Distance Vector Routing: overview

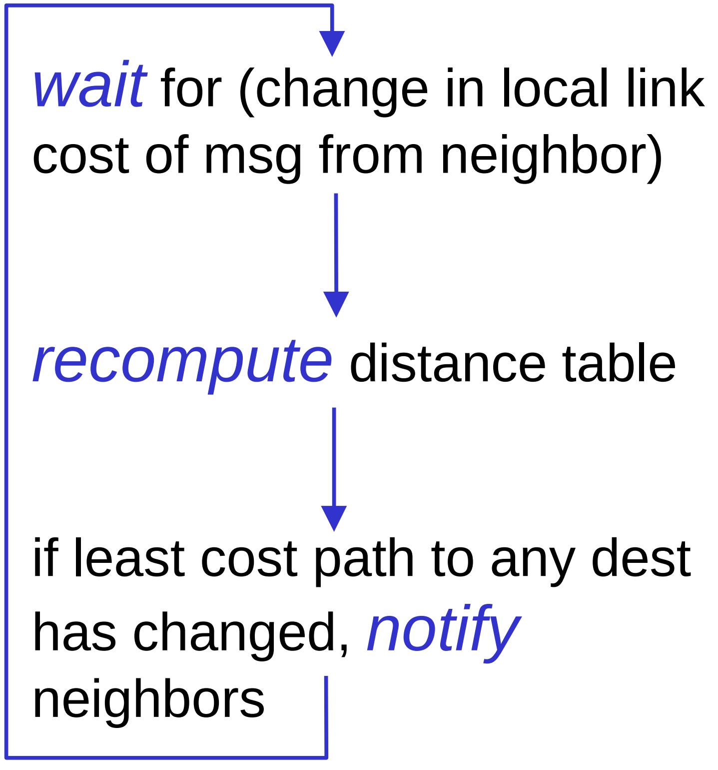
Iterative, asynchronous: each local iteration caused by:

- 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:



Distance Vector Algorithm:

At all nodes, X:

Initialization:

  for all adjacent nodes v:

    $D^X(*,v)$ = infty   /* the * operator means "for all rows" */

    $D^X(v,v) = c(X,v)$

  for all destinations, y

    send $\min_w D^X(y,w)$ to each neighbor /* w over all X's neighbors */

Distance Vector Algorithm (cont.):

loop

  wait (until I see a link cost change to neighbor V

      or until I receive update from neighbor V)

  if (c(X,V) changes by d)

    /* change cost to all dest's via neighbor v by d */

    /* note: d could be positive or negative */

    for all destinations y: $D^X(y,V) = D^X(y,V) + d$

  else if (update received from V wrt destination Y)

    /* shortest path from V to some Y has changed */

    /* V has sent a new value for its $\min_w DV(Y,w)$ */

    /* call this received new value is "newval" */

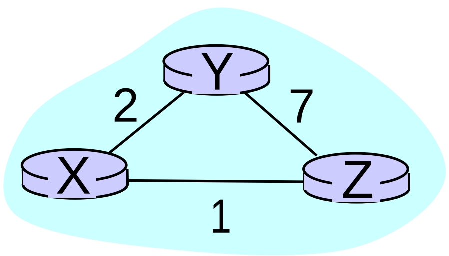
    for the single destination y: $D^X(Y,V) = c(X,V) + newval$

  if we have a new $\min_w D^X(Y,w)$ for any destination Y

     send new value of $\min_w D^X(Y,w)$ to all neighbors

forever

Distance Vector Algorithm: example



D^X (dest)	cost via Y	cost via Z
Y	2	∞
Z	∞	7

D^Y (dest)	cost via X	cost via Z
X	2	∞
Z	∞	1

D^Z (dest)	cost via X	cost via Y
X	7	∞
Y	∞	1

D^X (dest)	cost via Y	cost via Z
Y	2	8
Z	3	7

D^Y (dest)	cost via X	cost via Z
X	2	8
Z	9	1

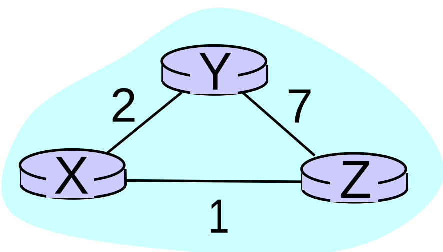
D^Z (dest)	cost via X	cost via Y
X	7	3
Y	9	1

D^X (dest)	cost via Y	cost via Z
Y		
Z		

D^Y (dest)	cost via X	cost via Z
X		
Z		

D^Z (dest)	cost via X	cost via Y
X		
Y		

Distance Vector Algorithm: example



D^X (cost via)

dest	Y	Z
Y	2	∞
Z	∞	7

D^Y (cost via)

dest	X	Z
X	2	∞
Z	∞	1

D^Z (cost via)

dest	X	Y
X	7	∞
Y	∞	1

D^X (cost via)

dest	Y	Z
Y	2	8
Z	3	7

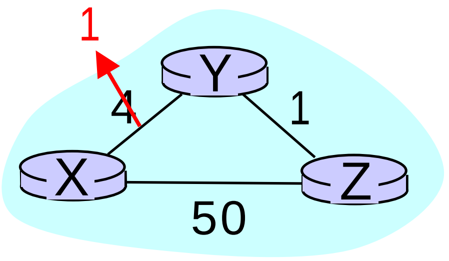
$$D^X(Y,Z) = c(X,Z) + \min_w\{D^Z(Y,w)\} = 7+1 = 8$$

$$D^X(Z,Y) = c(X,Y) + \min_w\{D^Y(Z,w)\} = 2+1 = 3$$

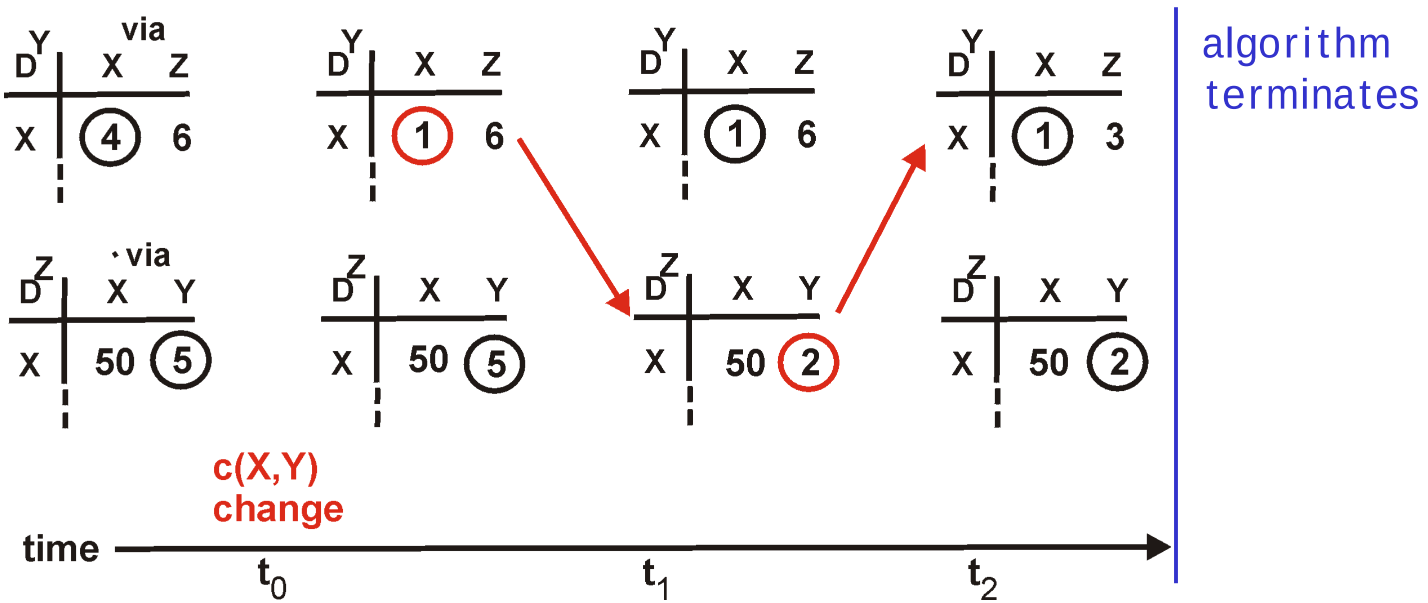
Distance Vector: link cost changes

Link cost changes:

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



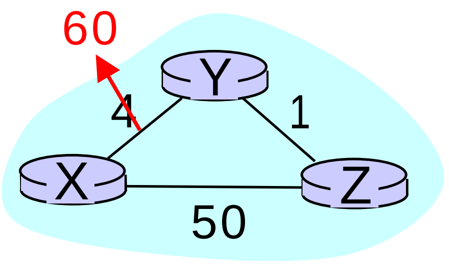
"good news travels fast"

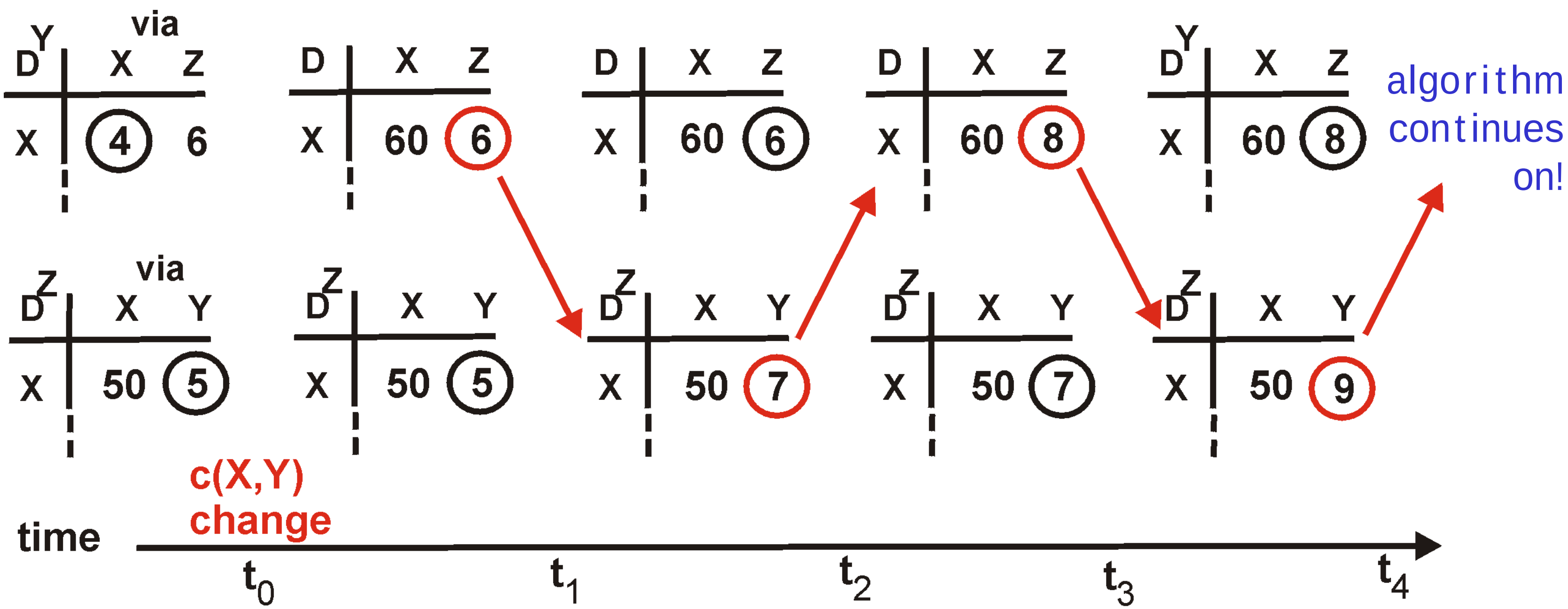


Distance Vector: link cost changes

Link cost changes:

- good news travels fast
- bad news travels slow - “count to infinity” problem!

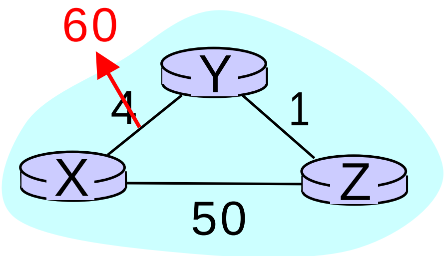


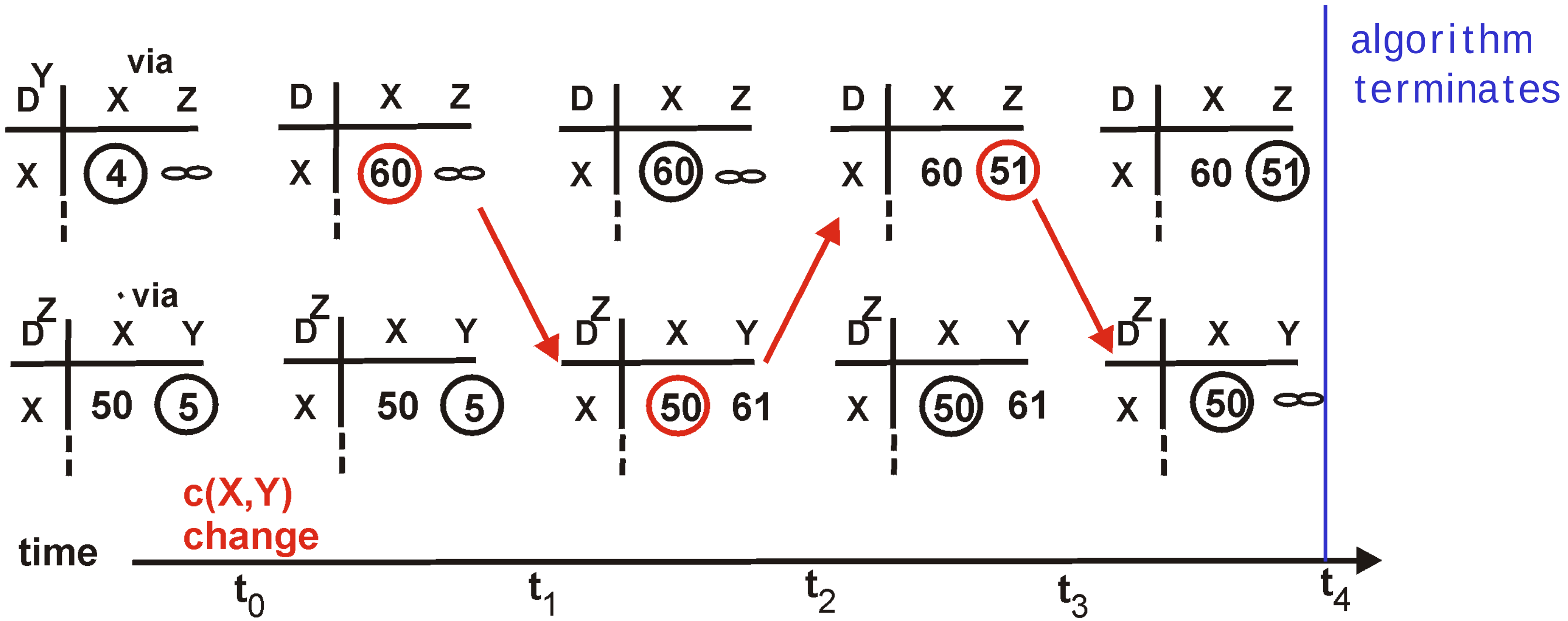


Distance Vector: poisoned reverse

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?





Comparison of LS and DV algorithms

Message complexity

- LS: with n nodes, E links, O(nE) msgs sent each
- DV: exchange between neighbors only
  - convergence time varies

Speed of Convergence

- LS: $O(n^{**}2)$ algorithm requires O(nE) msgs
  - may have oscillations
- DV: convergence time varies
  - may be 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
  - error propagate thru network

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

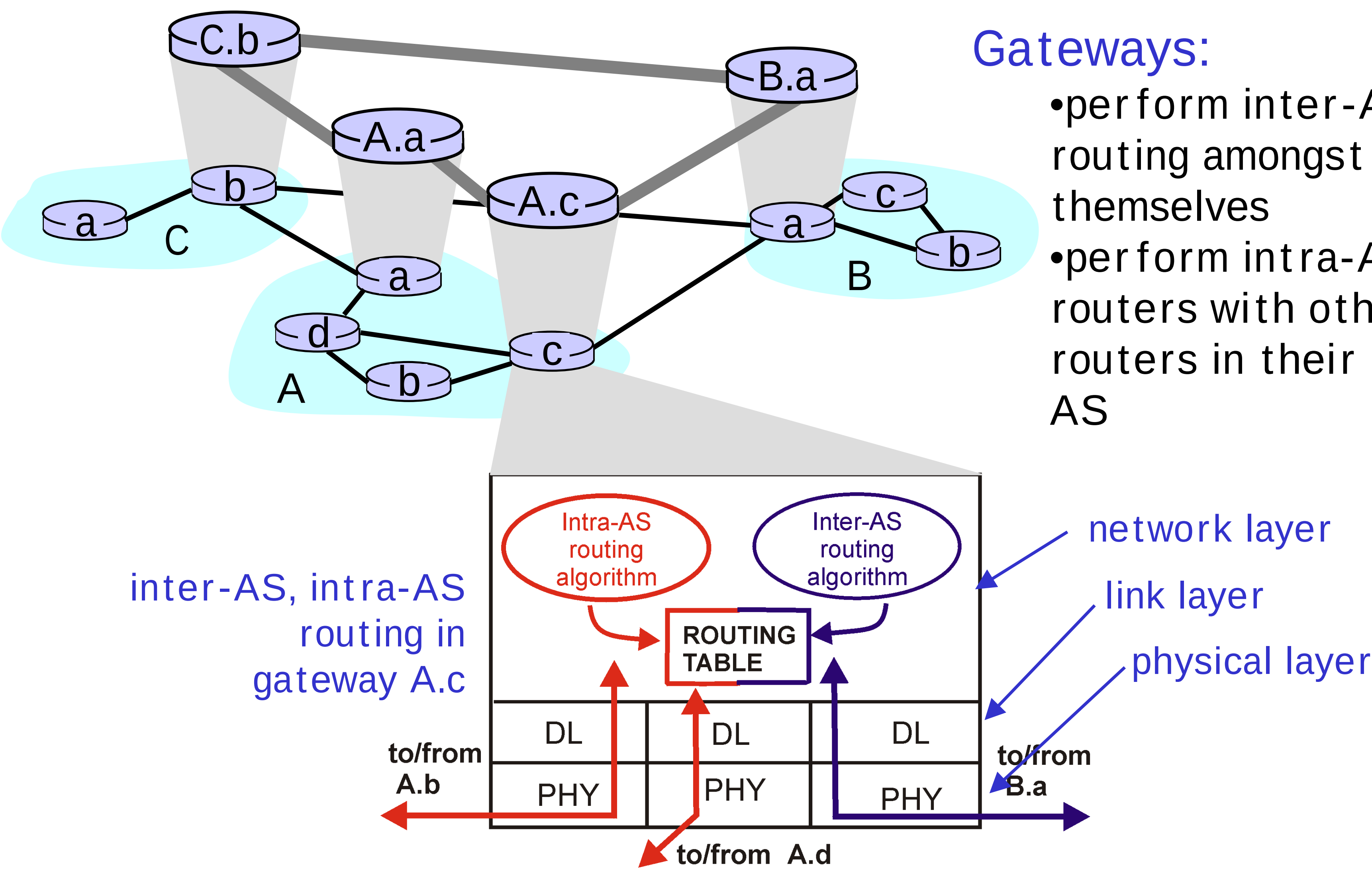
Hierarchical Routing

- aggregate routers into regions, "autonomous systems" (AS)
- routers in same AS run same routing protocol
  - "inter-AS" routing protocol
  - routers in different AS can run different inter-AS routing protocol

gateway routers

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

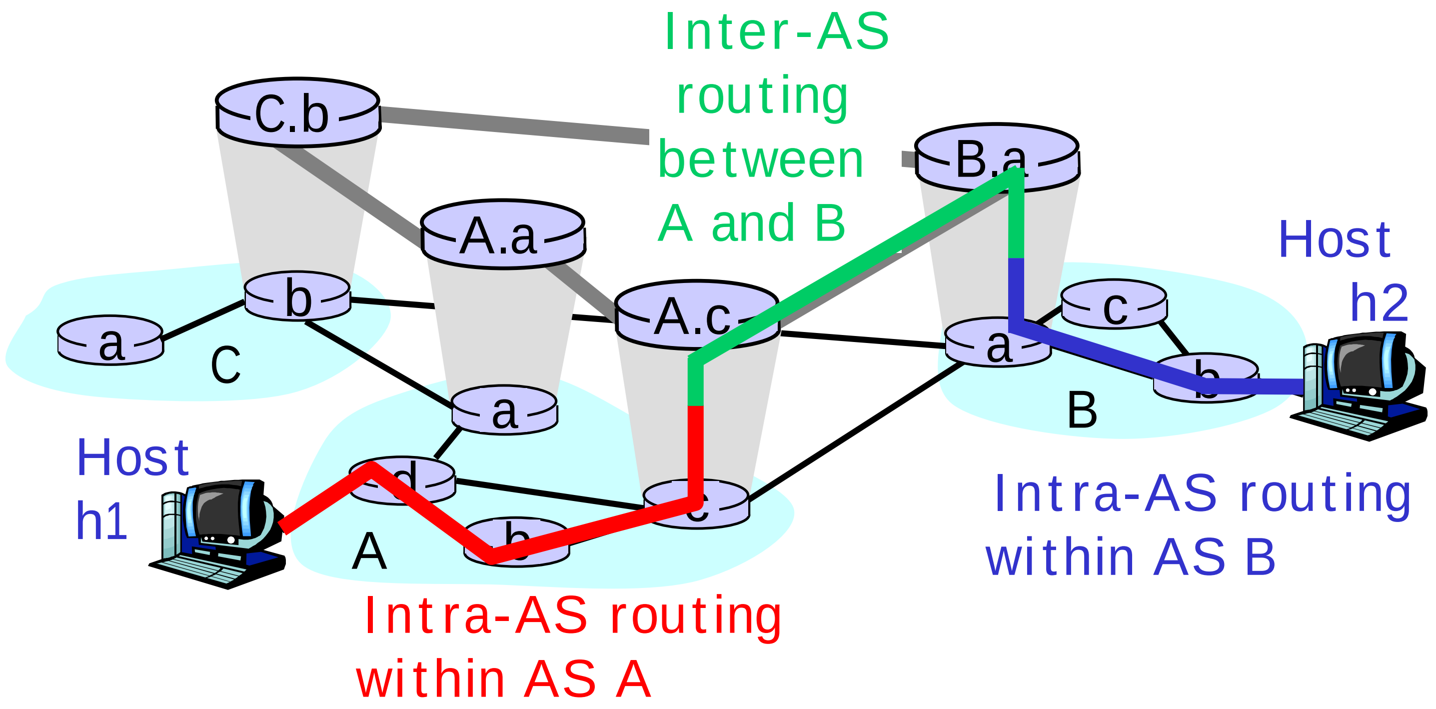
Intra-AS and Inter-AS routing



Gateways:

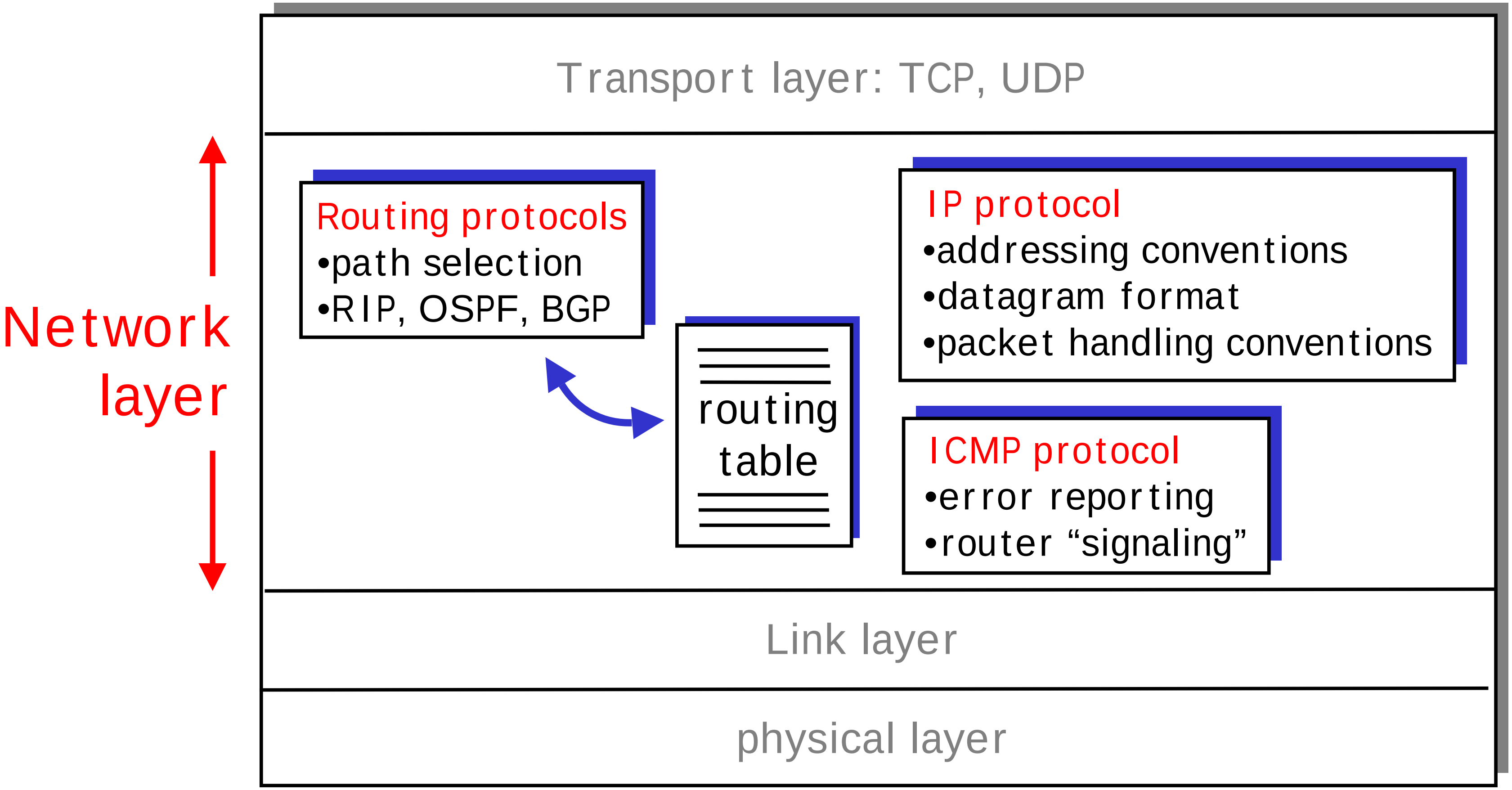
- perform inter-AS routing amongst themselves
- perform intra-AS routers with other routers in their AS

Intra-AS and Inter-AS routing



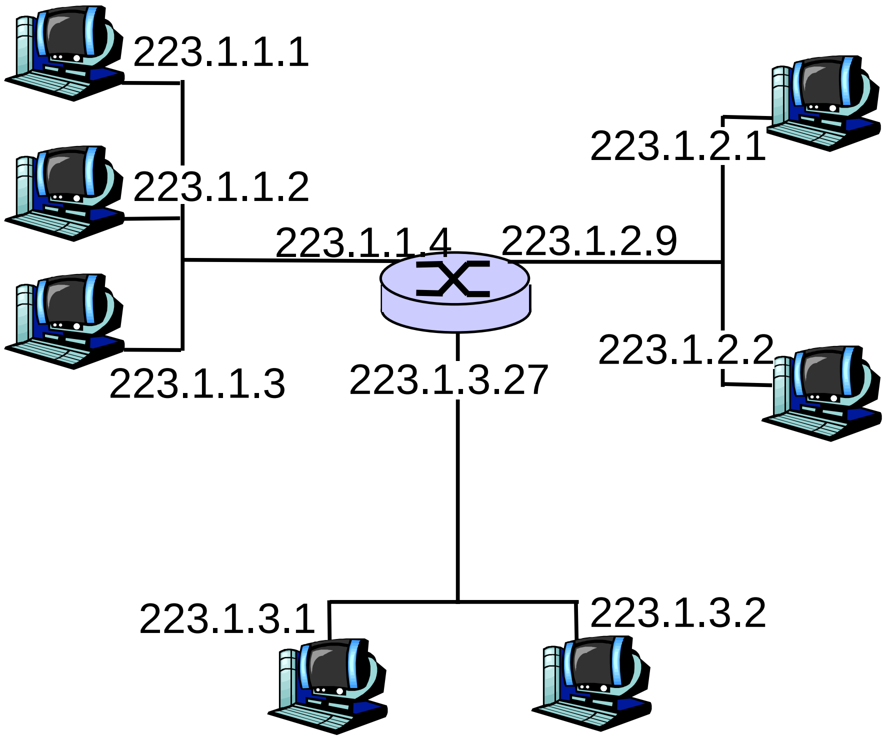
The Internet Network layer

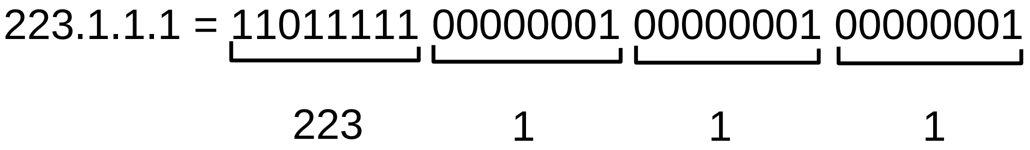
Host, router network layer functions:



IP Addressing

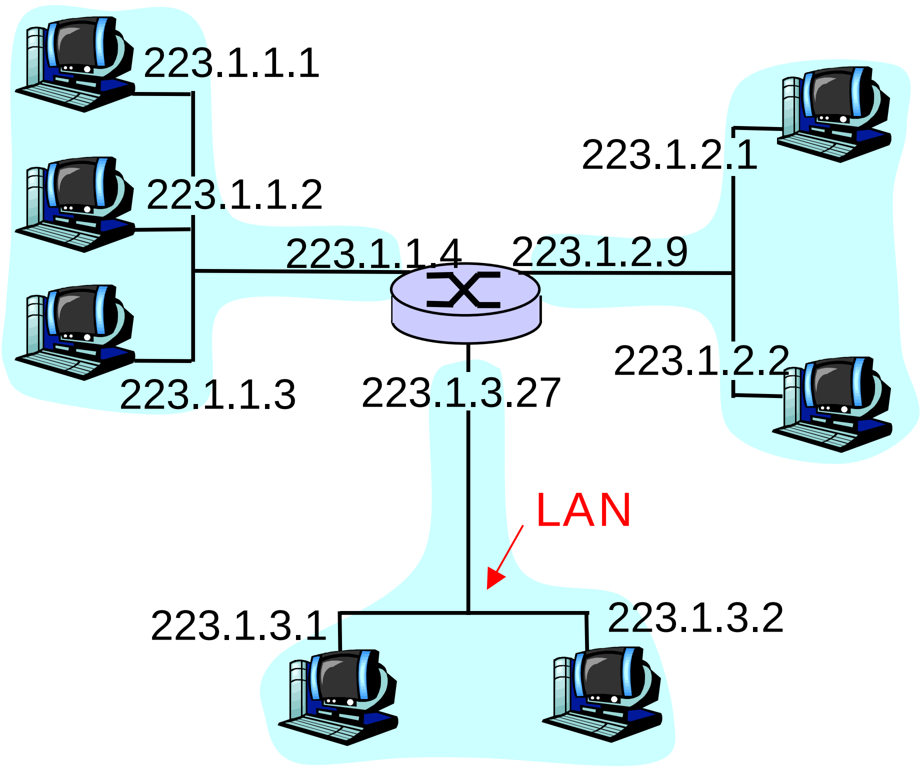
- **IP address:** 32-bit identifier for host, router *interface*
- *interface:* connection between host, router and physical link
  - router's typically have multiple interfaces
  - host may have multiple interfaces
  - IP addresses associated with interface, not host, router





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

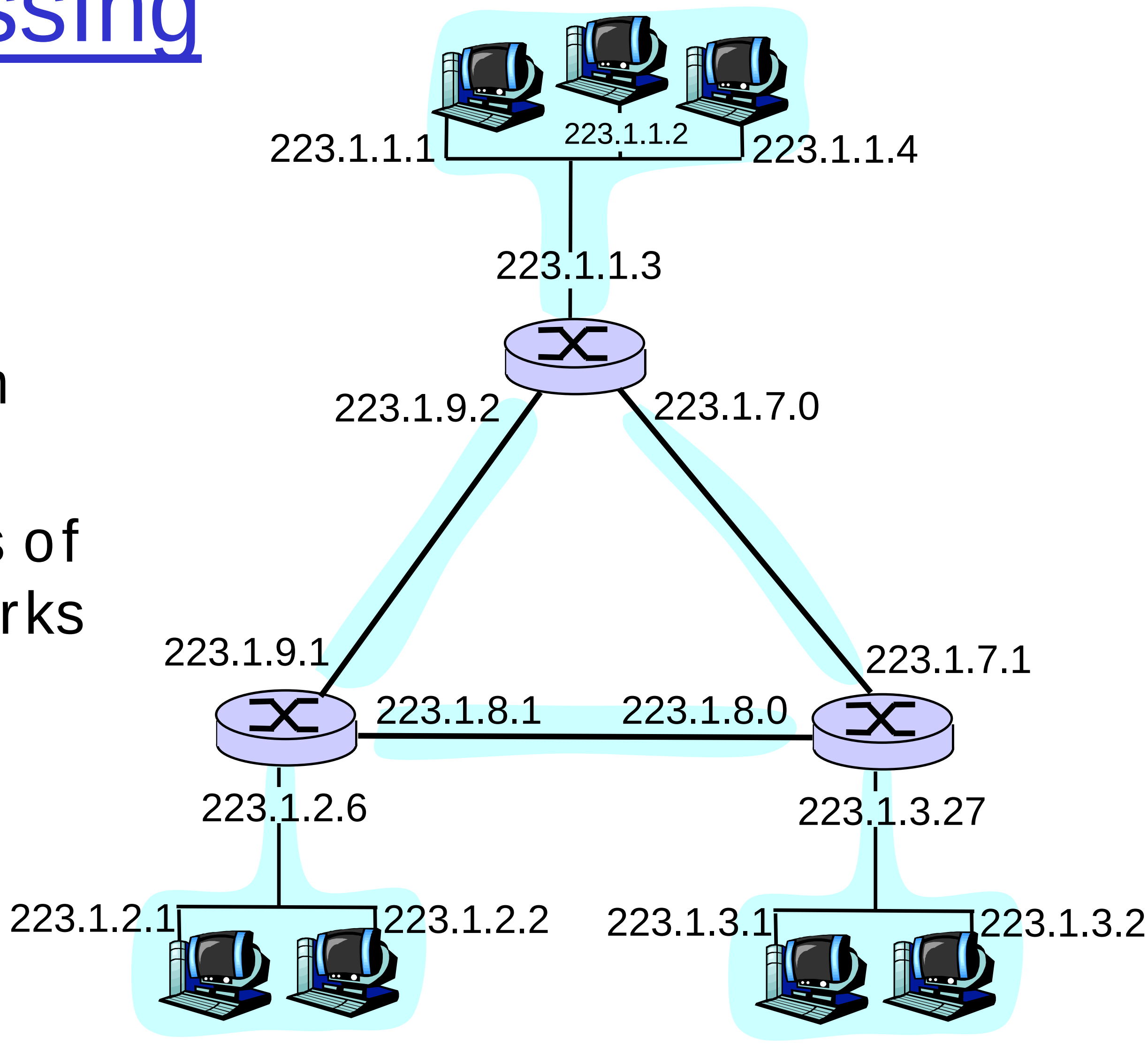


network consisting of 3 IP networks (for IP addresses starting with 223, first 24 bits are network address)

IP Addressing

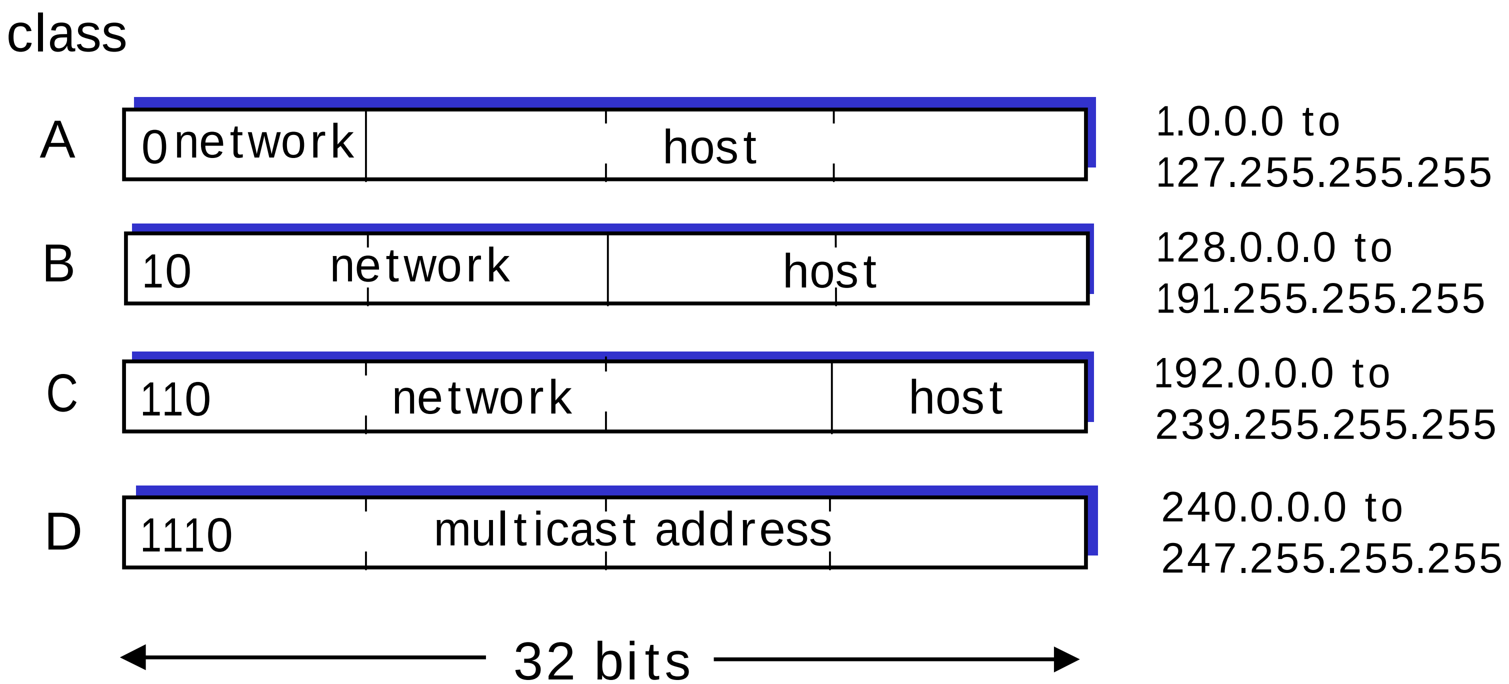
How to find the networks?

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



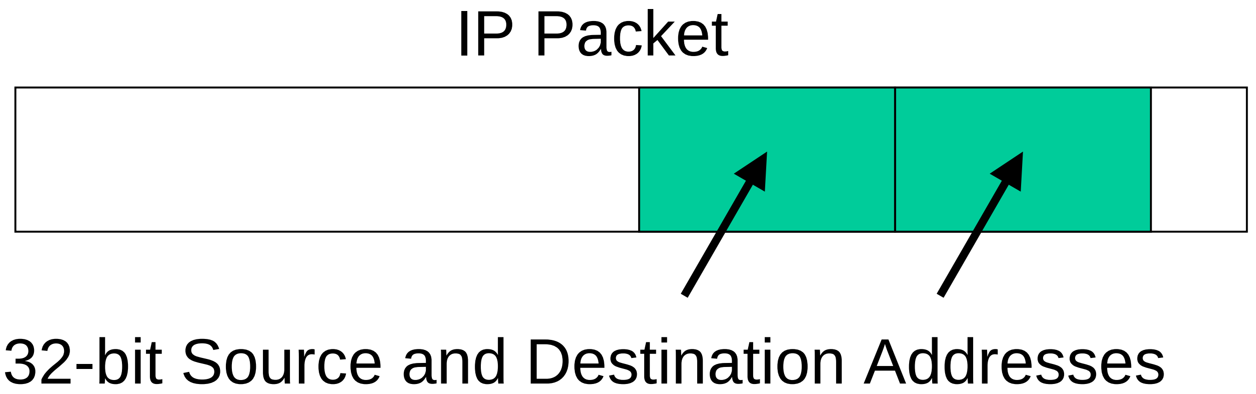
Interconnected system consisting of six networks

IP Addresses



Internet Addresses

- Also called IP addresses
  - Example: 128.171.17.13
  - Really 32-bit strings of ones and zeros
- Fit into source and destination address field of IP headers



Internet Address

- Hierarchical Addressing
- Two-Parts
  - Network part (organization on the Internet)
  - Local part (host on the network)
- Three-Parts
  - Network (organization on the Internet)
  - Subnet (suborganization)
  - Host on the subnet

Internet Addresses

- Two-Part
  - Divide Internet address into two parts
  - First part designates the network
  - Second (local) part designates the host on the network
  - Example:

Network Part

128.171.17.13

Local Part

Internet Addresses

- Three-Part
  - Local part is subdivided
  - Subnet part designates the subnet (suborganization)
  - Host part designates the host
  - Example:

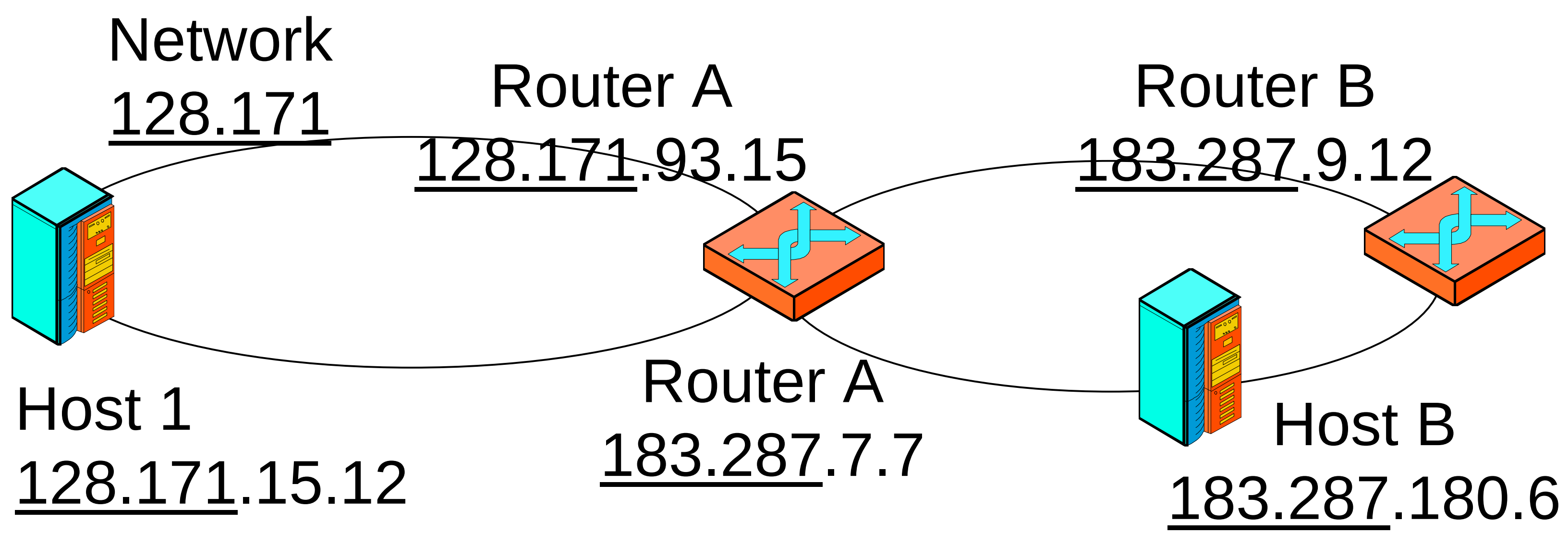
Network Part   Host Part

128.171.17.13

Subnet Part

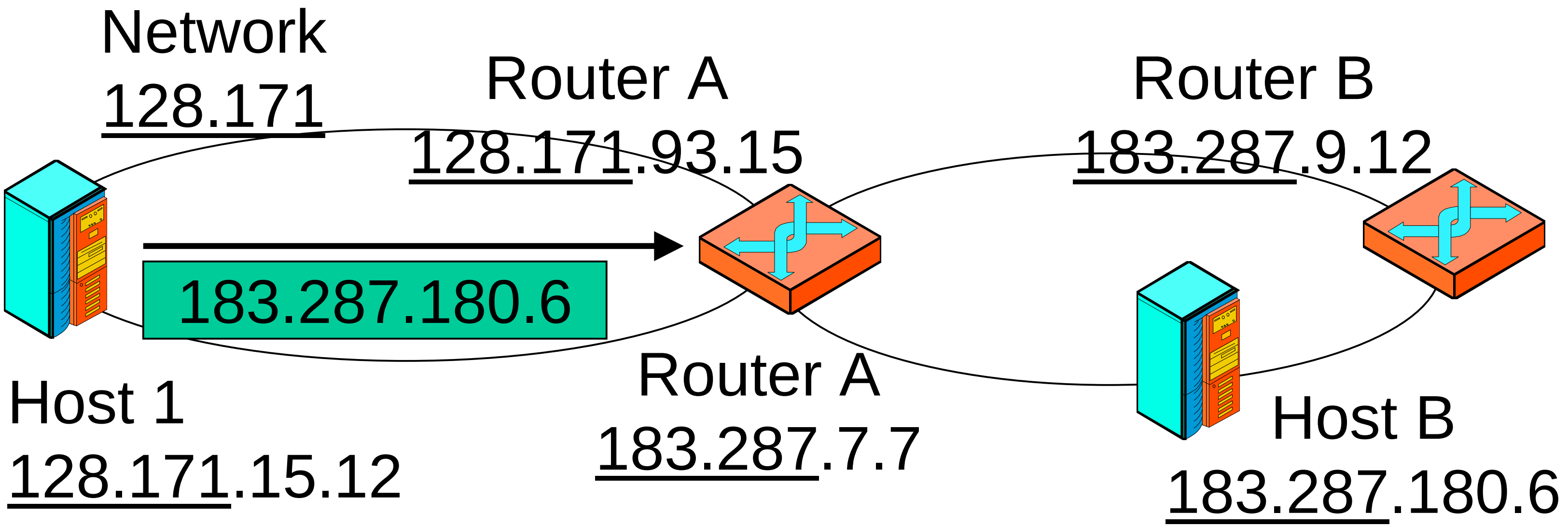
2-Part Internet Address and Routers

- Routers attach to multiple networks
  - Has an internet address on each network
    - 128.171.193.15 on network 128.171
    - 183.287.7.7 on network 183.287



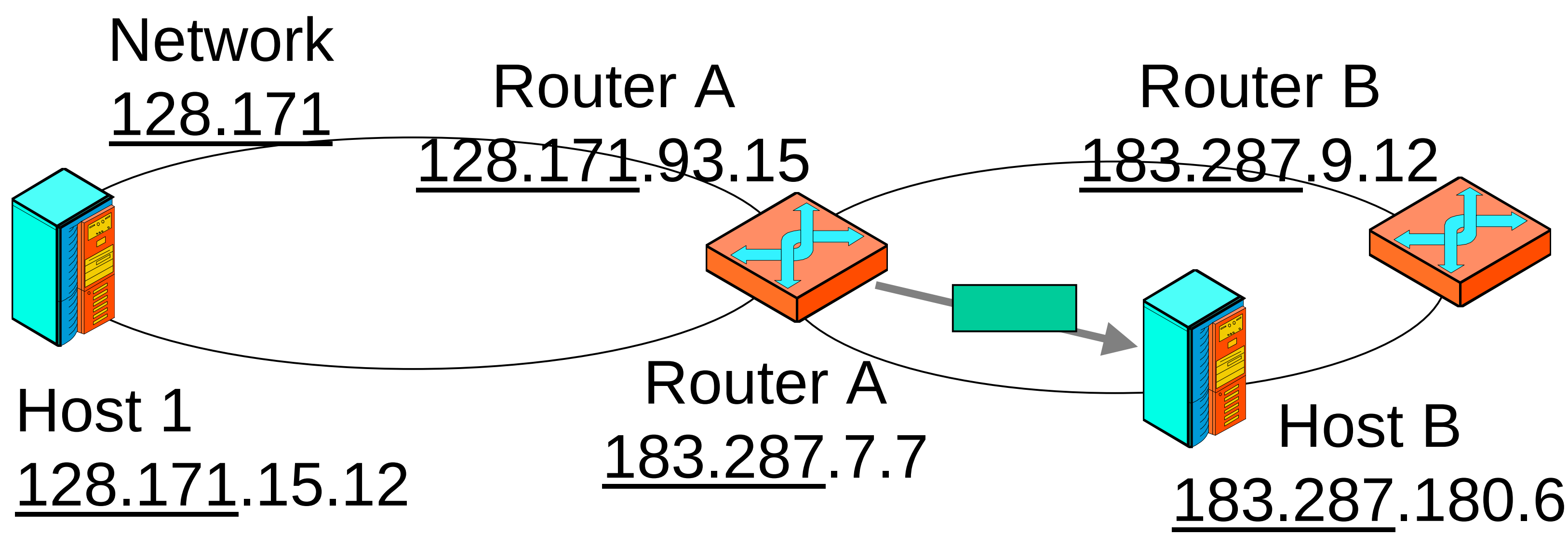
2-Part Internet Address and Routers

- Host sends IP packet to router
  - Router looks at destination address network part ONLY (183.287)
  - Compares to network parts of its own addresses (128,171, 183.287)



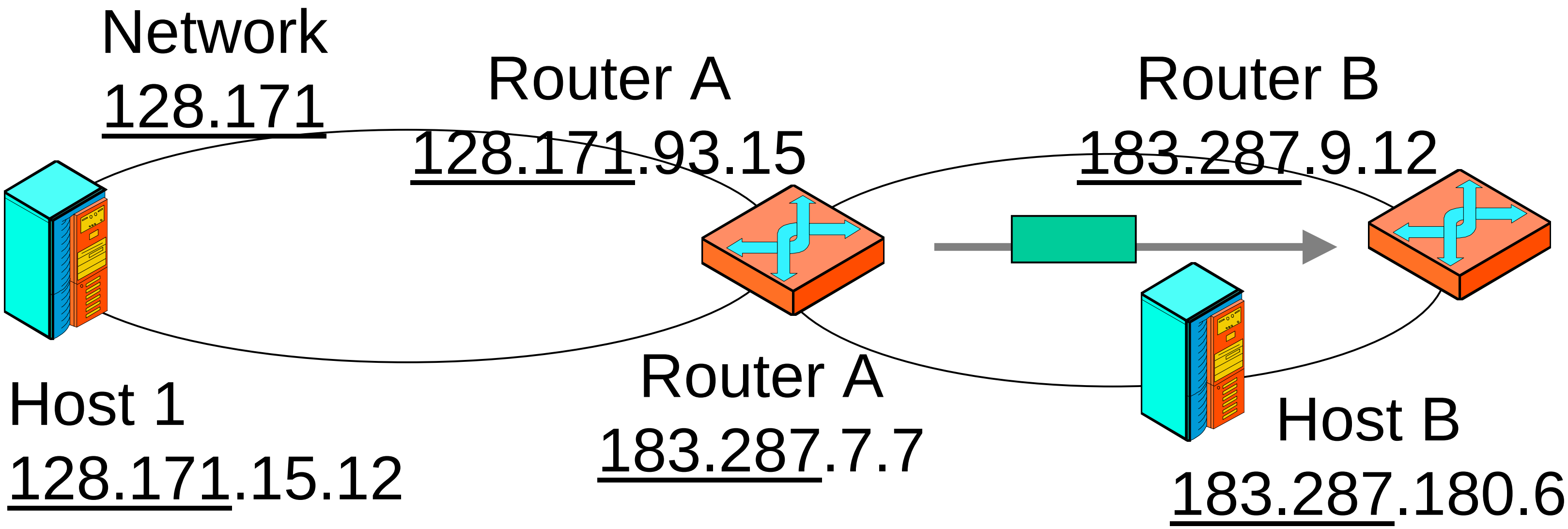
2-Part Internet Address and Routers

- If a network part matches (187.287)
  - The destination host is on that network
  - The router delivers it to the destination host



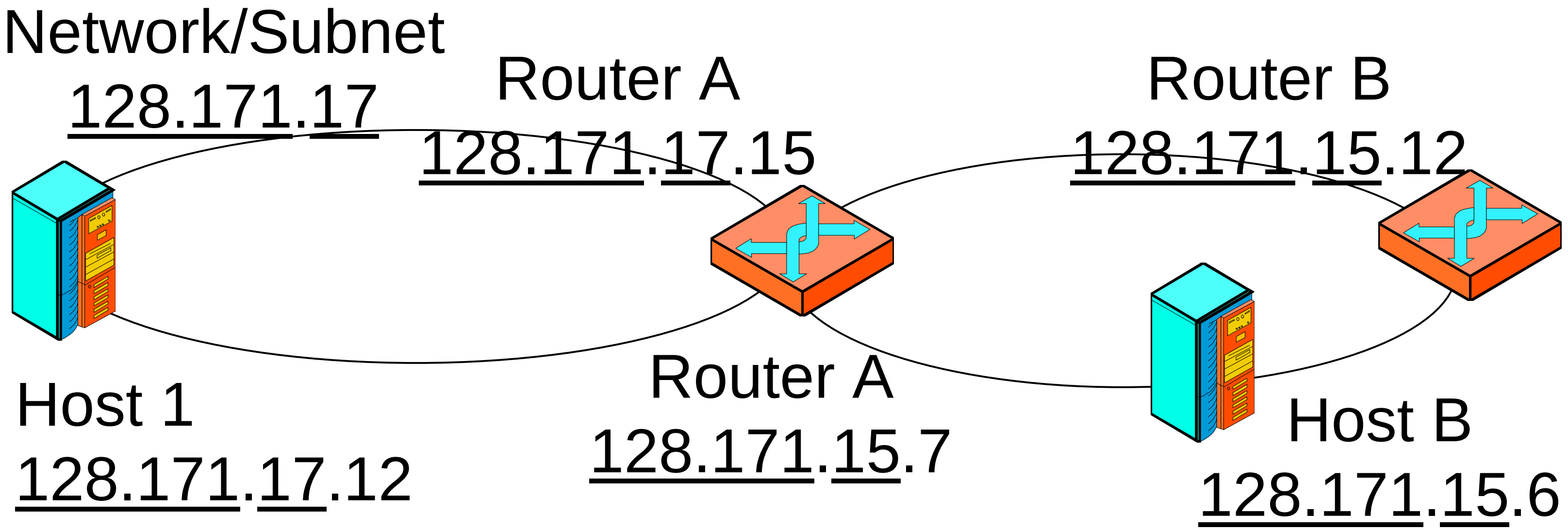
2-Part Internet Address and Routers

- If no network part matches
  - Destination host is not on one of the router's networks
  - Passes the IP packet onto another router



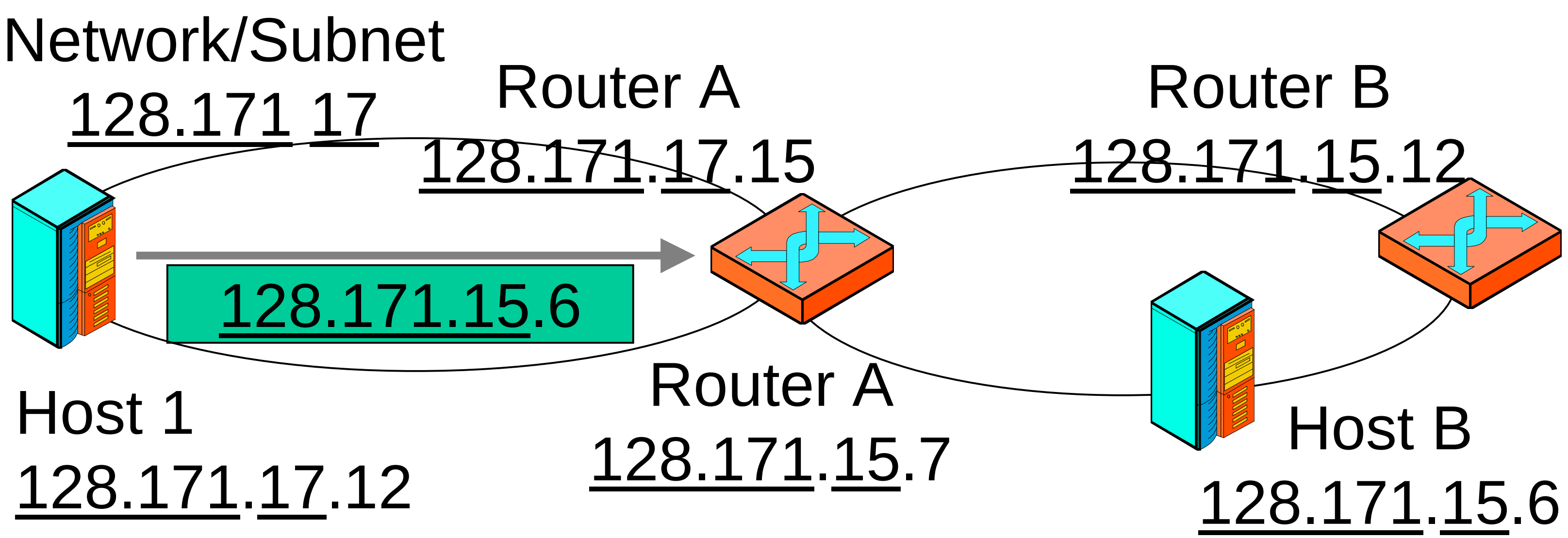
3-Part Internet Address and Routers

- Routers attach to multiple *subnets* (not networks)
  - Has an internet address on each subnet
  - Network PLUS subnet part underlined



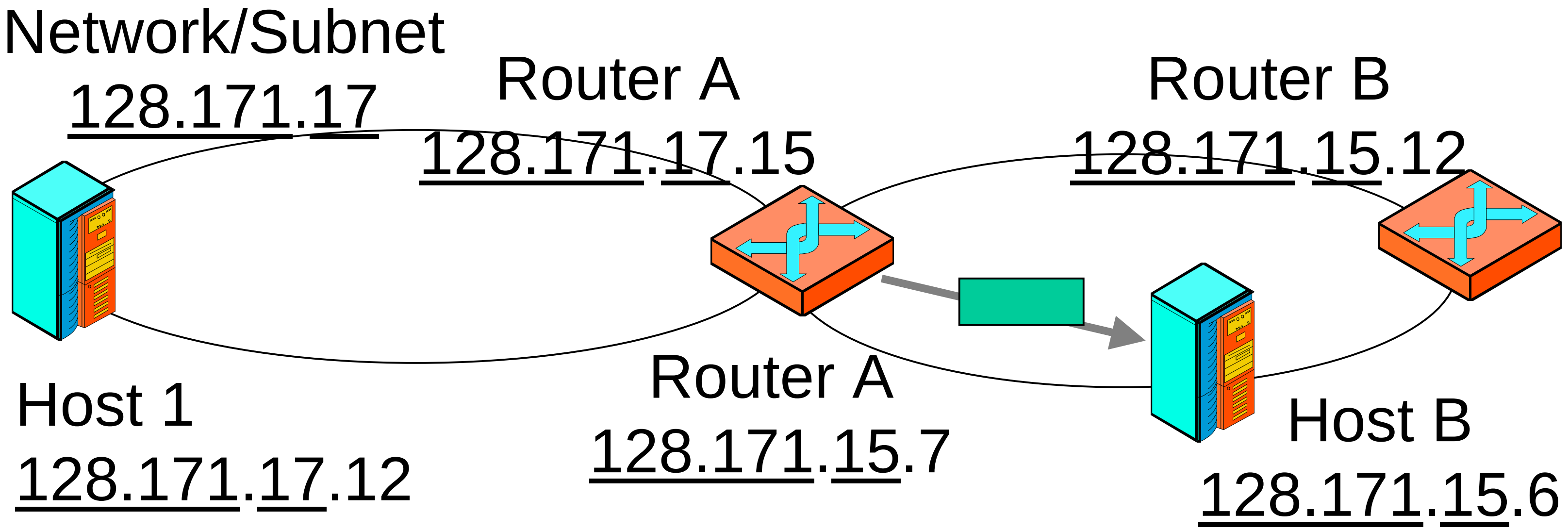
3-Part Internet Address and Routers

- Host sends IP packet to router
  - Router looks at destination address network plus parts ONLY (128.171.15)
  - Compares to network plus subnet parts of its own addresses (128.171.17, 128.171.15)



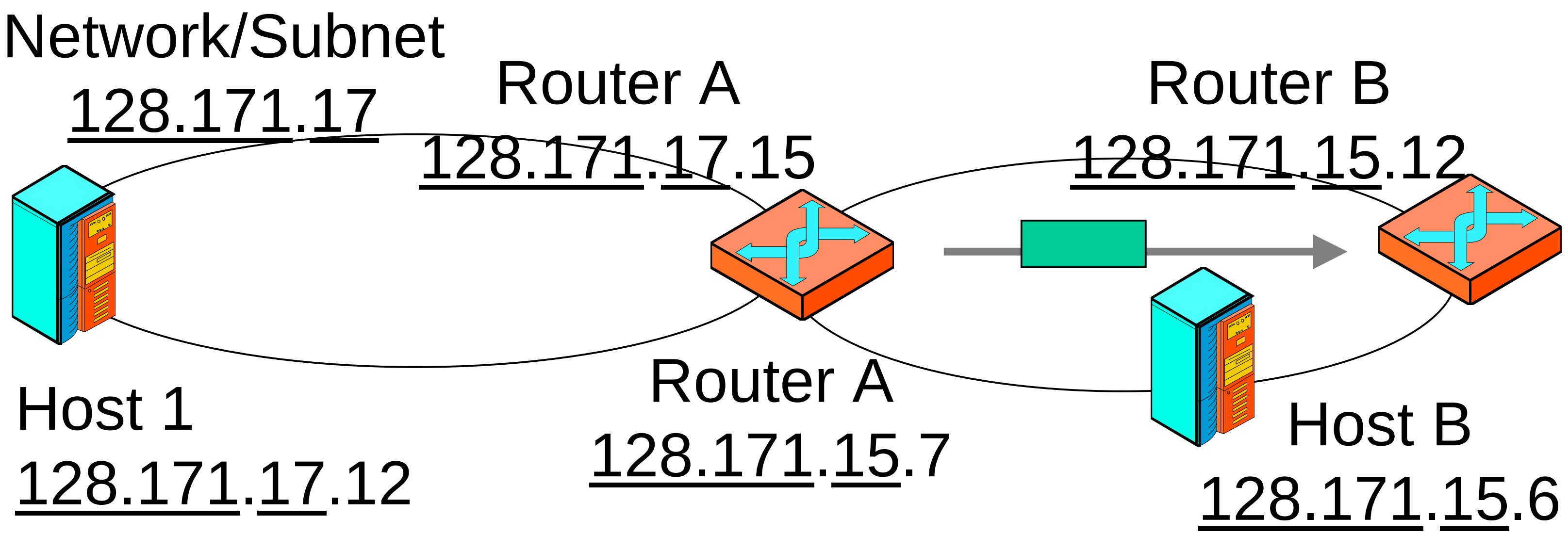
3-Part Internet Address and Routers

- If a network plus subnet part matches (128.171.15),
  - The destination host is on that subnet
  - The router delivers it



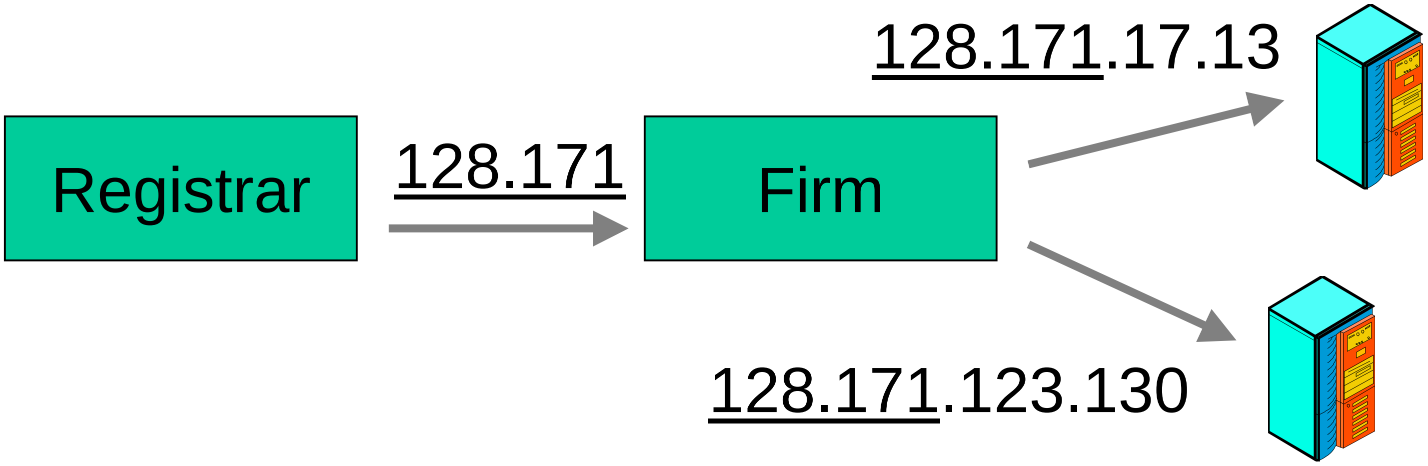
3-Part Internet Address and Routers

- If no network plus subnet part matches
  - Destination host is not on one of the router's subnets
  - Passes the IP packet onto another router



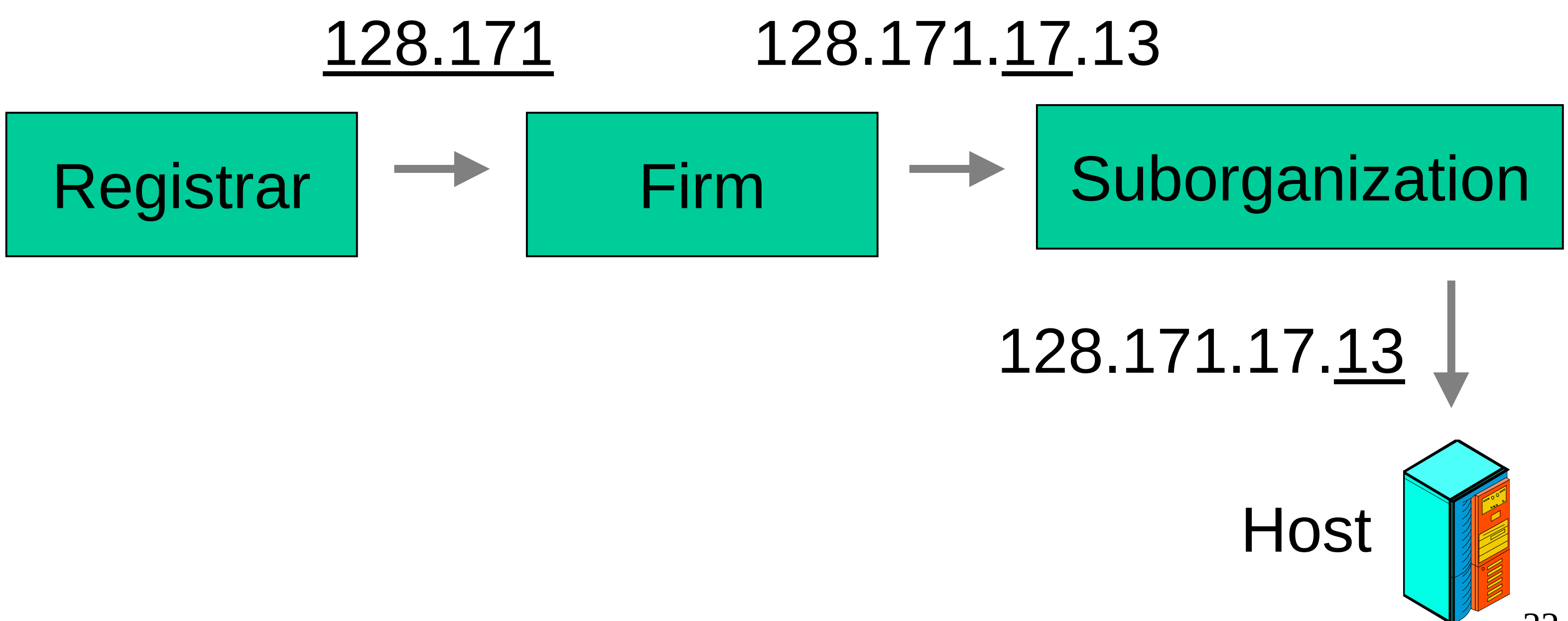
Assigning Two-Part Internet Addresses

- Organization applies to an internet registrar
  - It is given a two-part internet address
  - It assigns the local part to hosts internally
  - Only large organizations and ISPs get two-part addresses



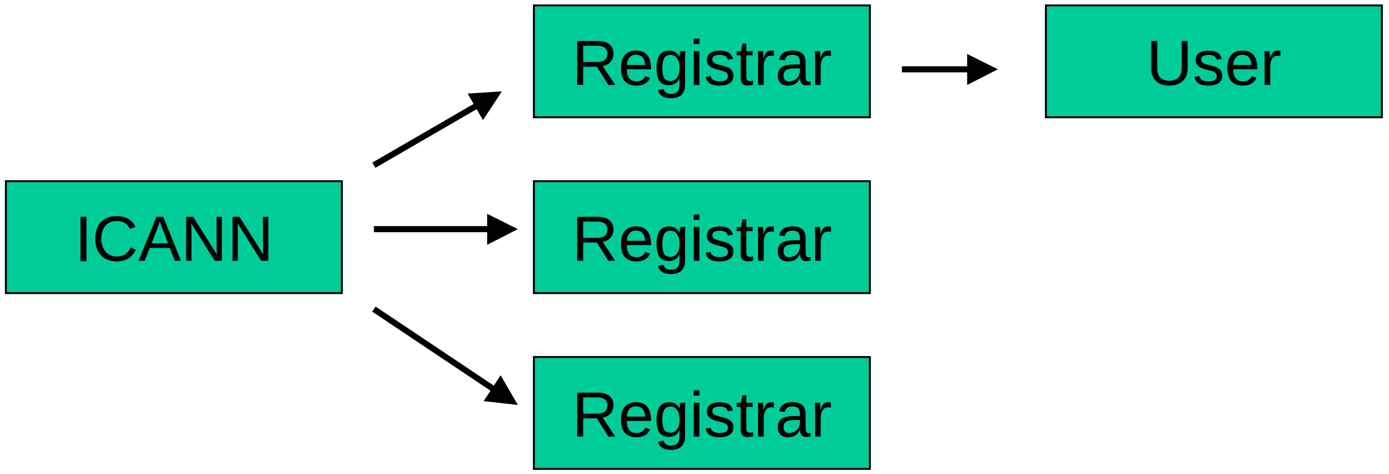
Assigning 3-Part Internet Addresses

- Organization has 2-part internet address
  - Assigns subnet part to suborganization
  - Suborganization assigns host bits to hosts

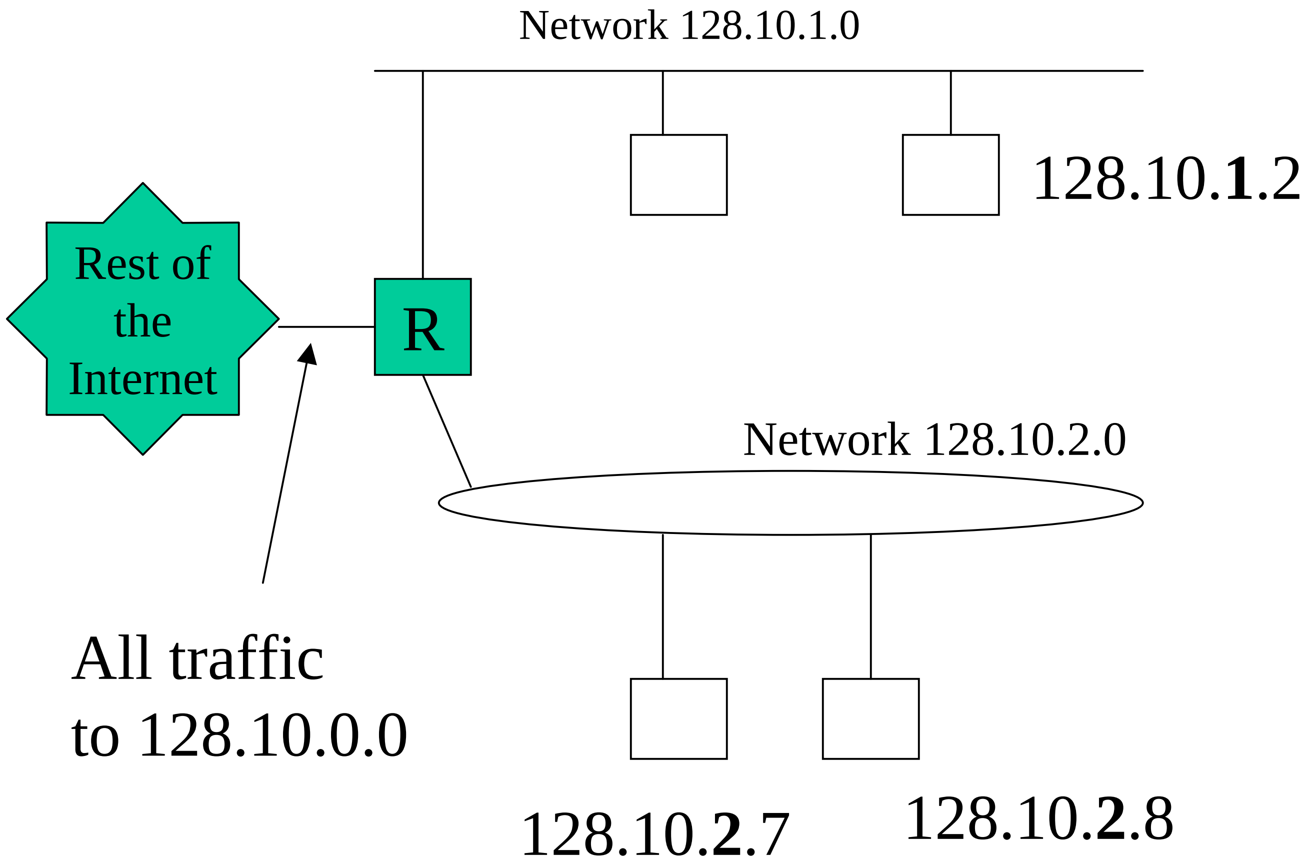


ICANN

- Internet Corporation for Assigned Numbers and Names
  - Is being given overall control over name and number assignments
  - Will work through multiple registrars.



Subnet addressing



Subnet masks

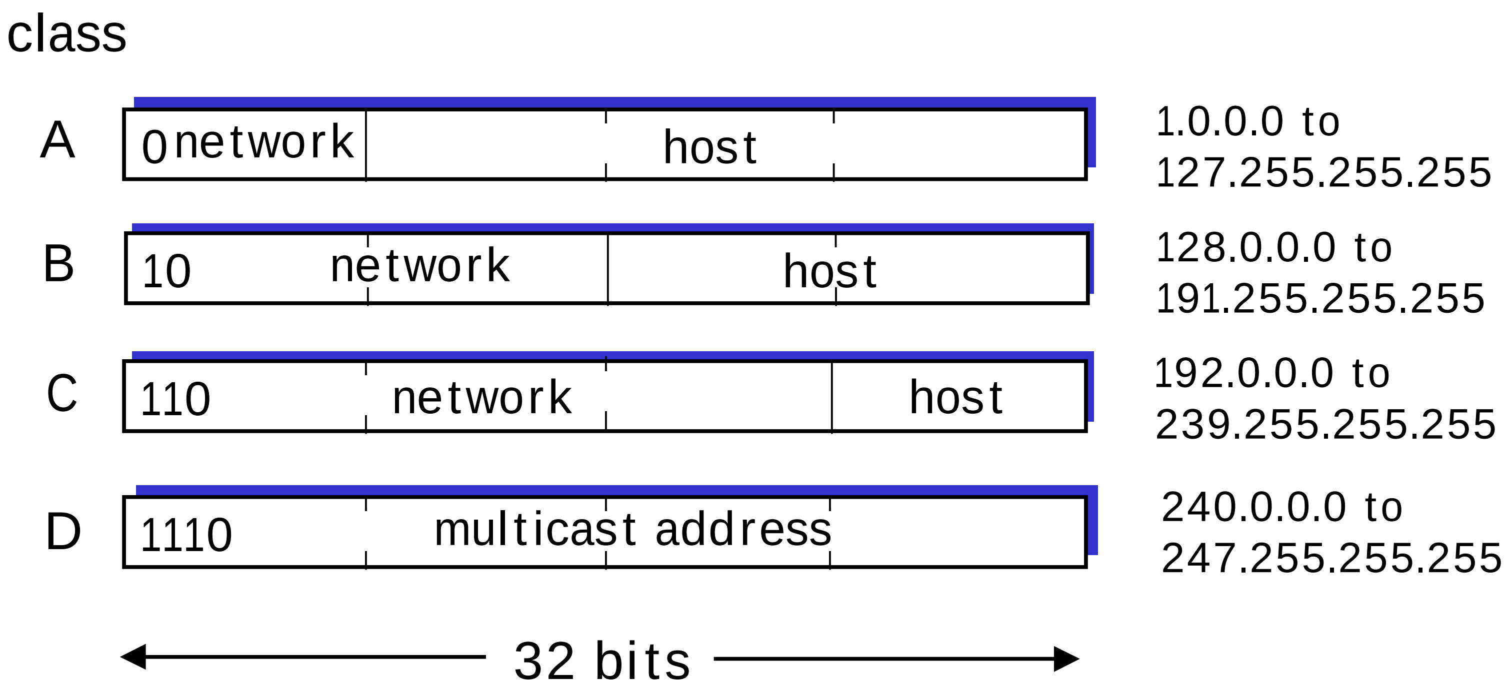
- Choosing a subnet addressing scheme is synonymous with choosing how to partition the local portion of an IP address into physical subnet and host part.
- The standard specifies that a site using subnet addressing must choose a 32-bit subnet mask for each network.
- Bits in the subnet mask are set to 1 if the network treats the corresponding bit in the IP address as part of the network address, 0 it it treats the bit as part of the host identifier.

  11111111 11111111 11111111 00000000 -> 255.255.255.0

Subnet Masks

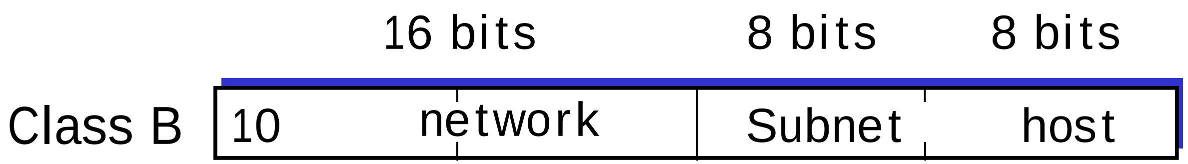
- By very definition of the network and host portions, each class network has a natural mask.
  - Class A natural mask 255.0.0.
  - Class B natural mask 255.255.0.0
  - Class C natural mask 255.255.255.0
- Without subnetting, network numbers would be of very limited use. The subnetting technique increases the number of subnetworks and reduces the number of hosts.
- A mask of 255.255 0.0 is applied to a network 10.0.0.0.
  - This increases from a single network 10.0.0.0 to 256 subnetworks ranging from 10.0.0.0 to 10.255.0.0.
  - This however decreases the number of hosts per each subnet from 16777216 to 65536 (ignoring boundaries).

IP Addressing



Subnetting

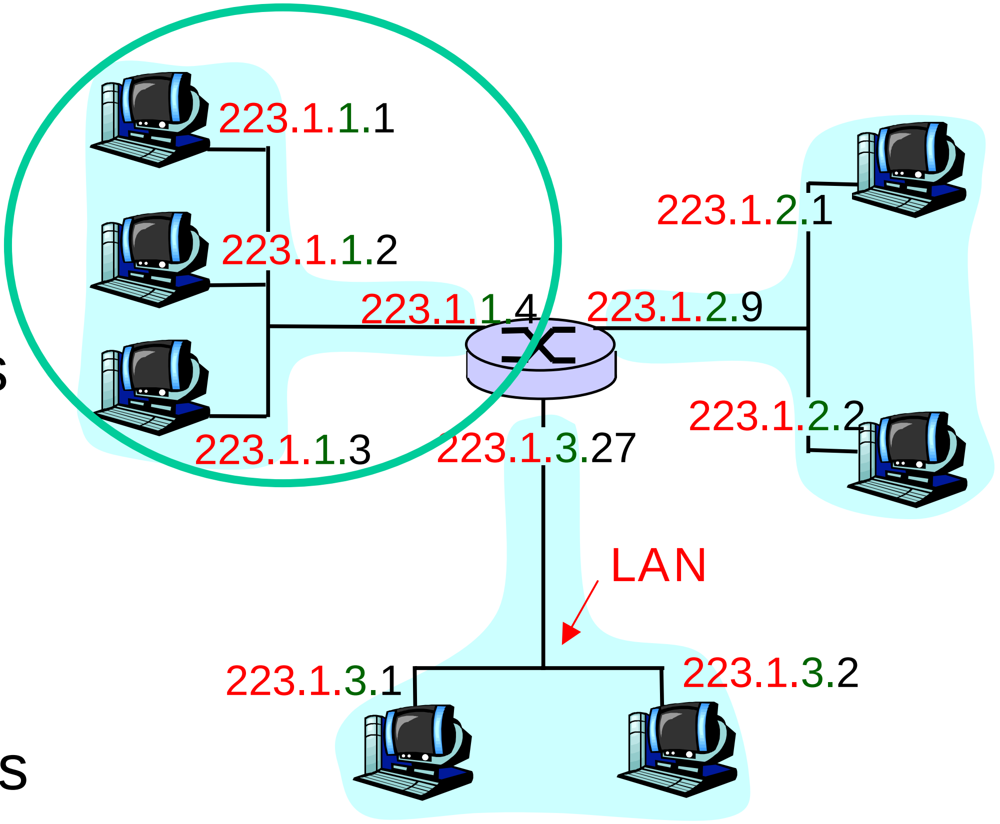
- Detailed in RFC 950



- Subnetting splits the host address into two parts
- Size of each part up to administrator.
- Only 8 bits to subnet in a class C address!

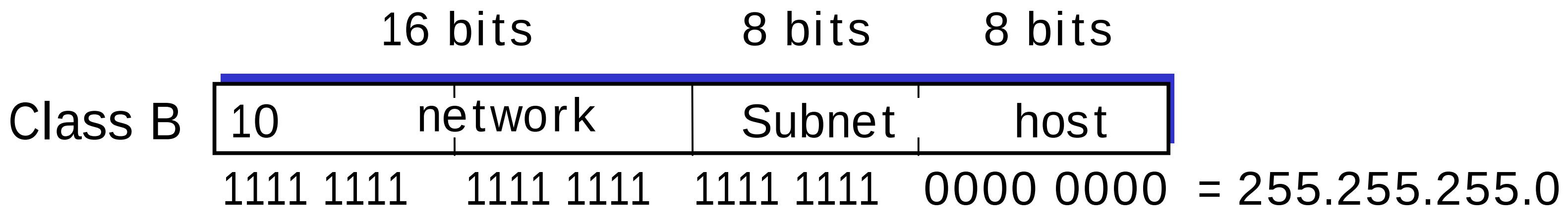
Subnetting

- Network ID. Subnet. Host
- Reduces the size of routing tables.
- One external class B routing table entry instead of 256 class C addresses.
- Changes to subnets does not require external announcements.



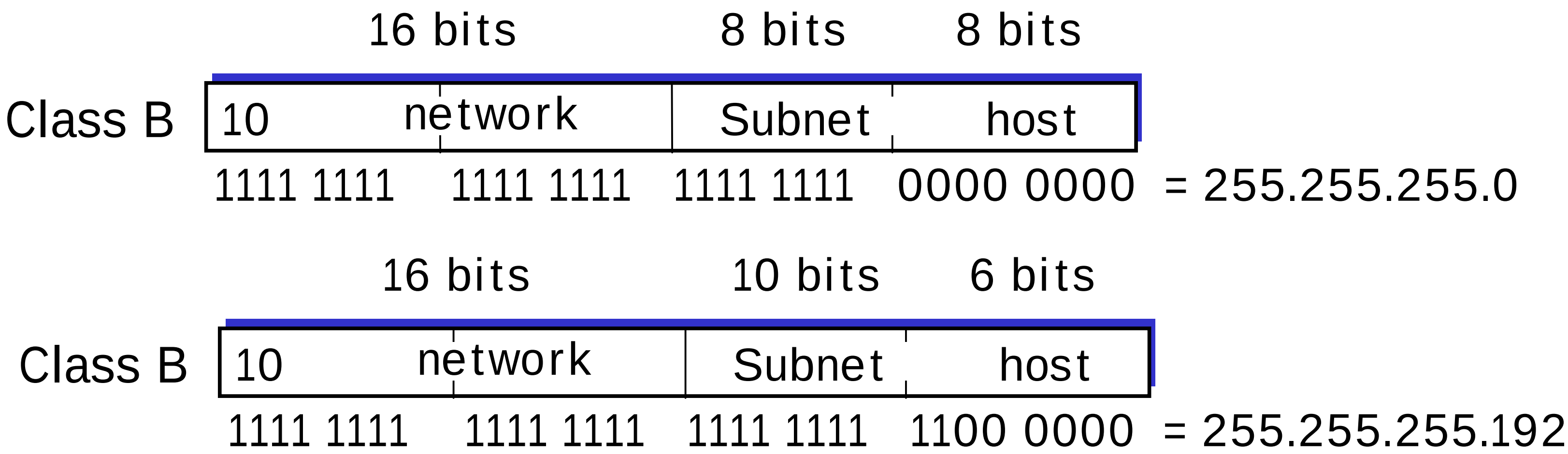
Subnetting inside a domain

- Internal routers must be aware of the subnet sizes.
- Subnet Masks identify the size of subnets.



- 32-bit value containing 1s over the network and subnet Ids, and 0s over the host IDs.
- & with any host to determine subnet number.

Subnet Masks



Using the high order bits of your IP address, you can determine your class (A,B, or C).

Netstat -nr (unix command)

```
> netstat -nr

Routing Table:
  Destination           Gateway            Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
128.114.48.0         128.114.49.15         U        3   1218  hme0
224.0.0.0            128.114.49.15         U        3      0  hme0
default              128.114.48.1          UG       0   7977
127.0.0.1            127.0.0.1             UH       0   4182  lo0
```

Subnet Arithmetic

- Host address 128.114.49.15
- subnet mask: 255.255.248.0

```
    10000000.1110010. 00110 001.00001111
&   11111111.11111111. 11111 000.00000000
----------------------------------------

             (Class B). 00110 000.00000000
 this host is on subnet 6
 or sometimes written as 128.114.48.0
```

Host routing with Subnets

Hosts search routing tables for

1. Matching host address (same LAN)
2. Matching subnet address
3. Matching network address
4. Default route

Address Resolution Protocol (ARP)

- Interface between Link layer and Network Layer.
- Allows hosts to query who owns an IP address on the same LAN.
- Owner responds with hardware address.
- Allows changes to link layer to be independent of IP addressing.
- (example to come)