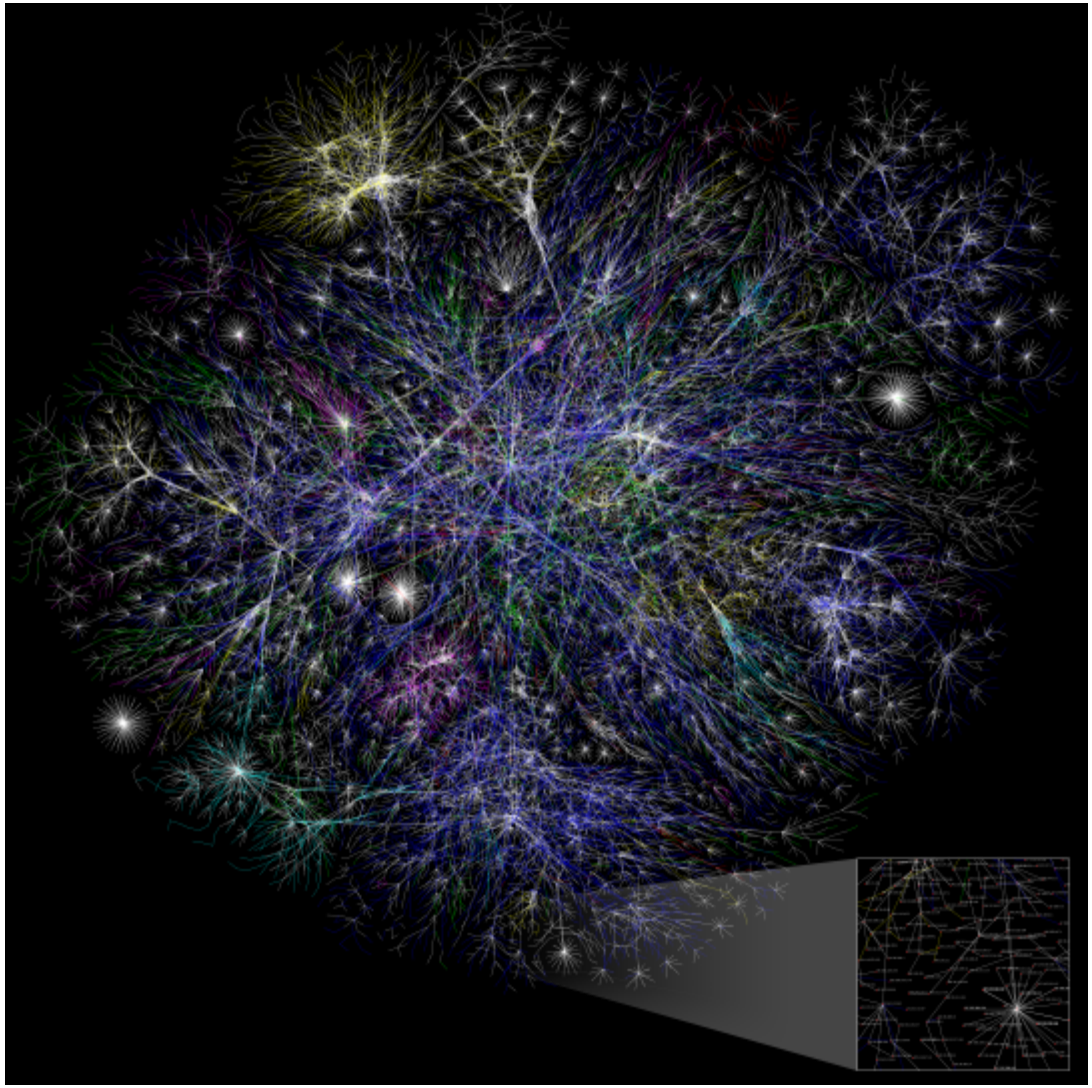
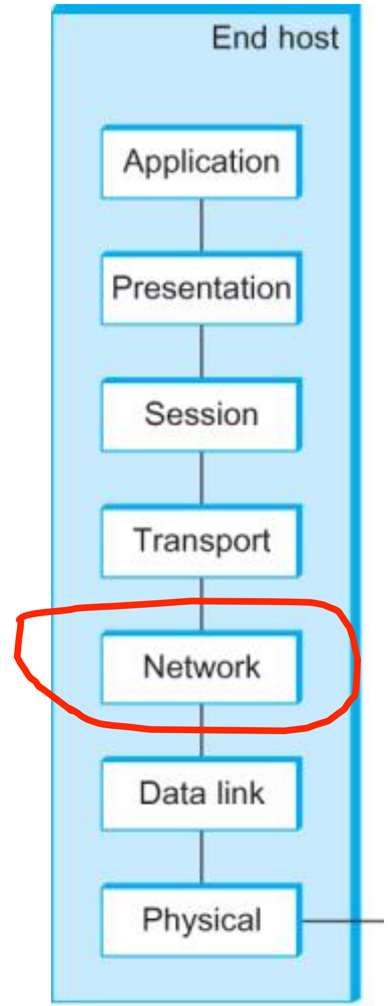
Routing and error reporting



Overview

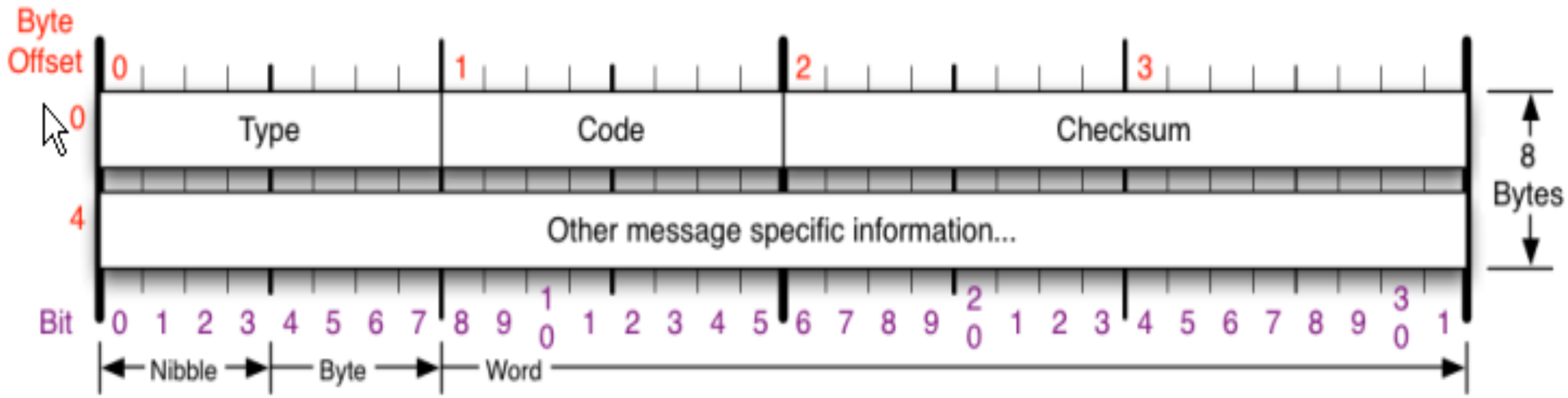
- Network error reporting
  - ICMP
- Inside a router
  - Routing versus forwarding
- Selecting a path
  - Given a known topology
- Learning the topology
  - How routers talk to each other



Network error reporting

- Internet Control Message Protocol (ICMP)
  - Rides on top of IP (like TCP/UDP)
  - Error messages sent back to host by routers
  - ICMP used by some user utilities:
    - traceroute
    - ping

ICMP



ICMP Message Types

Type	Code/Name
0	Echo Reply
3	Destination Unreachable
	0 Net Unreachable
	1 Host Unreachable
	2 Protocol Unreachable
	3 Port Unreachable
	4 Fragmentation required, and DF set
	5 Source Route Failed
	6 Destination Network Unknown
	7 Destination Host Unknown
	8 Source Host Isolated
	9 Network Administratively Prohibited
	10 Host Administratively Prohibited
	11 Network Unreachable for TOS

Type	Code/Name
3	Destination Unreachable (continued)
	12 Host Unreachable for TOS
	13 Communication Administratively Prohibited
4	Source Quench
5	Redirect
	0 Redirect Datagram for the Network
	1 Redirect Datagram for the Host
	2 Redirect Datagram for the TOS & Network
	3 Redirect Datagram for the TOS & Host
8	Echo
9	Router Advertisement
10	Router Selection

Type	Code/Name
11	Time Exceded
	0 TTL Exceeded
	1 Fragment Reassembly Time Exceeded
12	Parameter Problem
	0 Pointer Problem
	1 Missing a Required Operand
	2 Bad Length
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply
30	Traceroute

Checksum

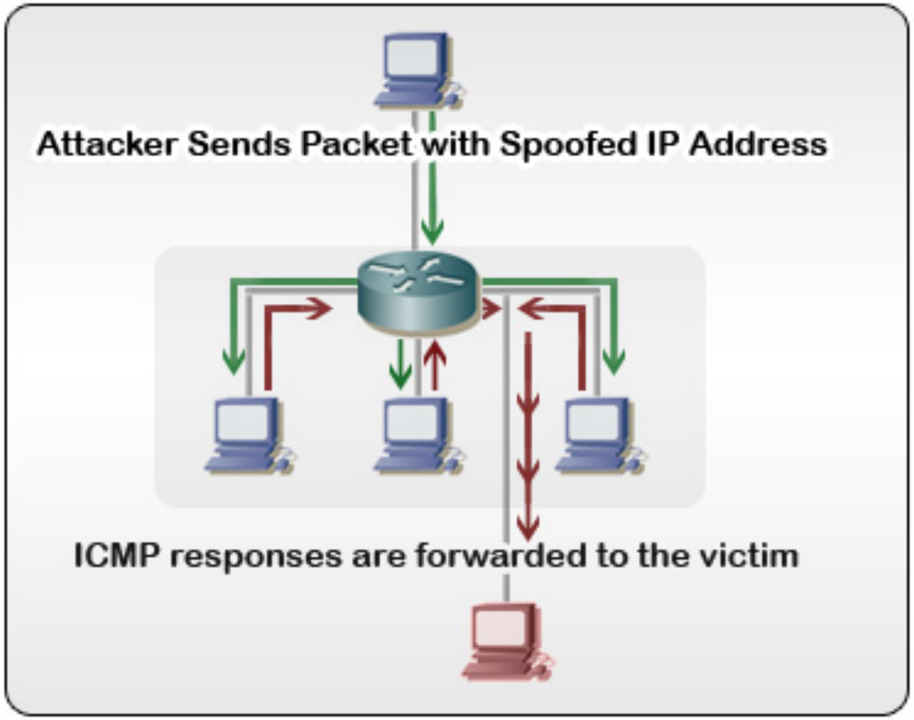
Checksum of ICMP header

RFC 792

Please refer to RFC 792 for the Internet Control Message protocol (ICMP) specification.

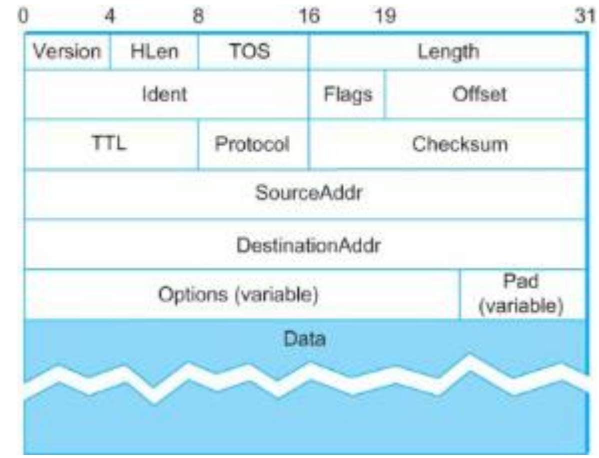
Smurf Attack

- Denial-of-Service attack
  - Attacker sends stream of ICMP echo request s
  - Sent to network broadcast address
  - Uses spoofed IP of victim
  - Generates large amounts of traffic on target network



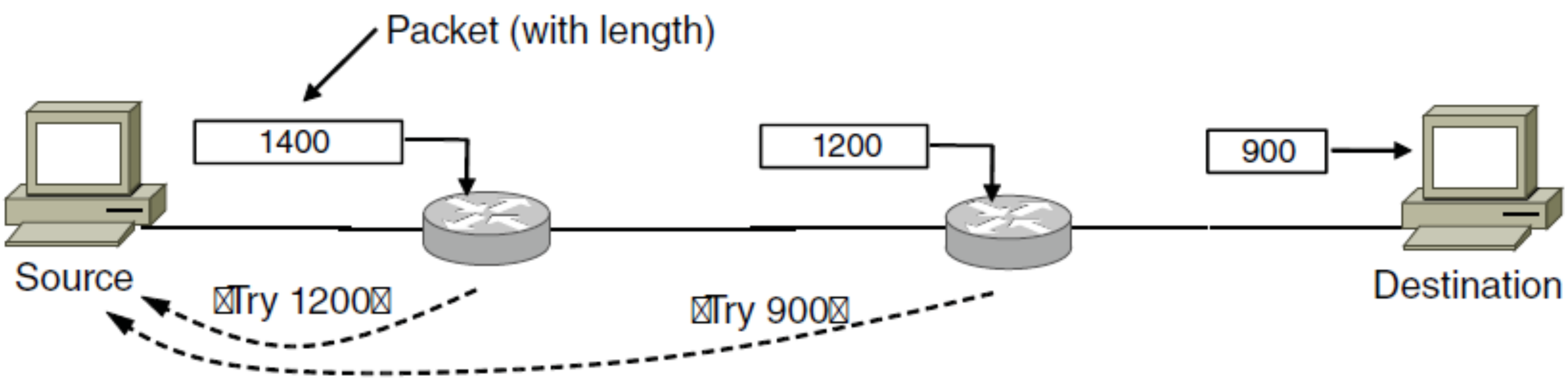
Path MTU discovery

- Set Don't Fragment (DF) bit in IP packet flags
- Any router with < MTU
  - Drop packet
  - Send back ICMP Fragmentation Required with MTU size
  - Host can then reduce its packet size



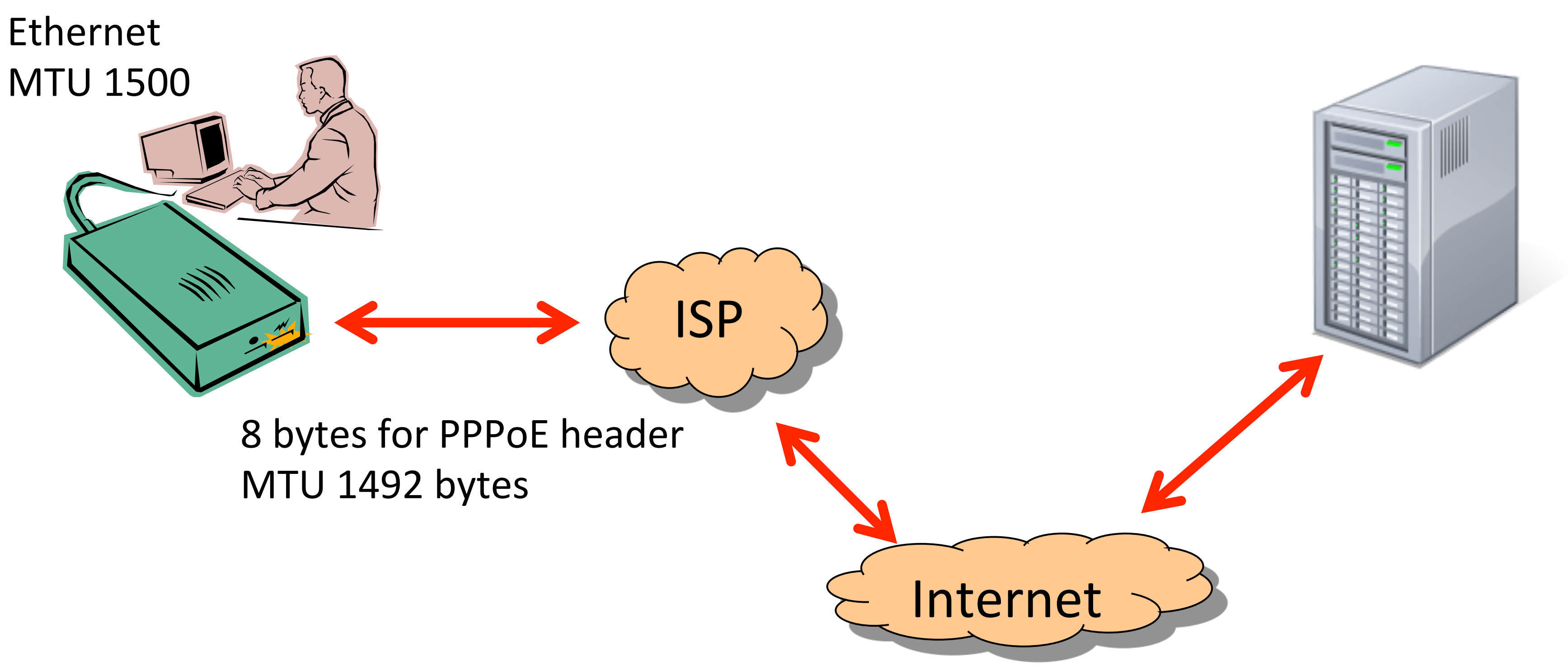
- Problems:
  - Some routers don't generate ICMP messages
  - Intermediate firewalls may filter ICMP messages

Path MTU discovery: success



1) Source sends off a 1400 byte message to destination with Do Not Fragment bit set.
2) First router refuses to send since its next hop MTU is 1200. Sends back ICMP message saying to use 1200.
3) Source sends 1200 byte message, second router rejects since its next hop MTU is 900.
4) Source sends a 900 byte message.

Path MTU discovery: failure

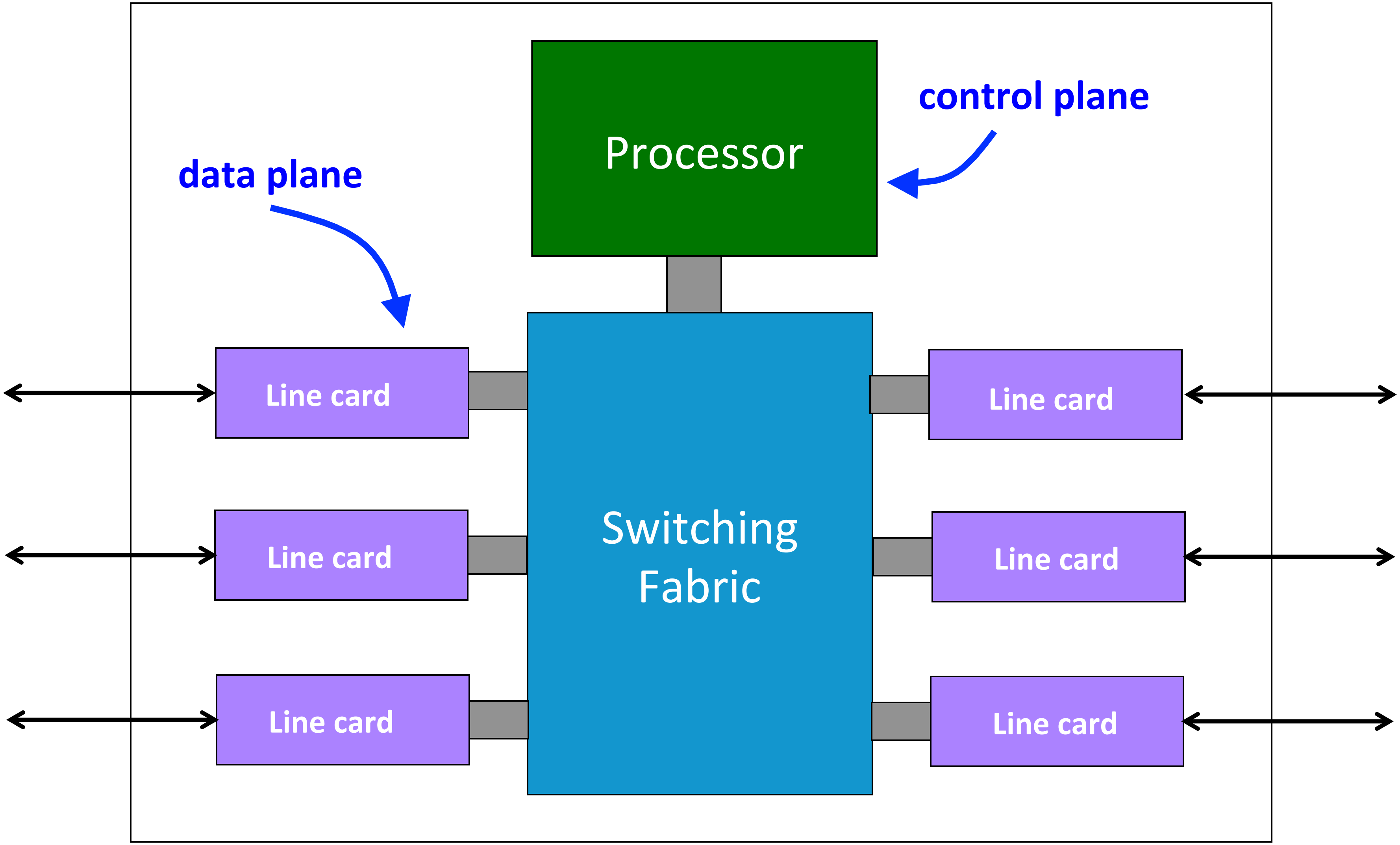


1) User sends a short packet requesting a web page.
2) Web server responds with a large 1500-byte packet.
3) ISP drops packet since > MTU, sends back an ICMP saying to use 1492 bytes.
4) ICMP gets filtered out somewhere or web server misconfigured.
5) Server eventually times out, resends 1500-byte packet
…

Forwarding vs. Routing

- Forwarding: data plane
  - Which outgoing link to place a packet
  - Router *uses* a forwarding table
- Routing: control plane
  - Computing paths for packets to follow
  - Routers communicate amongst themselves
  - Router *creates* a forwarding table

Data and Control Planes



Forwarding tables

- Forwarding tables
  - Map IP prefix to outgoing link
  - Optimized for fast lookup
- Entries could be statically configured
  - e.g. map 69.123.102.0/24 to link 3
- But what if:
  - Equipment fails
  - Equipment is added
  - A link becomes congested

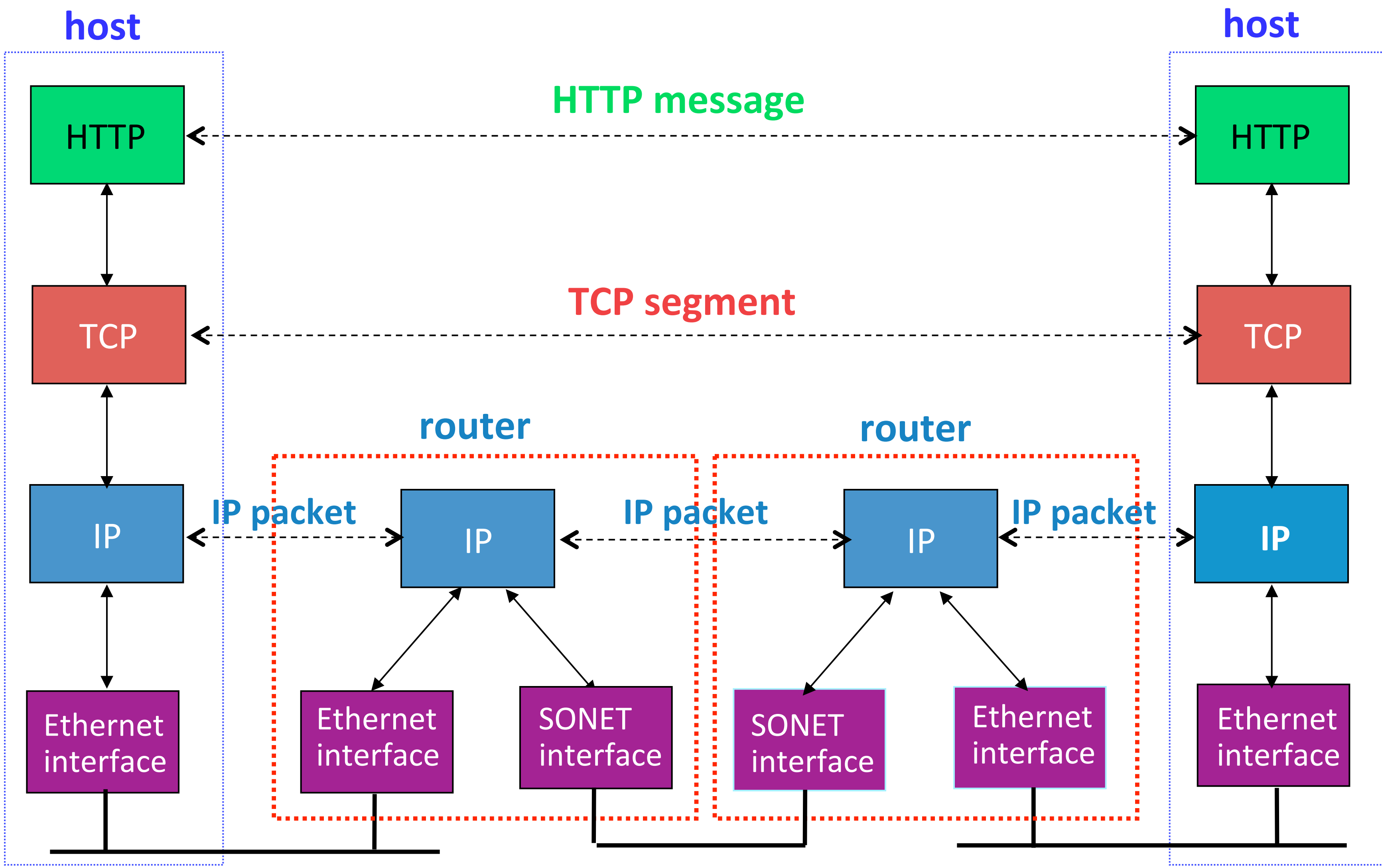
Prefix/Length	Interface	MAC Address
18/8	if0	8:0:2b:e4:b:1:2

Routing tables

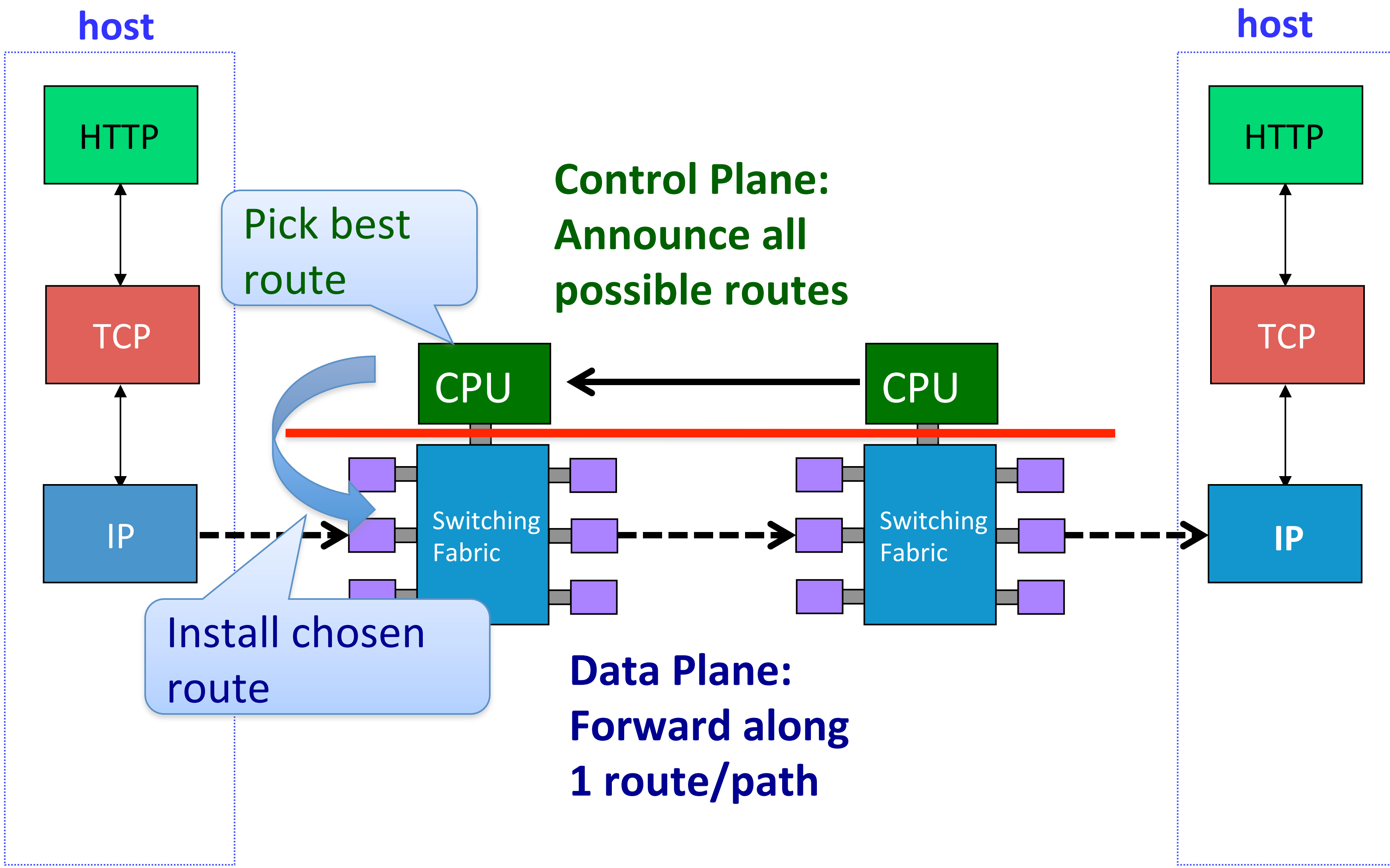
- Routing table:
  - Which router can serve a given IP prefix
  - What outgoing link reach that router
  - Perhaps metrics associated with routes
  - Represents the network topology
  - Used to build the forwarding table

Prefix/Length	Next Hop
18/8	171.69.245.10

Internet layering model

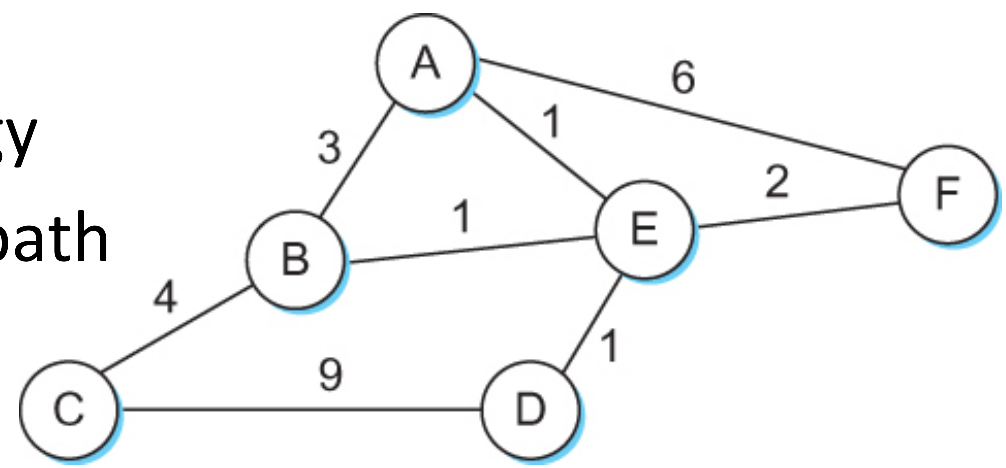


Internet layering model



Network as a graph

- Nodes:
  - Hosts, switches, routers, networks
- Edges:
  - Network links
  - May have an associated cost
- Basic problems:
  - Learning the topology
  - Finding lowest cost path



Routing protocols

- Distributed algorithm
  - Running on many devices
  - No central authority
  - Must deal with changing topology
- Two main classes for intradomain routing:
  - Distance vector routing
    - aka Bellman-Ford algorithm
    - Routing Information Protocol (RIP)
  - Link state routing
    - Open Shortest Path First Protocol (OSPF)

Distance vector routing

Distance vector routing

- Each node maintains state
  - Cost of direct link to each of your neighbors
  - Least cost route known to all destinations
- Routers send periodic updates
  - Send neighbor your array
  - When you receive an update from your neighbor
    - Update array entries if new info provides shorter route
  - Converges quickly (if no topology changes)

Distance vector example: step 1

Optimum 1-hop paths

Table for A			Table for B		
Dst	Cst	Hop	Dst	Cst	Hop
A	0	A	A	4	A
B	4	B	B	0	B
C	∞	–	C	∞	–
D	∞	–	D	3	D
E	2	E	E	∞	–
F	6	F	F	1	F

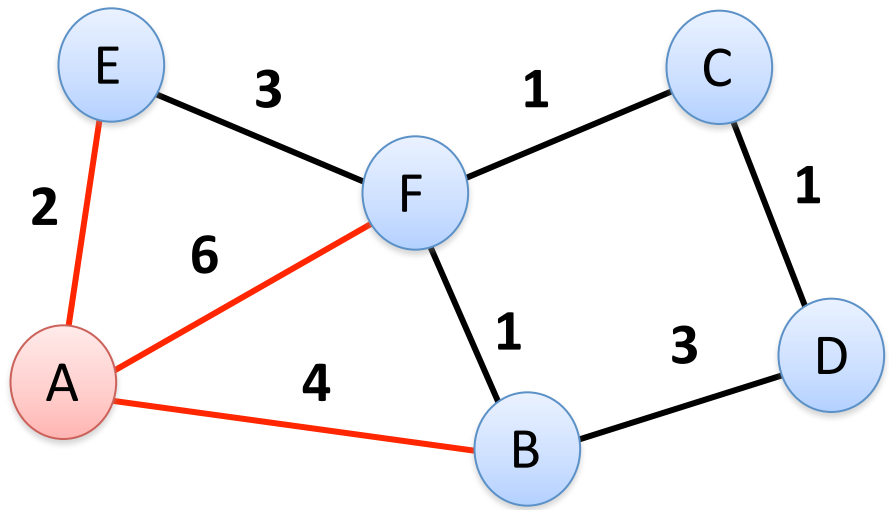


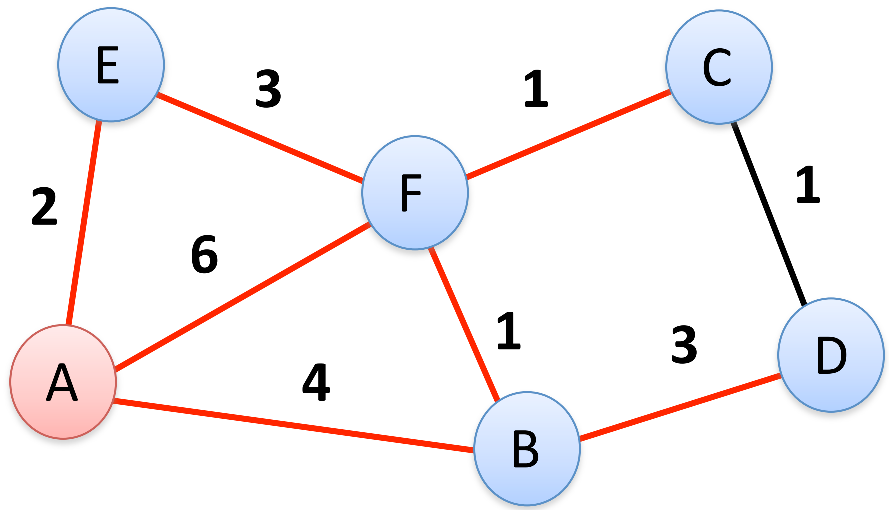
Table for C			Table for D			Table for E			Table for F		
Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop
A	∞	–	A	∞	–	A	2	A	A	6	A
B	∞	–	B	3	B	B	∞	–	B	1	B
C	0	C	C	1	C	C	∞	–	C	1	C
D	1	D	D	0	D	D	∞	–	D	∞	–
E	∞	–	E	∞	–	E	0	E	E	3	E
F	1	F	F	∞	–	F	3	F	F	0	F

Distance vector example: step 2

Optimum 2-hop paths

Table for A			Table for B		
Dst	Cst	Hop	Dst	Cst	Hop
A	0	A	A	4	A
B	4	B	B	0	B
C	7	F	C	2	F
D	7	B	D	3	D
E	2	E	E	4	F
F	5	E	F	1	F

Table for C			Table for D			Table for E			Table for F		
Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop
A	7	F	A	7	B	A	2	A	A	5	B
B	2	F	B	3	B	B	4	F	B	1	B
C	0	C	C	1	C	C	4	F	C	1	C
D	1	D	D	0	D	D	∞	–	D	2	C
E	4	F	E	∞	–	E	0	E	E	3	E
F	1	F	F	2	C	F	3	F	F	0	F



Distance vector example: step 3

Optimum 3-hop paths

Table for A			Table for B		
Dst	Cst	Hop	Dst	Cst	Hop
A	0	A	A	4	A
B	4	B	B	0	B
C	6	E	C	2	F
D	7	B	D	3	D
E	2	E	E	4	F
F	5	E	F	1	F

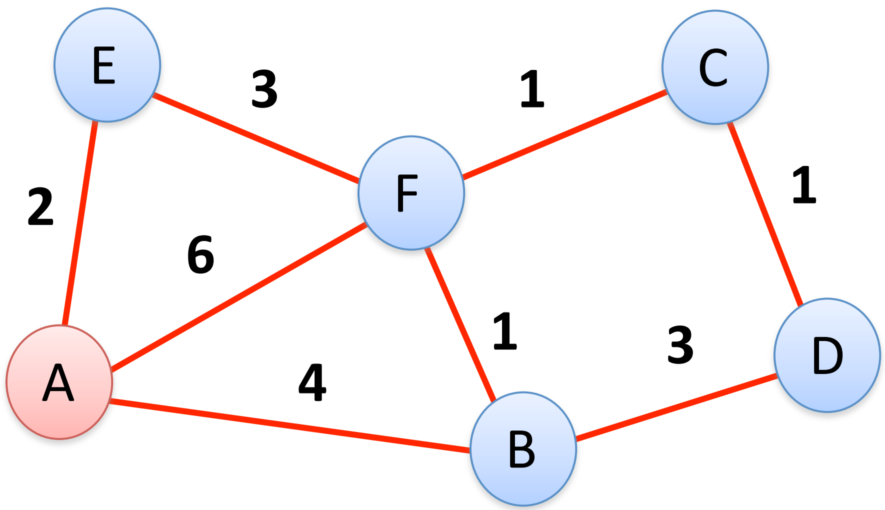
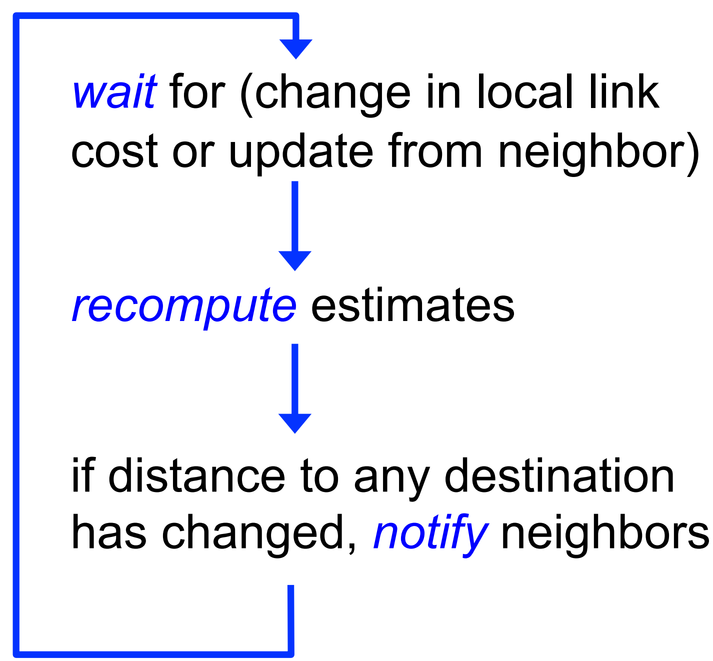


Table for C			Table for D			Table for E			Table for F		
Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop
A	6	F	A	7	B	A	2	A	A	5	B
B	2	F	B	3	B	B	4	F	B	1	B
C	0	C	C	1	C	C	4	F	C	1	C
D	1	D	D	0	D	D	5	F	D	2	C
E	4	F	E	5	C	E	0	E	E	3	E
F	1	F	F	2	C	F	3	F	F	0	F

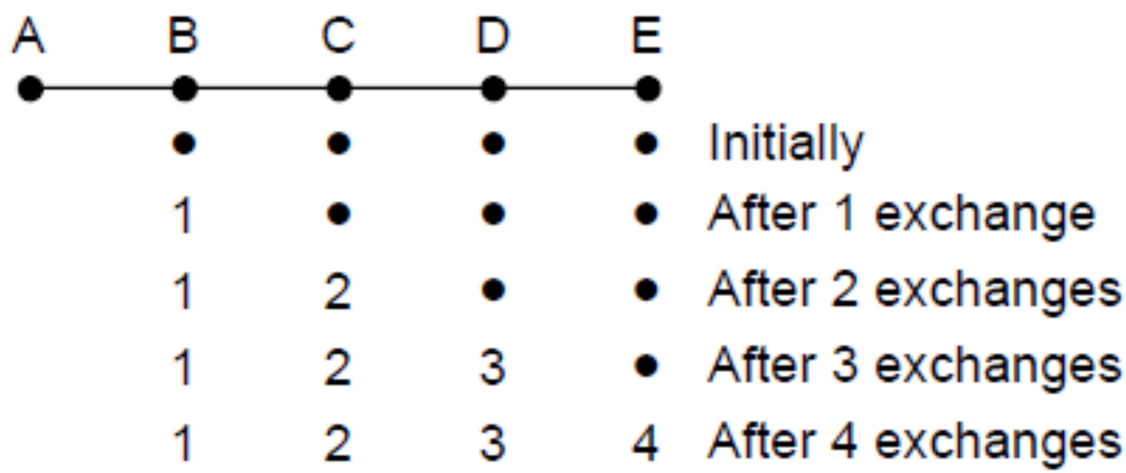
Distance vector updates

- Periodic updates
  - Automatically send update every so often
  - Lets other nodes know you are alive
- Triggered updates



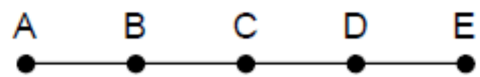
Link cost change

- What if link added or cost reduced?
  - Update propagates from point of change
  - Network with longest path of N hops:
    - N exchanges, everyone knows of new/improved link
  - "Good news travels fast"



Link cost change

- What if link deleted or cost increased?
  – Problem: Neighbor has a path somewhere, but you don't know if it goes through you
- Count to infinity problem
  – "Bad news travels slow"



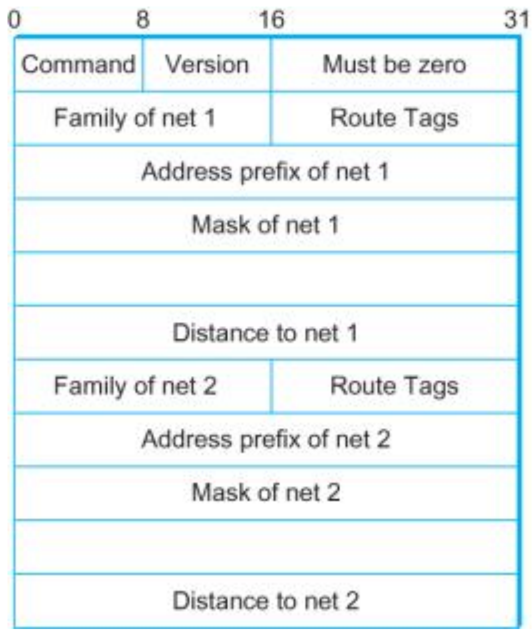
B	C	D	E	
1	2	3	4	Initially
3	2	3	4	After 1 exchange
3	4	3	4	After 2 exchanges
5	4	5	4	After 3 exchanges
5	6	5	6	After 4 exchanges
7	6	7	6	After 5 exchanges
7	8	7	8	After 6 exchanges
	⋮			
•	•	•	•	

Count-to-infinity

- Various ways to "fix":
  - – Use a small values for infinity, e.g. 16
    - Limits network size to 15 hops
  - – Split horizon with poisoned reverse
    - Track where you learned the route
    - e.g. (E, 2, A), I learned a cost 2 route to E from A
    - When B updates A, sends (E, ∞)
    - Only works for two node routing loops
  - – Holddown timer
    - Start a timer when a network becomes unreachable
    - Don't update until timer expires

RIP

- Routing Information Protocol (RIP)
  - Distance-vector protocol
  - Used in original ARPANET
  - All links costs 1
  - Advertise every 30 seconds
    - Can cause a lot of traffic
  - Small networks, < 16 hops
    - An Interior Gateway Protocol (IGP)
  - Runs over UDP



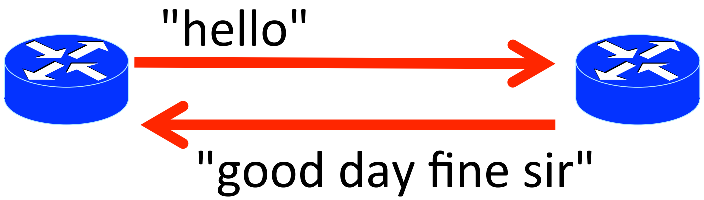
Link state routing

Link state routing

- Link state routing
  - Second major class of intradomain routing
  - Each router tracks its immediate links
    - Whether up or down
    - Cost of link
  - Each router broadcasts link state
    - Information disseminated to all nodes
    - Routers have global state from which to compute path
  - e.g. Open Shortest Path First (OSPF)

1. Learning about your neighbors

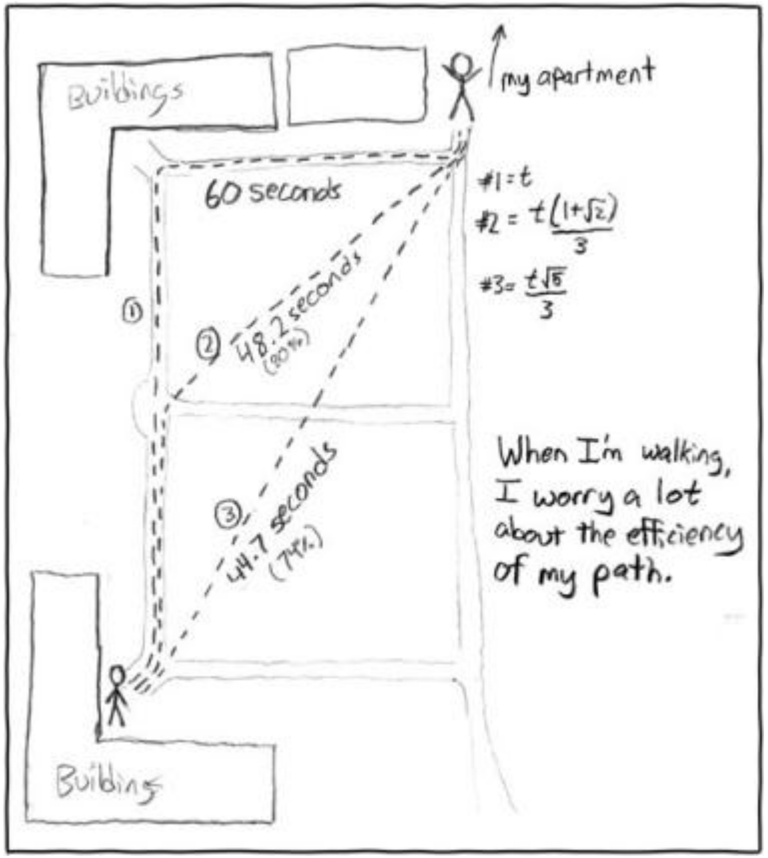
- Beaconing
  - Find out about your neighbors when you boot
  - Send periodic "hello" messages to each other
  - Detect a failure after several missed "hellos"



- Beacon frequency is tradeoff:
  - Detection speed
  - Bandwidth and CPU overhead
  - Likelihood of false detection

2. Setting link costs

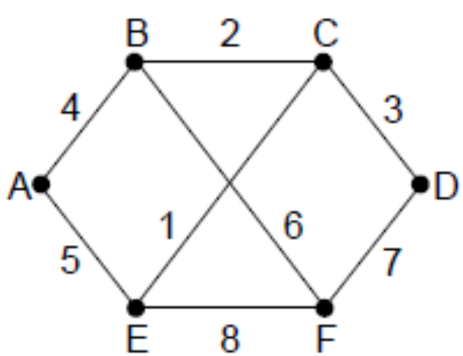
- Assign a link cost for each outbound link
  - Manual configuration
  - Automatic
    - Inverse of link bandwidth
      - 1-Gbps cost 1
      - 100-Mbps cost 10
    - Measure latency by sending an ECHO packet



http://xkcd.com/85/

3. Building link state packets

- Package info into a Link State Packet (LSP)
  - Identity of sender
  - List of neighbors
  - Sequence number of packet
  - Age of packet

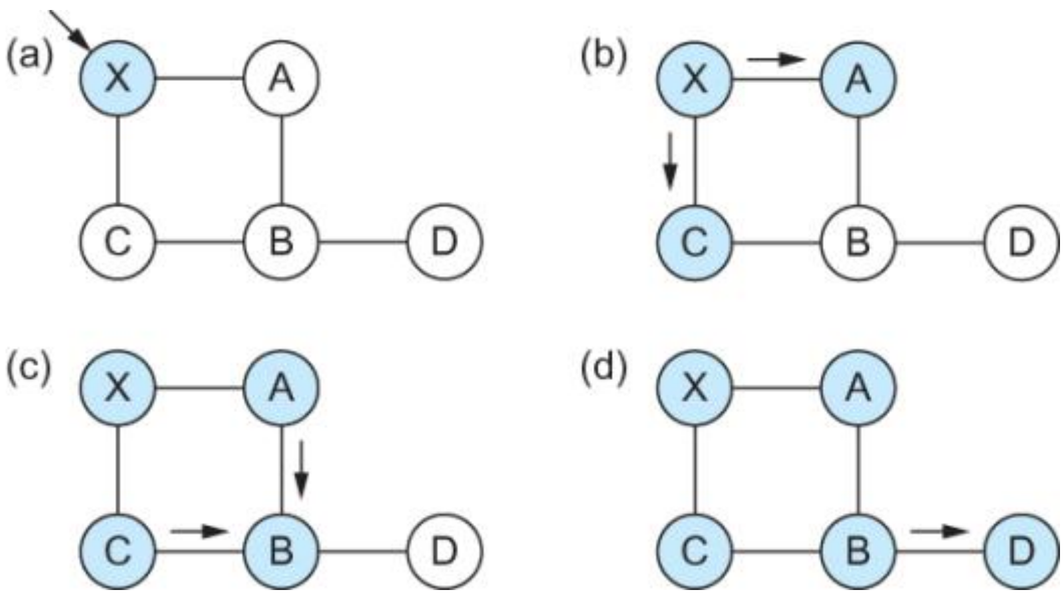


Link State Packets

A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

4. Distributing link state

- Flooding
  - Send your LSP out on all links
  - Next node sends LSP onward using its links
    - Except for link it arrived on



a) LSP arrives at node X
b) X floods LSP to A and C
c) A and C flood LSP to B (but not X)
d) flooding complete

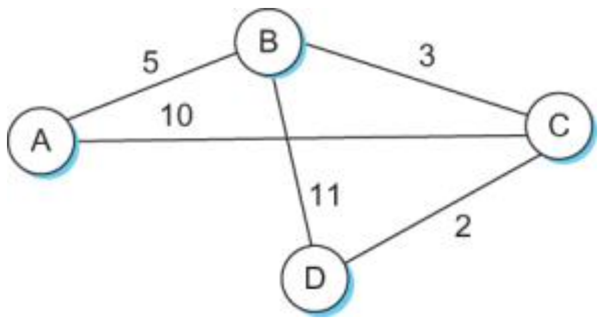
4. Distributing link state

- Making flooding reliable
  - Use acknowledgments and retransmissions between routers
  - Use sequence numbers
    - Discard info from packets older than your current info
  - Time-to-live TTL keeps LSP from being endlessly forwarded
- When to distribute?
  - Periodic timer
  - On detected change

5. Computing routes

- Router has accumulated full set of LSPs
  - Construct entire network graph
  - Shortest path from A to B?
  - Dijkstra's shortest path, forward search:
    - Maintain a tentative and confirmed list
    - Confirm yourself with cost 0
    - For last confirmed node, use its LSP to update tentative entries
      - Add new tentative entries, reduce cost using confirmed node
    - Confirm tentative with lowest cost

Shortest path routing

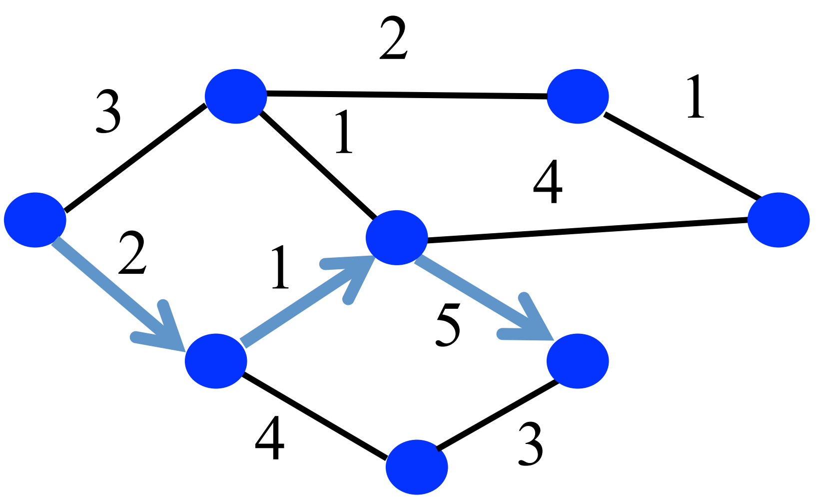


Step	Confirmed	Tentative	Comments
1	(D,0,–)		Since D is the only new member of the confirmed list, look at its LSP.
2	(D,0,–)	(B,11,B) (C,2,C)	D's LSP says we can reach B through B at cost 11, which is better than anything else on either list, so put it on Tentative list; same for C.
3	(D,0,–) (C,2,C)	(B,11,B)	Put lowest-cost member of Tentative (C) onto Confirmed list. Next, examine LSP of newly confirmed member (C).
4	(D,0,–) (C,2,C)	(B,5,C) (A,12,C)	Cost to reach B through C is 5, so replace (B,11,B). C's LSP tells us that we can reach A at cost 12.
5	(D,0,–) (C,2,C) (B,5,C)	(A,12,C)	Move lowest-cost member of Tentative (B) to Confirmed, then look at its LSP.
6	(D,0,–) (C,2,C) (B,5,C)	(A,10,C)	Since we can reach A at cost 5 through B, replace the Tentative entry.
7	(D,0,–) (C,2,C) (B,5,C) (A,10,C)		Move lowest-cost member of Tentative (A) to Confirmed, and we are all done.

Building routing table for node D.

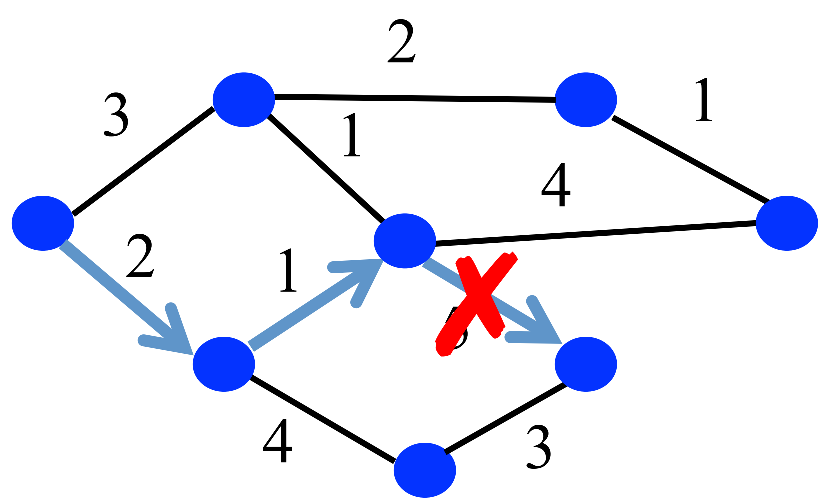
Link state convergence

- Consistent forwarding after convergence
  - All nodes have some link-state database
  - All nodes forward using shortest paths
  - The next router does what you think it will
    - Forward to the next hop in your shortest path calculation



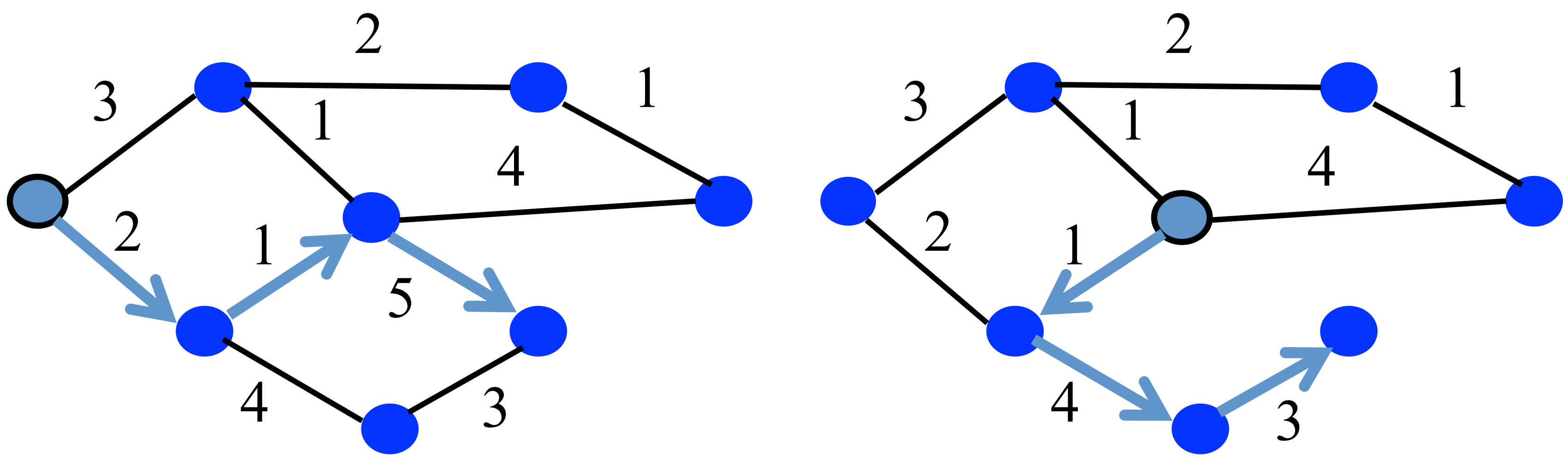
Transient disruptions

- Detection delay
  - Failures are not detected immediately
  - Router may forward packet into a "blackhole"
  - Chance depends on frequency of "hello" messages



Transient disruptions

- Inconsistent link-state
  - Some routers know about a failure, others don't
  - Shortest path no longer consistent
  - Can causes transient forwarding loops



Convergence delay

- Sources of delay:
  - Time to detect failure
  - Time to flood link-state info
  - Shortest path computation
  - Creating the forwarding table
- Before convergence:
  - Lost packets due to blackholes, TTL expiry
  - Looping packets
  - Out of order packets
  - Bad for Voice over IP, gaming, video

Reducing convergence delay

- Detect failures faster
  - Increase beacon frequency
  - Link-layer technologies that can detect failures
- Faster flooding
  - Flood immediately on a change
  - LSP sent with high-priority
- Faster computation
  - Faster processors in routers
  - Faster algorithms
    - e.g. incremental Dijkstra's
  - Faster forwarding table update
    - e.g. data structures supporting incremental updates

Distance vector vs. Link state

Distance vector	Link state
Knowledge of neighbors' distance to destinations	Knowledge of every router's links (entire network graph)
Router has O(# neighbors * # nodes)	Router has O(# edges)
Messages only between neighbors	Messages between all nodes
Trust a peer's routing computation	Trust a peer's info Do routing yourself
Bellman-Ford algorithm	Dijkstra's algorithm
Advantages: Less info has to be stored Lower computation overhead	**Advantages:** Fast to react to changes

Summary

- Error reporting (ICMP)
  - Router-to-router communications
  - Support user level tools, e.g. ping, traceroute
- Forwarding vs. Routing
- Two major types of routing
  - Distance vector
    - Router only know about its neighbors
  - Link state
    - Full state of network known by each router