

Chapter 4: Network Layer

Chapter goals:

- ✱ understand principles behind network layer services:
 - routing (path selection)
 - dealing with scale
 - how a router works
 - advanced topics: IPv6, multicast
- ✱ instantiation and implementation in the Internet

Overview:

- ✱ network layer services
- ✱ routing principle: path selection
- ✱ hierarchical routing
- ✱ IP
- ✱ Internet routing protocols
- ✱ reliable transfer
 - intra-domain
 - inter-domain
- ✱ what's inside a router?
- ✱ IPv6
- ✱ multicast routing

1

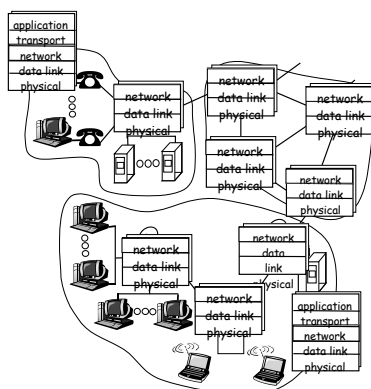
Quiz

- ✱ What is the main role of routing?
- ✱ Describe a centralized routing protocol
 - Main operation functions
 - Comment on complexity, robustness
- ✱ Describe a table driven routing protocol
 - Main functions
 - Comment on complexity
- ✱ How can the Internet routing scale?
 - Describe the elements that make it work
 - What does a router need to keep in its memory
 - Describe a routing table
- ✱ What happens to a packet when it arrives at a router?
 - What kind of “hardware” does it go through?
 - Where does the delay and loss come into play?

2

Network layer functions

- ✱ transport packet from sending to receiving hosts
 - ✱ network layer protocols in every host, router
- three important functions:
- ✱ **path determination:** route taken by packets from source to dest. *Routing algorithms*
 - ✱ **switching:** move packets from router's input to appropriate router output
 - ✱ **call setup:** some network architectures require router call setup along path before data flows



3

Network service model

Q: What service model for “channel” transporting packets from sender to receiver?

- ✱ guaranteed bandwidth?
- ✱ preservation of inter-packet timing (no jitter)?
- ✱ loss-free delivery?
- ✱ in-order delivery?
- ✱ congestion feedback to sender?

service abstraction

The most important abstraction provided by network layer:

virtual circuit
or
datagram?

4

Virtual circuits

“source-to-dest path behaves much like telephone circuit”

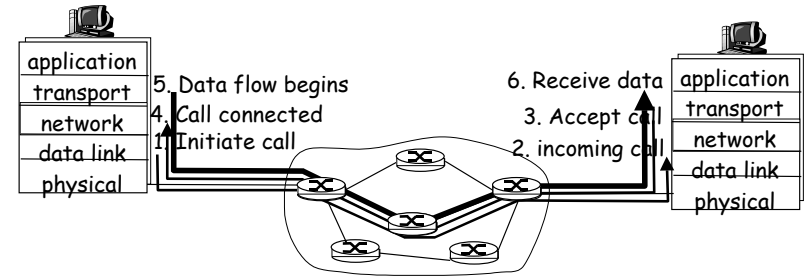
- performance-wise
- network actions along source-to-dest path

- ✱ call setup, teardown for each call *before* data can flow
- ✱ each packet carries VC identifier (not destination host ID)
- ✱ every router on source-dest path maintains “state” for each passing connection
 - transport-layer connection only involved two end systems
- ✱ link, router resources (bandwidth, buffers) may be *allocated* to VC
 - to get circuit-like perf.

5

Virtual circuits: signaling protocols

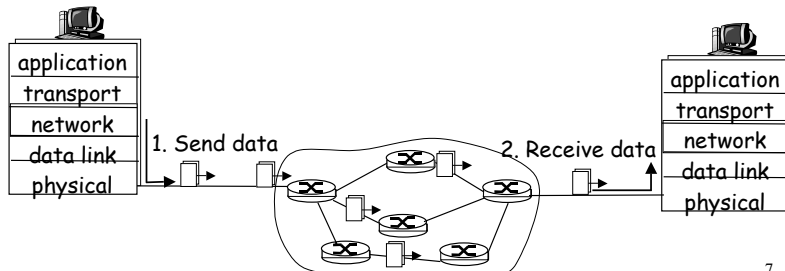
- ✱ used to setup, maintain, teardown VC
- ✱ used in ATM, frame-relay, X.25
- ✱ not used in today’s Internet



6

Datagram networks: the Internet model

- ✱ no call setup at network layer
- ✱ routers: no state about end-to-end connections
 - no network-level concept of “connection”
- ✱ packets typically routed using destination host ID
 - packets between same source-dest pair may take different paths



7

Datagram or VC network: why?

Internet

- ✱ data exchange among computers
 - “elastic” service, no strict timing req.
- ✱ “smart” end systems (computers)
 - can adapt, perform control, error recovery
 - simple inside network, complexity at “edge”
- ✱ many link types
 - different characteristics
 - uniform service difficult

ATM

- ✱ evolved from telephony
- ✱ human conversation:
 - strict timing, reliability requirements
 - need for guaranteed service
- ✱ “dumb” end systems
 - telephones
 - complexity inside network

8

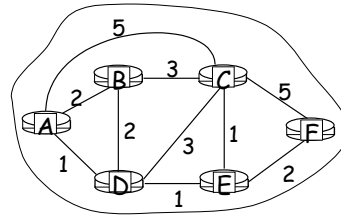
Routing

Routing protocol

Goal: determine "good" path (sequence of routers) thru network from source to dest.

Graph abstraction for routing algorithms:

- graph nodes are routers
- graph edges are physical links
 - link cost: delay, \$ cost, or congestion level



"good" path:
typically means
minimum cost path
other def's possible

9

Routing Algorithm classification

Global or decentralized information?

- Global:**
- all routers have complete topology, link cost info
 - "link state" algorithms
- Decentralized:**
- router knows physically-connected neighbors, link costs to neighbors
 - iterative process of computation, exchange of info with neighbors
 - "distance vector" algorithms

Static or dynamic?

- Static:**
- routes change slowly over time
- Dynamic:**
- routes change more quickly
 - periodic update
 - in response to link cost changes

10

A Link-State Routing Algorithm

Dijkstra's algorithm

- net topology, link costs known to all nodes
 - accomplished via "link state broadcast"
 - all nodes have same info
- computes least cost paths from one node ("source") to all other nodes
 - gives routing table for that node
- iterative: after k iterations, know least cost path to k dest.'s

11

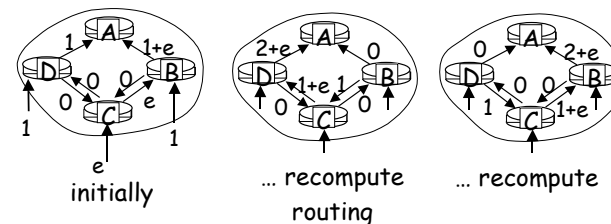
Dijkstra's algorithm, discussion

Algorithm complexity: n nodes

- each iteration: need to check all nodes, w, not in N
- $n(n+1)/2$ comparisons: $O(n^2)$
- more efficient implementations possible: $O(n \log n)$

Oscillations possible:

- e.g., link cost = amount of carried traffic



12

Distance Vector Routing Algorithm

iterative:

- continues until no nodes exchange info.
- self-terminating:** no "signal" to stop

asynchronous:

- nodes need *not* exchange info/iterate in lock step!

distributed:

- each node communicates *only* with directly-attached neighbors

Distance Table data structure

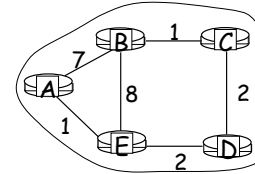
- each node has its own
- row for each possible destination
- column for each directly-attached neighbor to node
- example: in node X, for dest. Y via neighbor Z:

$$D^X(Y,Z) = \text{distance from X to Y, via Z as next hop}$$

$$= c(X,Z) + \min_w \{D^Z(Y,w)\}$$

13

Distance Table: example



$$D^E(C,D) = c(E,D) + \min_w \{D^D(C,w)\}$$

$$= 2 + 2 = 4$$

From node E: to go to C
What is the cost thru D?

D ^E ()	Via		
	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

14

Distance table gives routing table

D ^E ()	cost to destination via		
	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

destination	Outgoing link to use, cost	
	link	cost
A	A	1
B	D	5
C	D	4
D	D	2

Distance table → Routing table

15

Distance Vector Routing: overview

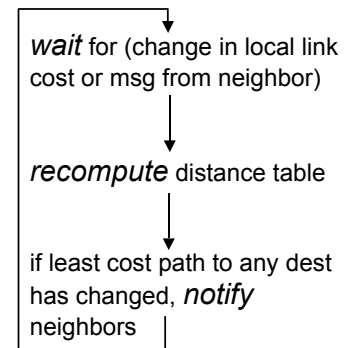
Iterative, asynchronous: each local iteration caused by:

- local link cost change
- message from neighbor: its least cost path change from neighbor

Distributed:

- each node notifies neighbors *only* when its least cost path to any destination changes
 - neighbors then notify their neighbors if necessary

Each node:

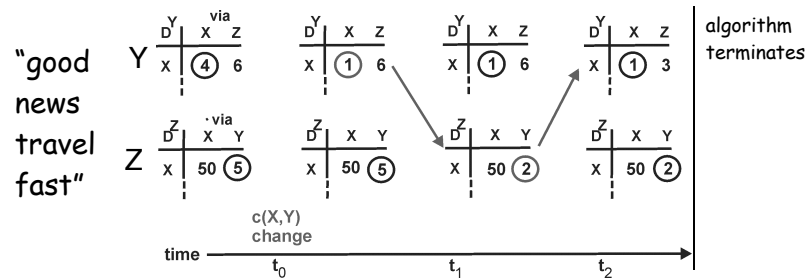
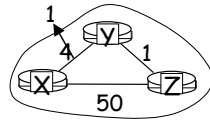


16

Distance Vector: link cost changes

Link cost changes:

1. node detects local link cost change
- updates distance table (line 15)
2. if cost change in least cost path, notify neighbors (lines 23,24)

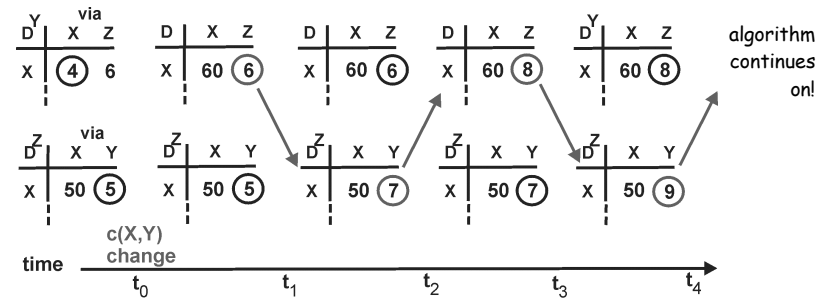
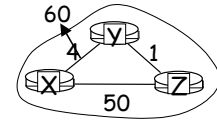


17

Distance Vector: link cost changes

Link cost changes:

bad news travels slow - "count to infinity" problem!



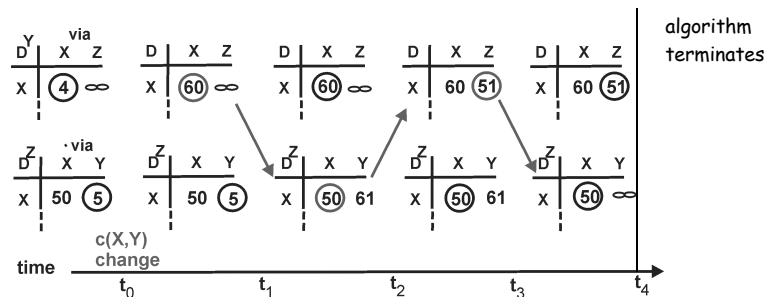
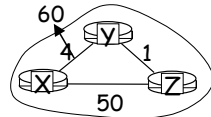
18

Distance Vector: poisoned reverse

If Z routes through Y to get to X :

Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z)

will this completely solve count to infinity problem?



19

Summary: Distributed Routing Techniques

Link State

- Topology information is **flooded** within the routing domain
- Best end-to-end paths are computed locally at each router.
- Best end-to-end paths determine next-hops.
- Advertises: link info
- Works only if policy is shared and uniform
- Examples: OSPF, IS-IS

Vectoring

- Each router knows little about network topology
- Only best next-hops are chosen by each router for each destination network.
- Best end-to-end paths result from composition of all next-hop choices
- Advertises: path info, distance
- Does not require uniform policies at all routers
- Examples: RIP, BGP

20

Comparison of LS and DV algorithms

Message complexity

- ☼ **LS:** with n nodes, E links, $O(nE)$ msgs sent each round
- ☼ **DV:** exchange between neighbors only periodically

Speed of Convergence

- ☼ **LS:** $O(n^2)$ algorithm
 - may have oscillations
- ☼ **DV:** convergence time varies
 - may have routing loops
 - count-to-infinity problem

Robustness: what happens if router malfunctions?

LS:

- node can advertise incorrect *link* cost
- each node computes only its *own* table

DV:

- DV node can advertise incorrect *path* cost
- each node's table used by others
 - errors propagate thru network

21

Hierarchical Routing

Our routing study thus far - idealization
all routers identical
network “flat”
... *not* true in practice

scale: with 50 million destinations:

- ☼ can't store all dest's in routing tables!
- ☼ routing table exchange would swamp links!

administrative autonomy

- ☼ internet = network of networks
- ☼ each network admin may want to control routing in its own network

22

Hierarchical Routing

- ☼ aggregate routers into regions, “autonomous systems” (AS)
- ☼ routers in same AS run same routing protocol
 - “intra-AS” routing protocol
 - routers in different AS can run different intra-AS routing protocol

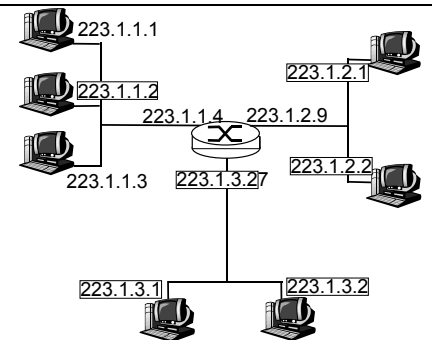
gateway routers

- ☼ special routers in AS
- ☼ run intra-AS routing protocol with all other routers in AS
- ☼ also responsible for routing to destinations outside AS
 - run *inter-AS* routing protocol with other gateway routers

23

IP Addressing: introduction

- ☼ IP address: 32-bit identifier for host, router *interface*
- ☼ *interface*: connection between host, router and physical link
 - router's typically have multiple interfaces
 - host may have multiple interfaces
 - IP addresses associated with interface, not host, router



223.1.1.1 = 11011111 00000001 00000001 00000001
223 1 1 1

24

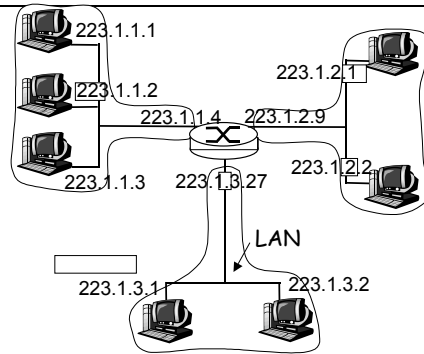
IP Addressing

IP address:

- network part (high order bits)
- host part (low order bits)

What's a network ? (from IP address perspective)

- device interfaces with same network part of IP address
- can physically reach each other without intervening router



network consisting of 3 IP networks
(for IP addresses starting with 223,
first 24 bits are network address)

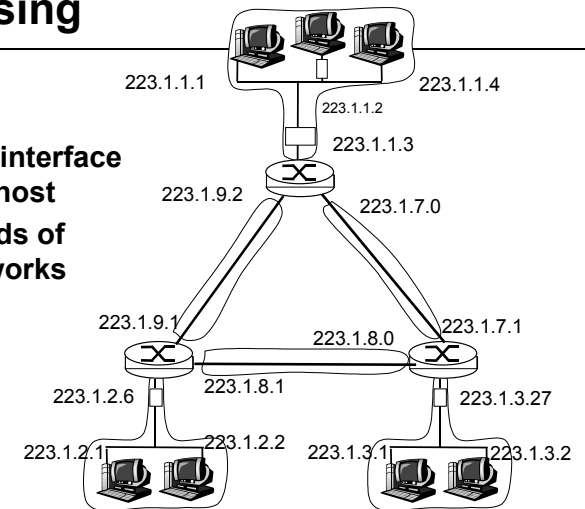
25

IP Addressing

How to find the networks?

- Detach each interface from router, host
- create "islands of isolated networks"

Interconnected
system consisting
of six networks



26

IP Addresses

given notion of "network", let's re-examine IP addresses:

"class-full" addressing:

class

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

← 32 bits →

27

IP addressing: need for change

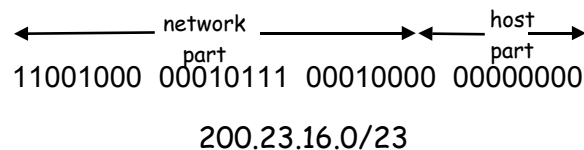
classful addressing:

- inefficient use of address space, address space exhaustion
- e.g., class B net allocated enough addresses for 65K hosts, even if only 2K hosts in that network

28

IP addressing: CIDR

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



29

IP Addresses and Prefixes

- ☀ IP addresses have 32 bits: 4 octets of bits (IPv4)
- ☀ A prefix is a group of IP addresses
- ☀ 128.32.101.5 is an IP address (32 bits)
- ☀ 128.32.0.0/16 is a prefix of the 16 first bits:
 - 128.32.0.0 – 128.32.255.255 (2^{16} addresses)
- ☀ 128.32.4.0/24 is a prefix of the 24 first bits - longer

30

IP addresses: how to get one?

Hosts (host portion):

- ☀ hard-coded by system admin in a file
- ☀ **DHCP: Dynamic Host Configuration Protocol:**
dynamically get address: “plug-and-play”
 - host broadcasts “DHCP discover” msg
 - DHCP server responds with “DHCP offer” msg
 - host requests IP address: “DHCP request” msg
 - DHCP server sends address: “DHCP ack” msg

31

IP addresses: how to get one?

Network (network portion):

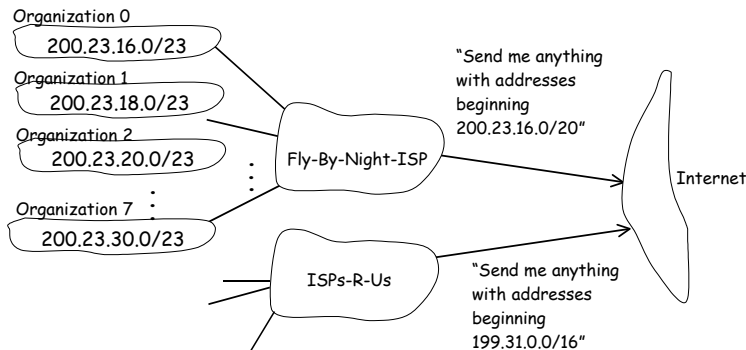
- ☀ get allocated portion of ISP's address space:

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

32

Hierarchical addressing: route aggregation

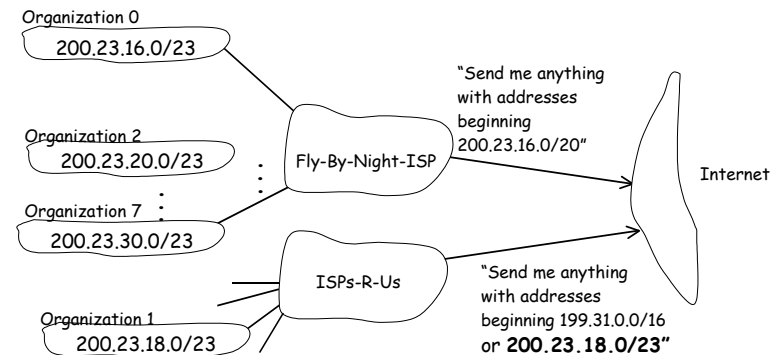
Hierarchical addressing allows efficient advertisement of routing information:



33

Hierarchical addressing: more specific routes

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



34

Routing is Based on Prefixes

- ☛ A BGP Routing table has prefixes for entries
- ☛ For a IP address of a packet, find longest match
- ☛ Example: packet IP 128.32.101.1
- ☛ Matching:
- ☛ 128.1.1.4 matches the first 8 bits – no match!
- ☛ 128.32.0.0/16 match for 16 bits
- ☛ 128.32.101.0/24 is a longer match

35

Prefix Matching in More Detail

- ☛ For a IP address of a packet, find longest match
- ☛ Example: Compare
 - packet IP 128.32.101.1
 - With 128.32.0.0/16
 - IP : 01000000. 00100000. 01100101 .00000001
 - Mask : 11111111. 11111111. 00000000 .00000000
 - AND : 01000000. 00100000. 00000000 .00000000
 - Prefix : 01000000. 00100000. 00000000. 00000000
 - Equal? Yes
 - We have a match of length 16

36

IP addressing: the last word...

Q: How does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers

- allocates addresses
- manages DNS
- assigns domain names, resolves disputes

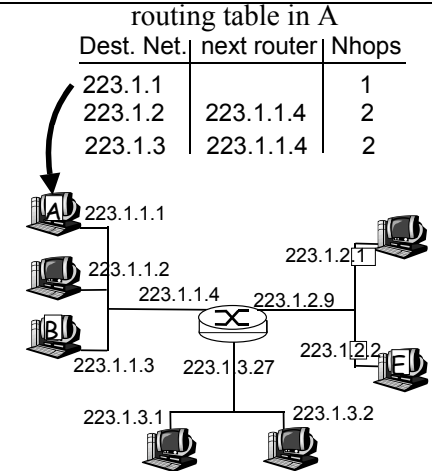
37

Getting a datagram from source to dest.

IP datagram:

misc fields	source IP addr	dest IP addr	data
-------------	----------------	--------------	------

datagram remains unchanged, as it travels source to destination
addr fields of interest here



38

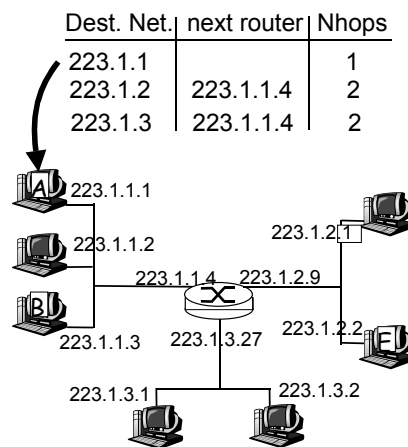
Getting a datagram from source to dest.

misc fields	223.1.1.1	223.1.1.3	data
-------------	-----------	-----------	------

Starting at A, given IP datagram addressed to B:

look up net. address of B
find B is on same net. as A
link layer will send datagram directly to B inside link-layer frame

B and A are directly connected



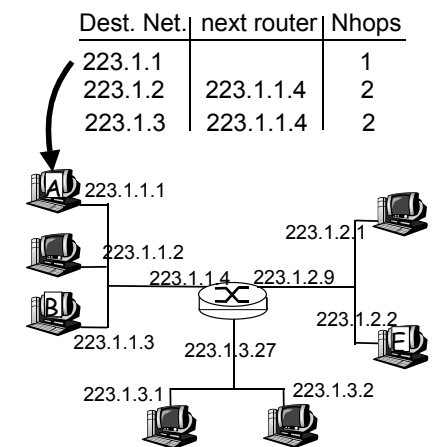
39

Getting a datagram from source to dest.

misc fields	223.1.1.1	223.1.2.2	data
-------------	-----------	-----------	------

Starting at A, dest. E:
look up network address of E
E on *different* network

A, E not directly attached
routing table: next hop router to E is 223.1.1.4
link layer sends datagram to router 223.1.1.4 inside link-layer frame
datagram arrives at 223.1.1.4
continued.....



40

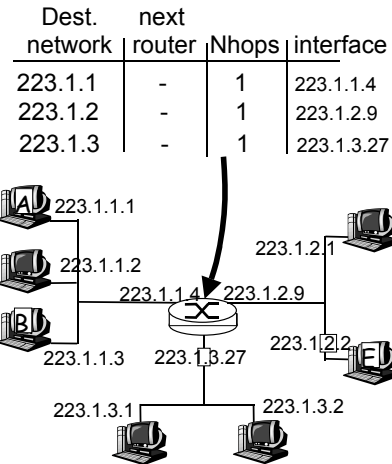
Getting a datagram from source to dest.

misc	223.1.1.1	223.1.2.2	data
fields			

Arriving at 223.1.4, destined for 223.1.2.2

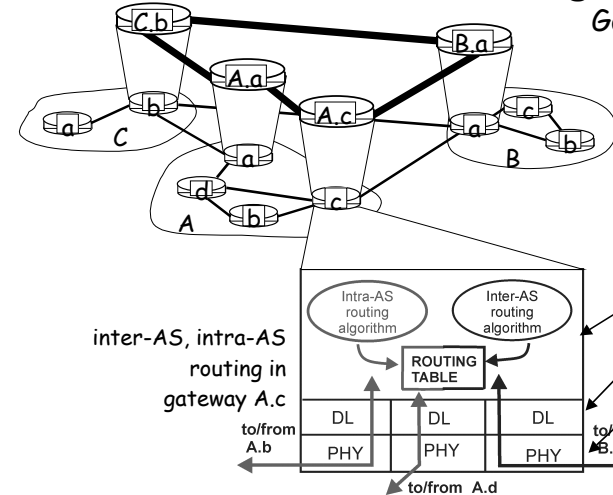
look up network address of E
E on *same* network as router's
interface 223.1.2.9

router, E directly attached
link layer sends datagram to 223.1.2.2
inside link-layer frame via interface
223.1.2.9
datagram arrives at 223.1.2.2!!!
(hooray!)



41

Intra-AS and Inter-AS routing



Gateways:

- perform inter-AS routing amongst themselves
- perform intra-AS routers with other routers in their AS

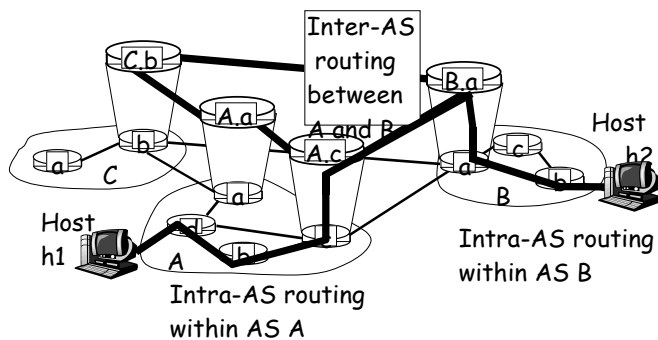
network layer

link layer

physical layer

42

Intra-AS and Inter-AS routing

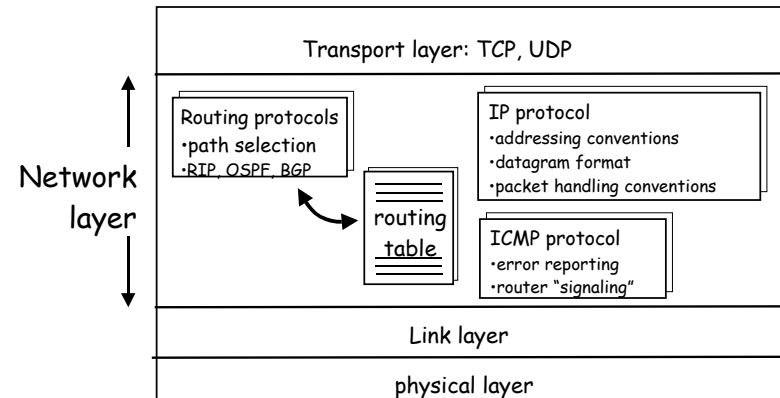


We'll examine specific inter-AS and intra-AS Internet routing protocols shortly

43

The Internet Network layer

Host, router network layer functions:



44

Measurements in the Internet

- ☼ **Difficulties in measuring**
- ☼ **Measuring tools (traceroute)**
- ☼ **Misc issues**

45

Measuring and Modeling Is not Easy

- ☼ **Constantly changing environment**
- ☼ **How much data is enough**
 - Recently: we need to measure more than 24h!
- ☼ **How frequently should I be measuring?**
- ☼ **Are the measurements representative?**

46

Operation versus Measurements

- ☼ **Operators do not care about**
 - Measurements
 - Academic Research
- ☼ **Why?**
 - Takes away resources
 - Can create problems
 - Complicates their lives
- ☼ **Luckily, there are measurement centers**
 - CAIDA, NLANR, routeviews, RIPE

47

Types of Measurement Tools

- ☼ **Application level:**
 - Install application agents at two measuring entries
 - REALITI tool from UCR
 - More control over process
- ☼ **Network level:**
 - Use the Internet control functionality (ICMP)
 - Trick the network to provide information

48

Ping: the tool

- ✱ Uses ICMP ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway
- ✱ Reports
 - Round trip time
 - Packets loss
- ✱ Many available options: packet type, size etc
- ✱ Limitation: >1sec measurement frequency
- ✱ Read manual: man ping

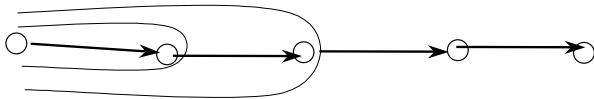
49

Traceroute: the tool

- ✱ Traceroute measures
 - the path and the round trip time
- ✱ Traceroute: ingenious (ab)use of the network layer by Van Jacobson
- ✱ Main ideas:
 - send “bad” packets to receive ICMP: “packet died”
 - Recursive probing to identify the path
 - Send three packets at a time
- ✱ Read manual: man traceroute

50

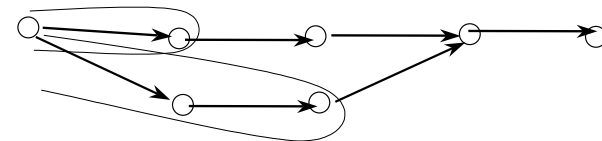
The ingenuity of traceroute



- ✱ Send a packet for every hop of the path
- ✱ Set TTL = 1, packet expires, ICMP returns
- ✱ Increase TTL by one, and repeat
- ✱ At the destination, port number is wrong: return an ICMP packet, port not found

51

Traceroute: Some Limitations



- ✱ In traceroute, you may be exploring multiple paths without knowing it
- ✱ Delays for each part of the path correspond to different measurements: ie they don't sum up

52

Identifying The Router Topology

☼ Several efforts rely on multiple traceroute

- Govindan et al INFOCOM 2000
- Cheswick and Burch Internet Mapping Project

☼ Main idea:

- Do thousands of traceroutes
- Collect all adjacent nodes
- Generate a graph

53

End

Router Graphs: A Complication

☼ Routers have multiple IP addresses

- One for each interface

☼ How do we resolve this?

☼ Only heuristics exist [Govindan]

☼ Heuristic: Send packets to one interface and hope that they will respond with the other interface

54

Proposals

☼ Overall: decent

☼ Not enough motivation/background

- All related papers

☼ Not enough thought on what you will do

- Spend an evening thinking what you will do and how

☼ Try to clarify goals early and talk to me

☼ Photocopy them, and give me the original

- It's the contract

55

56

Practical Tips

- ☀ **The earlier, the better**
- ☀ **Talk to me early**
- ☀ **Look for tools, data, previous work**
 - It can save you a lot of time in the long run
- ☀ **Try to focus on a topic**

57

Side Note

- ☀ **Important things in research:**
 - Asking the right questions
 - Identifying the right topic
 - Context of research
 - *Motivation*
 - *Previous and related work*
 - *Importance of problem*
 - Thoroughness of work

58