CSC458 – Lecture 4
Bridging LANs and IP

Administrivia

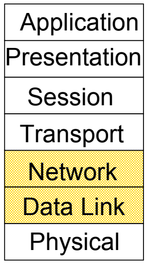
- Homework:
  - # 1 due today
  - # 2 out today and due in two weeks
- Readings:
  - Chapters 3 and 4
- Project:
  - # 2 due next week
- Tutorial today:
  - Joe Lim on project 2

Last Time …

- Medium Access Control (MAC) protocols
  - Part of the Link Layer
  - At the heart of Local Area Networks (LANs)
- How do multiple parties share a wire or the air?
  - Random access protocols (CSMA/CD)
  - Contention-free protocols (turn-taking, reservations)
  - Wireless protocols (CSMA/CA and RTS/CTS)

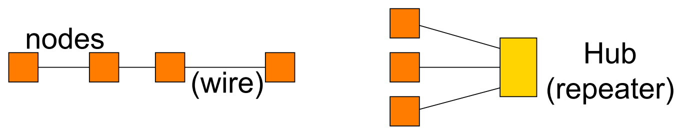
This Time -- Switching (a.k.a. Bridging)

- Focus:
  - What to do when one shared LAN isn't big enough?
- Interconnecting LANs
  - Bridges and LAN switches
  - A preview of the Network layer



Limits of a LAN

- One shared LAN can limit us in terms of:
  - Distance
  - Number of nodes
  - Performance



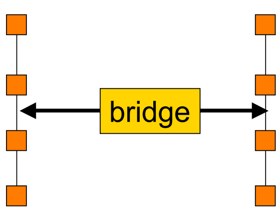
- How do we scale to a larger, faster network?
  - We must be able to interconnect LANs

Switching (a.k.a. Bridging)

- Transferring a packet from one LAN to another LAN
  - Build an "extended LAN"

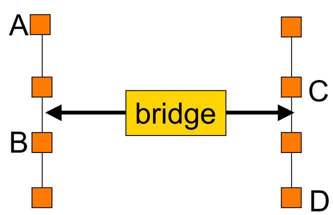
Bridges and Extended LANs

- "Transparently" interconnect LANs with bridge
  - Receive frames from each LAN and forward to the other
  - Each LAN is its own collision domain; bridge isn't a repeater
  - Could have many ports



Learning Bridges

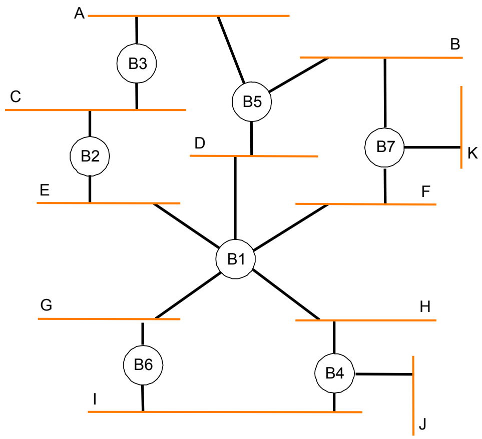
- To optimize overall performance:
  - Shouldn't forward A→B or C→D, should forward A→C and D→B



- How does the bridge know?
  - Learn who is where by observing <u>source</u> addresses and prune
  - Forward using destination address; age for robustness

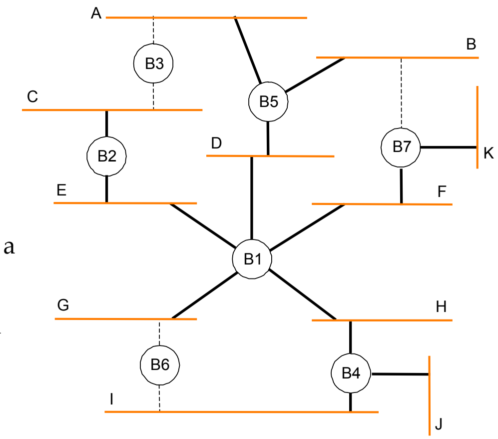
Why stop at one bridge?

- Allows you to incrementally build out network, across organizations
- But must avoid loops -- bridge must forward only on some bridge ports!
  - The Spanning Tree algorithm does this
  - It is separate from previous idea of learning



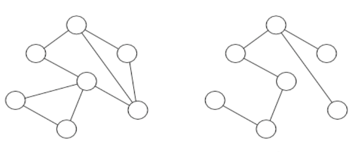
Spanning Tree Example

- Spanning tree uses select bridge ports so there are no cycles
  - Prune some ports
  - Only one tree
- Q: How do we find a spanning tree?
  - Automatically with a distributed algorithm



Spanning Tree

- Compute ST with *a* bridge as *root* such that
  - Root forwards onto all of its outgoing ports
  - Other bridges forward TO the root if a packet is coming from a bridge further from the root, else they forward away from the root
    - Packet traversal: forwards (UP)* then (DOWN*)



Spanning tree vs. learning

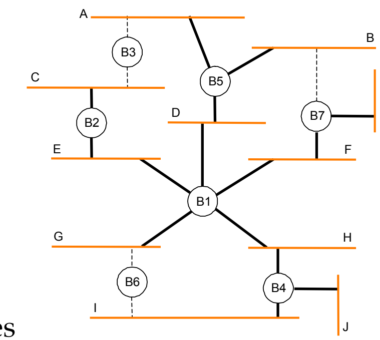
- Once the spanning tree is in place…
  - the bridge uses the regular learning algorithm to figure out which ports to forward / flood packet on
- Job of spanning tree algorithm is to disable some ports to eliminate cycles

Spanning Tree Algorithm

- Distributed algorithm to compute spanning tree
  - Robust against failures, needs no organization
  - Developed by Radia Perlman at DEC
    - IEEE 802.1 spec
    - http://www1.cs.columbia.edu/~ji/F02/ir02/p44-perlman.pdf
- Outline: Goal is to turn some bridge ports off
  1. Elect a root node of the tree (lowest address)
  2. Grow tree as shortest distances from the root (using lowest address to break distance ties)
     - All done by bridges sending periodic configuration messages over ports for which they are the "best" path
     - Then turn off ports that aren't on "best" paths

Algorithm Overview

- Each bridge has a unique id
  - e.g., B1, B2, B3
- Select the bridge with the smallest id as root
- Select bridge on each LAN that is closest to the root as that LAN's designated bridge
  - use ids to break ties
- Each bridge forwards frames over each LAN on which it is the designated bridge



Algorithm continued

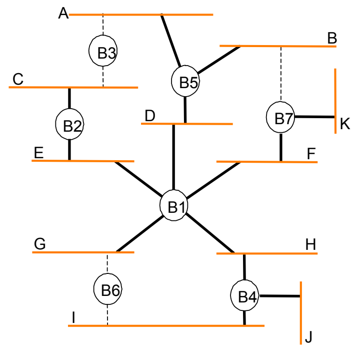
- Bridges exchange configuration messages, containing:
  - id for bridge sending the message
  - id for what the sending bridge believes to be the root bridge
  - distance (hops) from sending bridge to root bridge
- Each bridge records current best configuration message for each port
- Initially, each bridge believes it is the root
  - when learn not root, stop generating configuration messages
  - instead, forward root's configuration message
    - incrementing distance field by 1
  - in steady state, only root generates configuration messages

Algorithm More…

- When learn not designated bridge on LAN, stop forwarding configuration messages
  - in steady state, only designated bridges forward configuration messages
- Root bridge continues to send configuration messages periodically
- If a bridge does not receive config. message after a period of time:
  - assumes topology has changed
  - starts generating configuration messages claiming to be root

Algorithm Example

- Message format:
  - (root, dist-to-root, sending bridge)
- Sample messages sequences to and from B3:
  1. B3 sends (B3, 0, B3) to B2 and B5
  2. B3 receives (B2, 0, B2) and (B5, 0, B5) and accepts B2 as root
  3. B3 sends (B2, 1, B3) to B5
  4. B3 receives (B1, 1, B2) and (B1, 1, B5) and accepts B1 as root
  5. B3 could send (B1, 2, B3) but doesn't as its nowhere "best"
     B2 and B5 are better choices.
     so B3 is NOT a designated bridge
  6. B3 receives (B1, 1, B2) and (B1, 1, B5) again … stable
     B3 turns off data forwarding to LANs A and C

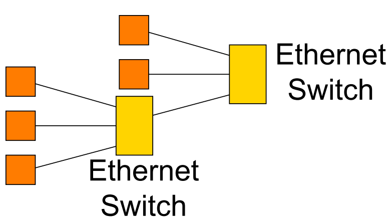


Some other tricky details

- Configuration information is aged
  - If the root fails a new one will be elected
- Reconfiguration is damped
  - Adopt new spanning trees slowly to avoid temporary loops

LAN Switches

- LAN switches are multi-port bridges
  - Modern, high performance form of bridged LANs
  - Looks like a hub, but frames are switched, not shared
  - Every host on a separate port, or can combine switches



Limitations of Bridges/Switches

- LAN switches form an effective small-scale network
  - Plug and play for real!
- Why can't we build a large network using bridges?
  - Little control over forwarding paths
  - Size of bridge forwarding tables grows with number of hosts
  - Broadcast traffic flows freely over whole extended LAN
  - Spanning tree algorithm limits reconfiguration speed
  - Poor solution for connecting LANs of different kinds

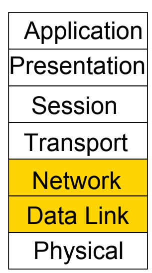
Key Concepts

- We can overcome LAN limits by interconnection
  - Bridges and LAN switches
  - But there are limits to this strategy …
- Next Topic: Routing and the Network layer
  - How to grow large and really large networks

Part 2: IP

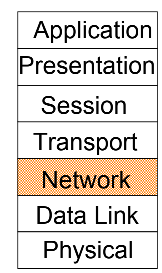
Last Time

- Focus:
  - What to do when one shared LAN isn't big enough?
- Interconnecting LANs
  - Bridges and LAN switches
  - But there are limits …



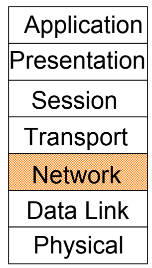
This Lecture

- Focus:
  - How do we build large networks?
- Introduction to the Network layer
  - Internetworks
  - Service models
  - IP, ICMP



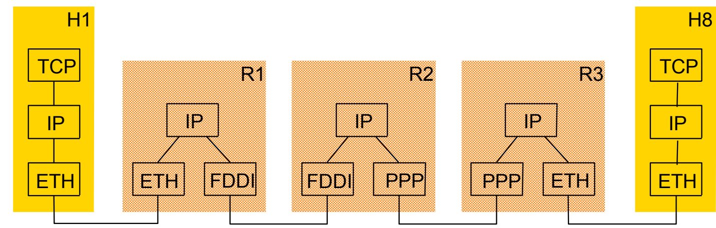
The Network Layer

- Job is to provide end-to-end data delivery between hosts on an internetwork
- Provides a higher layer of addressing



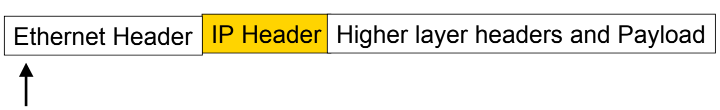
In terms of protocol stacks

- IP is the network layer protocol used in the Internet
- Routers are network level gateways
- Packet is the term for network layer PDUs



In terms of packet formats

- View of a packet on the wire on network 1 or 2
- Routers work with IP header, not higher
  - Higher would be a "layer violation"
- Routers strip and add link layer headers



Front of packet to left (and uppermost)

Network Service Models

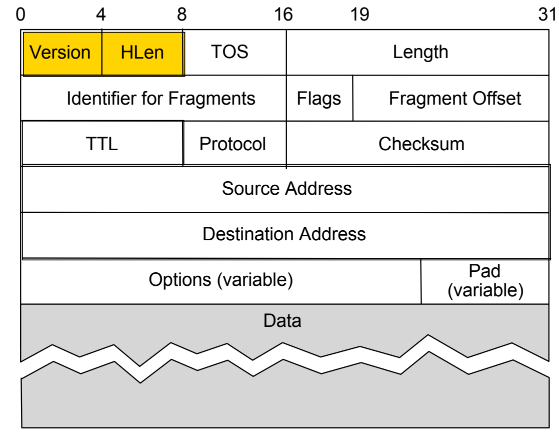
- Datagram delivery: postal service
  - connectionless, best-effort or unreliable service
  - Network can't guarantee delivery of the packet
  - Each packet from a host is routed independently
  - Example: IP
- Virtual circuit models: telephone
  - connection-oriented service
  - Signaling: connection establishment, data transfer, teardown
  - All packets from a host are routed the same way (router state)
  - Example: ATM, Frame Relay, X.25

Internet Protocol (IP)

- IP (RFC791) defines a datagram “best effort” service
  - May be loss, reordering, duplication, and errors!
  - Currently IPv4 (IP version 4), IPv6 on the way
- Routers forward packets using predetermined routes
  - Routing protocols (RIP, OSPF, BGP) run between routers to maintain routes (routing table, forwarding information base)
- Global, hierarchical addresses, not flat addresses
  - 32 bits in IPv4 address; 128 bits in IPv6 address
  - ARP (Address Resolution Protocol) maps IP to MAC addresses

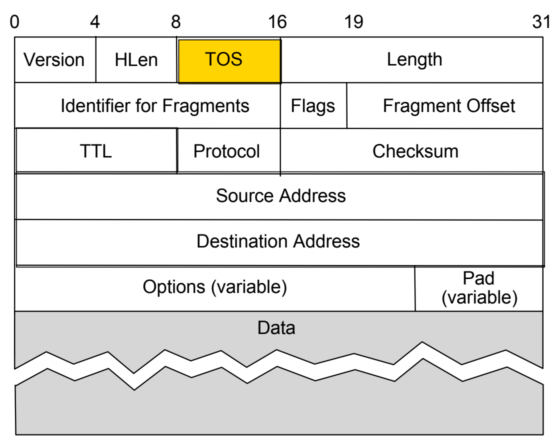
IPv4 Packet Format

- Version is 4
- Header length is number of 32 bit words
- Limits size of options



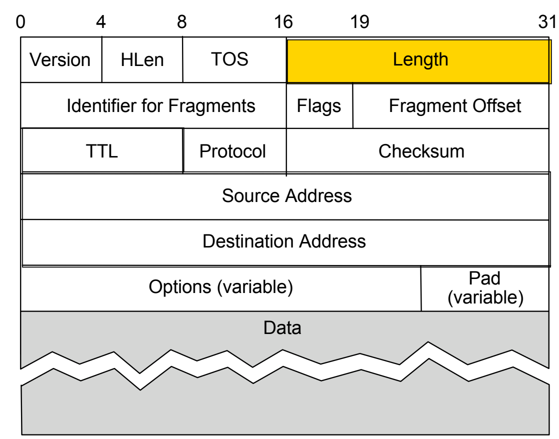
IPv4 Header Fields …

- Type of Service
- Abstract notion, never really worked out
  - Routers ignored
- But now being redefined for Diffserv



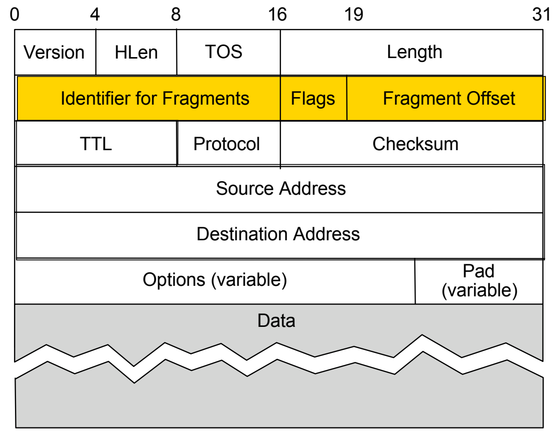
IPv4 Header Fields …

- Length of packet
- Min 20 bytes, max 65K bytes (limit to packet size)



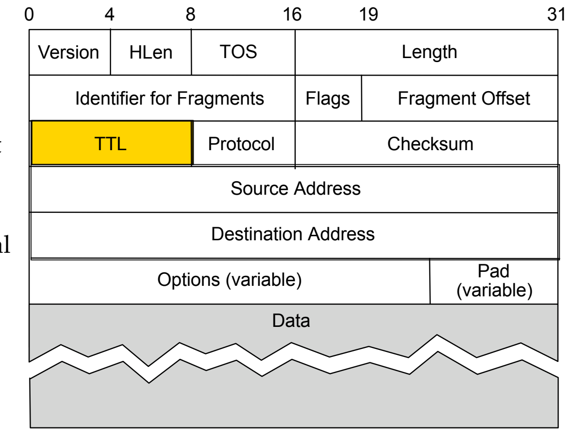
IPv4 Header Fields …

- Fragment fields
- Different LANs have different frame size limits
- May need to break large packet into smaller fragments



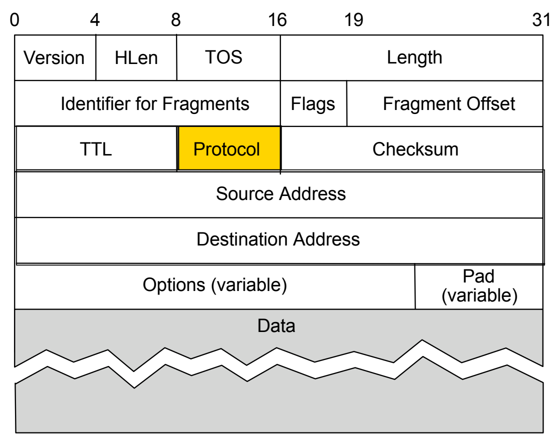
IPv4 Header Fields …

- Time To Live
- Decremented by router and packet discarded if = 0
- Prevents immortal packets



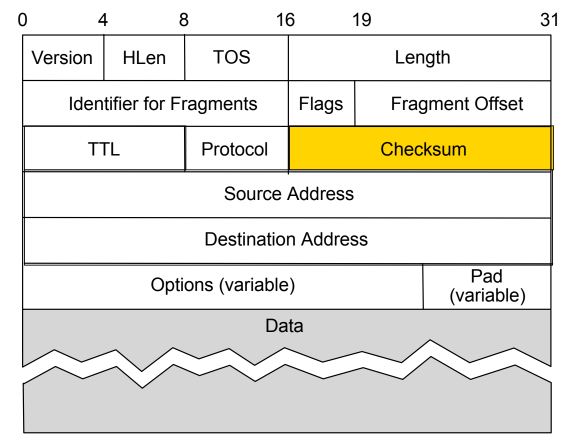
IPv4 Header Fields …

- Identifies higher layer protocol
  - E.g., TCP, UDP



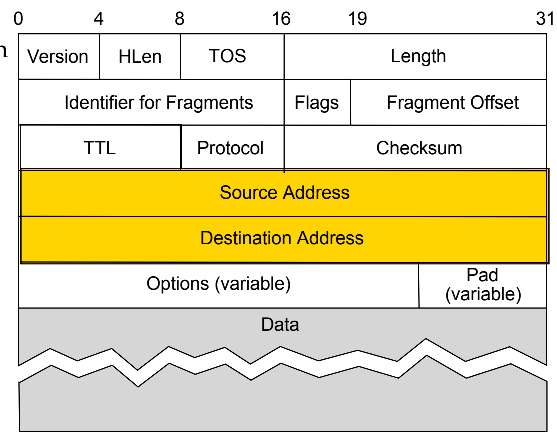
IPv4 Header Fields …

- Header checksum
- Recalculated by routers (TTL drops)
- Doesn't cover data
- Disappears for IPv6



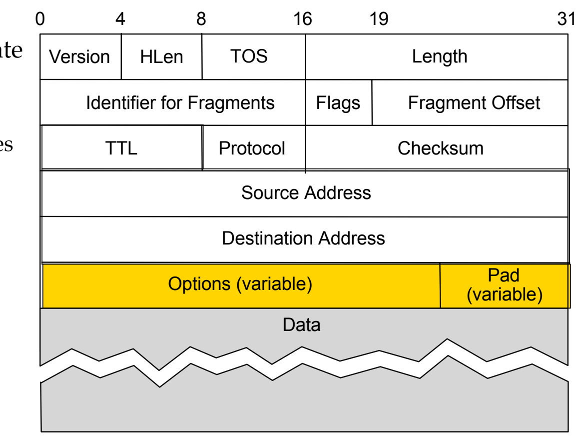
IPv4 Header Fields …

- Source/destination IP addresses
  - Not Ethernet
- Unchanged by routers
- Not authenticated by default



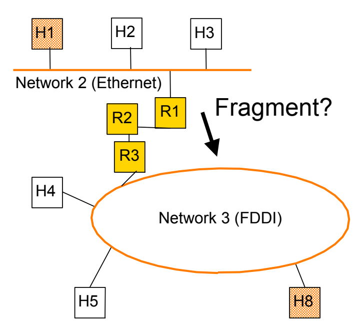
IPv4 Header Fields …

- IP options indicate special handling
  - Timestamps
  - "Source" routes
- Rarely used …



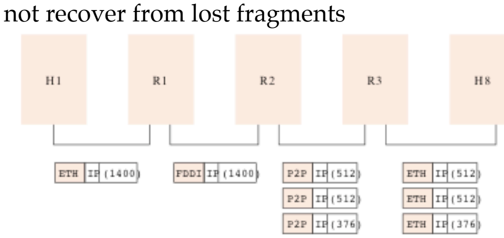
Fragmentation Issue

- Different networks may have different frame limits (MTUs)
  - Ethernet 1.5K, FDDI 4.5K
- Don't know if packet will be too big for path beforehand
  - IPv4: fragment on demand and reassemble at destination
  - IPv6: network returns error message so host can learn limit



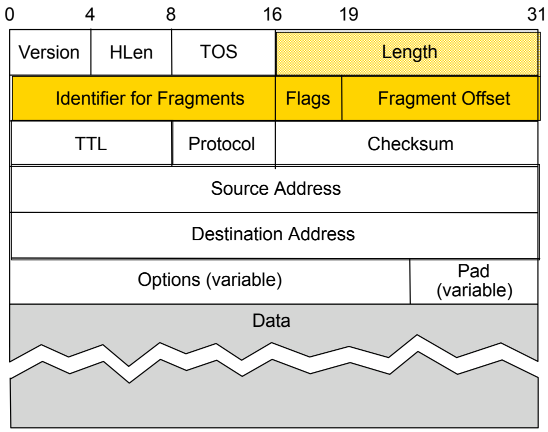
Fragmentation and Reassembly

- Strategy
  - fragment when necessary (MTU < Datagram size)
  - try to avoid fragmentation at source host
  - refragmentation is possible
  - fragments are self-contained IP datagrams
  - delay reassembly until destination host
  - do not recover from lost fragments

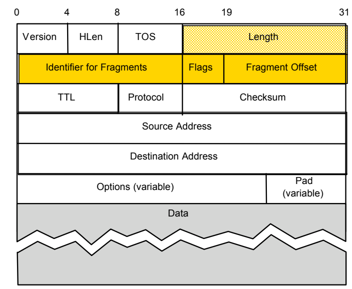


Fragment Fields

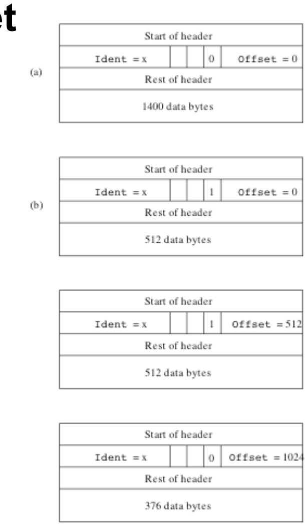
- Fragments of one packet identified by (source, dest, frag id) triple
  - Make unique
- Offset gives start, length changed
- Flags are More Fragments (MF) Don't Fragment (DF)



Fragmenting a Packet



Packet Format



Fragment Considerations

- Making fragments be datagrams provides:
  - Tolerance of loss, reordering and duplication
  - Ability to fragment fragments
- Reassembly done at the endpoint
  - Puts pressure on the receiver, not network interior
- Consequences of fragmentation:
  - Loss of any fragments causes loss of entire packet
  - Need to time-out reassembly when any fragments lost

Fragmentation Issues Summary

- Causes inefficient use of resources within the network
  - BW, CPU
- Higher level protocols must re-xmit entire datagram
  - on lossy network links, hard for packet to survive
- Efficient reassembly is hard
  - Lots of special cases
  - (think linked lists)

Avoiding Fragmentation

- Always send small datagrams
  - Might be too small
- "Guess" MTU of path
  - Use DF flag. May have large startup time
- Discover actual MTU of path
  - One RT delay w/help, much more w/o.
  - "Help" requires router support
- Guess or discover, but be willing to accept your mistakes

Path MTU Discovery

- Path MTU is the smallest MTU along path
  - Packets less than this size don't get fragmented
- Fragmentation is a burden for routers
  - We already avoid reassembling at routers
  - Avoid fragmentation too by having hosts learn path MTUs
- Hosts send packets, routers return error if too large
  - Hosts discover limits, can fragment at source
  - Reassembly at destination as before
- Learned lesson from IPv4, streamlined in IPv6

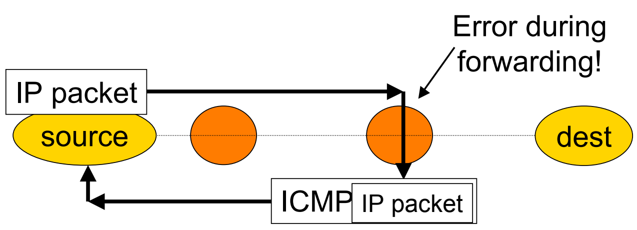
IP Addresses and IP Datagram Forwarding

- IP datagram (packet) contains destination address
- How the source gets the packet to the destination:
  - if source is on same network (LAN) as destination, source sends packet directly to destination host
  - else source sends data to a router on the same network as the source
  - router will forward packet to a router on the next network over
  - and so on…
  - until packet arrives at router on same network as destination; then, router sends packet directly to destination host
- Requirements
  - every host needs to know IP address of the router on its LAN
  - every router needs a routing table to tell it which neighboring network to forward a given packet on

ICMP

- What happens when things go wrong?
  - Need a way to test/debug a large, widely distributed system
- ICMP = Internet Control Message Protocol (RFC792)
  - Companion to IP – required functionality
- Used for error and information reporting:
  - Errors that occur during IP forwarding
  - Queries about the status of the network

ICMP Generation



Common ICMP Messages

- Destination unreachable
  - "Destination" can be host, network, port or protocol
- Packet needs fragmenting but DF is set
- Redirect
  - To shortcut circuitous routing
- TTL Expired
  - Used by the "traceroute" program
- Echo request/reply
  - Used by the "ping" program
- Cannot Fragment
- Busted Checksum

- ICMP messages include portion of IP packet that triggered the error (if applicable) in their payload

ICMP Restrictions

- The generation of error messages is limited to avoid cascades … error causes error that causes error!

- Don't generate ICMP error in response to:
  - An ICMP error
  - Broadcast/multicast messages (link or IP level)
  - IP header that is corrupt or has bogus source address
  - Fragments, except the first

- ICMP messages are often rate-limited too.

Key Concepts

- Network layer provides end-to-end data delivery across an internetwork, not just a LAN

  - Datagram and virtual circuit service models
  - IP/ICMP is the network layer protocol of the Internet

- Up next: More detailed look at routing and addressing