How/When Are IP Routing Tables Built?

- We already understand how routing tables are **used**
- The next questions are:
  - *What values should routing tables contain?*
  - *How can those values be obtained?*
- Depends on size/complexity of internet
- Static routing
  - Routes fixed at boot time
  - Useful only for simple networks
- Automatic routing
  - Table initialized at boot time
  - Values inserted/updated by protocols that propagate route information
  - Necessary in large internets

Routing with Partial Information

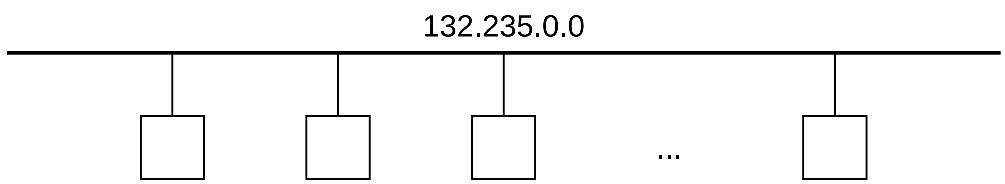
- Hosts and routers generally need different amounts of routing information
  - Many times, all a host needs is the address of a single router to use for all non-local traffic

    A host can route datagrams successfully even if it only has partial routing information because it can rely on a router.
  - A router generally needs more detailed information
- Default routes can be helpful in certain situations, but can lead to confusion:

  Pomeroy right lane, Marietta left lane, all others straight ahead

  - Will that lead me to Washington DC?
  - Will following that route yield the shortest path?

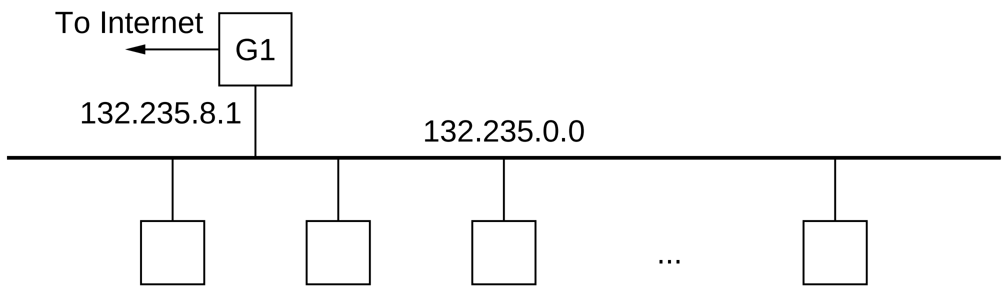
Default Host Routing Example #1



- Architecture: hosts on an isolated Ethernet
- Static routing
- Host routing table

Destination	Mask	Route
132.235.0.0	ffff0000	direct

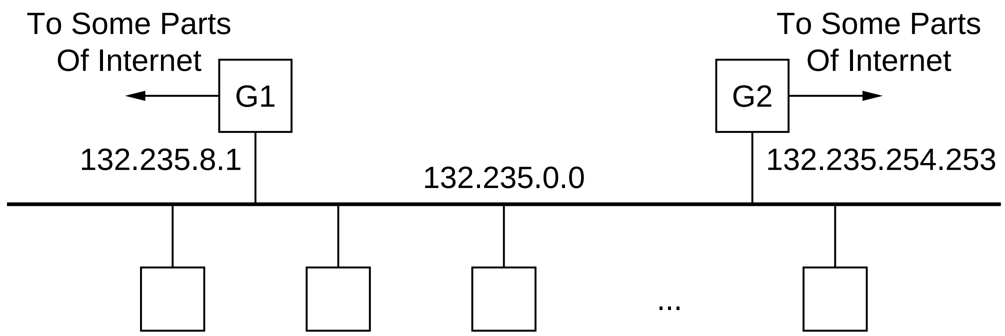
Default Host Routing Example #2



- Architecture: hosts on an Ethernet with one gateway
- Static routing
- Host routing table

Destination	Mask	Route
132.235.0.0	ffff0000	direct
default	00000000	132.235.8.1

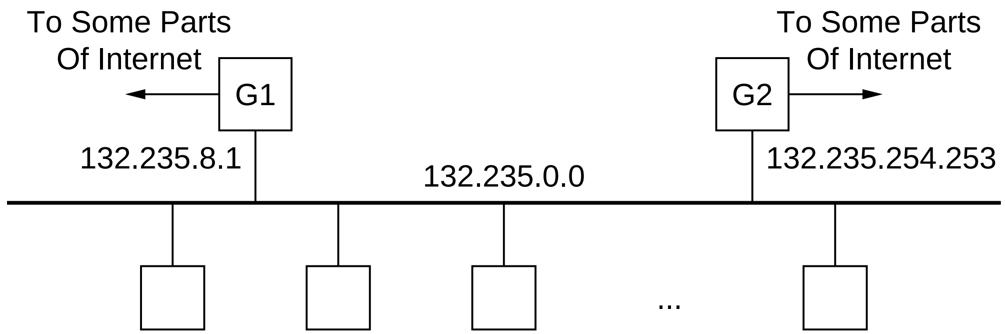
Host Routing Example #3



- Architecture: hosts on an Ethernet with two gateways
- Static routing + ICMP redirects
- Initial host routing table

Destination	Mask	Route
132.235.0.0	ffff0000	direct
default	00000000	132.235.8.1

Host Routing Example #3 (continued)

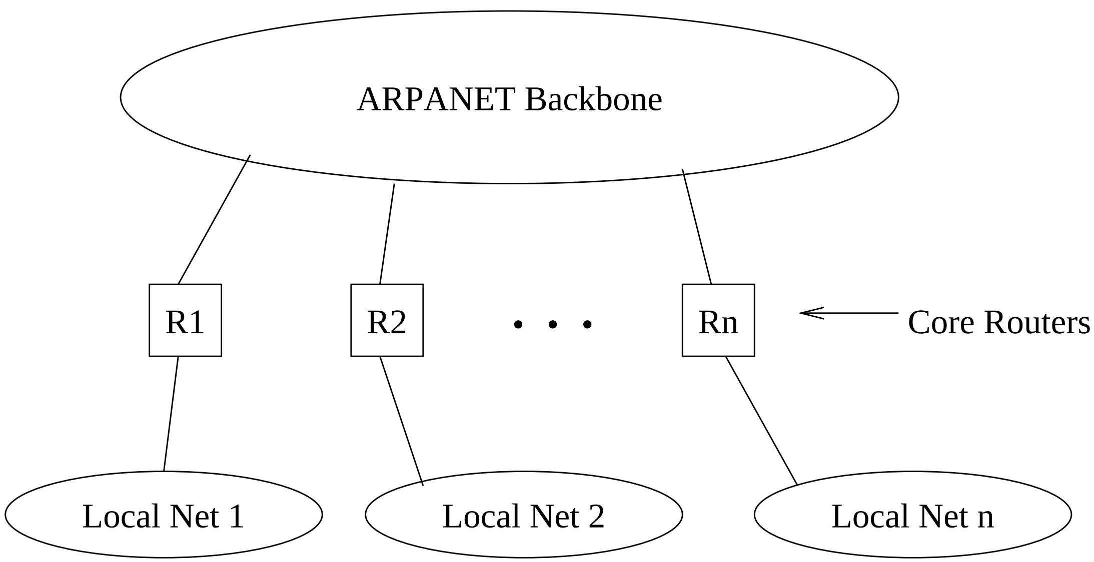


- Host generates datagram for 192.5.48.3, which lies beyond G2
- Host routes datagram to G1
- ICMP redirect from G1 updates table

Destination	Mask	Route
132.235.0.0	ffff0000	direct
192.5.48.0	ffffff00	132.235.254.253
default	00000000	132.235.8.1

Core Routers

- The old ARPANET had a very simple topology in which local networks were connected to the Arpanet through core routers
  - Local networks could rely on simple default routes
  - Only core routers needed detailed information



Why Must Core Routers be Smart?

- Even with this simple case, the core routers cannot rely on default routes
  - If they did, then routes across the core would not be efficient
  - For example, core router R_1 could use router R_2 as it's default, R_2 could use R_3, etc, with R_n using R_0
    - All packets would eventually get where they were going, but it would not be efficient
- There needs to be a way for core routers to exchange information about the local networks that they are connected to

Vector Distance Algorithm

- One gateway sends its routing table to another
- Table contains pairs of destination network and distance
- Receiver replaces entries in its table by routes to the sender if routing through the sender is less expensive than the current route
- Receiver propagates new routes next time it sends out an update
- Algorithm has several well-known shortcomings

GGP

- The original core routers used a vector-distance protocol called *gateway-to-gateway Protocol*, GGP
- Interesting, but obsolete protocol
- Updates traveled in IP datagrams, (like UDP)

GGP
Update Contents

- Each update contains:
  - Sequence number
  - Number up distances
    - Distance 1
      - Number of nets of distance 1
      - Network 1 address at distance 1
      - Network 2 address at distance 1
      - ...
    - Distance n
      - Number of nets of distance n
      - Network 1 address at distance n
      - Network 2 address at distance n
      - ...

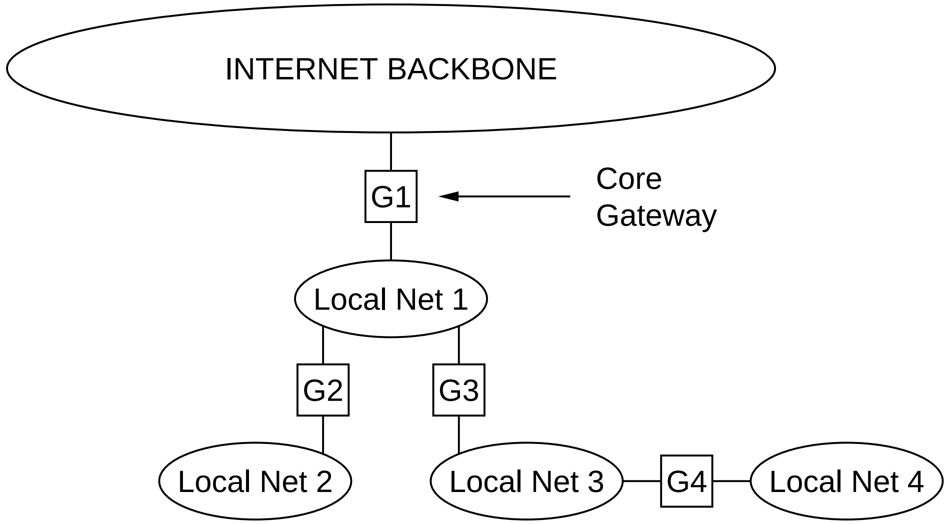
Link-State Algorithm

- All gateways know topology
- Think of gateways as nodes in a graph, and networks connecting them as edges or links
- All gateways propagate status of directly connected links periodically
- All gateways recompute routes from their copy of link information
- Also called *Shortest Path First* (SPF)
  - Comes from Dijkstra's short path algorithm

Gateway Types

- Gateways that live in the "middle" of the Internet are called Core gateways
- There are also noncore gateways
  - Any gateway that is not part of the core system
  - Might not be "trusted" by core gateways
  - Might not be maintained by the same group as the core gateways
  - Does not participate directly in core's routing information propagation algorithm
  - May not choose optimal routes if it uses the core except for local delivery

Example Of Hidden Networks



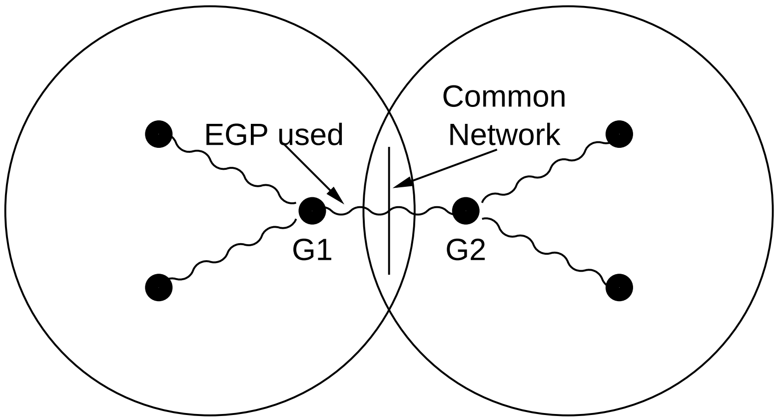
- Propagation of route information is independent of datagram routing

- Core gateway must learn routes from non-participating gateways

- Example: owner of networks 1 and 3 must tell core about route to network 4

Exterior Gateway Protocol (EGP)

- Standard Internet protocol
- Solves two problems
  - Allows noncore gateway to advertise networks hidden in its autonomous system
  - Allows noncore gateways to learn routes from the core
- Designed for communication with the Internet core system
- Now used primarily between pairs of autonomous systems

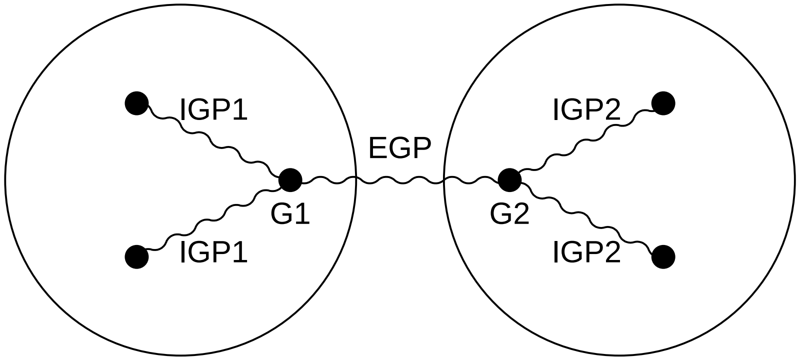


EGP Details

- Gateway in one autonomous system becomes the peer (neighbor) of a gateway in another autonomous system
- The two peers periodically poll each other
- Protocol keeps test of whether neighbor is alive separate from reachability update
- Polling rates may be asymmetric
- EGP supports messages for
  - Neighbor acquisition
  - Liveness test
  - Poll for update
  - Route update

Exchanging Routing Information Within An Autonomous System

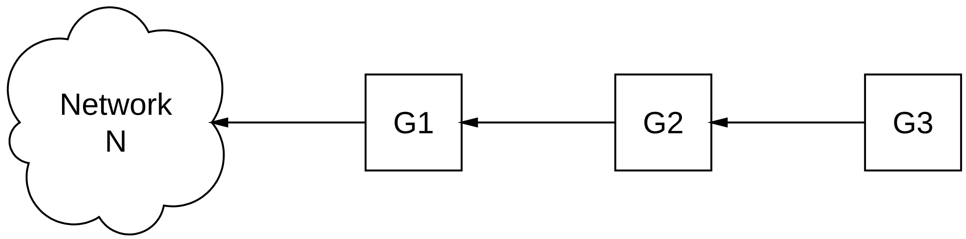
- Mechanisms called interior gateway protocols, IGPs
- Choice of IGP is made by autonomous system
- Some gateway in the autonomous system advertises network reachability to other autonomous systems with EGP
- Example IGPs are RIP, HELLO, and OSPF



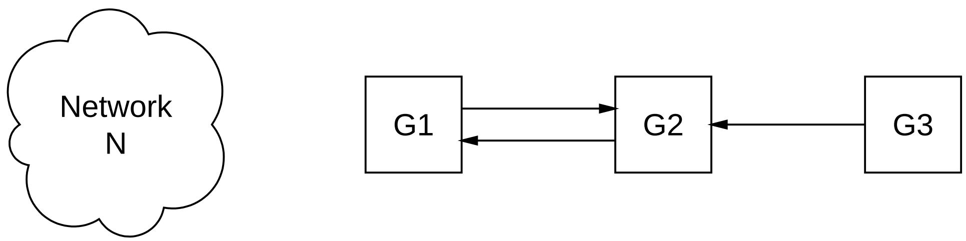
Routing Information Protocol (RIP)

- Implemented by 4BSD UNIX program *routed*
- Uses hop count metric
- Vector-distance protocol
- Relies on broadcast
- Assumes low-delay local area network
- Uses split horizon and poison reverse techniques to solve inconsistencies
- RIP2 includes subnet mask information

Slow Convergence Problem (Count To Infinity)



Gateways with routes to network N



G1 erroneously routes to G2 after failure

- Problem solved using *split horizon* and *hold down*

RIP Update Format

0	8	16	31
COMMAND	VERSION	RESERVED	
FAMILY OF NET 1		NET 1 ADDR., OCTETS 1 - 2	
NET 1 ADDRESS, OCTETS 3 - 6			
NET 1 ADDRESS, OCTETS 7 - 10			
NET 1 ADDRESS, OCTETS 11 - 14			
DISTANCE OF NETWORK 1			
FAMILY OF NET 2		NET 2 ADDR., OCTETS 1 - 2	
NET 2 ADDRESS, OCTETS 3 - 6			
NET 2 ADDRESS, OCTETS 7 - 10			
NET 2 ADDRESS, OCTETS 11 - 14			
DISTANCE OF NETWORK 2			
...			

- Uses family field to support multiple protocols
- Message travels in UDP datagram

HELLO Protocol

- Mostly of historical interest
- Developed by Dave Mills
- Used by NSFNET fuzzballs
- Uses metric based on delay
- Participants keep track of delay between pairs of gateways
- HELLO propagates delay information across net
- Route chosen to minimize total delay

OSPF Protocol

- Uses SPF algorithm for better scaling the vector-distance mechanisms
- Designed by the IETF
- Open standard
- Included *type of service routing*
  - Can install multiple routes to a destination based on the *type of service* field in the IP header
- Provides *load balancing*
  - Can specify multiple routes to a destination, OSPF will use them all
- Includes various *authentication* schemes
  - Only *trusted* routers can propogate routing information
- Supports host-specific routes and subnets

BGP - Border Gateway Protocol

! this slide is from Doug's online notes

- The most popular (virtually the only) EGP in use in the Internet
- Current version is BGP-4
- Allows two autonomous systems to communicate routing information
- Supports CIDR (mask accompanies each route)
- Each AS designates a border router to speak on its behalf
- Two border routers become BGP peers

Key Characteristics Of BGP

- Provides inter-autonomous system communication
- Propagates reachability information
- Follows next-hop paradigm
- Provides support for policies
- Sends path information
- Permits incremental updates
- Allows route aggregation
- Allows authentication