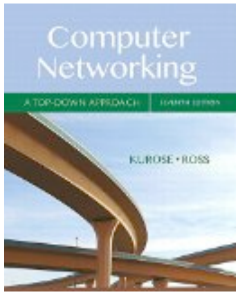
COMP 375: Lecture 31



- **News & Notes:**
  - Quiz #7 in class Friday
  - Project #5
    - **Protocol Spec Due**: Monday (April 23)
    - **Code Due**: Mon, April 30
- **Reading (Fri, Apr. 20)**
  - Sections 5.3 – 5.4 (Inter-AS Routing)

US-based ISPs are now free to implement "fast lanes" for certain traffic**.

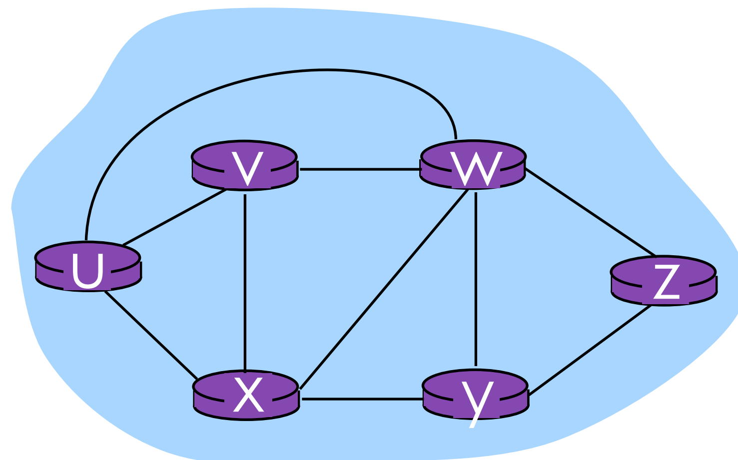
Discuss: Based on what we've seen, how do you think an ISP could prioritize traffic to a website like Netflix?

*** The FCC voted to end "net neutrality" in December 2017.*

Chapter 5

NETWORK CONTROL PLANE

We model networks as graphs, with nodes representing routers.

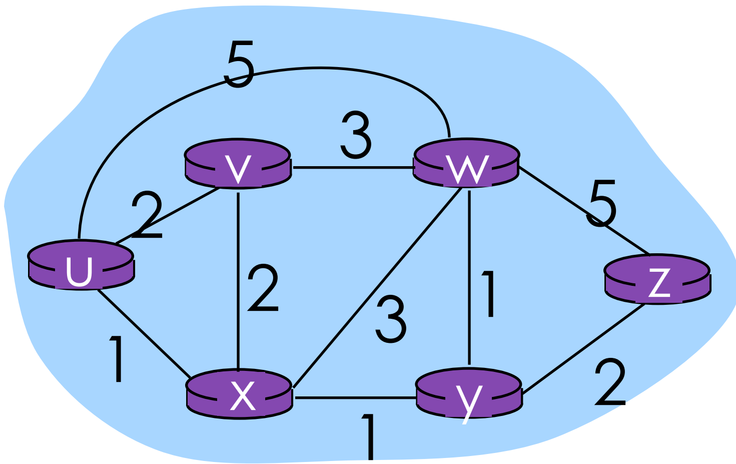


graph: G = (N,E)

N = set of routers = { u, v, w, x, y, z }

E = set of links = { (u,v), (u,x), (v,x), (v,w), (x,w), (u,w), (x,y), (w,y), (w,z), (y,z) }

Routing algorithms calculate the best path based on link costs.



$c(n_1,n_2)$: cost of link (n_1,n_2)

Cost of path $(n_1, n_2, n_3, \ldots, n_p) = c(n_1,n_2) + c(n_2,n_3) + \ldots + c(n_{p-1},n_p)$

How should link **costs** be determined?

A.	They should all be equal.
B.	They should be a function of link capacity.
C.	They should take current traffic characteristics into account (congestion, delay, etc.).
D.	They should be manually determined by network administrators.
E.	They should be determined in some other way.

Link costs are often all equal, with network admins tweaking them based on policy.

Discuss: *Why might a network admin change the cost of a link?*

Routing Challenges

1. *How to choose best path?*
2. *How to scale to millions of users?*
3. *How to adapt quickly to failures or changes?*

How much information should a router know about the network?

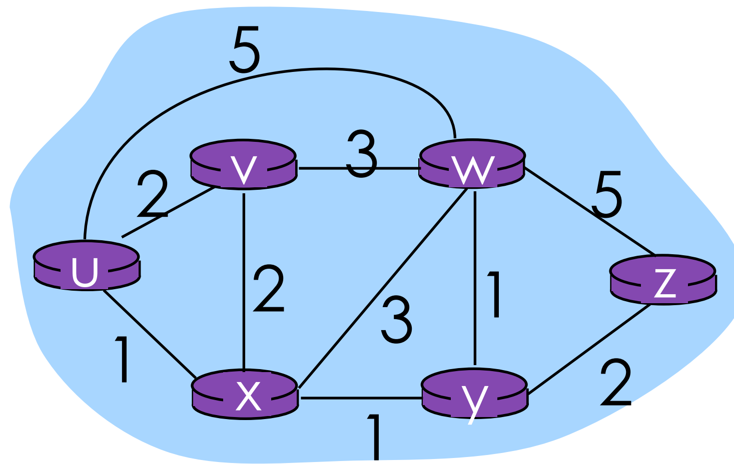
A.	The next hop and cost of forwarding to its neighbor(s).
B.	The next hop and cost of forwarding to any destination.
C.	The status and cost of every link in the network.
D.	Some other amount of information.

Less state.

Better decisions.

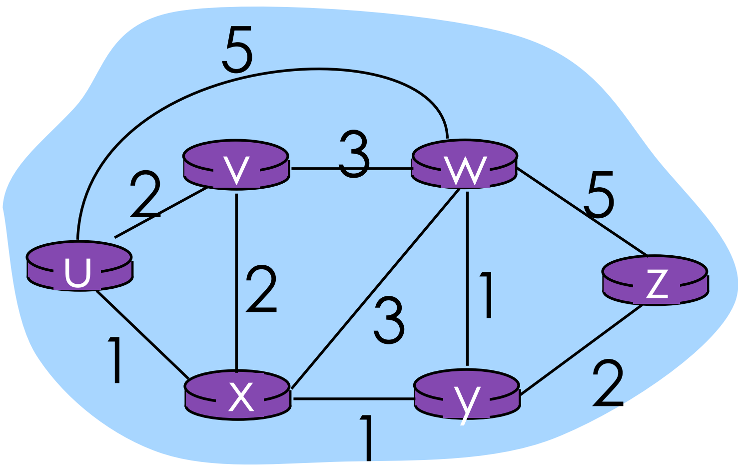
At a minimum, the routing table at U needs to know next hop for each possible destination.

Dest	Next Hop
V	V
X	X
W	X
Y	X
Z	X



Our routing table should probably contain more info (e.g. path cost).

Dest	Next Hop	Cost (Path)
V	V	2
X	X	1
W	X	4
Y	X	2
Z	X	4



Section 5.2

ROUTING ALGORITHMS

Today's routers use either link-state or distance-vector to calculate routes.

Link State (Global)

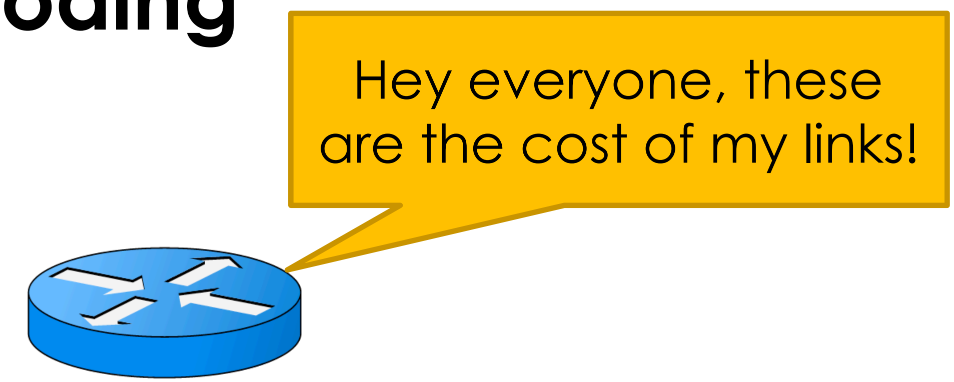
1. Routers maintain cost of each link in the network.
2. Connectivity/cost changes flooded to all routers.
3. Converges quickly (less inconsistency, looping, etc.).
4. Limited network sizes.

Distance Vector (Distributed)

1. Routers maintain next hop and cost of each destination.
2. Connectivity/cost changes iteratively propagate from neighbor to neighbor.
3. Requires multiple rounds to converge.
4. Scales to large networks.

Link-state (LS) routing has two phases: *reliable flooding* and *path calculation*.

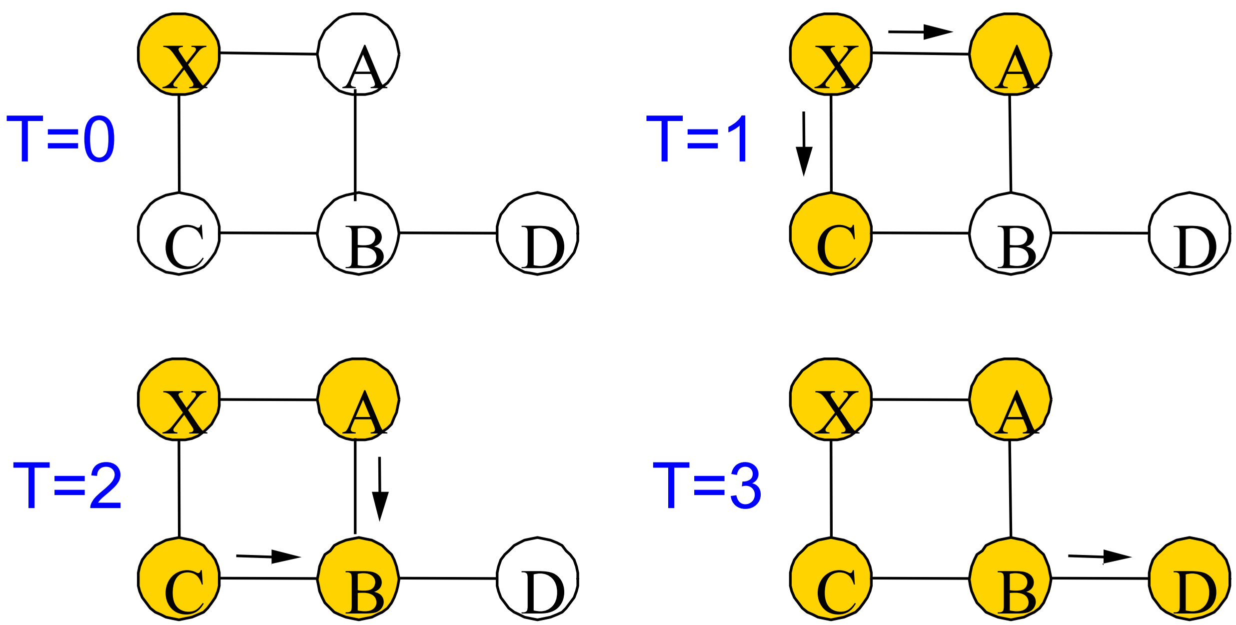
1. **Reliable flooding**



2. **Path calculation**

During flooding, routers forward link-state advertisements (LSAs).

Example: LSA generated by X at T=0



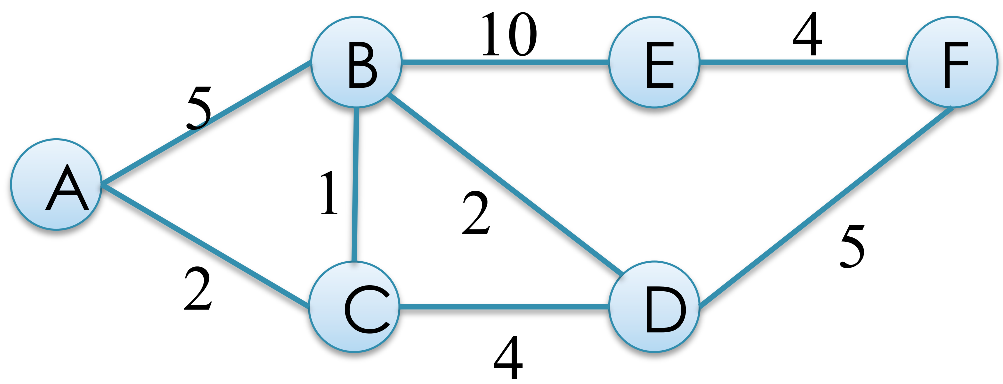
Dijkstra's Algorithm is one option for path calculation.

```
Initialization:
  N' = {u}
  for all nodes v
    if v adjacent to u
        then D(v) = c(u,v)
    else D(v) = ∞
```

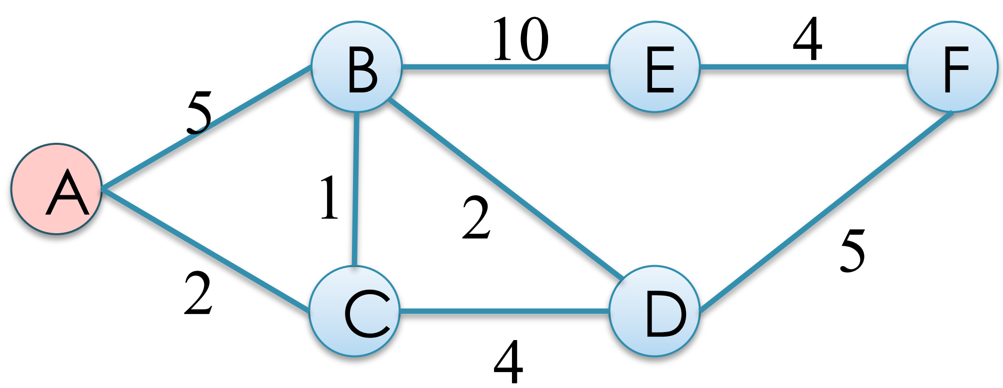
In Dijkstra’s Algorithm we calculate one new best cost each iteration.

```
Initialization:
  N' = {u}
  for all nodes v
    if v adjacent to u
        then D(v) = c(u,v)
    else D(v) = ∞
Loop
  find w not in N' such that D(w) is a minimum
  add w to N'
  update D(v) for all v adjacent to w and not in N' :
      D(v) = min( D(v), D(w) + c(w,v) )
  /* new cost to v is either old cost to v or known
   shortest path cost to w plus cost from w to v */
until all nodes in N'
```

In this example, we will try to determine the shortest paths from node A.



Dijkstra's Algorithm – Step 0



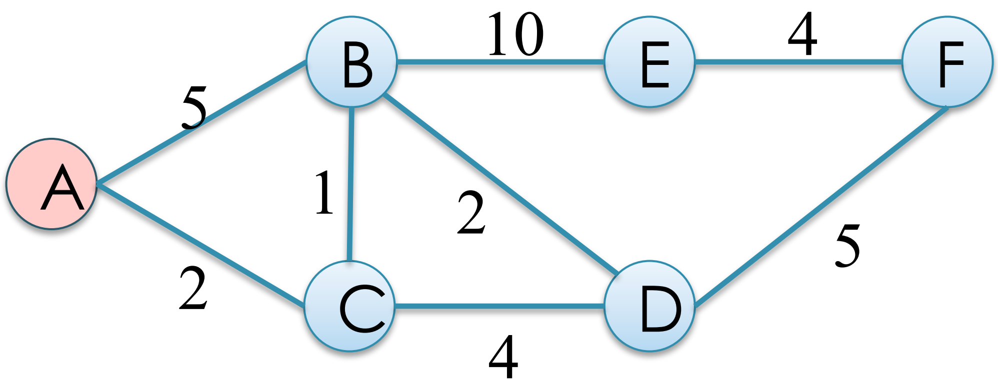
Previous Step

Dest	Path	Cost D(v)
A		
B		
C		
D		
E		
F		

This Step

Dest	Path	Cost D(v)
A	A	0
B	B	5
C	C	2
D	?	∞
E	?	∞
F	?	∞

Dijkstra's Algorithm – Step 1



Previous Step

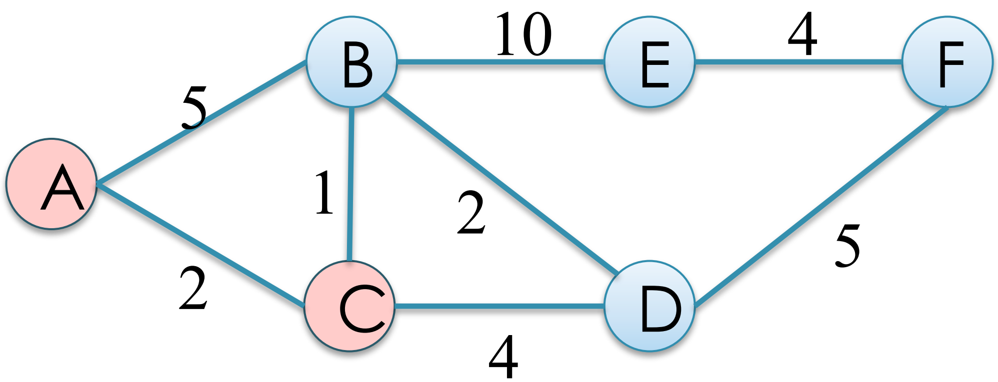
Dest	Path	Cost D(v)
A	A	0
B	B	5
C	C	2
D	?	∞
E	?	∞
F	?	∞

Pick Min

This Step

Dest	Path	Cost D(v)
A	A	0
B		
C		
D		
E		
F		

Dijkstra's Algorithm – Step 1



Can we find lower cost to any other node by going through C?

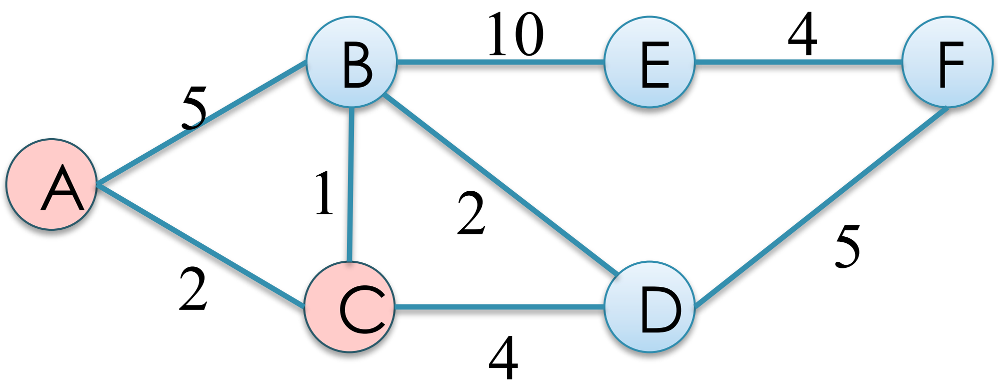
Previous Step

Dest	Path	Cost D(v)
A	A	0
B	B	5
C	C	2
D	?	∞
E	?	∞
F	?	∞

This Step

Dest	Path	Cost D(v)
A	A	0
B		
C	C	2
D		
E		
F		

Dijkstra's Algorithm – Step 1



Consider path to B:

D(B)
or
D(C) + cost(C, B)

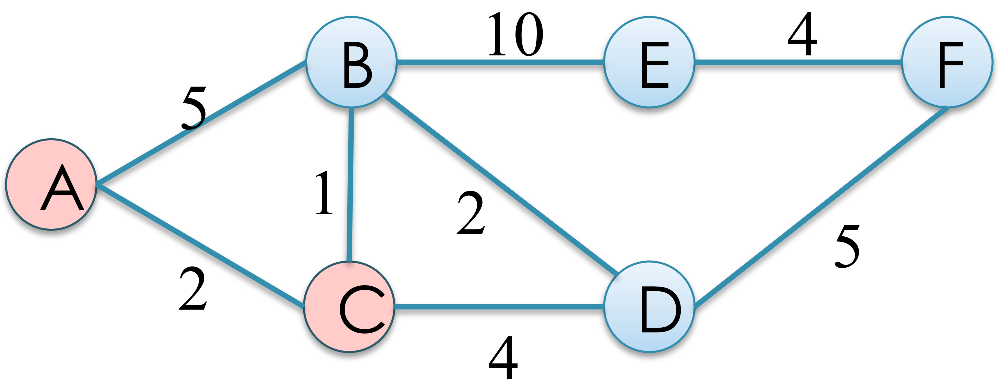
Previous Step

Dest	Path	Cost D(v)
✓ A	A	0
B	B	5
C	C	2
D	?	∞
E	?	∞
F	?	∞

This Step

Dest	Path	Cost D(v)
✓ A	A	0
B		
✓ C	C	2
D		
E		
F		

Dijkstra's Algorithm – Step 1



Consider path to B:

D(B) = 5
or
D(C) + cost(C, B)
2 + 1 = 3

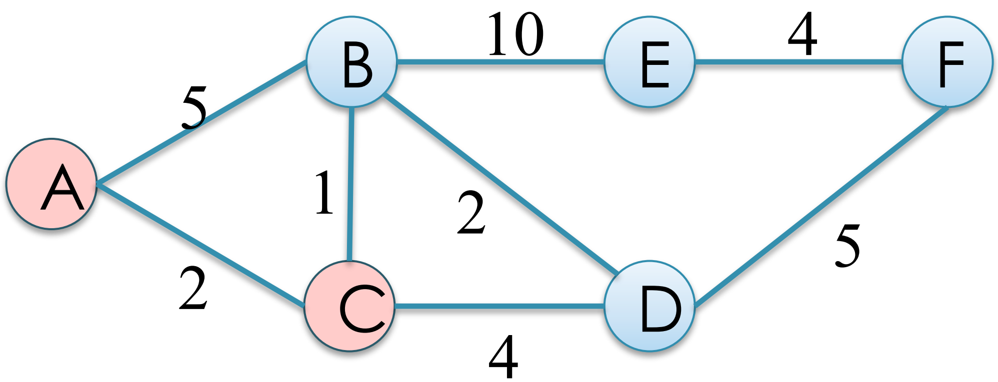
Previous Step

Dest	Path	Cost D(v)
A	A	0
B	B	5
C	C	2
D	?	∞
E	?	∞
F	?	∞

This Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D		
E		
F		

Dijkstra's Algorithm – Step 1



Consider path to D:

D(D) = ∞
or
D(C) + cost(C, D)
2 + 4 = 6

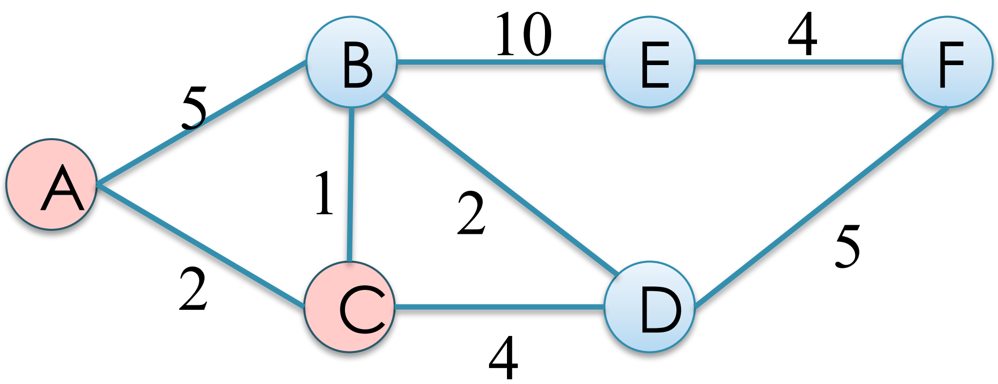
Previous Step

Dest	Path	Cost D(v)
A	A	0
B	B	5
C	C	2
D	?	∞
E	?	∞
F	?	∞

This Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, D	6
E		
F		

Dijkstra's Algorithm – Step 1



Still no information about E or F.

Previous Step

Dest	Path	Cost D(v)
A	A	0
B	B	5
C	C	2
D	?	∞
E	?	∞
F	?	∞

This Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, D	6
E	?	∞
F	?	∞