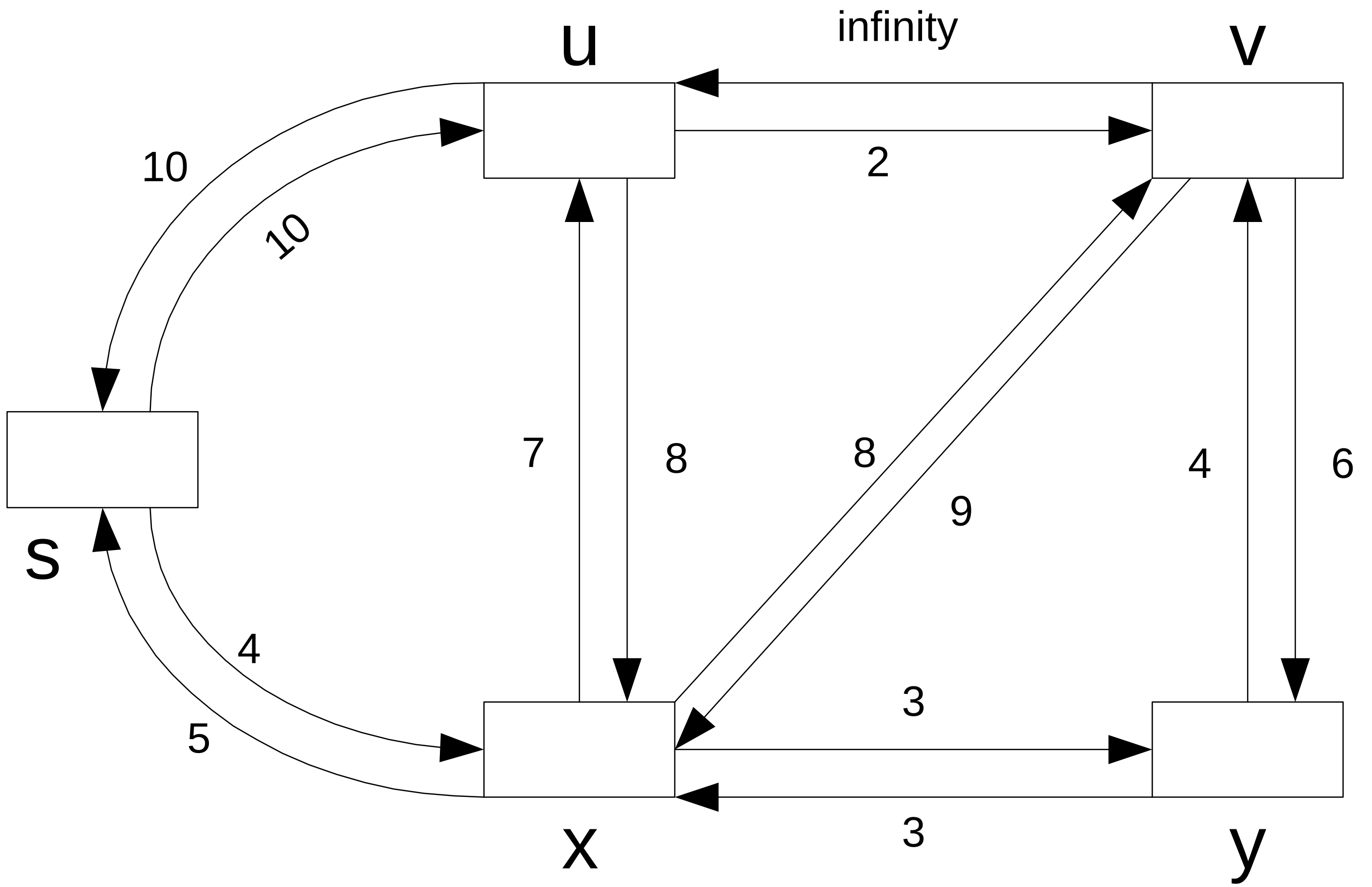
Today's Agenda

Review of Internet routing algorithms

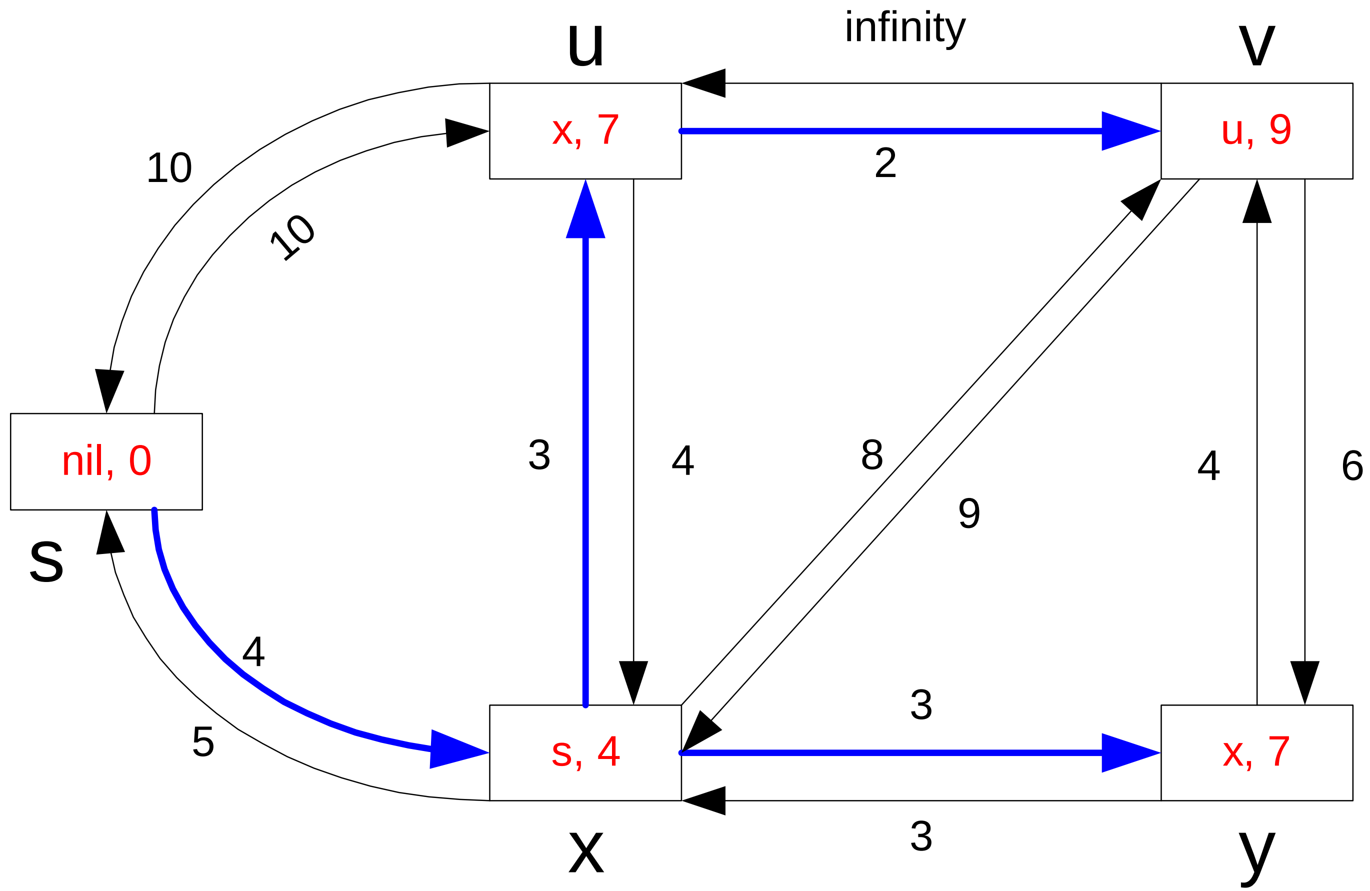
- Shortest path algorithms
  - Dijkstra algorithm
  - Bellman-Ford algorithm
- Routing algorithms in practice
  - Link-state routing
  - Distance vector routing
- Routing protocols
  - Intra-AS routing: OSPF, RIP, IS-IS, Static
  - Inter-AS routing: BGP

Networks as Graphs



- Weights: combination of bandwidth, load, delay, ...

Shortest Path Tree



- Directed tree, links go out from root to leaves
- The unique path from root to any node v is the shortest path from the root to v

Single-source Shortest Path Algorithms

- Dijkstra algorithm
- Bellman-Ford algorithm
- Input:
  - Direct graph $G = (V,E)$
  - A weight function $w: E \rightarrow R^+$
  - A source s
- Output:
  - A shortest path tree rooted at s

Basic Data Structures and Functions

- To build the SPT, each node maintain two fields:
  - *p[v]*: the pointer to the parent of v in the tree
  - *c[v]*: the least cost from s to v
- Init():
  - For each vertex v, $c[v] = \infty$, $p[v] =$ NIL
  - $c[s] = 0$
- Improve(u,v), where (u,v) is a directed edge of G
  - if $c[v] > c[u] + w(u,v)$ then
    $c[v] = c[u] + w(u,v)$
    $p[v] = u$

Dijkstra Algorithm

- Init()
- T = empty-set
- while (T != V)
  - $u \leftarrow$ a vertex not in T with minimum $c[u]$
  - $T = T \cup \{u\}$ // add u to T
  - for each v not in T so that (u,v) is an edge

    Improve(u,v)
- Note:
  - negative edges, the algorithm is slightly different
  - it can also fail if there is a negative cycle

Tips and Tricks 8

Some Dijkstra's quotes:

- "The question of whether a computer can think is no more interesting than the question of whether a submarine can swim."
- "Computer science is no more about computers than astronomy is about telescopes."
- "Object-oriented programming is an exceptionally bad idea which could only have originated in California."

Bellman-Ford Algorithm

- Init()
- For i=1 to |*V*|-1
  - for each edge (*u,v*) in G
    Improve(*u,v*)
- Can be modified to allow negative weights and negative cycles

Link State and Distance Vector Basics

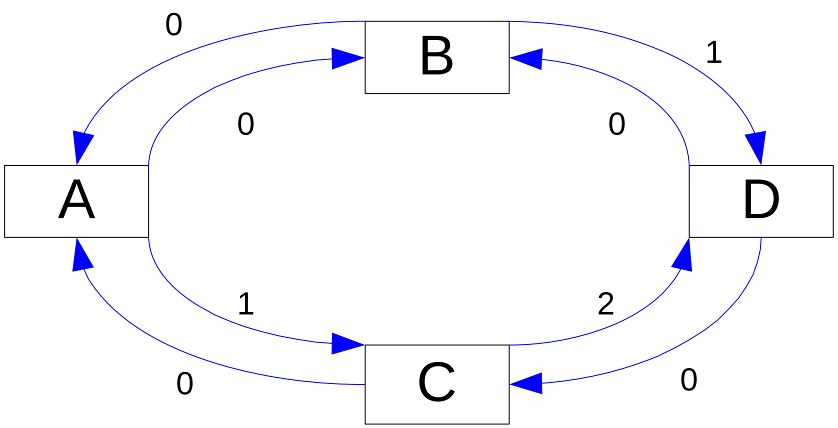
- Which corresponds to Dijkstra, Bellman-Ford?

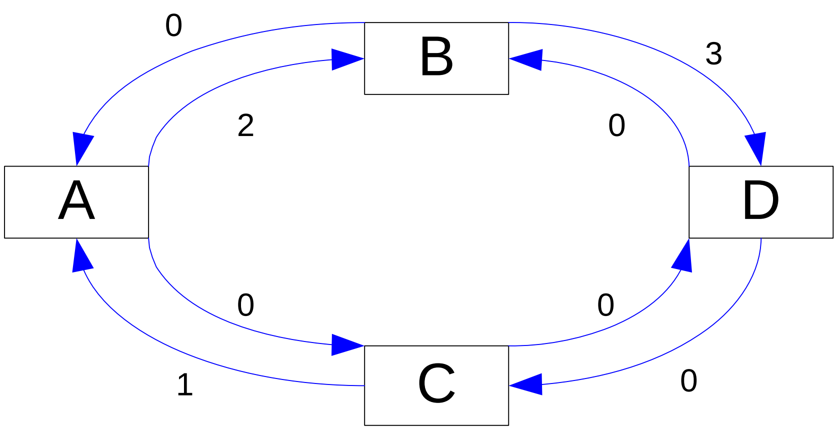
- Link State Routing
  - Each node sends neighboring link costs to all nodes
  - Each node computes routing table separately
  - Question: what if two nodes compute two different SPTs (given the same graph?)

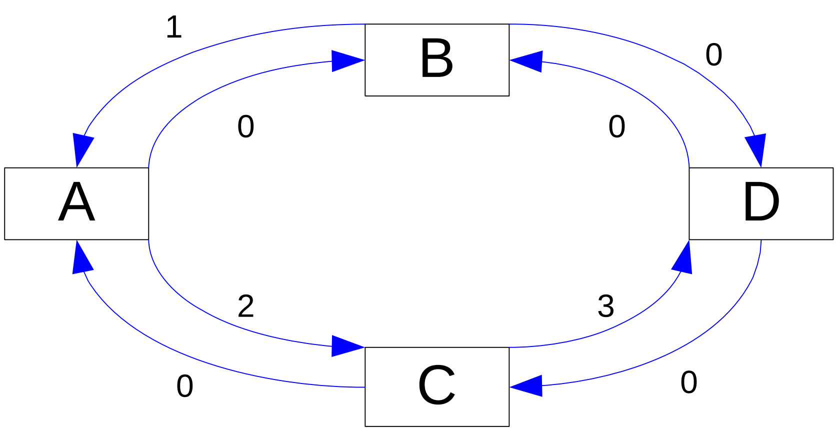
- Distance Vector
  - Each nodes sends estimates to all neighbors
  - Each node updates routing table accordingly

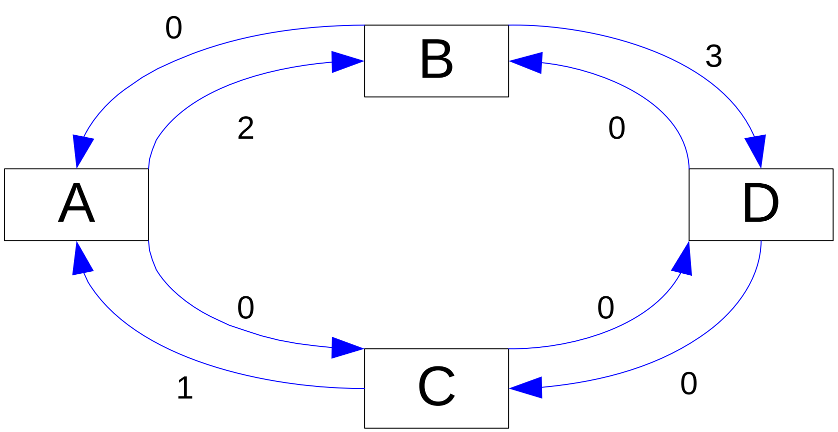
Oscillation Problem

- If link load is part of the cost









Count-to-infinity Problem

A	B	C	D	E	
	•	•	•	•	Initially
	1	•	•	•	After 1 exchange
	1	2	•	•	After 2 exchanges
	1	2	3	•	After 3 exchanges
	1	2	3	4	After 4 exchanges

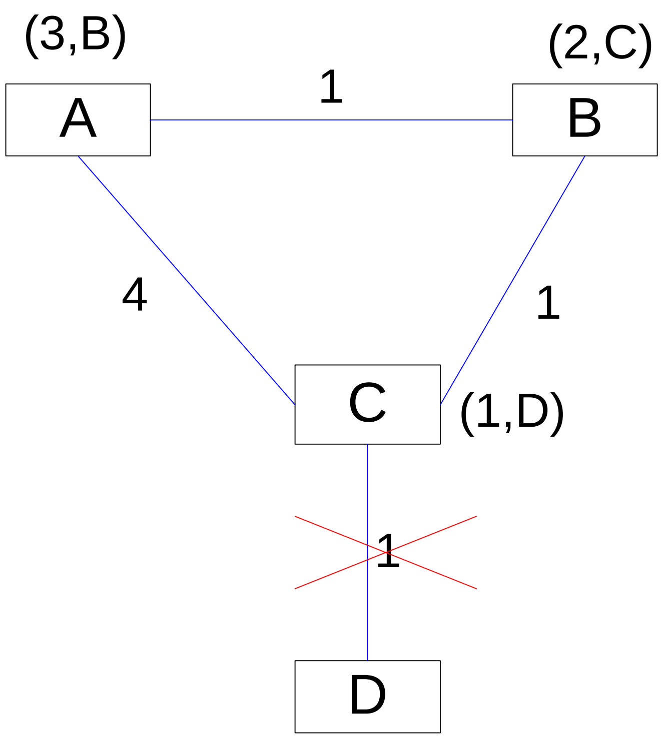
(a)

A	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
		⋮			
	•	•	•	•	

(b)

- What are some solutions?
  - Split horizon
  - With poisoned reverse (does it solve the problem?)
  - Triggered Updates

Poisoned Reverse Fails Too



A	B	C
(3,B)	(2,C)	(1,D)
(3,B)	∞	(7,A)
∞	(8,C)	(7,A)
(9,B)	(8,C)	∞
(9,B)	∞	(13,A)
∞	(14,C)	(13,A)
…	…	…

Triggered Updates

- How bad is count-to-infinity: it takes sometime to report unreachable destination
- To make bad news propagate faster, in practice we use triggered updates
  - send link status updates really quick hopefully before regular exchanges are done
  - Still do not solve the problem

Link State vs. Distance Vector (1)

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

Link State vs. Distance Vector (2)

- Message complexity
  - LS requires more messages to be exchanged, DV requires larger messages
- Speed of convergence
  - DV can converge slower, but LS needs more overhead to flood messages
- Robustness
  - LS is somewhat more robust since each node calculate all the routes by itself, less dependent on a few routers' errors
- Efficiency
  - Hard to say, neither is a winner, both are used in practice

Tips and Tricks 9

- What is a *file descriptor leakage* ?

Why Hierarchical Routing?

Suppose a single routing algorithm is used

- Does not scale well
  - 200 Mil destinations can’t be stored in memory entirely
  - LS: overhead required to broadcast link status
  - DV: likely never converge
- Politically incorrect

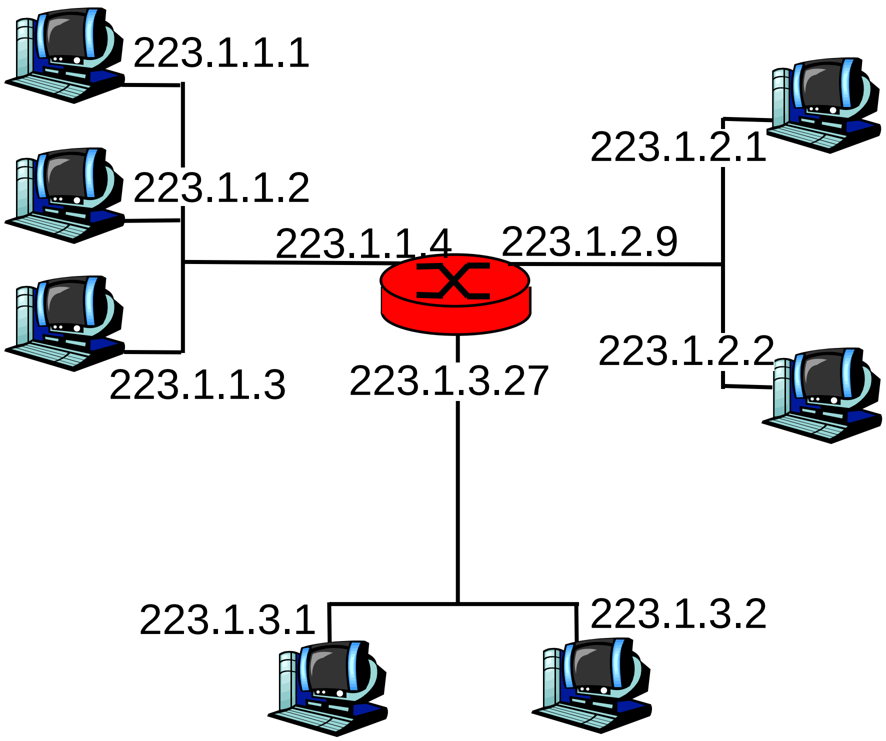
Hierarchical Routing (cont.)

- Intra-AS routing protocol or Interior Gateway Protocol
  - **Static**: used in very small domains
  - [DV] **RIP**: used in some small domains (has limitations)
  - [LS] **OSPF**: widely used in enterprise networks
  - [LS] **IS-IS**: widely used in ISP networks
  - Cisco's **IGRP** and **EIGRP**

- Inter-AS protocol or Exterior Gateway Protocol
  - **BGP** (v4)

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 each interface



223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

Classful IP Addresses

A	0	Few large organizations	1.0.0.0 to 126.0.0.0
B	10	Medium organizations	128.1.0.0 to 191.255.0.0
C	110	Small organizations	192.0.1.0 to 223.255.255.0
D	1110	Multicasting	224.0.0.0 to 239.255.255.255
E	1111	Reserved	240.0.0.0 to 254.255.255.255

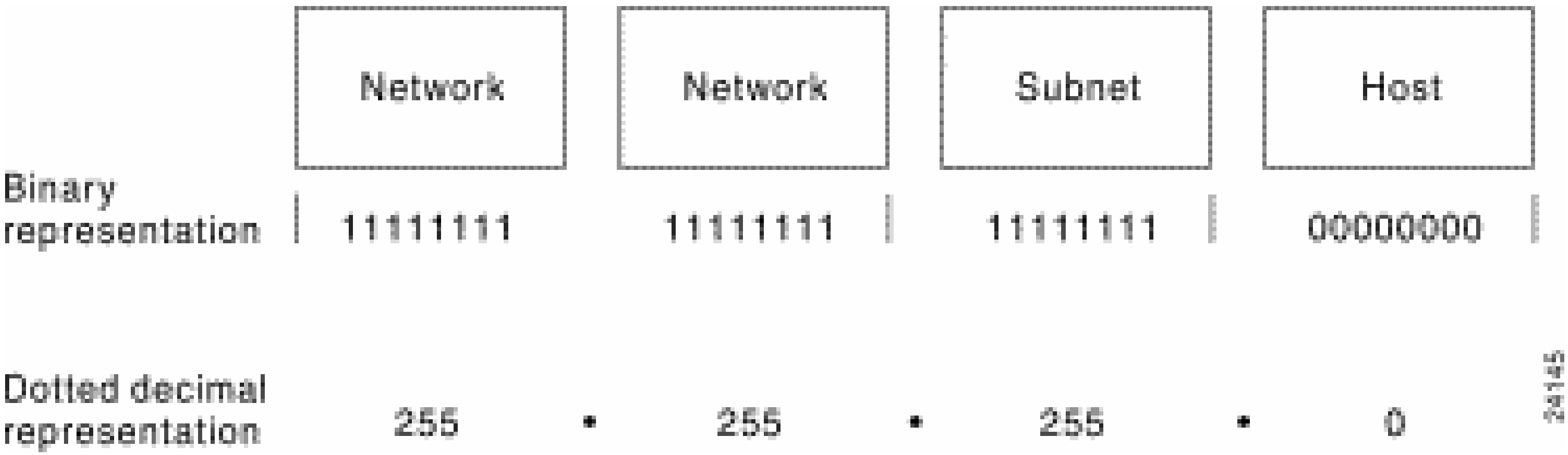
Are we missing something ?

Some Problems with Classful IP

- Waste lots of addresses
  - A class C network is too small
  - A class B network (65534 hosts) is too large
  - Running out of Class B networks
  - Giving out class C blocks increase the routing table sizes of core routers

- What are current solutions?

First Solution: Subnetting



- The network-prefix and the subnet number form the extended network-prefix
- The rest is host-number
- Modern terminology: /x networks
  - e.g. a class B is /16, class C is /24, class A is /8
  - a subnet of a class B could be /20, ...

Problem with Subnetting

- Each organization has a fixed number of subnets of fixed sized
  - CSE should have more hosts than philosophy dept.

- Reason: earlier routing protocols (RIPv1 – *routed*) did not pass the masks along with routes → a single mask used through out an organizational network

Second Solution: VLSM

- Variable Length Subnet Mask (RFC 1009 – 1987)
- Much more efficient use of address space
- Allows route aggregation (what is it?), but require:
  - Routing protocols to carry extended network prefix
  - Routers must implement “longest match” (why?)
  - Addresses must have topological significant
- Examples:
  - /8 into multiple /16
  - Some /16 into multiple /24
  - Some /16 into multiple /18
  - Some /24 into multiple /27

Longest Match Forwarding

- Destination:
  - 11.1.2.5 = 00001011.00000001.00000010.00000101
- Route 1:
  - 11.1.2.0/24 = 00001011.00000001.00000010.00000000
- Route 2:
  - 11.1.0.0/16 = 00001011.00000001.00000000.00000000
- Route 3:
  - 11.0.0.0/8 = 00001011.00000000.00000000.00000000
- Router must pick route 1

Third Solution: CIDR

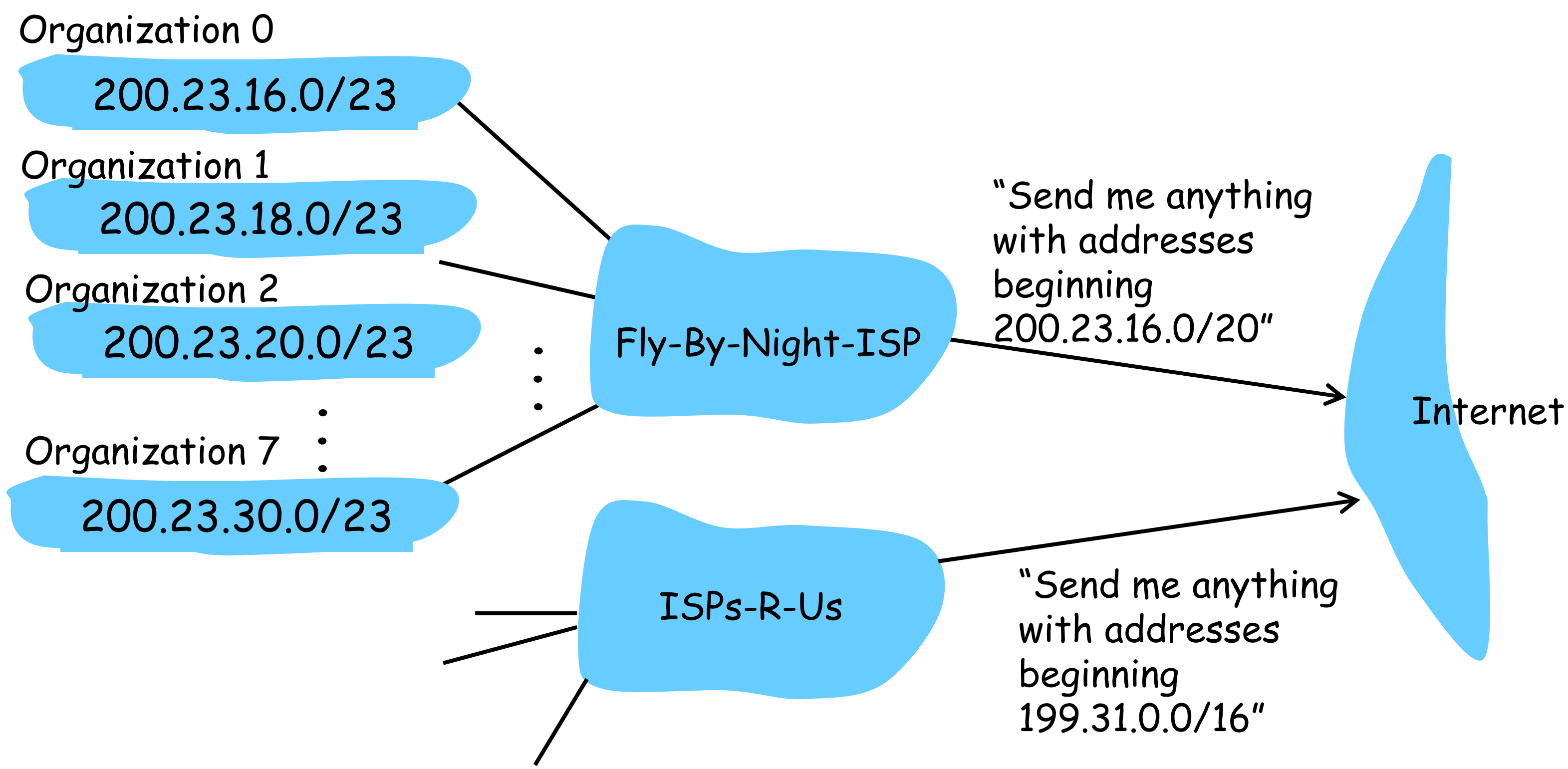
- Just the same as VLSM, but
  - VLSM is performed on previously assigned space to an organization, invisible to the internet
  - CIDR is global
- Example:
  - an organization gets a /20 from a /8 of an ISP
  - later the organization switches to another ISP
  - changing all IPs of all computers is crazy
  - so, keep the IPs and use CIDR, the second ISP gateway router advertise the subnet of the organization too
  - requires Longest Match to work

Another Solution

- Using NAT and Private Address Space (RFC 1918)
  - 10.0.0.0 – 10.255.255.255
  - 172.16.0.0 – 172.31.255.255
  - 192.168.0.0 – 192.168.255.255

Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:



Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1