



# Use NS to Simulate Adhoc Routing Protocol

Zhi Ping Lee



# Outline

1. Node configuration
2. Set trace file
3. Create node
4. Create node movements
5. Pause time
6. Set network traffic
7. Trace file



# Node configuration

1. -channelType Channel/WirelessChannel\  
2. -topoInstance \$topo \  
3. -agentTrace ON \  
4. -routerTrace ON \  
5. -macTrace ON \  
6. -movementTrace ON

# Set trace file

1. `set tracefd [open trace.tr w]`
2. `$ns_ trace-all $tracefd`
3. `set namfd [open trace.nam w]`
4. `$ns_ namtrace-all-wireless $namfd 1000 1000`

# Create node

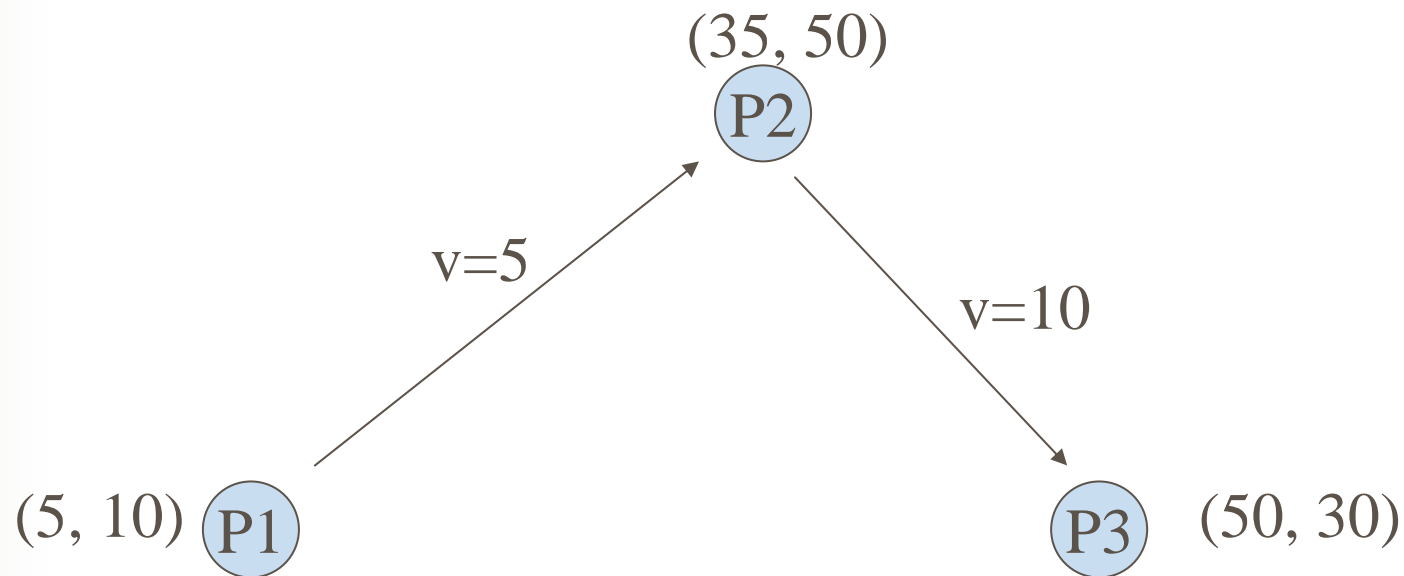
1. create-god 2
2. set node1 [\$ns\_ node]
3. set node2 [\$ns\_ node]
  
4. create-god 50
5. for {set i 0} {\$i < 50 } {incr i} {
6.     set node\_(\$i) [\$ns\_ node]
7. }



# Create node movements

1. `$node1 set X_ 5.0`
2. `$node1 set Y_ 2.0`
3. `$node1 set Z_ 0.0`
4. `$ns_ at 10.0 "$node1 setdest 20.0 18.0 1.0"`

# Pause time



1. Node start moving from p1 at 5.0s
2. Node move to p2 at 15.0s
3. Node start moving from p2 at 25.0s
4. Node move to p3 at 27.5s
5. Node start moving from p3 at 32.5s

Pause time: 10s

# Pause time

1. `$ns_ at 5.0 "$node1 setdest 35.0 50.0 5.0"`
2. `$ns_ at 25.0 "$node1 setdest 50.0 30.0 10.0"`
3. `$ns_ at 32.5 "$node1 setdest new_x new_y new_v"`

# Set network traffic

1. set udp\_ [new Agent/UDP]
2. set null\_ [new Agent/Null]
3. \$ns\_ attach-agent \$node1 \$udp\_
4. \$ns\_ attach-agent \$node2 \$null\_
5. \$ns\_ connect \$udp\_ \$null\_
6. set cbr\_ [new Application/Traffic/CBR]
7. \$cbr\_ attach-agent \$udp\_
8. \$cbr\_ set packetSize\_ 512
9. \$cbr\_ set interval\_ 0.1
10. \$cbr\_ set maxpkts\_ 10000
11. \$cbr\_ set random\_ 1
12. \$ns\_ at 20.0 "\$cbr\_ start"

# Simulation end

1. for {set i 0} {\$i < 50 } {incr i} {
2.     \$ns\_ at 100.0 "\$node\_(\$i) reset";
3. }
4. \$ns\_ at 100.0 "stop"
5. \$ns\_ at 100.01 "puts \"NS EXITING...\" ; \$ns\_ halt"
6. proc stop {} {
7.     global ns\_ tracefd namfd
8.     \$ns\_ flush-trace
9.     close \$tracefd
10.    close \$namfd
11. }



# Start simulation

1. `$ns_run`



# Trace file format

Format:

[Tag] [Time] [NodeID] [Level] [Information]

Tag- One character, s|r|D|T

The packet is send, receive, drop or just tag.

Time- The record time.

NodeID- Node id.

Level- AGT|RTR|MAC

The trace level.



# Trace file format

Information- packet size, packet id, packet type,  
source, destination etc.

# Trace file

```
s 0.127978400 _8_ AGT --- 0 cbr 512 [0 0 0 0] ----- [8:0 0:0 32 0] [0] 0 0  
r 0.330118537 _0_ AGT --- 0 cbr 532 [a2 0 a 800] ----- [8:0 0:0 28 0] [0] 3 0
```

[tag] [time] [nodeid] [level]

[why] [seq] [packet type] [packet size]

[[duration] [current\_hop][pre\_hop][ether\_type]]

(format\_mac in cmu-trace.cc)

----- [ [source:port] [destination:port] [ttl] [next hop] ]

(format\_ip in cmu-trace.cc)

[seq] [forward num] [opt num forwards]

(format\_rtp in cmu-trace.cc)

# Trace file

```
s 0.128053400 _8_ MAC --- 0 AODV 104 [0 ffffffff 8 800] ----- [8:255  
-1:255 1 0] [0x2 0 1 [0 0] [8 1]] (REQUEST)  
r 0.128469741 _29_ MAC --- 0 AODV 52 [0 ffffffff 8 800] ----- [8:255  
-1:255 1 0] [0x2 0 1 [0 0] [8 1]] (REQUEST)
```

[tag] [time] [nodeid] [level]

[why] [seq] [packet type] [packet size]

[[duration] [destination][source][ether\_type]]

(format\_mac in cmu-trace.cc)

----- [ [source:port] [destination:port] [ttl] [next hop] ]

(format\_ip in cmu-trace.cc)

[ [rq\_type] [hop count] [broadcast id] [[destination] [dest seqno]]

[[source] [src seqno]]

(format\_aodv in cmu-trace.cc)

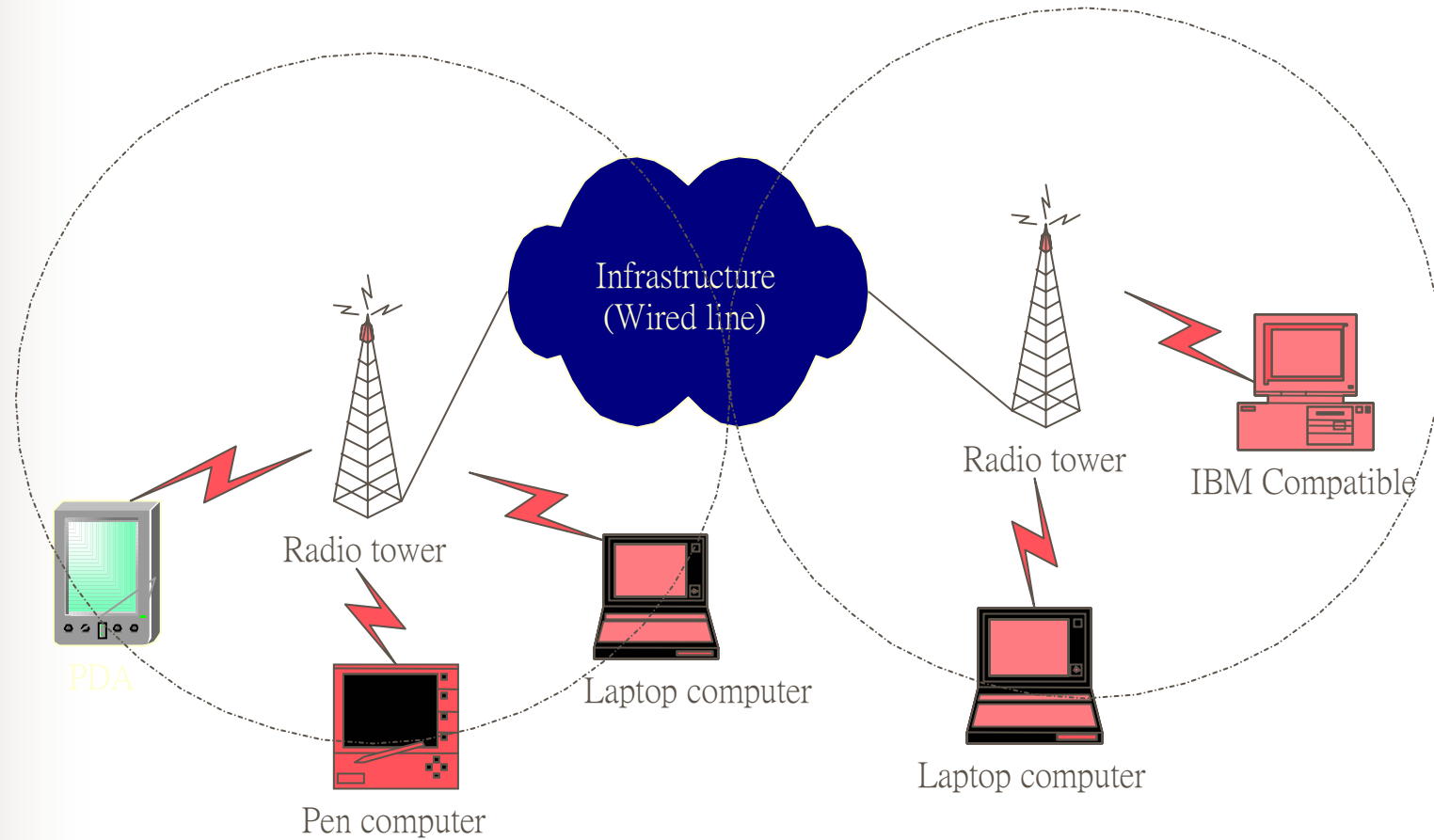


# Wireless networks

## ■ Two variations of mobile wireless networks

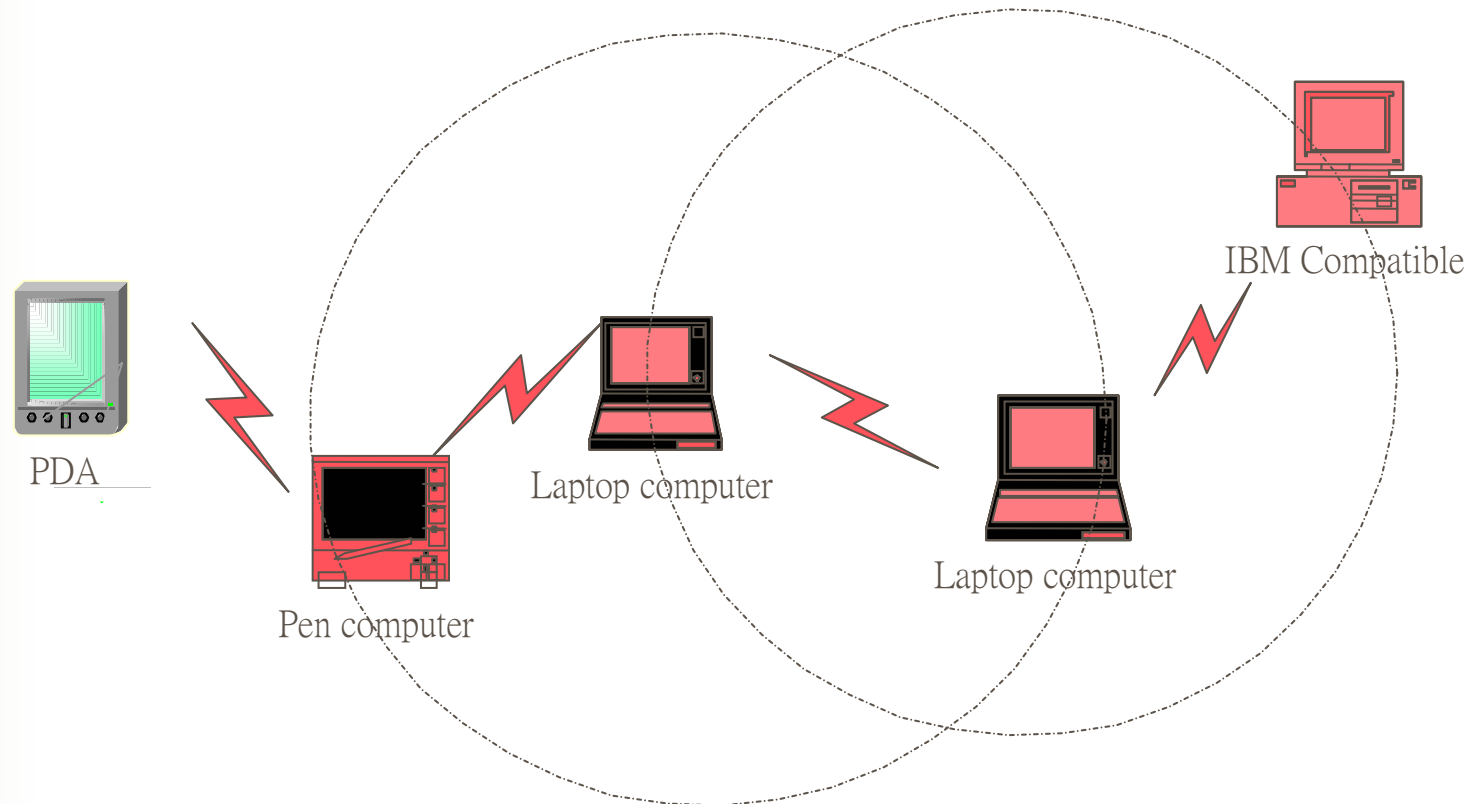
- infrastructured networks
- infrastructureless mobile networks
  - known as ad hoc networks
  - no fixed routers
  - all nodes are capable of movement and can be connected dynamically in an arbitrary manner

# Infrastructured networks



# Infrastructureless networks

- Ad-hoc networks





# Typical limitations of ad-hoc

- high power consumption
- low bandwidth
- high error rates
- high mobility of every network nodes



# Ad-Hoc Routing Protocol

- **AODV**

(Ad hoc on-demand distance vector routing)

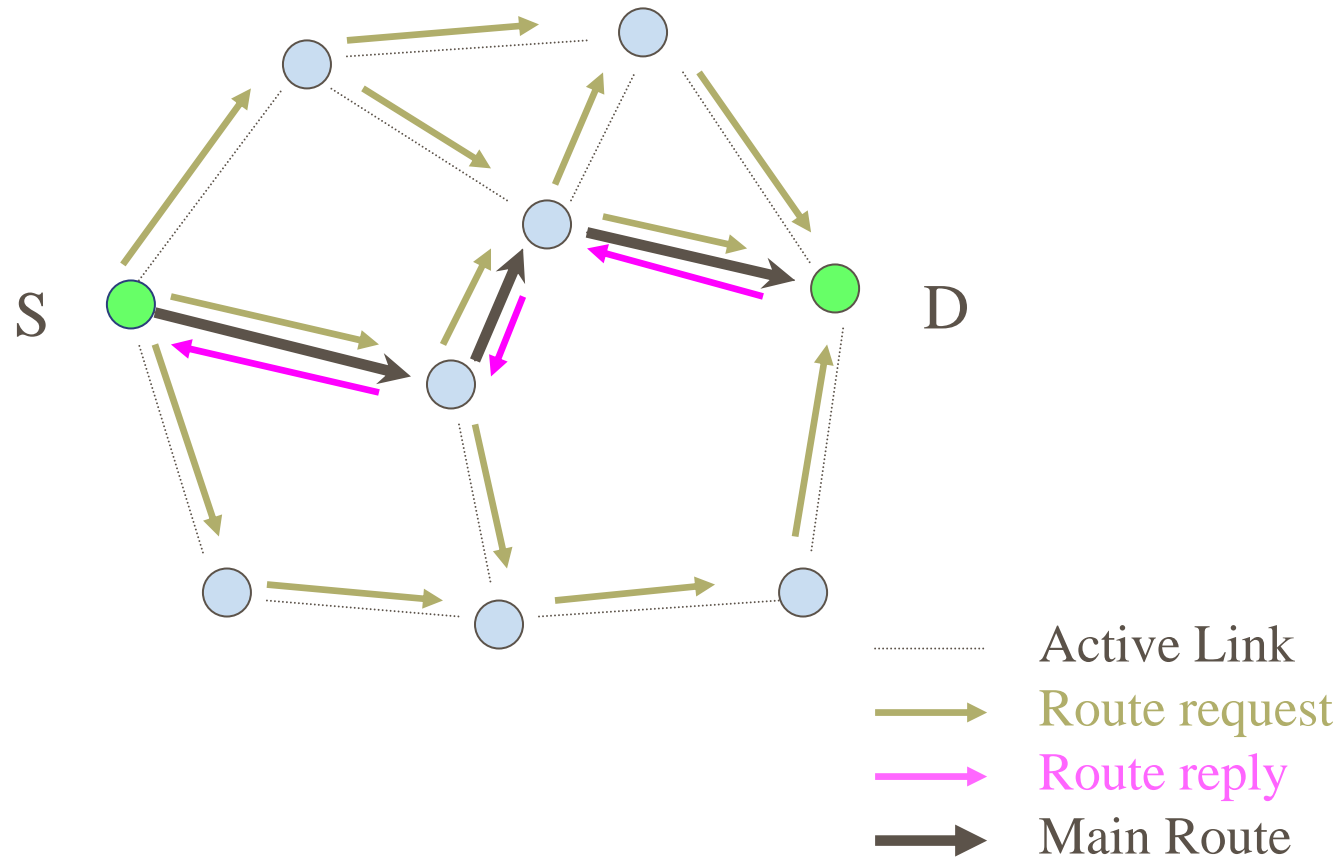
- **DSR**

(Dynamic source routing)

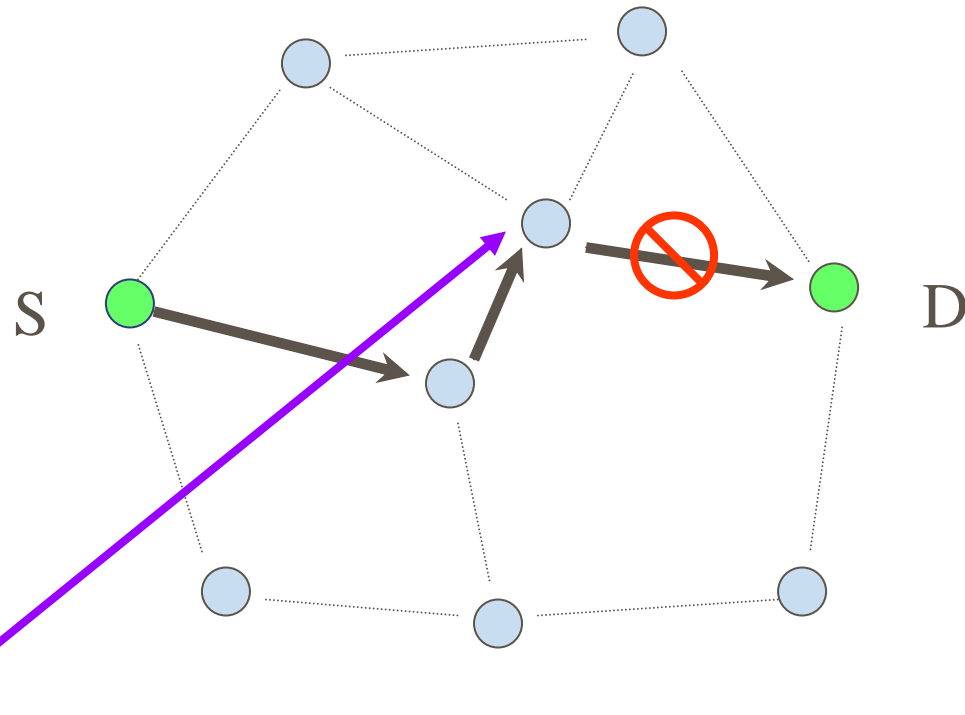
- **TORA**

(Temporally ordered routing algorithm)

# AODV - Route Construction

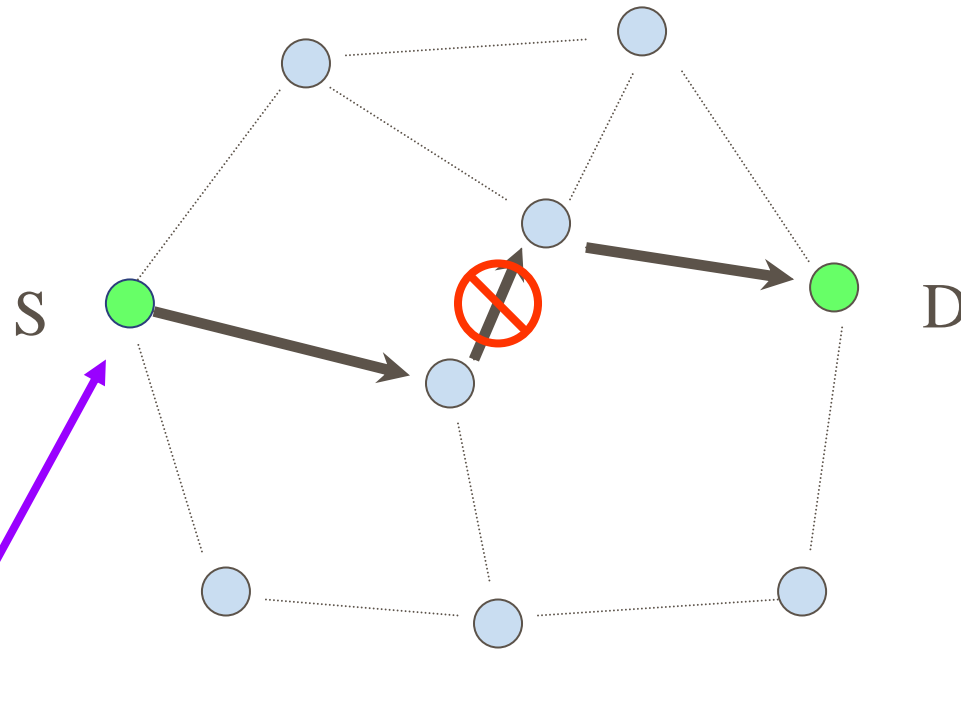


# AODV – Maintenance



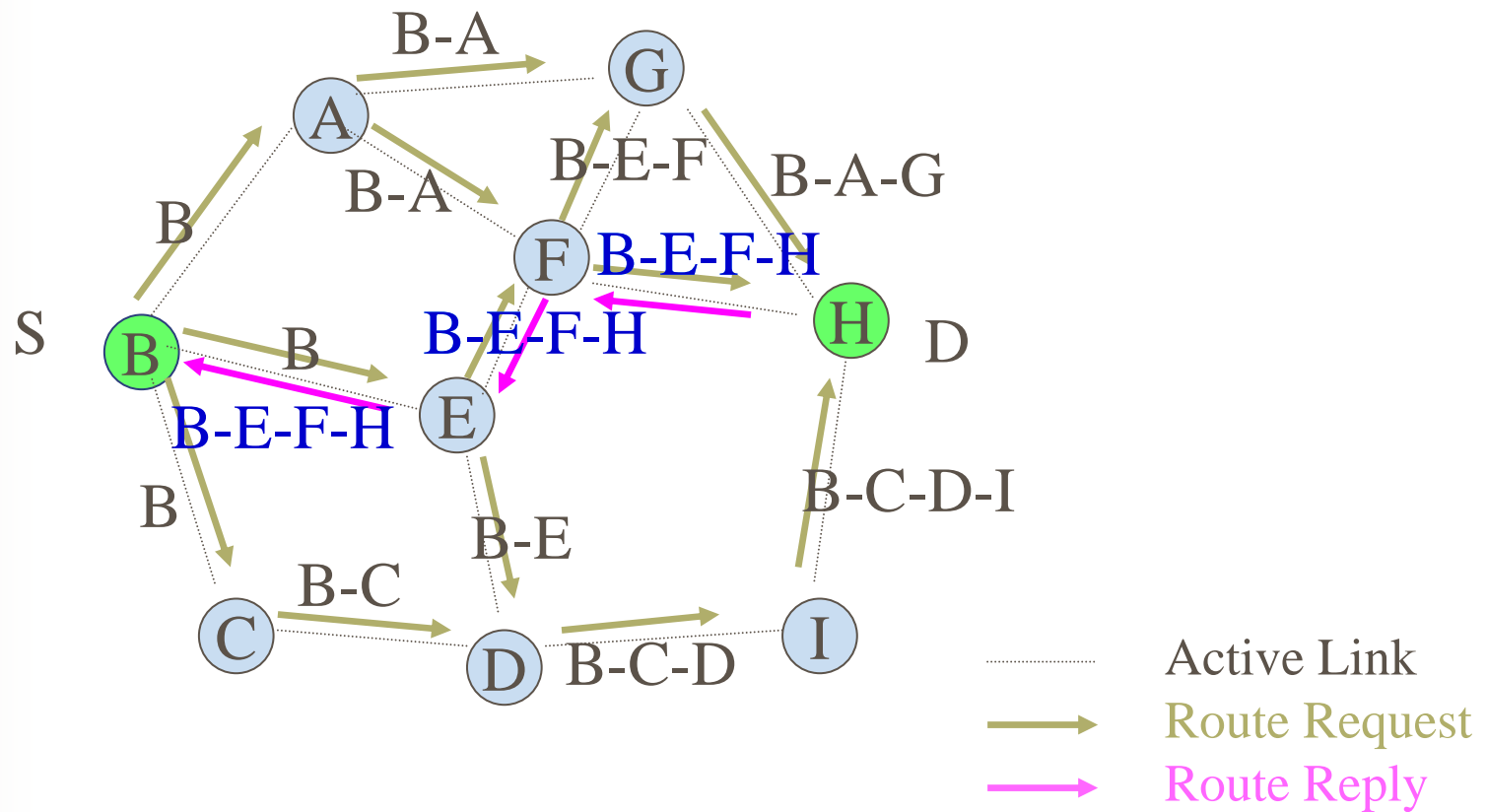
Flood **RREQ** to find D or an intermediate node to D

## AODV – Maintenance (cont.)

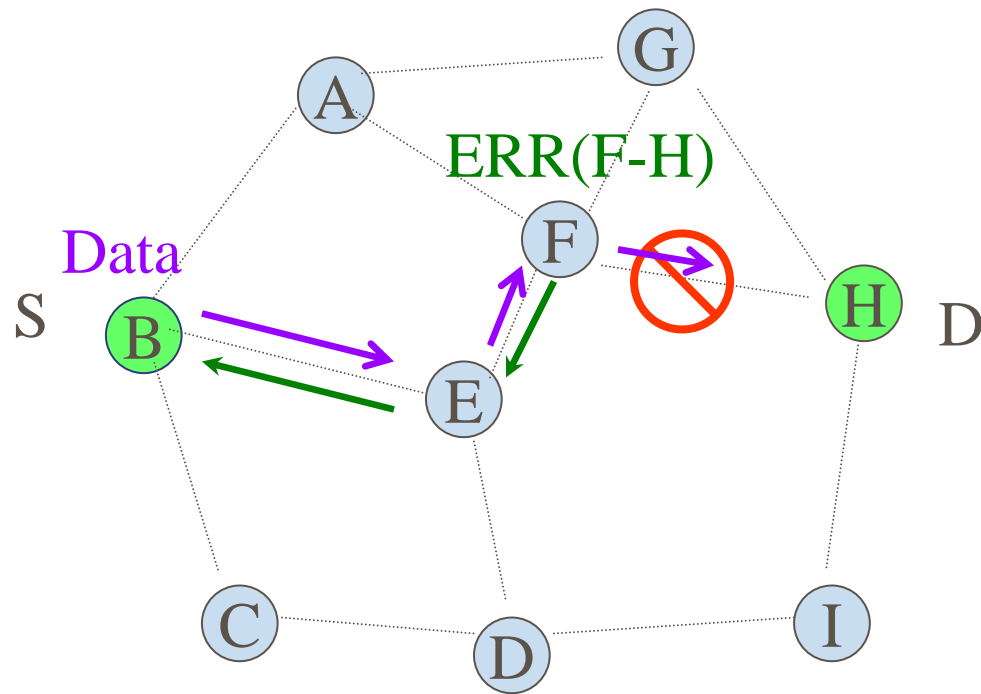


Flood **RREQ** to find D or an intermediate node to D

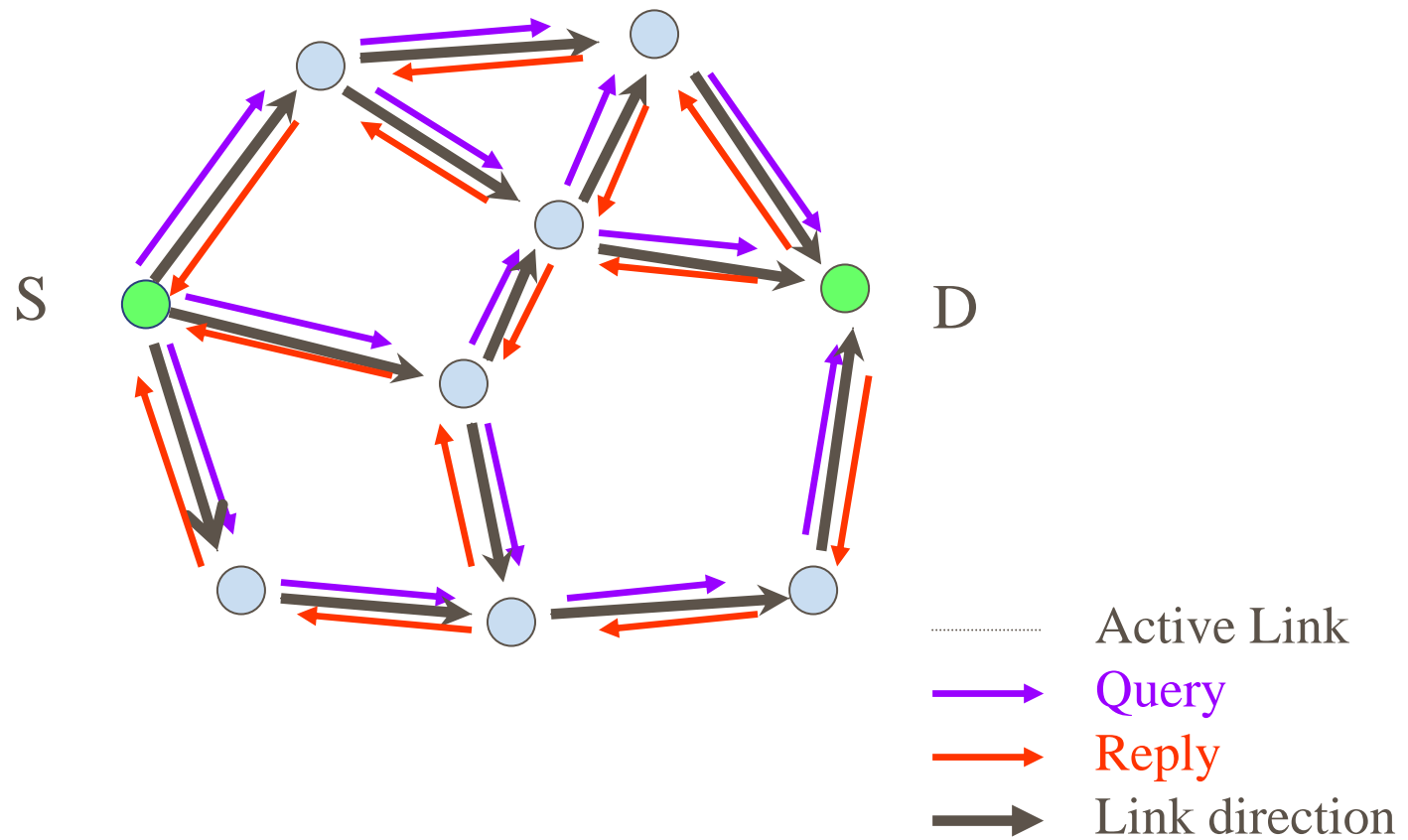
# DSR – Route Construction



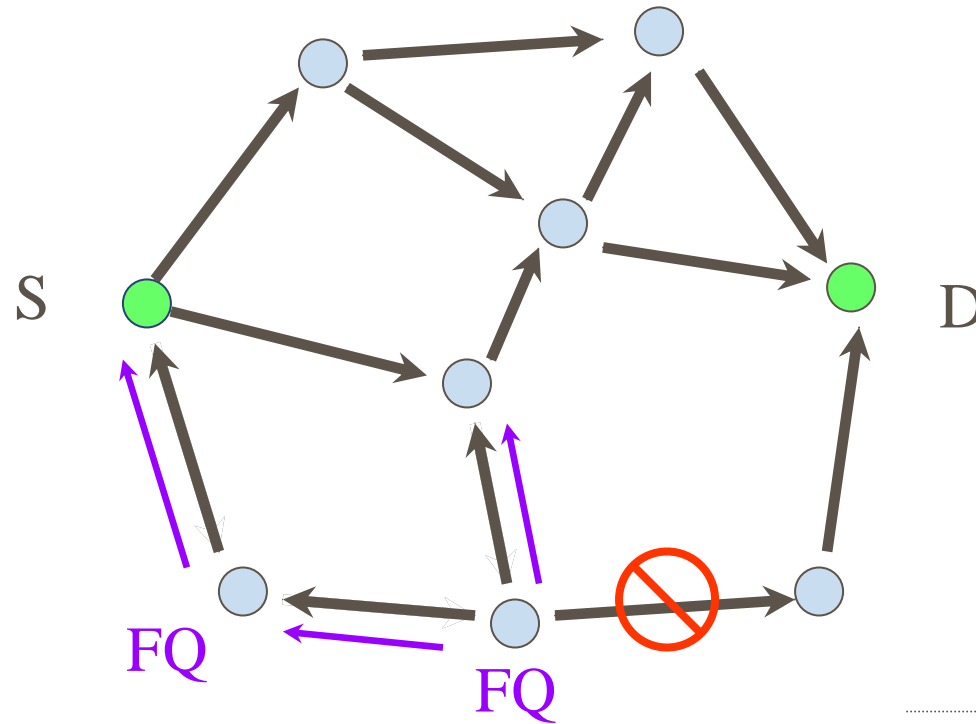
# DSR - Maintenance



# TORA – Route Construction



# TORA - Maintenance



FQ = Failure Query

- ..... Active Link
- Failure Query
- Link direction