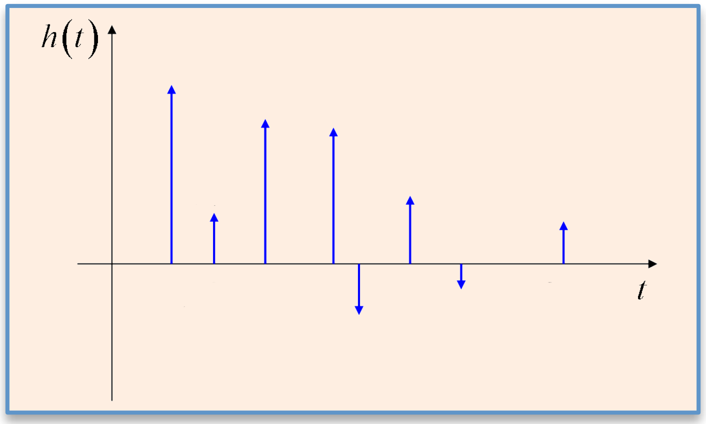
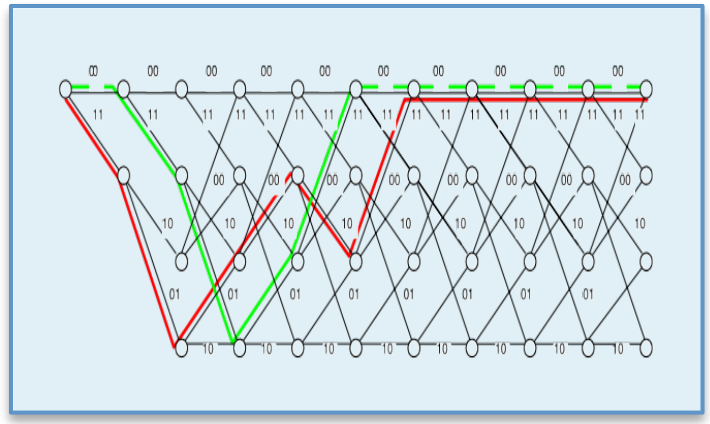
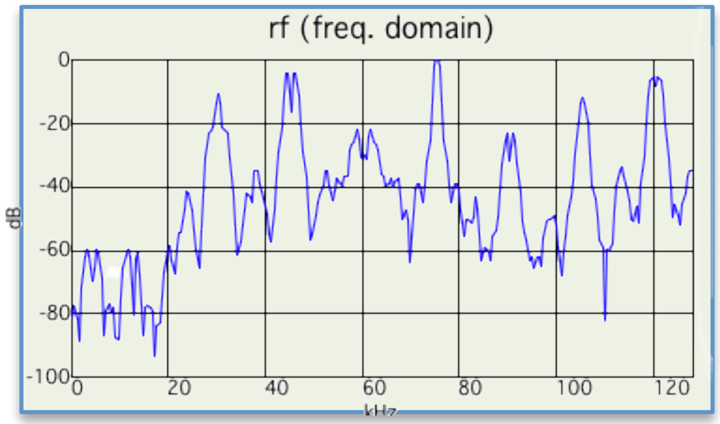
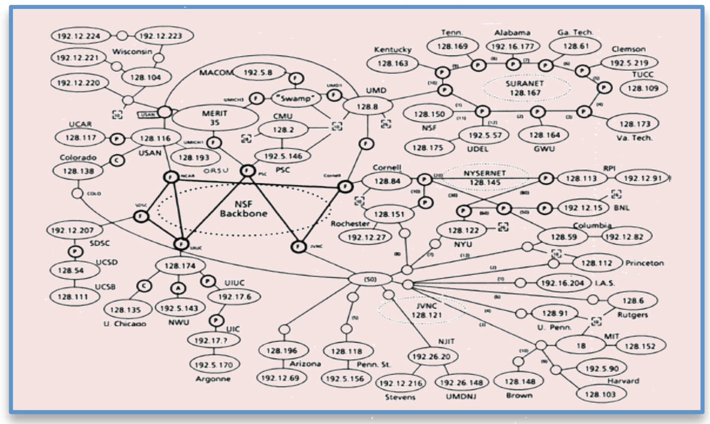








INTRODUCTION TO EECS II

DIGITAL COMMUNICATION SYSTEMS

6.02 Fall 2014
Lecture #21

- Failures in routing
  - Routing loops
  - Counting to infinity

Unanswered questions

(about packet-switched networks)

~~How do nodes **determine routes** to every other node?~~

Nodes determine routes via either **link-state** or **distance-vector** routing

How do nodes **route around link failures**?

How do nodes **communicate reliably** given that the network is best-effort?

Comparison of Routing Protocols

(no failures)

	Distance-vector	**Link-state**
Node X's advertisement format	list of all nodes X knows about and the costs to those nodes	list of all X's neighbors and the link costs to those nodes
Who receives X's advertisement	X's neighbors	all nodes (via flooding)
Integration	Bellman-Ford	Dijkstra's Algorithm
Convergence time	Proportional to the number of hops in the longest min-cost path	Proportional to flooding time + complexity of Dijkstra's

Comparison of Routing Protocols

(no failures)

	Distance-vector		Link-state
Amount of data consumed by advertisements	O(L)	**better for large networks?**	$O(L^2)$
Integration	Bellman-Ford		Dijkstra's Algorithm
Convergence time	Proportional to the number of hops in the longest min-cost path		Proportional to flooding time + complexity of Dijkstra's

Convergence

route validity: if node N's routing table contains D, then there is a usable path in the network from N to D, and the routing table reflects a usable path

path visibility: every router that has a usable path to a destination learns at least one valid route to that destination

Eventual Convergence

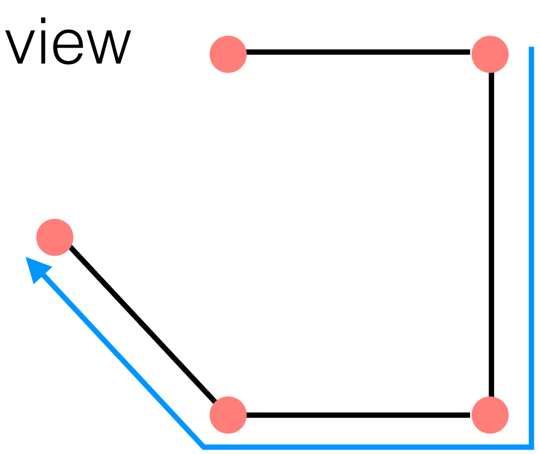
Given:
- initial state at time 0
- time t after which no changes occur to the topology and no routing advertisements or HELLO packets are lost

If the routing protocol converges in some finite amount of time after t, we say the routing protocol has **eventually converged**.

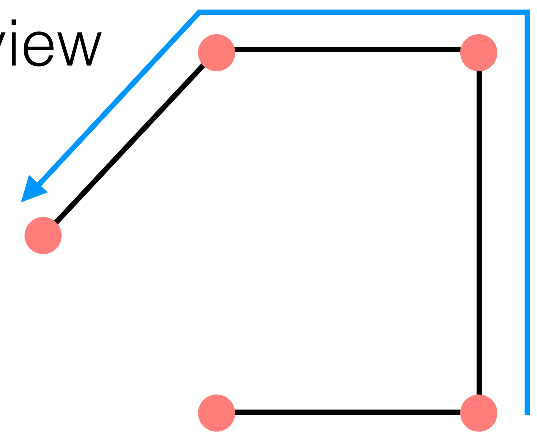
goal: understand how link-state and distance-vector perform when links fail so that we can decide when to use which protocol

Routing Loops

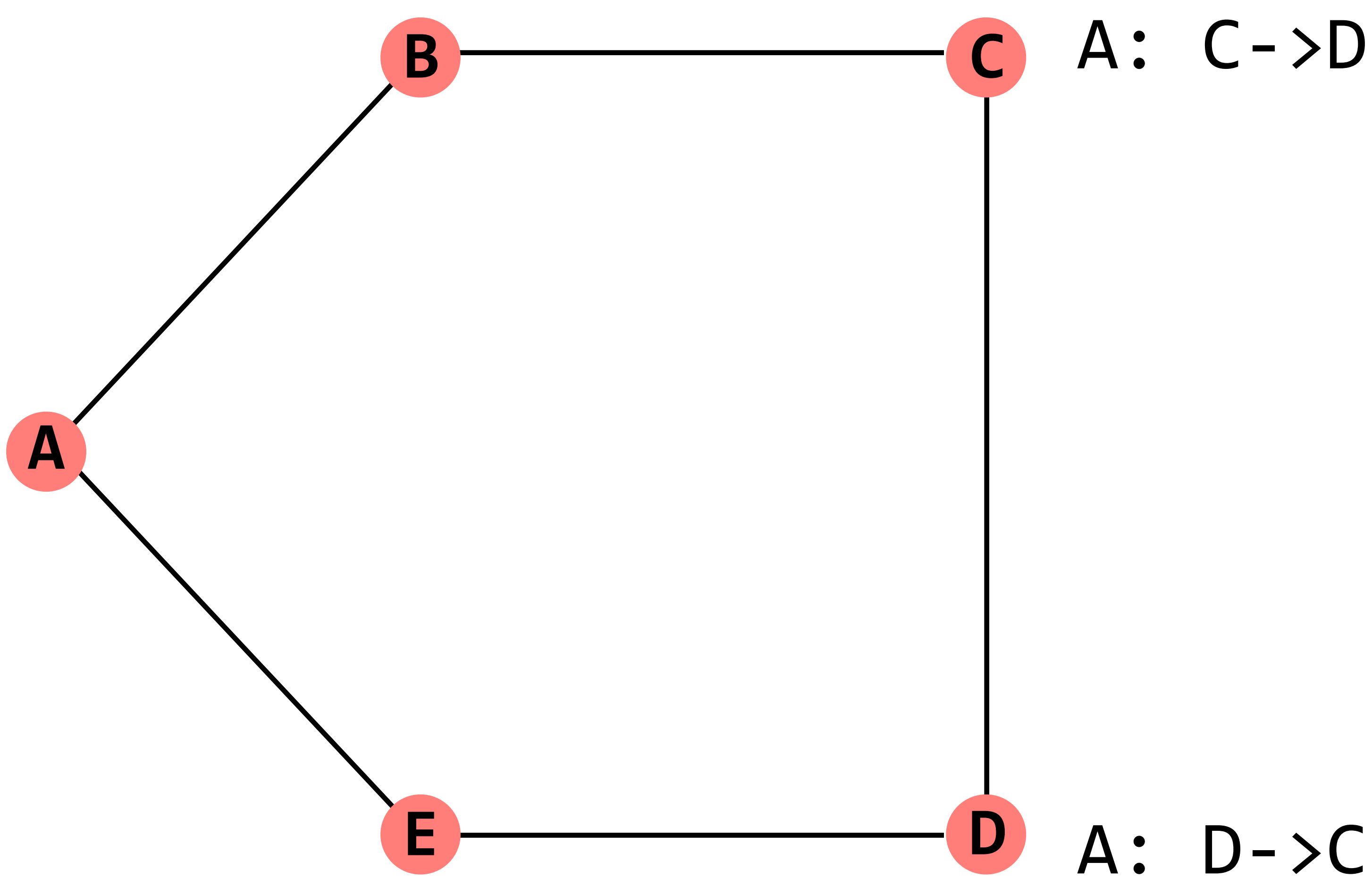
C’s view



D’s view

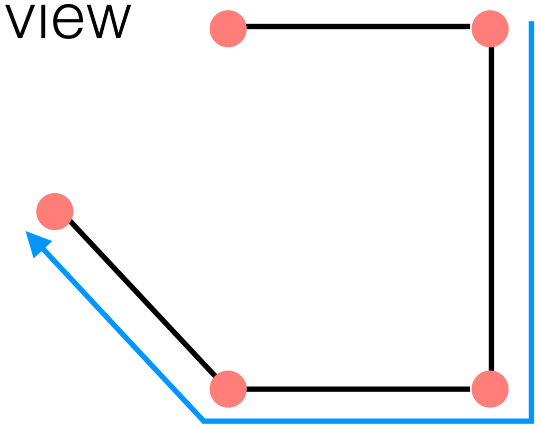


lost advertisements cause incorrect network views and thus incorrect routes

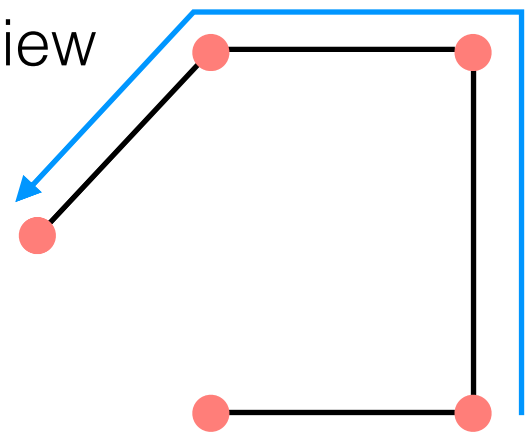


Routing Loops

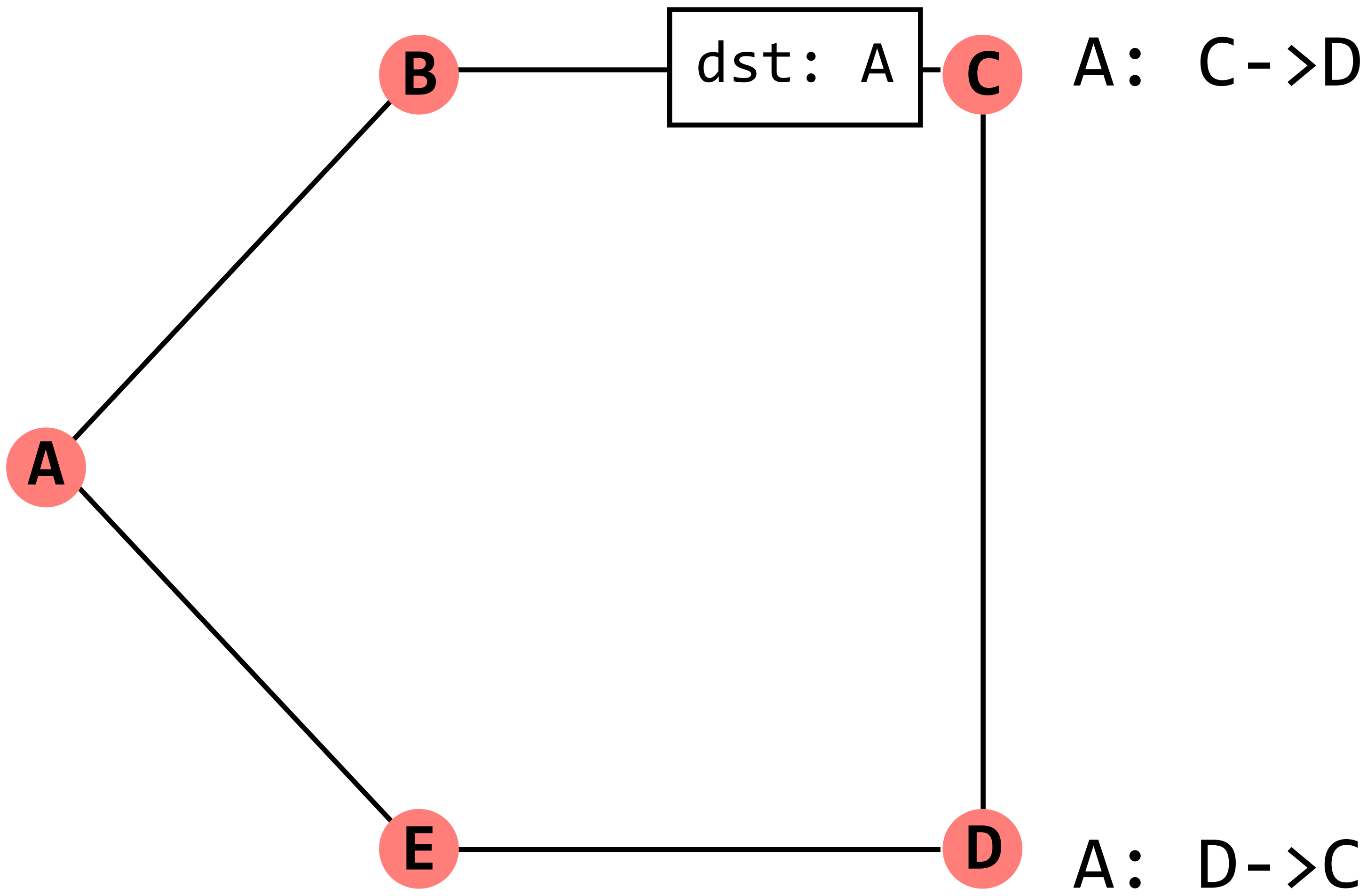
C's view



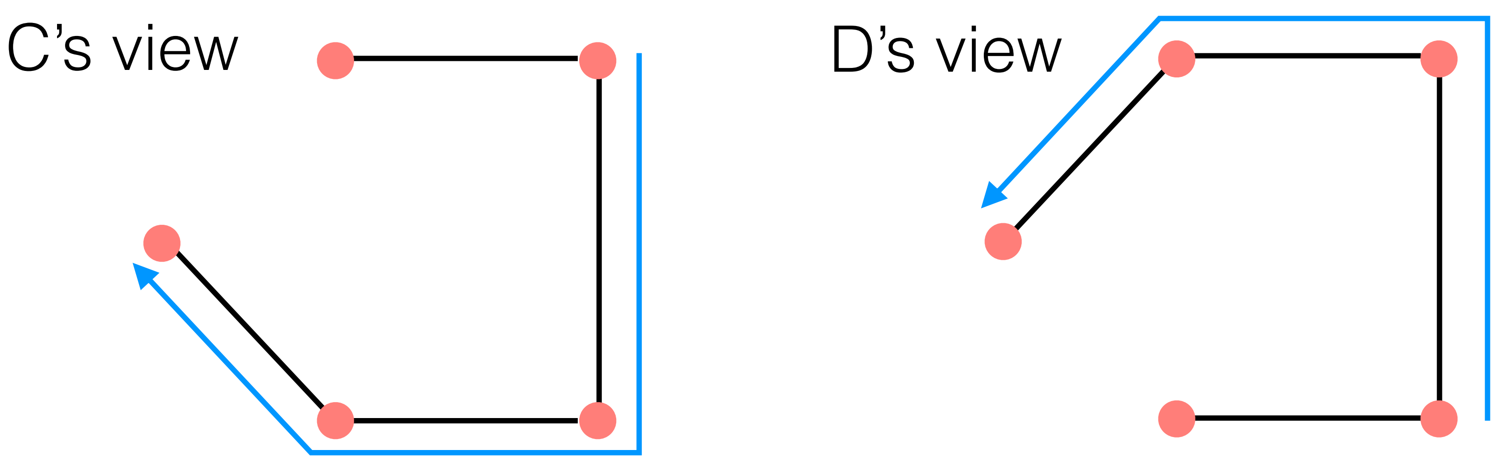
D's view



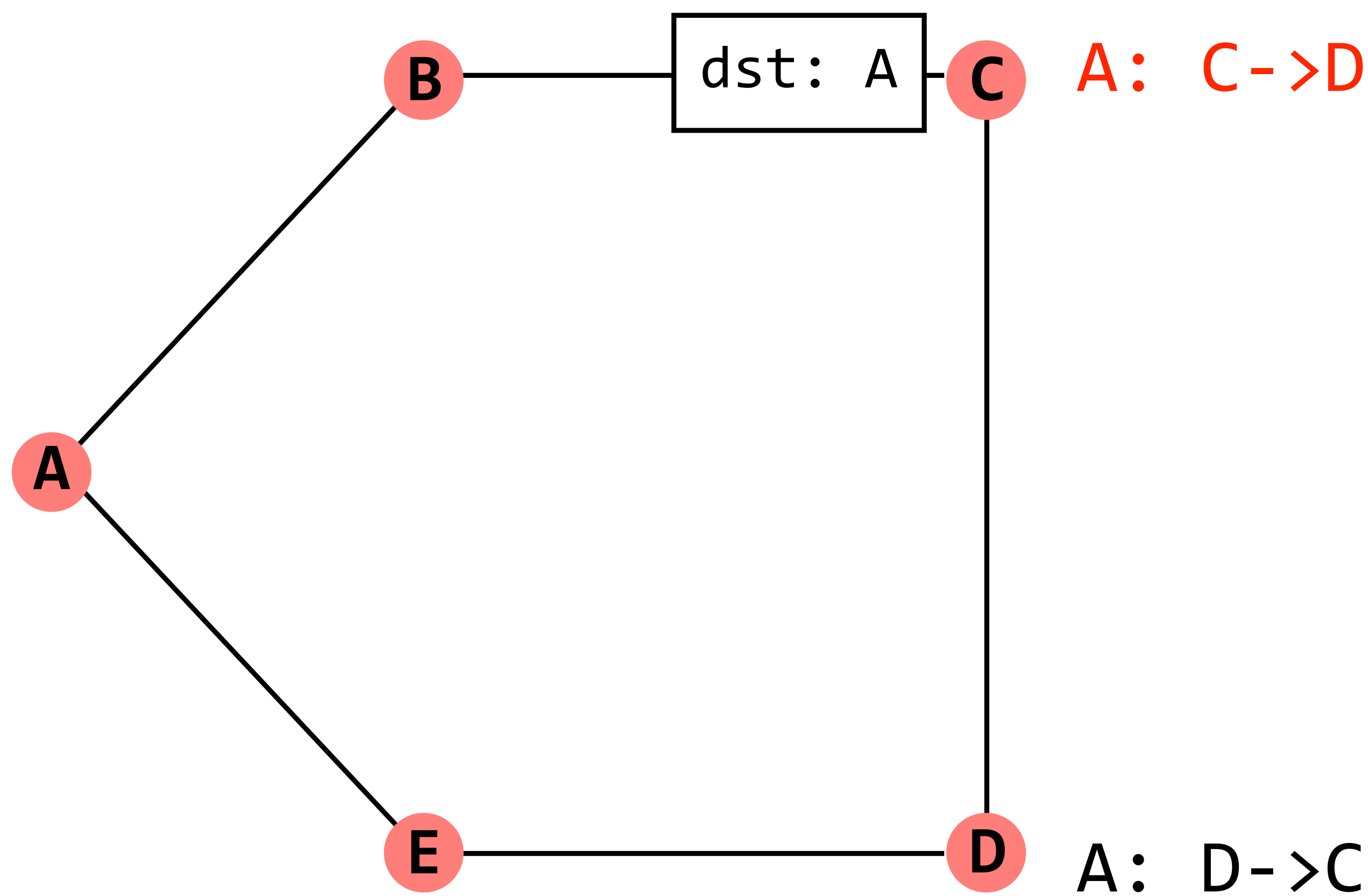
lost advertisements cause incorrect network views and thus incorrect routes



Routing Loops

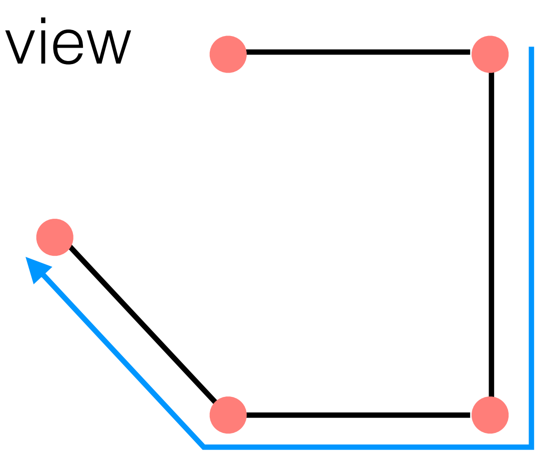


lost advertisements cause incorrect network views and thus incorrect routes

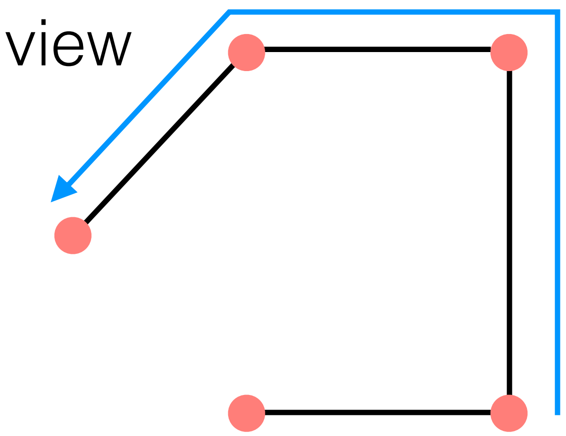


Routing Loops

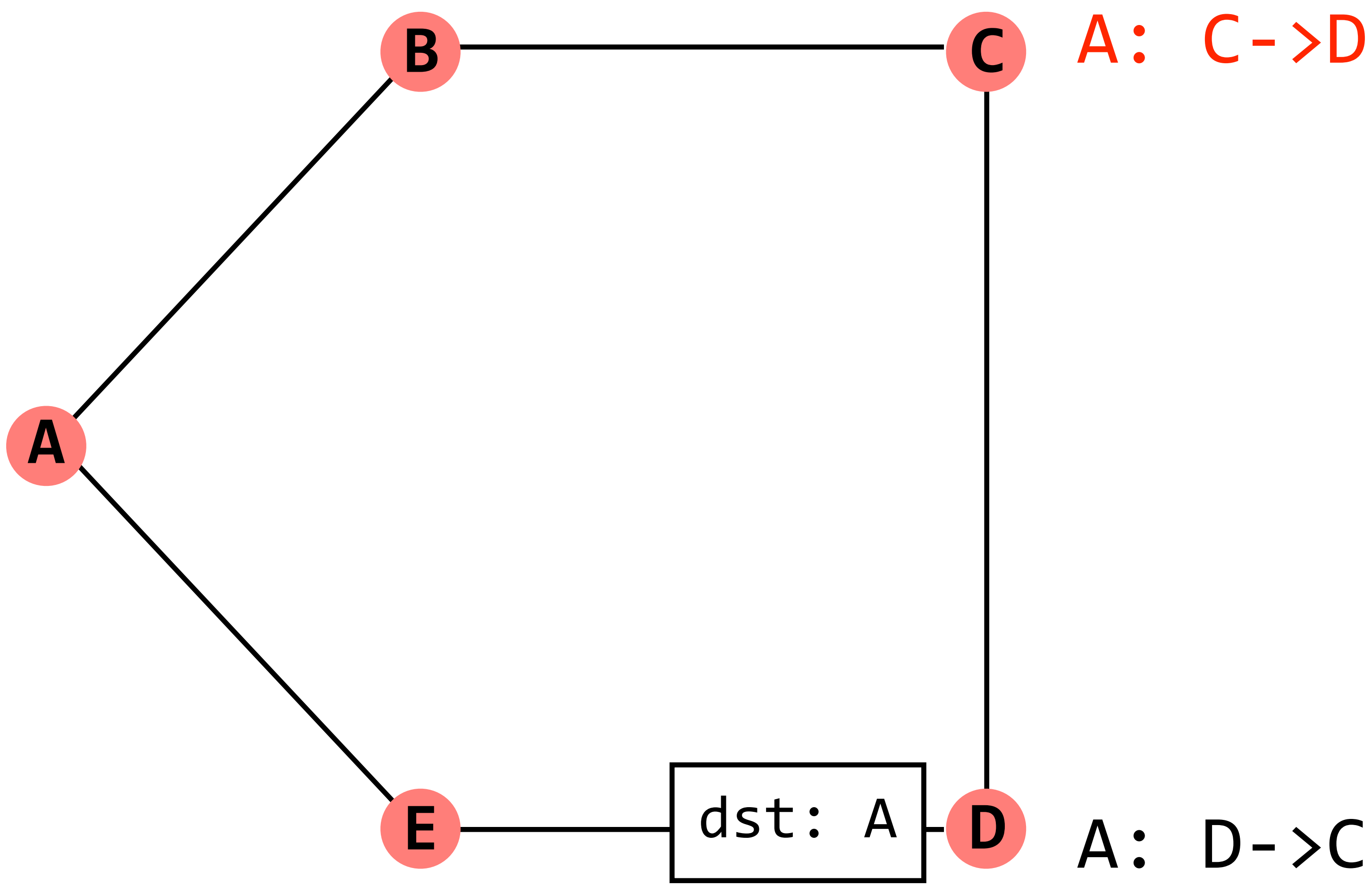
C’s view



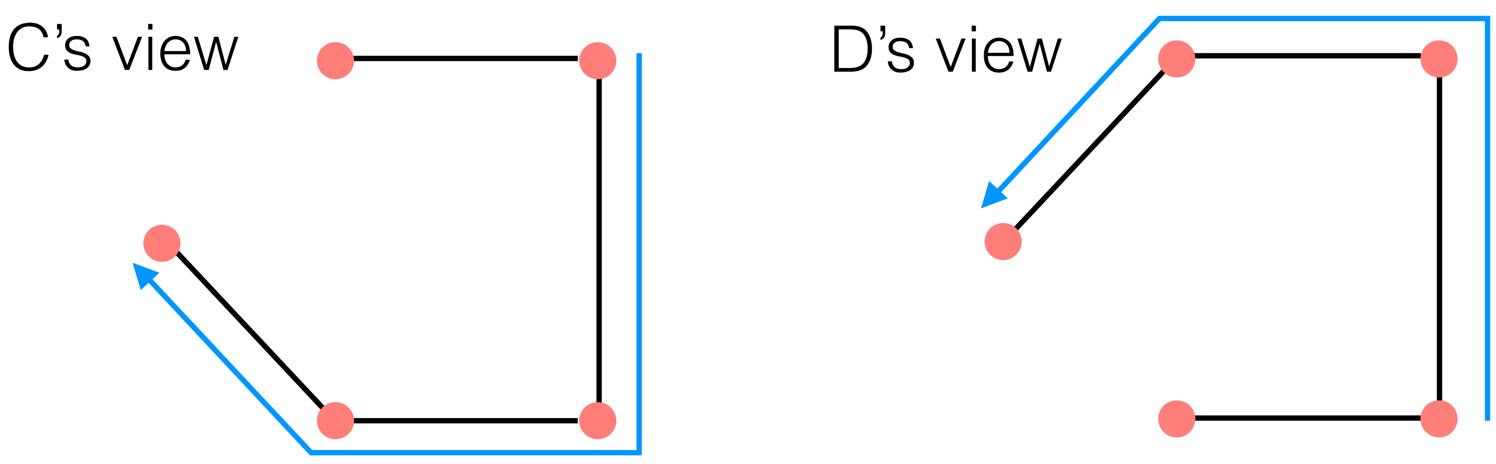
D’s view



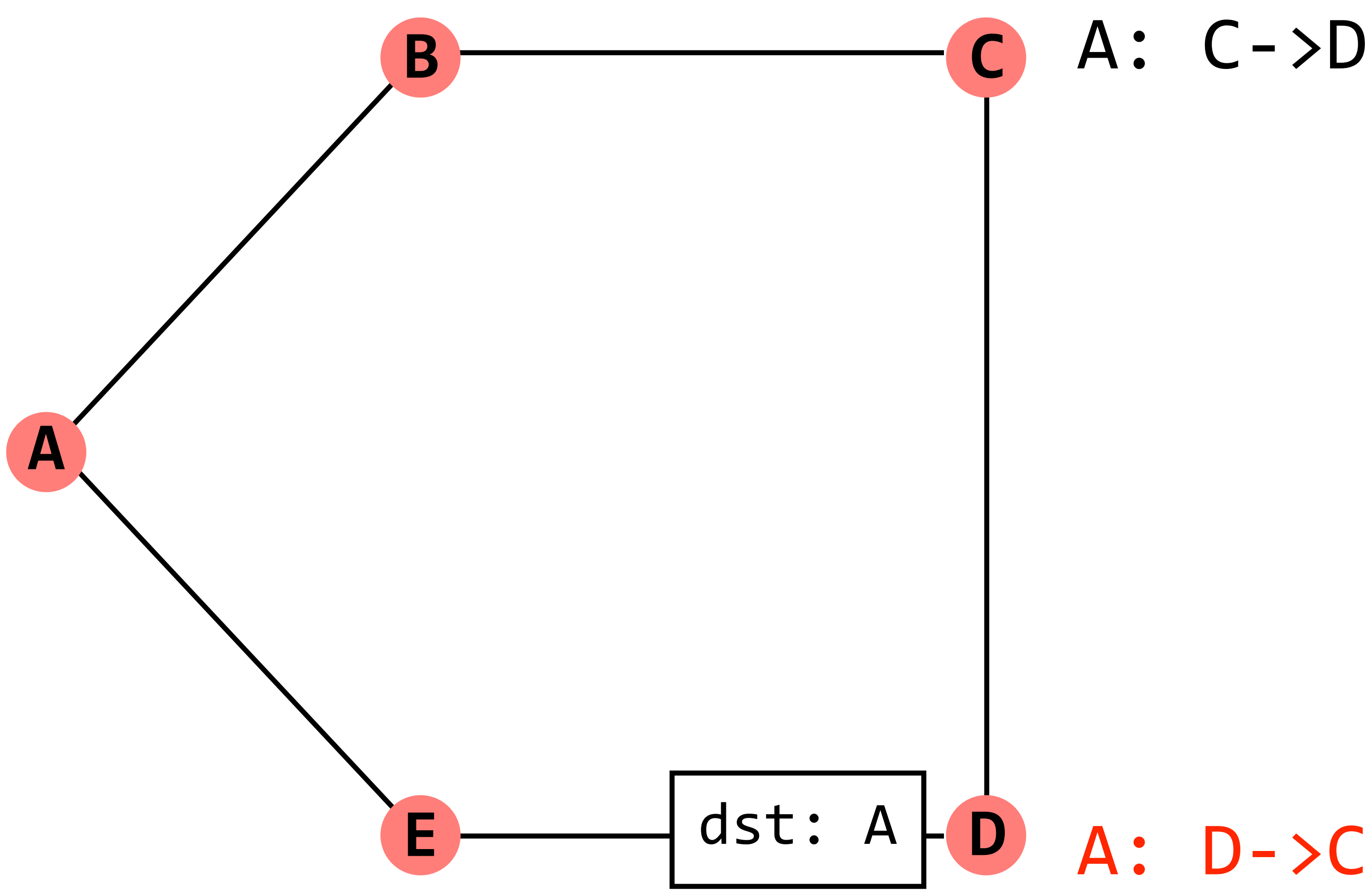
lost advertisements cause incorrect network views and thus incorrect routes



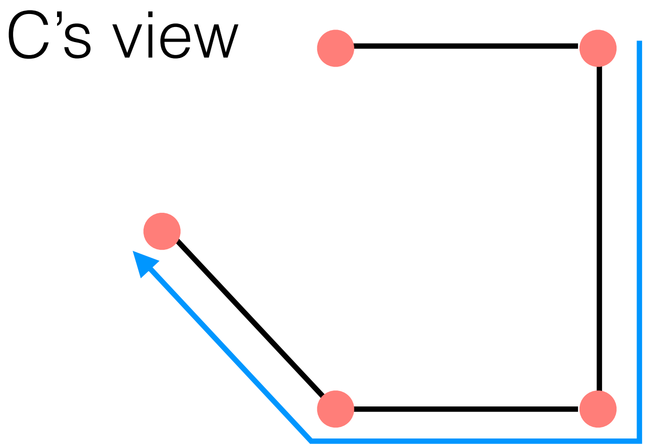
Routing Loops

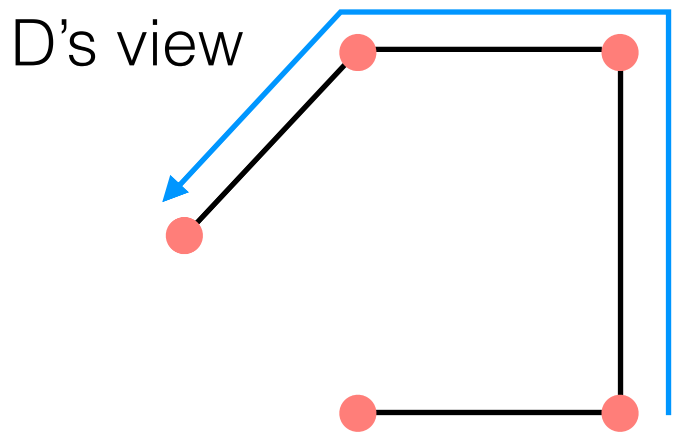


lost advertisements cause incorrect network views and thus incorrect routes

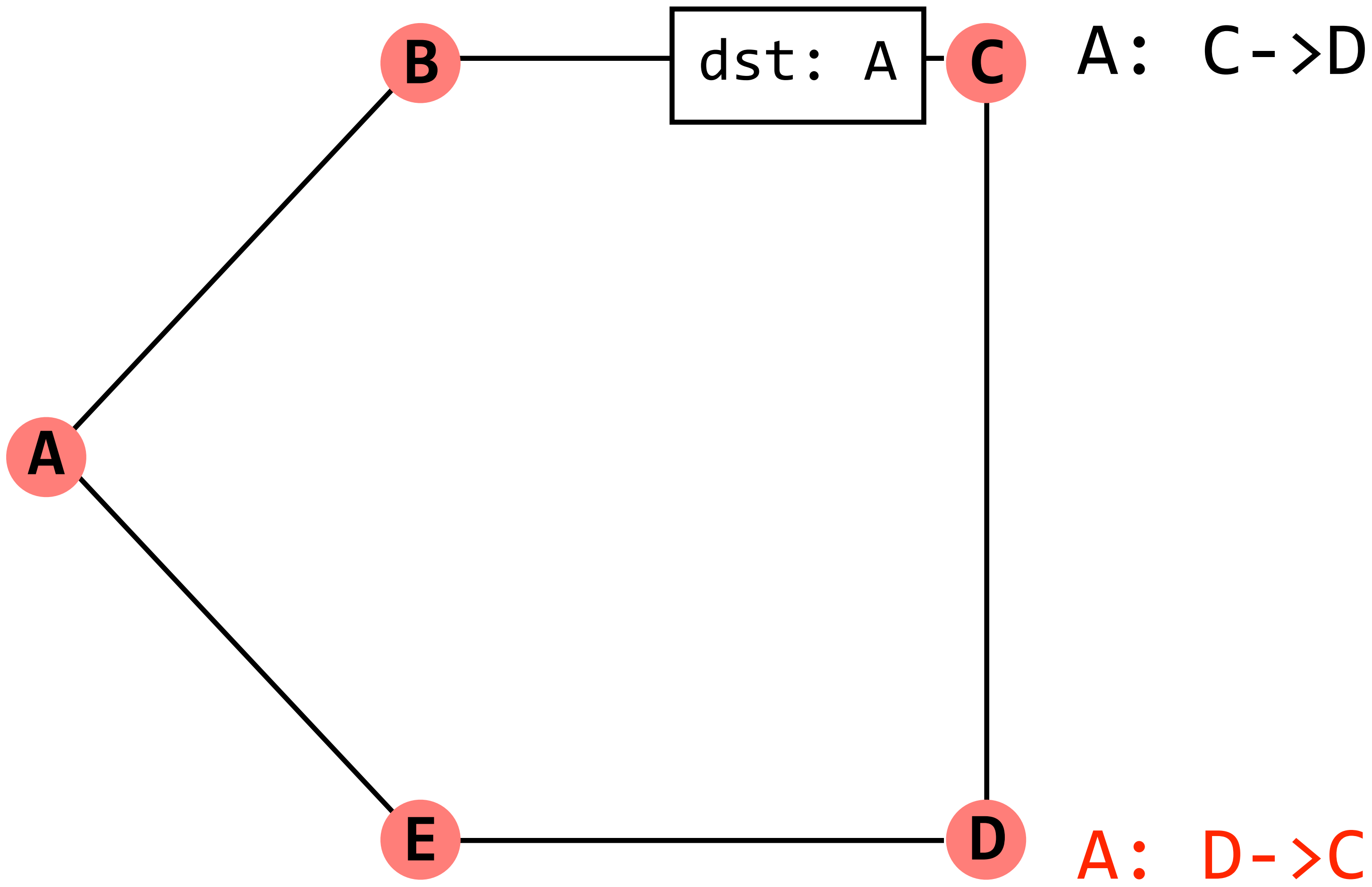


Routing Loops





lost advertisements cause incorrect network views and thus incorrect routes



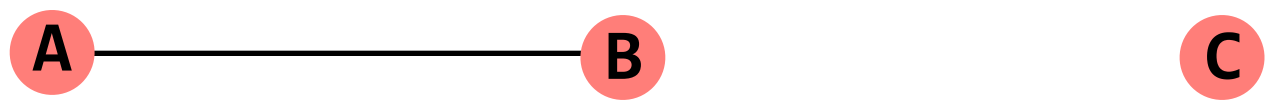
Comparison of Routing Protocols

(failures)

	Distance-vector	**Link-state**
Amount of data consumed by advertisements	Small (O(L)) **better for large networks?**	Large (O(L^2))
Convergence		Generally fast, routing loops are rare

INFINITY

A sends advertisements at t=0, 10, 20,..; B sends advertisements at t=5, 15, 25,..




```
A: Self, 0     A: B->A, 1
B: A->B, 1     B: Self, 0
C: A->B, 2     C: None, inf
```

t=9: B<->C fails

```
A: Self, 0     A: B->A, 1
B: A->B, 1     B: Self, 0
C: A->B, 2     C: B->A, 3    (2+1)
```

t=10: B receives the following advertisement from A:
[(A,0),(B,1),(C,2)]

```
A: Self, 0     A: B->A, 1
B: A->B, 1     B: Self, 0
C: A->B, 4     C: B->A, 3
```

t=15: A receives the following advertisement from B:
[(A,0),(B,1),(C,3)]

```
A: Self, 0     A: B->A, 1
B: A->B, 1     B: Self, 0
C: A->B, 4     C: B->A, 5
```

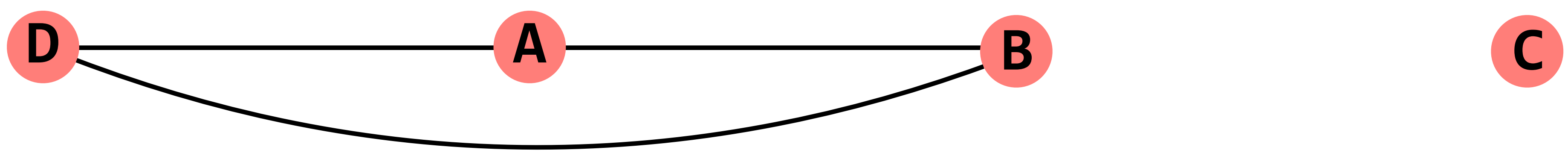
t=20: B receives the following advertisement from B:
[(A,0),(B,1),(C,4)]

continues until both costs to C are INFINITY

problem: distance-vector protocols can count to infinity, which increases the convergence time. can we solve the count-to-infinity problem?

Split-horizon

Don’t send advertisements about a route to the node providing the route



C: D->B, 2	C: A->B, 2	C: None, inf	B<->C fails
C: None, inf	C: A->B, 2	C: None, inf	B’s advertisement to A gets lost (so A makes no changes)
C: D->A, 3	C: A->B, 2	C: None, inf	A advertises about C to D (not to B because of split horizon)
C: D->A, 3	C: A->B, 2	C: B->D, 4	D advertises about C to B
C: D->A, 3	C: A->B, 5	C: B->D, 4	B advertises about C to A

continues until all costs to C are INFINITY

Comparison of Routing Protocols

(failures)

	Distance-vector	**Link-state**
Amount of data consumed by advertisements	Small (O(L)) **better for large networks?**	Large ($O(L^2)$)
Convergence	Can depend on value of INFINITY; the larger INFINITY is, the slower convergence is **so.. not good for large networks?**	Generally fast, routing loops are rare

Comparison of Routing Protocols

(failures)

	Distance-vector	Link-state	Path-vector
Amount of data consumed by advertisements	Small (O(L))	Large ($O(L^2)$)	Fairly small in practice (but larger than DV)
Convergence	Can depend on value of INFINITY	Generally fast, routing loops are rare	Not as fast as LS, but does not depend on INFINITY
	good for very small networks where we can make guarantees about (a lack of) routing loops	**good for small (university-sized) networks where the overhead of the advertisements doesn't overwhelm**	**good for large networks (the Internet!)**

- **Distance-vector Routing**
  Low overhead, but slow convergence (count-to-infinity)

- **Link-state Routing**
  High overhead, but faster convergence (routing loops can happen, but are rare)

- **Path-vector Routing**
  An improvement on distance-vector routing that avoids counting to infinity

- **Which protocol to use depends on the environment, particularly on the size of the network**