DSDV

Destination-Sequenced Distance-Vector Routing Protocol

Outline

- Introduction
  - Distance-Vector
- DSDV Protocol
- Summary

Introduction

- The property of ad-hoc networks
  - Topology may be quite dynamic
  - No administrative host
  - Hosts with finite power

Introduction

- The properties of the ad-hoc network routing protocol
  - Simple
  - Less storage space
  - Loop free
  - Short control message (Low overhead)
  - Less power consumption
  - Multiple disjoint routes
  - Fast rerouting mechanism

Introduction

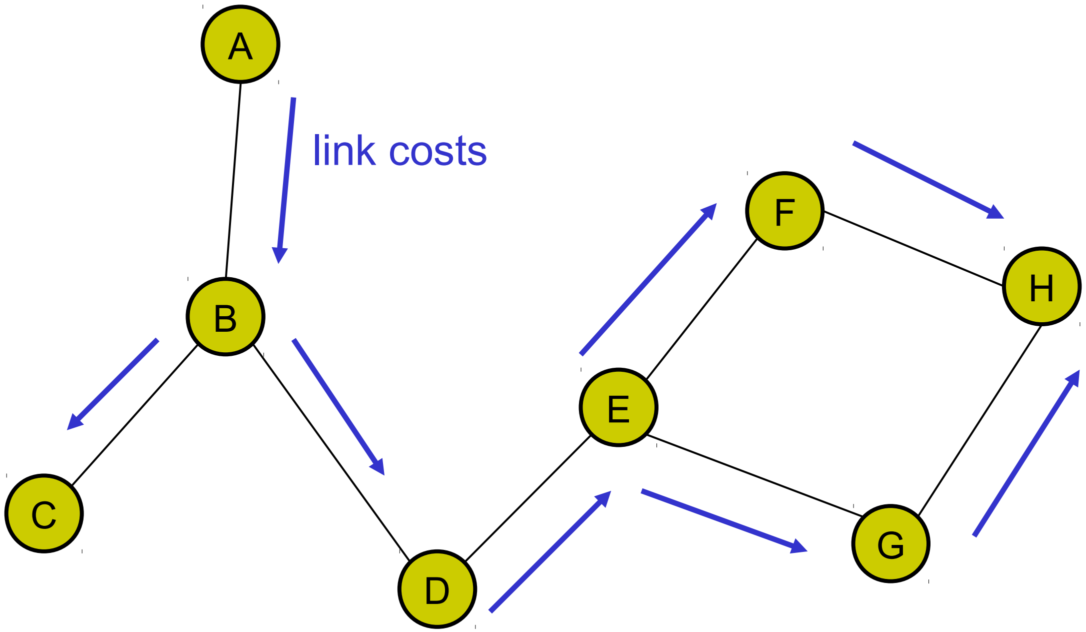
- Routing Protocol:
  - Table-driven (proactive)
  - Source-initiated on-demand (reactive)
  - Hybrid

- Routing Algorithm
  - Link-State algorithm:
    - Each node maintains a view of the network topology
  - Distance-Vector algorithm:
    - Every node maintains the distance of each destination

Link-State

- Like the shortest-path computation method
- Each node maintains a view of the network topology with a cost for each link
- Periodically broadcast link costs to its outgoing links to all other nodes such as flooding

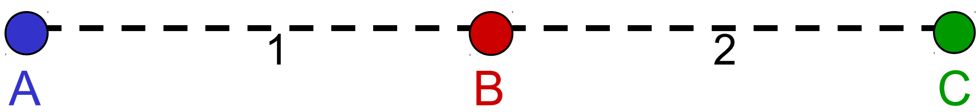
Link-State



Distance-Vector

- known also as Distributed Bellman-Ford or RIP (Routing Information Protocol)
- Every node maintains a routing table
  - all available destinations
  - the next node to reach to destination
  - the number of hops to reach the destination
- Periodically send table to all neighbors to maintain topology

Distance Vector (Tables)



A

Dest.	Next	Metric	…
A	A	0	
B	B	1	
C	B	3	

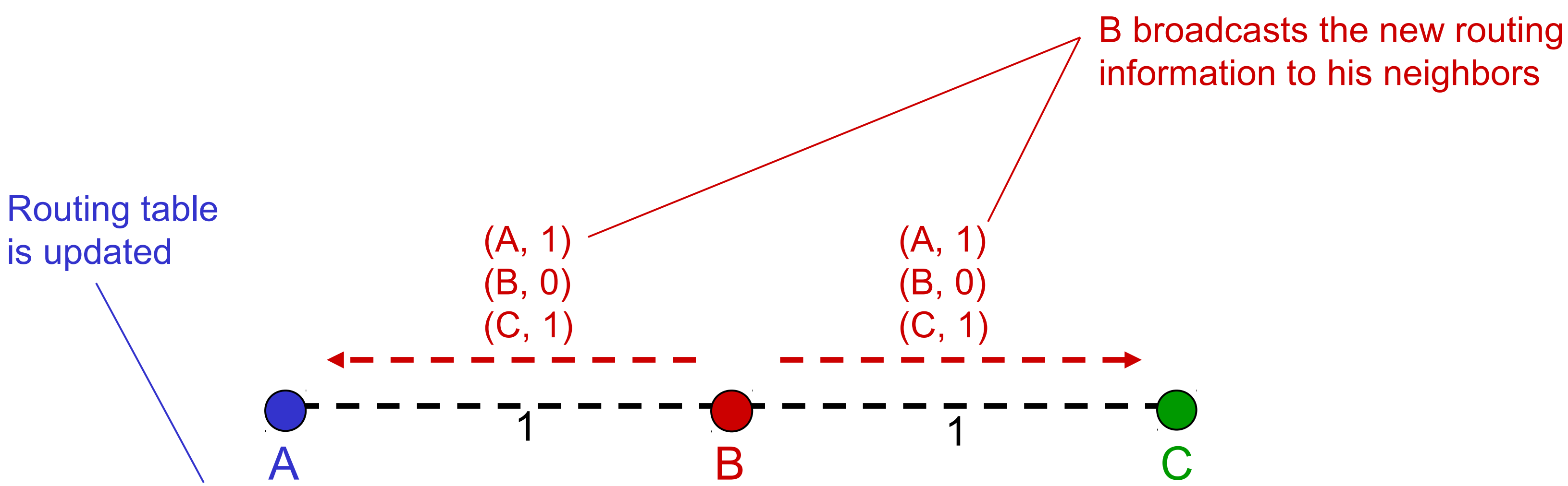
B

Dest.	Next	Metric	…
A	A	1	
B	B	0	
C	C	2	

C

Dest.	Next	Metric	…
A	B	3	
B	B	2	
C	C	0	

Distance Vector (Update)



A

Dest.	Next	Metric	…
A	A	0	
B	B	1	
C	B	~~3~~ 2	

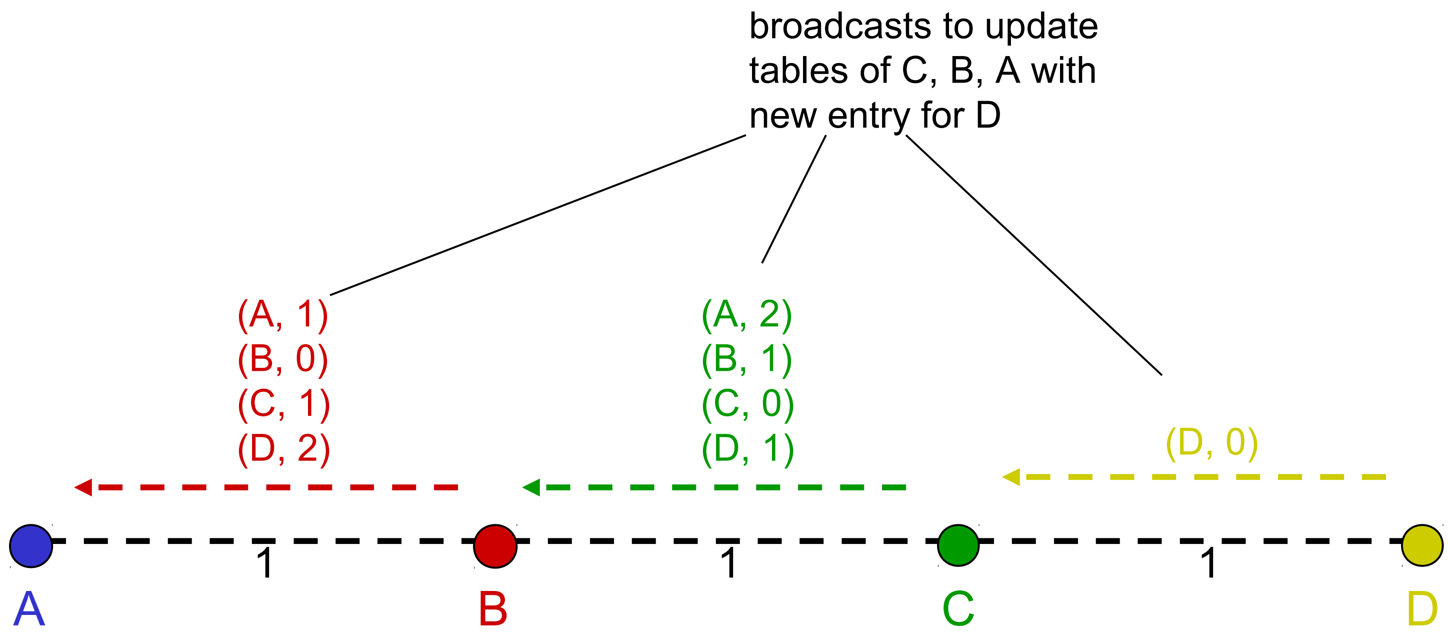
B

Dest.	Next	Metric	…
A	A	1	
B	B	0	
C	C	1	

C

Dest.	Next	Metric	…
A	B	~~3~~ 2	
B	B	1	
C	C	0	

Distance Vector (New Node)

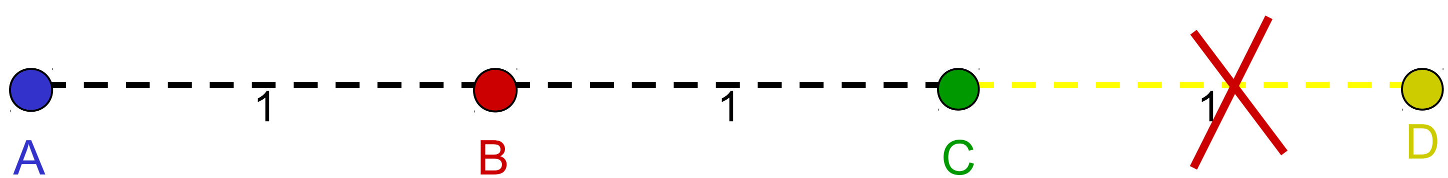


Dest.	Next	Metric	…
A	A	0	
B	B	1	
C	B	2	
D	B	3	

Dest.	Next	Metric	…
A	A	1	
B	B	0	
C	C	1	
D	C	2	

Dest.	Next	Metric	…
A	B	2	
B	B	1	
C	C	0	
D	D	1	

Distance Vector (Broken Link)

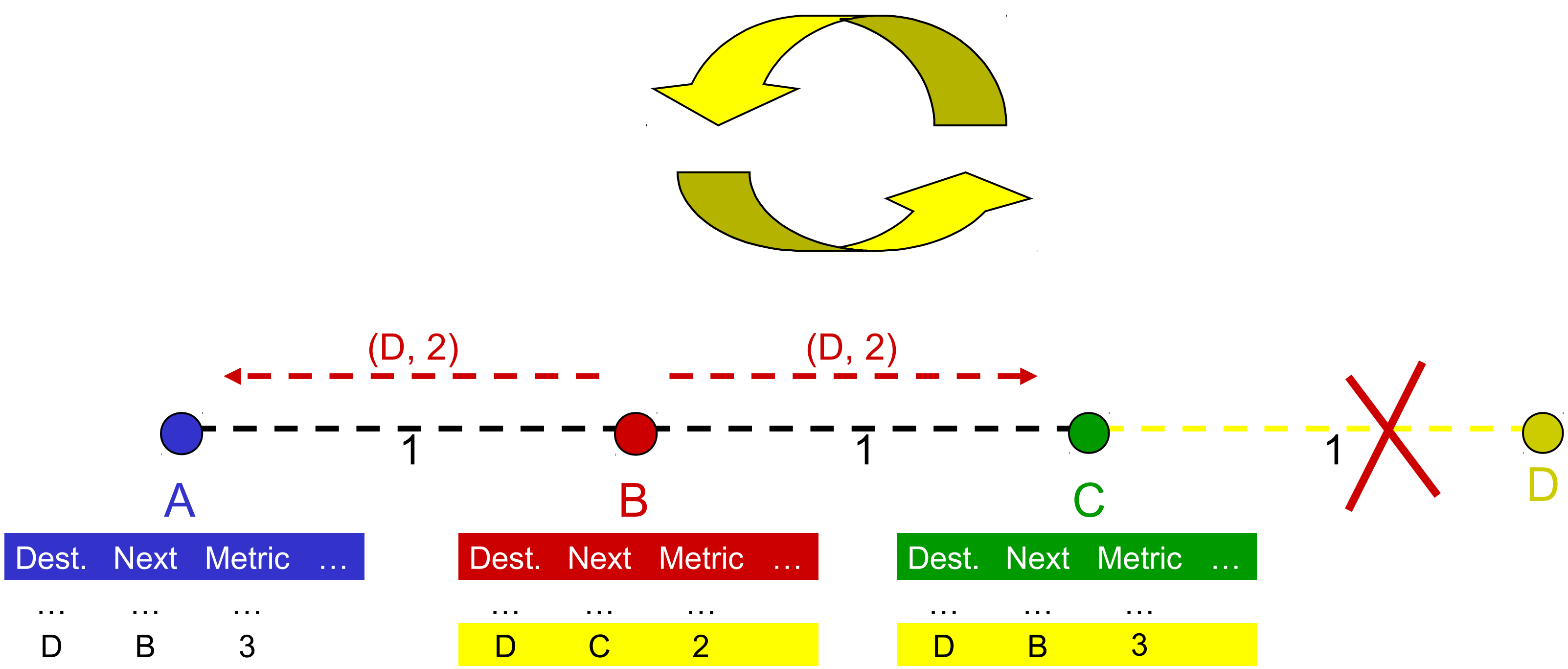


Dest.	Next	Metric	…
…	…	…	
D	B	3	

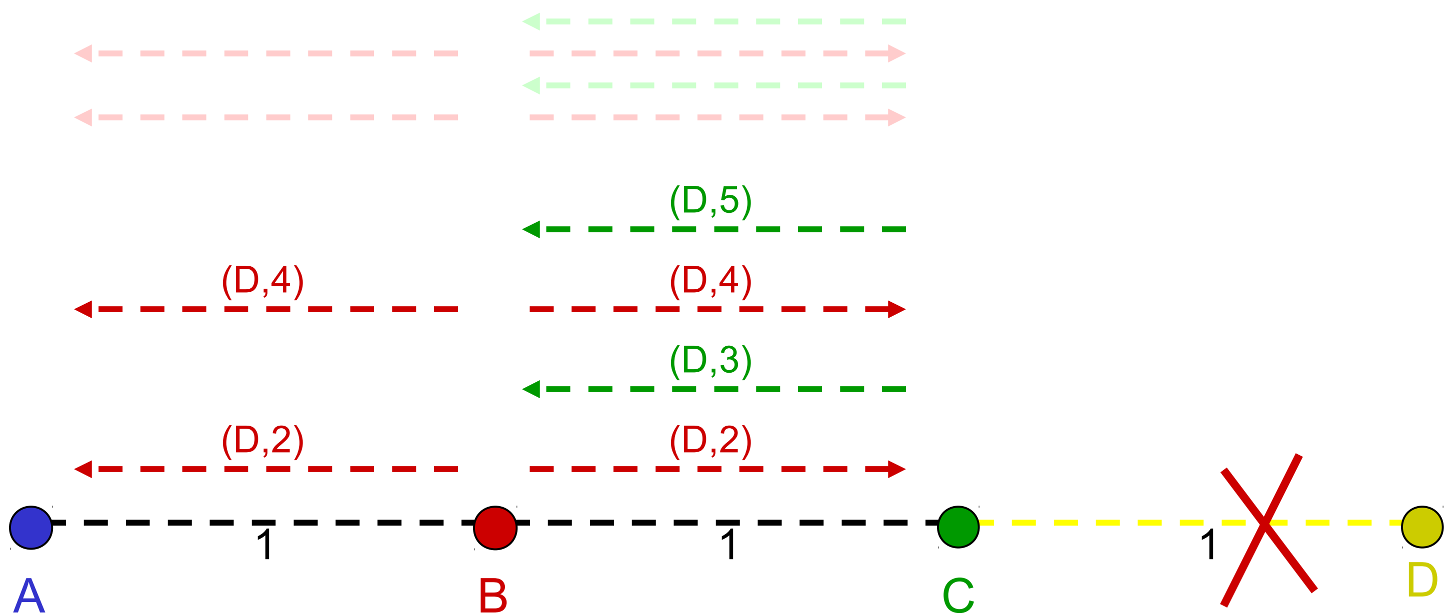
Dest.c	Next	Metric	…
…	…	…	
D	C	2	

Dest.	Next	Metric	…
…	…	…	
D	D	∞	

Distance Vector (Loops)



Distance Vector (Count to Infinity)



Dest.	Next	Metric	…
…	…	…	
D	B	3, 5, …	

Dest.c	Next	Metric	…
…	…	…	
D	C	2, 4, 6…	

Dest.	Next	Metric	…
…	…	…	
D	B	3, 5, …	

Distance Vector

- DV not suited for ad-hoc networks!
  - Loops
  - Count to Infinity

- New Solution -> DSDV Protocol

DSDV Protocol

- DSDV is Destination Based
- No global view of topology

DSDV Protocol

- DSDV is Proactive (Table Driven)
  - Each node maintains routing information for all known destinations
  - Routing information must be updated periodically
  - Traffic overhead even if there is no change in network topology
  - Maintains routes which are never used

DSDV Protocol

- Keep the simplicity of Distance Vector
- Guarantee Loop Freeness
  - New Table Entry for Destination Sequence Number
- Allow fast reaction to topology changes
  - Make immediate route advertisement on significant changes in routing table
  - but wait with advertising of unstable routes (damping fluctuations)

DSDV (Table Entries)

Destination	Next	Metric	Seq. Nr	Install Time	Stable Data
A	A	0	A-550	001000	Ptr_A
B	B	1	B-102	001200	Ptr_B
C	B	3	C-588	001200	Ptr_C
D	B	4	D-312	001200	Ptr_D

- **Sequence number** originated from destination. Ensures loop freeness.
- **Install Time** when entry was made (used to delete stale entries from table)
- **Stable Data** Pointer to a table holding information on how stable a route is. Used to damp fluctuations in network.

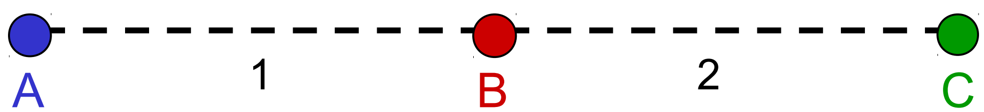
DSDV (Route Advertisements)

- Advertise to each neighbor own routing information
  - Destination Address
  - Metric = Number of Hops to Destination
  - Destination Sequence Number
- Rules to set sequence number information
  - On each advertisement increase own destination sequence number (use only even numbers)
  - If a node is no more reachable (timeout) increase sequence number of this node by 1 (odd sequence number) and set metric = ∞

DSDV (Route Selection)

- Update information is compared to own routing table
  - 1. Select route with higher destination sequence number (This ensure to use always newest information from destination)
  - 2. Select the route with better metric when sequence numbers are equal.

DSDV (Tables)

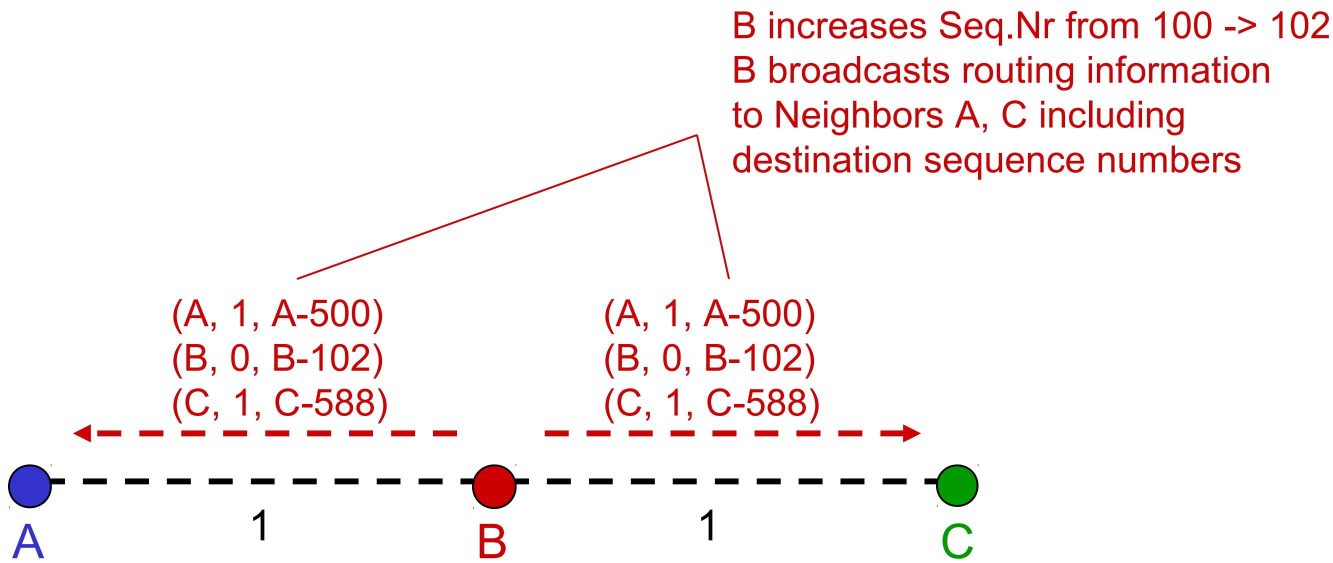


Dest.	Next	Metric	Seq
A	A	0	A-550
B	B	1	B-100
C	B	3	C-586

Dest.	Next	Metric	Seq
A	A	1	A-550
B	B	0	B-100
C	C	2	C-588

Dest.	Next	Metric	Seq.
A	B	1	A-550
B	B	2	B-100
C	C	0	C-588

DSDV (Route Advertisement)



Dest.	Next	Metric	Seq
A	A	0	A-550
B	B	1	B-102
C	B	2	C-588

Dest.	Next	Metric	Seq
A	A	1	A-550
B	B	0	B-102
C	C	1	C-588

Dest.	Next	Metric	Seq.
A	B	2	A-550
B	B	1	B-102
C	C	0	C-588

DSDV (Respond to Topology Changes)

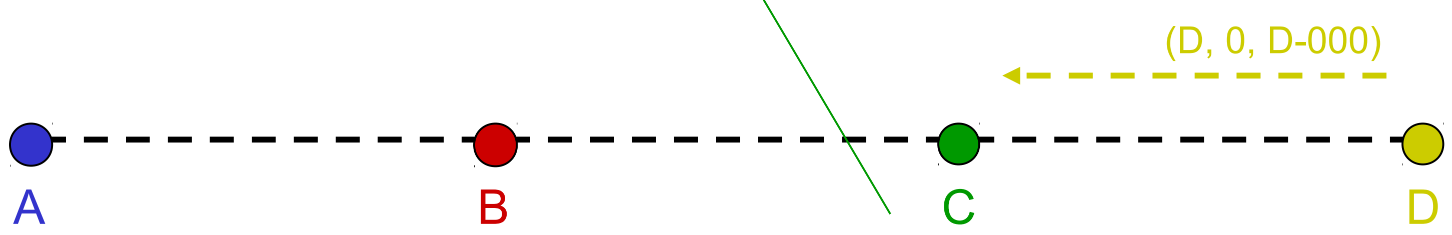
- Immediate advertisements
  - Information on new Routes, broken Links, metric change is immediately propagated to neighbors.

- Full/Incremental Update:
  - Full Update: Send all routing information from own table.
  - Incremental Update: Send only entries that has changed. (Make it fit into one single packet)

DSDV (New Node)

2. Insert entry for D with sequence number D-000
Then immediately broadcast own table

1. D broadcast for first time
Send Sequence number D-000



A

Dest.	Next	Metric	Seq.
A	A	0	A-550
B	B	1	B-104
C	B	2	C-590

B

Dest.	Next	Metric	Seq.
A	A	1	A-550
B	B	0	B-104
C	C	1	C-590

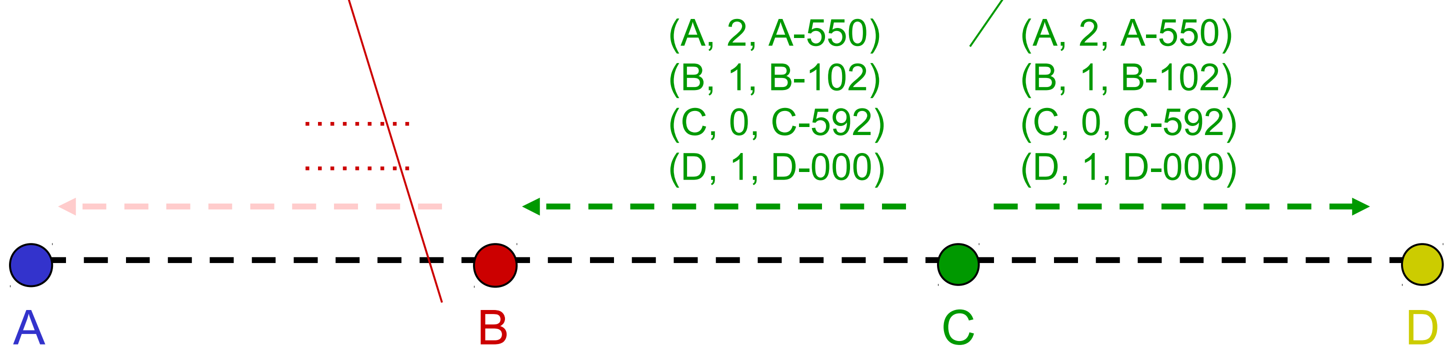
C

Dest.	Next	Metric	Seq.
A	B	2	A-550
B	B	1	B-104
C	C	0	C-590
D	D	1	D-000

DSDV (New Node cont.)

3. C increases its sequence number to C-592 then broadcasts its new table.

4. B gets this new information and updates its table…….



Dest.	Next	Metric	Seq.
A	A	0	A-550
B	B	1	B-104
C	B	2	C-590

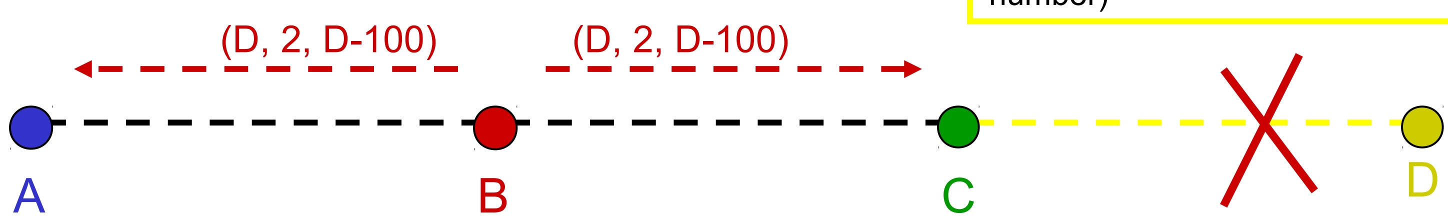
Dest.	Next	Metric	Seq.
A	A	1	A-550
B	B	0	B-102
C	C	1	C-592
D	C	2	D-000

Dest.	Next	Metric	Seq.
A	B	2	A-550
B	B	1	B-102
C	C	0	C-592
D	D	1	D-000

DSDV (no loops, no count to infinity)

2. B does its broadcast
-> no affect on C (C knows that B has stale information because C has higher seq. number for destination D)
-> no loop -> no count to infinity

1. Node C detects broken Link:
-> Increase Seq. Nr. by 1
(only case where not the destination sets the sequence number -> odd number)



Dest.	Next	Metric	Seq.
…	…	…	
D	B	3	D-100

Dest.c	Next	Metric	Seq.
…	…	…	
D	C	2	D-100

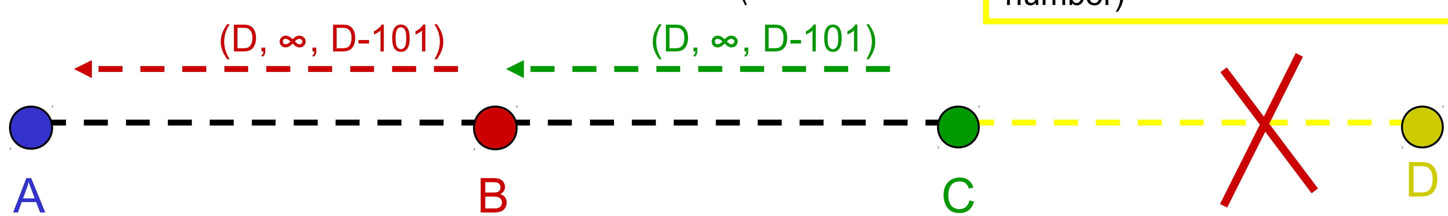
Dest.	Next	Metric	Seq.
…	…	…	
D	D	∞	D-101

DSDV (Immediate Advertisement)

1. Node C detects broken Link:
-> Increase Seq. Nr. by 1
(only case where not the destination sets the sequence number -> odd number)

2. Immediate propagation
C to B:
(update information has higher Seq. Nr. -> replace table entry)

3. Immediate propagation
B to A:
(update information has higher Seq. Nr. -> replace table entry)



A

Dest.	Next	Metric	Seq.
…	…	…	...
~~D~~	~~B~~	~~3~~	~~D-100~~
D	B	∞	D-101

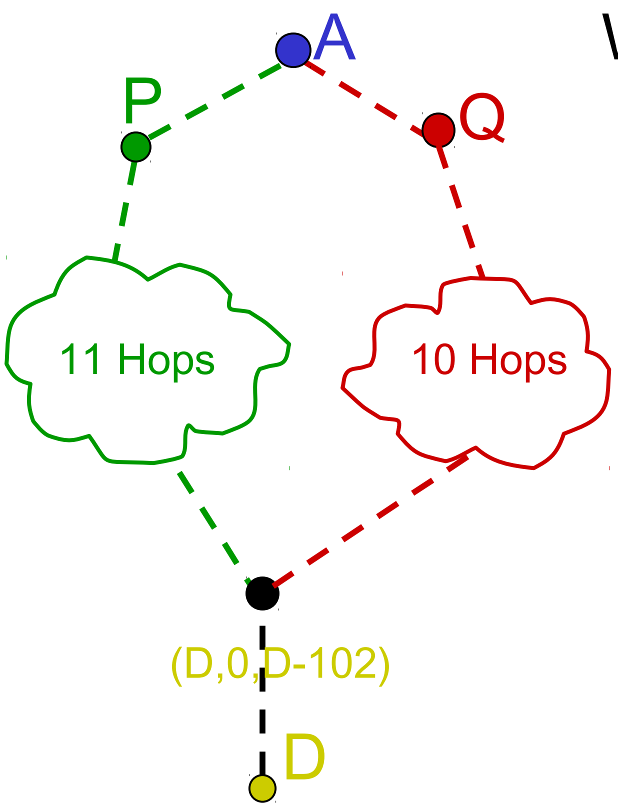
B

Dest.c	Next	Metric	Seq.
…	…	…	...
~~D~~	~~C~~	~~2~~	~~D-100~~
D	C	∞	D-101

C

Dest.	Next	Metric	Seq.
…	…	…	
~~D~~	~~D~~	~~1~~	~~D-100~~
D	D	∞	D-101

DSDV (Problem of Fluctuations)

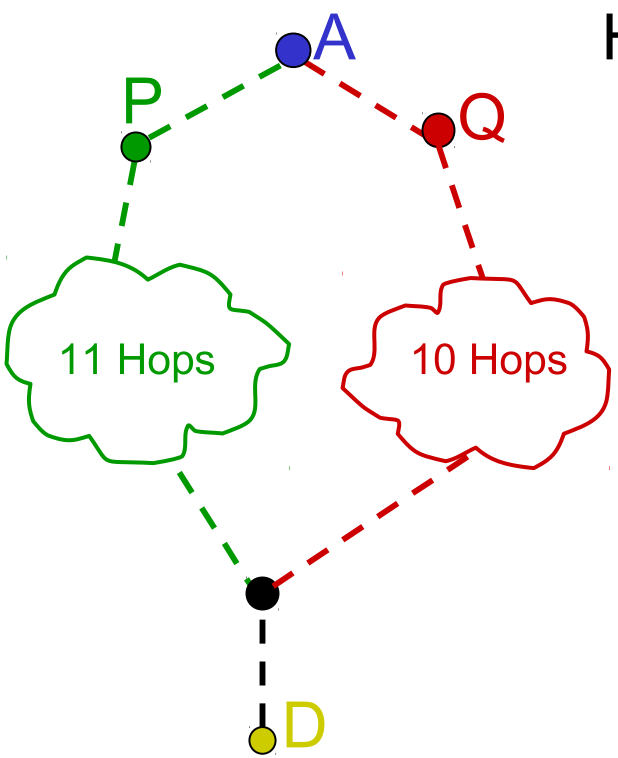


What are Fluctuations

- Entry for D in A: [D, Q, 14, D-100]
- D makes Broadcast with Seq. Nr. D-102
- A receives from P Update (D, 15, D-102)
  -> Entry for D in A: [D, P, 15, D-102]
  A must propagate this route immediately.
- A receives from Q Update (D, 14, D-102)
  -> Entry for D in A: [D, Q, 14, D-102]
  A must propagate this route immediately.

This can happen every time D or any other node does its broadcast and lead to unnecessary route advertisements in the network, so called fluctuations.

DSDV (Damping Fluctuations)



How to damp fluctuations

- Record last and avg. Settling Time of every Route in a separate table. (Stable Data)
  Settling Time = Time between arrival of first route and the best route with a given seq. nr.
- A still must update his routing table on the first arrival of a route with a newer seq. nr., but he can wait to advertising it. Time to wait is proposed to be 2*(avg. Settling Time).
- Like this fluctuations in larger networks can be damped to avoid unececarry adverdisment, thus saving bandwith.

Summery

- Advantages
  - Simple (almost like Distance Vector)
  - Loop free through destination seq. numbers
  - No latency caused by route discovery

- Disadvantages
  - No sleeping nodes
  - Overhead: most routing information never used