Dept. of Mathematics "Tullio-Levi Civita"
Master's Degree in Computer Science

Wireless Networks - A.Y. 2018-2019
04/12/2018

BLR AND ROFF:

TWO ROUTING ALGORITHMS FOR MANETS

Professor: Claudio Enrico Palazzi

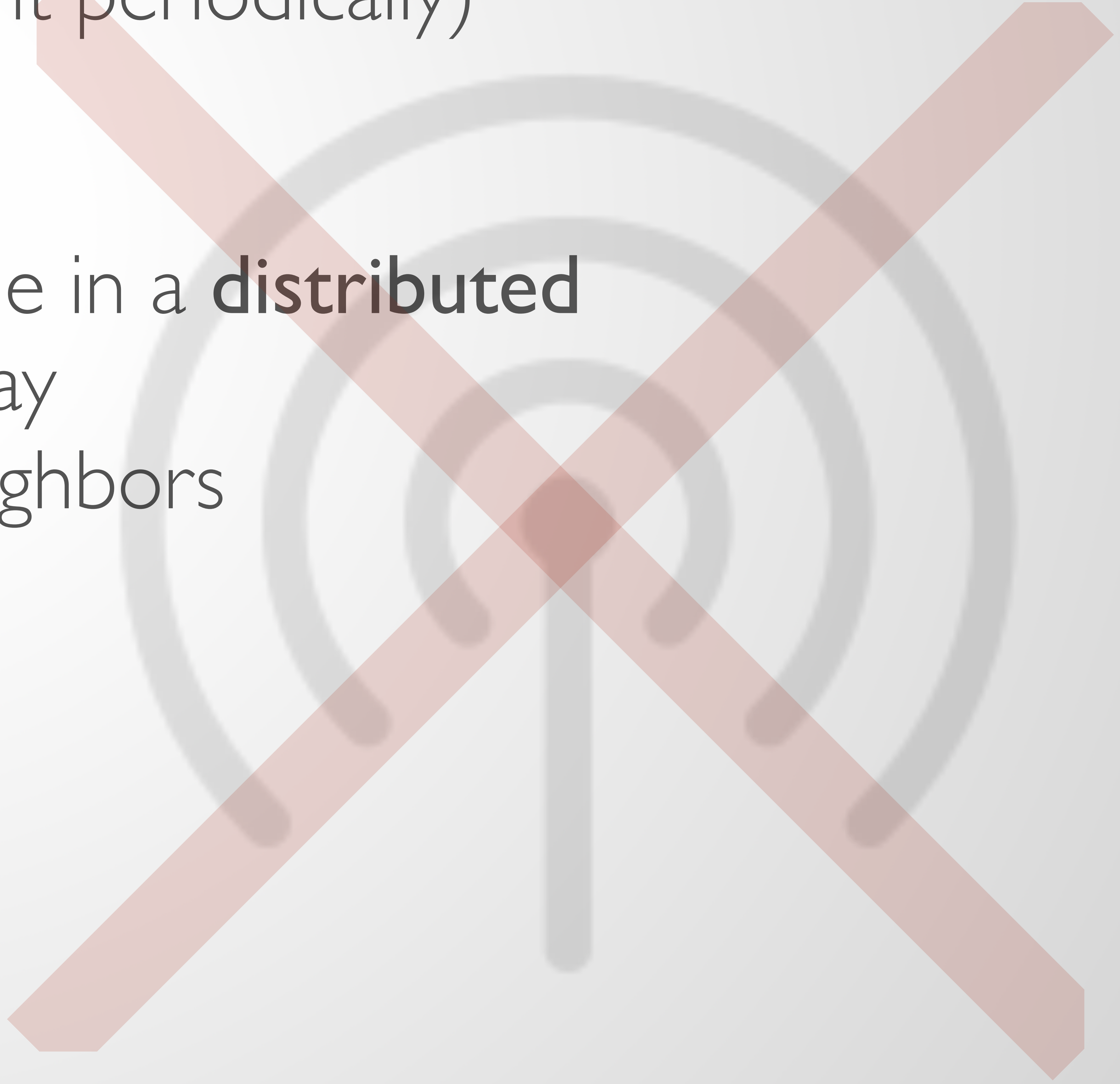
Jordan Gottardo - Master's Degree in Computer Science undergraduate

BLR

BLR: **B**eacon-**L**ess **R**outing protocol

- **No beacons** (in the sense of messages sent periodically)
- e.g. Hello Messages in Fast Broadcast

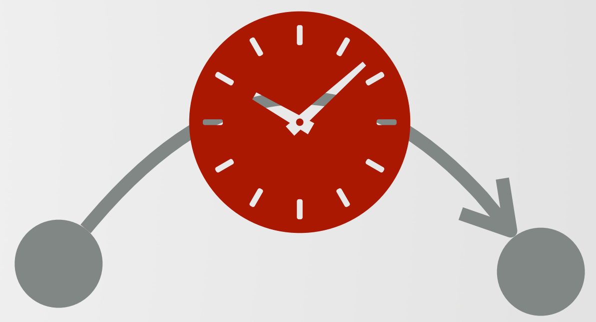
- Main idea: each node selects the next node in a **distributed** manner, based on dynamic forwarding delay
- **No information** needed about nodes' neighbors



ASSUMPTIONS

- Each node is **aware** of its position (e.g. via GPS)

- Two **parameters** need to be set:
  - `max_delay`: maximum delay per hop
  - `r`: maximum tranmission radius



BASIC MODE 1/3

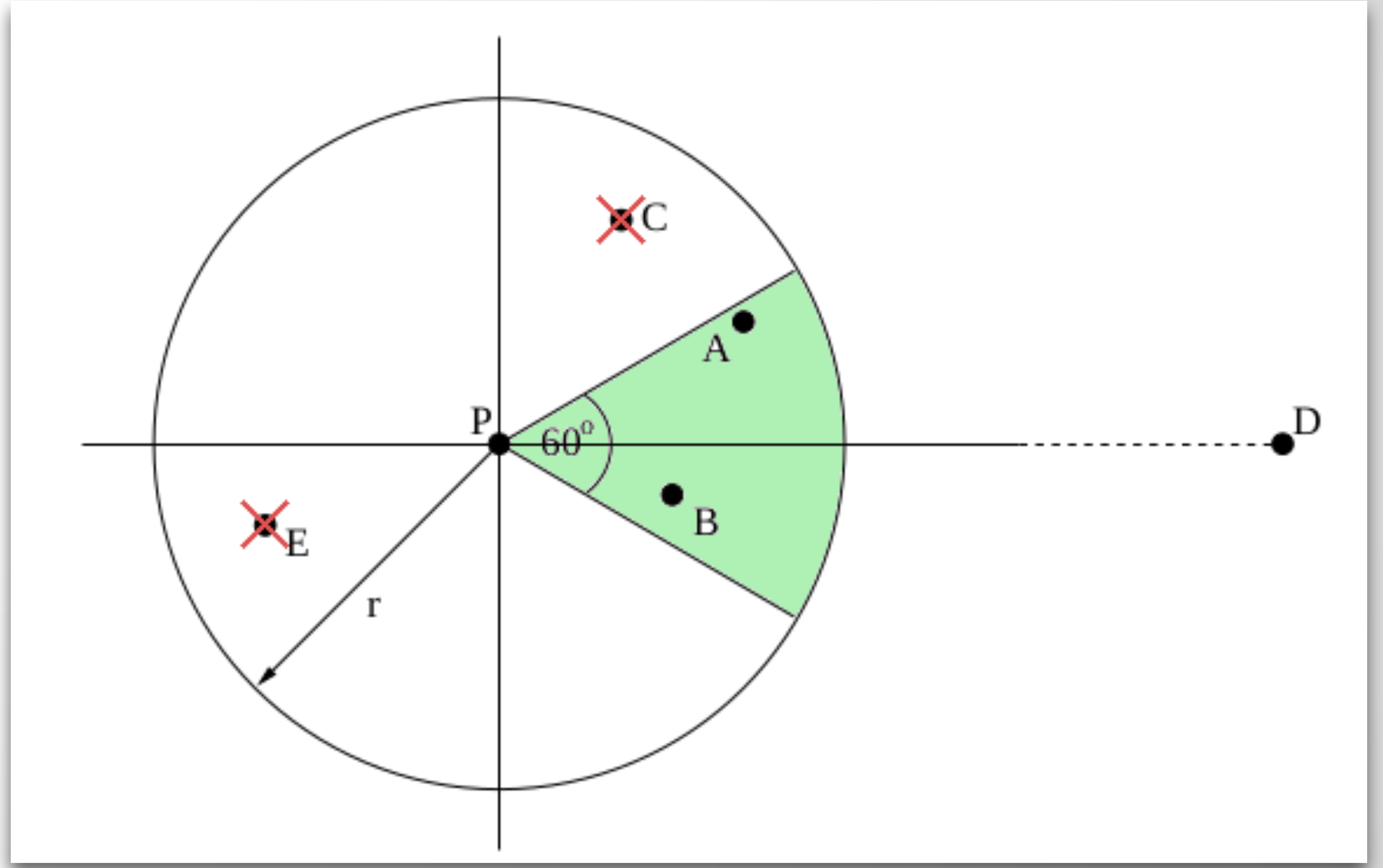
- Basic mode is used when sending packets normally
- Let's suppose P wants to send a packet to D
  1. P determines D's position and stores it in the header of the packet (`destination.position`)
  2. Then it stores its position in the header as well (`previous.position`)
  3. And eventually broadcasts the packet

Structure of a packet's header	
destination.position	previous.position

BASIC MODE 2/3

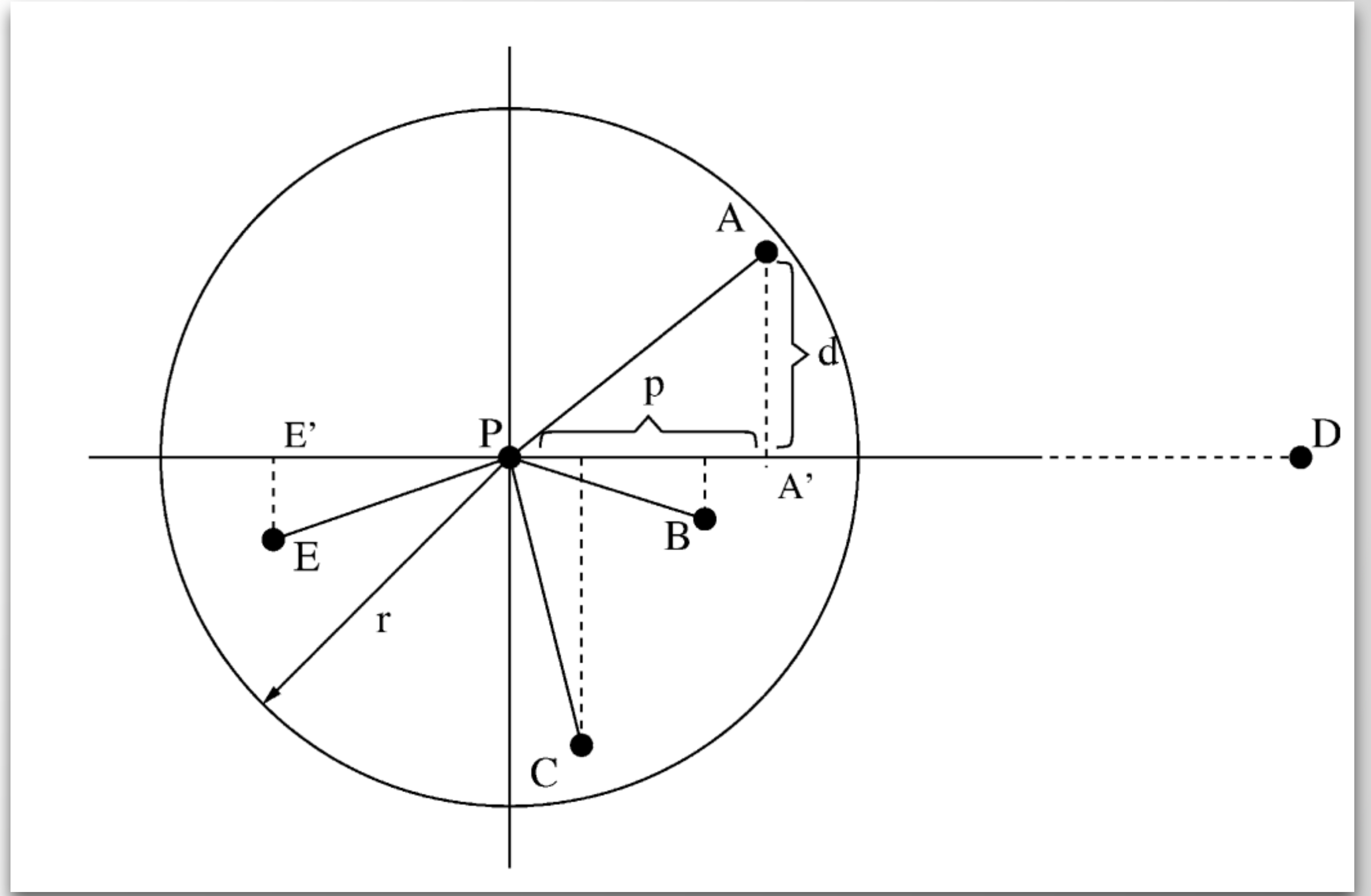
After P's broadcast, every neighbor (inside radius `r`) receives the packet

- Each neighbor participates in contention to forward the packet **only if** it is inside a **60° sector** starting from P towards D
- This sector is built in a way that every node inside the sector is **far no more than** `r` from each other node inside the sector
- Nodes outside the sector discard the packet



DEFINITION OF PROGRESS

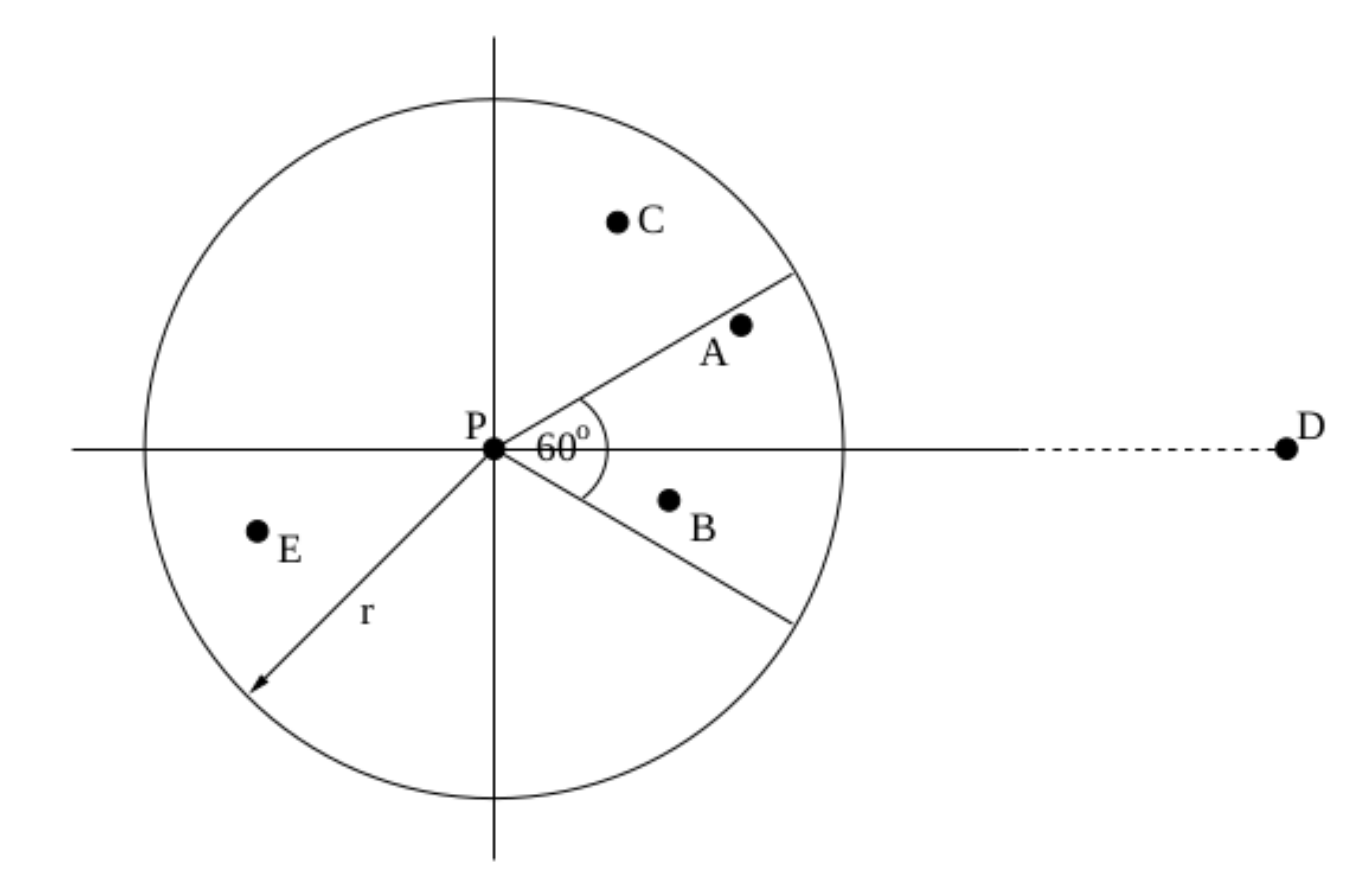
- P wants to send a message to D
- P **broadcasts** the message:
  each of its neighbors receives it
- From A's point of view, we define A's **progress (p)** as the projection of $\overrightarrow{PA}$ over $\overrightarrow{PD}$



BASIC MODE 3/3

After reception, each node calculates its progress **p** towards D

- then it waits for $Add\_delay = Max\_delay \frac{r-p}{r}$
- we have that `0 ≤ add_delay ≤ max_delay`
- nodes with smaller progress wait for more compared to ones with greater progress



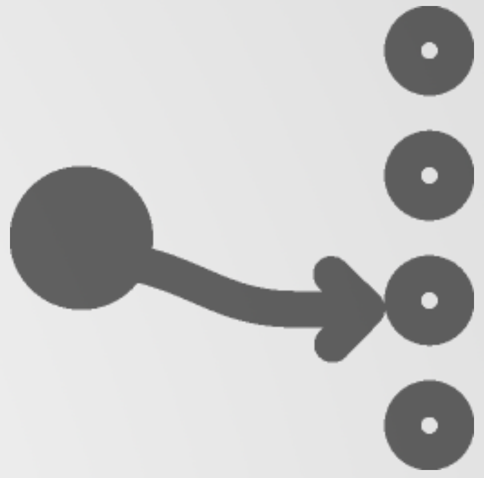
When a node has forwarded the message, other nodes in the sector receive the forwarding and back off from transmission

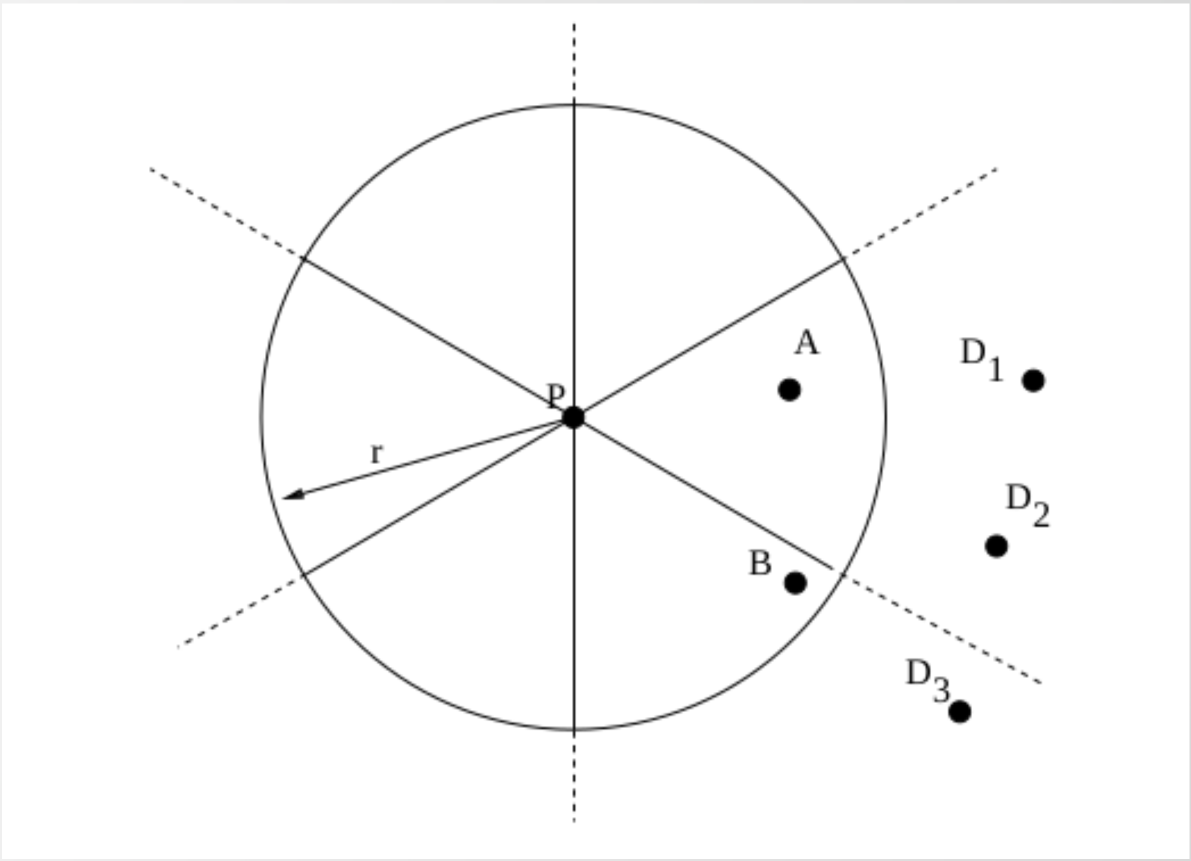
The algorithms is then repeated on the next nodes

No ack needed (**passive acknowledgements**)

OTHER FEATURES

1. **Unicast packets**:
   - After having discovered a candidate with maximum progress, subsequent packets can be sent via **unicast** at lower power and without delay





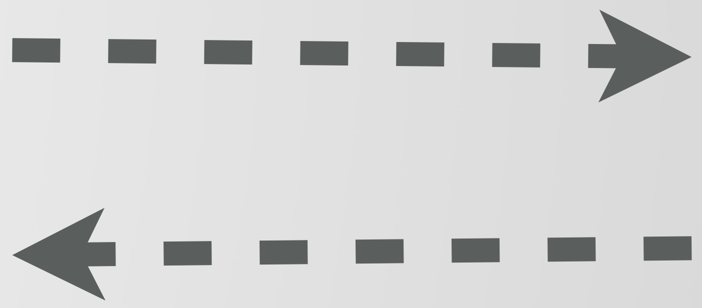
2. **Aggregation of paths**:
   - Messages towards D1 and D2 (same sector) can be forwarded **through A** via unicast without a full discovery process (for the duration of `Beacon_Threshold`)

BACKUP MODE

If the sender P doesn't detect a forwarding within `max_delay`,
it assumes that the basic greedy mode has failed
and switches to **backup mode**

Two approaches are possible:

1. **Request-Response Approach**: P asks **all** its neighbors for their position, then sends the packet via unicast to the node with the **most forward progress** (right hand rule if no neighbor has forward progress)

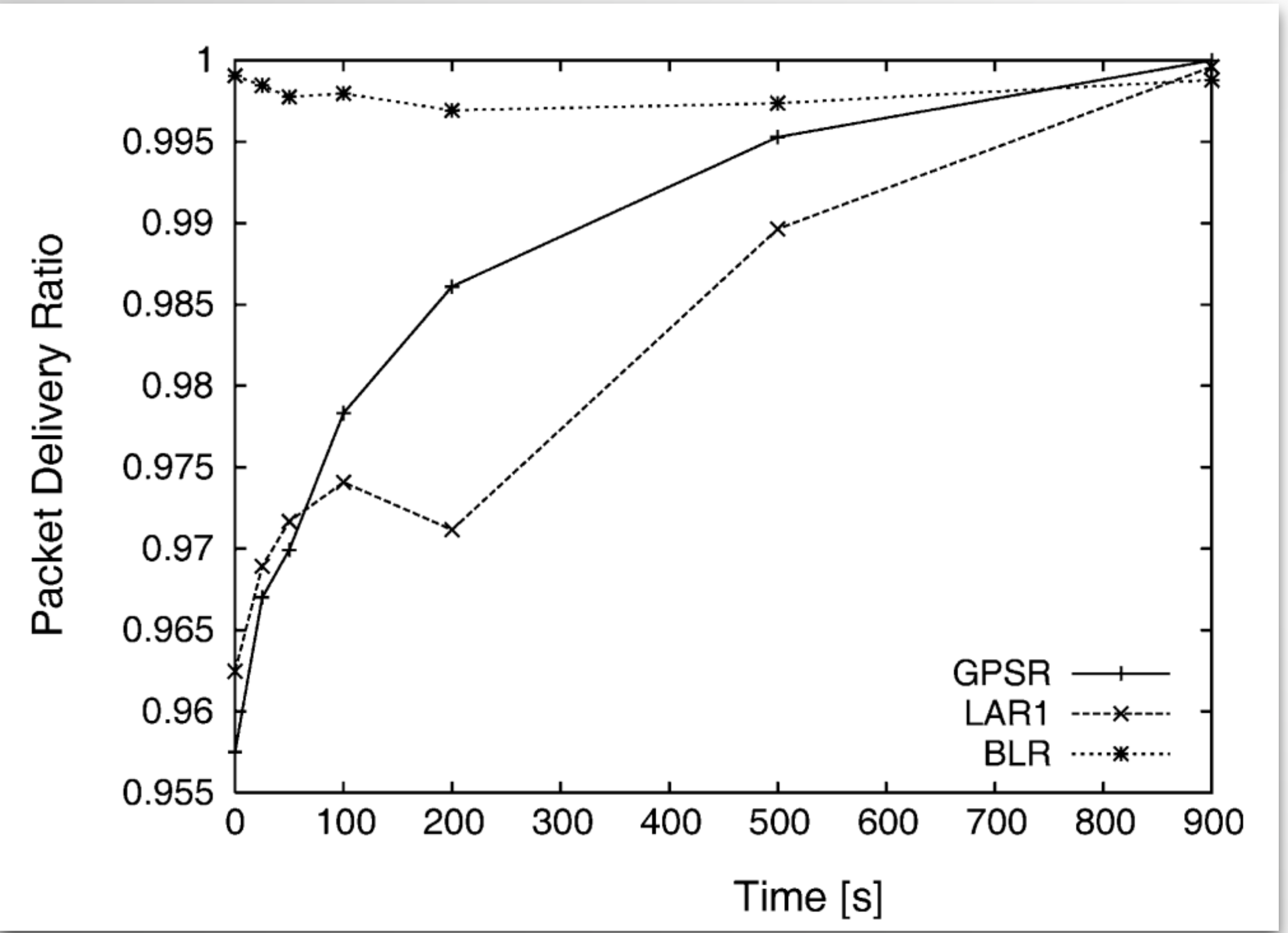


2. **Clockwise-relaying Approach**: nodes forward the packet in a clockwise manner

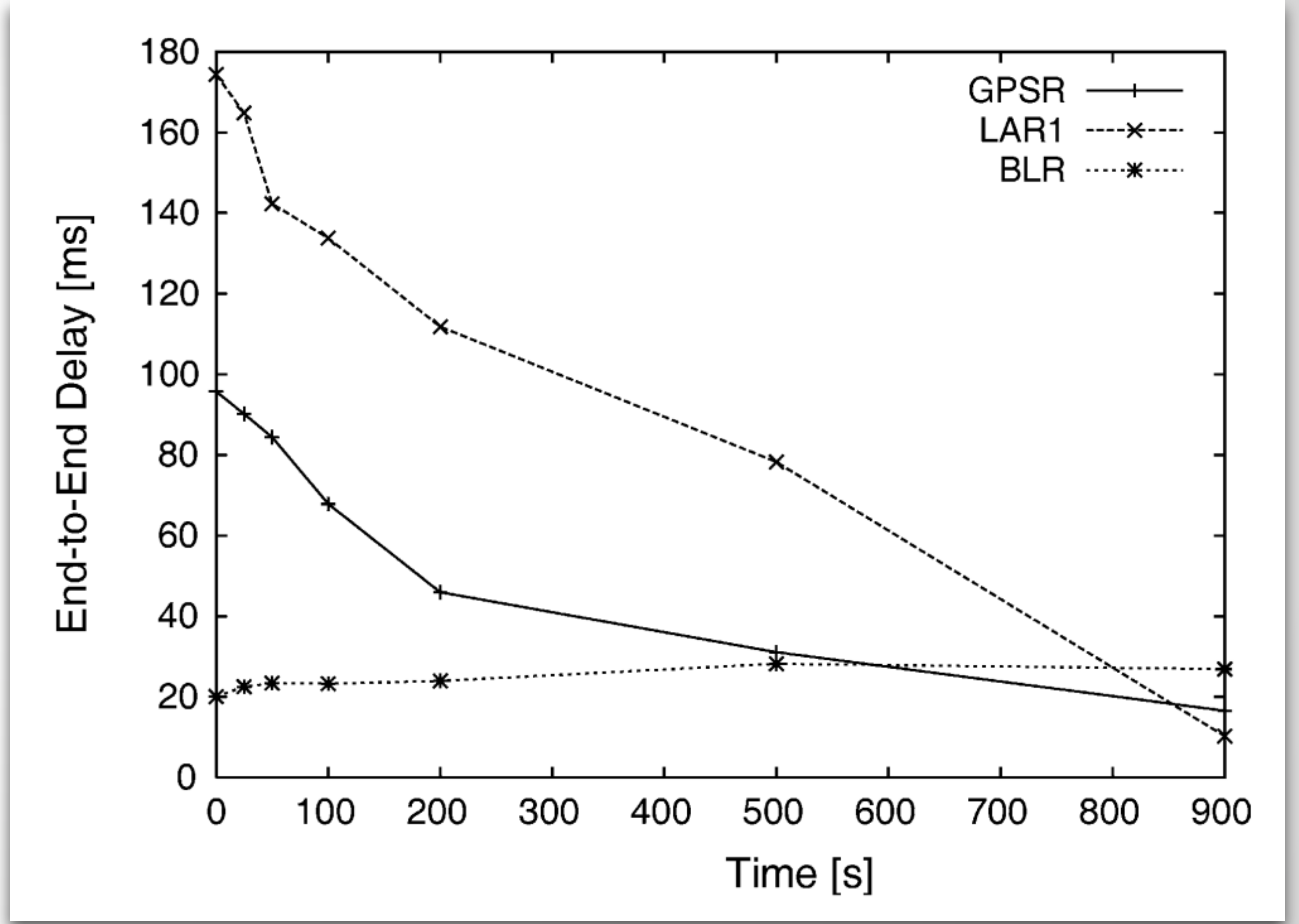
TESTBED ENVIRONMENT

Simulator	QualNet
Algorithms tested	BLR, GPSR, LAR1
Number of nodes	200
Area of simulation	600 x 3000 m^2
Duration of simulation	15m
Mobility model	Random waypoint
Speed of nodes	[1, 40] m/s
Traffic generator	Two 64B UDP packets per second
Traffic start time	180 s
Traffic end time	880 s
MAC standard	802.11 DCF
Bitrate	2 Mbps
Transmission range r	450 m
Max_delay	40 ms
Node density	111 nodes per km^2

PERFORMANCE EVALUATION



Packet Delivery Ratio vs Time



End-to-End Delay vs Time

PROS AND CONS

Pros

- Absence of beaconing helps with **saving battery power**
- BLR's performance is basically **independent of node mobility**
- Possibility to exploit directional antennas

Cons

- **Medium-high node densities** (~110 nodes/km^2) required in order to operate almost always in basic mode
- DCF overhead introduced with RTS-CTS-DATA-ACK makes some advantages of BLR useless

ROFF

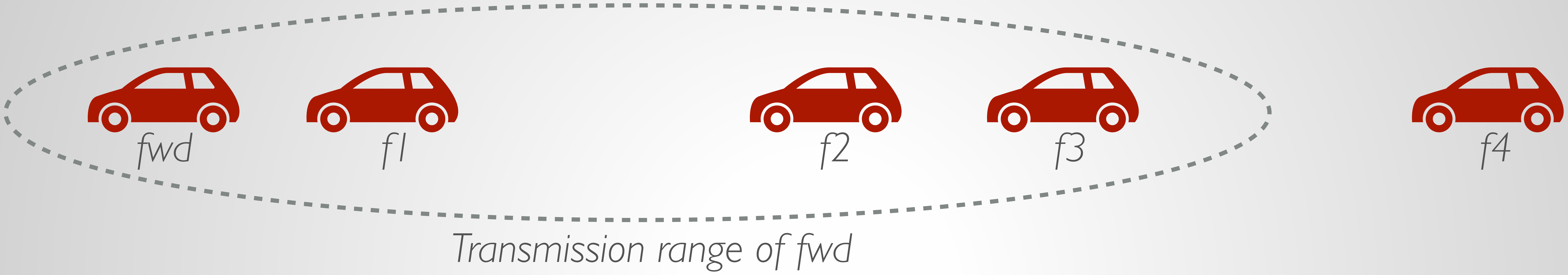
ROFF: **RO**bust and **F**ast **F**orwarding in VANETs

- Multi-hop broadcast protocol, similar to Fast Broadcast

Main problems tackled by ROFF:

1. Lack of consideration of **vehicles distribution**
2. Short difference between waiting times may cause **collisions**

MAIN IDEA OF EXISTING ALGORITMS



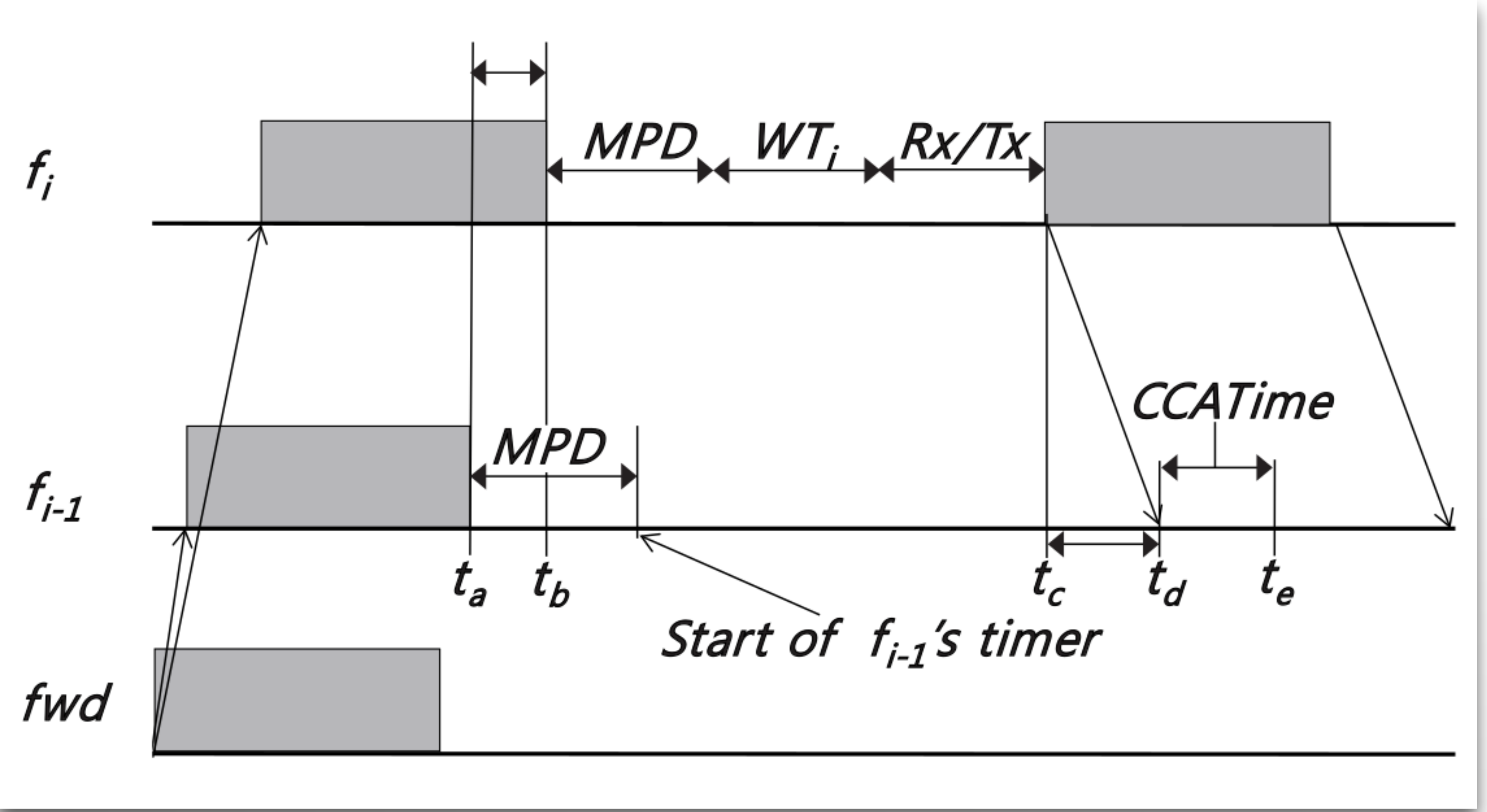
Potential forwarders' (f1...f3) waiting time is i**nversely proportional** to their distance from fwd

- f3 should be the one which forwards the message immediately, f1 and f2 should suppress their transmission

Problem with this approach: f1 and f2 can successfully suppress their transmission only if their **timer is long enough** to detect the transmission from f3

MINDIFF

ROFF defines the **`minDiff`** between two forwarders
(i.e. the minimum waiting time difference in order to successfully suppress transmission)



- **pd**: propagation delay between two nodes
- **MPD**: Mac Processing Delay, time to process a message after reception
- **Rx/Tx** turnaround time: time required to switch interfaces from reception to transmission
- **CCATime**: time before the MAC layer receives indication of reception by the PHY layer

$$minDiff_{fi,\ fi-1} = (pd_{fwd,fi} - pd_{fwd,fi-1}) + pd_{fi,fi-1} + Rx/Tx + CCATime$$

ANOTHER PROBLEM: EMPTY SPACES 1/2

In this situation, if B doesn't receive the message (for example due to lossy environment) A will be the designed forwarder and its waiting time will be slightly longer than B's since their distance is not too great

ANOTHER PROBLEM: EMPTY SPACES 2/2

However, in this situation A's waiting will be **needlessly longer** than B's due to the **great distance** between them

Problem: distances and empty spaces influence waiting times

ROFF's solution consists in giving each vehicle a **unique priority** based on distance instead of making waiting time proportional to distance

ASSUMPTIONS

- Each vehicle is equipped with a **GPS** and a digital map

- Vehicles periodically (~100 milliseconds) broadcast **beacons** (e.g. Hello Messages) containing their GPS position, speed, etc.

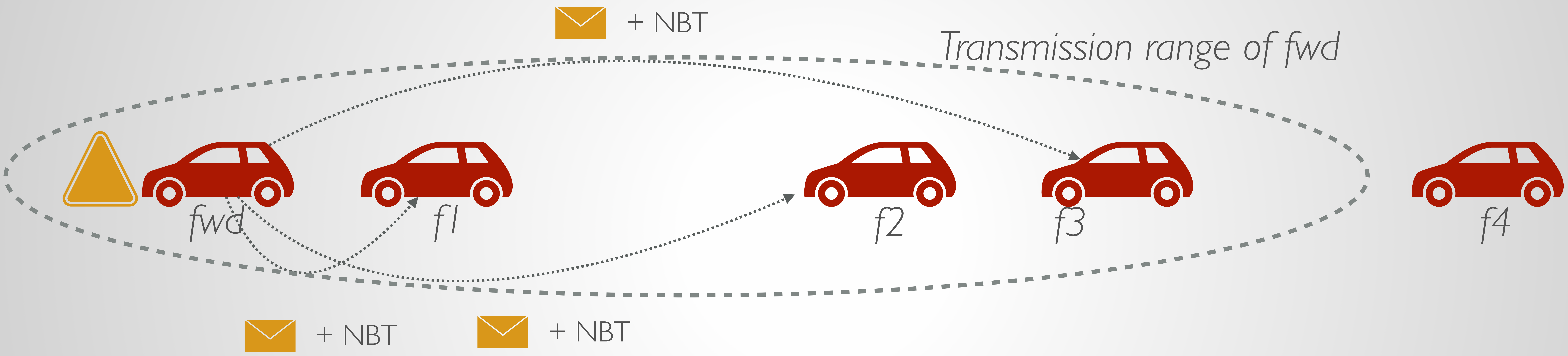
NEIGHBOR TABLE (NBT)

Each vehicle builds a **Neighbor Table (NBT)** to monitor its neighbors

- One entry for each neighbor
- All information needed to build it are included in the beacons
- The corresponding entry is updated at every beacon reception

NBT of node 1640327e9e05efc2		
ID of neighbor	Location	Beacon reception time
38243a54e71b4301	(x, y, z)	13:22:35
682284f29091750a	(x', y', z')	13:22:36

PRIORITY ACQUISITION 1/2

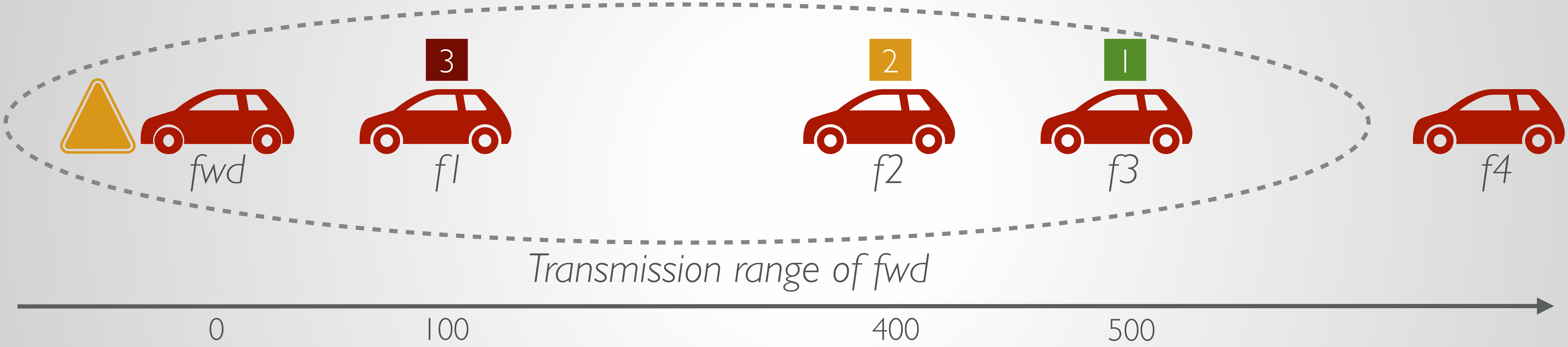


fwd sends an Alert Message

- The NBT of fwd is **piggybacked on the Alert Message**
- Both the AM and the NBT are broadcast to all following vehicles

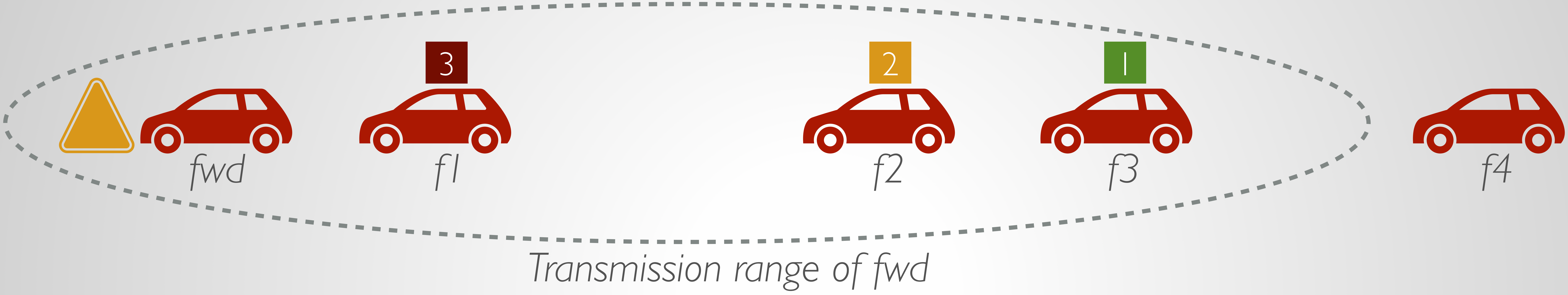
In this example, fwd's NBT will contain entries for f1, f2 and f3

PRIORITY ACQUISITION 2/2



Each potential forwarder receives fwd's NBT and calculates its **forwarding priority** based on its distance and other neighbors' distances to fwd

WAITING TIME ASSIGNMENT 1/2



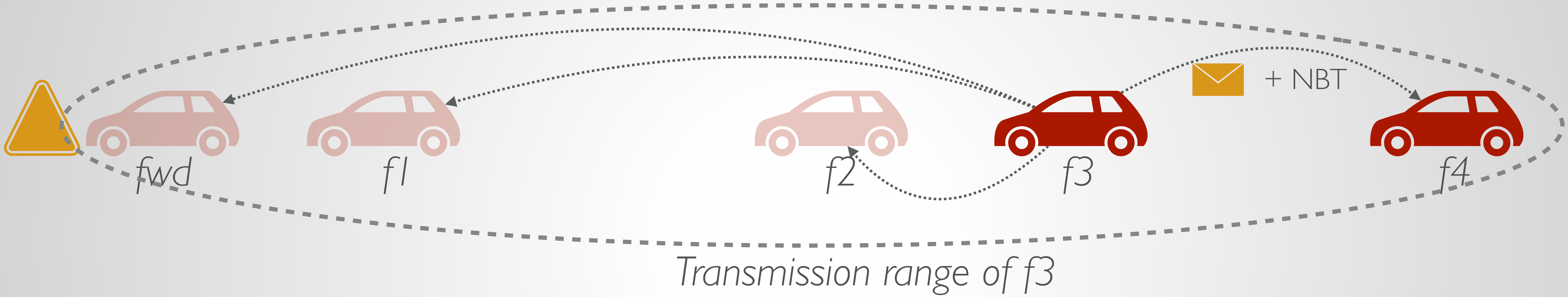
Each potential forwarder calculates its **waiting time** based on its priority using the following formula. PFC(i) indicates a potential forwarder with priority equal to i

$$WT_{PFC(k)} = \sum_{i=2}^{k} minDiff_{PFC(i),PFC(i-1)} (k \geq 2).$$

WAITING TIME ASSIGNMENT 2/2

Node	Priority	Waiting time
f3	1	0
f2	2	minDiff(f2, f3)
f1	3	minDiff(f2, f3) + minDiff(f2, f1)

BROADCAST CONTINUATION



Now f3 broadcasts immediately the Alert Message and piggybacks **its** NBT

- Priority acquisition is always carried out based on the sender's local view

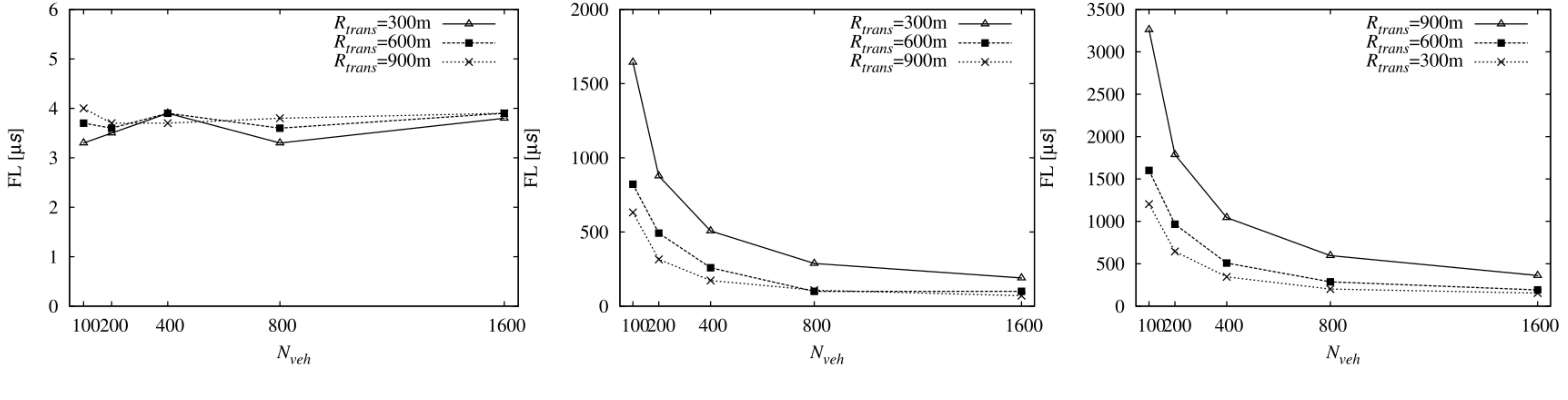
fwd, f1 and f2 discard the message since they have already received it

TESTBED ENVIRONMENT

Simulator	ns-2
Algorithms tested	ROFF, STF_5ms, STF_10ms
Number of nodes	100, 200, 400, 800, 1600
Area of simulation	4km straight highway, 6 lanes
Mobility model	Freeway
Propagation model	Nakagami
Speed of nodes (avg)	60 km/h
Bitrate	6 Mbps
Transmission range	300, 600, 900 m
Rx/Tx turnaround time	2 μs
CCATime	8 μs

PERFORMANCE EVALUATION 1/2

Forwarding latency = time from when a previous forwarder has finished transmitting the message to when another node broadcasts the message



(a) ROFF (b) STF_5ms (c) STF_10ms

Number of vehicles vs Forwarding Latency

PERFORMANCE EVALUATION 2/2

BDR = % of vehicles inside a vehicle's transmission range discovered through beacons

Number of vehicles vs Collision Probability (%)

PROS AND CONS

Pros

Collision avoidance thanks to minDiff in waiting times

Independence on empty spaces between vehicles

Cons

Might not be fit for **highly lossy** environments due to imperfect detection of neighbors in NBT (performance evaluation not yet performed)

Great overhead due to IDs in NBTs (but it is possible to overcome this problem "compressing" NBTs via **bitmaps**)

REFERENCES

- Marc Heissenbüttel, Torsten Braun, Thomas Bernoulli, Markus Wälchli. BLR: beacon-less routing algorithm for mobile ad hoc networks (2004).

  https://www.sciencedirect.com/science/article/pii/S0140366404000271?via%3Dihub

- Hongseok Yoo, Dongkyun Kim. ROFF: RObust and Fast Forwarding in Vehicular Ad-Hoc Networks (2014).

  https://ieeexplore.ieee.org/document/6906275