecards (input)
Linecards
utput)
1
2
N
Interconnect Fabric
1
2
N
A decision for each packet.

20


Router architecture ov
Input and Output for the same port are on one physical linecard
high-level view of generic router architecture:
routing, management control plane (software) operates in millisecond
routing processor
Processes packets on their way in
Processes packets before they leave
high-speed switching fabric
timeframe
router input ports
Transfers packets from input to output ports

21

Input port functions



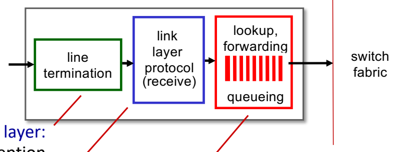
physical layer:
bit-level reception

link layer:
e.g., Ethernet
(Topic3)

decentralized switching:
- using header field values, lookup output port using forwarding table in input port memory *("match plus action")*
- goal: complete input port processing at 'line speed'
- input port queuing: if datagrams arrive faster than forwarding rate into switch fabric

22

Input port functions



physical layer:
bit-level reception

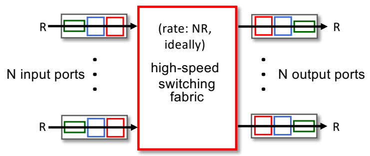
link layer:
e.g., Ethernet
(chapter 6)

decentralized switching:
- using header field values, lookup output port using forwarding table in input port memory *("match plus action")*
- destination-based forwarding: forward based only on destination IP address (traditional)
- generalized forwarding: forward based on any set of header field values

23

Switching fabrics

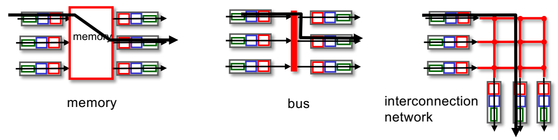
- transfer packet from input link to appropriate output link
- switching rate: rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable



24

Switching fabrics

- transfer packet from input link to appropriate output link
- switching rate: rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable
- three major types of switching fabrics:

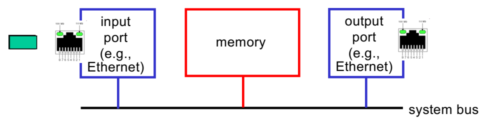


25

Switching via memory

first generation routers:

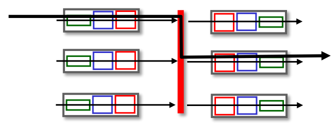
- traditional computers with switching under direct control of CPU
- packet copied to system's memory
- speed limited by memory bandwidth (2 bus crossings per datagram)



26

Switching via a bus

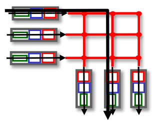
- datagram from input port memory to output port memory via a shared bus
- *bus contention:* switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access routers



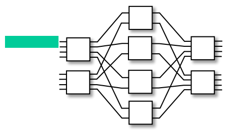
27

Switching via interconnection network

- Crossbar, Clos networks, other interconnection nets initially developed to connect processors in multiprocessor
- multistage switch: *nxn* switch from multiple stages of smaller switches
- exploiting parallelism:
  - fragment datagram into fixed length cells on entry
  - switch cells through the fabric, reassemble datagram at exit



3x3 crossbar

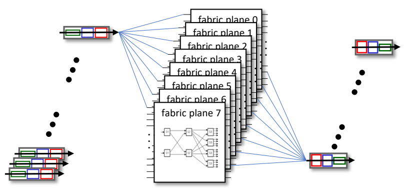


8x8 multistage switch
built from smaller-sized switches

28

Switching via interconnection network

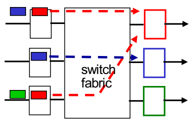
- scaling, using multiple switching "planes" in parallel:
  - speedup, scaleup via parallelism
- Cisco CRS router:
  - basic unit: 8 switching planes
  - each plane: 3-stage interconnection network
  - up to 100's Tbps switching capacity



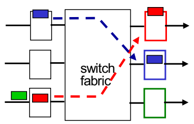
29

Input port queuing

- If switch fabric slower than input ports combined -> queueing may occur at input queues
  - queueing delay and loss due to input buffer overflow!
- Head-of-the-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward



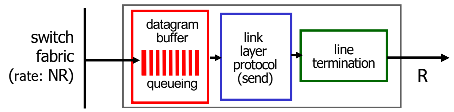
output port contention: only one red datagram can be transferred. lower red packet is *blocked*



one packet time later: green packet experiences HOL blocking

30

Output port queuing



This is a really important slide

- *Buffering* required when datagrams arrive from fabric faster than link transmission rate. *Drop policy:* which datagrams to drop if no free buffers?

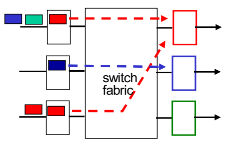
Datagrams can be lost due to congestion, lack of buffers

- *Scheduling discipline* chooses among queued datagrams for transmission

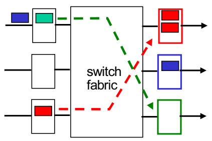
Priority scheduling – who gets best performance, network neutrality

31

Output port queuing



at *t*, packets more from input to output



one packet time later

- buffering when arrival rate via switch exceeds output line speed
- *queueing (delay) and loss due to output port buffer overflow!*

32

How much buffering? (related material in Topic 5)

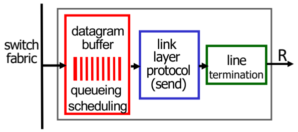
- RFC 3439 rule of thumb: average buffering equal to "typical" RTT (say 250 msec) times link capacity C
  - e.g., C = 10 Gbps link: 2.5 Gbit buffer
- more recent recommendation: with *N* flows, buffering equal to

$$\frac{RTT \cdot C}{\sqrt{N}}$$

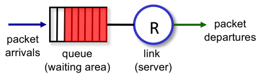
- but *too* much buffering can increase delays (particularly in home routers)
  - long RTTs: poor performance for realtime apps, sluggish TCP response
  - recall delay-based congestion control: "keep bottleneck link just full enough (busy) but no fuller"

33

Buffer Management



Abstraction: queue



buffer management:

- drop: which packet to add, drop when buffers are full
  - tail drop: drop arriving packet
  - priority: drop/remove on priority basis
- marking: which packets to mark to signal congestion (ECN, RED)

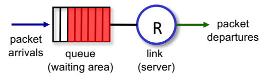
34

Packet Scheduling: FCFS

packet scheduling: deciding which packet to send next on link

- first come, first served
- priority
- round robin
- weighted fair queueing

Abstraction: queue



FCFS: packets transmitted in order of arrival to output port

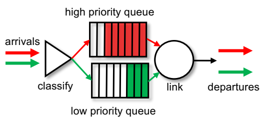
- also known as: First-in-first-out (FIFO)
- real world examples?

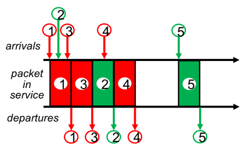
35

Scheduling policies: priority

Priority scheduling:

- arriving traffic classified, queued by class
  - any header fields can be used for classification
- send packet from highest priority queue that has buffered packets
  - FCFS within priority class



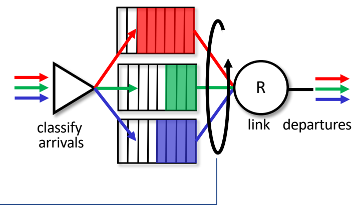


36

Scheduling policies: round robin

Round Robin (RR) scheduling:

- arriving traffic classified, queued by class
  - any header fields can be used for classification
- server cyclically, repeatedly scans class queues, sending one complete packet from each class (if available) in turn



37

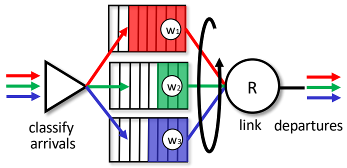
Scheduling policies: weighted fair queueing

Weighted Fair Queuing (WFQ):

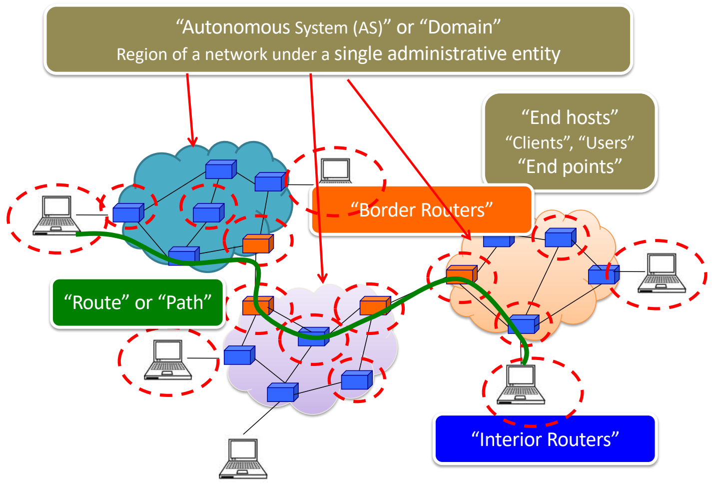
- generalized Round Robin
- each class, *i*, has weight, w_i, and gets weighted amount of service in each cycle:

$$\frac{w_i}{\Sigma_j w_j}$$

- minimum bandwidth guarantee (per-traffic-class)

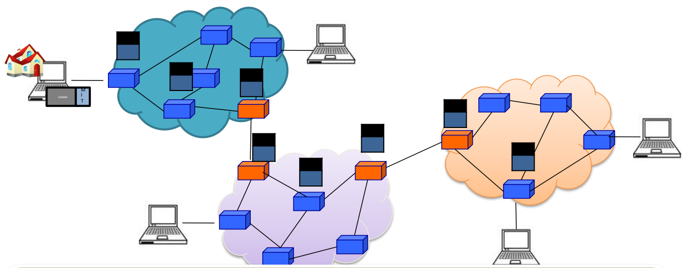


38



39

Context and Terminology

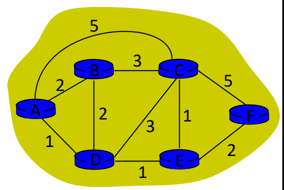


Internet routing protocols are responsible for constructing and updating the forwarding tables at routers

40

Routing Protocols

- Routing protocols implement the core function of a network
  - Establish paths between nodes
  - Part of the network's "control plane"
- Network modeled as a graph
  - Routers are graph vertices
  - Links are edges
  - Edges have an associated "cost"
    - e.g., distance, loss



- Goal: compute a "good" path from source to destination
  - "good" usually means the shortest (least cost) path

41

Internet Routing

- Internet Routing works at two levels
- Each AS runs an intra-domain routing protocol that establishes routes within its domain
  - (AS -- region of network under a single administrative entity)
  - Link State, e.g., Open Shortest Path First (OSPF)
  - Distance Vector, e.g., Routing Information Protocol (RIP)
- ASes participate in an inter-domain routing protocol that establishes routes between domains
  - Path Vector, e.g., Border Gateway Protocol (BGP)

42

42

Addressing (to date)

- a reminder -

Recall each host has a unique ID (address)

- No particular structure to those IDs
  (e.g. *Ethernet*)
- IP addressing – in contrast – has implicit structure

43

43

Outline

- Popular Routing Algorithms:
  - Link State Routing
  - Distance Vector Algorithm
- Routing: goals and metrics

44

44

Link-State Routing

Examples:

Open Shortest Path First (**OSPF**) or
Intermediate System to Intermediate System
(written as **IS-IS**/**ISIS** and pronounced eye-esss-eye-esss)

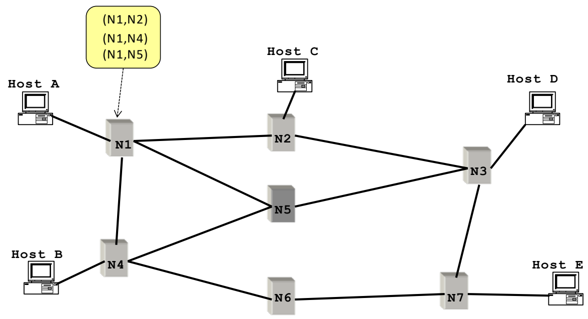
The two common Intradomain routing or
interior gateway protocols (IGP)

45

45

Link State Routing

- Each node maintains its local "link state" (LS)
  - i.e., a list of its directly attached links and their costs

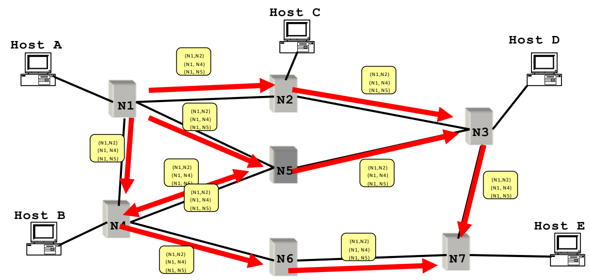


46

46

Link State Routing

- Each node maintains its local "link state" (LS)
- Each node floods its local link state
  - on receiving a new LS message, a router forwards the message to all its neighbors other than the one it received the message from

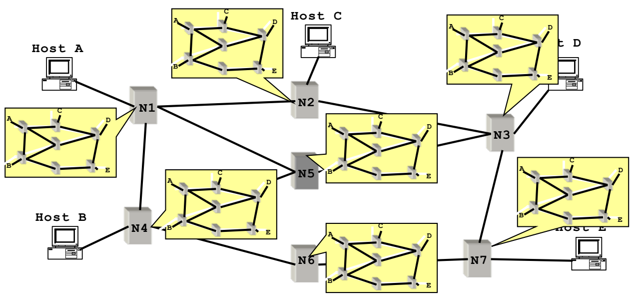


47

47

Link State Routing

- Each node maintains its local "link state" (LS)
- Each node floods its local link state
- Hence, each node learns the entire network topology
  - Can use Dijkstra's to compute the shortest paths between nodes



48

48

Dijkstra's Shortest Path Algorithm

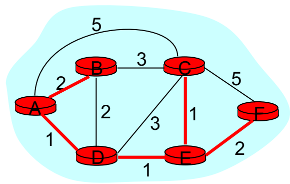
- INPUT:
  - Network topology (graph), with link costs
- OUTPUT:
  - Least cost paths from one node to all other nodes
- Iterative: after *k* iterations, a node knows the least cost path to its *k* closest neighbors
- This is covered in Algorithms

49

49

The Forwarding Table

- Running Dijkstra at node A gives the shortest path from A to all destinations
- We then construct the *forwarding table*



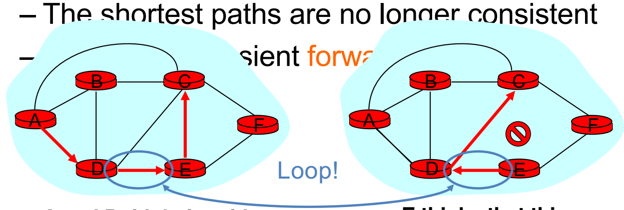
Destination	Link
B	(A,B)
C	(A,D)
D	(A,D)
E	(A,D)
F	(A,D)

Issue #1: Scalability

- How many messages needed to flood link state messages?
  - O(N x E), where N is #nodes; E is #edges in graph
- Processing complexity for Dijkstra's algorithm?
  - $O(N^2)$, because we check all nodes w not in S at each iteration and we have O(N) iterations
  - more efficient implementations: O(N log(N))
- How many entries in the LS topology database? O(E)
- How many entries in the forwarding table? O(N)

Issue#2: Transient Disruptions

- Inconsistent link-state database
  - Some routers know about failure before others
  - The shortest paths are no longer consistent
  - sient forwa



A and D think that this is the path to C

E thinks that this is the path to C

Distance Vector Routing

Learn-By-Doing

Let's try to collectively develop
distance-vector routing from first principles

54

54

Experiment

- Your job: find the (route to) the youngest person in the room
- Ground Rules
  - **You may not** leave your seat, nor shout loudly across the class
  - **You may** talk with your immediate neighbors (N-S-E-W only)
  (hint: "exchange updates" with them)
- At the end of 5 minutes, I will pick a victim and ask:
  - who is the youngest person in the room? (date&name)
  - which one of your neighbors first told you this info.?

EQUIPMENT REQUIRED: PIECE OF PAPER and a PEN (or your emotional equivalent)

55

55

Go!

56

56

Distance-Vector Routing

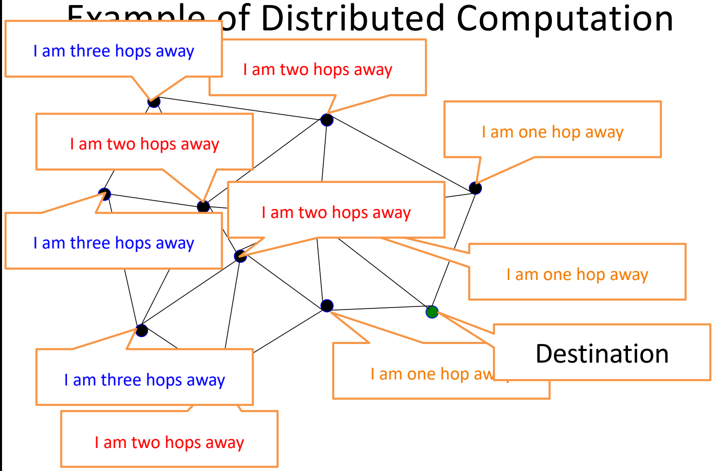
Example:

Routing Information Protocol (RIP)

57

57

Example of Distributed Computation



58

58

Distance Vector Routing

Each router sends its knowledge about the "whole" network to its neighbors. Information sharing at regular intervals.

- Each router knows the links to its neighbors
  - Does *not* flood this information to the whole network
- Each router has provisional "shortest path" to every other router
  - E.g.: Router A: "I can get to router B with cost 11"
- Routers exchange this distance vector information with their neighboring routers
  - Vector because one entry per destination
- Routers look over the set of options offered by their neighbors and select the best one
- Iterative process converges to set of shortest paths

59

59

A few other inconvenient truths

- What if we use a non-additive metric?
  - E.g., maximal capacity
- What if routers don't use the same metric?
  - I want low delay, you want low loss rate?
- What happens if nodes lie?

60

60

Can You Use Any Metric?

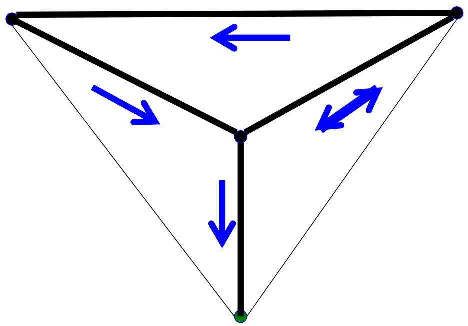
- I said that we can pick any metric. Really?
- What about maximizing capacity?

61

61

What Happens Here?

Problem: "cost" does not change around loop



Additive measures avoid this problem!

62

62

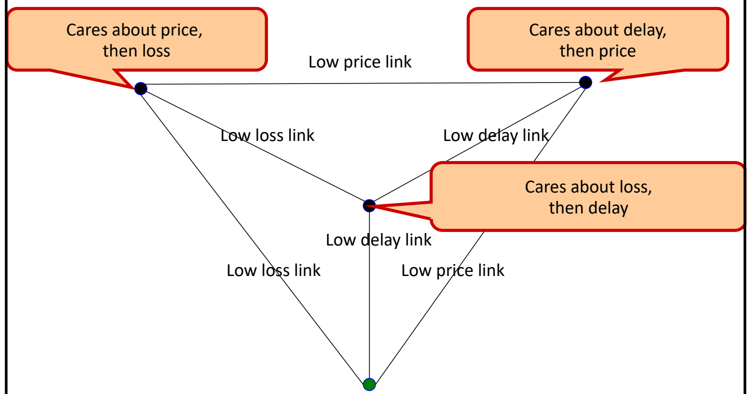
No agreement on metrics?

- If the nodes choose their paths according to different criteria, then bad things might happen
- Example
  - Node A is minimizing latency
  - Node B is minimizing loss rate
  - Node C is minimizing price
- Any of those goals are fine, if globally adopted
  - Only a problem when nodes use different criteria
- Consider a routing algorithm where paths are described by delay, cost, loss

63

63

What Happens Here?



64

64

Must agree on loop-avoiding metric

- When all nodes minimize same metric
- And that metric increases around loops
- Then process is guaranteed to converge

65

65

What happens when routers lie?

- What if a router claims a 1-hop path to everywhere?
- All traffic from nearby routers gets sent there
- How can you tell if they are lying?
- Can this happen in real life?
  - It has, several times….

66

66

Link State vs. Distance Vector

- Core idea
  - LS: tell all nodes about your immediate neighbors
  - DV: tell your immediate neighbors about (your least cost distance to) all nodes

67

67

Link State vs. Distance Vector

- LS: each node learns the complete network map; each node computes shortest paths independently and in parallel
- DV: no node has the complete picture; nodes cooperate to compute shortest paths in a distributed manner

→LS has higher messaging overhead
→LS has higher processing complexity
→LS is less vulnerable to looping

68

68

Link State vs. Distance Vector

Message complexity

- LS: O(NxE) messages;
  - N is #nodes; E is #edges
- DV: O(#Iterations x E)
  - where #Iterations is ideally O(network diameter) but varies due to routing loops or the count-to-infinity problem

Processing complexity

- LS: $O(N^2)$
- DV: O(#Iterations x N)

Robustness: what happens if router malfunctions?

- LS:
  - node can advertise incorrect *link* cost
  - each node computes only its *own* table
- DV:
  - node can advertise incorrect *path* cost
  - each node's table used by others; error propagates through network

69

69

Routing: Just the Beginning

- Link state and distance-vector are the deployed routing paradigms for intra-domain routing
- Inter-domain routing (BGP)
  - more Part II (Principles of Communications)
  - A version of DV

70

70

What are desirable goals for a routing solution?

- "Good" paths (least cost)
- Fast convergence after change/failures
  - no/rare loops
- Scalable
  - #messages
  - table size
  - processing complexity
- Secure
- Policy
- Rich metrics (more later)

71

71

Delivery models

- What if a node wants to send to more than one destination?
  - broadcast: send to all
  - multicast: send to all members of a group
  - anycast: send to any member of a group
- What if a node wants to send along more than one path?

72

72

Metrics

- Propagation delay
- Congestion
- Load balance
- Bandwidth (available, capacity, maximal, bbw)
- Price
- Reliability
- Loss rate
- Combinations of the above

In practice, operators set abstract "weights" (much like our costs); how exactly is a bit of a black art

73

73

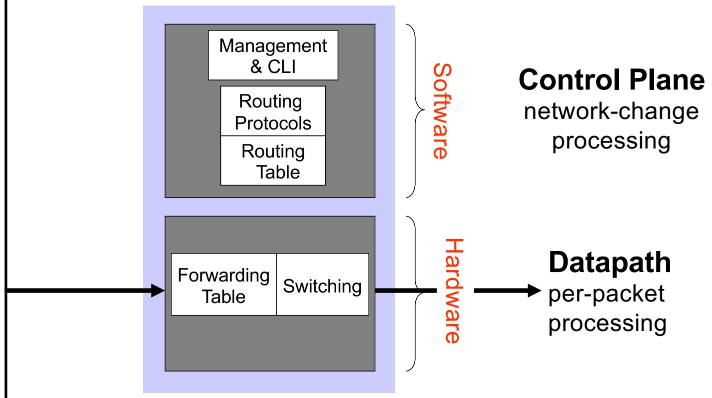
From Routing back to Forwarding

- Routing: “control plane”
  - Computing paths the packets will follow
  - Routers talking amongst themselves
  - Jointly creating the routing state
- Forwarding: “data plane”
  - Directing a data packet to an outgoing link
  - Individual router using routing state
- Two very different timescales....

74

74

Basic Architectural Components of an IP Router



75

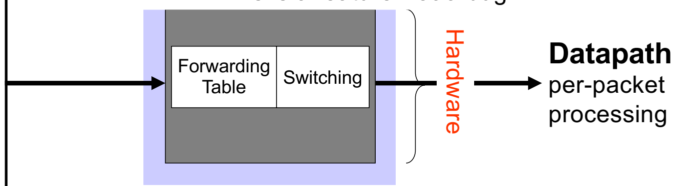
75

Independent operation!

If the control-plane **fails**…..

The data-path is **not affected**…
like a loyal pet it will keep going using the current (last) table update

This is a feature **not** a bug



76

76

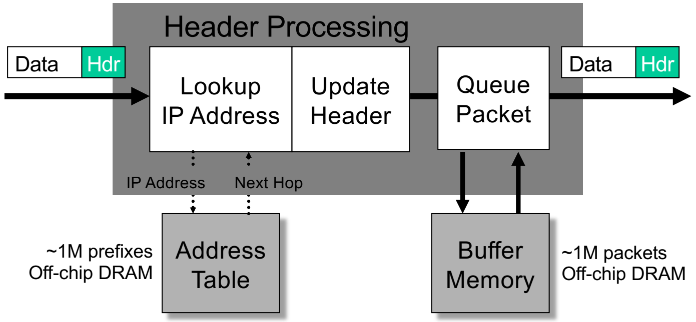
Per-packet processing in an IP Router

1. Accept packet arriving on an incoming link.
2. Lookup packet destination address in the forwarding table, to identify outgoing port(s).
3. Manipulate packet header: e.g., decrement TTL, update header checksum.
4. Send packet to the outgoing port(s).
5. Buffer packet in the queue.
6. Transmit packet onto outgoing link.

77

77

Generic Router Architecture



78

78

Forwarding tables

IP address } 32 bits wide → ~ 4 billion unique address

Naïve approach:
One entry per address

Entry	Destination	Port
1	0.0.0.0	1
2	0.0.0.1	2
⋮	⋮	⋮
2^{32}	255.255.255.255	12

} **~ 4 billion entries**

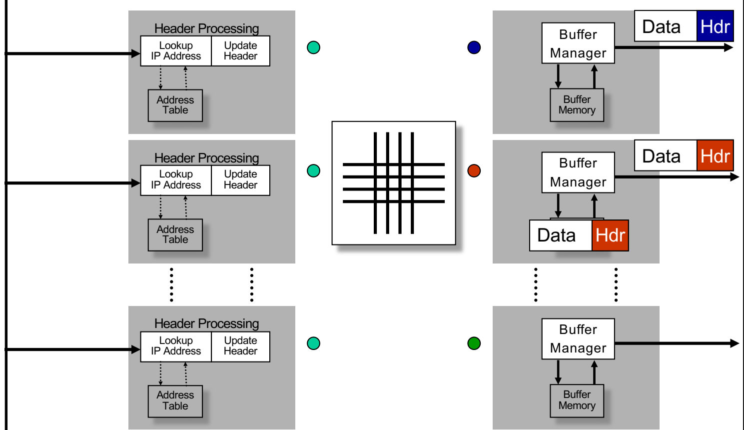
Improved approach:
Group entries to reduce table size

Entry	Destination	Port
1	0.0.0.0 – 127.255.255.255	1
2	128.0.0.1 – 128.255.255.255	2
⋮	⋮	⋮
50	248.0.0.0 – 255.255.255.255	12

79

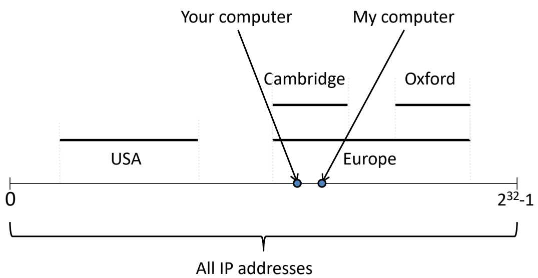
79

Generic Router Architecture



80

IP addresses as a line



Entry	Destination	Port
1	Cambridge	1
2	Oxford	2
3	Europe	3
4	USA	4
5	Everywhere (default)	5

81

81

Longest Prefix Match (LPM)

Entry	Destination	Port
1	Cambridge	1
2	Oxford	2
3	Europe	3
4	USA	4
5	Everywhere (default)	5

Universities

Continents

Planet

Matching entries:
- Cambridge — Most specific
- Europe
- Everywhere

To: Cambridge	Data

82

82

Longest Prefix Match (LPM)

Entry	Destination	Port
1	Cambridge	1
2	Oxford	2
3	Europe	3
4	USA	4
5	Everywhere (default)	5

Universities

Continents

Planet

Matching entries:
- Europe — Most specific
- Everywhere

To: France	Data

83

83

Implementing Longest Prefix Match

Entry	Destination	Port	
1	Cambridge	1	Searching
2	Oxford	2	
3	Europe	3	
4	USA	4	FOUND
5	Everywhere (default)	5	

Most specific

Least specific

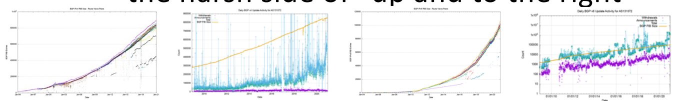
84

84

Forwarding table realities

- High Speed: Must be “packet-rate” lookup
  - about 200M lookups / second for 100Gbps
- Large (messy) tables – (BGP Jan 2021 stats)
  - 866,000+ routing prefix entries for IPv4
  - 104,000+ routing prefix entries for IPv6
- Changing and Growing

the harsh side of “up and to the right”



Open problems : continual growth is continual demand for innovation opportunities in control, algorithms, & network hardware

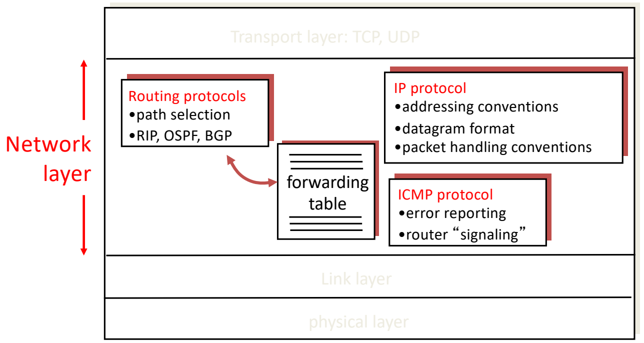
85

Hudson 2020 report https://blog.apnic.net/2021/01/05/bgp-in-2020-the-bgp-table/

85

The Internet version of a Network layer

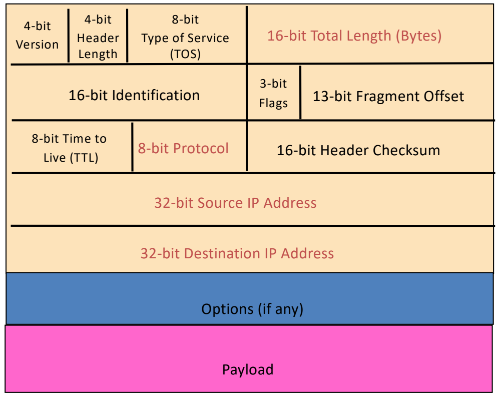
Host, router network layer functions:



86

86

IPv4 Packet Structure
20 Bytes of Standard Header, then Options



87

87

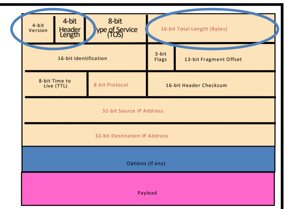
(Packet) Network Tasks One-by-One

- Read packet correctly
- Get packet to the destination
- Get responses to the packet back to source
- Carry data
- Tell host what to do with packet once arrived
- Specify any special network handling of the packet
- Deal with problems that arise along the path

88

88

Reading Packet Correctly

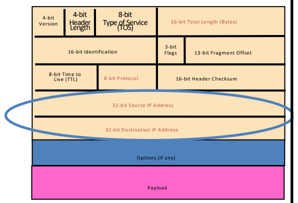


- Version number (4 bits)
  - Indicates the version of the IP protocol
  - Necessary to know what other fields to expect
  - Typically "4" (for IPv4), and sometimes "6" (for IPv6)
- Header length (4 bits)
  - Number of 32-bit words in the header
  - Typically "5" (for a 20-byte IPv4 header)
  - Can be more when IP options are used
- Total length (16 bits)
  - Number of bytes in the packet
  - Maximum size is 65,535 bytes (2^{16} -1)
  - … though underlying links may impose smaller limits

89

89

Getting Packet to Destination and Back

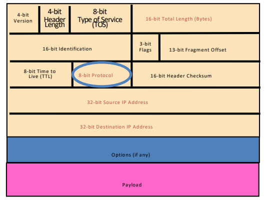


- Two IP addresses
  - Source IP address (32 bits)
  - Destination IP address (32 bits)
- Destination address
  - Unique identifier/locator for the receiving host
  - Allows each node to make forwarding decisions
- Source address
  - Unique identifier/locator for the sending host
  - Recipient can decide whether to accept packet
  - Enables recipient to send a reply back to source

90

90

Telling Destination Host How to Handle Packet



- Protocol (8 bits)
  - Identifies the higher-level protocol
  - Important for demultiplexing at receiving host
- Most common examples
  - E.g., “6” for the Transmission Control Protocol (TCP)
  - E.g., “17” for the User Datagram Protocol (UDP)

protocol=6	protocol=17
IP header	IP header
TCP header	UDP header

91

91

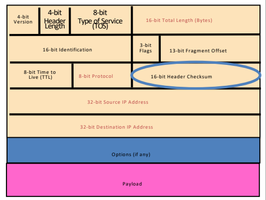
Potential Problems

- Header Corrupted: **Checksum**
- Loop: **TTL**
- Packet too large: **Fragmentation**

93

93

Header Corruption

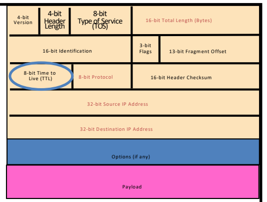


- Checksum (16 bits)
  - Particular form of checksum over packet header
- If not correct, router discards packets
  - So it doesn’t act on bogus information
- Checksum recalculated at every router
  - **Why?**
  - **Why include TTL?**
  - **Why only header?**

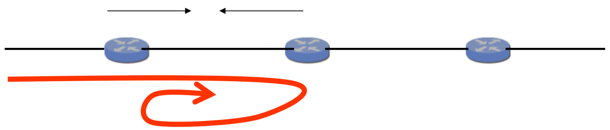
94

94

Preventing Loops
(aka Internet Zombie plan)



- Forwarding loops cause packets to cycle forever
  - As these accumulate, eventually consume **all** capacity

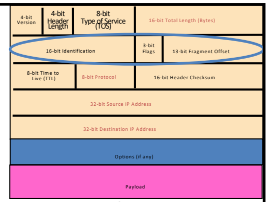


- Time-to-Live (TTL) Field (8 bits)
  - Decremented at each hop, packet discarded if reaches 0
  - …and "time exceeded" message is sent to the source
    - Using "ICMP" control message; basis for **traceroute**

95

95

Fragmentation
(some assembly required)



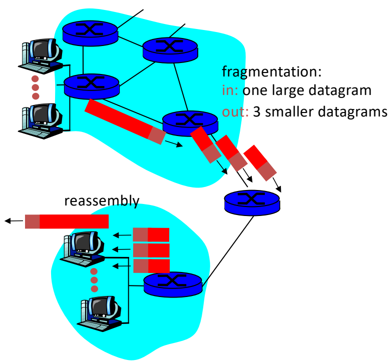
- Fragmentation: when forwarding a packet, an Internet router can split it into multiple pieces ("fragments") if too big for next hop link
- Must reassemble to recover original packet
  - Need fragmentation information (32 bits)
  - Packet identifier, flags, and fragment offset

96

96

IP Fragmentation & Reassembly

- network links have MTU (max.transfer size) - largest possible link-level frame.
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments
- IPv6 does things differently…

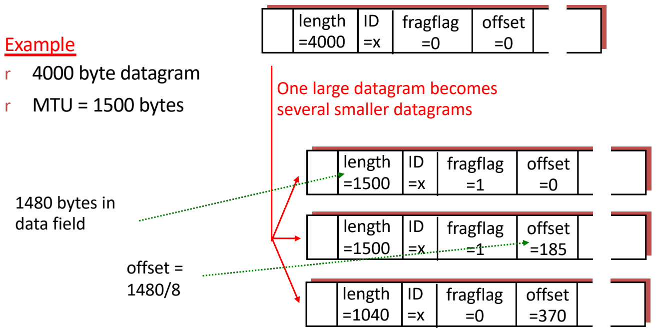


97

97

IP Fragmentation and Reassembly

Example
- 4000 byte datagram
- MTU = 1500 bytes

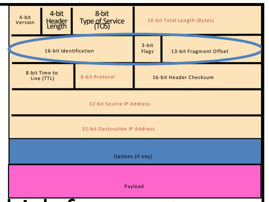


Question: What happens when a fragment is lost?

98

98

Fragmentation Details



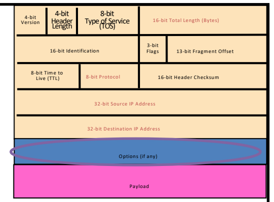
- Identifier (16 bits): used to tell which fragments belong together
- Flags (3 bits):
  - Reserved **(RF):** unused bit
  - Don't Fragment **(DF):** instruct routers to **not** fragment the packet even if it won't fit
    - Instead, they **drop** the packet and send back a "Too Large" ICMP control message
    - Forms the basis for "Path MTU Discovery"
  - More (**MF**): this fragment is not the last one
- Offset (13 bits): what part of datagram this fragment covers in 8-byte units

Pop quiz question: Why do frags use offset and not a frag number?

99

99

Options



- End of Options List
- No Operation (padding between options)
- Record Route
- Strict Source Route
- Loose Source Route
- Timestamp
- Traceroute
- Router Alert
- .....

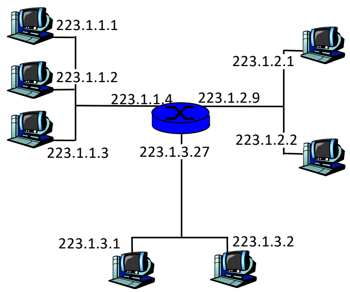
Few are used as each requires special handling in an IP router.

100

100

IP Addressing: introduction

- IP address: 32-bit identifier for host, router *interface*
- *interface:* connection between host/router and physical link
  - routers typically have multiple interfaces
  - host typically has one interface
  - IP addresses associated with each interface



223.1.1.1 = 11011111 00000001 00000001 00000001

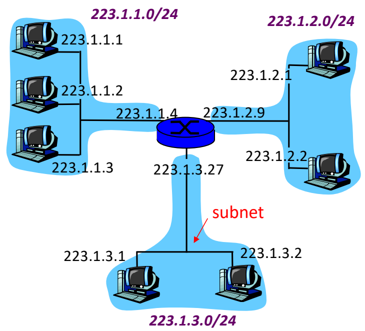
223 1 1 1

101

101

Subnets

- IP address:
  - subnet part (high order bits)
  - host part (low order bits)
- *What's a subnet ?*
  - device interfaces with same subnet part of IP address
  - can physically reach each other without intervening router



subnet part | host part

11011111 00000001 00000011 00000000

223.1.3.0/24

CIDR: Classless InterDomain Routing
- subnet portion of address of arbitrary length
- address format: a.b.c.d/x, where x is # bits in subnet portion of address

Subnet mask: /24

network consisting of 3 subnets

102

102

IP addresses: how to get one?

Q: How does a *host* get IP address?

- hard-coded by system admin in a file
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config (circa 1980's your mileage will vary)
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - "plug-and-play"

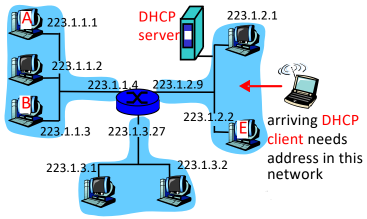
103

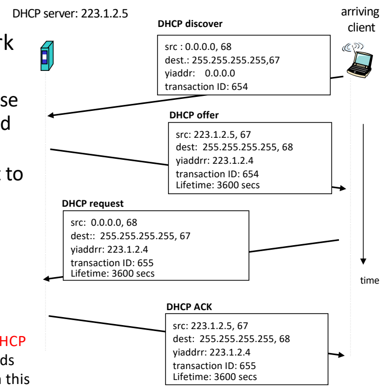
103

DHCP client-server scenario

Goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- Can renew its lease on address in use
- Allows reuse of addresses (only hold address while connected an "on")
- Support for mobile users who want to join network (more shortly)





104

104

IP addresses: how to get one?

Q: How does *network* get subnet part of IP addr?

A: gets allocated portion of its provider ISP's address space

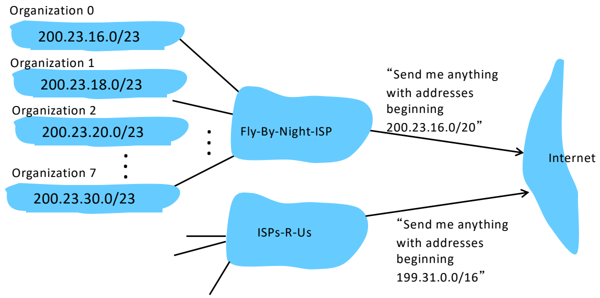
ISP's block	11001000 00010111 00010000	00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100	00000000	200.23.20.0/23
...	.....	....	....
Organization 7	11001000 00010111 00011110	00000000	200.23.30.0/23

105

105

Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:

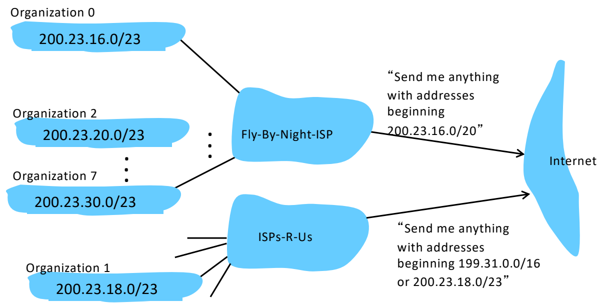


106

106

Hierarchical addressing: more specific routes

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



107

107

IP addressing: the last word...

Q: How does an ISP get a block of addresses?

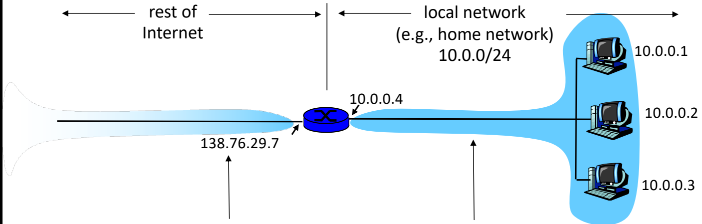
A: ICANN: Internet Corporation for Assigned Names and Numbers
- allocates addresses
- manages DNS
- assigns domain names, resolves disputes

108

108

Cant get more IP addresses? well there is always.....

NAT: Network Address Translation



All datagrams *leaving* local network have same single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

109

109

NAT: Network Address Translation

- Motivation: local network uses just one IP address as far as outside world is concerned:
  - range of addresses not needed from ISP: just one IP address for all devices
  - can change addresses of devices in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - devices inside local net not explicitly addressable, visible by outside world (a security plus).

110

110

NAT: Network Address Translation

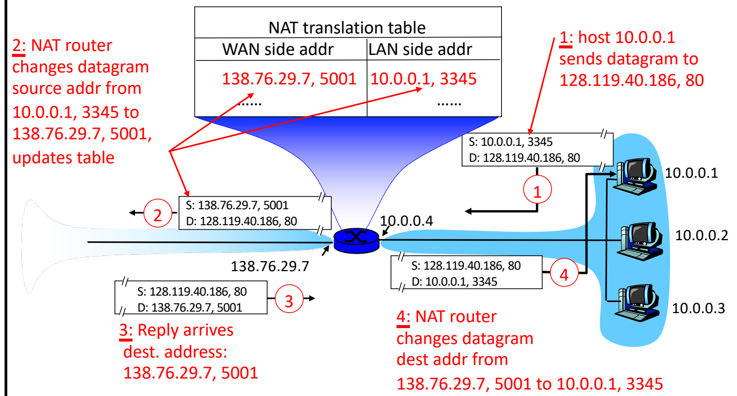
Implementation: NAT router must:

- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  - . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr.
- *remember (in NAT translation table)* every (source IP address, port #) to (NAT IP address, new port #) translation pair
- *incoming datagrams: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

111

111

NAT: Network Address Translation



112

112

NAT: Network Address Translation

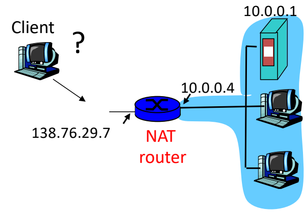
- 16-bit port-number field:
  - 60,000+ simultaneous connections with a single WAN-side address!
- NAT is controversial:
  - routers should only process up to layer 3
  - violates end-to-end argument (?)
    - NAT possibility must be taken into account by app designers, eg, P2P applications
  - address shortage should instead be solved by IPv6

113

113

NAT traversal problem

- client wants to connect to server with address 10.0.0.1
  - server address 10.0.0.1 local to LAN (client can't use it as destination addr)
  - only one externally visible NATted address: 138.76.29.7
- solution 1: statically configure NAT to forward incoming connection requests at given port to server
  - e.g., (138.76.29.7, port 2500) always forwarded to 10.0.0.1 port 25000



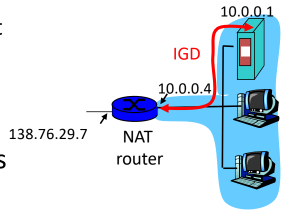
114

114

NAT traversal problem

- solution 2: Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATted host to:
  - ❖learn public IP address (138.76.29.7)
  - ❖add/remove port mappings (with lease times)

i.e., automate static NAT port map configuration

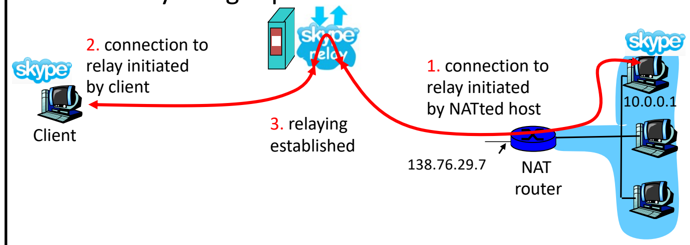


115

115

NAT traversal problem

- solution 3: relaying (was used in (really old) Skype)
  - NATed client establishes connection to relay
  - External client connects to relay
  - relay bridges packets between to connections



116

116

Remember this? Traceroute at work…

traceroute: rio.cl.cam.ac.uk to munnari.oz.au

(tracepath on windows is similar)

Three delay measurements from rio.cl.cam.ac.uk to gatwick.net.cl.cam.ac.uk

trans-continent link

```
traceroute munnari.oz.au
traceroute to munnari.oz.au (202.29.151.3), 30 hops max, 60 byte packets
 1  gatwick.net.cl.cam.ac.uk (128.232.32.2)  0.416 ms  0.384 ms  0.427 ms
 2  cl-sby.route-nwest.net.cam.ac.uk (193.60.89.9)  0.393 ms  0.440 ms  0.494 ms
 3  route-nwest.route-mill.net.cam.ac.uk (192.84.5.137)  0.407 ms  0.448 ms  0.501 ms
 4  route-mill.route-enet.net.cam.ac.uk (192.84.5.94)  1.006 ms  1.091 ms  1.163 ms
 5  xe-11-3-0.camb-rbr1.eastern.ja.net (146.97.130.1)  0.300 ms  0.313 ms  0.350 ms
 6  ae24.lowdss-sbr1.ja.net (146.97.37.185)  2.679 ms  2.664 ms  2.712 ms
 7  ae28.londhx-sbr1.ja.net (146.97.33.17)  5.955 ms  5.953 ms  5.901 ms
 8  janet.mx1.lon.uk.geant.net (62.40.124.197)  6.059 ms  6.066 ms  6.052 ms
 9  ae0.mx1.par.fr.geant.net (62.40.98.77)  11.742 ms  11.779 ms  11.724 ms
10  ae1.mx1.mad.es.geant.net (62.40.98.64)  27.751 ms  27.734 ms  27.704 ms
11  mb-so-02-v4.bb.tein3.net (202.179.249.117)  138.296 ms  138.314 ms  138.282 ms
12  sg-so-04-v4.bb.tein3.net (202.179.249.53)  196.303 ms  196.293 ms  196.264 ms
13  th-pr-v4.bb.tein3.net (202.179.249.66)  225.153 ms  225.178 ms  225.196 ms
14  pyt-thairen-to-02-bdr-pyt.uni.net.th (202.29.12.10)  225.163 ms  223.343 ms  223.363 ms
15  202.28.227.126 (202.28.227.126)  241.038 ms  240.941 ms  240.834 ms
16  202.28.221.46 (202.28.221.46)  287.252 ms  287.306 ms  287.282 ms
17  * * *
18  * * *
19  * * *
20  coe-gw.psu.ac.th (202.29.149.70)  241.681 ms  241.715 ms  241.680 ms
21  munnari.OZ.AU (202.29.151.3)  241.610 ms  241.636 ms  241.537 ms
```

* means no response (probe lost, router not replying)

117

117

Traceroute and ICMP

- Source sends series of UDP segments to dest
  - First has TTL =1
  - Second has TTL=2, etc.
  - Unlikely port number
- When nth datagram arrives to nth router:
  - Router discards datagram
  - And sends to source an ICMP message (type 11, code 0)
  - Message includes name of router& IP address
- When ICMP message arrives, source calculates RTT
- Traceroute does this 3 times

Stopping criterion

- UDP segment eventually arrives at destination host
- Destination returns ICMP "host unreachable" packet (type 3, code 3)
- When source gets this ICMP, stops.

118

118

ICMP: Internet Control Message Protocol

- used by hosts & routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

119

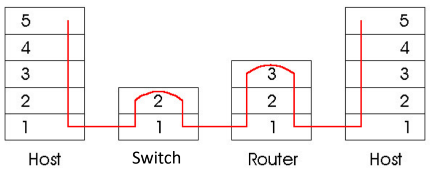
Gluing it together:

How does my Network (address) interact with my Data-Link (address) ?

120

Switches vs. Routers Summary

- both store-and-forward devices
  - routers: network layer devices (examine network layer headers eg IP)
  - switches are link layer devices (examine Data-Link-Layer headers eg Ethernet)
- Routers: implement routing algorithms, maintain routing tables of the network – create network forwarding tables from routing tables
- Switches: implement learning algorithms, learn switch/DLL forwarding tables



121

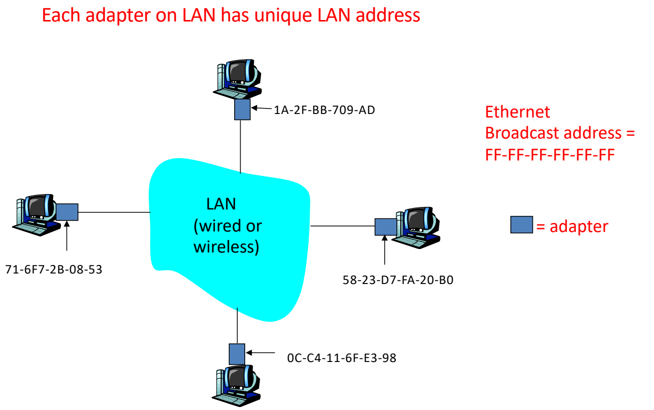
MAC Addresses (and IPv4 ARP)

or How do I glue my network to my data-link?

- 32-bit IP address:
  - *network-layer* address
  - used to get datagram to destination IP subnet
- 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, firmware, etc.

122

LAN Addresses and ARP



123

123

Address Resolution Protocol

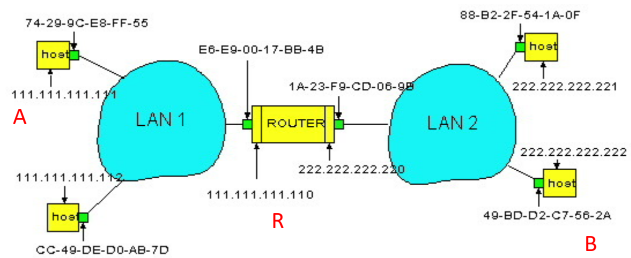
- Every node maintains an ARP table
  – <IP address, MAC address> pair
- Consult the table when sending a packet
  – Map destination IP address to destination MAC address
  – Encapsulate and transmit the data packet
- But: what if IP address not in the table?
  – Sender broadcasts: "**Who has IP address 1.2.3.156**?"
  – Receiver responds: "**MAC address 58-23-D7-FA-20-B0**"
  – Sender caches result in its ARP table

124

124

Example: A Sending a Packet to B

How does host A send an IP packet to host B?

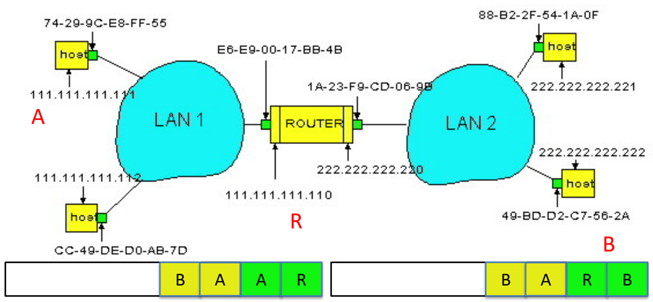


125

125

Example: A Sending a Packet to B

How does host A send an IP packet to host B?



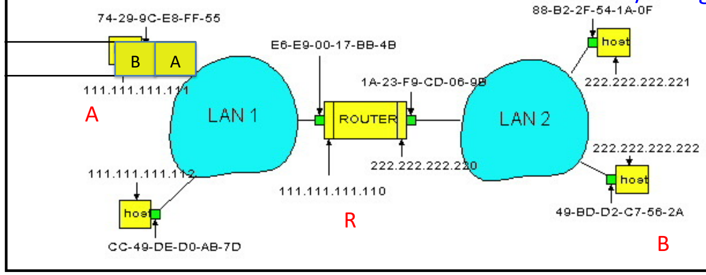
1. **A sends packet to R.**
2. **R sends packet to B.**

126

126

Host A Decides to Send Through R

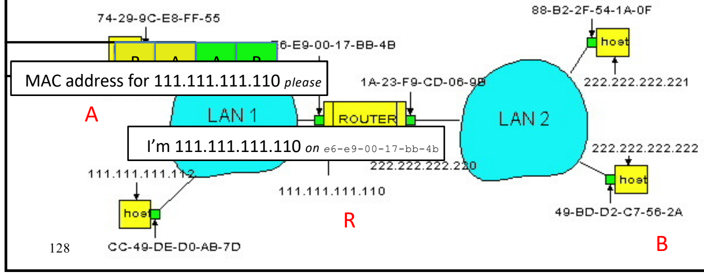
- Host A constructs an IP packet to send to B
  - Source 111.111.111.111, destination 222.222.222.222
- Host A has a gateway router R
  - Used to reach destinations outside of 111.111.111.0/24
  - Address 111.111.111.110 for R learned via DHCP/config



127

Host A Sends Packet Through R

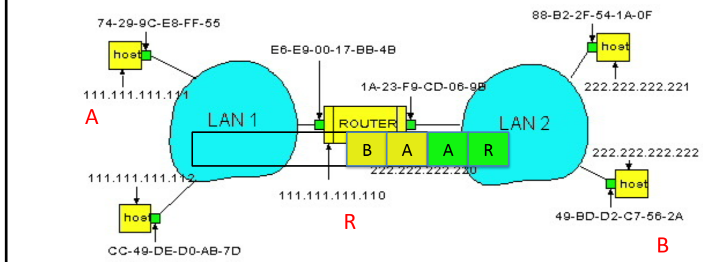
- Host A learns the MAC address of R's interface
  - ARP request: broadcast request for 111.111.111.110
  - ARP response: R responds with E6-E9-00-17-BB-4B
- Host A encapsulates the packet and sends to R



128

R Decides how to Forward Packet

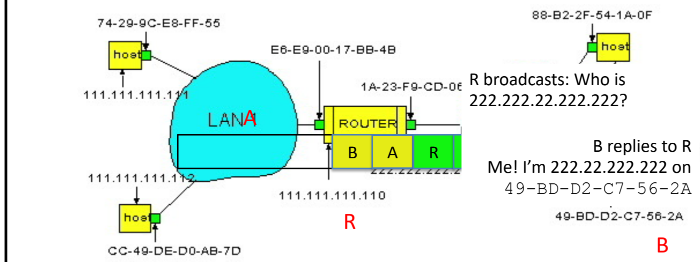
- Router R's adaptor receives the packet
  - R extracts the IP packet from the Ethernet frame
  - R sees the IP packet is destined to 222.222.222.222
- Router R consults its forwarding table
  - Packet matches 222.222.222.0/24 via other adaptor



129

R Sends Packet to B

- Router R's learns the MAC address of host B
  - ARP request: broadcast request for 222.222.222.222
  - ARP response: B responds with 49-BD-D2-C7-52A
- Router R encapsulates the packet and sends to B



130

Security Analysis of ARP

- Impersonation
  - Any node that hears request can answer …
  - … and can say whatever they want
- Actual legit receiver never sees a problem
  - Because even though later packets carry its IP address, its NIC doesn't capture them since the (naughty) packets are not its MAC address

131

131

Key Ideas in Both ARP and DHCP

- Broadcasting: Can use broadcast to make contact
  - Scalable because of limited size
- Caching: remember the past for a while
  - Store the information you learn to reduce overhead
  - Remember your own address & other host's addresses
- Soft state: eventually forget the past
  - Associate a time-to-live field with the information
  - … and either refresh or discard the information
  - Key for robustness in the face of unpredictable change

132

132

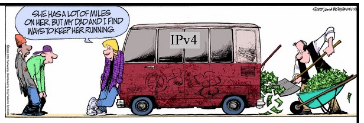
Why Not Use DNS-Like Tables?

- When host arrives:
  - Assign it an IP address that will last as long it is present
  - Add an entry into a table in DNS-server that maps MAC to IP addresses
- Answer:
  - Names: explicit creation, and are plentiful
  - Hosts: come and go without informing network
    - Must do mapping on demand
  - Addresses: not plentiful, need to reuse and remap
    - Soft-state enables dynamic reuse

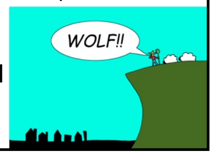
133

133

IPv6



- Motivated prematurely by address exhaustion
  - addresses are larger
  - packet headers are laid out differently
  - address management and configuration are completely different
  - some DNS behavior changes
  - some sockets code changes
  - *everybody now has a hard time parsing IP addresses*
- Steve Deering focused on simplifying IP
  - Got rid of all fields that were not absolutely necessary
  - "Spring Cleaning" for IP
- Result is an elegant, if unambitious, protocol



134

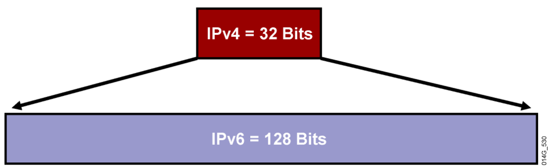
134

IPv4	IPv6
Addresses are 32 bits (4 bytes) in length.	Addresses are 128 bits (16 bytes) in length
Address (A) resource records in DNS to map host names to IPv4 addresses.	Address (AAAA) resource records in DNS to map host names to IPv6 addresses.
Pointer (PTR) resource records in the IN-ADDR.ARPA DNS domain to map IPv4 addresses to host names.	Pointer (PTR) resource records in the IP6.ARPA DNS domain to map IPv6 addresses to host names.
IPSec is optional and should be supported externally	IPSec support is not optional
Header does not identify packet flow for QoS handling by routers	Header contains Flow Label field, which Identifies packet flow for QoS handling by router.
Both routers and the sending host fragment packets.	Routers do not support packet fragmentation. Sending host fragments packets
Header includes a checksum.	Header does not include a checksum.
Header includes options.	Optional data is supported as extension headers.
ARP uses broadcast ARP request to resolve IP to MAC/Hardware address.	Multicast Neighbor Solicitation messages resolve IP addresses to MAC addresses.
Internet Group Management Protocol (IGMP) manages membership in local subnet groups.	Multicast Listener Discovery (MLD) messages manage membership in local subnet groups.
Broadcast addresses are used to send traffic to all nodes on a subnet.	IPv6 uses a link-local scope all-nodes multicast address.
Configured either manually or through DHCP.	Does not require manual configuration or DHCP.
Must support a 576-byte packet size (possibly fragmented).	Must support a 1280-byte packet size (without fragmentation).

136

Larger Address Space

- IPv4 = 4,294,967,295 addresses
- IPv6 = 340,282,366,920,938,463,374,607,432,768,211,456 addresses
- 4x in number of bits translates to **huge** increase in address space!

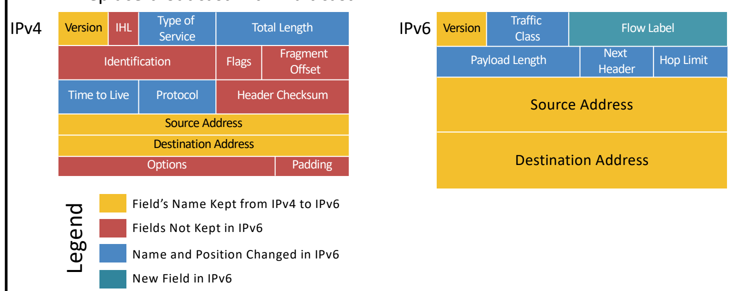


137

137

Other Significant Protocol Changes - 1

- Increased minimum MTU from 576 to 1280
- No enroute fragmentation… fragmentation only at source
- Header changes (20bytes to 40bytes)
- Replace broadcast with multicast



138

138

Other Significant Protocol Changes - 2

operation is intended to be simpler within the network:

- no *in-network* fragmentation
- no checksums in IPv6 header
- UDP checksum required (wasn't in IPv4) rfc6936: **No more zero**
- optional state carried in *extension headers*
  - Extension headers notionally replace IP options
  - Each extension header indicates the type of the *following* header, so they can be chained
  - The final 'next header' either indicates there is no 'next', or escapes into an transport-layer header (e.g., TCP)

139

139

IPv6 Basic Address Structure

IPv6 addresses are split into two primary parts:

0 — 64	64 — 128
Routing Prefix	Interface Identifier

- 64 bits is dedicated to an addressable interface (equivalent to the host, if it only has one interface)
- The network prefix allocated to a network by a registry can be up to 64-bits long
- An allocation of a /64 (i.e. a 64-bit network prefix) allows *one* subnet (it cannot be subdivided)
- A /63 allows two subnets; a /62 offers four, etc. /48s are common for older allocations (RFC 3177, obsoleted by RFC 6177).
- Longest-prefix matching operates as in IPv4.

140

140

IPv6 Address Representation (quick)

IPv6 addresses represented as eight 16-bit blocks (4 hex chars) separated by colons:

- `2001:4998:000c:0a06:0000:0000:0002:4011`

But we can condense the representation by removing leading zeros in each block:

- `2001:4998:c:a06:0:0:2:4011`

And by reducing the consecutive block of zeros to a "`::`" (this double colon rule can only be applied once)

- `2001:4998:c:a06::2:4011`

141

141

IPv6 Address Families

The address space is carved, like v4, into certain categories [1]:

host-local : localhost; `::1` is equivalent to `127.0.0.1`

link-local : not routed: `fe80::/10` is equivalent to `169.254.0.0/16`

site-local : not routed *globally*: `fc00::/7` is equivalent to `192.168.0.0/16` or `10.0.0.0/8`

global unicast : `2000::/3` is basically any v4 address not reserved in some other way

multicast : `ff00::/8` is equivalent to `224.0.0.0/4`

[1] http://www.ripe.net/lir-services/new-lir/ipv6_reference_card.pdf

142

142

Problem with /64 Subnets

- Scanning a subnet becomes a DoS attack!
  - Creates IPv6 version of 2^{64} ARP entries in routers
  - Exhaust address-translation table space
- So now we have:

`ping6 ff02::1` All nodes in broadcast domain
`ping6 ff02::2` All routers in broadcast domain

- Solutions
  - RFC 6164 recommends use of /127 to protect router-router links
  - RFC 3756 suggest "clever cache management" to address more generally

143

143

Neighbour Discovery

- The Neighbour Discovery Protocol[2] specifies a set of ICMPv6 message types that allow hosts to discover other hosts or routing hardware on the network
  - neighbour solicitation
  - neighbour advertisement
  - router solicitation
  - router advertisement
  - redirect
- In short, a host can *solicit* neighbour (host) state to determine the layer-2 address of a host *or* to check whether an address is in use
- or it can solicit router state to learn more about the network configuration
- In both cases, the solicit message is sent to a well-known multicast address

[2] http://tools.ietf.org/html/rfc4861

144

144

IPv6 Dynamic Address Assignment

We have the two halves of the IPv6 address: the network component and the host component. Those are derived in different ways.

Network (top 64 bits):

- Router Advertisements (RAs)

  Interface

Identifier (bottom 64 bits):

- Stateless, automatic: SLAAC
- Stateful, automatic: DHCPv6

145

145

SLAAC: overview

SLAAC is:

- ... intended to make network configuration easy without manual configuration *or even a DHCP server*
- ... an algorithm for hosts to automatically configure their network interfaces (set up addresses, learn routes) without intervention

146

146

SLAAC: overview

- When a host goes live or an interface comes up, the system wants to know more about its environment
- It *can* configure link-local addresses for its interfaces: it uses the interface identifier, the EUI-64
- It uses this to ask (solicit) router advertisements sooner than the next periodic announcements; ask the network for information

147

147

SLAAC: overview

The algorithm (assuming one interface):

1. Generate potential link-local address
2. Ask the network (multicast[4]) if that address is in use: *neighbour solicitation*
3. Assuming no responses, assign to interface

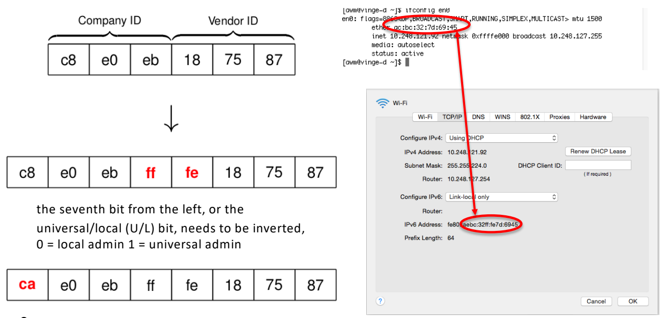
[4] https://tools.ietf.org/html/rfc2373

148

148

The EUI-64 Interface Identifier

- IEEE 64-bit Extended Unique Identifier (EUI-64)[3]
- There are various techniques to derive a 64-bit value, but often times we derive from the 48-bit MAC address



[3] http://tools.ietf.org/html/rfc2373

149

149

SLAAC: overview; Router Solicitation

Then,
- Once the host has a unique *link-local* address, it can send packets to anything else sharing that link substrate
  ... but the host doesn't yet know any routers, or public routes
  ... bootstrap: routers listen to a well-known multicast address

4. host asks the network (multicast) for router information: *router solicitation*

5. responses from the routers are sent directly (unicast) to the host that sent the router solicitation

6. the responses *may* indicate that the host should do more (e.g., use DHCP to get DNS information)

150

150

Router Advertisement

Without solicitation, router advertisements are generated intermittently by routing hardware.

Router Advertisements:
- nodes that forward traffic periodically advertise themselves to the network
- periodicity and expiry of the advertisement are configurable

Router Advertisement (RA), among other things, tells a host where to derive its network state with two flags: M(anaged) and O(ther info):
- M: “Managed Address Configuration”, which means: use DHCPv6 to find your host address (and ignore option O)
- O: Other information is available via DHCPv6, such as DNS configuration

151

151

Uh-oh

What problem(s) arises from totally decentralised address configuration?

Concerns that arise from using an EUI-64:

- Privacy: SLAAC interface identifiers don't change over time, so a host can be identified across networks
- Security: embedding a MAC address into an IPv6 address will carry that vendor's ID(s)[5], a possible threat vector

[5]http://standards.ieee.org/develop/regauth/oui/public.html

152

152

Address Configuration: SLAAC Privacy Addresses

Privacy extensions for SLAAC[6]

- temporary addresses for initiating outgoing sessions
- generate one temporary address per prefix
- when they expire, they are not used for new sessions, but can continue to be used for existing sessions
- the addresses should appear random, such that they are difficult to predict
- lifetime is configurable; this OSX machine sets an 86,400s timer (1 day)

[6]https://tools.ietf.org/html/rfc4941

153

153

Address Configuration: SLAAC Privacy Addresses

The algorithm:

- Assume: a stored 64-bit input value from previous iterations, or a pseudo-randomly generated value

1. take that input value and append it to the EUI-64
2. compute the MD5 message digest of that value
3. set bit 6 to zero
4. compare the leftmost 64-bits against a list of reserved interface identifiers and those already assigned to an address on the local device. If the value is unacceptable, re-run using the rightmost 64 bits of the result instead of the historic input value in step 1
5. use the leftmost 64-bits as the randomised interface identifier
6. store the rightmost 64-bits as the history value to be used in the next iteration of the algorithm

154

154

IPv6: why has the transition taken so long?

IPv4 and IPv6 are not compatible:

- different packet formats
- different addressing schemes

as the Internet has grown bigger and accumulated many IPv4-only services, transition has proven ... Tricky

Incentive issues

Virgin Media policy in 2010

....When IPV6 is rolled out across the whole of the Internet then a lot of the ISP's will roll out IPV6, ...

155

155

IPv6: why has the transition taken so long?

- IPv4 has/had the momentum
  ... which led to CIDR
  ... and encouraged RFC1918 space and NAT

- IPv4 NAT was covered earlier in this topic (reminder)
  - your ISP hands you only one IPv4 address
  - you share that across multiple devices in your household
  - The NAT handles all the translation between internal ("private") and external ("public") space

156

156

Transition tech: outline

- Tunnelling
- dual-stacked services, and happy eyeballs
- DNS64 and NAT64[8]
- 464XLAT
- DNS behaviour

[8] https://tools.ietf.org/html/rfc6146

157

157

Transition tech: outline

- Tunnelling

HURRICANE ELECTRIC
INTERNET SERVICES

Hurricane Electric Free IPv6 Tunnel Broker

IPv6 Tunnel Broker

Think of it as an IPv6 VPN service; which is essentially what it is

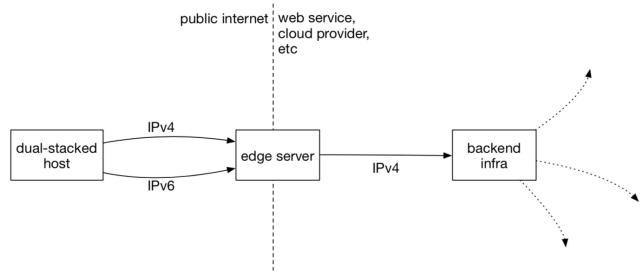
[8] https://tools.ietf.org/html/rfc6146

158

158

Dual-Stack Services: Common Deployment

It's common for web services to play conservatively: dual-stack your edge services (e.g., load balancers), leaving some legacy infrastructure for later:



159

159

Dual-Stack Services: Common Deployment

Aim is to reduce the pain:
- You can dual-stack the edge hosts, and carry state in, say, HTTP headers indicating the user's IP address (common over v4 anyway)
- You can dual-stack the backend opportunistically, over a longer period of time
- You use DNS to enable/disable the v6 side last (if there is no AAAA record in DNS, no real users will connect to the IPv6 infrastructure

160

160

Happy Eyeballs and DNS

- The introduction of IPv6 carried with it an obligation that applications attempt to use IPv6 before falling back to IPv4.
- What happens though if you try to connect to a host which doesn't exist?[9]
- But the presence of IPv6 modifies the behaviour of DNS responses and response preference[10]

[9] https://tools.ietf.org/html/rfc5461
[10] https://tools.ietf.org/html/rfc3484

161

161

Happy Eyeballs

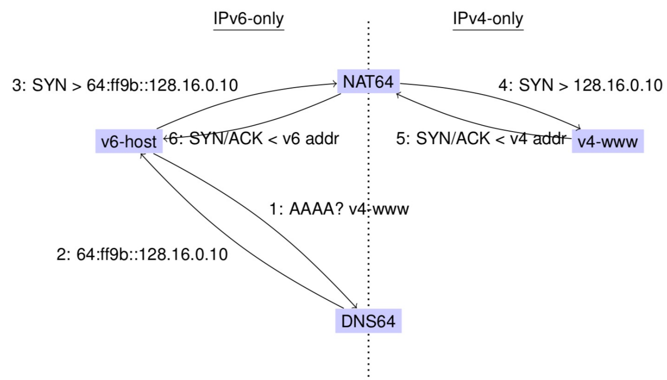
- Happy Eyeballs[11] was the proposed solution
  - the eyeballs in question are yours, or mine, or whoever is sitting in front of their browser getting mad that things are unresponsive
- Modifies application behaviour

[11] https://tools.ietf.org/html/rfc8305

162

162

DNS64 & NAT64



163

163

464XLAT

- Problem: IPv6-only to the host, but an IPv4-only app trying to access an IPv4-only service
  - Some *applications* do not understand IPv6, so having an IPv6 address doesn't help
  - 464XLAT[12] solves this problem
  - In essence, DNS64 + NAT64 + a shim layer on the host itself to offer IPv4 addresses to apps

[12]https://tools.ietf.org/html/rfc6877

164

164

Improving on IPv4 and IPv6?

- Why include unverifiable source address?
  - Would like accountability ***and*** anonymity (now neither)
  - Return address can be communicated at higher layer
- Why packet header used at edge same as core?
  - Edge: host tells network what service it wants
  - Core: packet tells switch how to handle it
    - One is local to host, one is global to network
- Some kind of payment/responsibility field?
  - Who is responsible for paying for packet delivery?
  - Source, destination, other?
- Other ideas?

167

167

Summary Network Layer

- understand principles behind network layer services:
  - network layer service models
  - forwarding versus routing (versus switching)
  - how a switch & router works
  - routing (path selection)
  - IPv6
- Algorithms
  - Two routing approaches (LS vs DV)
  - One of these in detail (LS)
  - ARP
- Other Core ideas
  - Caching, soft-state, broadcast
  - Fate-sharing in practice….

168

168