Modeling of data networks by example: NS-2 (III) Wireless Networks

Holger Füßler, Hannes Hartenstein

Course overview

1. Introduction
2. Building block: RNG
3. Building block: Generating random variates I and modeling examples
4. Building block: Generating random variates II and modeling examples
5. Algorithmics: Management of events
6. NS-2: Introduction
7. NS-2: Fixed networks
8. NS-2: Wireless networks
9. Output analysis: single system
10. Output analysis: comparing different configurations
11. Omnet++ / OPNET
12. Simulation lifecycle, summary

Lecture overview

Retrace and understand a typical use case for simulation of (wireless) computer networks

- **Part I: Basics on Mobile Ad-Hoc Network Routing**
- **Part II: Ns-2 and the cmu wireless extensions**
- **Part III: Ns-2 scripts for MANET simulation**
- **Part IV: Output of MANET simulations**
  - **the CMU trace file**
  - **vizualization and analysis**

Lecture overview

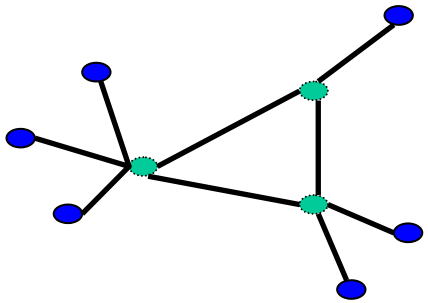
Retrace and understand a typical use case for simulation of (wireless) computer networks

- **Part I: Basics on Mobile Ad-Hoc Network Routing**
- **Part II: Ns-2 and the cmu wireless extensions**
- **Part III: Ns-2 scripts for MANET simulation**
- **Part IV: Output of MANET simulations**
  - **the CMU trace file**
  - **vizualization and analysis**

Mobile Ad-Hoc Networking

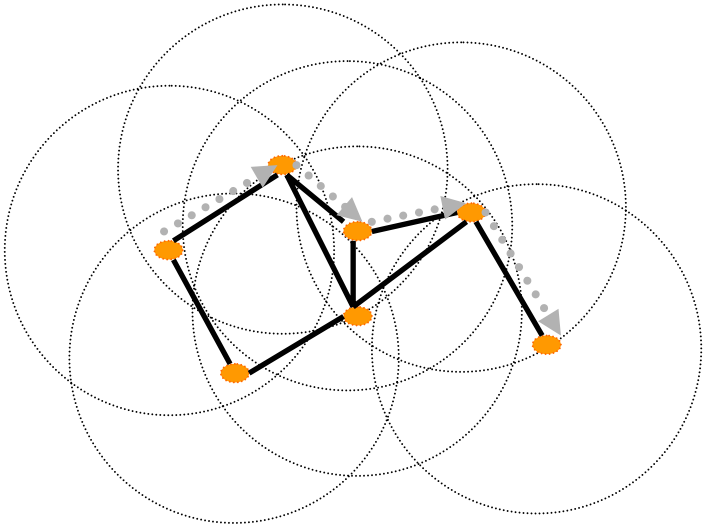
Characterization:

- All (network) nodes are equipped with radio technology
- nodes can be mobile
- Networks are forming up spontaneously
- Nodes are routers *and* end-systems



End-System

Router



MANET-node

nom. radio range

Mobile Ad-Hoc Network Routing

- Routing is the „classical“ MANET problem
  - links often unidirectional
  - radio range unstable
  - strict energy constraints
  - mobility- /radio-induced topology changes
- But also:
  - link / physical layer research
  - cooperativeness
  - security
  - transport layer (TCP ceases to work over wireless links)

MANET example 1: Sensor networks

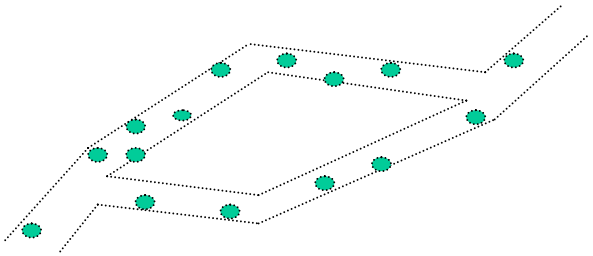
- usually high node density
- but also low mobility
- dominating node property is the availability of sensors (light/heat/movement)
- single nodes are highly restricted on energy and computation power

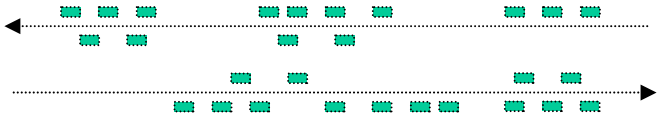
Embedded Sensor Board, FU Berlin (http://www.scatterweb.de)

MANET example 2: Vehicular Ad-Hoc Networks

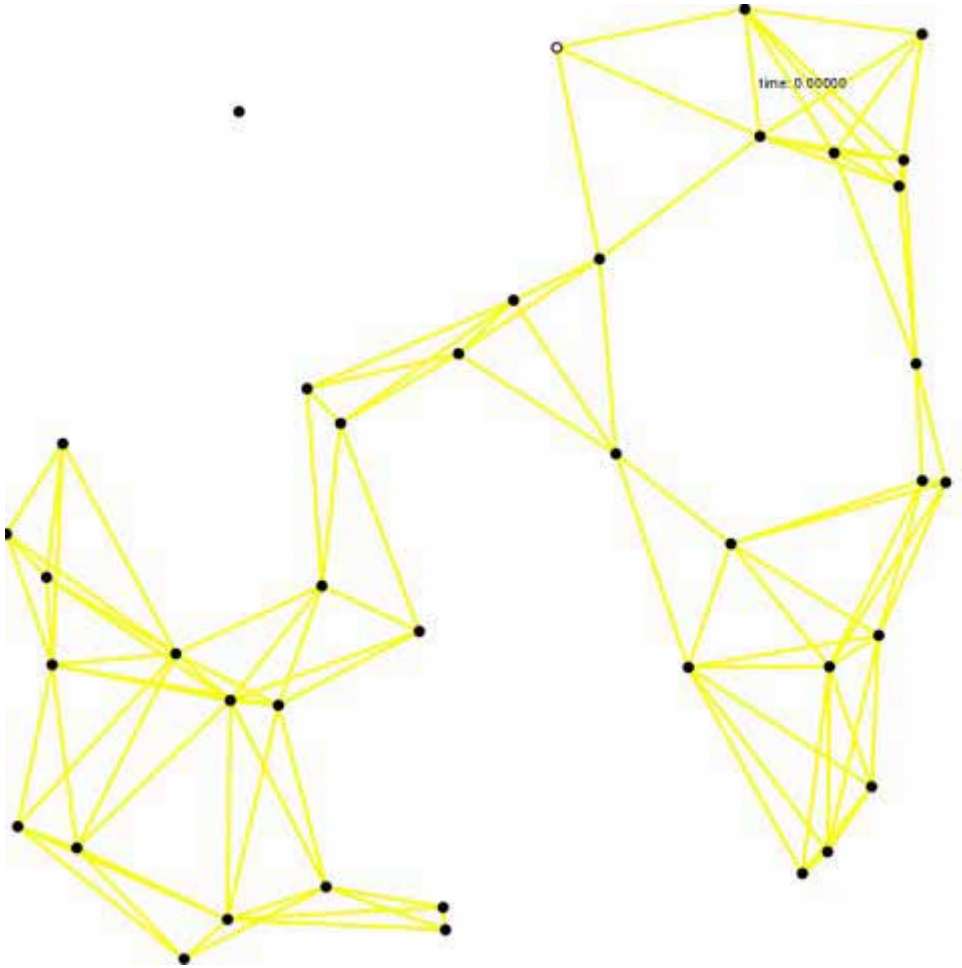
FleetNet

- Ad-Hoc Networks between street-bound cars based on ad-hoc network principles (see http://www.fleetnet.de)
- additional sensors (position / movement / possibly radar)
- very high mobility
- almost no energy restrictions

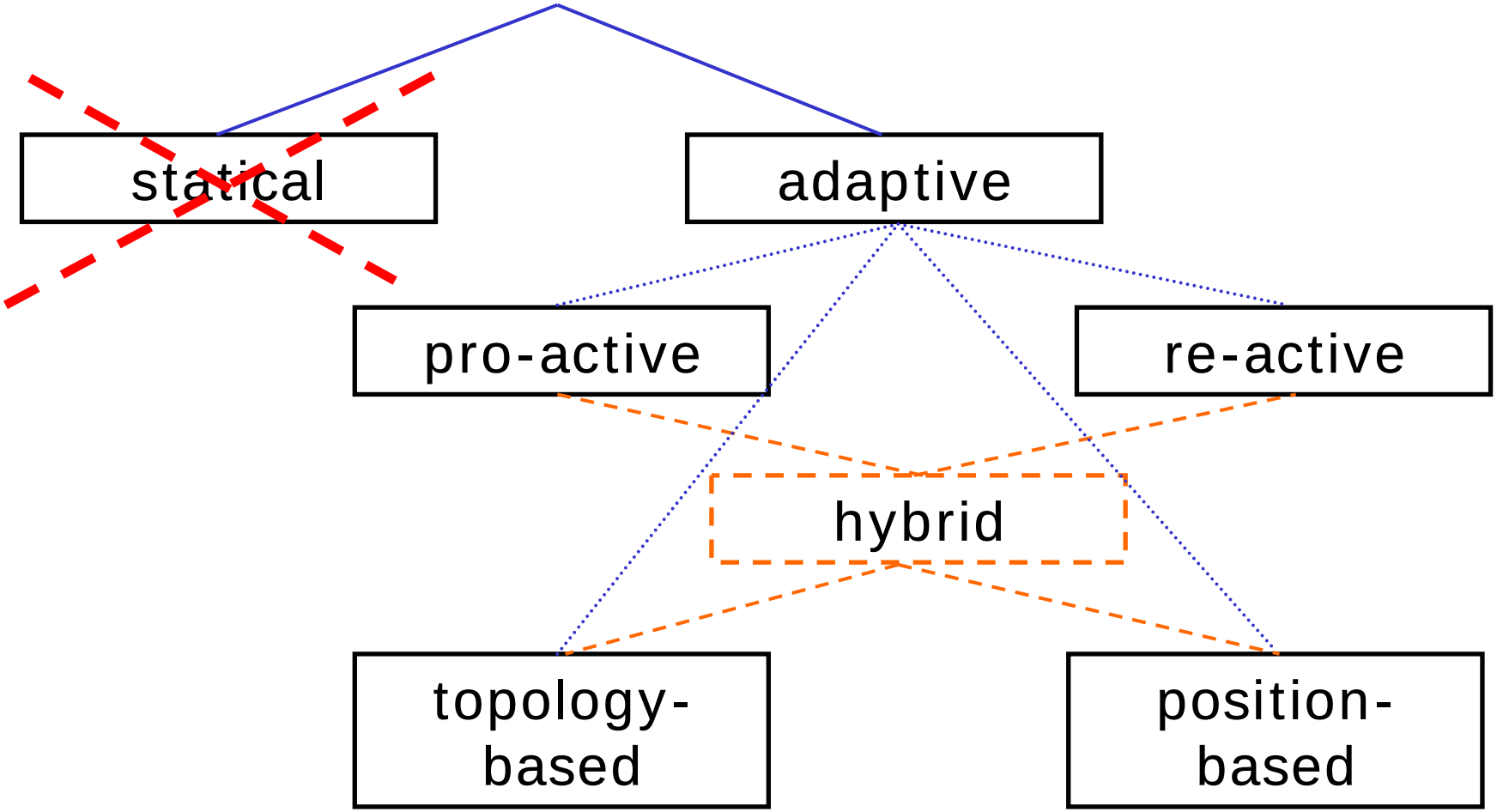




Node Mobility – Why is routing hard?



Classification of Routing Algorithms



Proactive vs. Reactive

- Proactive Algorithms:
  - all nodes permanently keep routes to all other nodes
- Reactive Algorithms:
  - only when communication is desired, the nodes „build up“ a routing information

- Discussion:
  - proactive is more suited for „equally distributed“ communication
  - But: Above a certain mobility rate, pro-active routing will fully load the network

Classification of Routing Algorithms

- Topology-Based Algorithms:
  - a (distributed) network topology is built based on the neighborhood relationship between nodes
  - the actual routing is done by
    - Source Routing
    - Distance-Vector Routing
    - Shortest-Path Routing (Link-State)
- Position-Based Algorithms:
  - Routing is done mainly in a greedy way minimizing the remaining distance to the destination
  - only possible, when nodes know about their current position
  - if no „greedy route“ is found, a recovery strategy is used

The IETF MANET group

- Topology-Based Algorithms:
  - reactive
    - AODV (Ad-Hoc On-Demand Distance Vector Routing)
    - DSR (Dynamic Source Routing)
  - proactive
    - TBRPF (Truncated Reverse-Path Broadcasting)
    - OLSR (Optimized Link-State Routing)
- Position-Based Algorithms:
  - none (so far ;-))

Classification of Routing Algorithms

- Perkins / Royer 1999 (download available)
- Re-Active Routing Method (i.e. on-demand)
- Topology-Based Distance-Vector Routing
- meanwhile RFC status (experimental RFC 3561)

AODV operation

- Route Requests (RREQ) are flooded on-demand

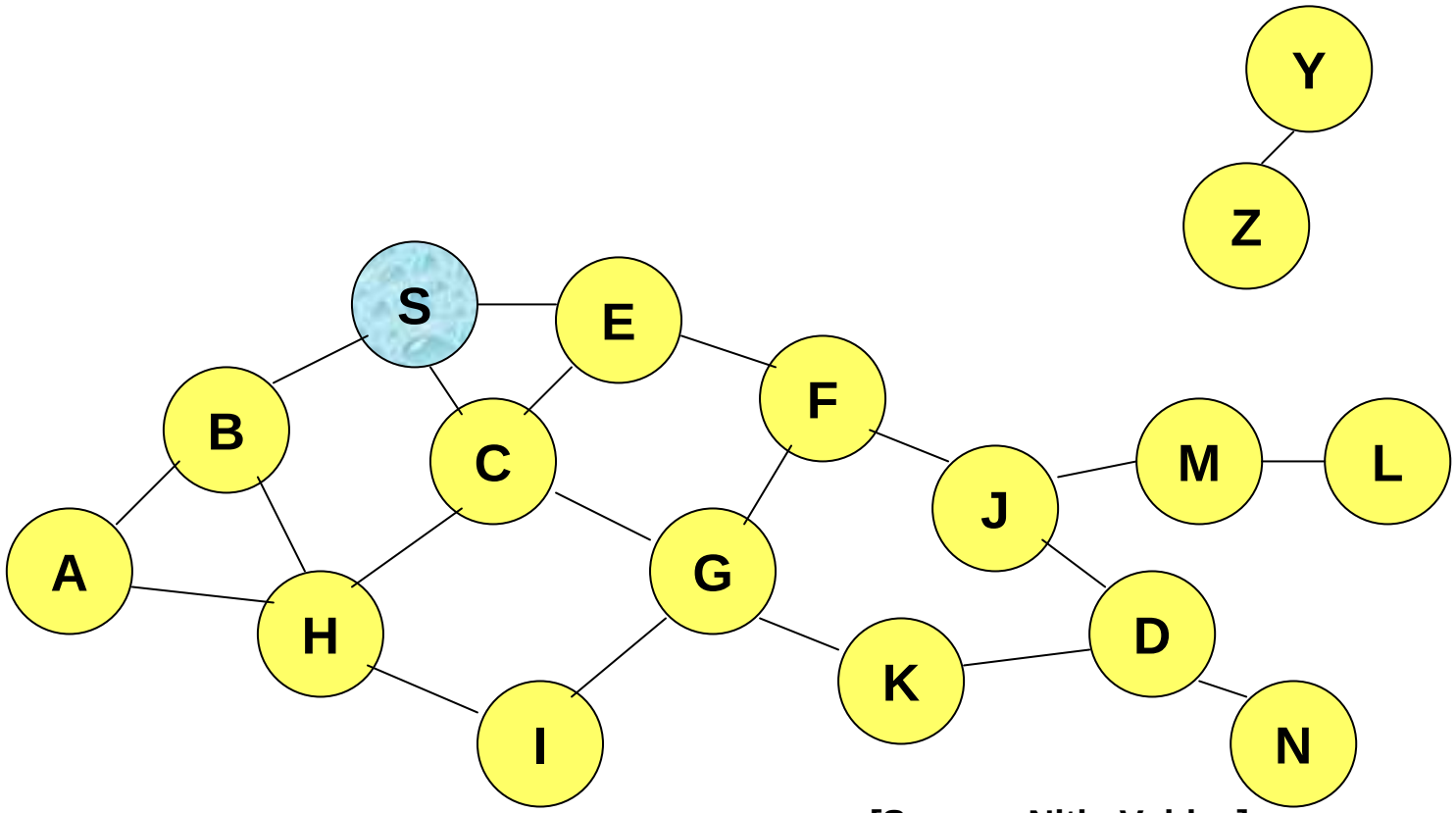
- When a node re-broadcasts a Route Request, it sets up a reverse path pointing towards the source
  - AODV assumes symmetric (bi-directional) links

- When the intended destination receives a Route Request, it replies by sending a Route Reply

- Route Reply travels along the reverse path set-up when Route Request is forwarded

[Source: Nitin Vaidya]

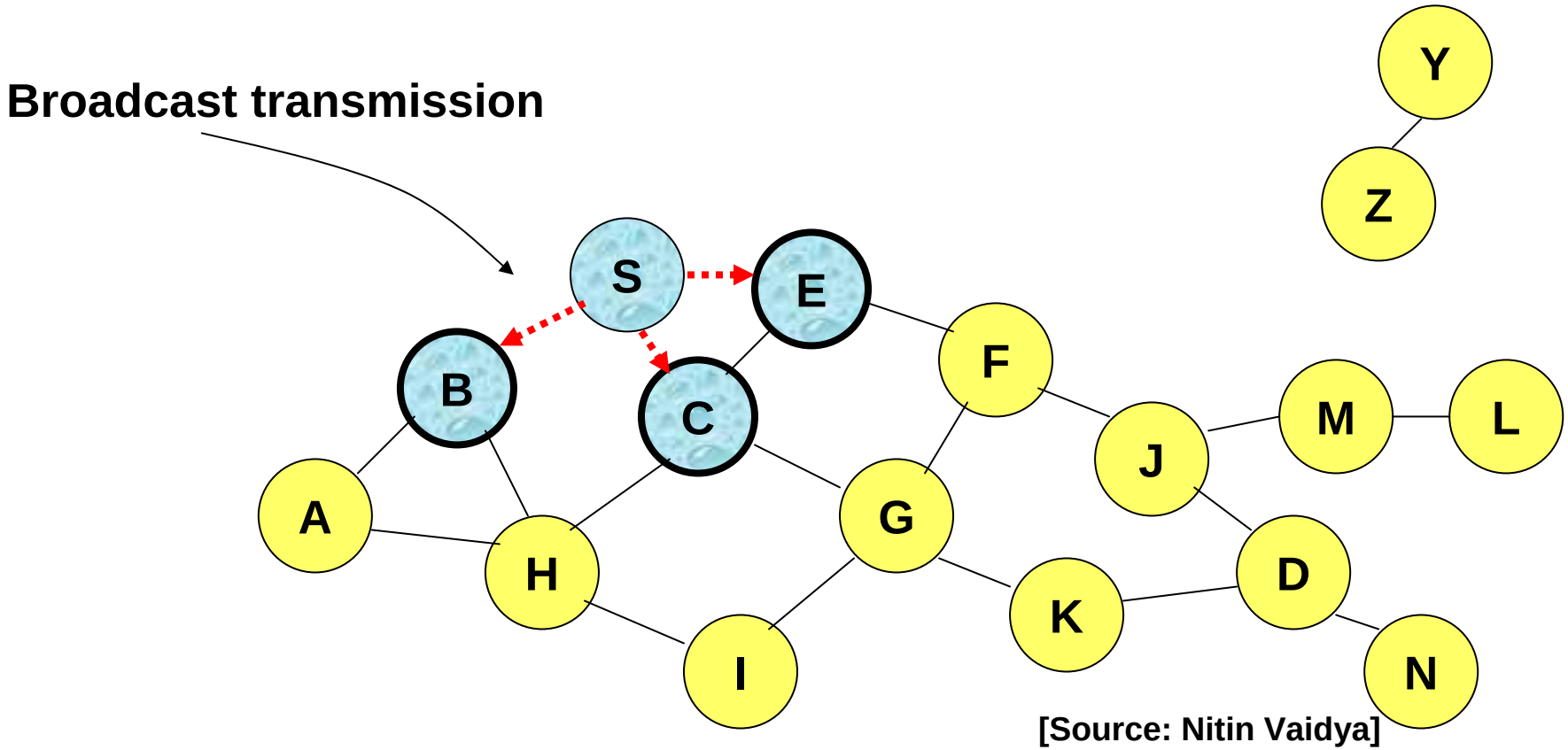
Route Requests in AODV



[Source: Nitin Vaidya]

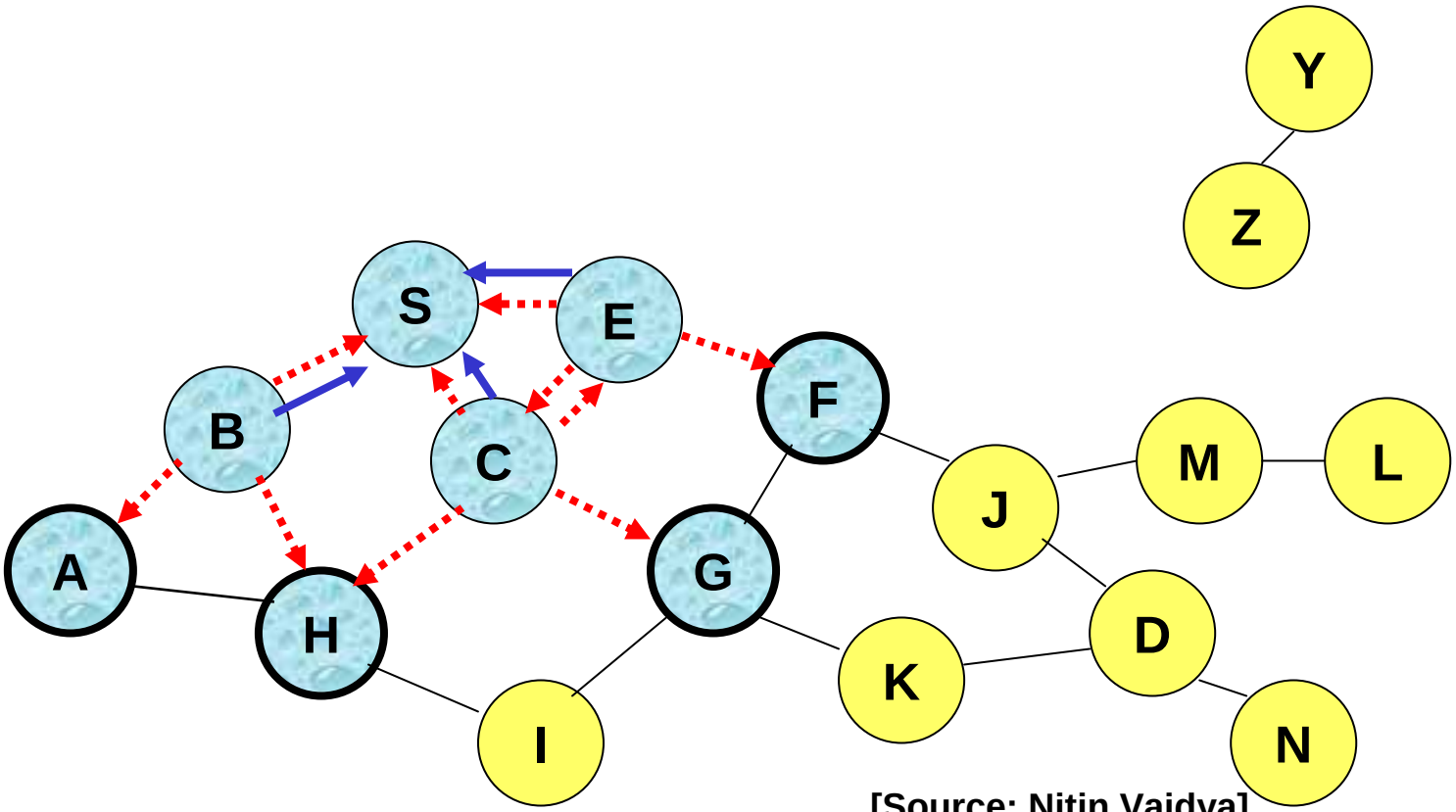
Represents a node that has received RREQ for D from S

Route Requests in AODV



Represents transmission of RREQ

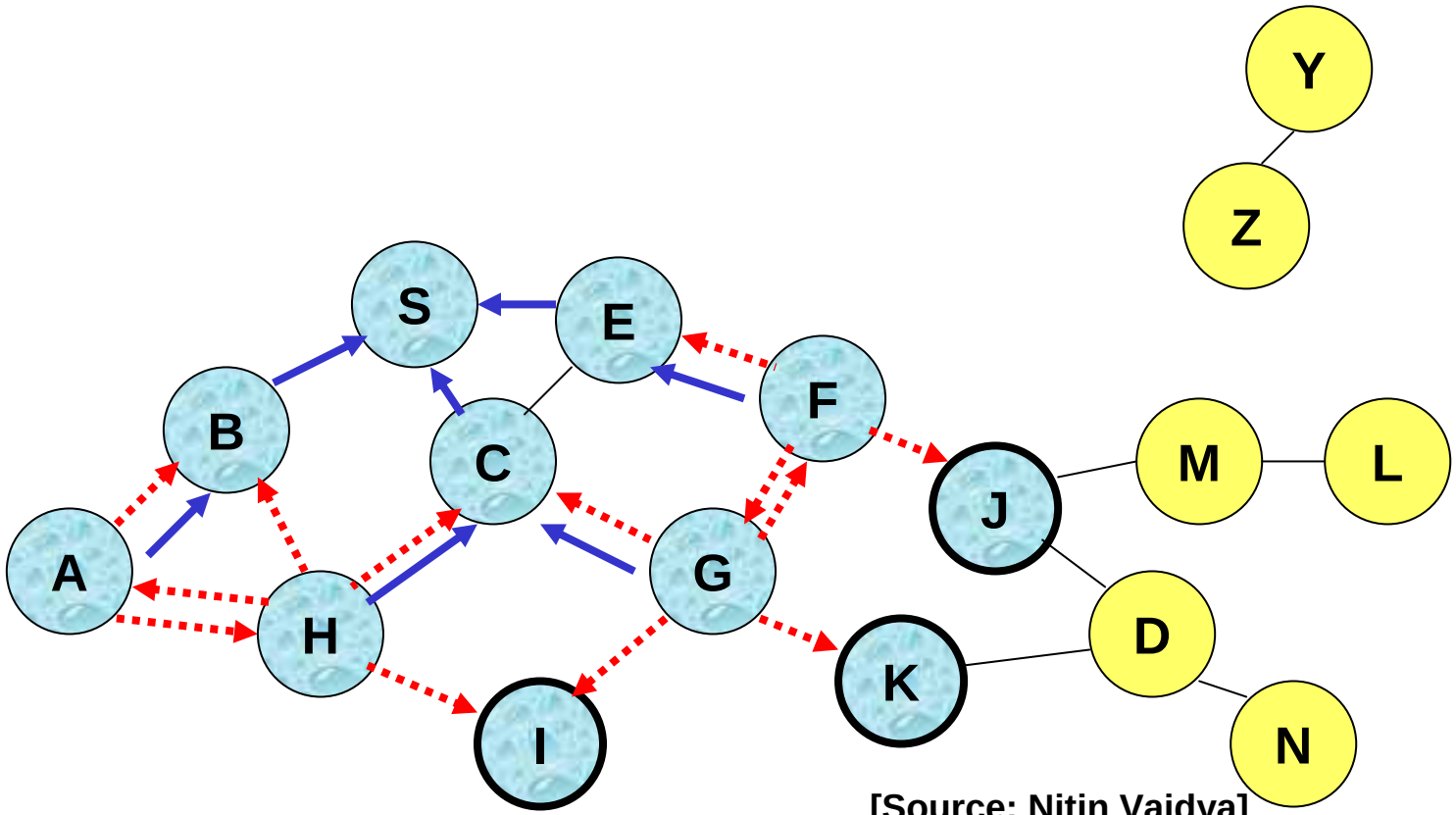
Route Requests in AODV



[Source: Nitin Vaidya]

Represents links on Reverse Path

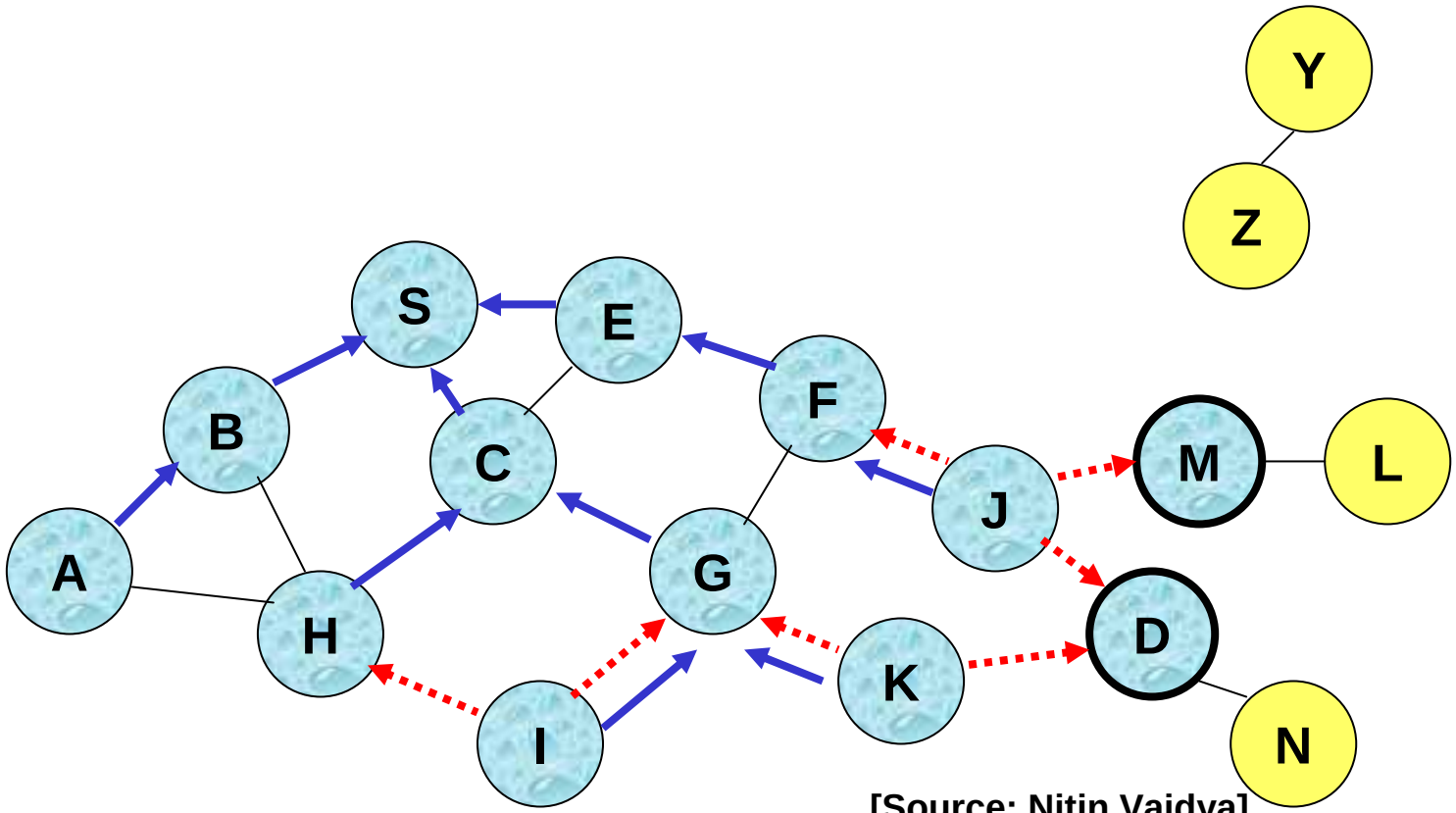
Reverse Path Setup in AODV



[Source: Nitin Vaidya]

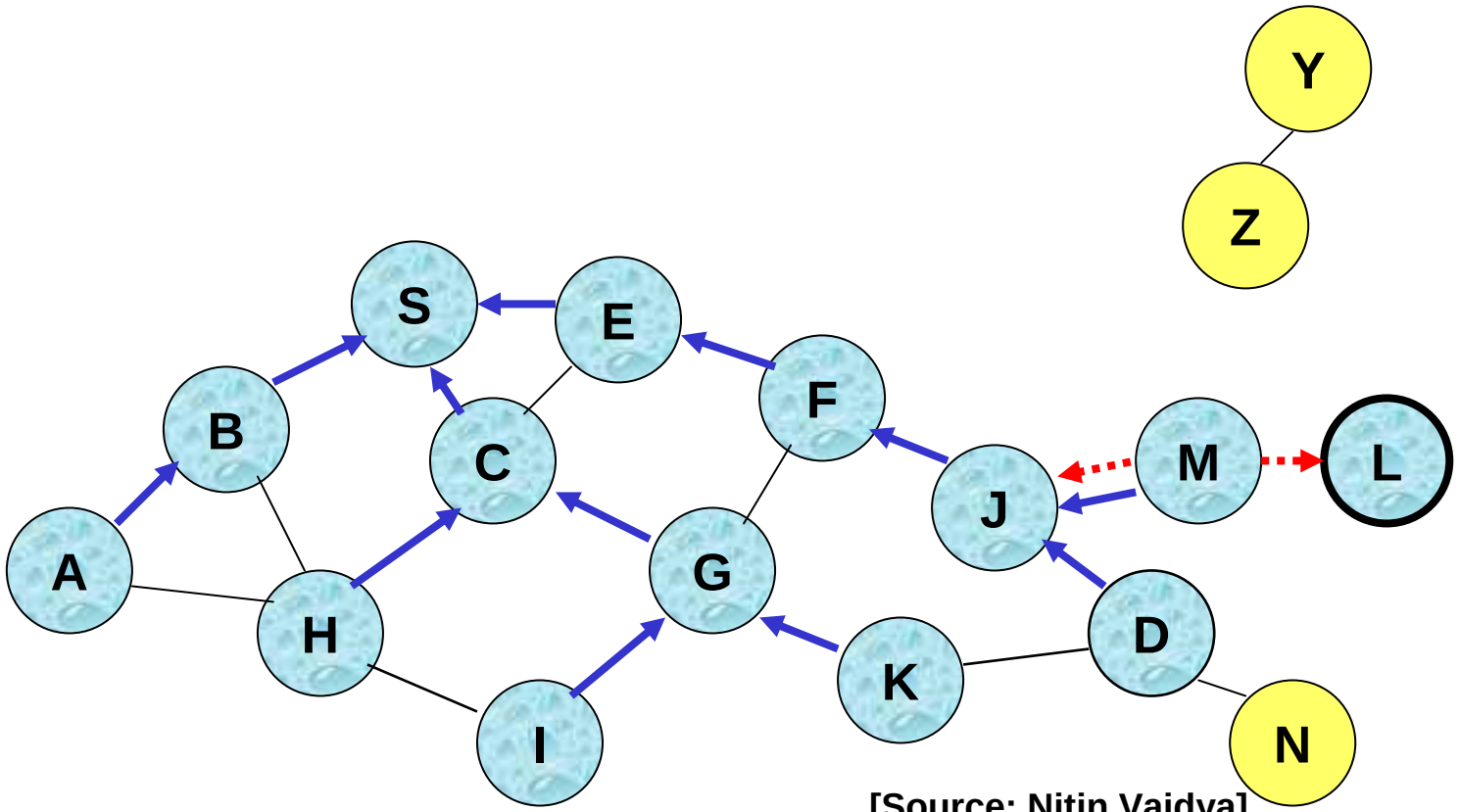
- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

Reverse Path Setup in AODV



[Source: Nitin Vaidya]

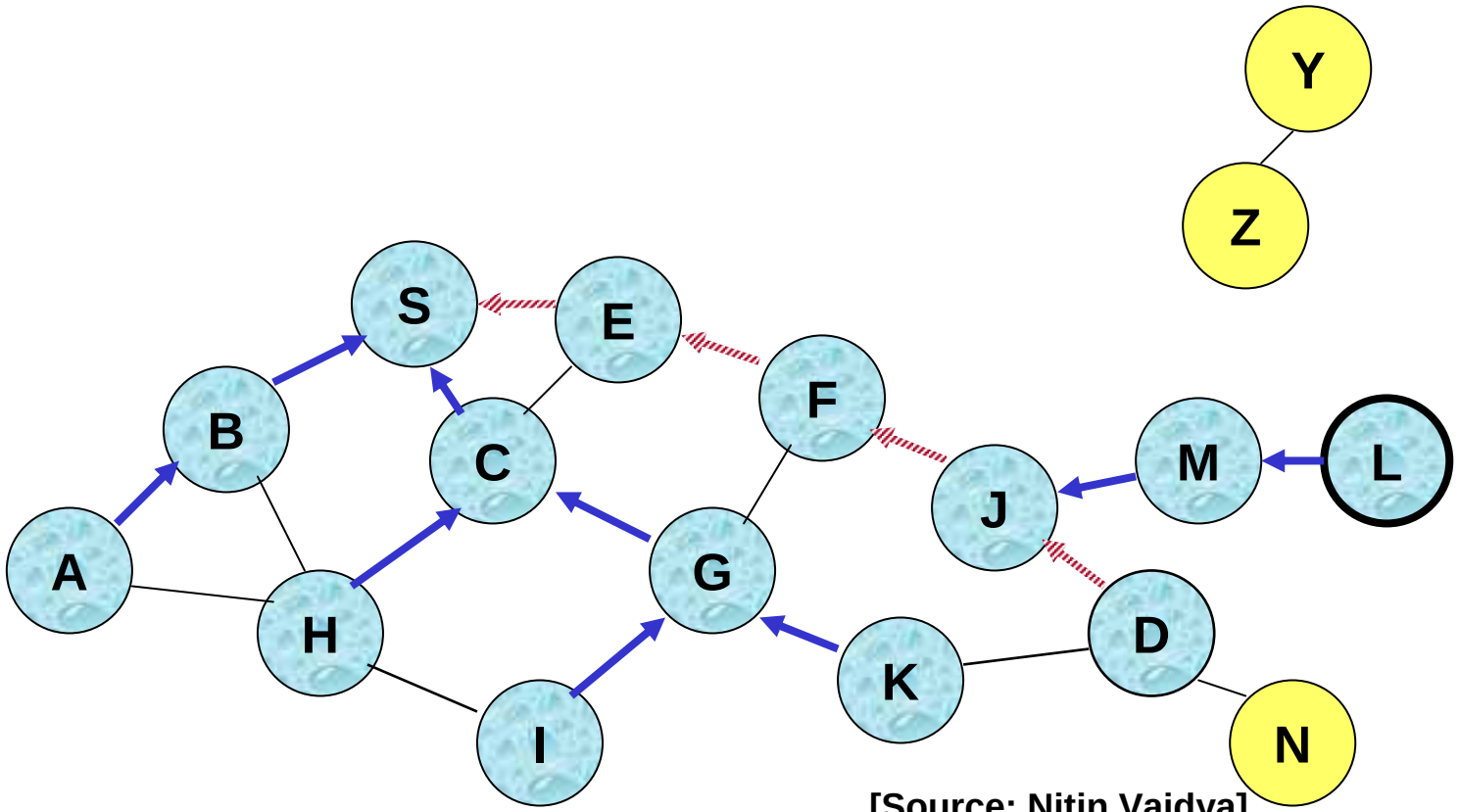
Reverse Path Setup in AODV



[Source: Nitin Vaidya]

- Node D does not forward RREQ, because node D is the intended target of the RREQ

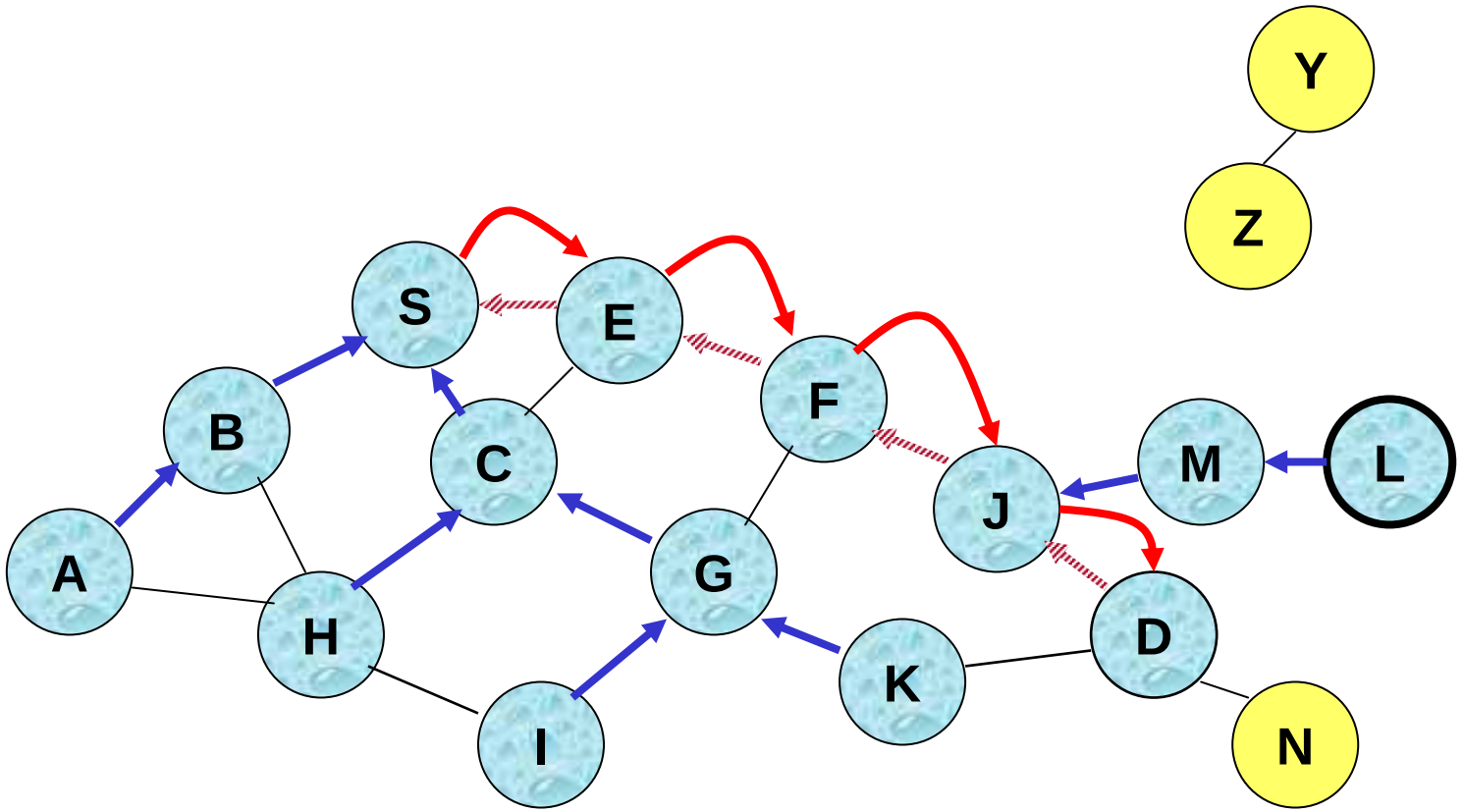
Route Reply in AODV



[Source: Nitin Vaidya]

Represents links on path taken by RREP

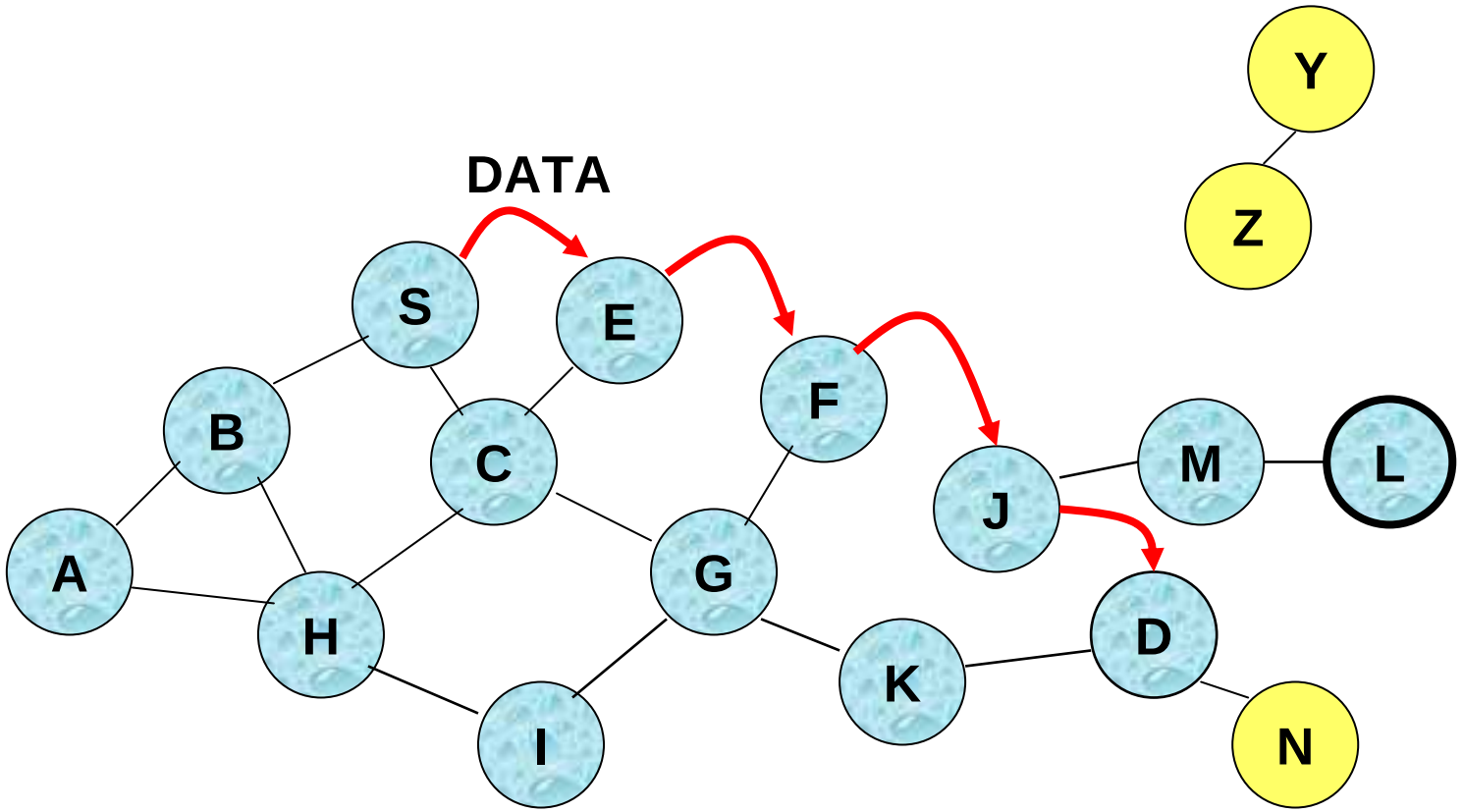
Forward Path Setup in AODV



Forward links are setup when RREP travels along the reverse path

Represents a link on the forward path

Data Delivery in AODV



Routing table entries used to forward data packet.
Route is *not* included in packet header.

Additional Comments

- AODV is (in reality) much more complicated
- sequence-number mechanism, among other things for count-to-infinity protection
- lots of protocol parameters (time-out values, link-layer notification, backward path setup via flooding)

Lecture overview

Retrace and understand a typical use case for simulation of (wireless) computer networks

- **Part I: Basics on Mobile Ad-Hoc Network Routing**
- **Part II: Ns-2 and the CMU wireless extensions**
- **Part III: Ns-2 scripts for MANET simulation**
- **Part IV: Output of MANET simulations**
  - **the CMU trace file**
  - **vizualization and analysis**

NS-2's Wireless Extensions

- Originally, ns-2 has no support for wireless networks
- CMU monarch wireless extensions (1998) (http://www.monarch.cs.cmu.edu/cmu-ns.html)
  - mobile nodes with programmable trajectories
  - IEEE 802.11 DCF MAC protocol
  - ARP / DSR / DSDV / TORA
  - wireless networking (Lucent WaveLan DSSS radio)
  - two ray ground radio propagation
  - utility scripts (movement, analysis, vizualization)
  - ...
- already included in actual ns-2 releases

ns-2 wireless node

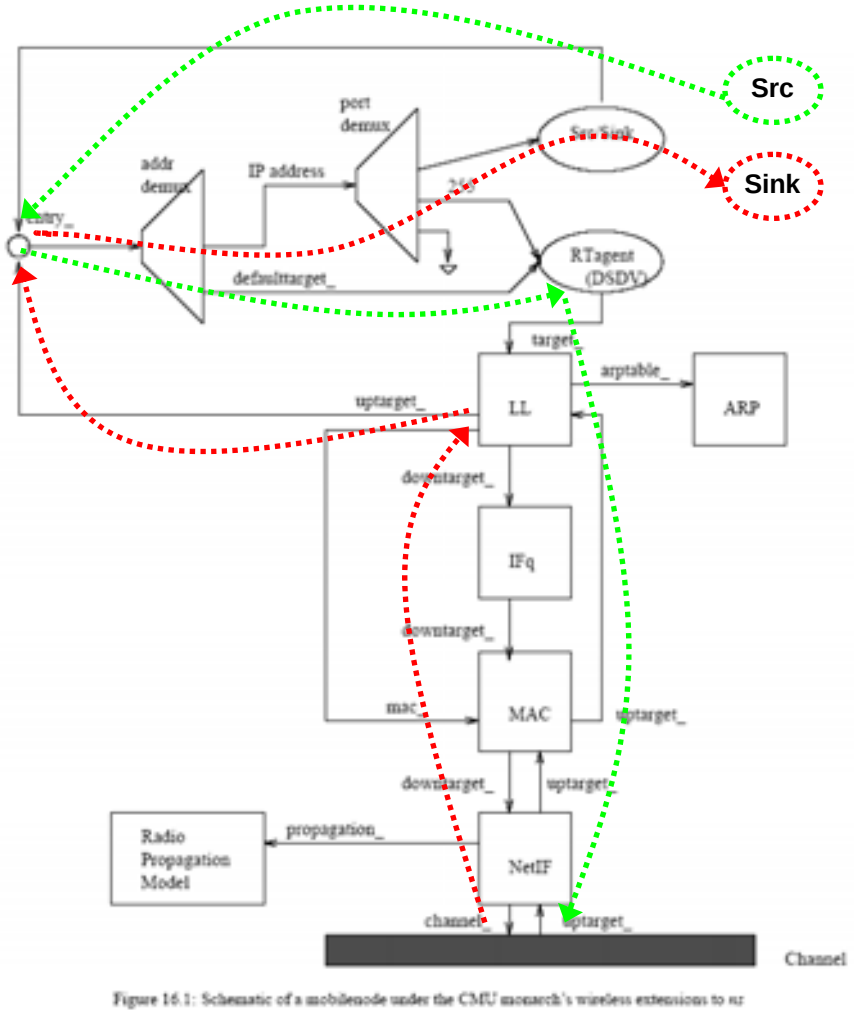


Figure 16.1: Schematic of a mobilenode under the CMU monarch's wireless extensions to ns

1. **Packet injected**
2. **non-local → RTAgent (e.g. AODV)**
   - **do route request if necessary**
   - **drop packet or select next hop (add to packet header)**
3. **hand packet to link layer**
   - **do ARP if necessary (IP)**
4. **hand packet to interface queue**
5. **MAC: get packets one-by-one**
   - **perform Media Access**
6. **Radio Propagation Model represents radio characteristics**
7. **NetIf is interface to channel**
   - **knows who is sending and if this is jamming my transmission**
8. **Reception of Packets**

Lecture overview

Retrace and understand a typical use case for simulation of (wireless) computer networks

- **Part I: Basics on Mobile Ad-Hoc Network Routing**
- **Part II: Ns-2 and the CMU wireless extensions**
- **Part III: Ns-2 scripts for MANET simulation**
- **Part IV: Output of MANET simulations**
  - **the CMU trace file**
  - **visualization and analysis**

Wireless “scripting basics” – Part 1

Simulation Script

```
# Radio Stuff
set val(chan)    Channel/WirelessChannel
set val(prop)    Propagation/TwoRayGround
set val(netif)   Phy/WirelessPhy
set val(rr)      250.0
set val(mac)     Mac/802_11
set val(bw)      2.0e6
set val(ifq)     Queue/DropTail/PriQueue
set val(ll)      LL
set val(ant)     Antenna/OmniAntenna
set val(ifqlen)  50

# Basic Sim Setup
set val(nn)      11
set val(rp)      AODV
set val(x)       2000
set val(y)       300
set val(simtime) 30

God set rrange_ $val(rr)
Mac/802_11 set rrange_ $val(rr)
```

Simulation Script Cont‘d

```
# Initialize Global Variables
set ns_          [new Simulator]
set tracefd      [open aodv-static_line.tr w]
set god_         [create-god $val(nn)]

$ns_ trace-all $tracefd
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
set channel_ [new $val(chan)]

$ns_ node-config -adhocRouting $val(rp) \
                 -llType $val(ll) \
                 -macType $val(mac) \
                 -ifqType $val(ifq) \
                 -ifqLen $val(ifqlen) \
                 -antType $val(ant) \
                 -phyType $val(netif) \
                 -topoInstance $topo \
                 -agentTrace ON \
                 -routerTrace ON \
                 -macTrace ON \
                 -movementTrace ON \
                 -channel $channel_ \
                 -propType $val(prop)
```

Wireless “scripting basics” – node-config options

option	available values	default
general		
addressType	flat, hierarchical	flat
MPLS	ON, OFF	OFF
both satellite- and wireless-oriented		
wiredRouting	ON, OFF	OFF
llType	LL, LL/Sat	""
macType	Mac/802_11, Mac/Csma/Ca, Mac/Sat, Mac/Sat/UnslottedAloha, Mac/Tdma	""
ifqType	Queue/DropTail, Queue/DropTail/PriQueue	""
phyType	Phy/WirelessPhy, Phy/Sat	""
wireless-oriented		
adhocRouting	DIFFUSION/RATE, DIFFUSION/PROB, DSDV, DSR, FLOODING, OMNIMCAST, AODV, TORA	""
propType	Propagation/TwoRayGround, Propagation/Shadowing	""
propInstance	Propagation/TwoRayGround, Propagation/Shadowing	""
antType	Antenna/OmniAntenna	""
channel	Channel/WirelessChannel, Channel/Sat	""
topoInstance	<topology file>	""
mobileIP	ON, OFF	OFF
energyModel	EnergyModel	""
initialEnergy	<value in Joules>	""
rxPower	<value in W>	""
txPower	<value in W>	""
idlePower	<value in W>	""
agentTrace	ON, OFF	OFF
routerTrace	ON, OFF	OFF
macTrace	ON, OFF	OFF
movementTrace	ON, OFF	OFF
errProc	UniformErrorProc	""
FECProc	?	?
toraDebug	ON, OFF	OFF
satellite-oriented		
satNodeType	polar, geo, terminal, geo-repeater	""
downlinkBW	<bandwidth value, e.g. "2Mb">	""

» **node-config options see ns-doc**

Wireless “scripting basics” – Part 2

Simulation Script Cont‘d

```
for {set i 0} {$i < $val(nn)} {incr i} {
      set node_($i) [$ns_ node]
      $node_($i) random-motion 0
}

# source the mvmnt pattern and the comm pattern
source move-static_line.tcl
source comm-static_line.tcl

for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at $val(simtime) "$node_($i) reset";
}

$ns_ at $val(simtime).2 "stop"
$ns_ at $val(simtime).21 "puts \"NS EXITING...\" ; $ns_
      halt"
proc stop {} {
    global ns_ tracefd aggrfd
    $ns_ flush-trace
    close $tracefd
}

puts "Starting Simulation..."
$ns_ run
```

The Comm Pattern

```
set commsrcno 0
set commdstno 8


set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_($commsrcno) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_($commdstno) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 4.0
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 40
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 2.5 "$cbr_(0) start"
```

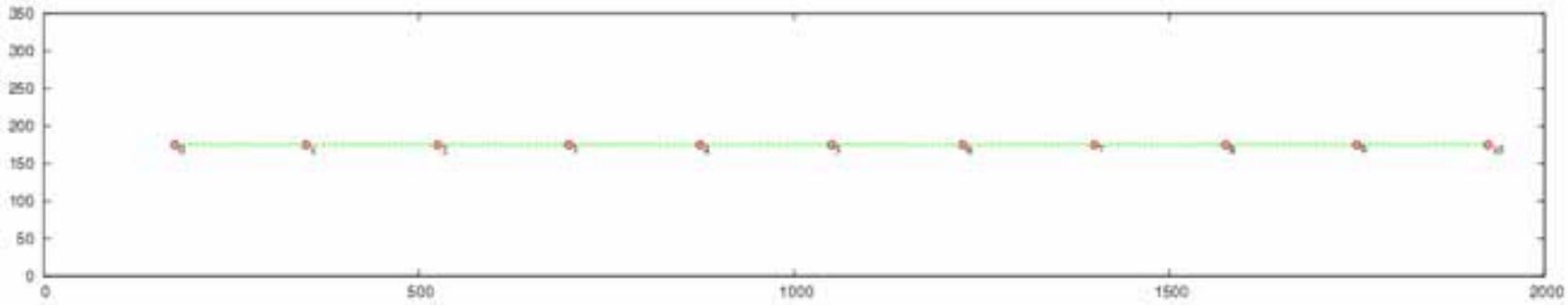
Wireless “scripting basics” – The “Movement Pattern”

move-static_line.tcl

```
$node_(0) set X_      175.00
$node_(0) set Y_      175.00
$node_(0) set Z_        0.00
$node_(1) set X_      350.00
$node_(1) set Y_      175.00
$node_(1) set Z_        0.00
. . .
```

move-static_line.tcl cont‘d

```
. . .
$node_(10) set X_    1925.00
$node_(10) set Y_     175.00
$node_(10) set Z_       0.00
```



Excursion: Real Movement

- **Real Movement, i.e. nodes changing position is supported as follows:**

  $ns at $time “$node setdest x2 y2 <speed>”

- **letting the node $node move from the position it holds at simulation time $time to x2 / y2 with <speed> m/s beginning at time $time**

- **Usually, a utility like “setdest” (in the indep-utils dir) is used to generate random waypoint traffic**

- **Or: Use real-live movements “converted” to ns-2 input**

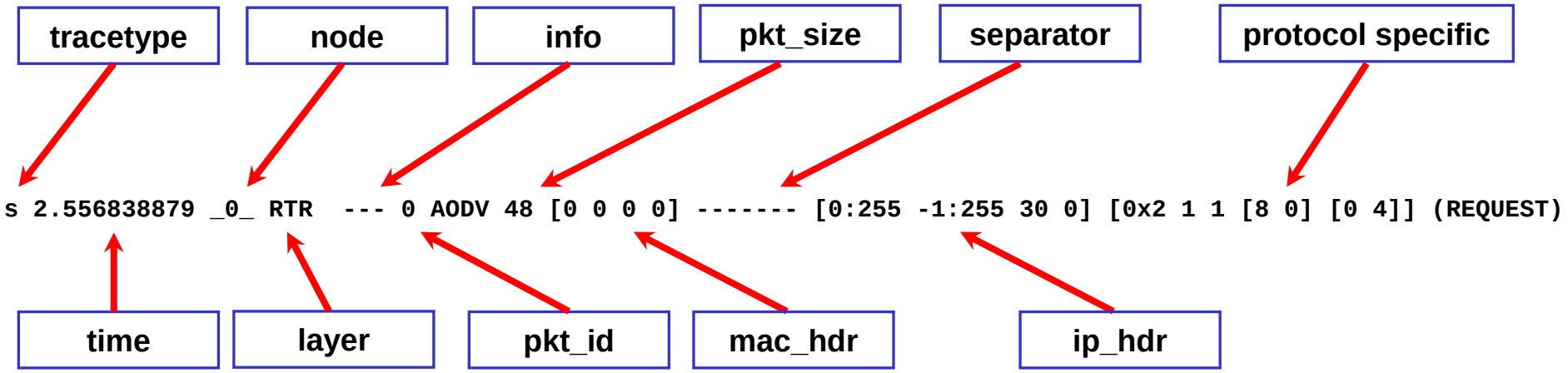
Lecture overview

Retrace and understand a typical use case for simulation of (wireless) computer networks

- **Part I: Basics on Mobile Ad-Hoc Network Routing**
- **Part II: Ns-2 and the CMU wireless extensions**
- **Part III: Ns-2 scripts for MANET simulation**
- **Part IV: Output of MANET simulations**
  - **the CMU trace file**
  - **vizualization and analysis**

Trace Output

- The content of the file is controlled by node-config
- Structure different from wired traces
- typical line:



Trace Output (Details)

- **Mac Header:**

  `[duration dst src pkt_type]`

  - **Duration: Only for RTS/CTS (Network Allocation Vector)**
  - **dst / src: MAC dst/src, for broadcast ffffffff**
  - **pkt_type: Packet Type of enclosed packet (800 for IP)**

- **IP Header:**

  `[src:port dst:port ttl next_hop]`

  - **src / dst: network address of node (end-2-end), -1 for broadcast**
  - **ports: port numbers (255 for routing)**
  - **ttl: time to live in hops**
  - **next_hop (from IPs point of view)**

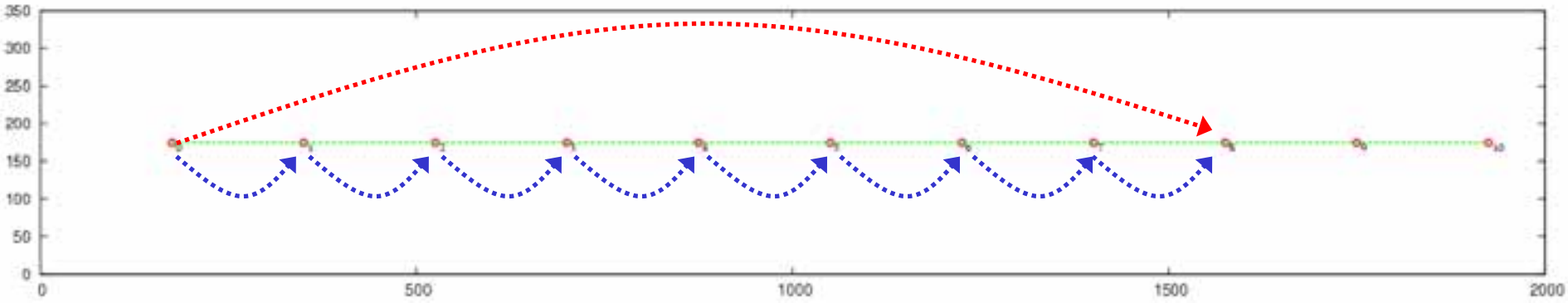
- **further details: “ns-2.27/trace/cmu-trace.(h|cc)”**

Tracing: Degrees of Freedom

- also: new trace format of form `-<valueDescription> value` activated with $ns new-trace
  - `Better to parse`
  - `Harder to read (for me [HMF])`
  - `Almost twice as Big, but bzip2 handles it pretty well`
- also: node position logging in every line
- also: mechanism for custom lines in trace file

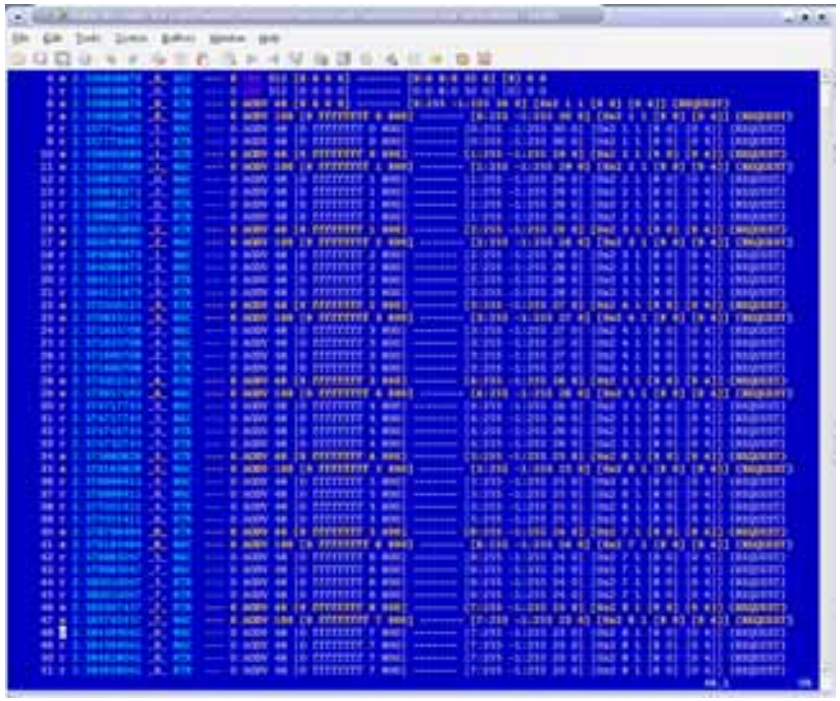
Example Scenario

- 11 nodes in a row
- Routing Protocol: AODV
- CBR traffic: node 0 with node 8



Example Trace

- **Open Trace File in Editor**
  - **AODV-011-static-0.tr (old trace format)**
  - **AODV-011-static-0.ntr (new trace format)**
- **Follow Route Request / Reply in scenario**



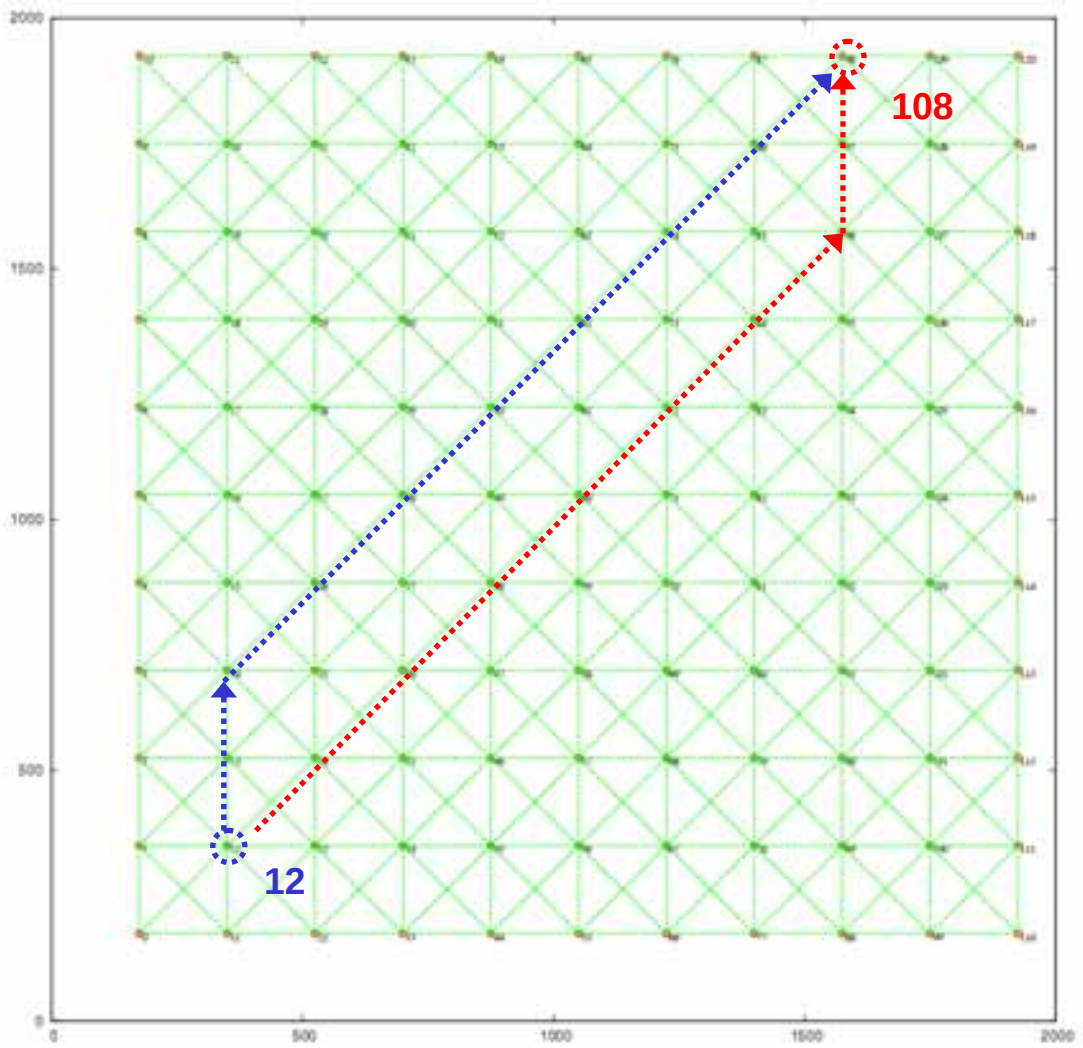
Evaluation

- **Typical metrics for evaluation of “Routing Protocol Performance” (see also RFC 2501):**
  - **Packet Delivery Ratio (PDR)**
  - **(avg) End-2-End Delay / Route Acquisition Time**
  - **Overhead / Cost**
    - **on Routing / MAC Layer**
    - **overall / per packet or payload bit**
- **Warning: Do not generate a wrong feeling of linearity!**
  - **Histogram or Function Plot (Interpolation)?**
  - **Absolute Values per Simulation Run / per Packet?**

Evaluation (End-2-End Delay Line Scenario)

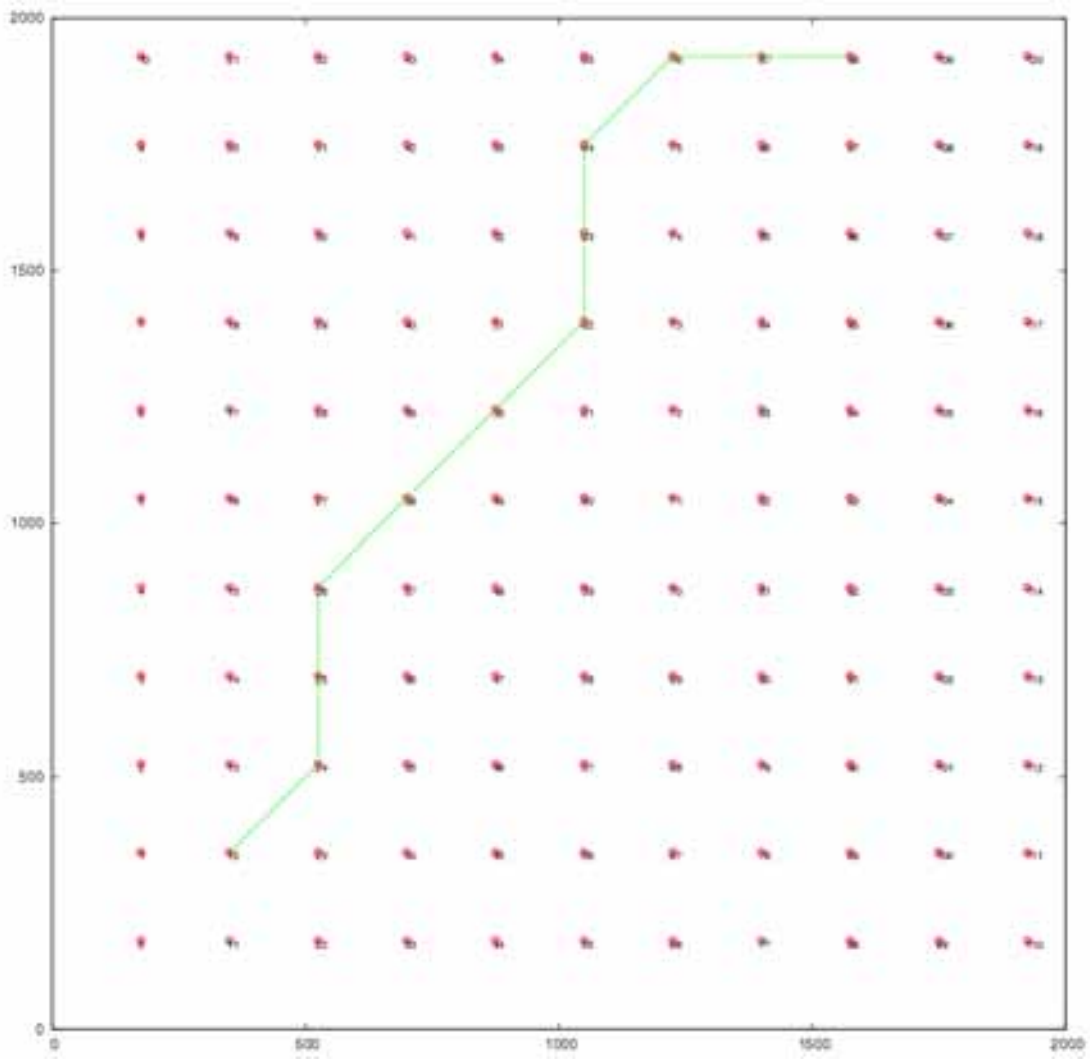
- Histogram shows the end-2-end-packet delay for the nine packets sent
- „First Packet Delay“ higher because of „Route Acquisition Time“
- Remaining Packets fairly stable

The Grid Scenario (Setup)



- 11 × 11 = 121 nodes (0..120)
- CBR: 12 → 108
- Which are the shortest routes to be expected?
- Hop Length: 9

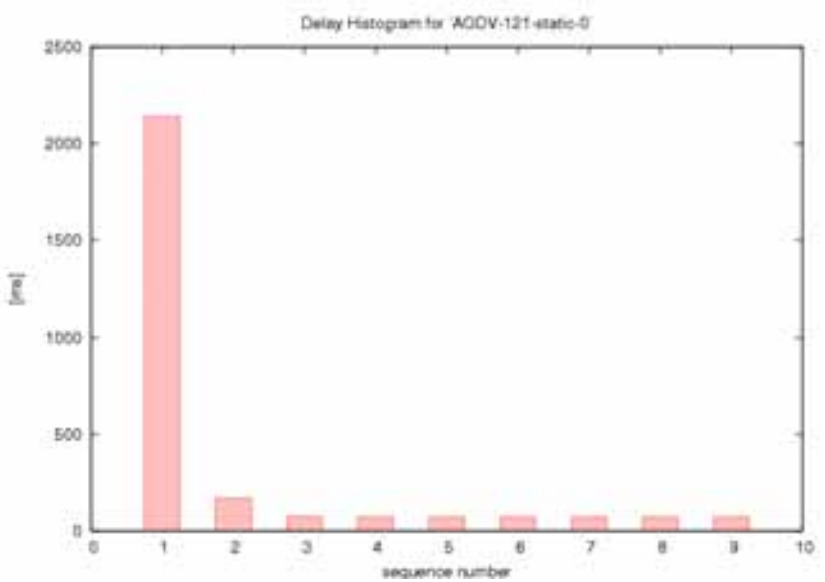
The Grid Scenario (Route Taken)



- **Why is that so?**
- **AODV immediately answers the RREQ, even if it‘s suboptimal**
- **Protocol Variant: Answer Each RREQ and improve route**

Evaluation (End-2-End Delay Grid Scenario)

- Histogram shows the end-2-end-packet delay for the nine packets sent
- „First Packet Delay“ higher because of „Route Acquisition Time“
- Remaining Packets fairly stable
- But: „First Packet Delay“ one order of magnitude higher than in line scenario



There is still more…

- play with downloadable examples
- parser scripts with different purposes
- have a look at the trace files

Wrap-Up

- Basics on Mobile Ad-Hoc Network Routing
  - Challenges
  - Classification of Algorithms
- Learned about the wireless extensions of ns-2
- Created simple AODV simulations
- Learned to read the wireless traces

References

- The ns-2 user manual http://www.isi.edu/nsnam/ns
- S. Corson, J. Macker: RFC 2501: Mobile Ad-Hoc Networking (MANET) Routing Protocol Performance Issues and Evaluation Considerations
- C. Perkins, E. Royer “Ad-Hoc On-Demand Distance-Vector Routing”

  In " Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA) ", pp. 90-100, New Orleans, LA, February 1999
- C.Perkins, E. Belding-Royer, Samir Das: RFC 3561: Ad-Hoc On-Demand Distance-Vector (AODV) Routing, IETF
- Samir R. Das, Charles E. Perkins, Elizabeth M. Royer, “Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks”, IEEE Infocom 2000, Tel Aviv, Israel, March 2000