Midterm Review

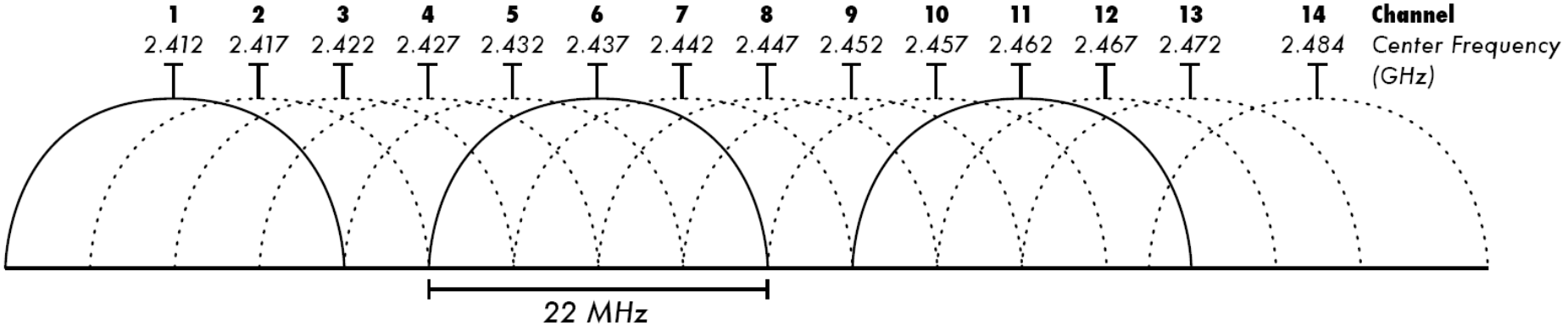
DSR, AODV, DSDV, Sequence Numbers, Link Metrics

Outline

- 802.11 transmission channels
- DSR, AODV, DSDV
- Multiple transmission channels
- Sequence numbers to prevent loops for distance vector algorithms
- How to compute ETX and ETT in practice

802.11 Transmission Channels

- 802.11 = 802.11b, 802.11g, 802.11a, 802.11n
- 802.11b/g: ISM band at 2.5 GHz
  - 2.400–2.500 GHz
- 802.11a: 5 GHz U-NII band
  - From 5.15 to 5.825 GHz



Picture: wikipedia

DSDV

- DSDV: Destination Sequence Distance Vector
- Distance vector adapted to wireless
- Sequence number against loops
- Full dump infrequently, incremental updates more frequently

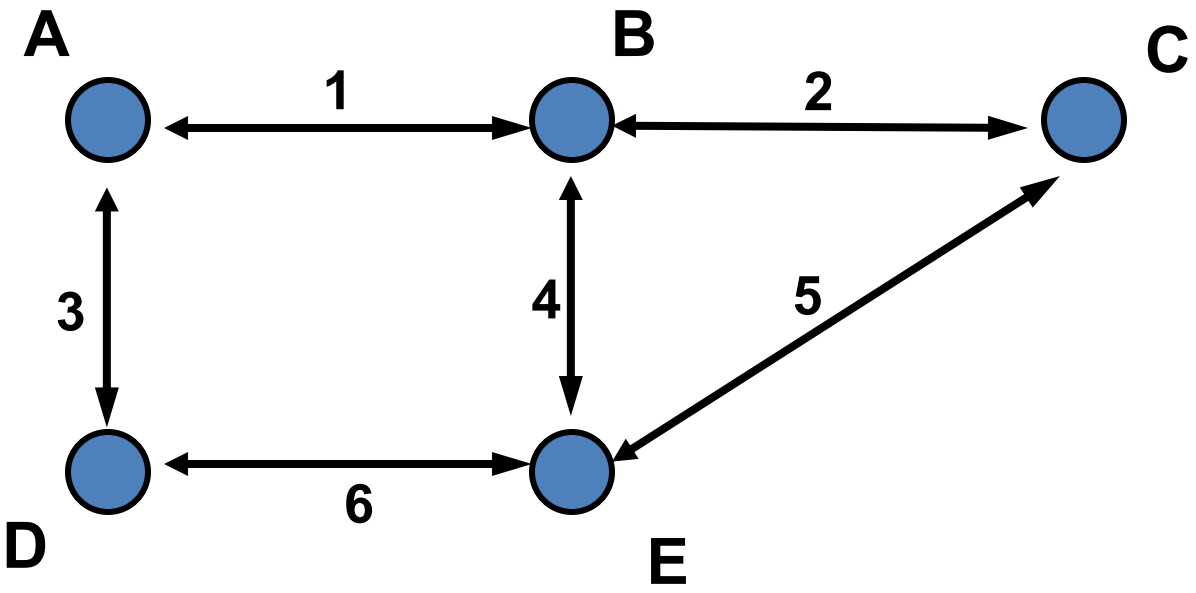
DSR

- On-demand algorithm
- Source routing
- RREQ, RREP, RERR
- Caching
- Many optimizations

AODV

- On-demand algorithm
- Distance vector: routing table at nodes
- RREQ, RREP
  - Path discovery, reverse path setup, forward path setup
- Sequence number

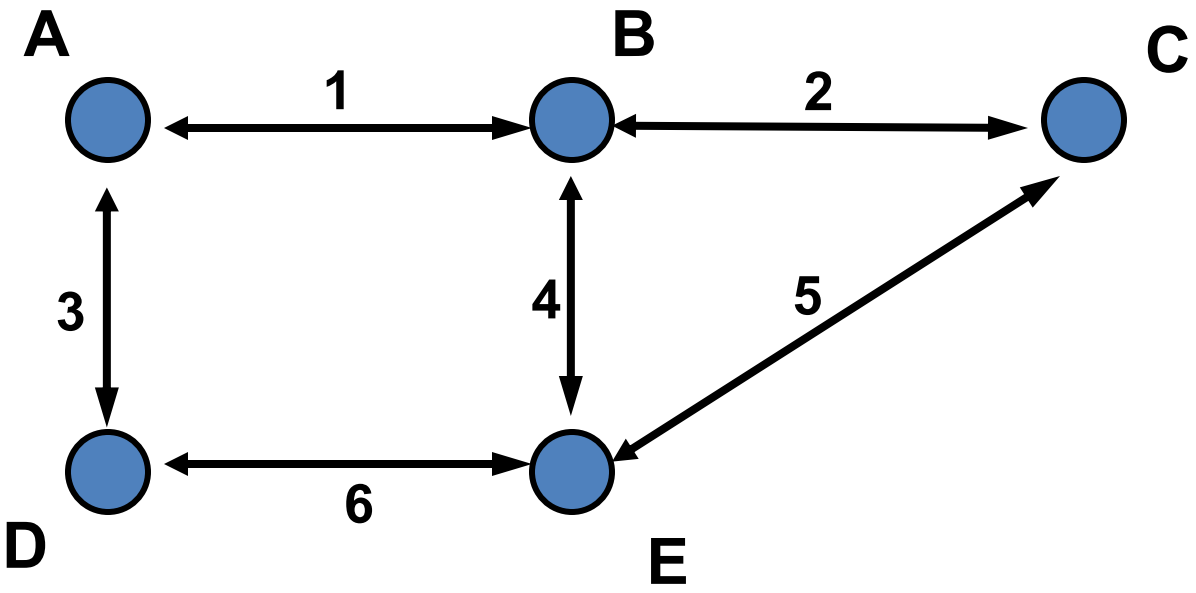
Distance Vector Example



- Symmetric links, link cost = 1
- A, B… E are addresses
- Local knowledge

Example borrowed from "Routing in the Internet", Huitema, 1995

Cold Start

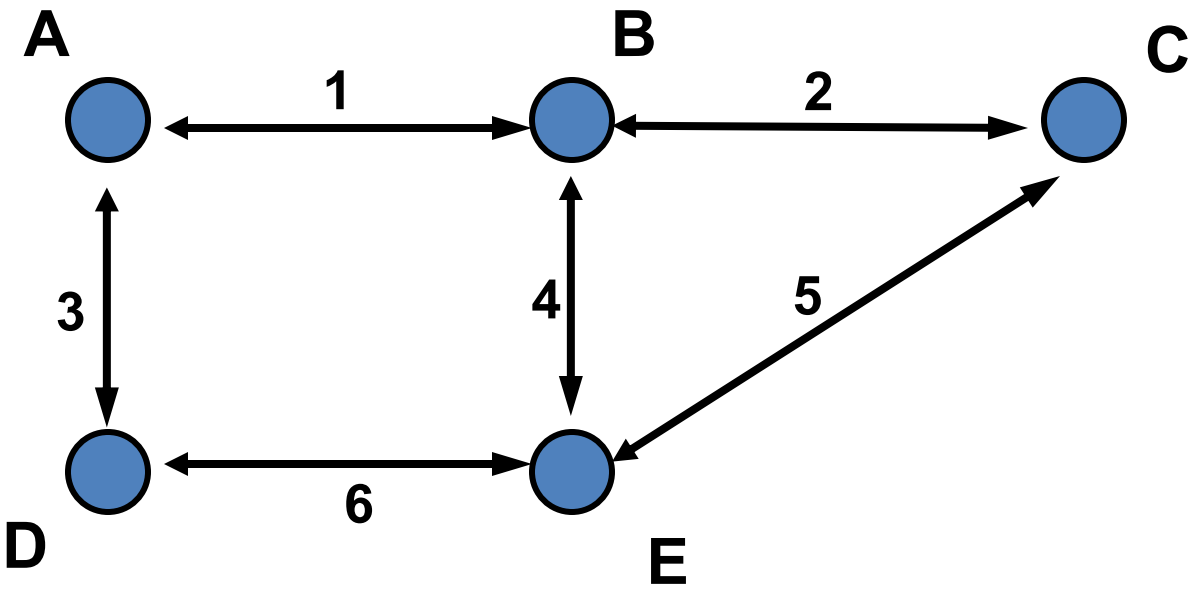


From A to	Link	Cost
A	Local	0

From B to	Link	Cost
B	Local	0

- Tx: DV(A): A=0 [1,3]

Update at B and D



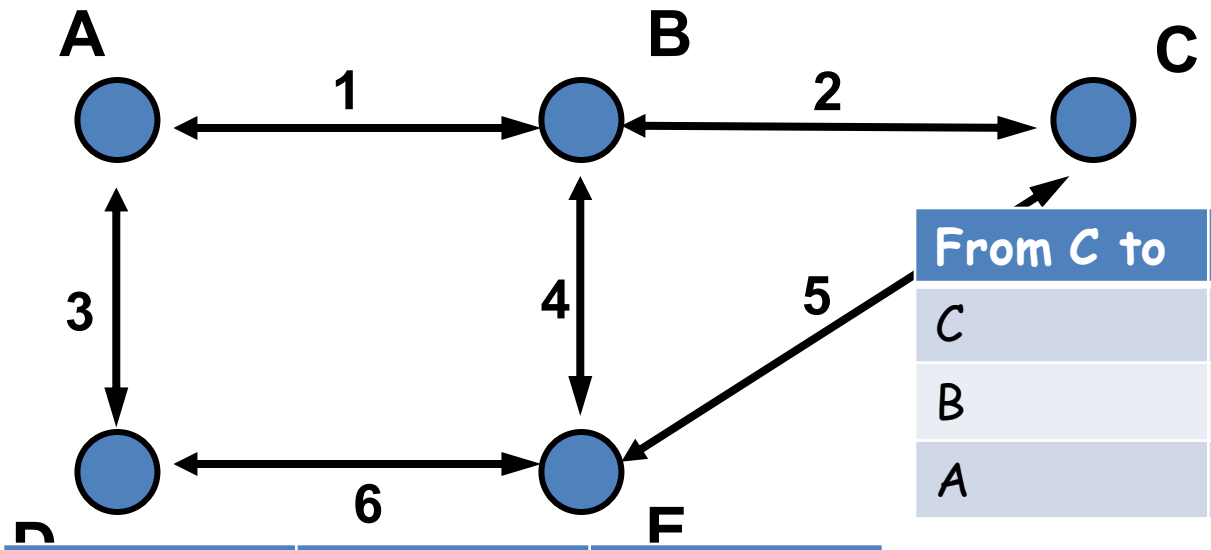
From A to	Link	Cost
A	Local	0

From B to	Link	Cost
B	Local	0
A	1	1

- Tx: DV(B): B=0, A=1 [1,2,4]
- Tx: DV(D): D=0, A=1 [3,6]

From D to	Link	Cost
D	Local	0
A	3	1

Message from B Received First



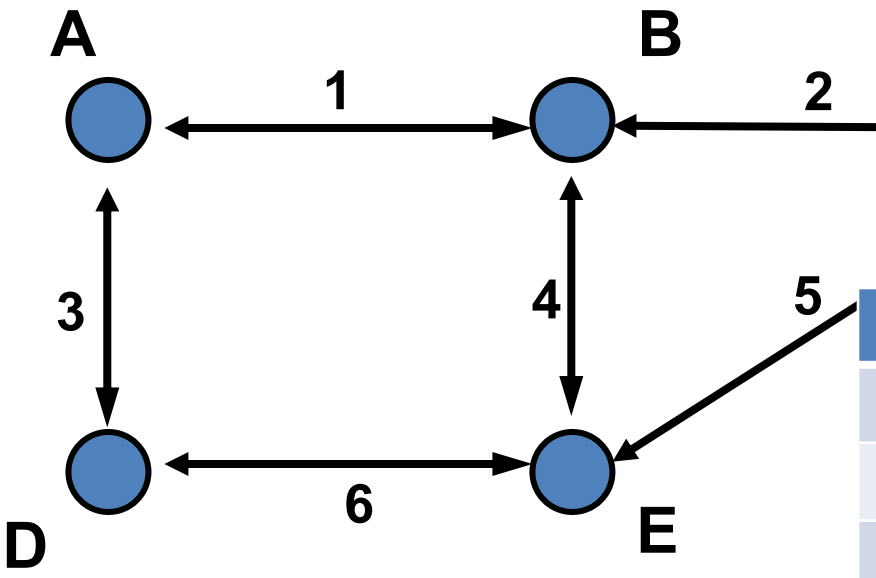
From A to	Link	Cost
A	Local	0
B	1	1
D	3	1

From C to	Link	Cost
C	Local	0
B	2	1
A	2	1

From E to	Link	Cost
E	Local	0
B	4	1
A	4	2

- Rx: DV(B): B=0, A=1
- Rx: DV(D): D=0, A=1

E Receives Message From D



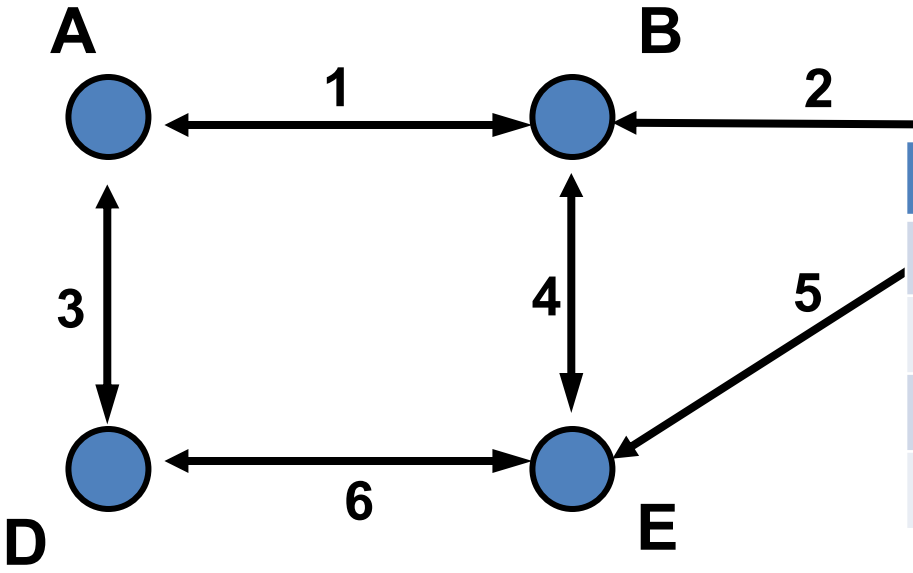
From E to	Link	Cost
E	Local	0
B	4	1
A	4	2
D	6	1

- Rx: DV(D): D=0, A=1

No update of the entry for A

- Tx: DV(A): A=0,B=1,D=1 [1,3]
- Tx: DV(C): C=0, B=1, A=2 [2,5]
- Tx: DV(E): E=0,B=1,A=2,D=1 [4,5,6]

Routing Tables at B,D and E are Updated

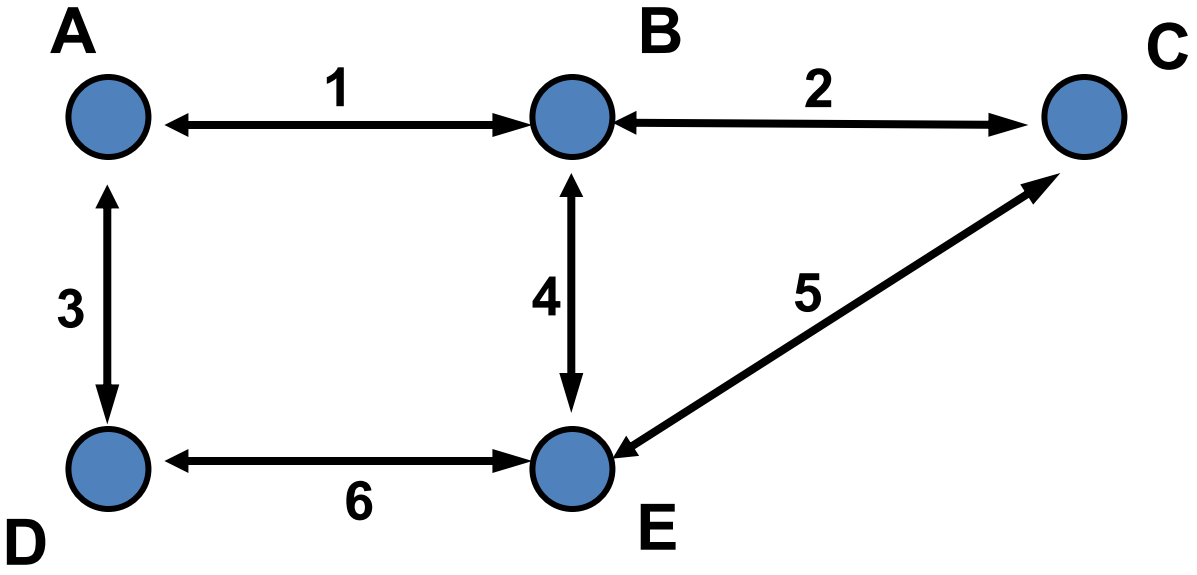


From D to	Link	Cost
D	Local	0
A	3	1
B	3	2
E	6	1

From B to	Link	Cost
B	Local	0
A	1	1
D	1	2
C	2	1
E	4	1

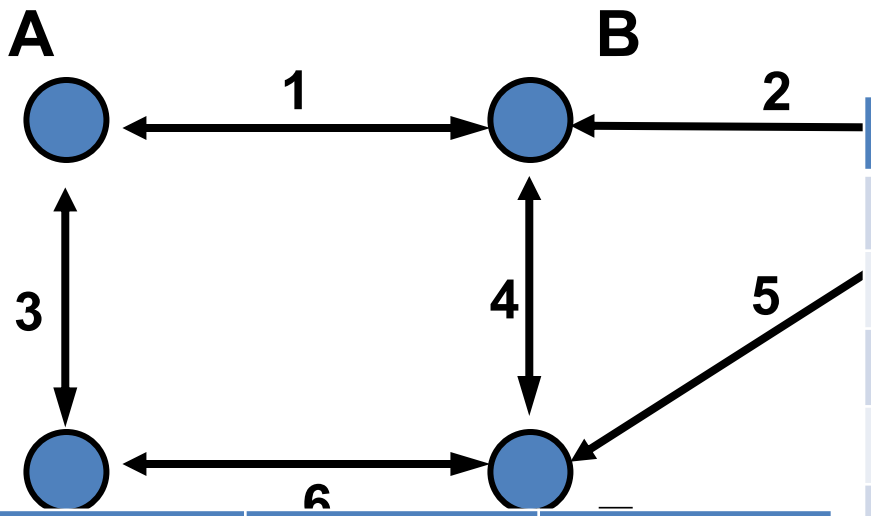
From E to	Link	Cost
E	Local	0
B	4	1
A	4	2
D	6	1
C	5	1

And New DVs are Sent



- DV(B): B=0,A=1,D=2, C=2, E=1 [1,3,4]
- DV(D): D=0, A=1, B=2, E=1 [3,6]
- DV(E): E=0,B=1,A=2,D=1,C=1 [4,5,6]

Routing Tables at A,C and D are Updated



From A to	Link	Cost
A	Local	0
B	1	1
D	3	1
C	1	2
E	1	2

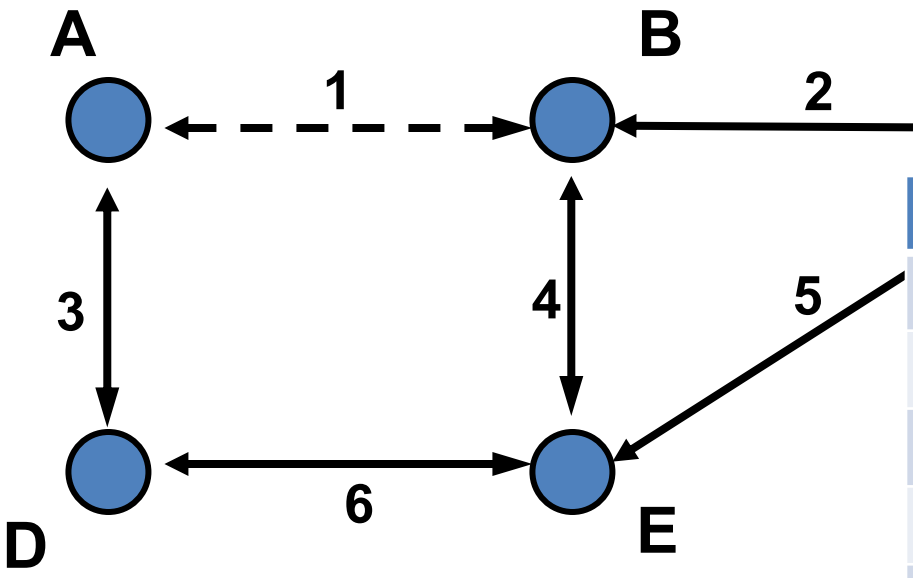
From C to	Link	Cost
C	Local	0
B	2	1
A	2	2
E	5	1
D	5	2

From D to	Link	Cost
D	Local	0
A	3	1
B	3	2
E	6	1
C	6	2

- No more update necessary

If Link 1 Breaks?

- Update at A and B: inf for all routes through link 1

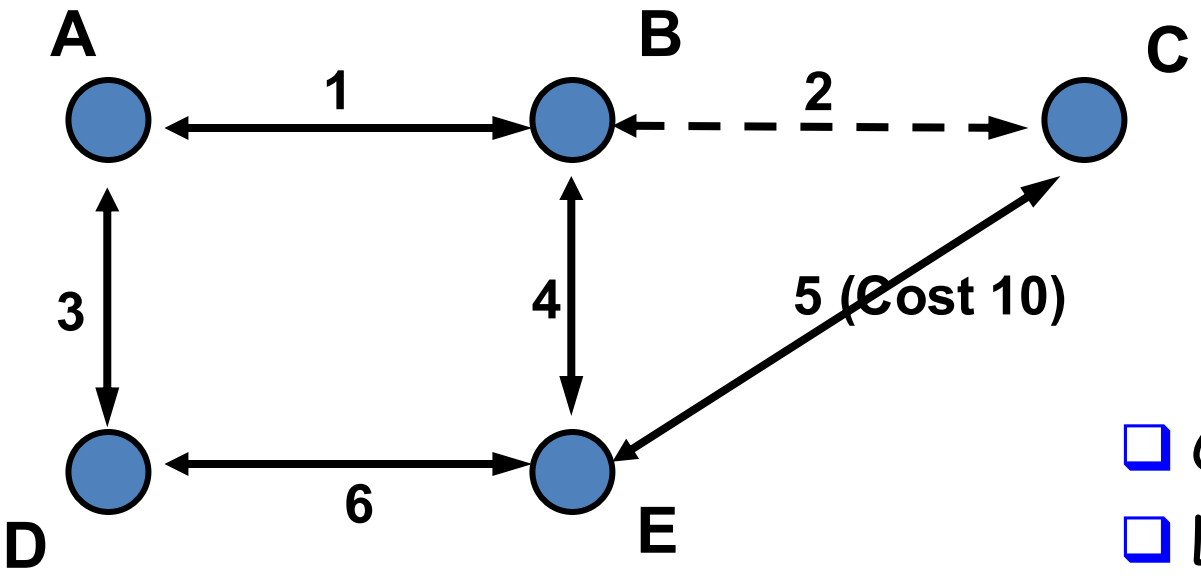


From A to	Link	Cost
A	Local	0
B	1	Inf
D	3	1
C	1	Inf
E	1	Inf

From B to	Link	Cost
B	Local	0
A	1	Inf
D	1	Inf
C	2	1
E	4	1

- Tx: DV(A) … [3]
- Tx: DV(B) … [4,2]
- …
- Low convergence time

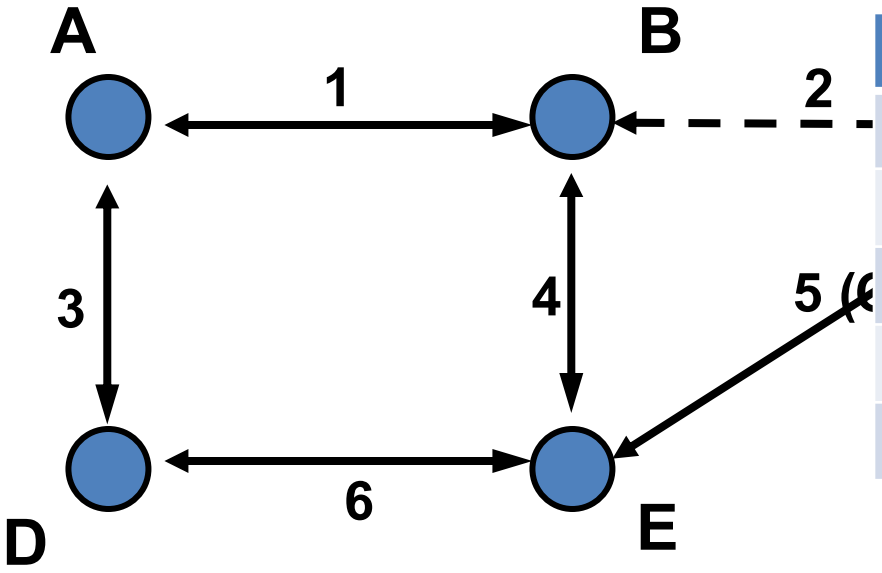
Distance Vector Issues: Temporary Loop



- Cost(5)=10
- Link 2 breaks

From X to C	Link	Cost
A	1	2
B	2	1
C	Local	0
D	3	3
E	4	2

Distance Vector Issues: Temporary Loop

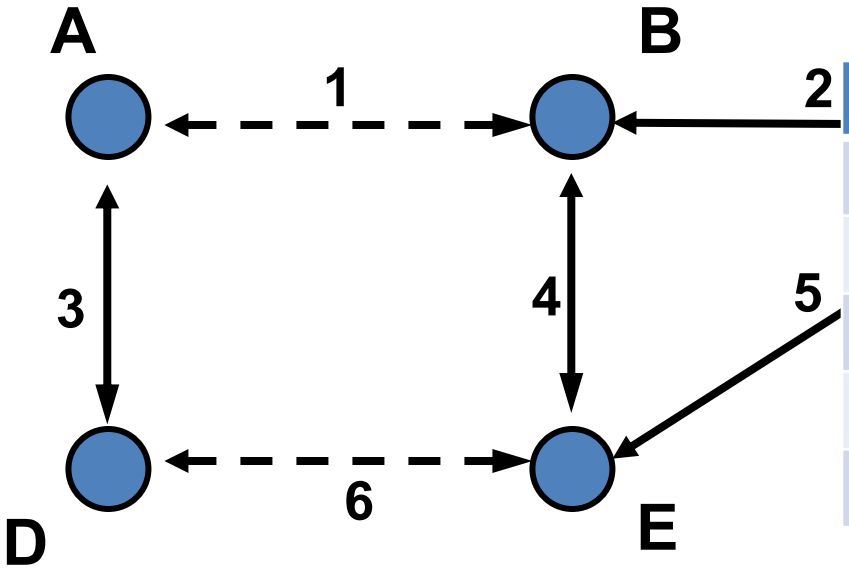


From X to C	Link	Cost
A	1	2
B	2	Inf
C	Local	0
D	3	3
E	4	2

- B notice failure
- A send DV refresh before B:
  DV(A) A=0,B=1,C=2,D=1,E=2
- B send DV update
  DV(B) A=1,B=0,C=3,D=2,E=1
- Temp. loop A-B to C

From X to C	Link	Cost
A	1	4
B	1	3
C	Local	0
D	3	3
E	4	4

Distance Vector Issue: Count to Infinity

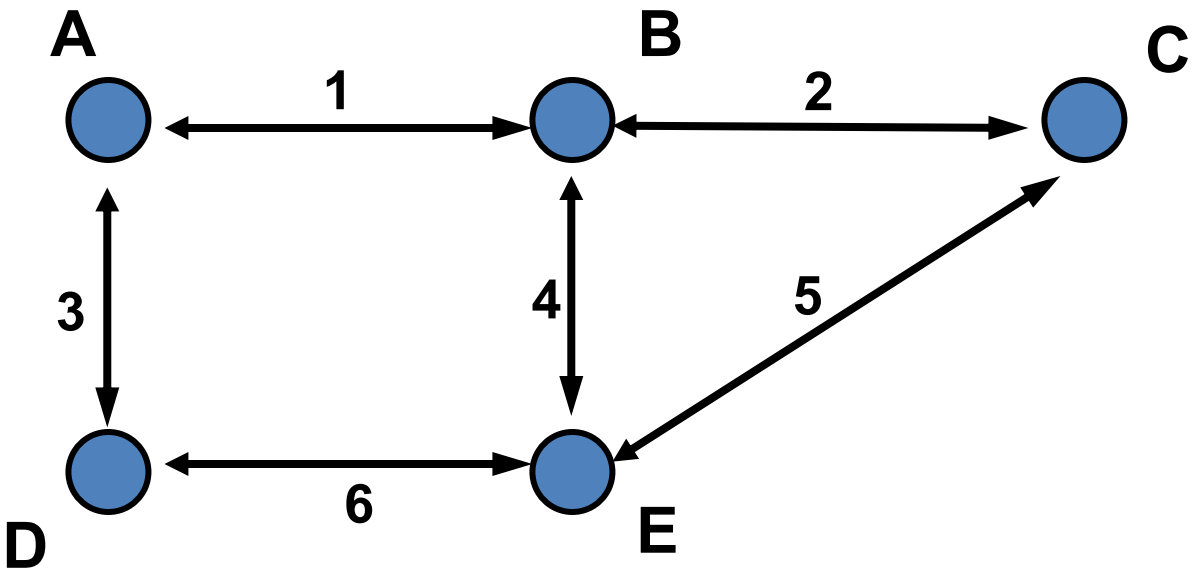


From D to	Link	Cost
D	Local	0
A	3	1
B	3	Inf
E	6	Inf
C	6	Inf

- Link 6 breaks after link 2
- D notice failure
- A send DV refresh before D:
  DV(A) A=0,B=3,C=3,D=1,E=3
- ~~Temp.~~ loop A-B

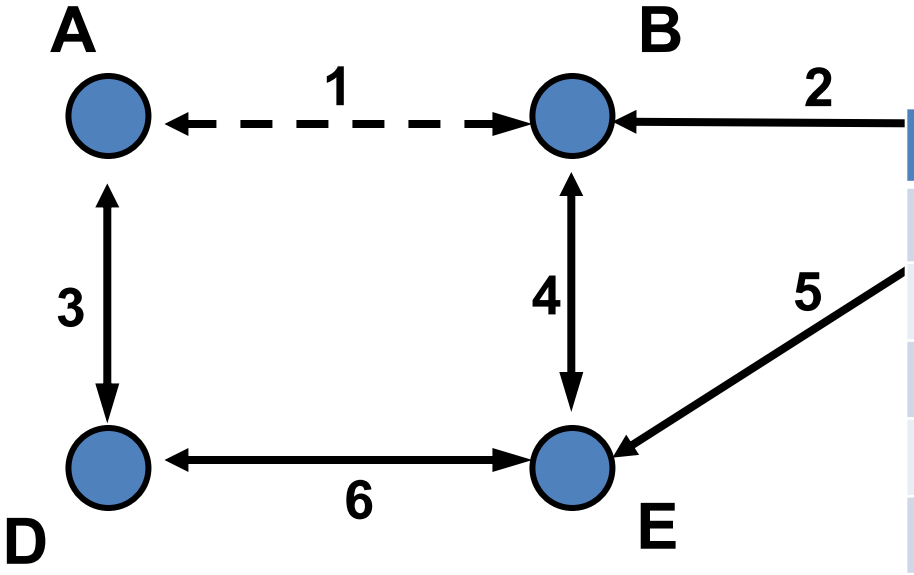
From D to	Link	Cost
D	Local	0
A	3	1
B	3	4
E	6	4
C	6	4

Split Horizon and Co.



- Split horizon: if A to X through B, then B should not try to reach X through A
  - A does not announce route to X to B
- With poisonous reverse: announce route but set cost to Inf

Split Horizon is not Enough



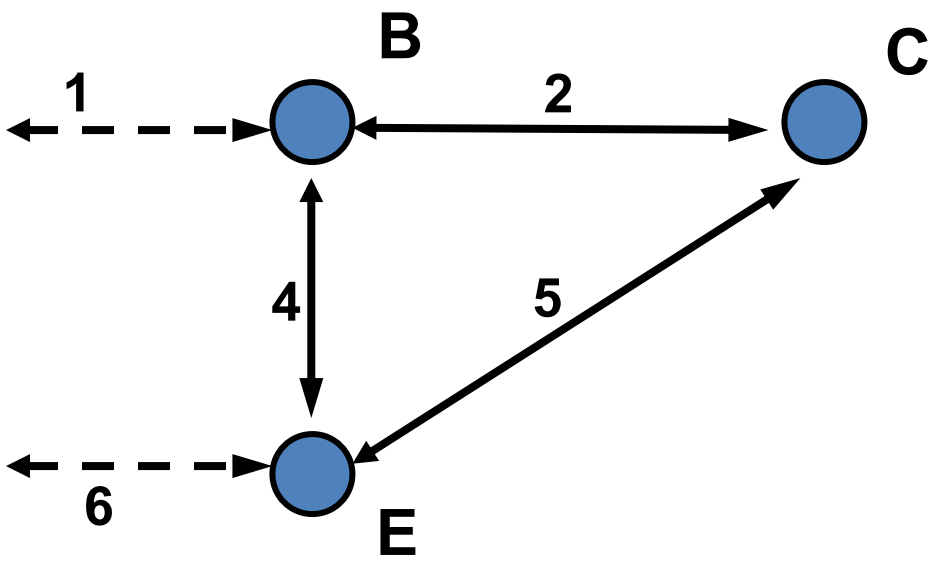
From A to	Link	Cost
A	Local	0
B	3	3
D	3	1
C	3	3
E	3	2

- Split Horizon: DV(A) *empty* [3]
- With poisonous reverse:

  DV(A) B=Inf, D=Inf, C=Inf, E=Inf [3]

Split Horizon Does Not Prevent all Loops

- E-D failure



From X to D	Link	Cost
B	4	2
C	5	2
E	6	Inf

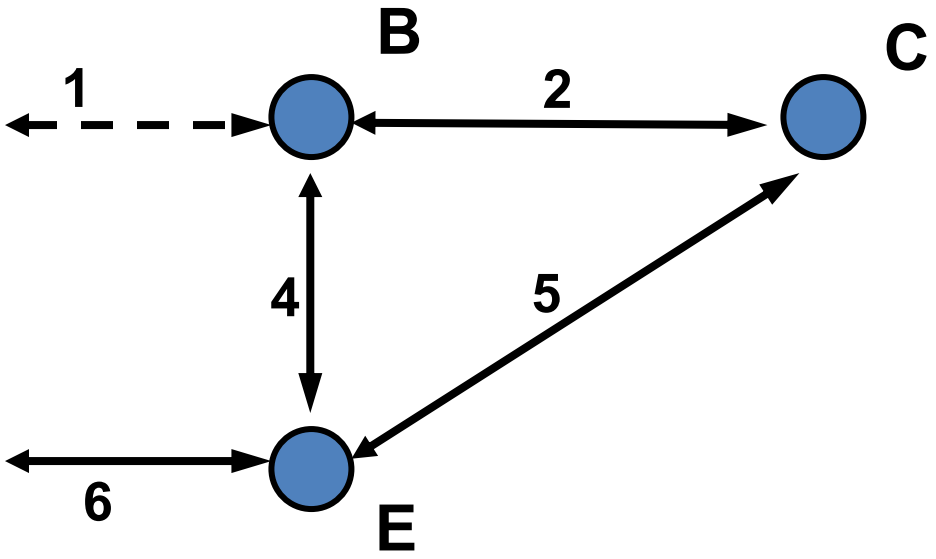
From X to D	Link	Cost
B	4	Inf
C	5	2
E	6	Inf

- DV(E) … D=Inf [4,~~5~~] , tx error on 5
- DV(C) E=Inf… [5], DV(C) E=1,D=2 [2]
- DV(B) D=3… [4]

From X to D	Link	Cost
B	2	3
C	5	2
E	4	4

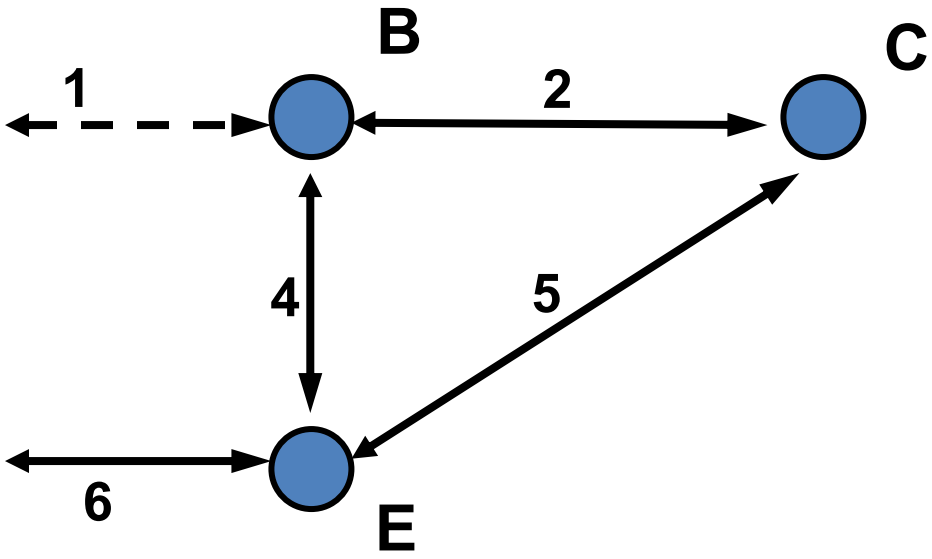
DSDV Use Destination Sequence Numbers to Avoid Loops

- (destination) sequence number for each node
  - Each node increments and appends its sequence number when sending its local routing table
  - This sequence number attached to route entries created for this node



DSDV Use Destination Sequence Numbers to Avoid Loops

From B to	Link	Cost	Seq
A	4	3	50
B	Local	0	60
C	2	1	70
D	4	2	80
E	4	1	90



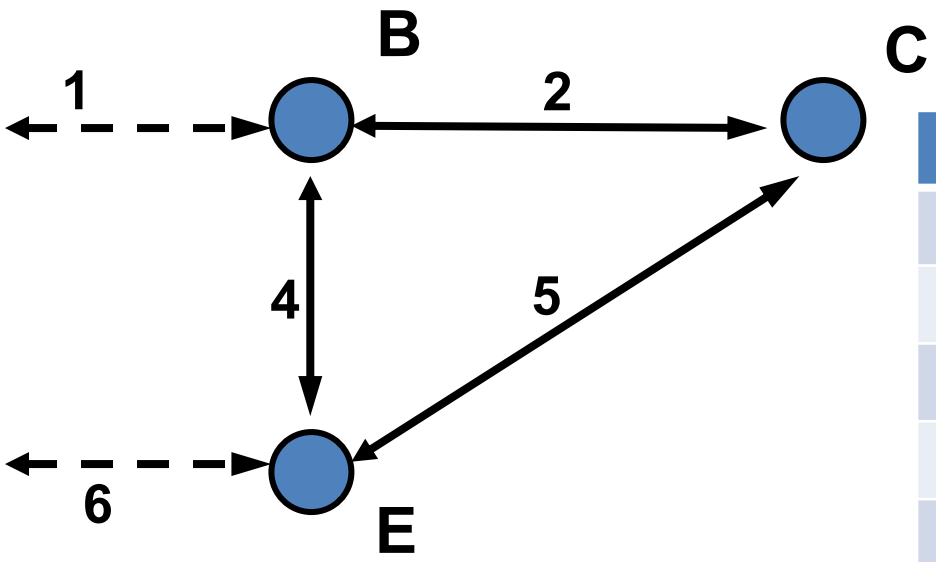
From C to	Link	Cost	Seq
A	5	3	50
B	2	1	60
C	Local	0	70
D	5	2	80
E	5	1	90

From E to	Link	Cost	Seq
A	6	2	50
B	4	1	60
C	5	1	70
D	6	1	80
E	Local	0	90

Destination Sequence Number and Routing Update

1. If S(X) > S(Y), then X ignores the routing information received from Y
2. If S(X) = S(Y), and cost of going through Y is smaller than the route known to X, then X sets Y as the next hop to Z
3. If S(X) < S(Y), then X sets Y as the next hop to Z, and S(X) is updated to equal S(Y)

With Previous Example



From B to	Link	Cost	Seq
A	4	3	50
B	Local	0	60
C	2	1	70
D	4	2	80
E	4	1	90

- DV(E) … D=Inf(80) [4,~~5~~] , tx error on 5
- DV(C) E=Inf(90),… [5], DV(C) E=1(90),D=2(80) [2]
  - Identical sequence number, but longer route: rule 2, update rejected

AODV Uses "Route Request" Sequence Number

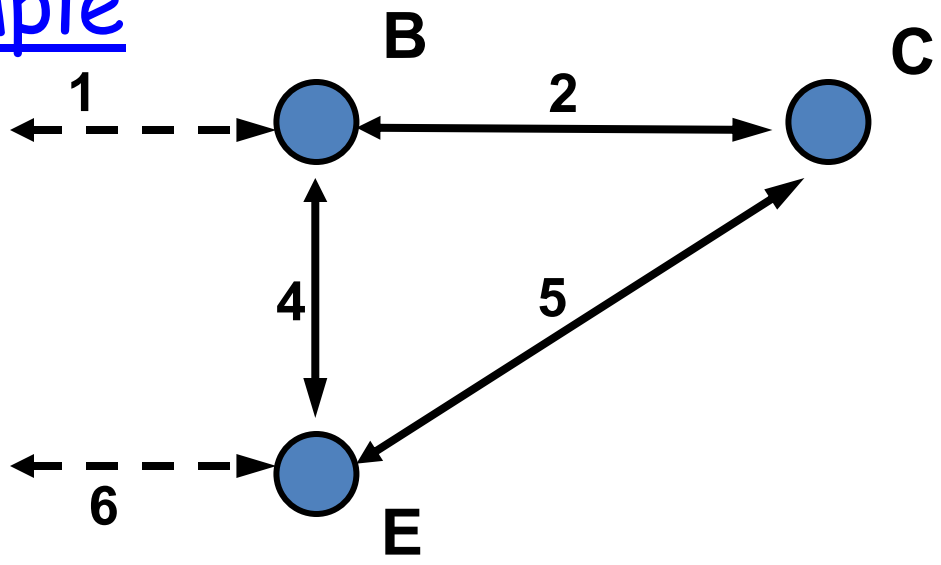
- E unable to forward packet P (from E to D) on link 6, generates a RERR message
- E increments the destination sequence number for D cached at E
- Incremented sequence number *N* included in RERR
- E initiates RREQ for D, use destination sequence number > *N*

- D receives RREQ with destination sequence number N, D set its sequence number to N, unless it is already larger than N

AODV Sequence Number Rule

- If RREQ sequence number > sequence number in table, intermediate node does not answer RREQ

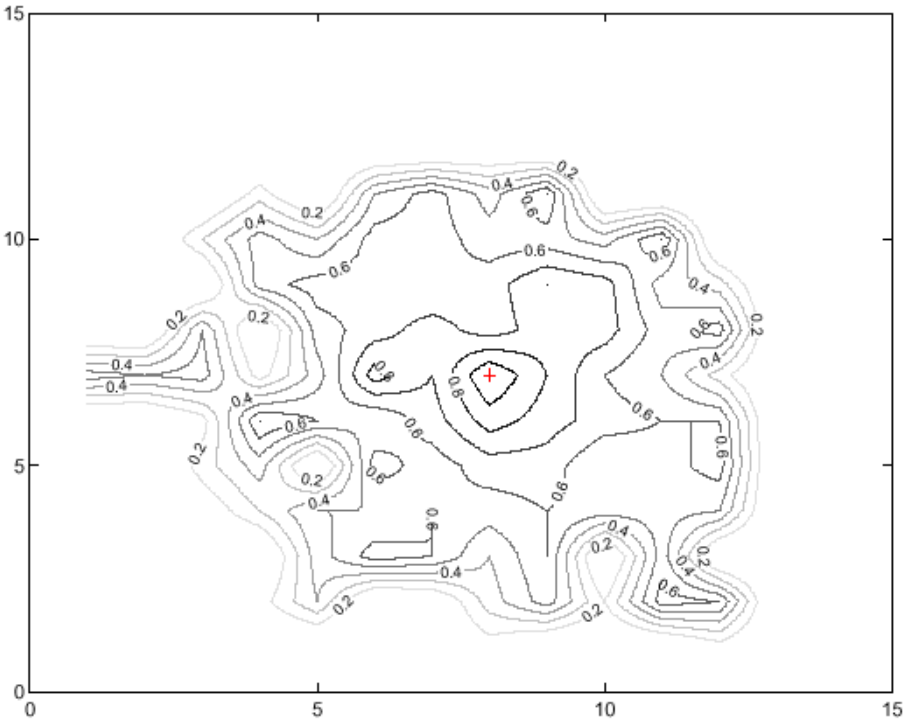
With Previous Example



- E sends to D, link 6 breaks
- E sends RREP(D,Inf), lost on link 4
- E sends RREQ(D,N+2)
- B does not send RREP since N+2 > N

Non-Uniform Packet Reception Probability

- Contour prob. of packet reception around a single node
- Setup: around 160 nodes in a grid with 60 cm spacing
- Heavy-tail behavior

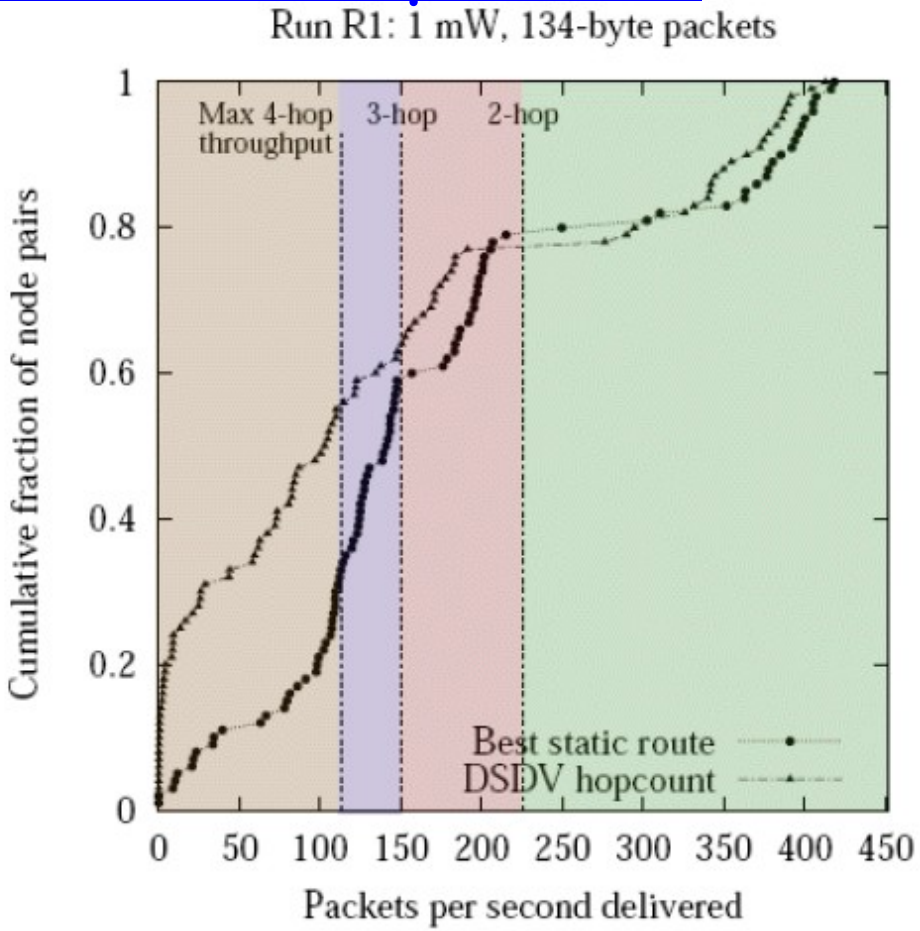


Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks, Ganesan et al., 2002

Performance of Hop Count

x axis:

throughput

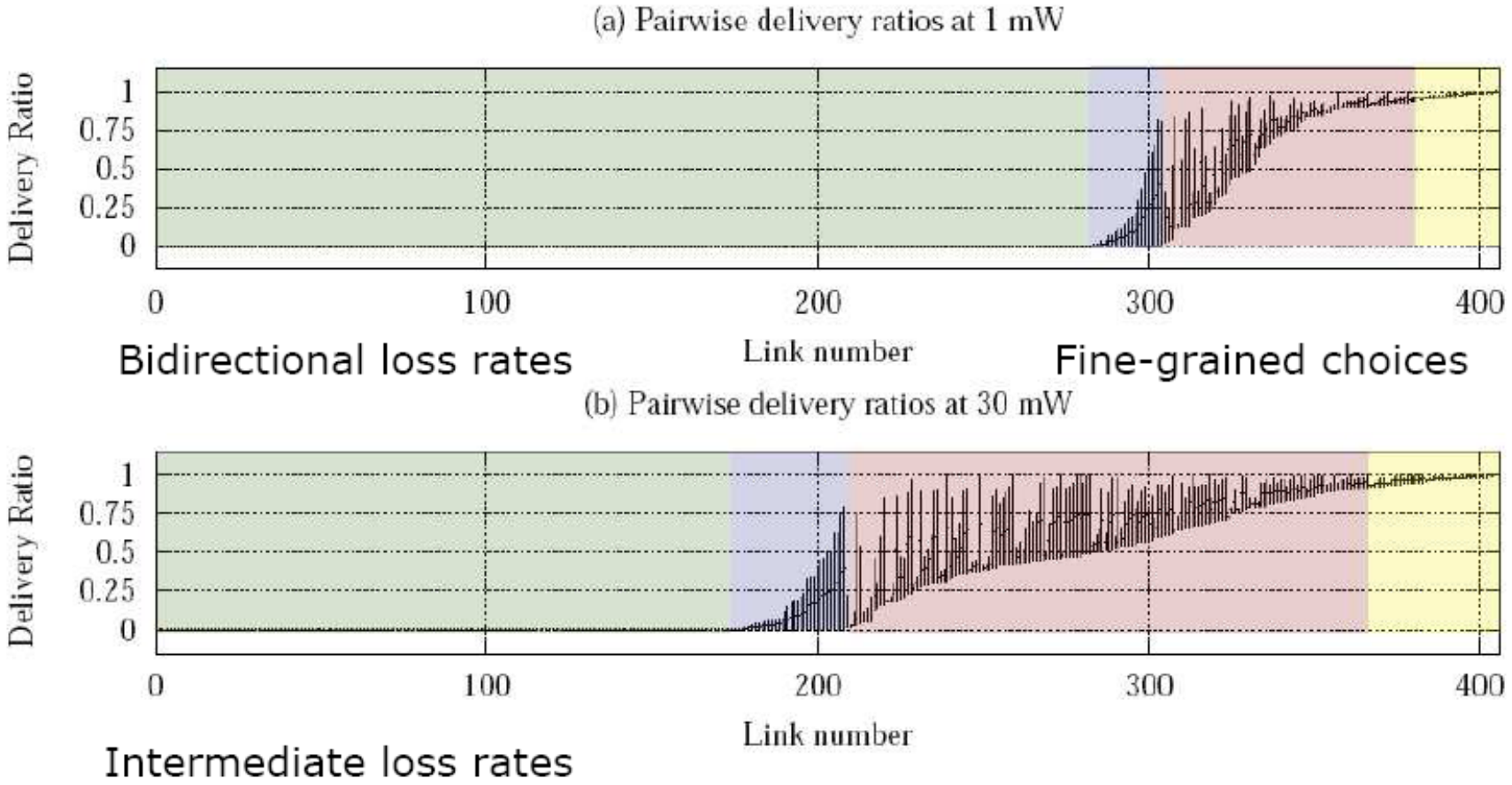


y axis:
fraction of pairs with less throughput

- Setup: 100 random node pairs, throughput CDF, DSDV with min. hop-count vs "best" route

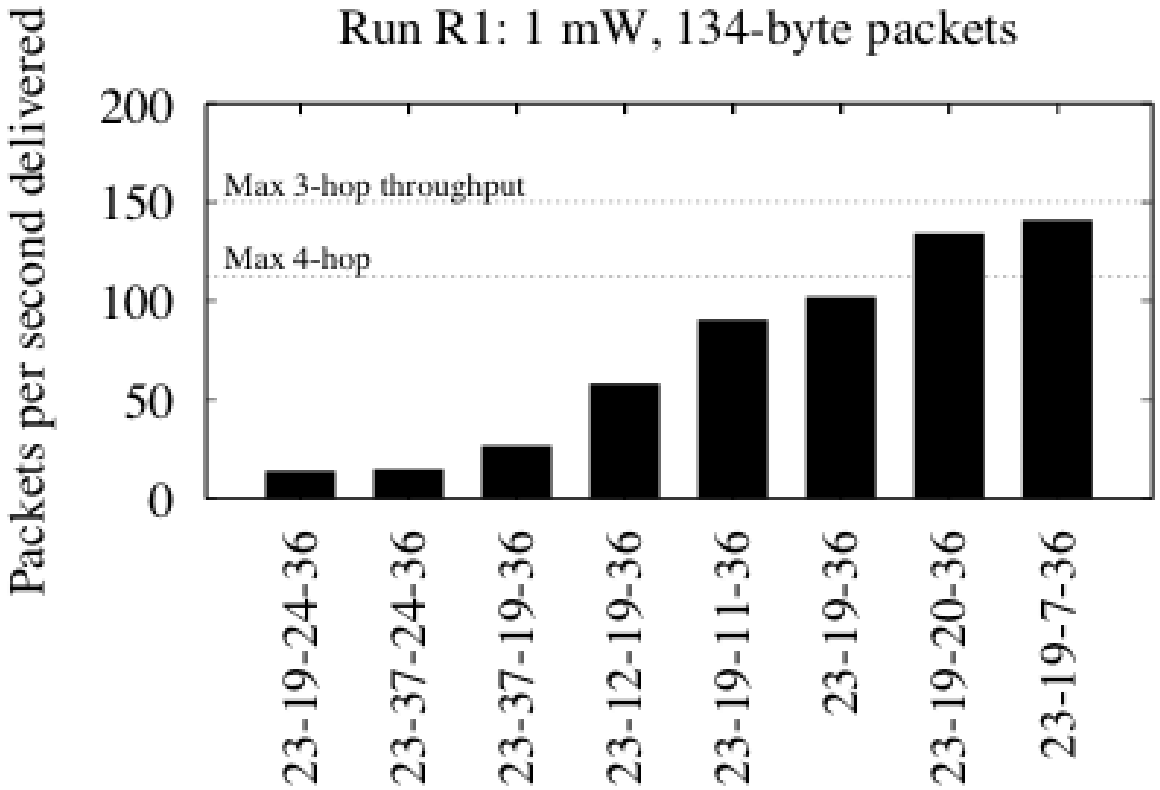
A High-Throughput Path Metric for Multi-Hop Wireless Routing, De Couto et al., 2003

Motivation for a Better Metric



- Distribution of link loss ratio
  - Many links are asymmetric, wide range of loss ratios

Same Hop Count, Different Throughput



Expected Transmission Count (ETX)

- In theory: $$ETX = \frac{1}{d_f \times d_r}$$
  - d_f : forward delivery ratio
  - d_r : reverse delivery ratio
  - Assumptions: fixed rate, fixed power, ignore packet size

ETX Original Implementation

- Each node broadcast probe of fixed size at *average* period τ

$$r(t) = \frac{count(t - w, t)}{w / \tau}$$

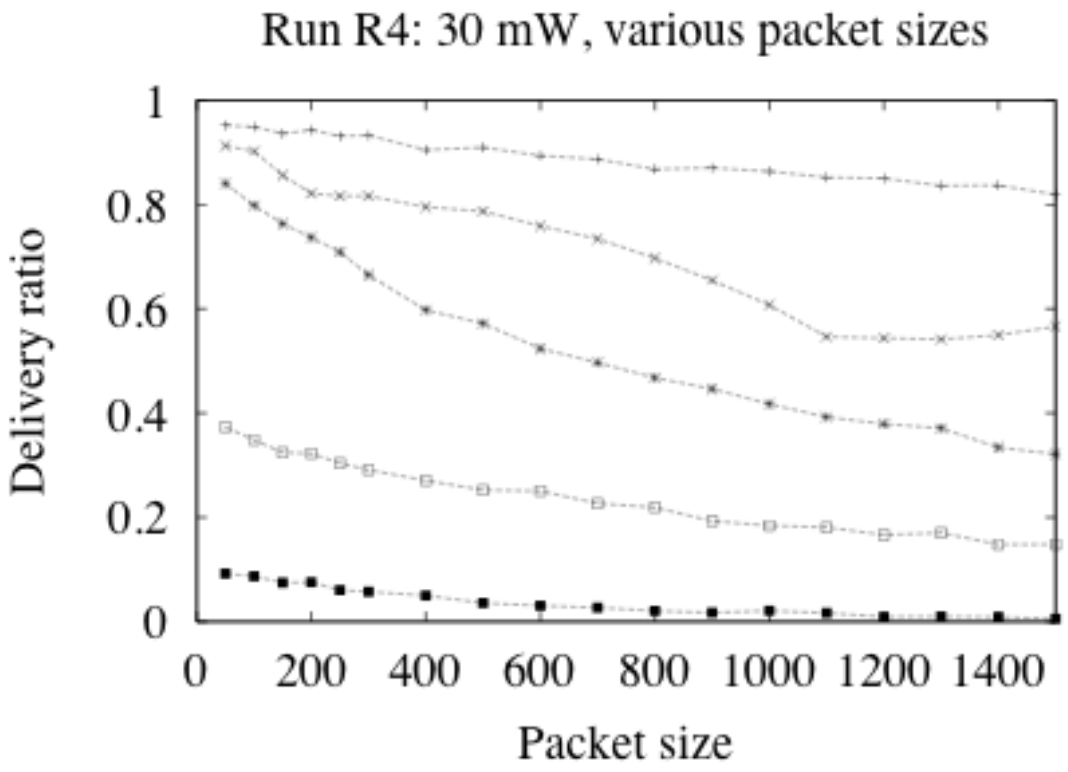
  - Typical: size 134 bytes, $w = 10, \tau =$

- Each probe contains # probes received from each neighbor

- Use 802.11 broadcast frames, 1 Mbps, 1 mW
  - No ack, no retransmission

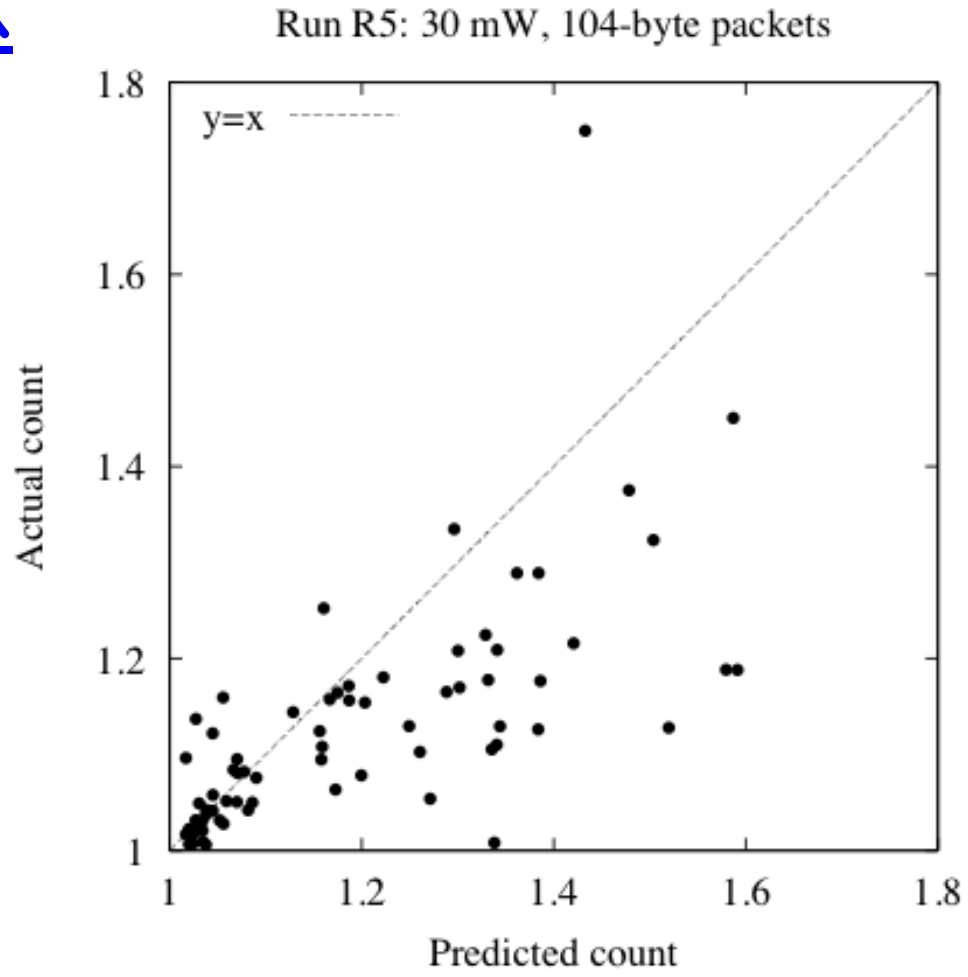
A High-Throughput Path Metric for Multi-Hop Wireless Routing, De Couto et al., 2003

Packet Loss Varies with Packet Size



- Packet size typically larger than probe size (193 bytes with overhead)

Accuracy of ETX



- ETX overestimates packet delivery (why? Probably underestimate ACK delivery ratio)

ETX in OLSR: olsr.org implementation

- Link quality computed from HELLO packets reception
  - Sent every two seconds
- New LQ Hello messages (broadcast link quality computed for each neighbor)
  - Non RFC 3626 compliant
- Extend TC messages (how good links are)
  - Non RFC 3626 compliant

http://www.olsr.org/docs/README-Link-Quality.html

Expected Transmission Time (ETT)

- Start with ETX, multiply with link bandwidth

$$ETT = \frac{S}{B} \times ETX$$

  - S: packet size, B: link bandwidth (raw data rate)

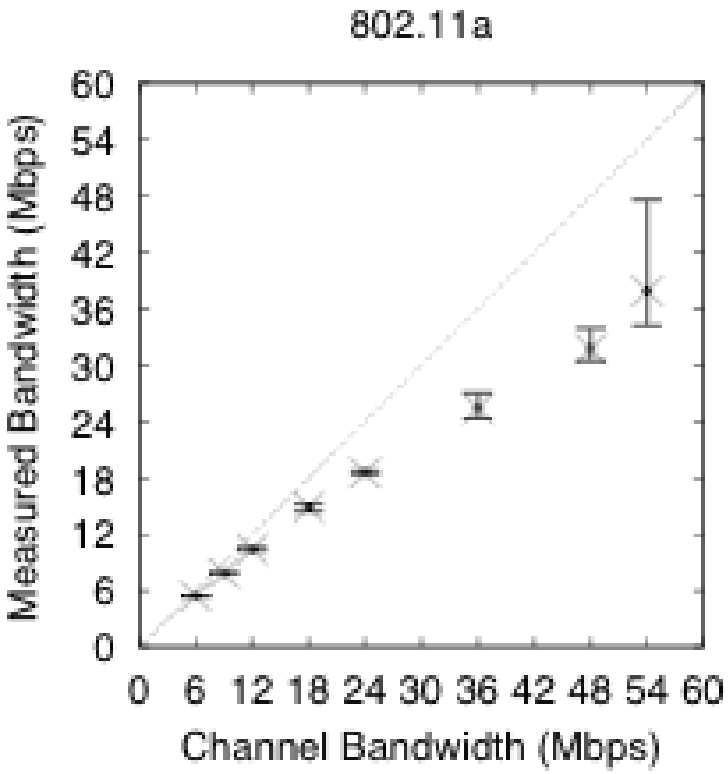
- How to find B?
  - Fix rate / autorate / …
  - Use measurement: packet pairs

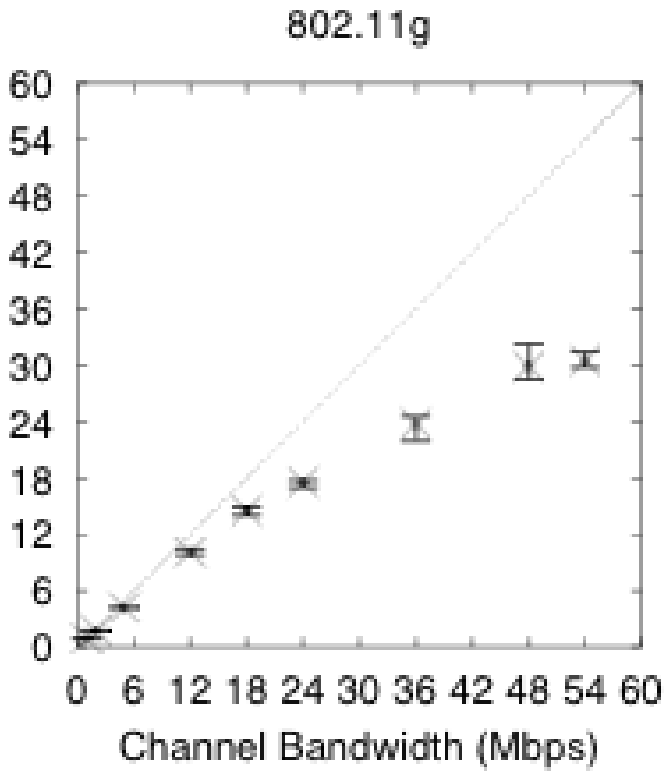
Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks, Draves et al., 2004

How to Compute the ETT = Compute B

- Packet pairs:
  - Send two packets back-to-back: a short (137) followed by a long (1137)
  - Measure time difference of reception
  - Send back to transmitter
  - Min of 10 consecutive samples, divide size of long packet by min value
- Implementation
  - 802.11 broadcast frame for loss rate
  - Unicast packet for bandwidth (because use of autorate)
- Why not RTT: $O(n^2)$

Accuracy of Packet Pairs Measurements

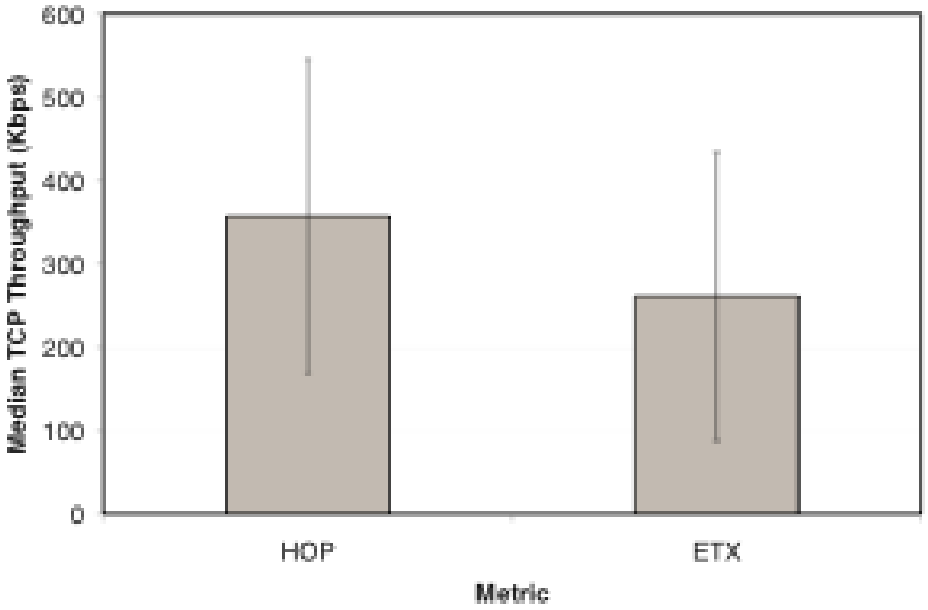




Another ETT Implementation

- Measure delivery probability and compute ETT
- Delivery prob. measurement
  - For each 802.11b rate: send periodic 1500-byte broadcasts
  - Periodic minimum size 60-byte broadcast at 1 Mbps
  - Nodes keep track of received broadcasts and report back to neighbors
  - Delivery prob. = fraction rx @ 1500 * fraction rx @ 60
  - Accounts for ACK, for each rate
- ETT computation (for best delivery prob.)
  - Delivery prob. * Transmission time of 1500-byte frame

ETX vs Hop Count with Mobility



- Scenario: 45 1-min. TCP transmissions with a mobile sender in a static mesh network

Comparison of Routing Metrics for Static Multi-Hop Wireless Networks, Draves et al., 2004

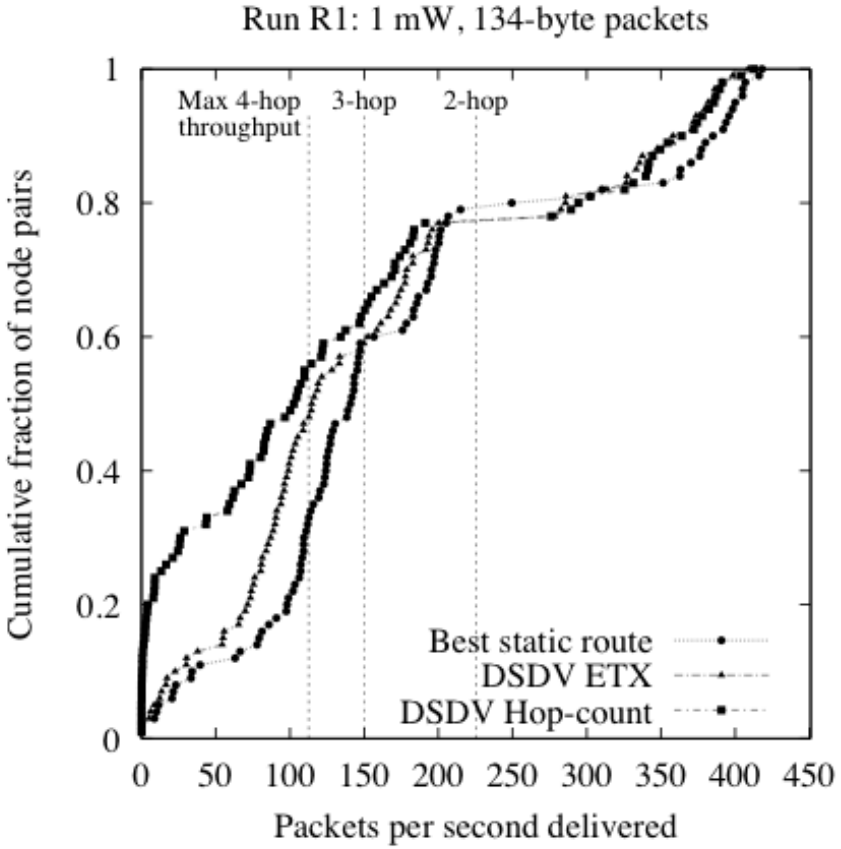
DSR with ETX

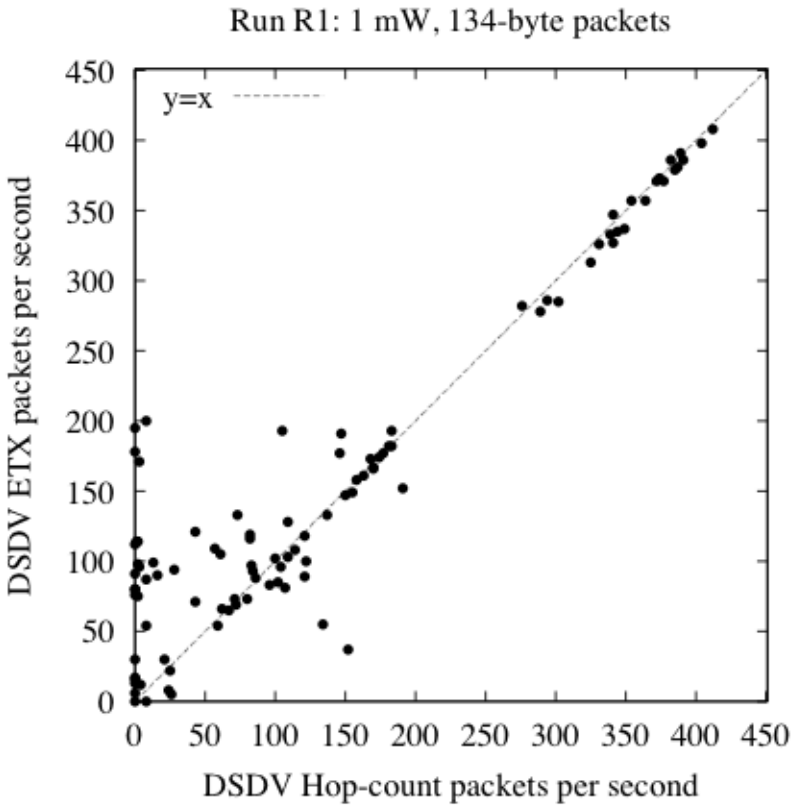
- DSR Modifications
  - *Link* cache, run Dijkstra to find best route (at source)
  - ETX measurement using periodic broadcast link probes
  - RREQ forwading: forwarder address + ETX of incoming link
  - IF RREQ with better accumulated metric for given ID: forward again
  - Metrics included in RREP
  - No overhearing, no salvaging, no reply from cache

DSDV with ETX

- Use short (134 bytes payload) probe every second
- Remember probes over last 10 seconds

Do We Gain Anything?

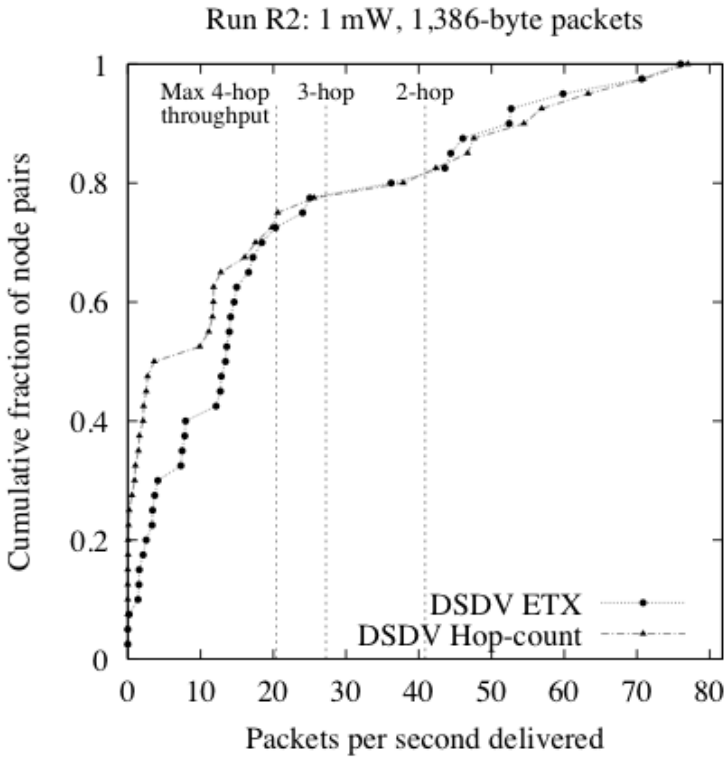


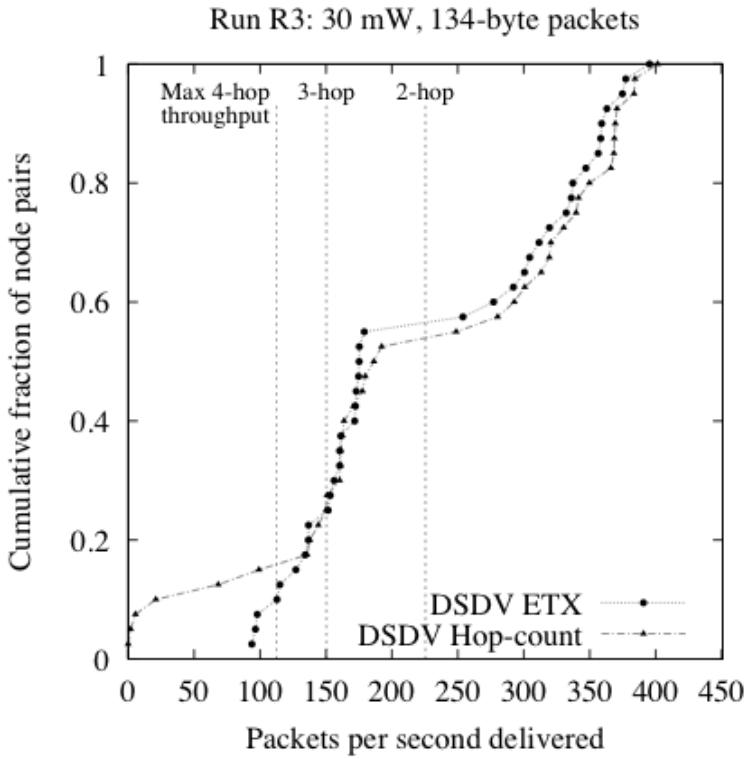


- Setup: 100 random node pairs, throughput CDF

A High-Throughput Path Metric for Multi-Hop Wireless Routing, De Couto et al., 2003

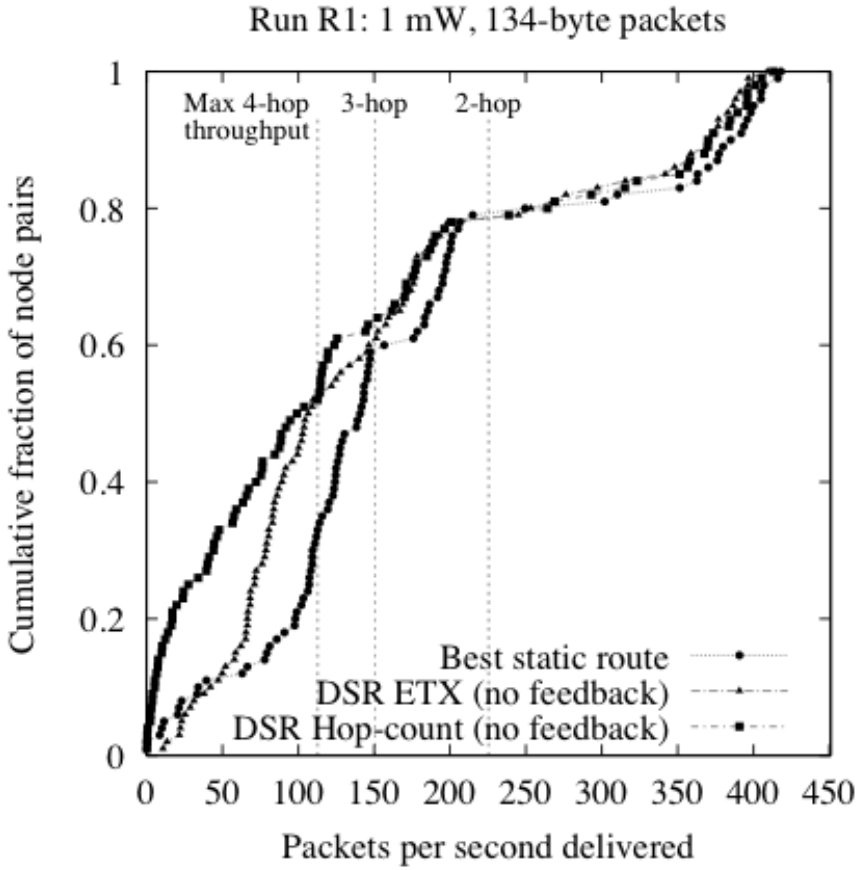
DSDV: Hop Count vs ETX





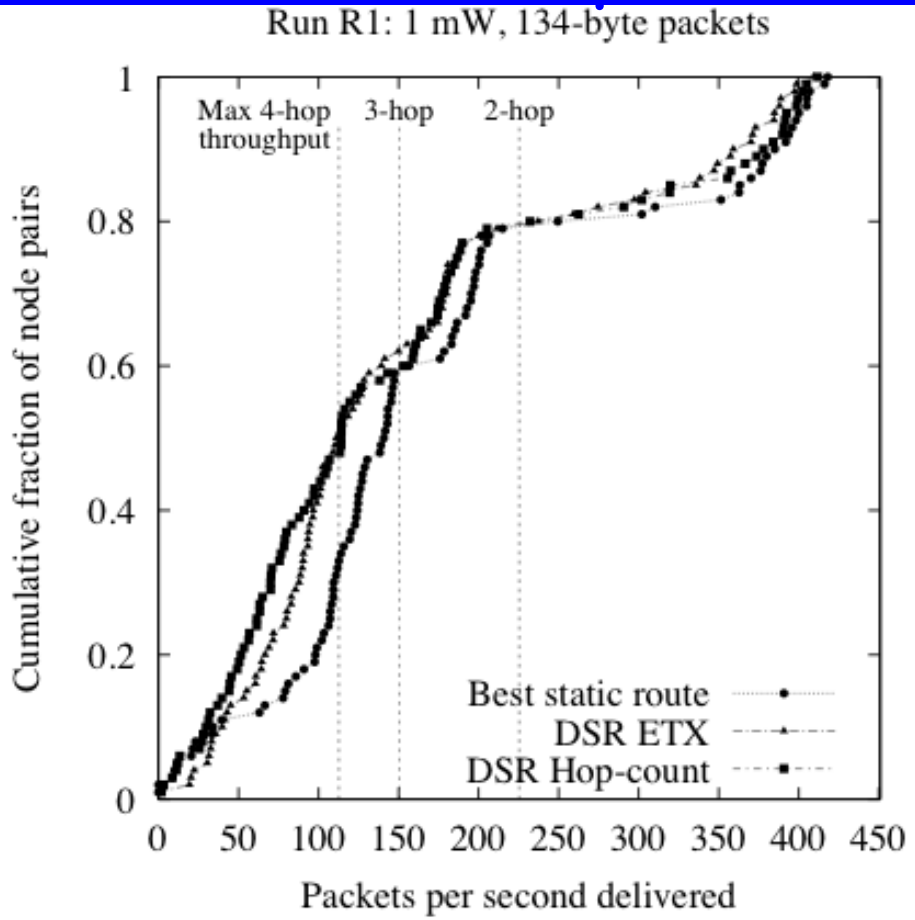
- Setup: 40 pairs from the 100 random node pairs
- Higher tx power increases connectivity

DSR: Hop Count vs ETX



- ETX helps for initial route choice

DSR with feedback: Hop Count vs ETX



- Failure feedback removes low throughput routes

Summary

- Link metrics play a crucial role in the performance of a routing algorithm for wireless networks
- They are still an active area of research
- In practice (today), ETX appears to be the most and widely used metric
- Look at the system overall result