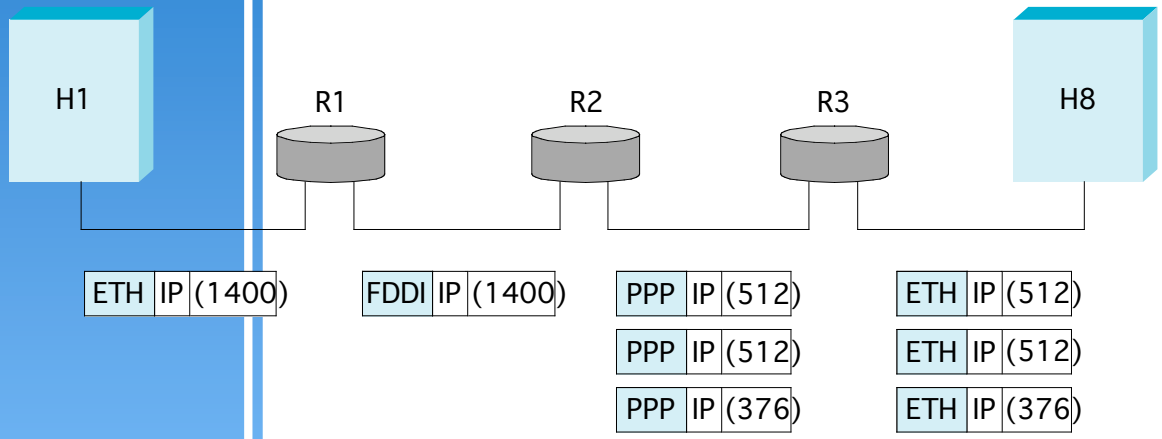
- Internetworking - We are heterogeneity to our network (variable network technologies, bandwidth, MTU, latency, etc. etc.)
  - Goal is to use this opportunity (and not to find the lowest common denominator performance)

Issue 2: Fragmentation and Reassembly

- Cannot expect all networks to deal with the same packet size, can choose the absolute smallest - but that would mean poor performance for all networks
  - Each network has some MTU (maximum transmission unit)
- Design decisions
  - fragment when necessary (MTU < Datagram)
  - re-fragmentation is possible
  - fragments are self-contained datagrams
  - delay reassembly until destination host
  - do not recover from lost fragments
    - Suppose we send a 64k packet over ether (1500 byte MTU), 1 packet = 44 fragments. Losing 1 of 44 fragments = lose the entire packet
  - try to avoid fragmentation at source host

Example



(a)

Start of header				
Ident = x			0	Offset = 0
Rest of header				
1400 data bytes				

(b)

Start of header				
Ident = x			1	Offset = 0
Rest of header				
512 data bytes				

Start of header				
Ident = x			1	Offset = 64
Rest of header				
512 data bytes				

Start of header				
Ident = x			0	Offset = 128
Rest of header				
376 data bytes				

Gateway13 example

- Ifconfig eth0 in gateway13.cse.nd.edu

Link encap:Ethernet HWaddr 00:07:E9:3C:8F:80

inet addr:129.74.154.198 Bcast:129.74.155.255
Mask:255.255.252.0

inet6 addr: fe80::207:e9ff:fe3c:8f80/64 Scope:Link

UP BROADCAST RUNNING MULTICAST
MTU:9000 Metric:1

Issue 3: Datagram Forwarding

- Strategy
  - every datagram contains destination's address
  - if connected to destination network, then forward to host
  - if not directly connected, then forward to some router
  - forwarding table maps network number into next hop
  - each host has a default router
  - each router maintains a forwarding table

Example (R2)

Network number	Next
1	R3
2	R1
3	Interface 1
4	Interface 0

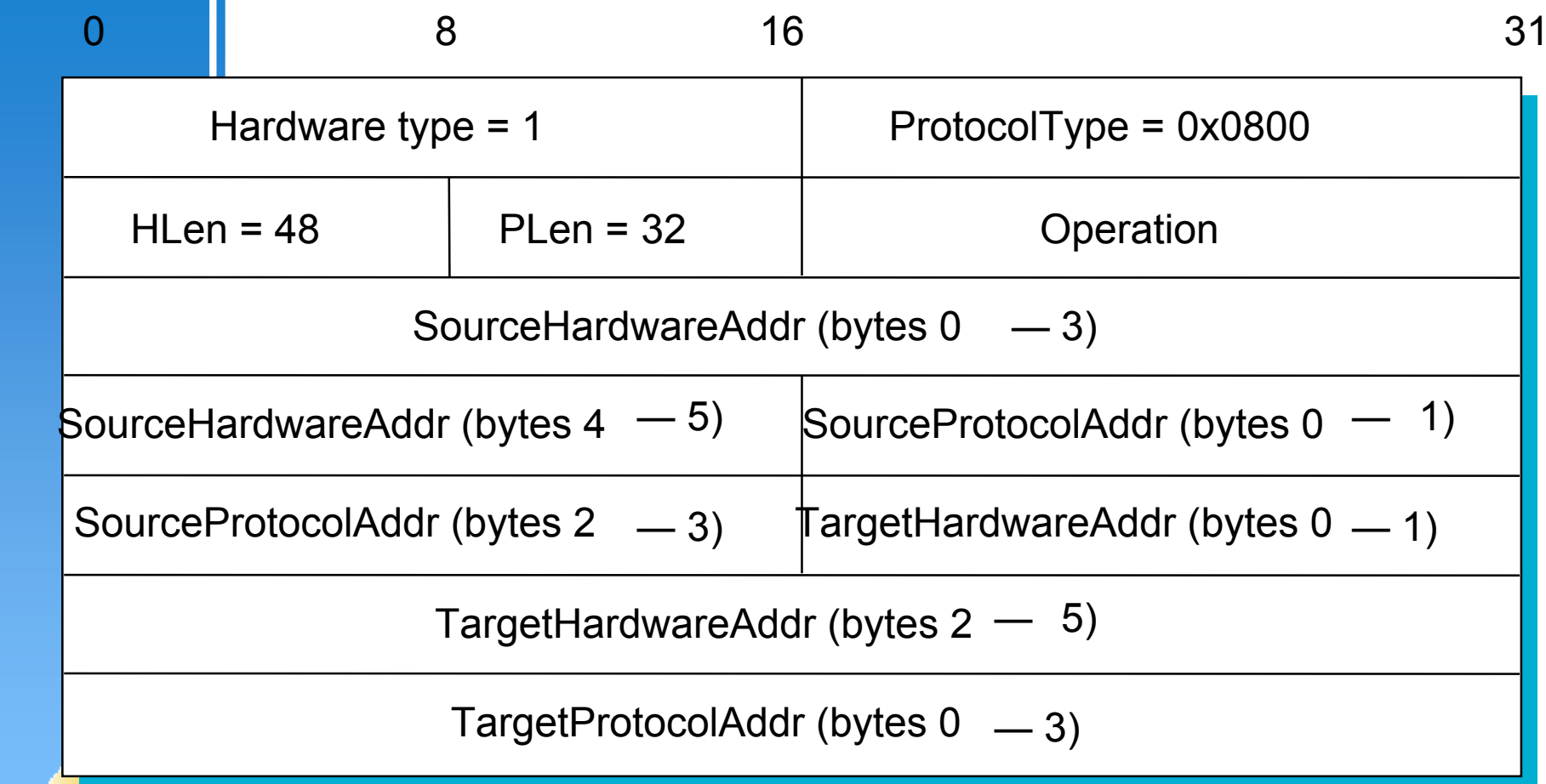
Address Translation

- Map IP addresses into physical addresses
  - destination host
  - next hop router
- Techniques
  - encode physical address in host part of IP address
  - table-based
- Mechanism to map IP to physical address: ARP
  - table of IP to physical address bindings
  - broadcast request if IP address not in table
  - target machine responds with its physical address
  - table entries are discarded if not refreshed

ARP Details

- Request Format
  - HardwareType: type of physical network (e.g., Ethernet)
  - ProtocolType: type of higher layer protocol (e.g., IP)
  - HLEN & PLEN: length of physical and protocol addresses
  - Operation: request or response
  - Source/Target-Physical/Protocol addresses
- Notes
  - table entries timeout in about 10 minutes
  - update table with source when you are the target
  - update table if already have an entry
  - do not refresh table entries upon reference

ARP Packet Format



Sample arp table in darwin.cc.nd.edu

- arp -a

Net to Media Table: IPv4

```
Device   IP Address               Mask      Flags   Phys Addr
------ -------------- --------- ----- ---------
hme0   eafs-e06.gw.nd.edu   255.255.255.255       00:d0:c0:d3:aa:40
hme0   bind.nd.edu          255.255.255.255       08:00:20:8a:5f:cf
hme0   honcho-jr.cc.nd.edu  255.255.255.255       00:b0:d0:82:83:7f
hme0   mail-vip.cc.nd.edu   255.255.255.255       02:e0:52:0c:56:c4
hme0   john.helios.nd.edu   255.255.255.255       08:00:20:85:db:c4
hme0   casper.helios.nd.edu 255.255.255.255       08:00:20:b1:f8:e1
hme0   pinky.helios.nd.edu  255.255.255.255       08:00:20:a9:88:30
```

ARP problems

- ARP trusts any response - no authentication method
  - Works great at home, how about Notre Dame
- Replies which do not correspond to requests are allowed to update cache in many instances
- New information must supercede old info

Internet Control Message Protocol (ICMP)

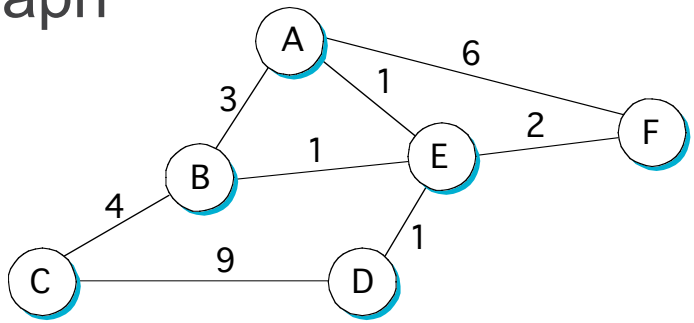
- Mechanisms to notify of errors (not mandatory)
  - Echo (ping)
  - Redirect (from router to source host)
  - Destination unreachable (protocol, port, or host)
  - TTL exceeded (so datagrams don't cycle forever)
  - Checksum failed
  - Reassembly failed
  - Cannot fragment

Next problem: Routing

- Routing is similar to bridging in functionality. We want to use multiple paths instead of removing redundancy (spanning tree alrogithm)

Overview 4.2: Routing

- Forwarding vs Routing
  - forwarding: to select an output port based on destination address and routing table
  - routing: process by which routing table is built
- Network as a Graph



- Problem: Find lowest cost path between two nodes
- Factors
  - static: topology
  - dynamic: load
  - Distributed algorithm: what info. should be exchanged?

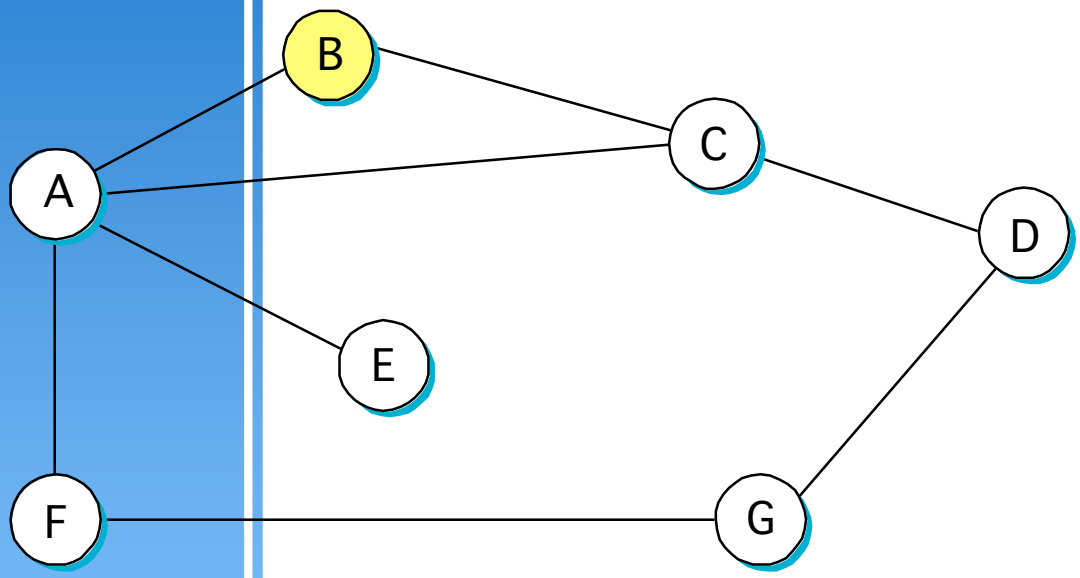
Two kinds of routing algorithms

- Distance vector:
  - Each router exchanges information with other routers and makes independent decisions
- Link state:
  - All routers exchange information, learn the state of the "world" and find the most optimal route (which is uniform across all routers)

Distance Vector (e.g. RIP v1)

- Each node maintains a set of triples
  - (Destination, Cost, NextHop)
- Directly connected neighbors exchange updates
  - periodically (on the order of several seconds)
  - whenever table changes (called triggered update)
- Each update is a list of pairs:
  - (Destination, Cost)
- Update local table if receive a “better” route
  - smaller cost
  - came from next-hop
- Refresh existing routes; delete if they time out

Example



Destination	Cost	NextHop
A	1	A
C	1	C
D	2	C
E	2	A
F	2	A
G	3	A

Routing Loops

- Example 1
  - F detects that link to G has failed
  - F sets distance to G to infinity and sends update to A
  - A sets distance to G to infinity since it uses F to reach G
  - A receives periodic update from C with 2-hop path to G
  - A sets distance to G to 3 and sends update to F
  - F decides it can reach G in 4 hops via A
- Example 2: count to infinity problem
  - link from A to E fails
  - A advertises distance of infinity to E
  - B and C advertise a distance of 2 to E
  - B decides it can reach E in 3 hops; advertises this to A
  - A decides it can read E in 4 hops; advertises this to C
  - C decides that it can reach E in 5 hops…

Loop-Breaking Heuristics

- Set infinity to 16
- Split horizon
- Split horizon with poison reverse

Link State (e.g. OSPF)

- Strategy
  - send to all nodes (not just neighbors) information about directly connected links (not entire routing table)
- Link State Packet (LSP)
  - id of the node that created the LSP
  - cost of link to each directly connected neighbor
  - sequence number (SEQNO)
  - time-to-live (TTL) for this packet

Link State (cont)

- Reliable flooding
  - store most recent LSP from each node
  - forward LSP to all nodes but one that sent it
  - generate new LSP periodically
    - increment SEQNO
  - start SEQNO at 0 when reboot
  - decrement TTL of each stored LSP
    - discard when TTL=0

Route Calculation

- Dijkstra's shortest path algorithm
- Let
  - N denotes set of nodes in the graph
  - l (i, j) denotes non-negative cost (weight) for edge (i, j)
  - s denotes this node
  - M denotes the set of nodes incorporated so far
  - C(n) denotes cost of the path from s to node n

- M = {s}
- for each n in N - {s}
  - C(n) = l(s, n)
- while (N != M)
  - M = M union {w} such that C(w) is the minimum for
  - all w in (N - M)
  - for each n in (N - M)
    - C(n) = MIN(C(n), C (w) + l(w, n))

Route cost metrics

- Original ARPANET metric
  - measures number of packets queued on each link
  - took neither latency or bandwidth into consideration
- New ARPANET metric
  - stamp each incoming packet with its arrival time (AT)
  - record departure time (DT)
  - when link-level ACK arrives, compute
    - Delay = (DT - AT) + Transmit + Latency
  - if timeout, reset DT to departure time for retransmission
  - link cost = average delay over some time period
- Fine Tuning
  - compressed dynamic range
  - replaced Delay with link utilization

Mobility

- What if nodes move
  - You need a new IP address when you move
  - Communications (sockets) have to be reestablished
  - One solution is to use Dynamic DNS with DHCP
    - Used at ND
    - When a host moves, DHCP gives it a new address and Dynamic DNS updates the DNS entry with the new DHCP address
    - For example, my laptop is called kural.cse.nd.edu, but may map into different IP addresses depending on where I am
    - Works for new connections, old connections break
    - Can only work within the same domain (because DNS servers are only administered for the domain)

Mobile IP

- Mobile host registers with Foreign Agent. FA informs Home Agent. HA tunnels packets to FA. Communications through the Home address.