Chapter 10: Routing Protocols on Vehicular Ad Hoc Networks

Prof. **Yuh-Shyan Chen**

Department of Computer Science and Information Engineering

National Taipei University

Goals of this chapter

- Introduce existing routing protocols on vehicular ad hoc networks.

Outline

Geographic Routing in City Scenarios

VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks

Connectivity-Aware Routing (CAR) in Vehicular Ad Hoc Networks

DIR: Diagonal-Intersection-Based Routing Protocol for Vehicular Ad Hoc Networks

Delay-Bounded Routing in Vehicular Ad-hoc Networks

Geographic Routing in City Scenarios

Christian Lochert, Martin Mauve, Holger FuBler,
Hannes Hartenstein

ACM SIGMOBILE Mobile Computing and Communications Review, 2005

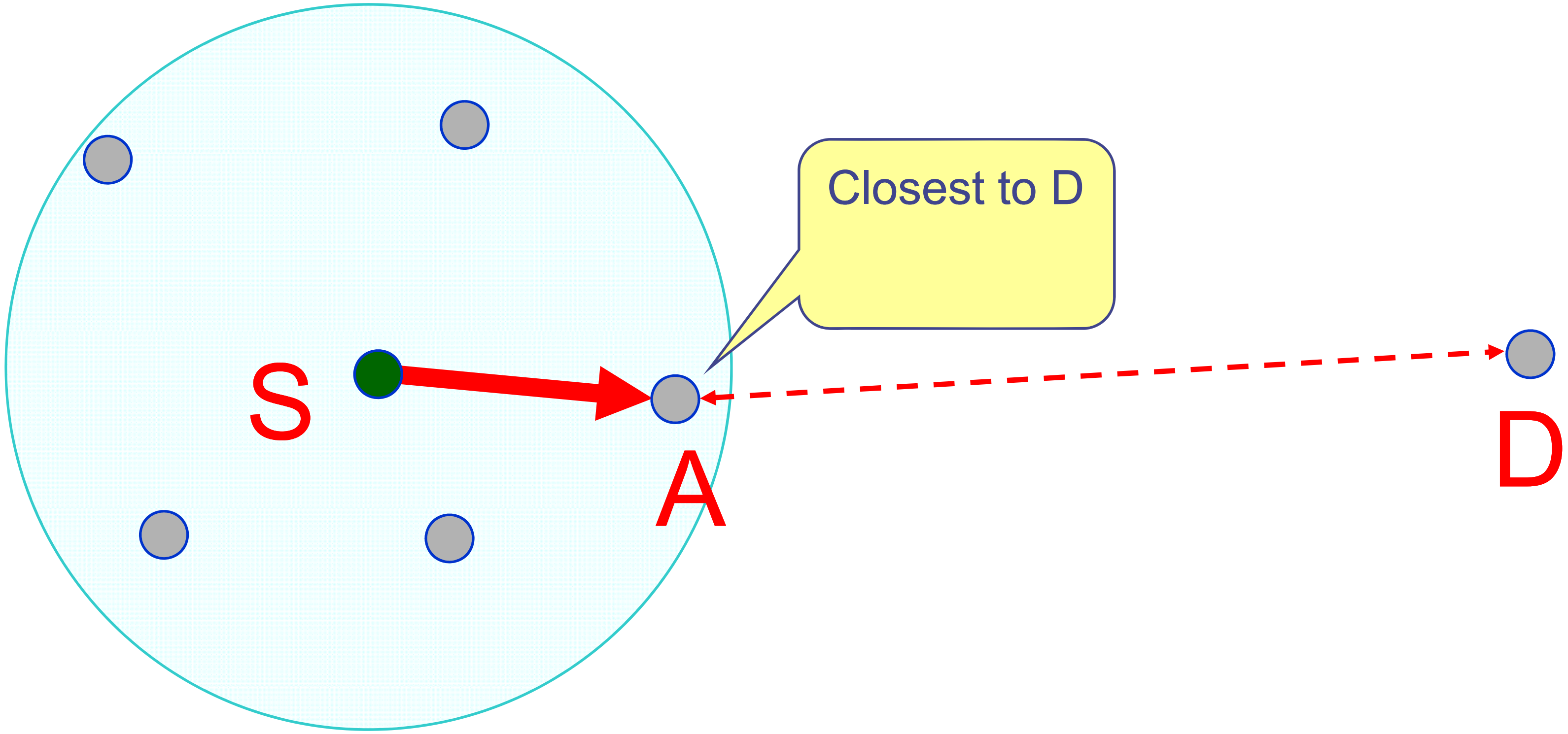
Position-based routing

- In existing position-based routing approaches an **intermediate node** forwards a packet to the direct neighbor which is closest to the geographic position of the destination. This is called **greedy forwarding**.
- For this task each node has to be aware of
  - *i)* its own position,
  - *ii)* the position of its direct neighbors and
  - *iii)* the position of the final destination.
- A node determines its own position by using GPS, the position of the neighbors is received through one hop beacon messages transmitted periodically by all nodes and the position of the final destination is provided by a location service or by a geocast application.

Cont.

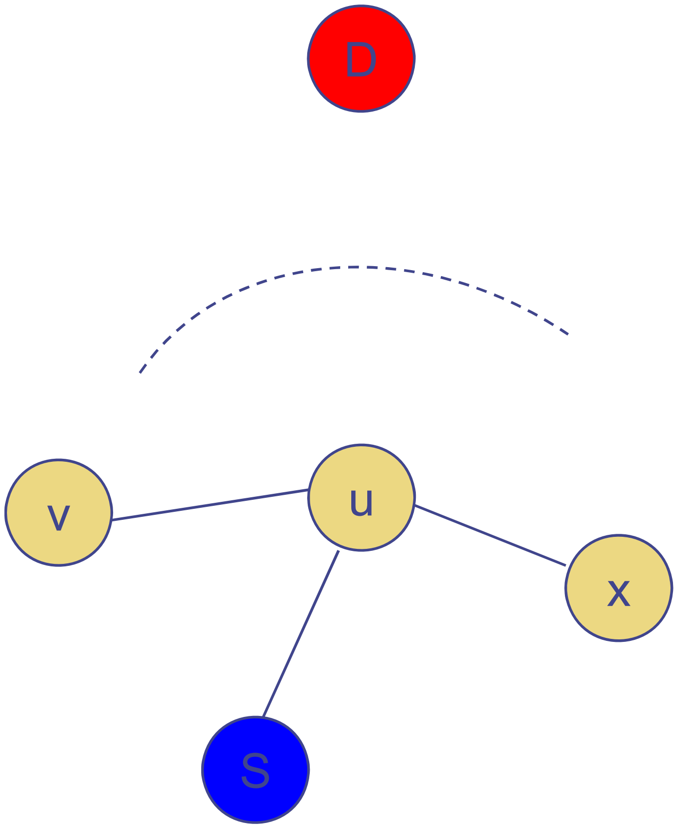
- Since greedy forwarding uses only local information a packet may reach a local optimum w.r.t. the distance to the destination, i.e. **no neighbor exists which is closer to the destination than the intermediate node itself**.
- In order to escape from a local optimum a repair strategy may be used.
  - The general aim of a repair strategy is to forward the packet to a node which is closer to the destination than the node where the packet encountered the local optimum.
  - Several repair strategies have been proposed, including Greedy Perimeter Stateless Routing and face-2.
  - However, it has been shown [4, 6] that existing repair strategies do not perform well in city environments because they rely on distributed algorithms for planarizing graphs.

Greedy Perimeter Stateless Routing (GPSR)



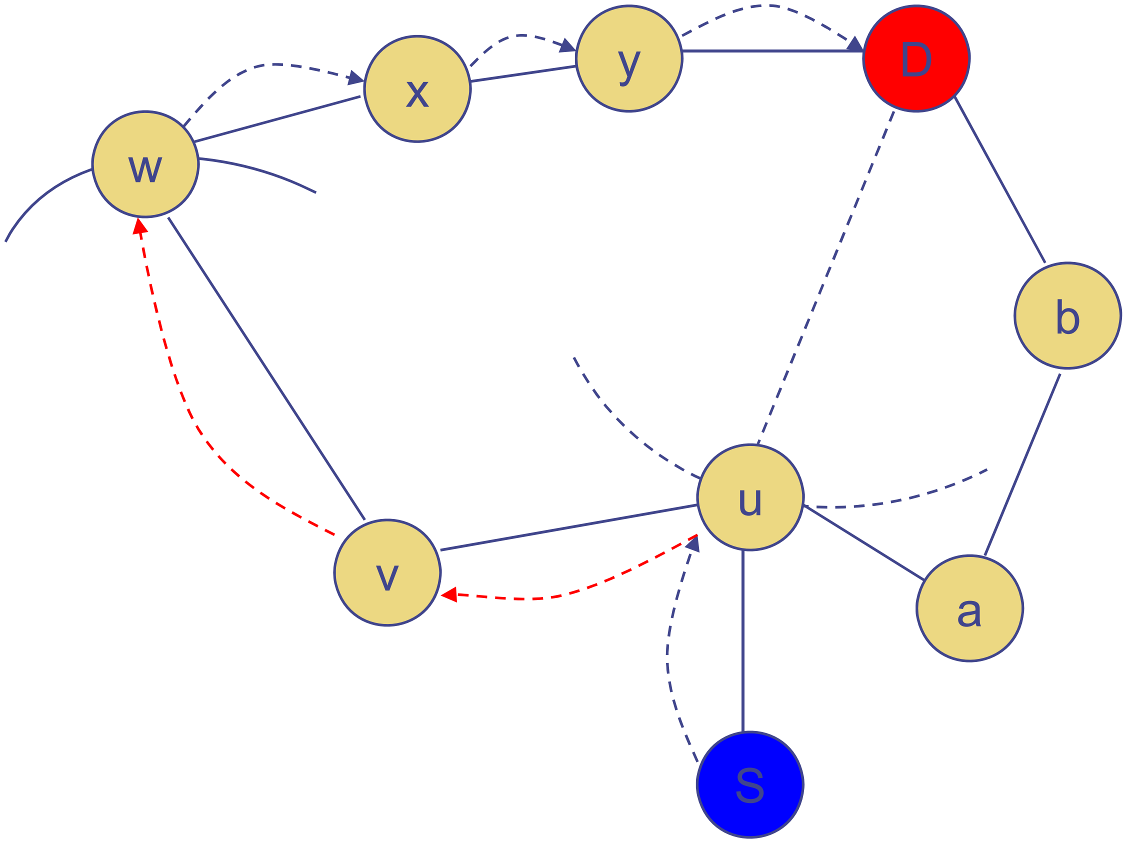
- Find neighbors who are the closer to the destination
- Forward the packet to the neighbor closest to the destination

GPSR: Local optimum



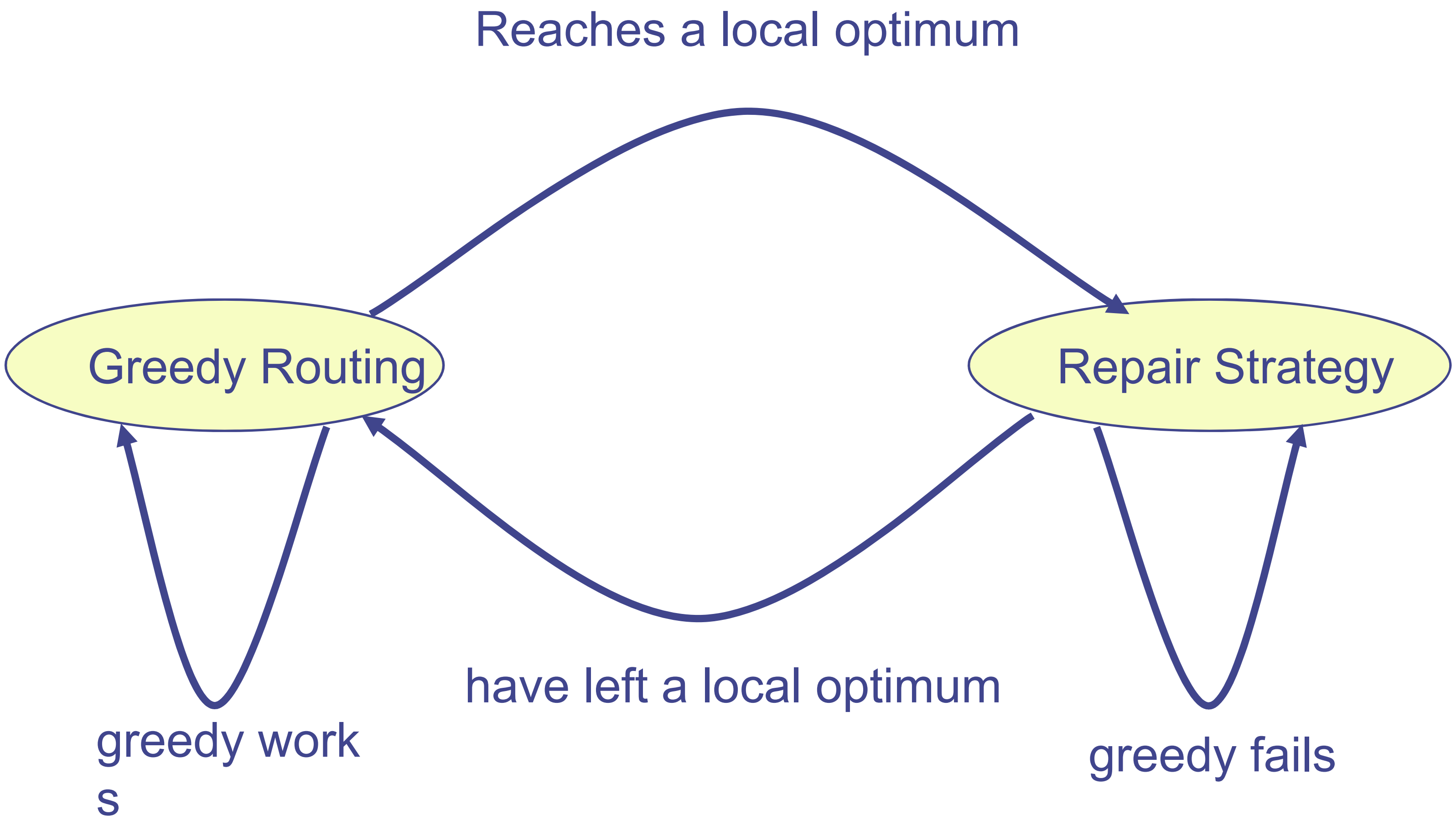
No neighbor exists which is closer to the destination than itself

GPSR: Repair Strategy

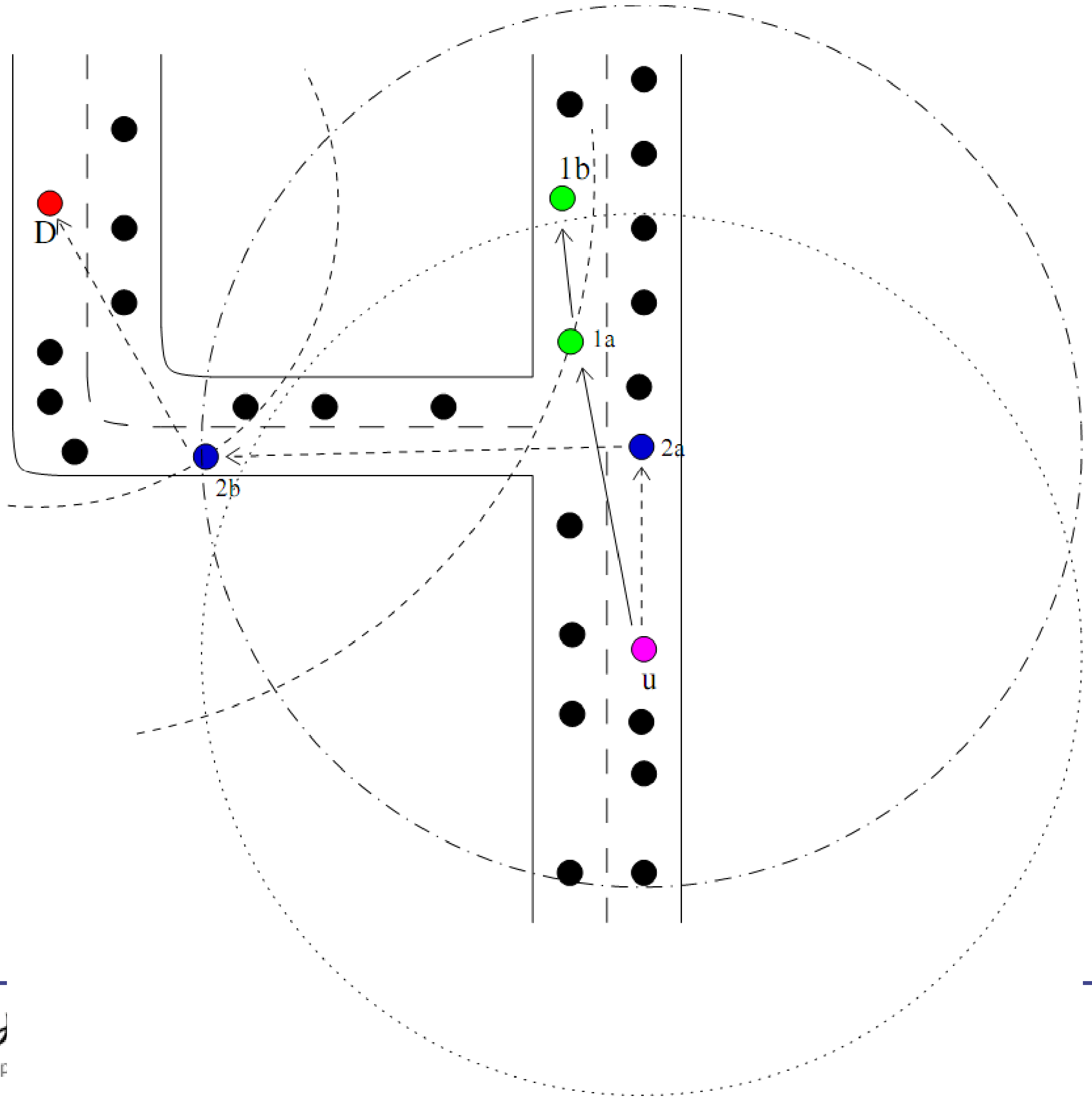


Perimeter Forwarding

GPSR


Reaches a local optimum
Greedy Routing
Repair Strategy
have left a local optimum
greedy work
s
greedy fails

GPSR: Challenges in a City Environment



Greedy Perimeter Coordinator Routing (GPCR)

- In the presence of radio obstacles the use of these algorithms frequently partitions an otherwise connected graph, making the delivery of packets impossible.
- A new routing approach for mobile Ad-Hoc Networks, called as Greedy Perimeter Coordinator Routing (GPCR), is introduced.

Greedy Perimeter Coordinator Routing

- Greedy Perimeter Coordinator Routing (GPCR) is a **position**-based routing protocol.
- The main idea of GPCR is to take advantage of the fact that **streets** and **junctions** form a natural planar graph, without using any global or external information such as a static street map.
- GPCR consists of two parts: **a restricted greedy forwarding** procedure and **a repair strategy** which is based on the topology of real-world streets and junctions and hence does not require a graph planarization algorithm.

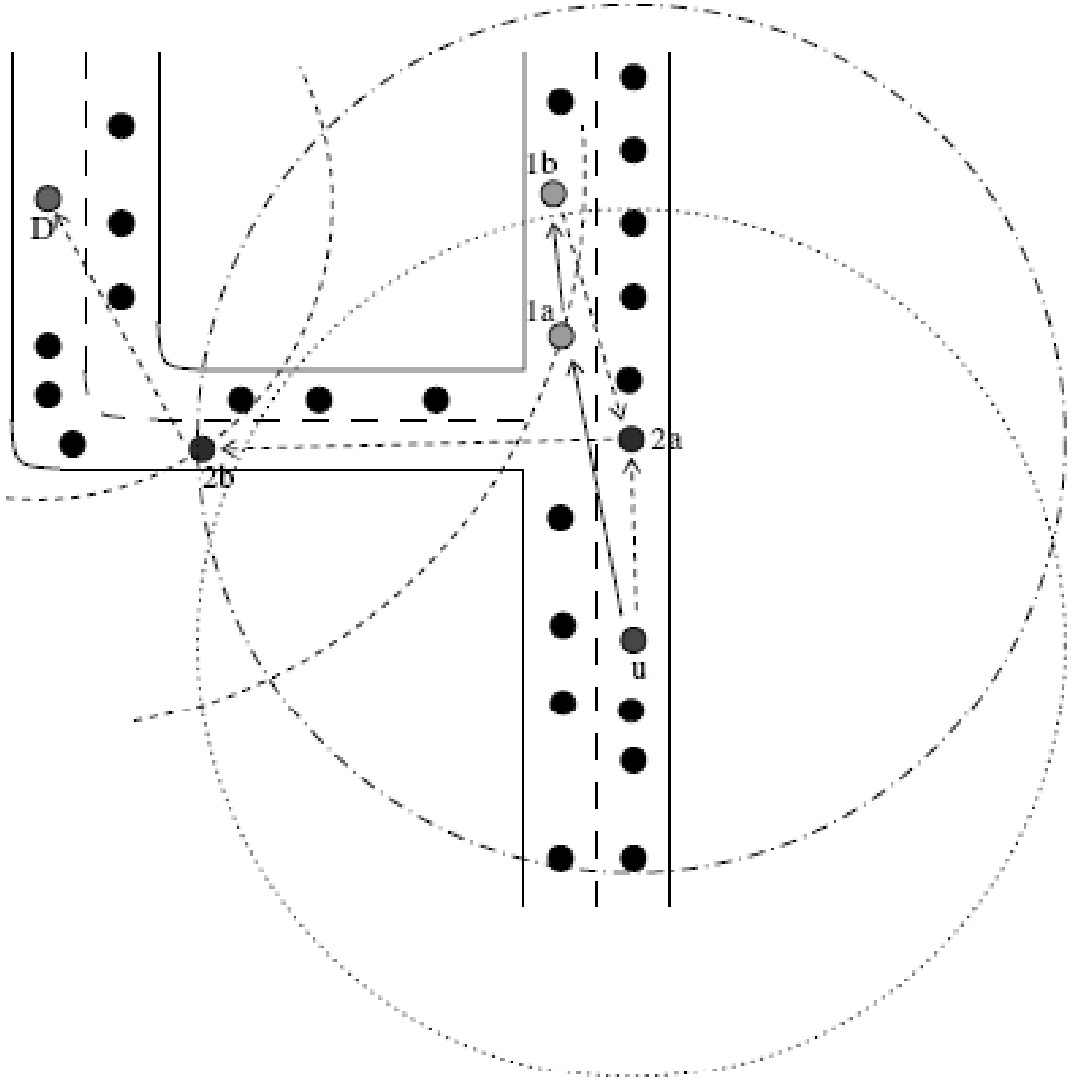
Restricted Greedy Routing

- Junctions are the only places where actual routing decision are taken.
- Therefore packets should always be forwarded to a node on a junction rather than beeing forwarded accross a junction.
- Node ***u*** would forward the packet beyond the junction to node **1*a*** if regular greedy forwarding is used.
- By forwarding the packet to node **2*a*** an alternative path to the destination node can be found without getting stuck in a local optimum.

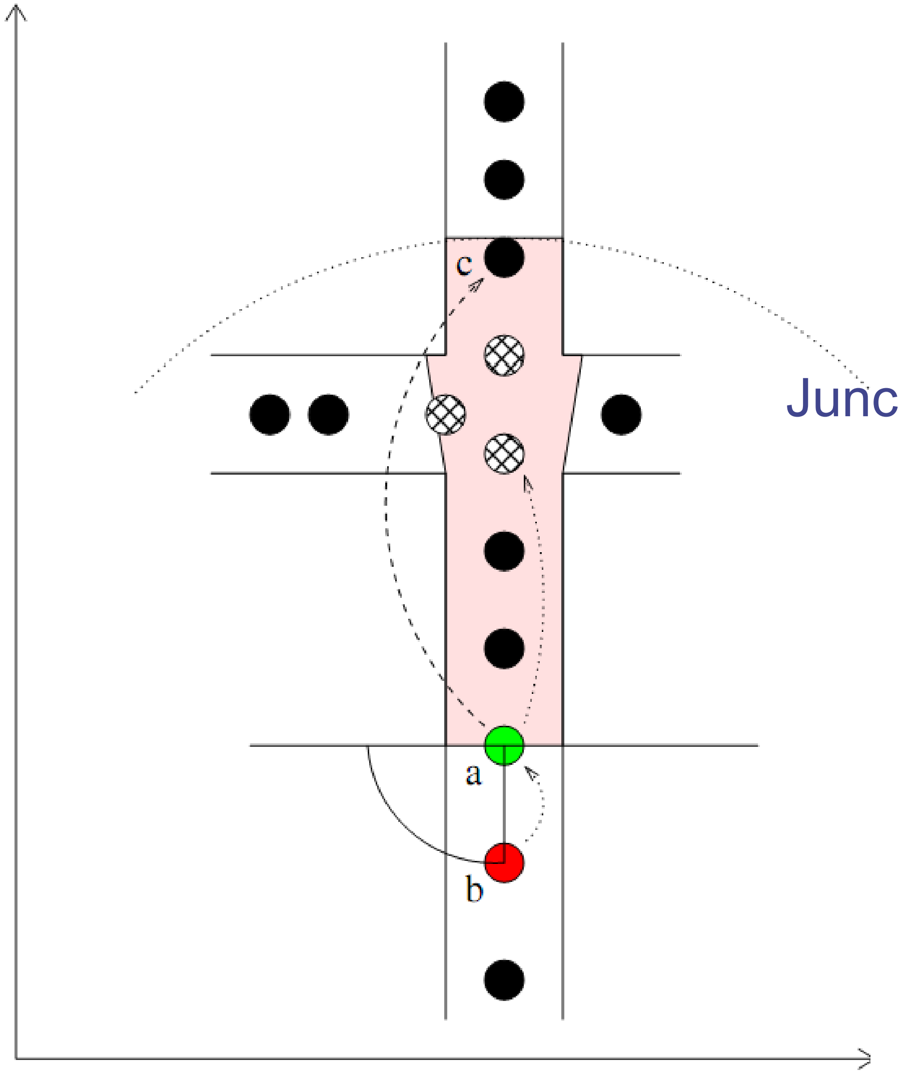
Cont.

- A coordinator broadcasts its role along with its position information. In a first step we assume that each node knows whether it is a coordinator (i.e., located in the area of a junction) or not.

Greedy Routing vs. Restricted Greedy Routing in the area of a junction.



Greedy Perimeter Coordinator Routing (GPCR)

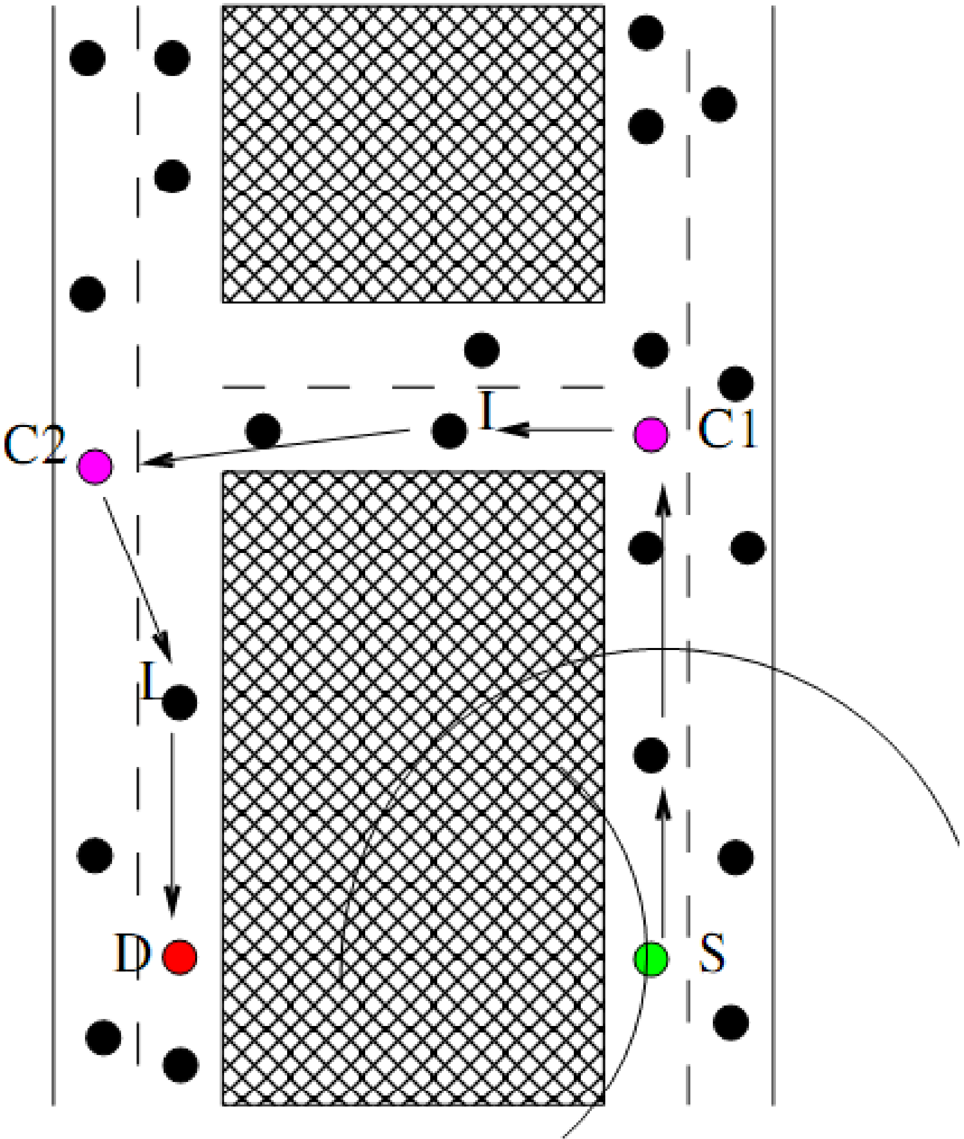


Junction (intersection)

Example

- Figure 2 shows an example of how the next hop is selected on a street.
- Node ***a*** receives a packet from node ***b***. Because ***a*** is located on a street and **not on a junction** it should forward the packet along this street.
- First the qualified neighbors of **a** are determined. Then it is checked whether at least one of them is a coordinator.
- As in this example there are **three coordinator nodes** that qualify as a next hop one of these coordinator nodes is chosen randomly and the packet will be forwarded to this coordinator.

Repair Strategy



Example

- A packet with destination *D* reaches a local optimum at node *S*.
- The forwarding of the packet is then switched to the **repair strategy** and it is routed along the street until it hits the first coordinator node.
- Node ***C1*** receives the packet and has to decide on the street the packet should follow.
- Using the **right-hand rule** it chooses the street that is the next one counterclock wise from the street the packet has arrived on.
- Therefore node ***I*** will be chosen to forward the packet. The packet will then be **forwarded along the street** until the next junction is reached.

Cont.

- When the packet arrives at the coordinator ***C2*** this node has to **decide again** on the **next street** that is to be taken and decides to forward the packet to node ***L***.
- At this point the distance to the destination is less than at the beginning of the repair strategy at node *S*.
- Hence the mode is switched back to the **greedy** strategy described above.

VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks

J Zhao and G Cao

IEEE INFOCOM 2006

Section Outline

- Introduction
- The VADD Model
- Vehicle-Assisted Data Delivery Protocols
- Performance Evaluations
- Conclusions

Introduction

- Multi-hop data delivery through VANET is complicated by the fact that vehicular networks are **highly mobile** and **frequently disconnected**.
- Existing data delivery schemes either pose too much control or no control at all on mobility, and hence not suitable for vehicular networks.
- We introduce a (vehicle-assisted data delivery) VADD protocol which can forward the packet to the best road with the lowest data delivery delay.
  - Adopt idea of the carry and forward
  - Based on the existing traffic pattern, a vehicle can find the next road to forward the packet to reduce the delay.

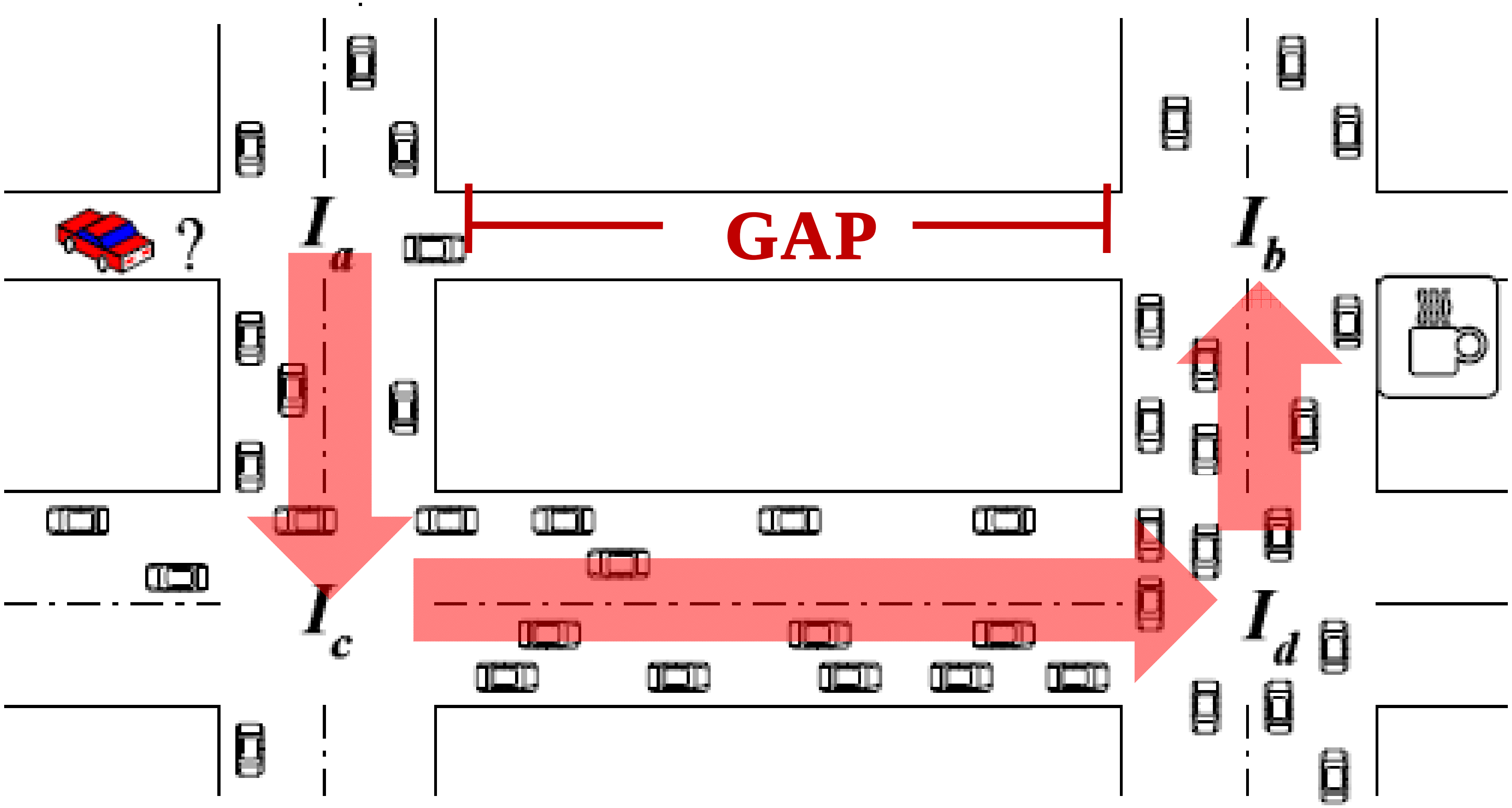
The VADD Model

- A vehicle knows its location by GPS device, and the packet delivery information such as *source id*, *source location*, *packet generation time*, *destination location*, *expiration time*, in the packet header.
- Vehicles can find their neighbors throguh periodic beacon messages, which also enclose the physical location of the sender.
- Vehicles are assumed to be equipped with pre-load digital maps, which provide street-level map and traffic statistics (such as traffic density and vehicle speed on roads at different times of the day)

VADD overview

- The most important issue is to select a forwarding path with the smallest packet delivery delay.
- Although geographical forwarding approaches such as GPSR which always chooses the next hop closer to the destination, it may not be suitable for sparsely connected vehicular networks.

VADD overview: find a path to the coffee shop



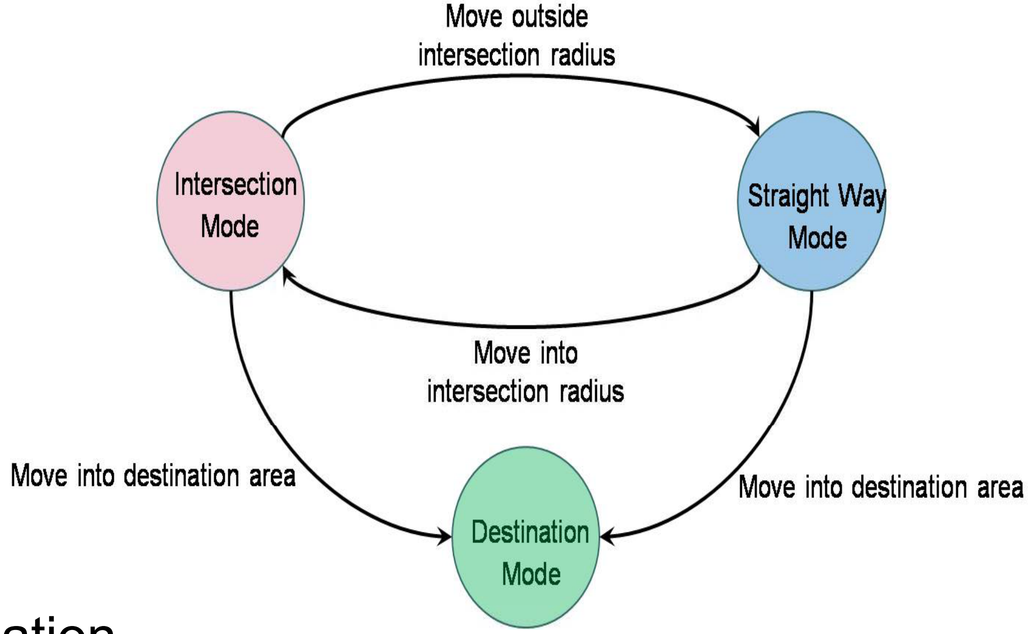
To forward the request through -> Ic , Ic -> Id , Id -> Ib would be faster than through Ia->Ib.

VADD basic principles

1. Transmit through wireless channels as much as possible.
2. If the packet gas to be carried through certain roads, the road with higher speed should be chosen.
3. Due to the unpredictable nature of vehicular ad-hoc networks, we cannot expect the packet to be successfully routed along the pre-computed optimal path, so dynamic path selection should continuously be executed throughout the packet forwarding process.

VADD: Three packet modes (based on location of the packet carrier)

- *Intersection* Mode
  - Optimize the packet forwarding direction
  - is the most critical and complicated one


Move outside intersection radius
Intersection Mode
Straight Way Mode
Move into intersection radius
Move into destination area
Move into destination area
Destination Mode

- *StraightWay* Mode
  - Geographically greedy forwarding towards next target intersection
- *Destination* Mode
  - Broadcast packet to destination

The VADD Delay Model

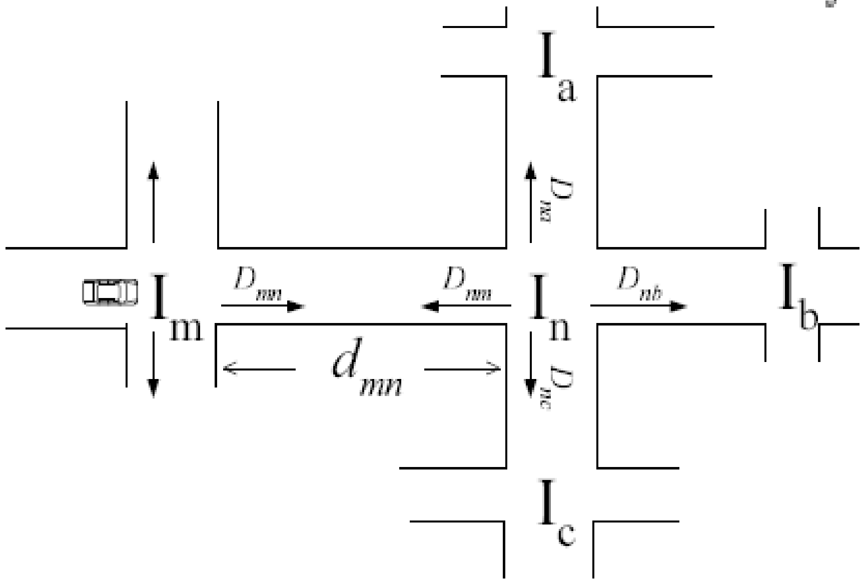


Fig. 3. An example of the VADD Delay Model

$$d_{ij} = (1 - e^{-R\cdot\rho_{ij}}) \cdot \frac{l_{ij}\cdot c}{R} + e^{-R\cdot\rho_{ij}} \cdot \frac{l_{ij}}{v_{ij}}$$

r_{ij}: the road from I_i to I_j.

l_{ij}: the Euclidean distance of r_{ij}.

ρ_{ij}: the vehicle density on r_{ij}.

v_{ij}: the average vehicle velocity on r_{ij}.

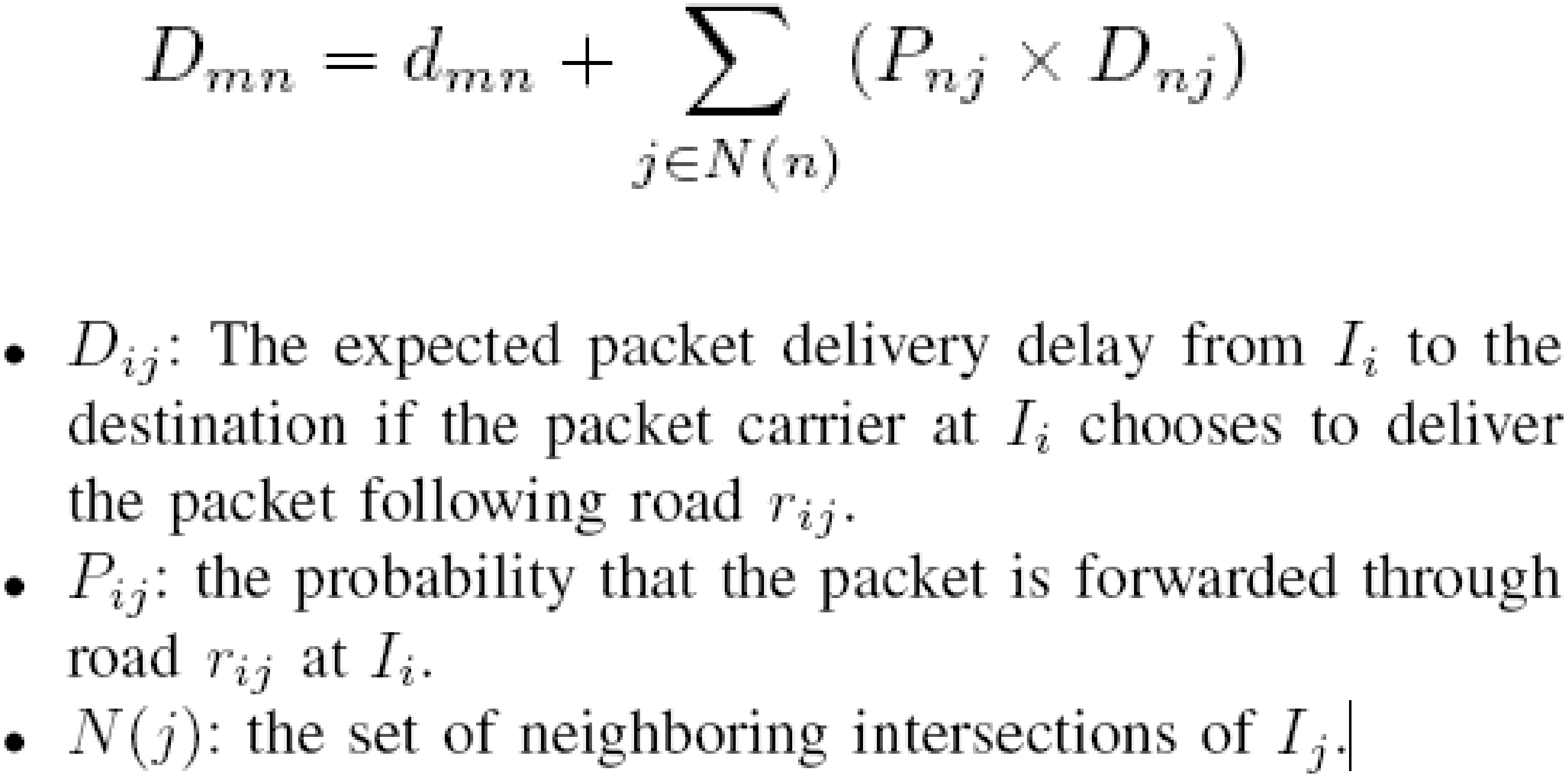
d_{ij}: the expected packet forwarding delay from I_i to I_j.

$$d_{ij} = \begin{cases} \alpha \cdot l_{ij}, & \text{if } \frac{1}{\rho_{ij}} \leq R \\ \frac{l_{ij}}{v_{ij}} - \beta \cdot \rho_{ij}, & \text{if } \frac{1}{\rho_{ij}} > R \end{cases}$$

Cont.

- The equation indicates that if the average distance between vehicles is smaller than R, wireless transmission is used to forward the packet. Otherwise, vehicles are used to carry the data.
- One way to view the VADD delay model is to represent the vehicular network as a directed graph, in which nodes represent intersections and edges represent the roads connecting adjacent intersections.
- The packet forwarding delay between two adjacent intersections is the weight of the edge.
- Given the weight on each edge, a naive optimal forwarding path selection scheme is to compute the shortest path from source to destination by applying *Dijkstra's* algorithm.

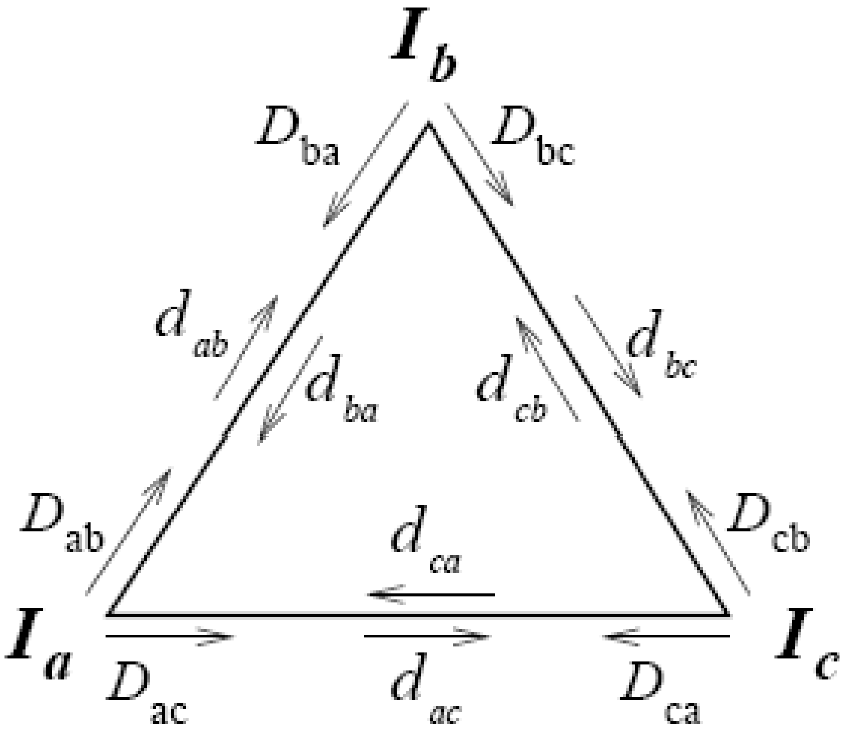
A stochastic model to estimate the data delivery delay



$$D_{mn} = d_{mn} + \sum_{j \in N(n)} (P_{nj} \times D_{nj})$$

- D_{ij}: The expected packet delivery delay from I_i to the destination if the packet carrier at I_i chooses to deliver the packet following road r_{ij}.
- P_{ij}: the probability that the packet is forwarded through road r_{ij} at I_i.
- $N(j)$: the set of neighboring intersections of I_j.

An example of VADD Delay Model



$$\begin{cases} D_{ac} = d_{ac} \\ D_{ab} = d_{ab} + P_{ba} \cdot D_{ba} + P_{bc} \cdot D_{bc} \\ D_{ba} = d_{ba} + P_{ab} \cdot D_{ab} + P_{ac} \cdot D_{ac} \\ D_{bc} = d_{bc} \\ D_{cb} = 0 \\ D_{ca} = 0 \end{cases}$$

Pij : the probability that the packet is forwarded through road *rij* at *Ii*

Suppose a data packet reaches *Ia*, and the destination is *Ic*.

$$D_{ac} = d_{ac} \qquad D_{ab} = \frac{1}{1 - P_{ab} \cdot P_{ba}} \times (d_{ab} + P_{ba} \cdot d_{ba} + P_{ba} \cdot P_{ac} \cdot d_{ac} + P_{bc} \cdot d_{bc})$$

The VADD Model (cont.)

we rename the unknown D_{ij} as x_{ij}.

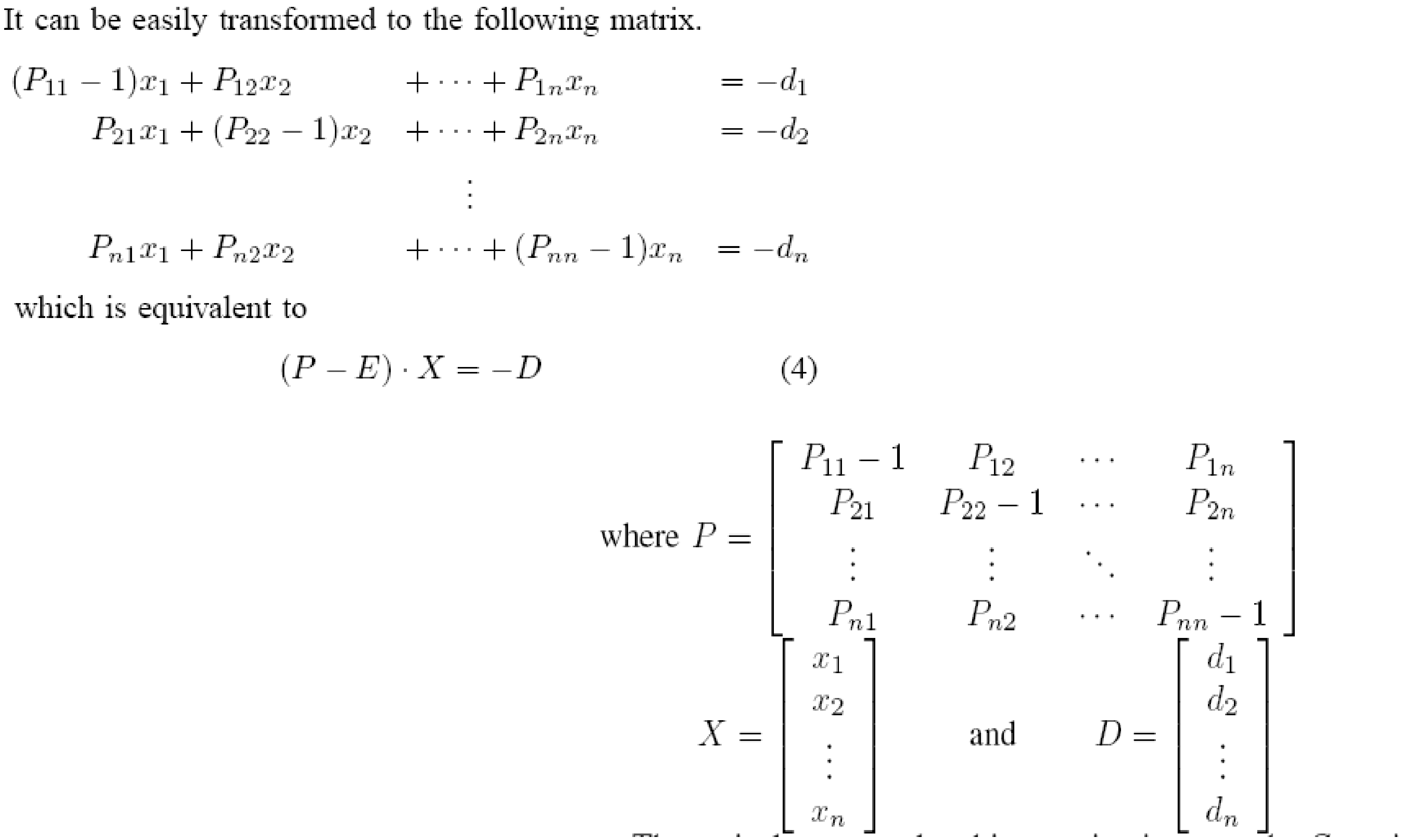
rename the

subscript ij of d_{ij} and x_{ij} with a unique number for each pair ij, and rename the subscript of P_{ij} by its position in the equations. Then, we can derive n linear equations with n unknowns $x_1, x_2, \cdots, x_n$, where n equals to the number of roads within the boundary:

$$
\begin{aligned}
x_1 &= d_1 + P_{11}x_1 + P_{12}x_2 + \cdots + P_{1n}x_n \\
x_2 &= d_2 + P_{21}x_1 + P_{22}x_2 + \cdots + P_{2n}x_n \\
&\vdots \\
x_n &= d_n + P_{n1}x_1 + P_{n2}x_2 + \cdots + P_{nn}x_n
\end{aligned}
$$

Hot spot
sender

The VADD Model (cont.)



It can be easily transformed to the following matrix.

$$\begin{aligned}
(P_{11}-1)x_1 + P_{12}x_2 &+ \cdots + P_{1n}x_n &= -d_1 \\
P_{21}x_1 + (P_{22}-1)x_2 &+ \cdots + P_{2n}x_n &= -d_2 \\
&\vdots \\
P_{n1}x_1 + P_{n2}x_2 &+ \cdots + (P_{nn}-1)x_n &= -d_n
\end{aligned}$$

which is equivalent to

$$(P - E) \cdot X = -D \qquad (4)$$

$$\text{where } P = \begin{bmatrix} P_{11}-1 & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22}-1 & \cdots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & \cdots & P_{nn}-1 \end{bmatrix}$$

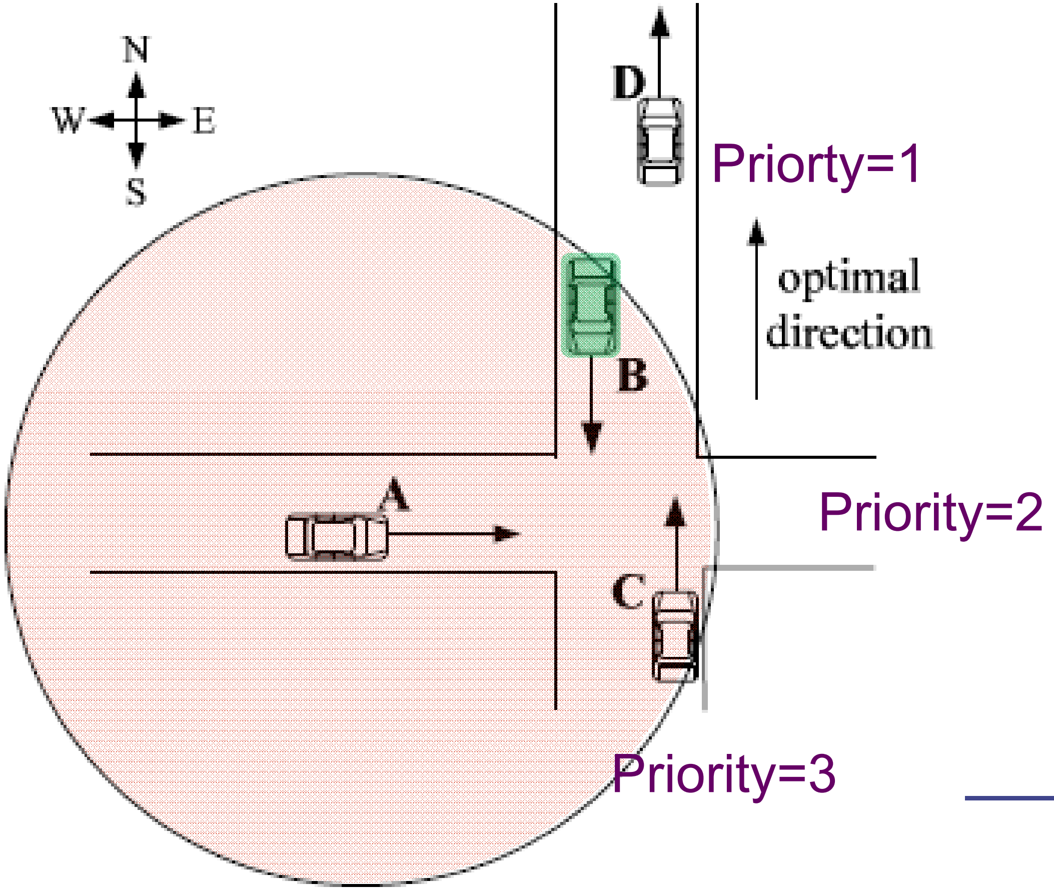
$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}$$

VADD Protocols

- VADD Protocols Used in the Intersection Mode
  - Location First Probe (L-VADD)
  - Direction First Probe (D-VADD)
  - Hybrid Probe (H-VADD)
- Data Forwarding in the StraightWay Mode

Location First Probe (L-VADD)

- Each outgoing road is assigned a priority where smaller D_{ij} has higher priority

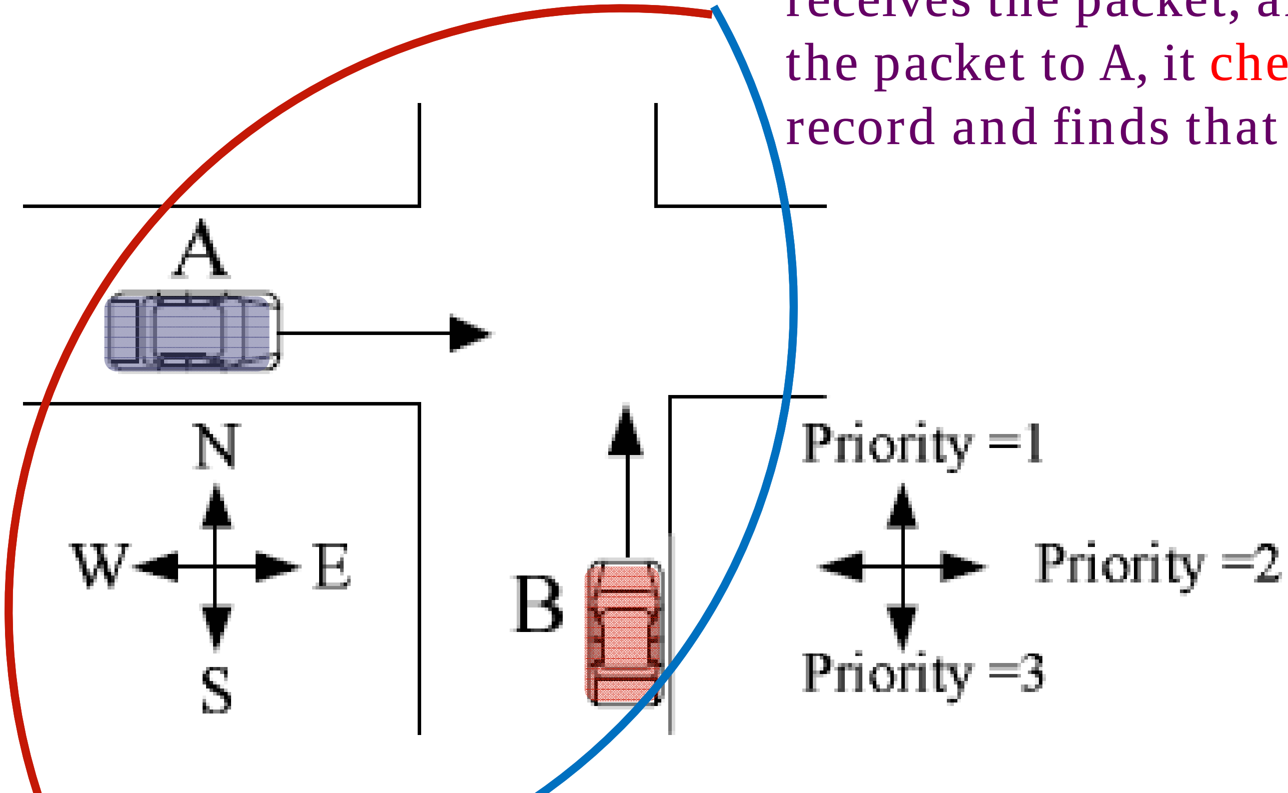


L-VADD (cont.)

- L-VADD may result in routing loops

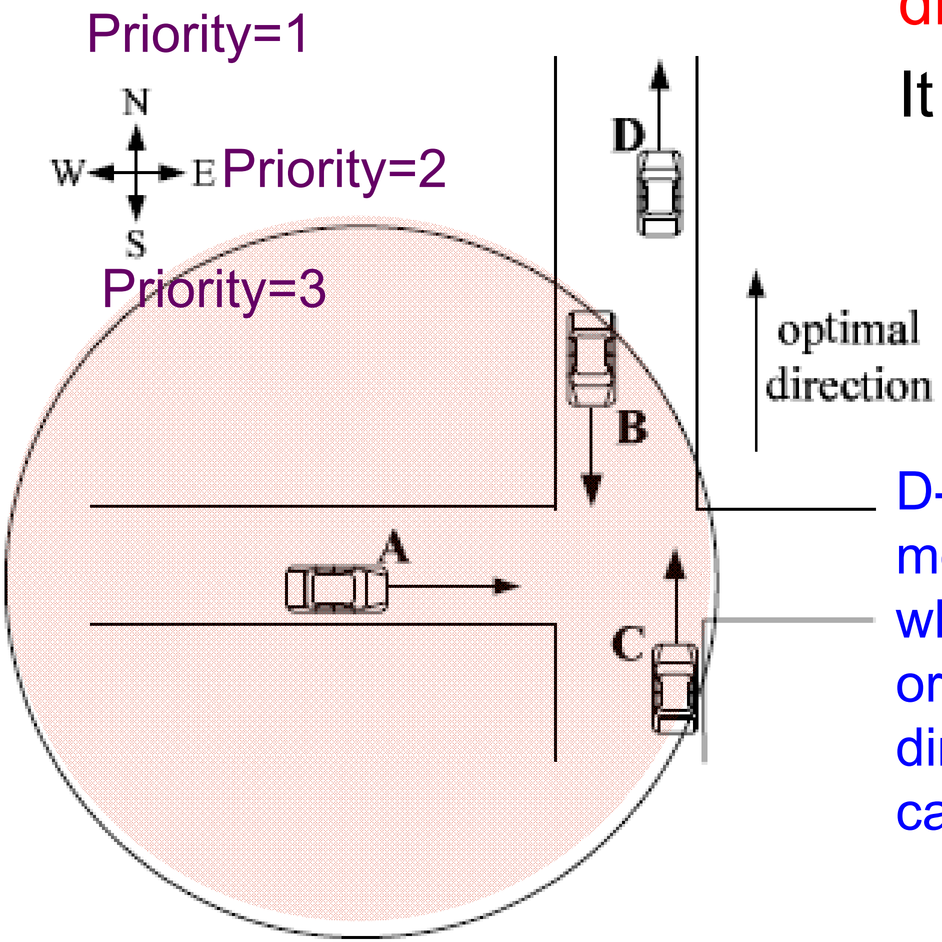
Solution:

A records its own id as the previous_hop before forwarding the packet to B. When B receives the packet, and decides to forward the packet to A, it checks the previous hop record and finds that A is the previous hop



Direction First Probe (D-VADD)

- D-VADD is free from routing loops at intersection areas



disadvantage:

It may suffer from long packet forwarding and long packet delivery delay.

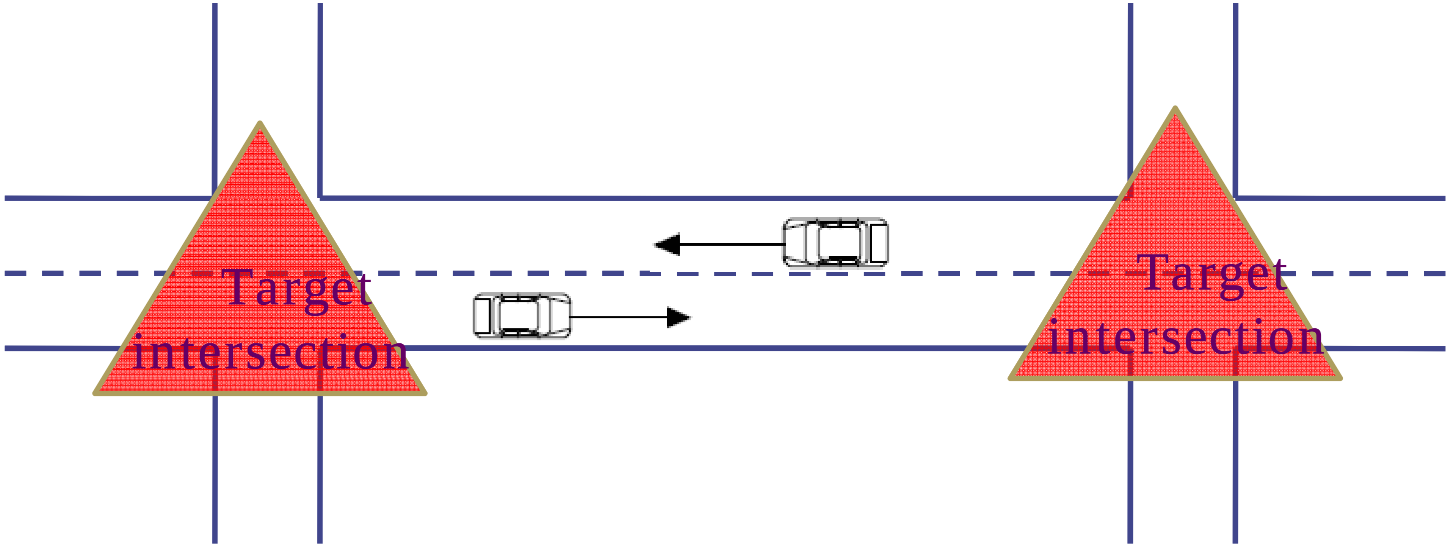
D-VADD only probes vehicles moving towards the direction whose priority is higher than or equal to the moving direction of current packet carrier.

Hybrid Probe (H-VADD)

- H-VADD works as follows:
  - Upon entering an intersection, H-VADD behaves like L-VADD. If a routing loop is detected, it immediately switches to use D-VADD until it exits the current intersection
  - H-VADD inherits the advantage of using the shortest forwarding path in L-VADD when there is no routing loop, and use D-VADD to address the routing loop problem of L-VADD

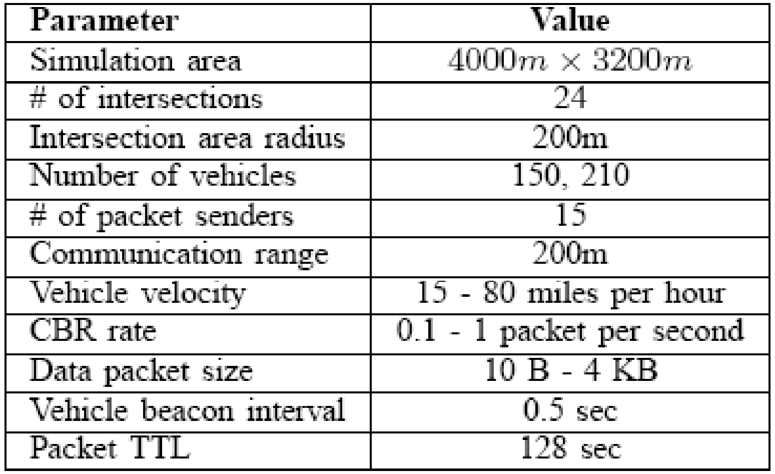
Data Forwarding in the StraightWay Mode

- If the identified target intersection is the intersection ahead, the packet is forwarded to the target intersection by GPSR
- If the identified target intersection is the intersection behind, the packet carrier keeps holding the packet, and waits for a vehicle in the opposite direction

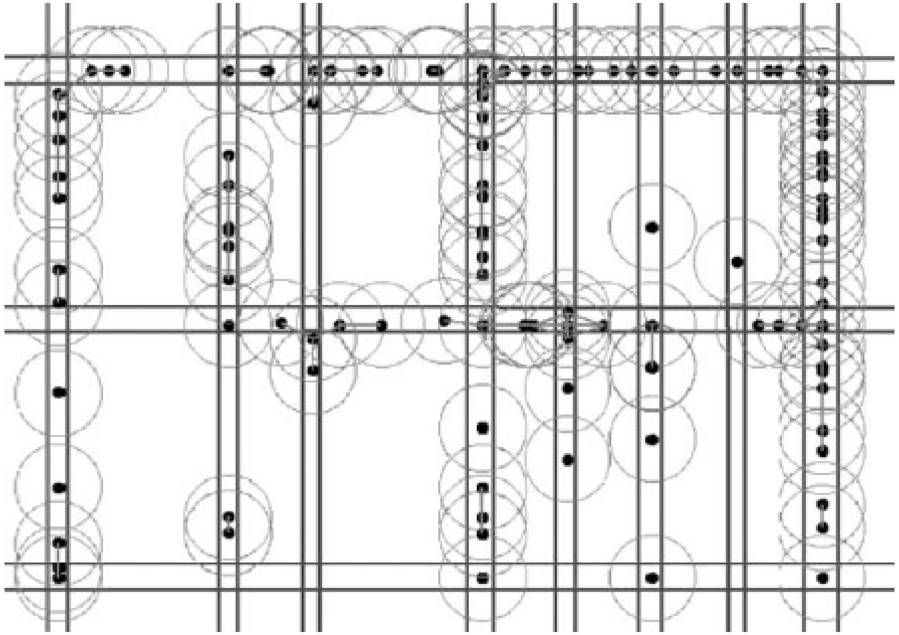


Performance Evaluations

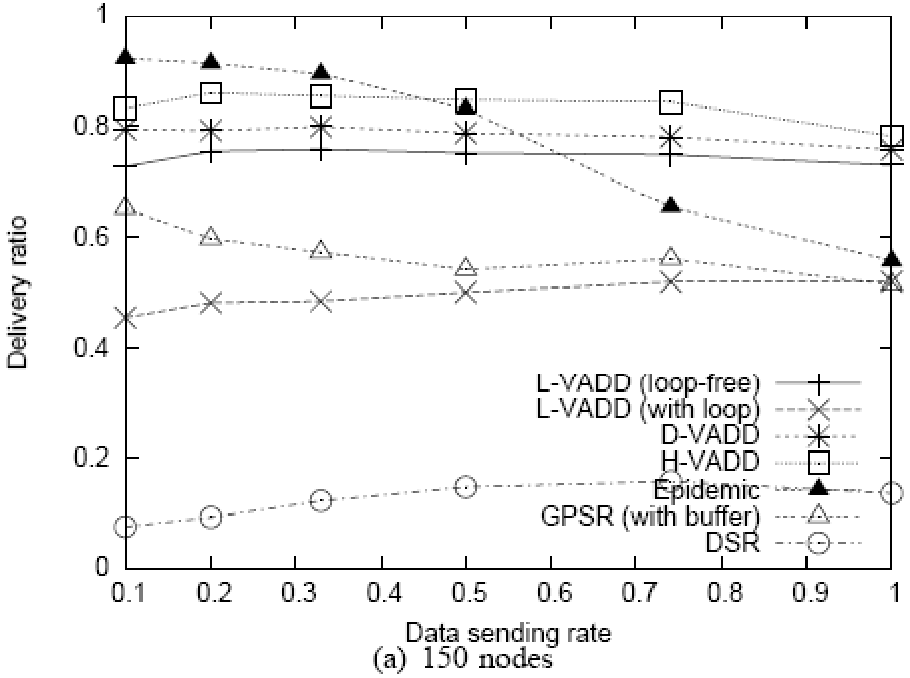
- Packet level simulation by ns2
- Metrics
  - Delivery ratio
  - Delivery delay
  - Data traffic overhead
- Compare the performance:
  - Epidemic Routing
  - GPSR (with buffer)
- Mobility Scenario
  - Traffic model derived from
    from TIGER database
  - Map data are transformed
    Into ns-2 readable data



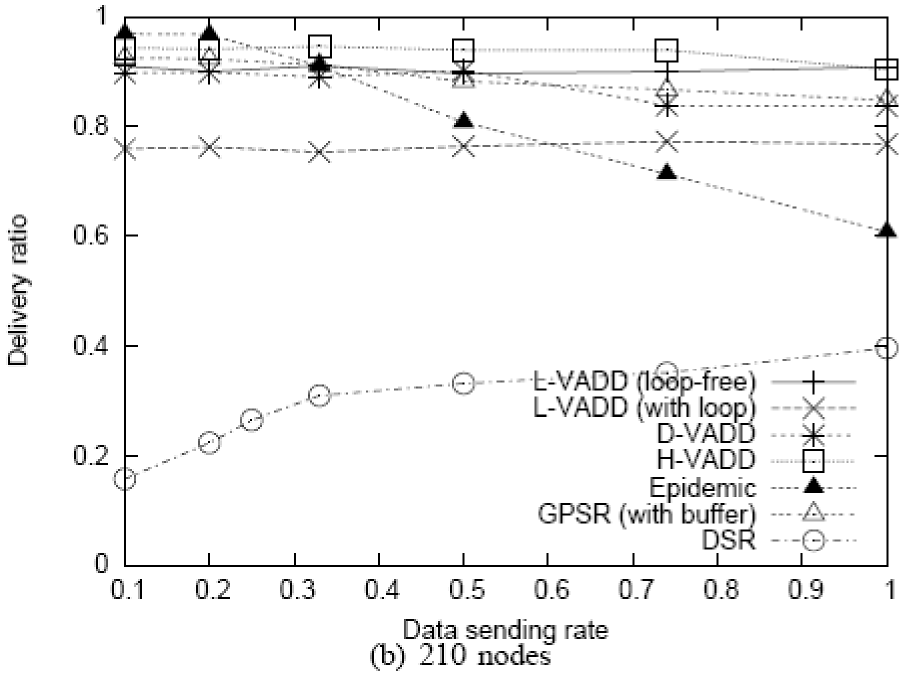
Parameter	Value
Simulation area	$4000m \times 3200m$
# of intersections	24
Intersection area radius	200m
Number of vehicles	150, 210
# of packet senders	15
Communication range	200m
Vehicle velocity	15 - 80 miles per hour
CBR rate	0.1 - 1 packet per second
Data packet size	10 B - 4 KB
Vehicle beacon interval	0.5 sec
Packet TTL	128 sec



Delivery ratio

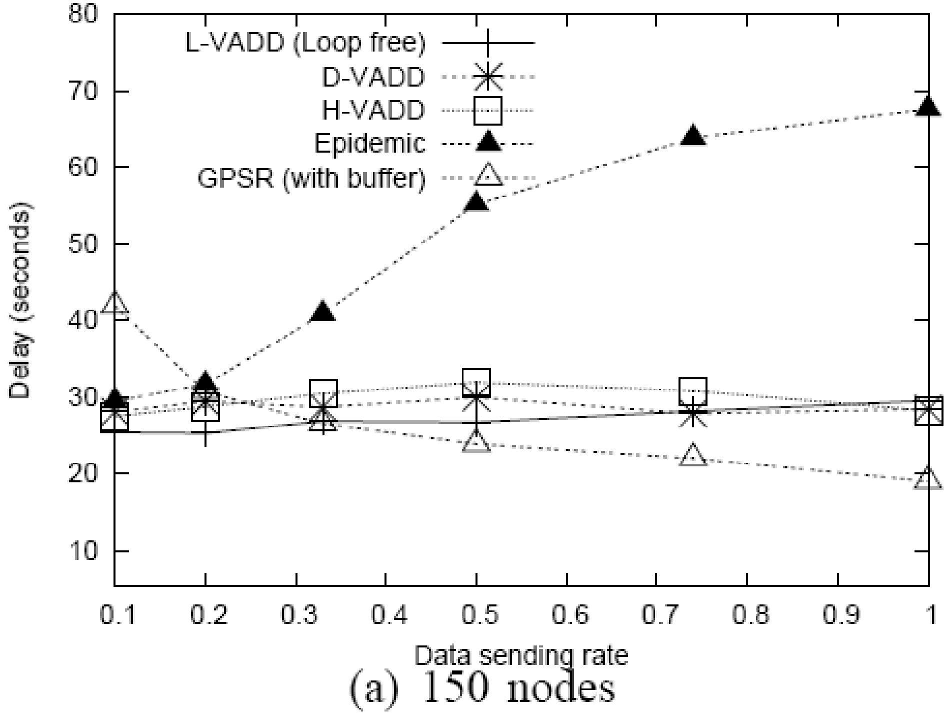


(a) Low node density

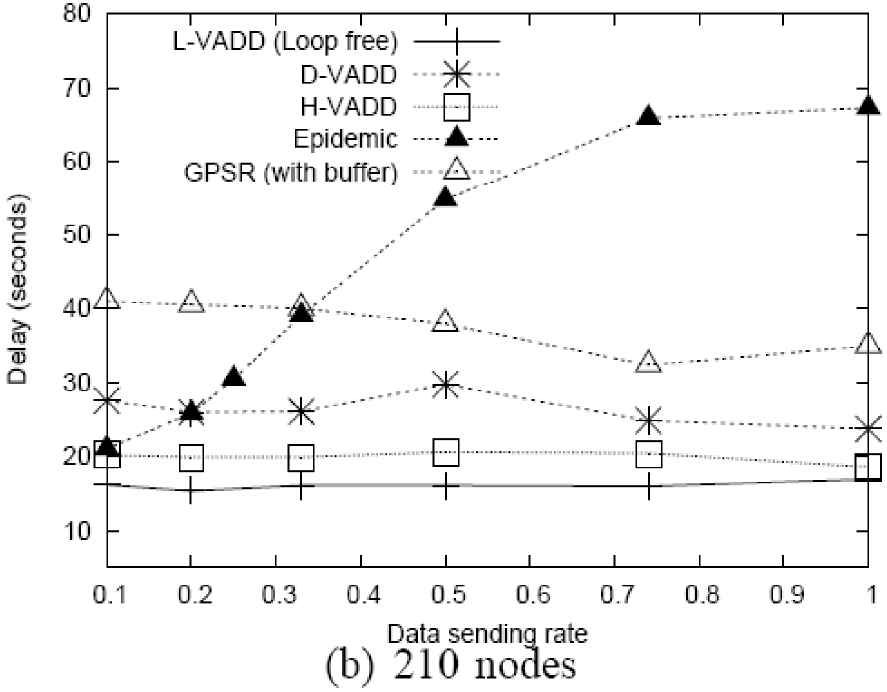


(b) High node density

Delivery delay



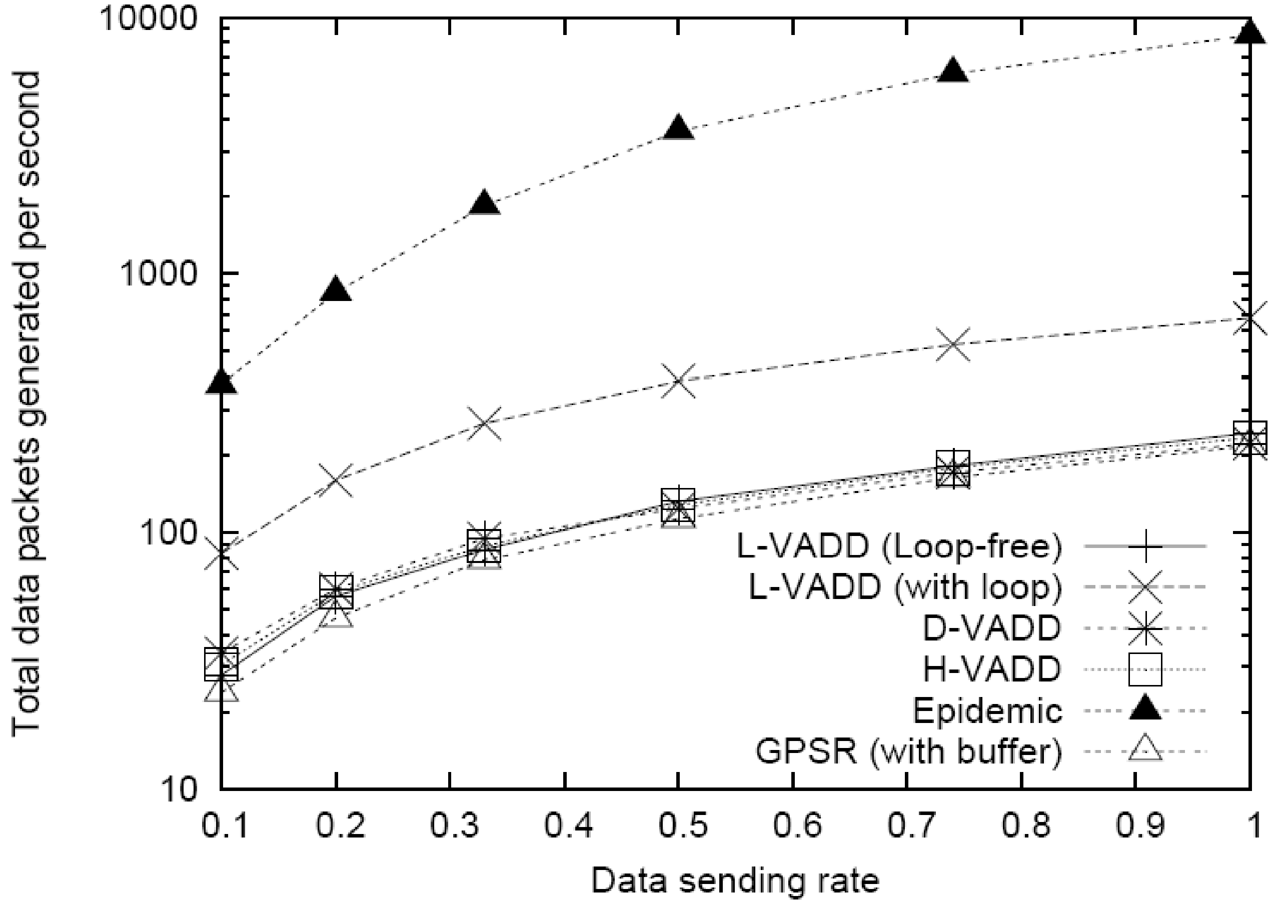
(a) Low node density



(b) High node density

Data traffic overhead

- Comparison between different protocols



Conclusions

- Different from existing carry and forward solutions, this work makes use of the predicable vehicle mobility, which is limited by the traffic pattern and road layout.

- Experimental results showed that the proposed VADD protocols outperform existing solutions in terms of packet delivery ratio, data packet delay, and traffic overhead.

Connectivity-Aware Routing (CAR) in Vehicular Ad Hoc Network

Valery Naumov and Thomas R.Gross

IEEE INFOCOM 2007

Section Outline

- Introduction
- Motivation
- Connectivity-Away Routing(CAR)
- Simulation
- Conclusions

Introduction

- Geographic routing protocol focus on geographically existing paths but do not take into account if a path between source and destination is populated.
  - Assume every node knows its *position*, *velocity*, and *direction* via GPS.
- This work presents a novel **position-based** routing scheme called Connectivity-Aware Routing (CAR)
  - is designed specifically for inter-vehicle communication in a city and/or highway environment.
  - CAR integrates locating destinations with finding connected paths **between source and destination**.
  - “Guards” help to track the current position of a destination.

Motivation

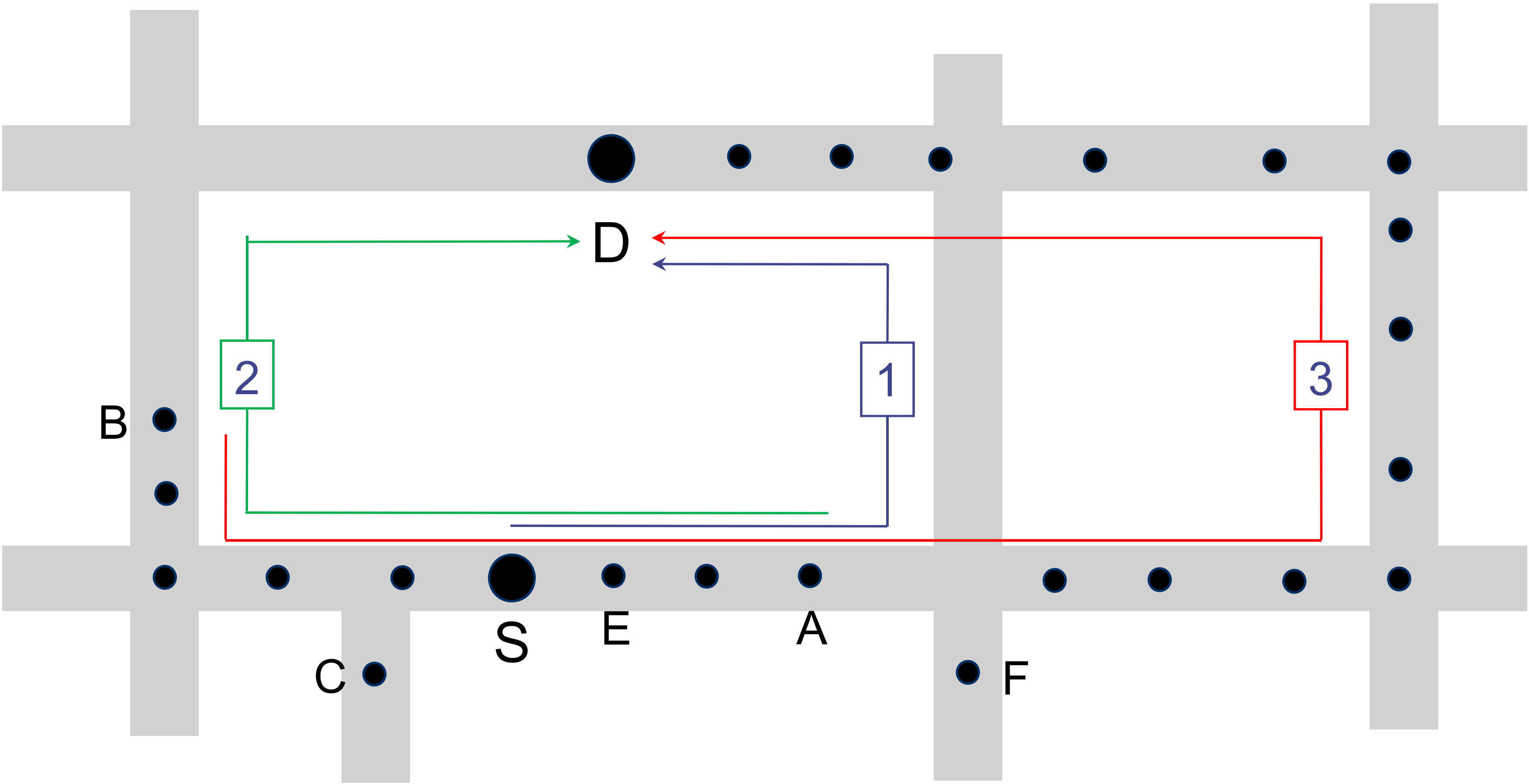


Fig. 1. Find path examples

Connectivity-Aware Routing (CAR)

- The CAR protocol consists of four main parts:
  1. Destination location and path discovery,
  2. Data packet forwarding along the found path,
  3. Path maintenance with the help of guards,
  4. Error recovery.

Neighbor tables and **adaptive** beaconing

- Adaptive beaconing
  - The HELLO beacon includes location, moving direction and speed.
  - The beaconing interval is changed according to the number of the registered nearby neighbors.
  - The fewer neighbors there are, the more frequent is a node’s HELLO beaconing.
  - Therefore Node 3 in Figure 2 beacons more frequently than Nodes 2 and 4 and much more frequently than Node 1.

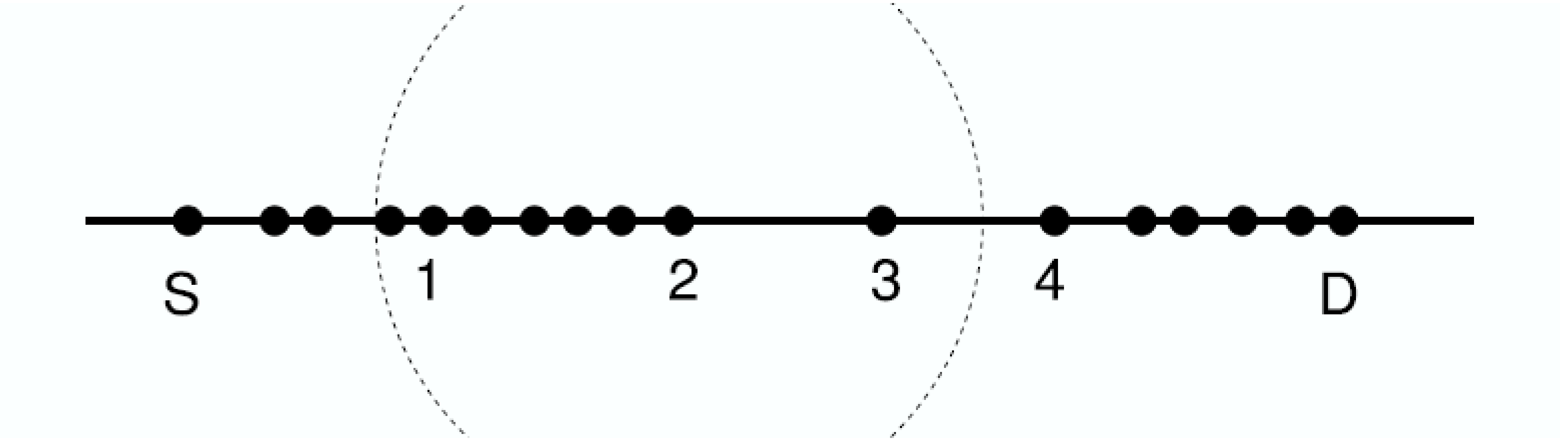


Fig. 2. Influence of the neighbor table accuracy. The accuracy of node 1 neighbor table is far less important for the communication between nodes *S* and *D* than those of nodes 2, 3, and 4.

Guards

- **Standing guards**
  - A standing guard (or guard for short) represents temporary state information that is tied to a **geographical area**, rather than to a specific node.
  - A guard is kept alive by the nodes located in the area.
  - A guard exists as an entry in the periodic HELLO beacon of a node.
  - This entry contains an id, a time-to-live (TTL) counter, a guarded position and radius, and some information that is naturally communicated to the neighbors by the nodes' usual periodic.
  - A node with a guard can filter or redirect packets or adds information to a packet that will eventually deliver this information to the packet's destination.
  - Once TTL reaches zero, the guard is removed from the node's HELLO beacon.

Path maintenance (standing guard)

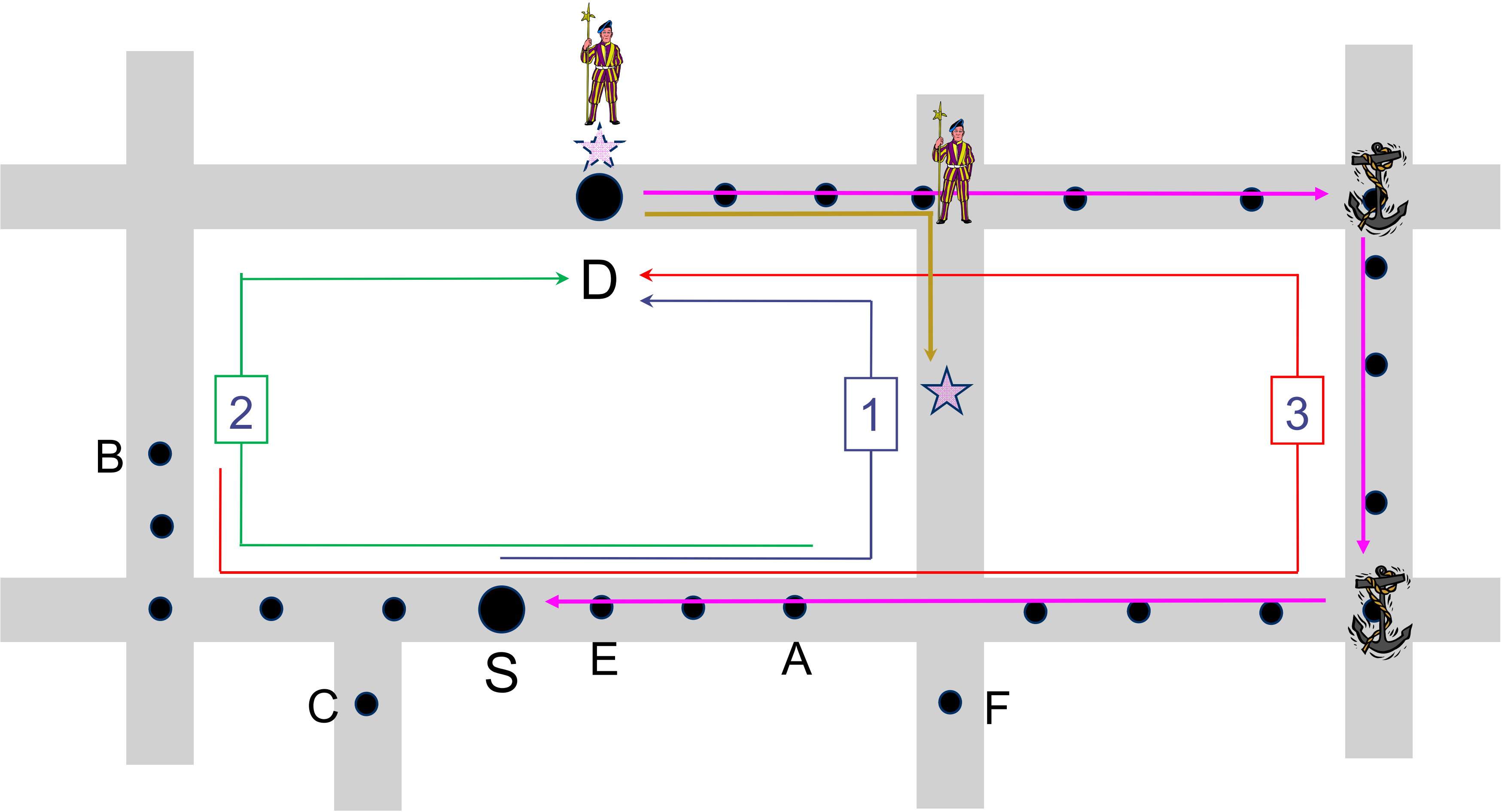


Fig. 1. Find path examples

Guards

- **Traveling guards**
  - A traveling guard contains also a velocity vector, in addition to the guarded position and radius.
  - Each node that receives a traveling guard records the time when the guard was received (or last sent).
  - As it is time for the next HELLO beacon, the node computes the **new guarded position** based on the **old guarded position**, the velocity vector of the guard, and the time passed since this guard was received.
  - Traveling guards allow the information carried by the guard to **travel with a certain speed** along the road. The age counter of the traveling guard is decreased with every retransmission.

Path maintenance (traveling guard)

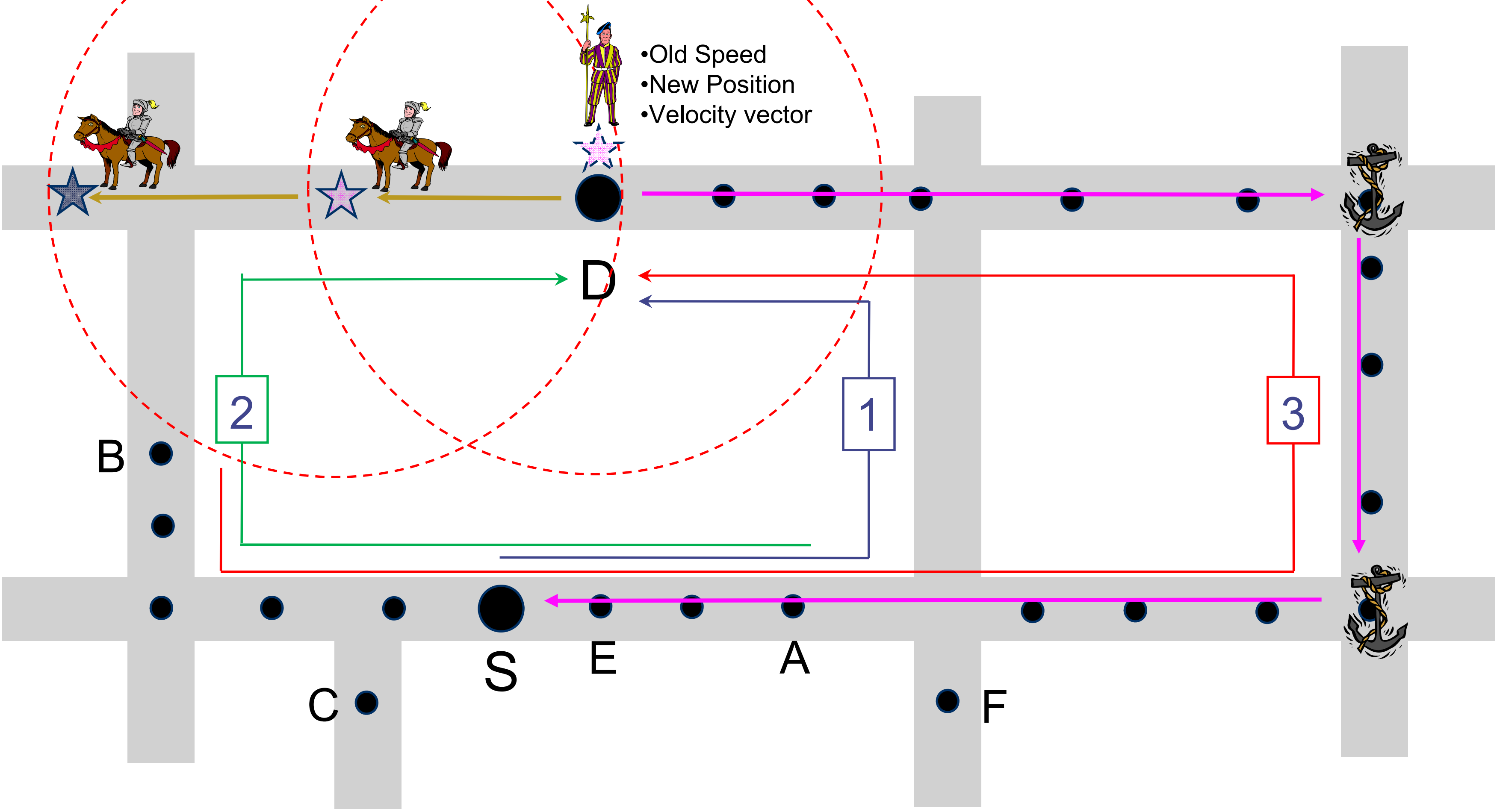


Fig. 1. Find path examples

1. Destination location discovery

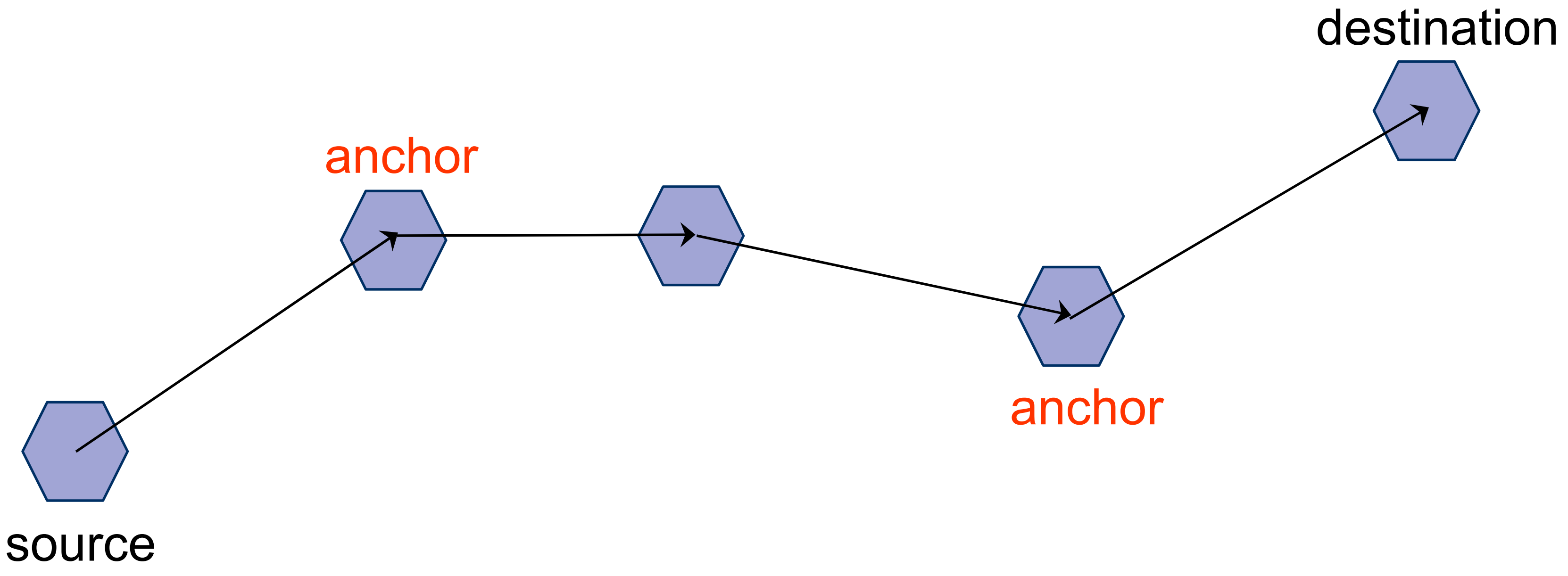
- Source initiates a ***PGB*** (Preferred Group Broadcasting) path discovery request.
- A path discovery packet consists of “PD id”, destination, previous forwarder’s coordinate/velocity vector, travel time, connectivity, anchor.
- To estimate the connectivity on the traveled path, each forwarder changes three other packet fields:
  - “Number of hops”
  - “Average number of neighbors”
  - “Minimum number of neighbors”

Cont.

- To find a destination and a path to it, CAR uses PGB in data dissemination mode. PGB optimizes broadcasts specifically for VANETs, it reduces control messages overhead by eliminating redundant transmissions.
  - C. Perkins and E. Royer, “Ad-hoc on-demand distance vector routing,” in *Proc. IEEE WMCSA’99*, Feb 1999, pp. 90–100.
  - A study of VANETs for realistic scenarios shows that **AODV** [17] (not a GR protocol) combined with Preferred Group Broadcasting (**PGB**), an optimization of broadcasting, provides better results than **GPSR**, a GR protocol, GPSR [18] even when GPSR is improved with **Advanced Greedy Forwarding (AGF)** [5].

Destination location discovery (cont.)

- If two velocity vectors angle > 18°, **anchor** is set.
  - Anchor contains coordinates and velocity vector of current node and previous node.



2. CAR route reply

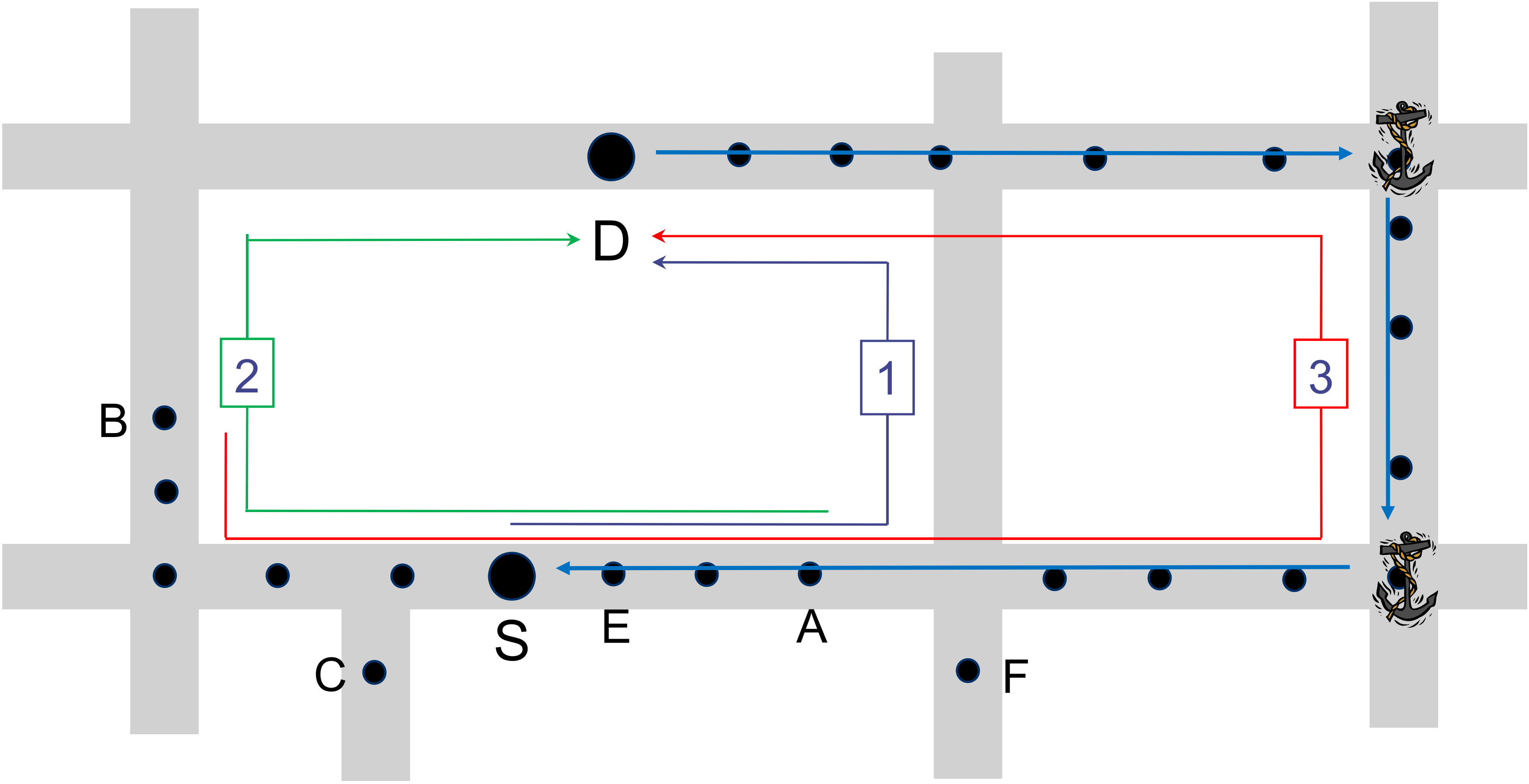


Fig. 1. Find path examples

Cont.

- Eventually a route reply is sent from the destination back to the source.
- A route reply is a unicast packet that contains the destination's coordinates and velocity vector, together with the information collected by the route request on its way to destination.
- **AGF** is used to forward the route reply back to the source via the **recorded anchor points**.
    - V. Naumov, R. Baumann, and T. Gross, "An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces," in *Proc. ACM MOBIHOC'06*, 2006, pp. 108–119.
- Data packets are forwarded in a greedy manner toward the destination through the set of anchor points using the same AGF algorithm.

Advantages of this approach to discover a destination's location

1. it finds the paths that are not only geographically possible but exist in reality;
2. it takes the connectivity into account;
3. there is no need for expensive trial-and-error route tests based on data packet transmissions.
4. only source-destination pairs keep anchored paths to each other.

Greedy forwarding over the **anchored path**

- The CAR protocol extends AGF to work with anchor points. AGF assumes that both the source and the destination inform each other about their velocity vectors.
- Instead of forwarding a data packet to a neighbor that is geographically closer to the destination, a neighbor closest to the next anchor point is chosen.
- Each forwarding node relays to anchor if the distance is less than half coverage.
  - To avoid multiple attempts to gradually get closer to the next anchor point.
  - each forwarding node checks if its position and the position of the next anchor point is separated by less than half the node’s coverage range.
- The process continues until the packet reaches its destination.

3. Path maintenance

- If an end node (source or destination) changes position or direction, **standing guard** will be activated to maintain the path.
  - Standing guard is tied to a geographical area, rather than a specific node.
  - The guard contains the old and the new velocity vectors of this node.
  - Right after activating a guard the node sends a notification packet to source.

Path maintenance

- If end node changes direction against the direction of communication, **traveling guard** will be activated.
  - A traveling guard contains velocity vector, position and radius.
  - A traveling guard runs as end node’s old direction and speed, and reroute the packets to the destination.
- If an end point node notices that due to speed changes.
  - Its true position become separated by more than 60% of the average coverage range, the node broadcasts a traveling guard, letting the guard travel with the old speed of the node.

4. Routing error recovery

- Error may occur due to:
  - A temporary gap between two vehicles or raised interference.
  - Long-term disconnection.
  - A packet arrives the estimated position but can not find the destination.

Routing error recovery

- Timeout algorithm with active waiting cycle.
  - Tell other nodes there is a disconnection, and buffer the packets.
  - Try to detect next-hop node.

- Walk-around error recovery
  - If the timeout algorithm is failed, the node will report to the source and **starts a local destination location discovery process**.
  - No matter the destination discovery succeed or not, the result will be reported to the source.

Conclusion

- The CAR protocol is based on PGB and AGF to provide a scalable low overhead routing algorithm for inter-vehicle communication both in the city and on the highway.
- CAR is able to locate destinations without using an idealized location service. Rather than relying solely on knowledge of the road layout, CAR adapts to current conditions to find a route with sufficient connectivity so as to maximize the chance of successful delivery.
- CAR is presented here as a unified protocol but the key concepts can also be incorporated into other protocols.
  - E.g., incorporating CAR's adaptive beaconing mechanism into GPSR improves GPSR's performance by up to 30%.

DIR: Diagonal-Intersection-Based Routing Protocol for Vehicular Ad Hoc Networks

Yuh-Shyan Chen and Ci-Yi Pan

Department of Computer Science and Information Engineering,
National Taipei University

Telecommunication System, Vol. 46, Issue 4, May 2011.

Outline

- Introduction
- Related work
- Motivation and basic idea
- Diagonal intersection based routing protocol
- Performance analysis
- Conclusions

Introduction

- Intelligent transportation system (ITS)
    - An emergent system to integrate with the advanced electronics, communications, information, and wireless sensor technology to provide safety and comfort of drivers in highway and urban
  - Road-to-vehicle communications (RVC)
  - Inter-vehicle communications (IVC)
- Wireless routing technologies
  - MANETs
  - VANETs
    - Very high mobility
    - Network topology changeable
    - Temporary network fragmentation

Introduction (Cont.)

- The goal of VANETs
  - To develop a quick and efficient information for the user
- The proposed diagonal-intersection-based routing (DIR) protocols
  - Forward the packet to a road with the lowest packet delivery delay

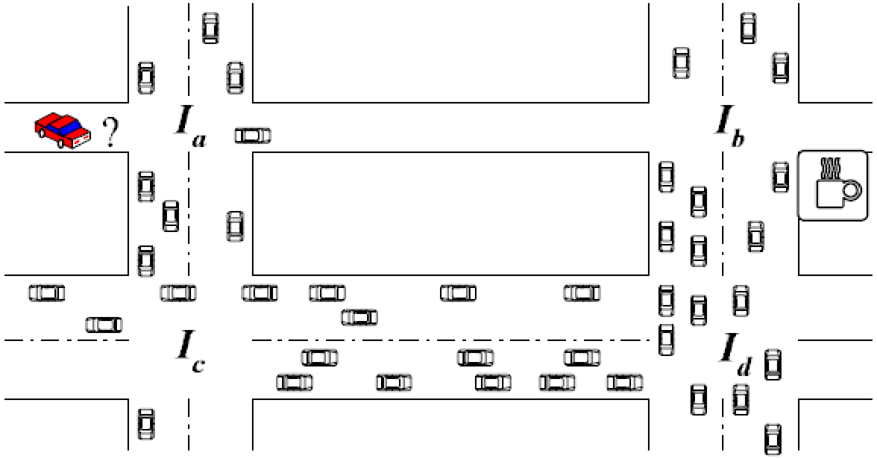
Related Work

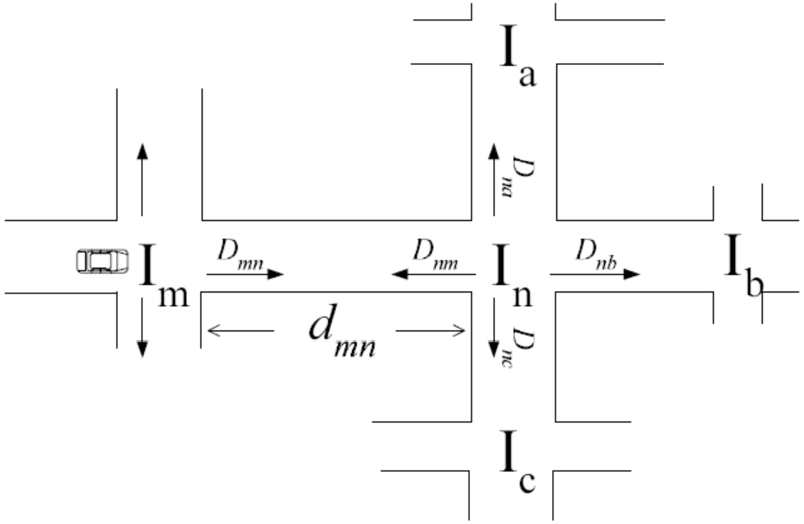
- Existing routing results in VANETs
  - Christian Lochert *et al*., “Geographic Routing in City Scenarios”. *ACM SIGMOBILE Mobile Computing and Communications Review*, 2005.
    - Proposed the position-based routing approach (greedy forwarding) such that an intermediate node forwards a packet to the direct neighbor which is closest to the geographic position of the destination.
  - Naumou *et al*., “An Evaluation of Inter-Vehicle Ad Hoc Networks based on realistic vehicular traces”. *ACM International Symposium on Mobile Ad Hoc Networking and Computing* (MOBIHOC2006), Florence, Italy, pp.108-119, May 2006.
    - Incorporated a velocity vector of speed and direction to improve the GPSR protocol by accurately determining the location of a destination.
  - Jerbi *et al*., “An Improved Vehicular Ad Hoc Routing Protocol for City Environments”. *IEEE International Conference on Communications* (ICC 2007), Glasgow, Scotland, pp. 3972-3979, 24-28 June 2007.
    - An improved greedy strategy used to forward packers between two junctions

Related Work (Cont.)

- Zhao *et al*., "VADD：Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks" *IEEE International Conference on Computer Communications* (INFOCOM 2006), Barcelona, Caralunya, Spain, pp. 1-12, 23-29 April 2006.
  - Adopt the idea of carry and forward
  - VADD protocol to forward the packet to the best road with the lowest data delivery delay
- Naumov *et al*., "Connectivity-Away Routing (CAR) in Vehicular Ad Hoc Networks". *IEEE International Conference on Computer Communications* (INFOCOM 2007), Anchorage,Alaska, USA, pp. 1919-1927, 6-12 May 2007.
  - The main property of CAR protocol is the ability to not only locate positions of destinations but also to find connected paths between source and destination vehicle

Zhao *et al.*'s VADD：Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks (**INFOCOM 2006**)

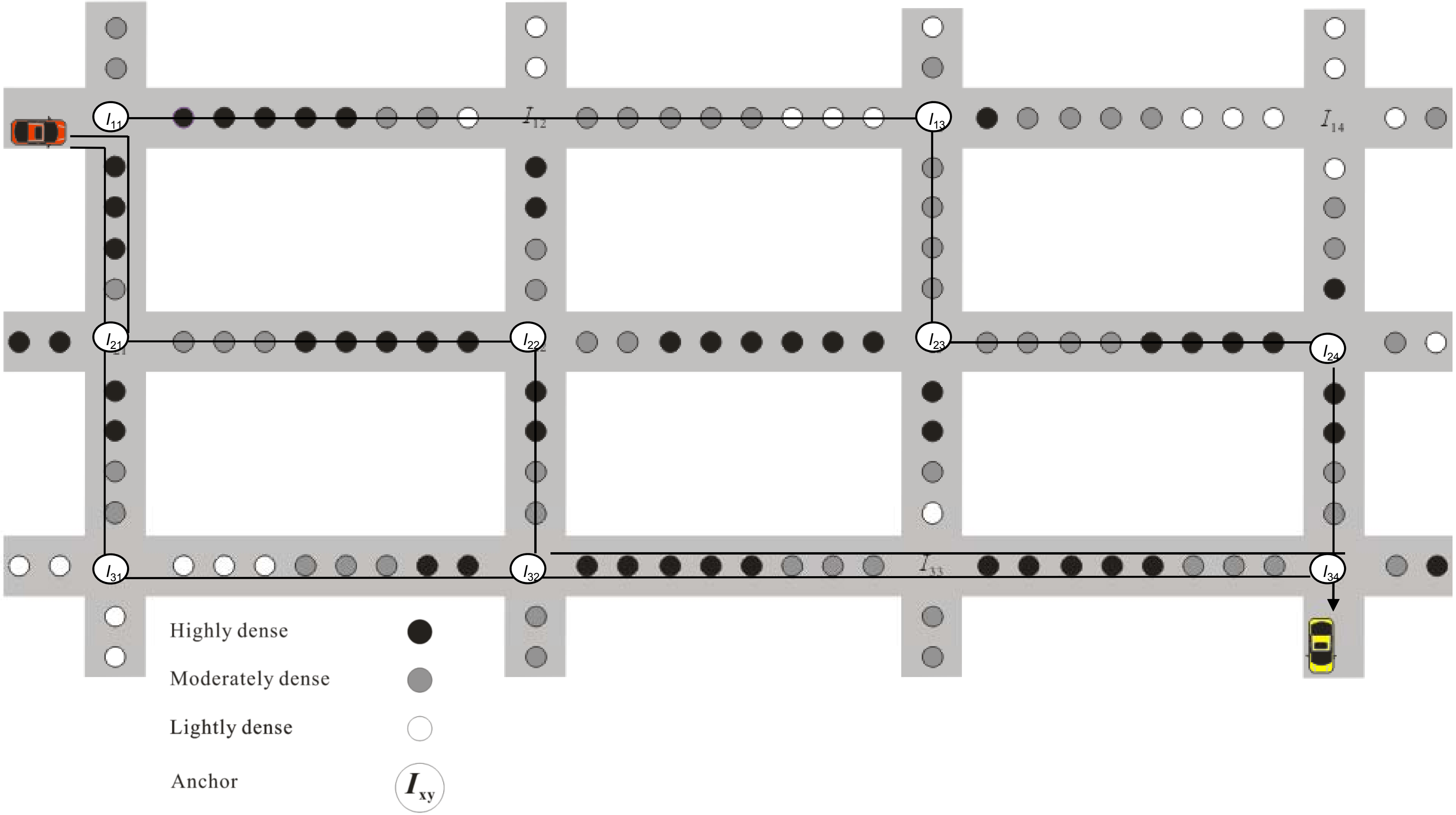




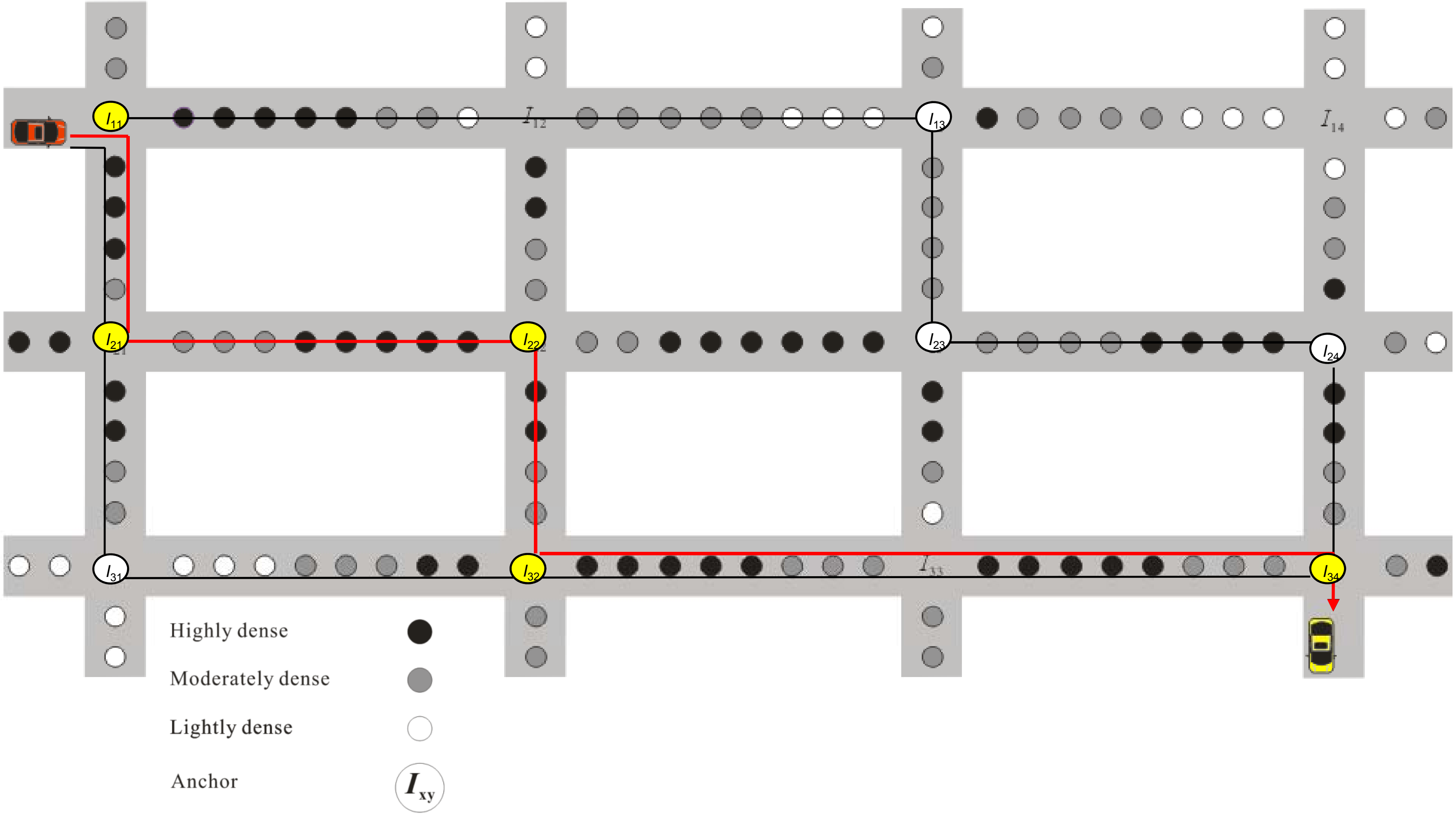
$$d_{ij} = (1 - e^{-R\cdot\rho_{ij}}) \cdot \frac{l_{ij}\cdot c}{R} + e^{-R\cdot\rho_{ij}} \cdot \frac{l_{ij}}{v_{ij}}$$

r_{ij}: the road from I_i to I_j.
l_{ij}: the Euclidean distance of r_{ij}.
ρ_{ij}: the vehicle density on r_{ij}.
v_{ij}: the average vehicle velocity on r_{ij}.
d_{ij}: the expected packet forwarding delay from I_i to I_j.

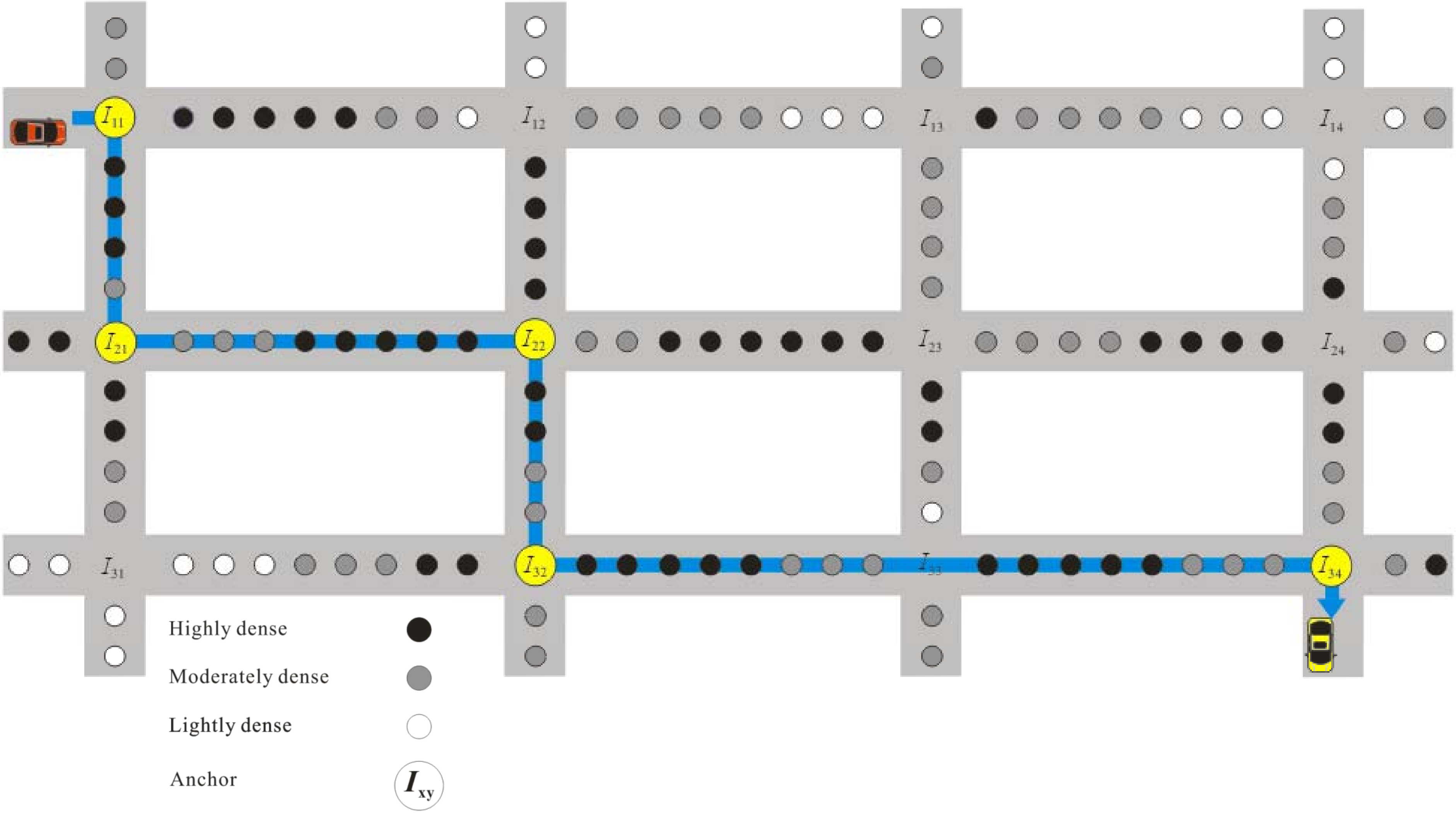
Connectivety-Away Routing (CAR) in Vehicular Ad Hoc Networks”. (**INFOCOM 2007**)



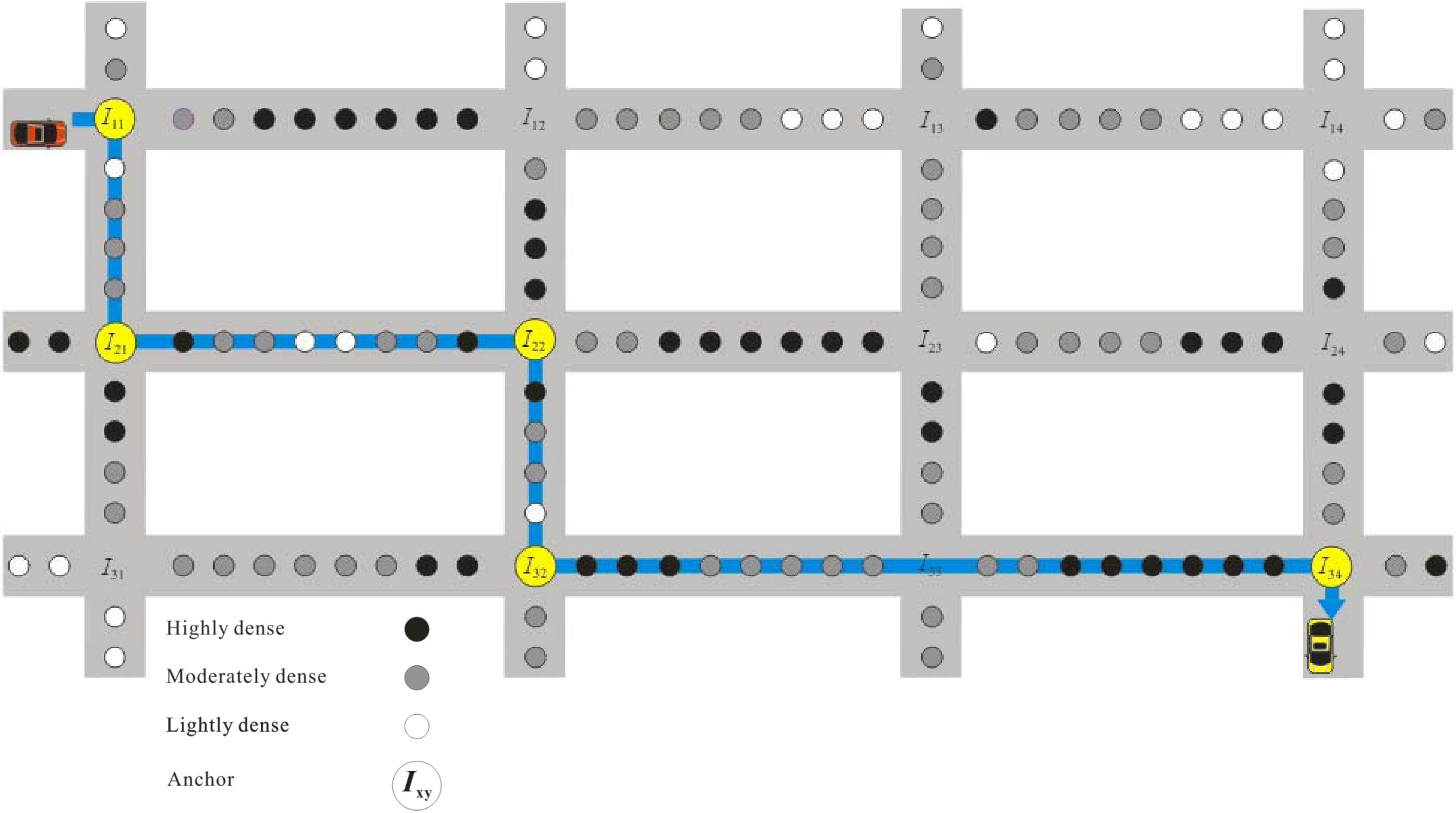
CAR Protocol



CAR Protocol



CAR protocol cannot adjust different sub-path when the traffic status is changed.



Motivation and Basic Idea

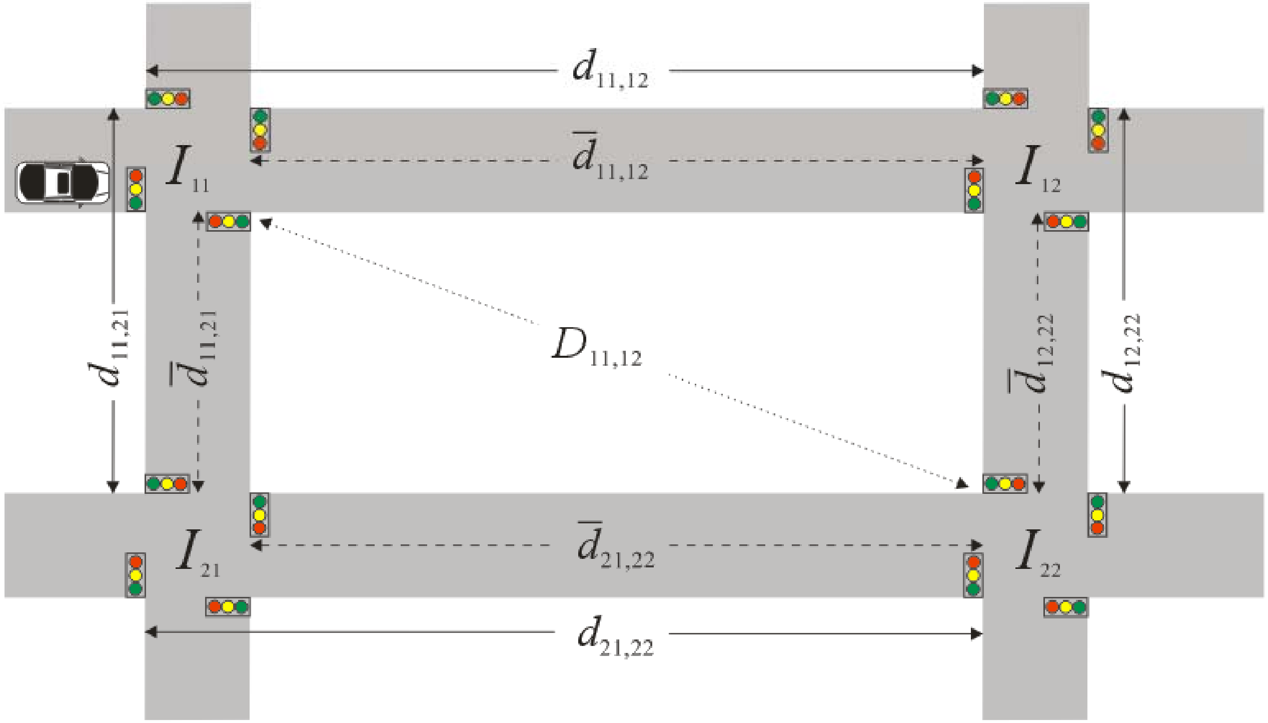
- The CAR protocol works with anchor point
    - Data forwarding to a neighbor that is geographically closer to the destination
        - CAR path constructed by a series of anchor points
        - A neighbor closest to the next anchor point is chosen
    - No path adjustability capability
    - Without considering traffic light model
- The DIR protocol works with diagonal anchor point
    - Low expected packet forwarding delay
        - Calculated to choose one sub-paths
    - Auto-adjustability capability
        - Search for a routing path with the lower data forwarding delay
    - Traffic light model is considered

System Model

- New delay model is modified from the VADD delay model

$$d_{x_1y_1,x_2y_2} = P_{x_1y_1} \cdot \bar{d}_{x_1y_1,x_2y_2} + \left(1 - P_{x_1y_1}\right) \cdot C_{x_1y_1} \cdot \alpha_{x_1y_1}$$

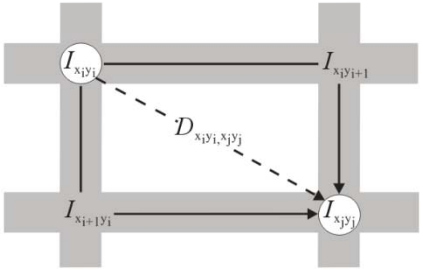
$$\bar{d}_{x_1y_1,x_2y_2} = \left(1 - e^{-R\cdot\rho_{x_1y_1,x_2y_2}}\right) \cdot \frac{l_{x_1y_1,x_2y_2} \cdot c}{R} + e^{-R\cdot\rho_{x_1y_1,x_2y_2}} \cdot \frac{l_{x_1y_1,x_2y_2}}{v_{x_1y_1,x_2y_2}}$$



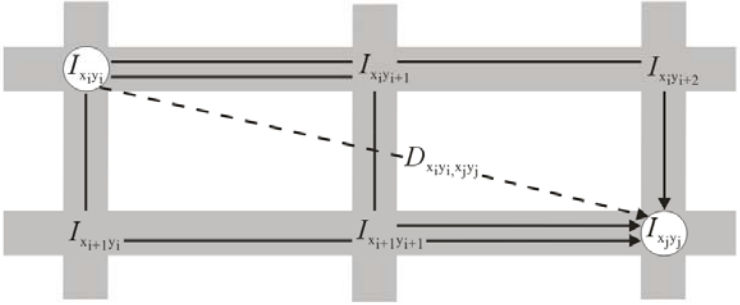
$$D_{m,n} = d_{m,n} + \sum_{j\in N(n)} \left(P_{n,j} \times D_{n,j}\right)$$

System Model (Cont.)

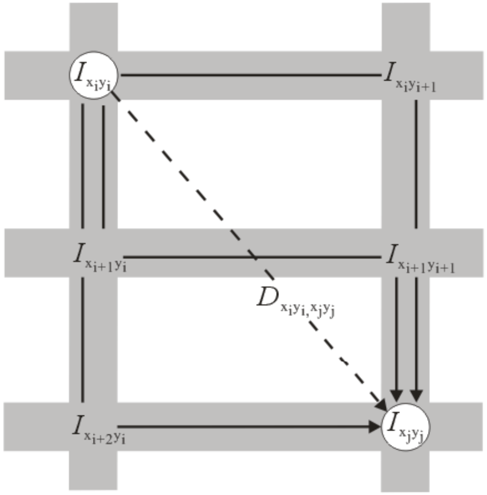
- Three different scenarios



(a)



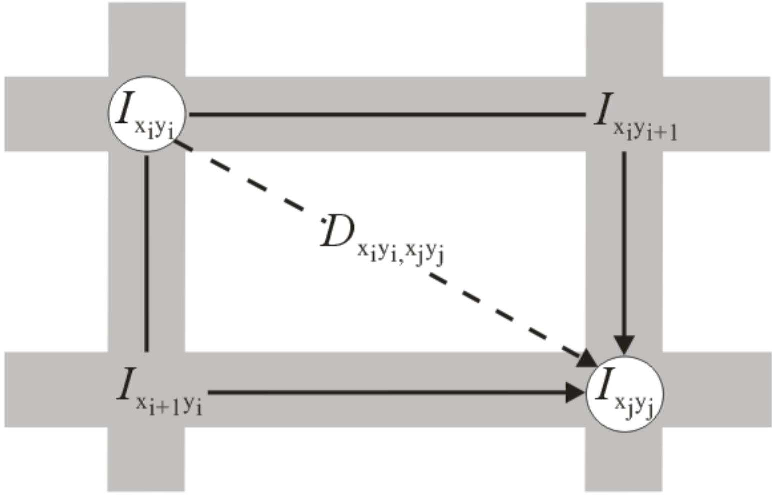
(b)



(c)

Scenario

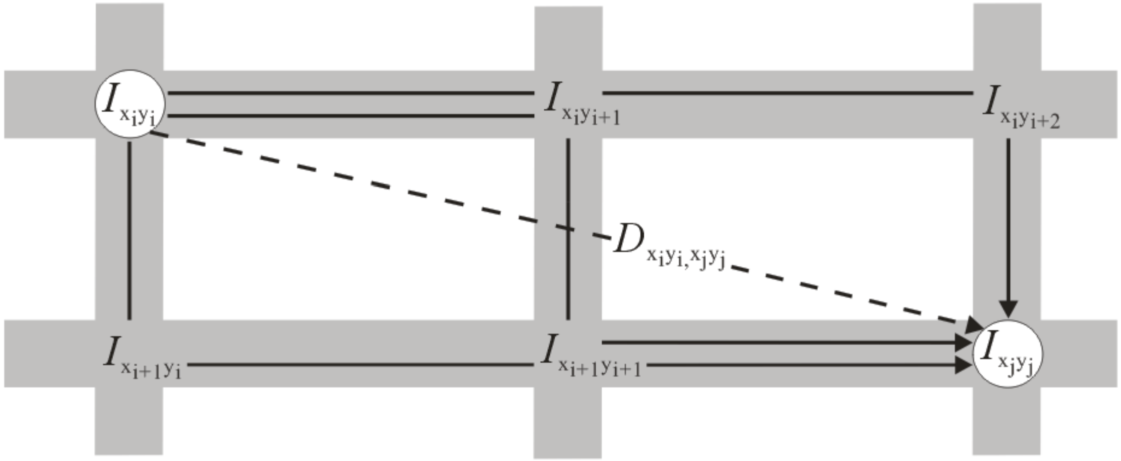
- $D_{m,n} = d_{m,n} + \sum_{j \in N(n)} (P_{n,j} \times D_{n,j})$
- $D_{xy,x+1y+1}$=min{$d_{xy,x+1y}$+$d_{x+1y,x+1y+1}$, $d_{xy,xy+1}$+$d_{xy+1,x+1y+1}$}
  - $I_{xy} \longrightarrow I_{x+1y} \longrightarrow I_{x+1y+1}$
  - $I_{xy} \longrightarrow I_{xy+1} \longrightarrow I_{x+1y+1}$



(a)

Scenario

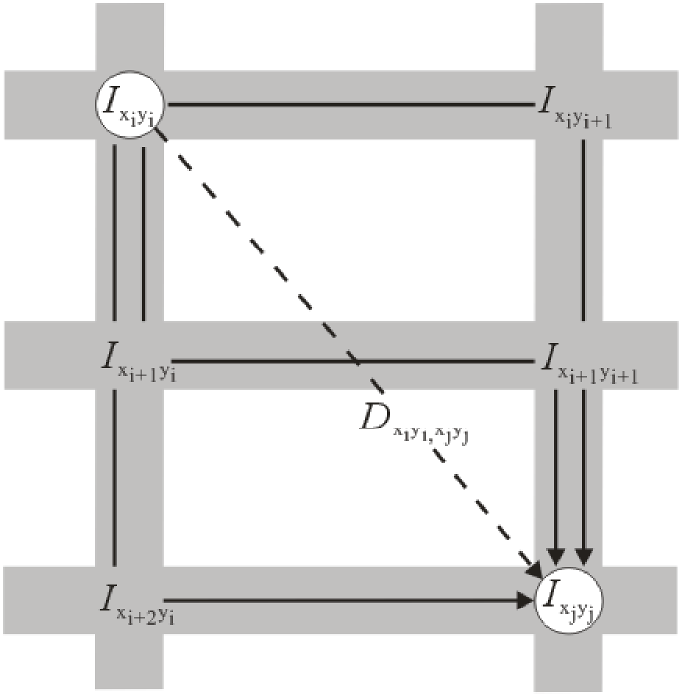
- $D_{m,n} = d_{m,n} + \sum_{j \in N(n)} (P_{n,j} \times D_{n,j})$
- $D_{xy,x+1y+2}$=min{$d_{xy,x+1y}$+$d_{x+1y,x+1y+1}$+$d_{x+1y+1,x+1y+2}$, $d_{xy,xy+1}$+$d_{xy+1,x+1y+1}$+$d_{x+1y+1,x+1y+2}$, $d_{xy,xy+1}$+$d_{xy+1,xy+2}$+$d_{xy+2,x+1y+2}$ }
  - $I_{xy} \longrightarrow I_{x+1y} \longrightarrow I_{x+1y+1} \longrightarrow I_{x+1y+2}$
  - $I_{xy} \longrightarrow I_{xy+1} \longrightarrow I_{x+1y+1} \longrightarrow I_{x+1y+2}$
  - $I_{xy} \longrightarrow I_{xy+1} \longrightarrow I_{xy+2} \longrightarrow I_{x+1y+2}$



(b)

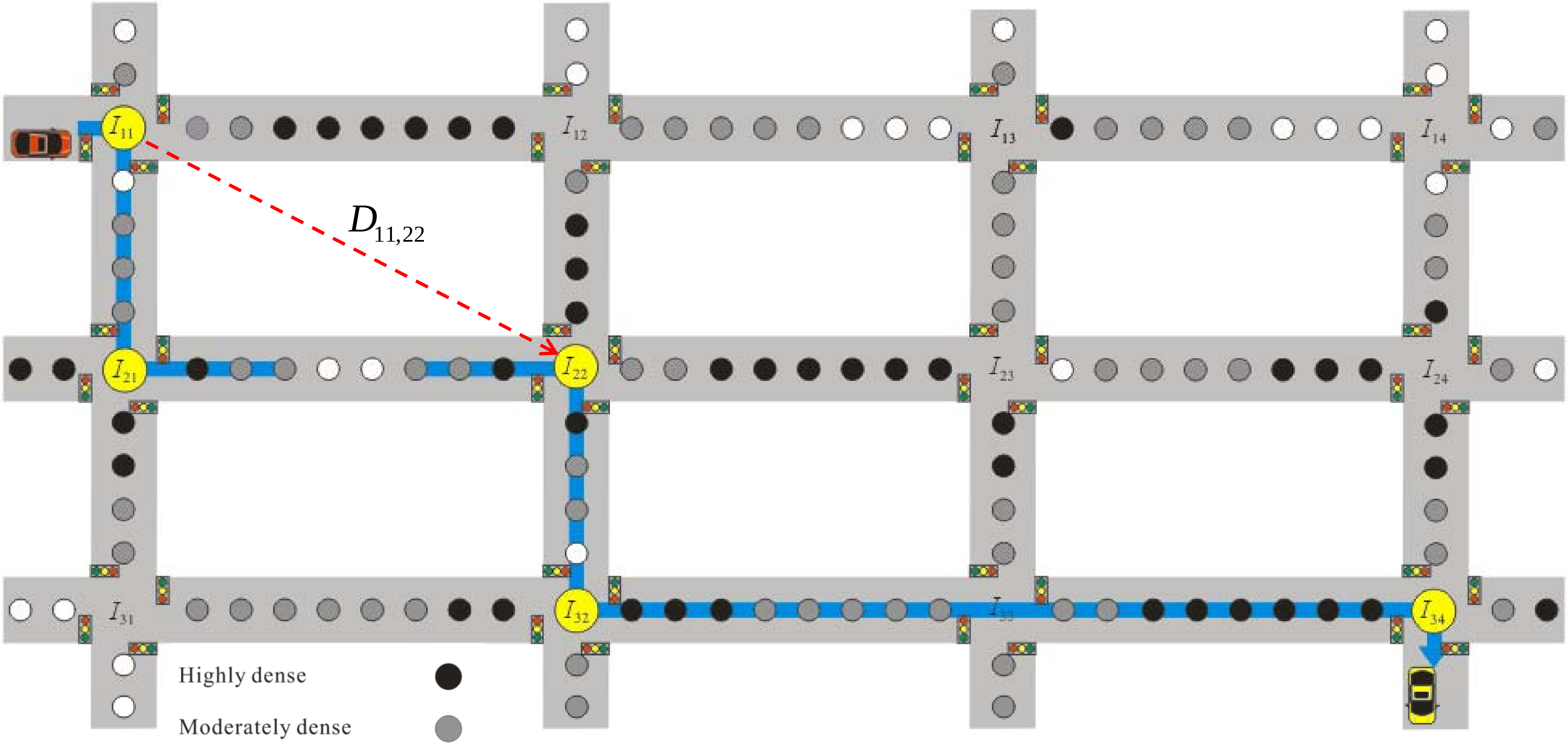
Scenario

- $D_{m,n} = d_{m,n} + \sum_{j \in N(n)} (P_{n,j} \times D_{n,j})$
- $D_{xy,x+2y+1} = \min\{d_{xy,x+1y} + d_{x+1y,x+2y} + d_{x+2y,x+2y+1}, d_{xy,x+1y} + d_{x+1y,x+1y+1} + d_{x+1y+1,x+2y+1}, d_{xy,xy+1} + d_{xy+1,x+1y+1} + d_{x+1y+1,x+2y+1}\}$
  - $I_{xy} \longrightarrow I_{x+1y} \longrightarrow I_{x+2y} \longrightarrow I_{x+2y+1}$
  - $I_{xy} \longrightarrow I_{x+1y} \longrightarrow I_{x+1y+1} \longrightarrow I_{x+2y+1}$
  - $I_{xy} \longrightarrow I_{xy+1} \longrightarrow I_{x+1y+1} \longrightarrow I_{x+2y+1}$



(c)

Diagonal-Intersection-Based Routing (DIR) Protocol



Highly dense ●

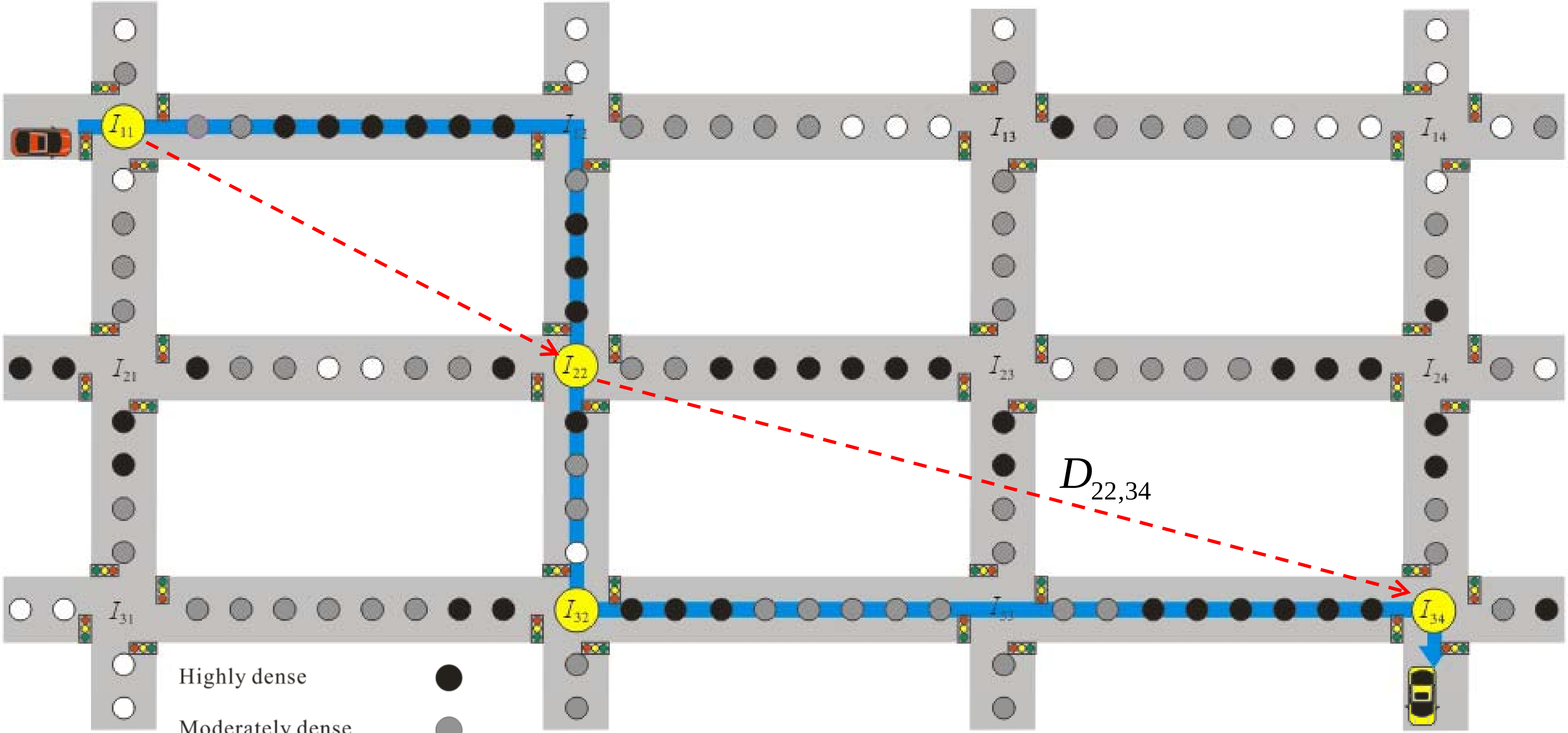
Moderately dense ●

Lightly dense ○

Anchor I_{xy}

$$D_{11,22} = \min\left(d_{11,12} + d_{12,22}, d_{11,21} + d_{21,22}\right)$$

Path Adjusts to Adapt Current Traffic



Highly dense ●

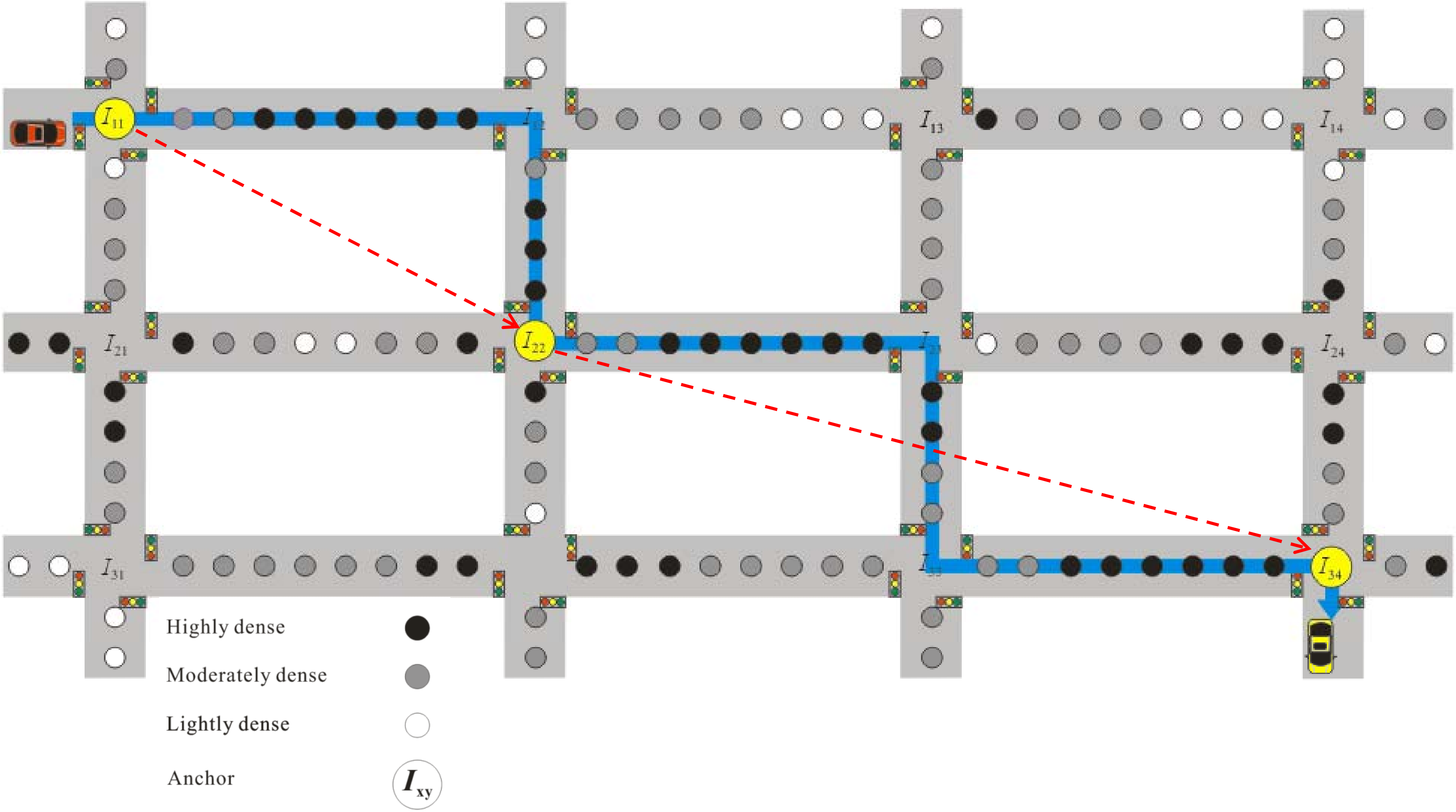
Moderately dense ◍

Lightly dense ○

Anchor I_{xy}

$$D_{22,34} = \min\left(d_{22,23} + d_{23,24} + d_{24,34}, d_{22,23} + d_{23,33} + d_{33,34}, d_{22,32} + d_{32,33} + d_{33,34}\right)$$

Diagonal-Intersection-Based Routing (DIR) Protocol



Diagonal-Intersection-Based Routing (DIR) Protocol

- Destination discovery
  - Algorithm A (consider fixed anchor-point list)
  - Algorithm B (dynamically consider anchor-point list)
- Data forwarding
- Path maintenance
  - Source and destination are fixed
  - Source and destination are mobile

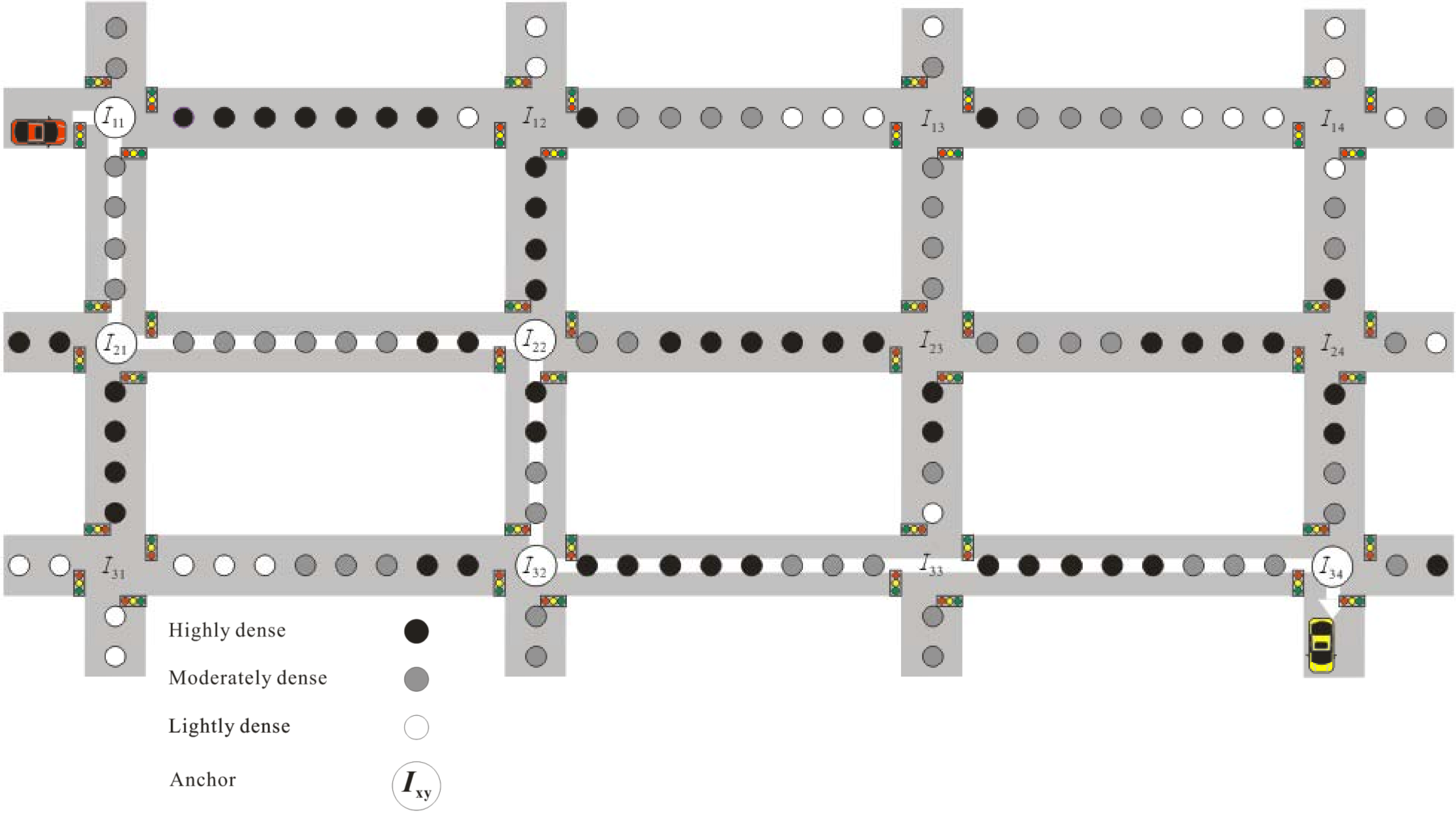
DIR-Algorithm A

- To construct a DIR route with least-delay
  - S1 ：Let $DI_1 = I_{x1y1} = I_1$ is diagonal-intersection list $DIL = [DI_1]$
  - S2 ：Let $d_{I_{xiyi}}(I_{x\alpha y\beta}) = D_{xiyi,xi'yi'} + d_{I_{xi'yi'}}(I_{x\alpha y\beta})$

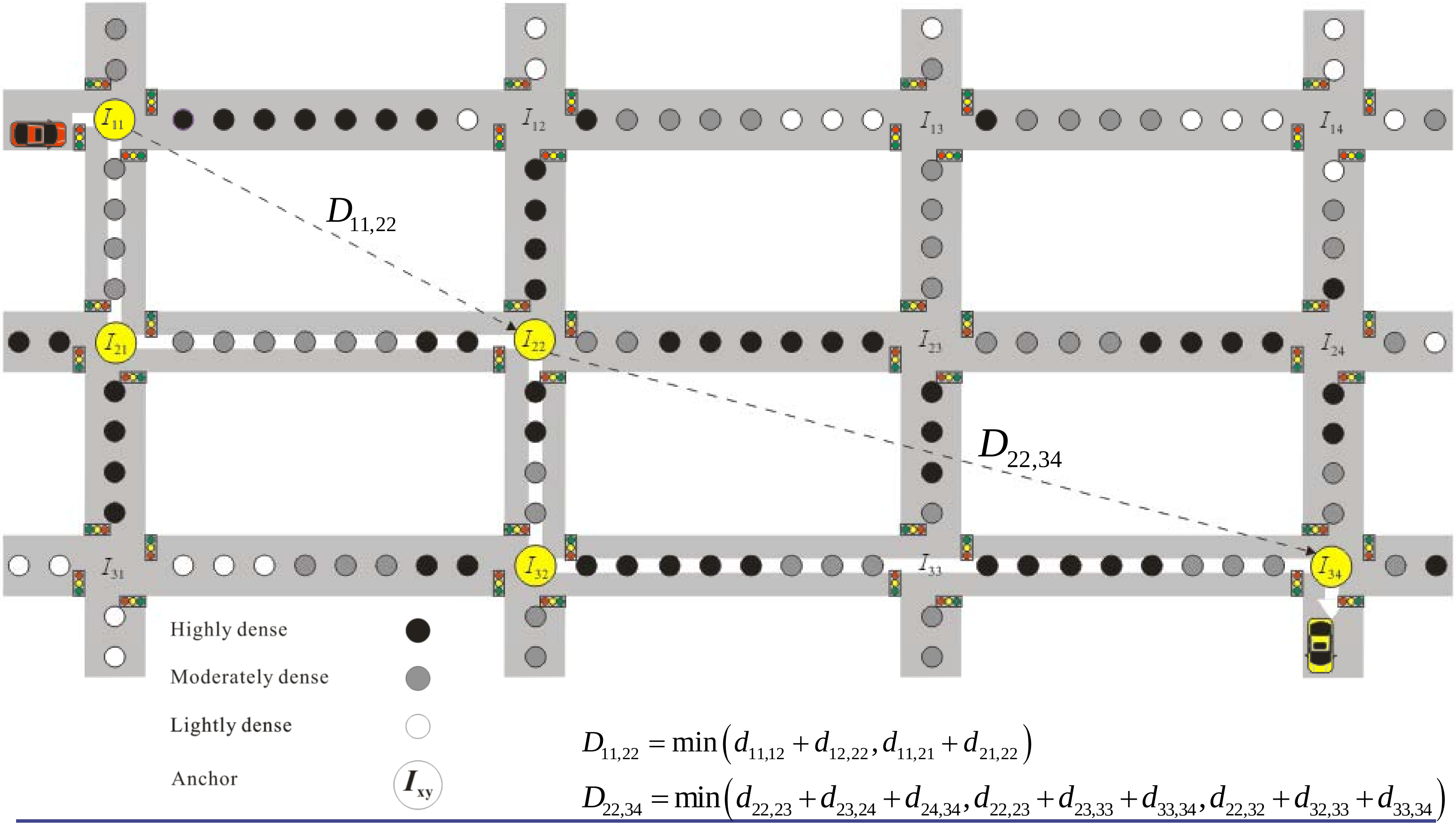
$$\begin{cases} if\ |x_i - x_i^{'}| = 1 \cap |y_i - y_i^{'}| = 1\ or \\ if\ |x_i - x_i^{'}| = 2 \cap |y_i - y_i^{'}| = 1\ or \\ if\ |x_i - x_i^{'}| = 1 \cap |y_i - y_i^{'}| = 2, \end{cases}$$

  - S3：Let $I_{xiyi} = I_{xi'yi'}$

Example of DIR-Algorithm A



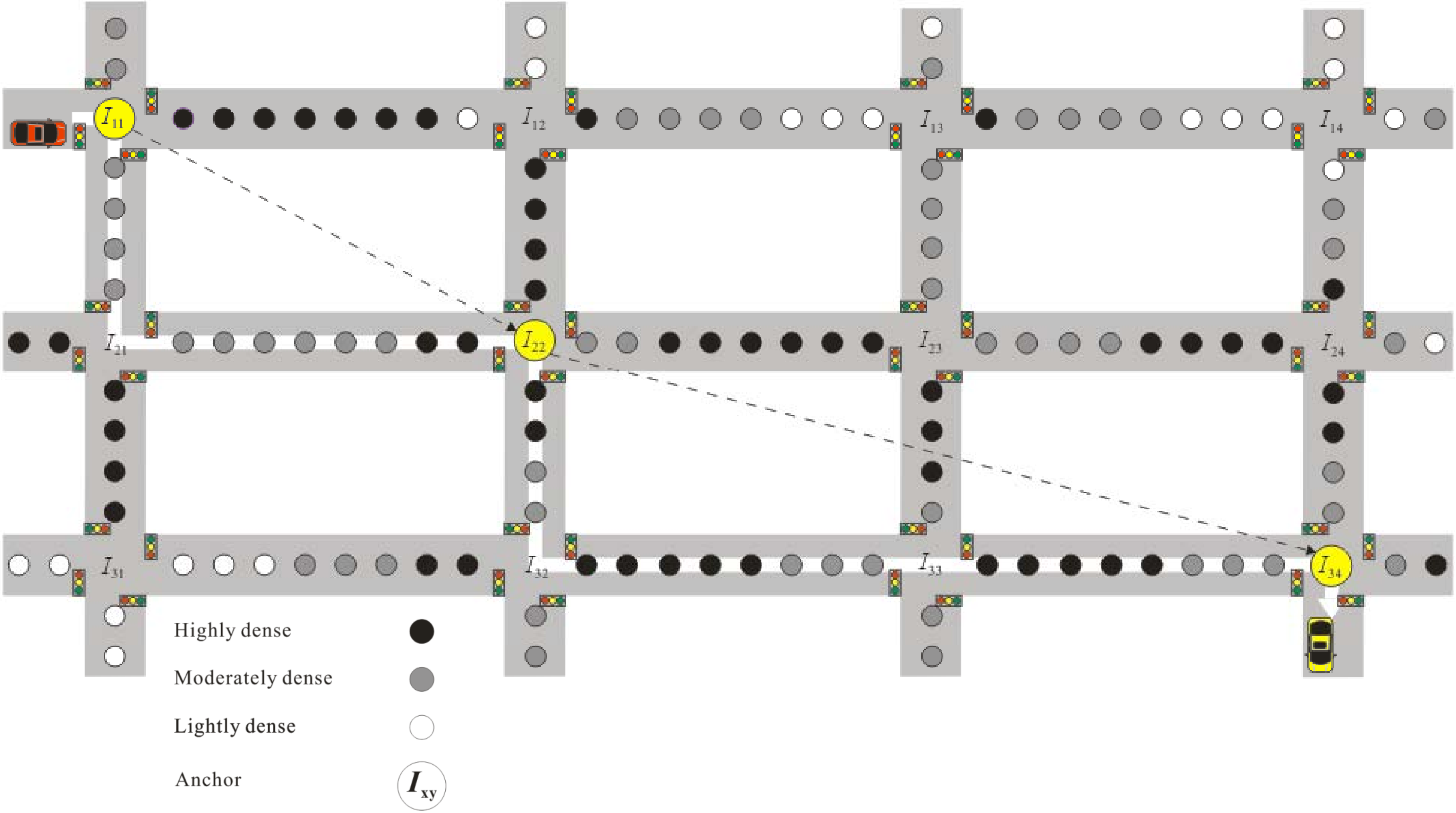
Example of DIR-Algorithm A



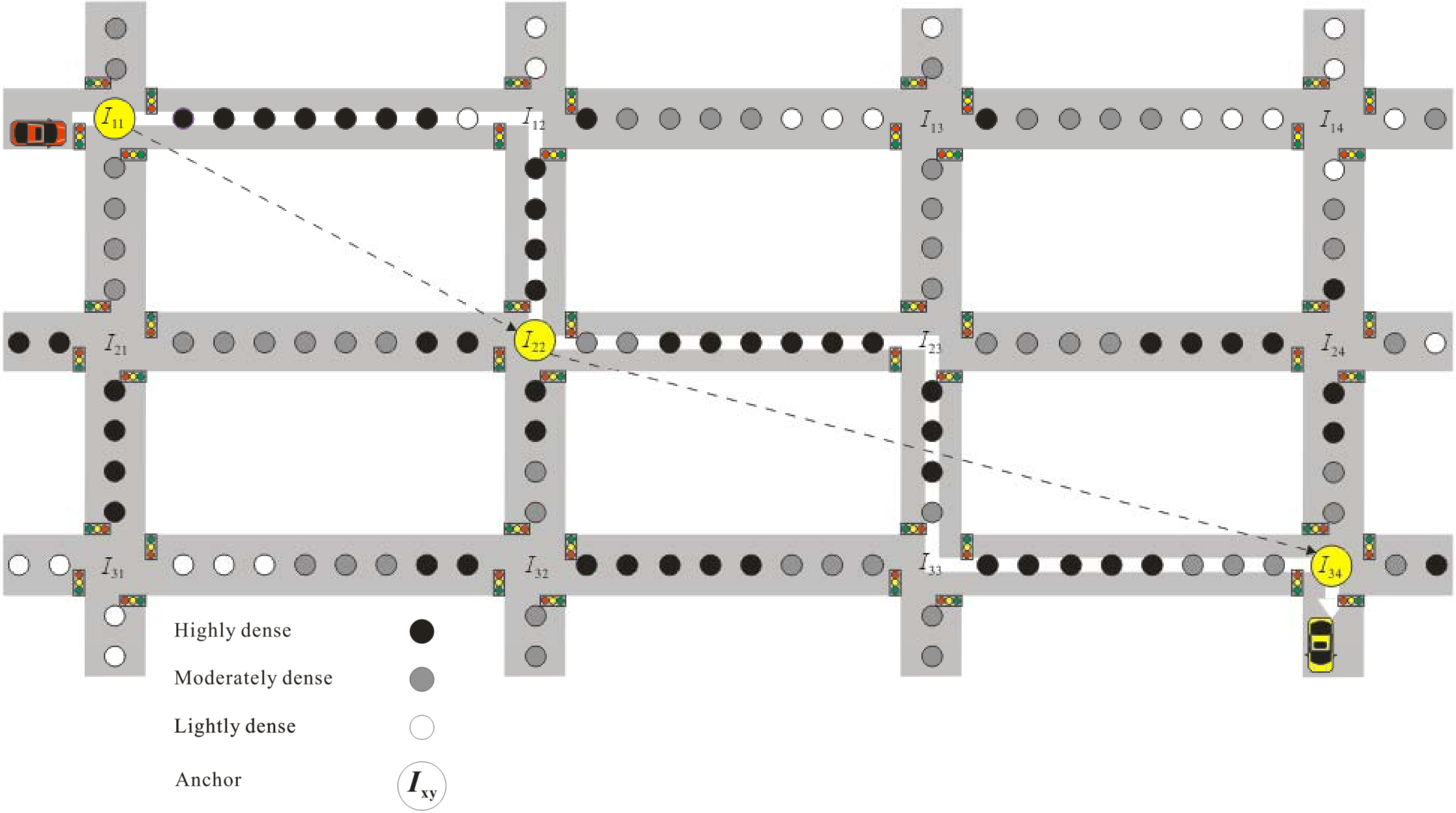
$$D_{11,22} = \min\left(d_{11,12} + d_{12,22}, d_{11,21} + d_{21,22}\right)$$

$$D_{22,34} = \min\left(d_{22,23} + d_{23,24} + d_{24,34}, d_{22,23} + d_{23,33} + d_{33,34}, d_{22,32} + d_{32,33} + d_{33,34}\right)$$

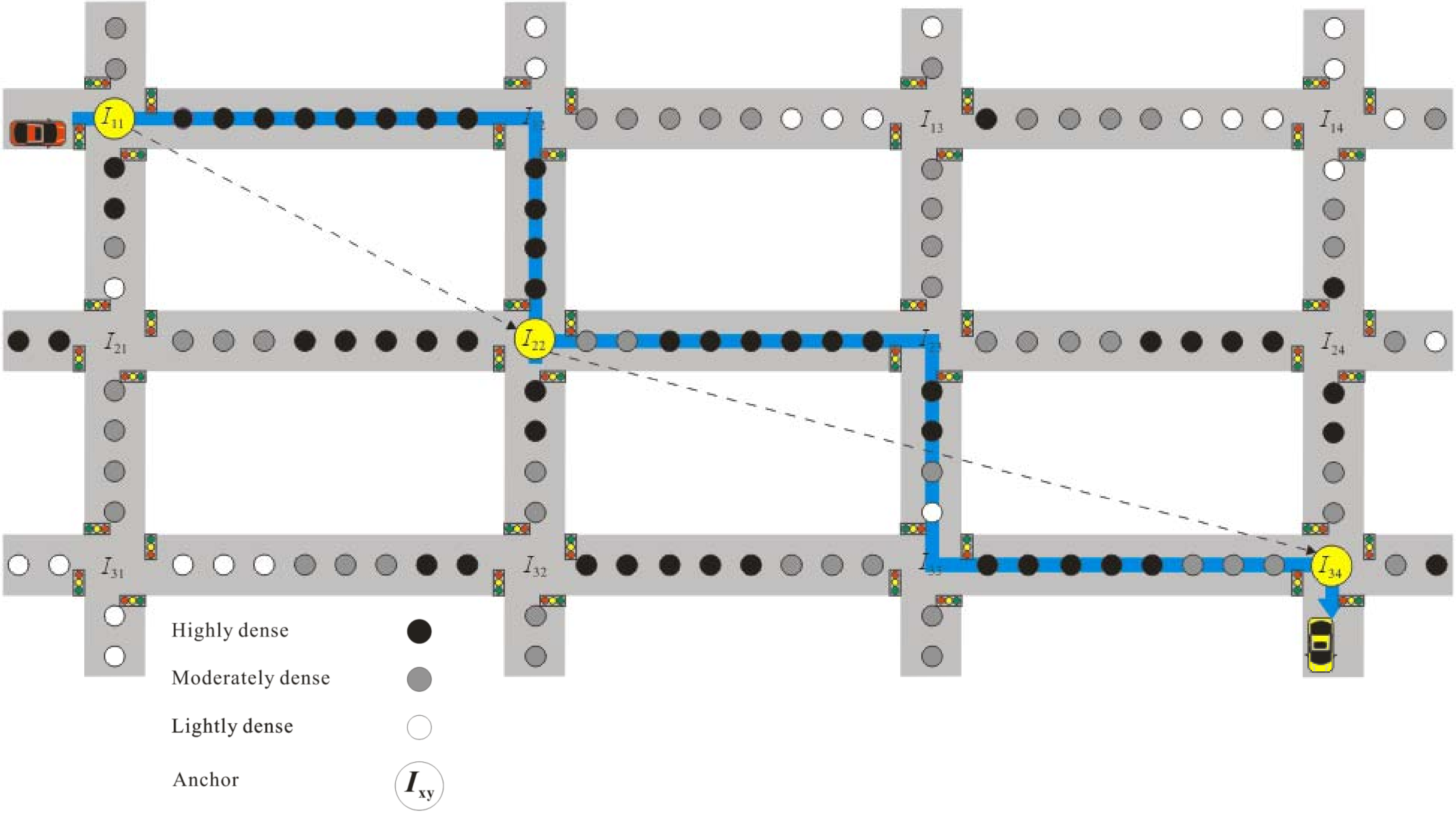
Example of DIR-Algorithm A



Example of DIR-Algorithm A

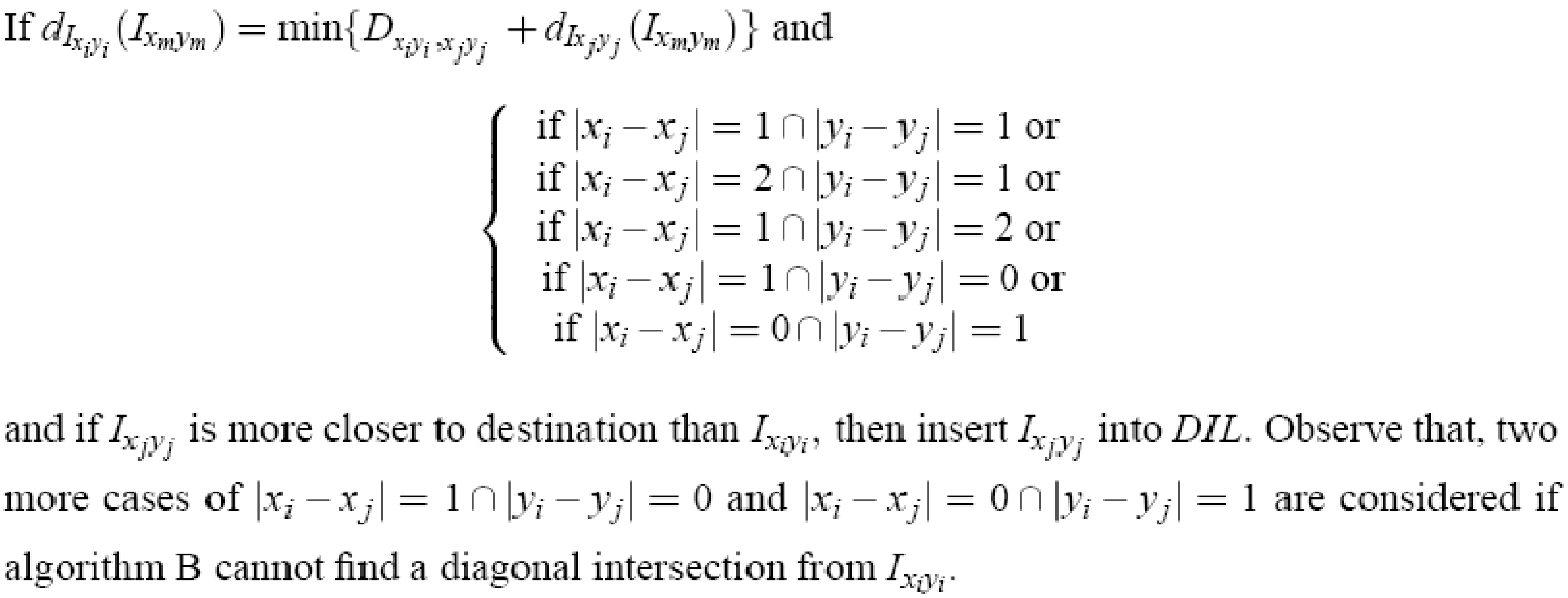


Example of DIR-Algorithm A



DIR-Algorithm B

- To construct a $DIL=[DI_1, DI_2, \ldots, DI_n]$
    - S1 ：Let $DIL=[I_{x_1y_1}]$ and $I_{x_iy_i}$
    - S2 ：



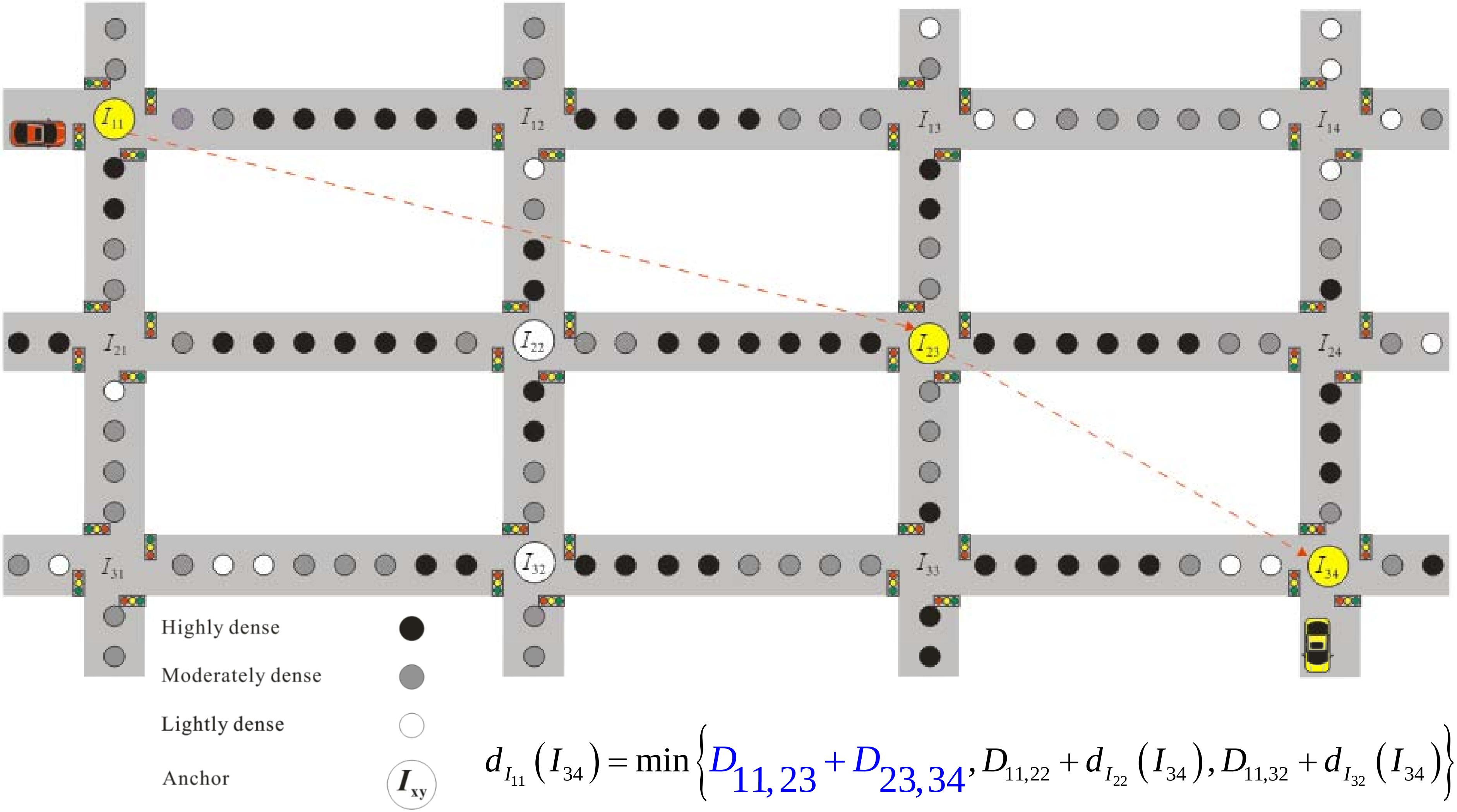
If $d_{I_{x_iy_i}}(I_{x_my_m}) = \min\{D_{x_iy_i,x_jy_j} + d_{I_{x_jy_j}}(I_{x_my_m})\}$ and

$$\begin{cases} \text{if } |x_i - x_j| = 1 \cap |y_i - y_j| = 1 \text{ or} \\ \text{if } |x_i - x_j| = 2 \cap |y_i - y_j| = 1 \text{ or} \\ \text{if } |x_i - x_j| = 1 \cap |y_i - y_j| = 2 \text{ or} \\ \text{if } |x_i - x_j| = 1 \cap |y_i - y_j| = 0 \text{ or} \\ \text{if } |x_i - x_j| = 0 \cap |y_i - y_j| = 1 \end{cases}$$

and if $I_{x_jy_j}$ is more closer to destination than $I_{x_iy_i}$, then insert $I_{x_jy_j}$ into DIL. Observe that, two more cases of $|x_i - x_j| = 1 \cap |y_i - y_j| = 0$ and $|x_i - x_j| = 0 \cap |y_i - y_j| = 1$ are considered if algorithm B cannot find a diagonal intersection from $I_{x_iy_i}$.

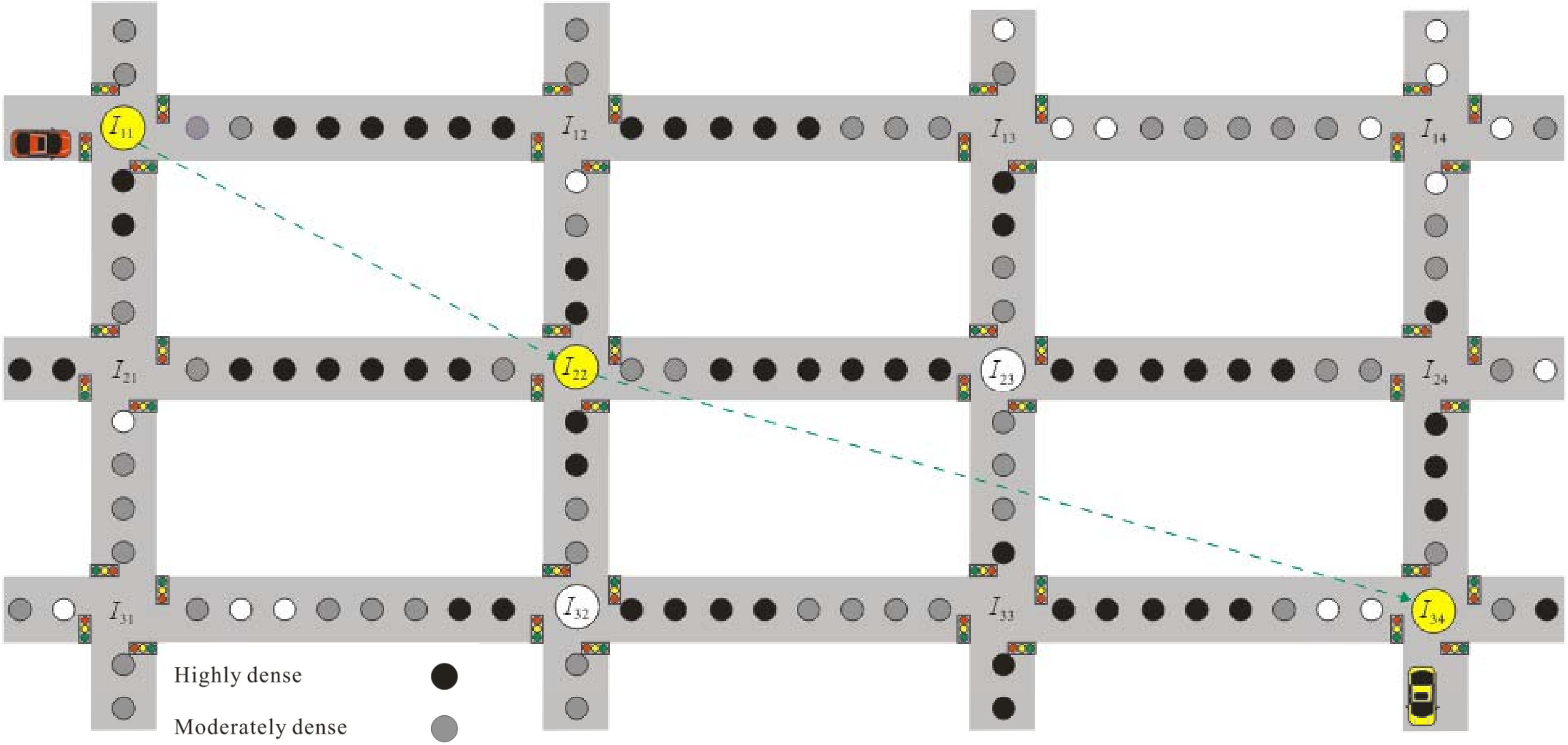
-
    - S3： $DIL=[DI_1, DI_2, \ldots, DI_n]$ is constructed

Example of DIR-Algorithm B



$$d_{I_{11}}(I_{34}) = \min\left\{D_{11,23} + D_{23,34}, D_{11,22} + d_{I_{22}}(I_{34}), D_{11,32} + d_{I_{32}}(I_{34})\right\}$$

Example of DIR-Algorithm B



Highly dense ●

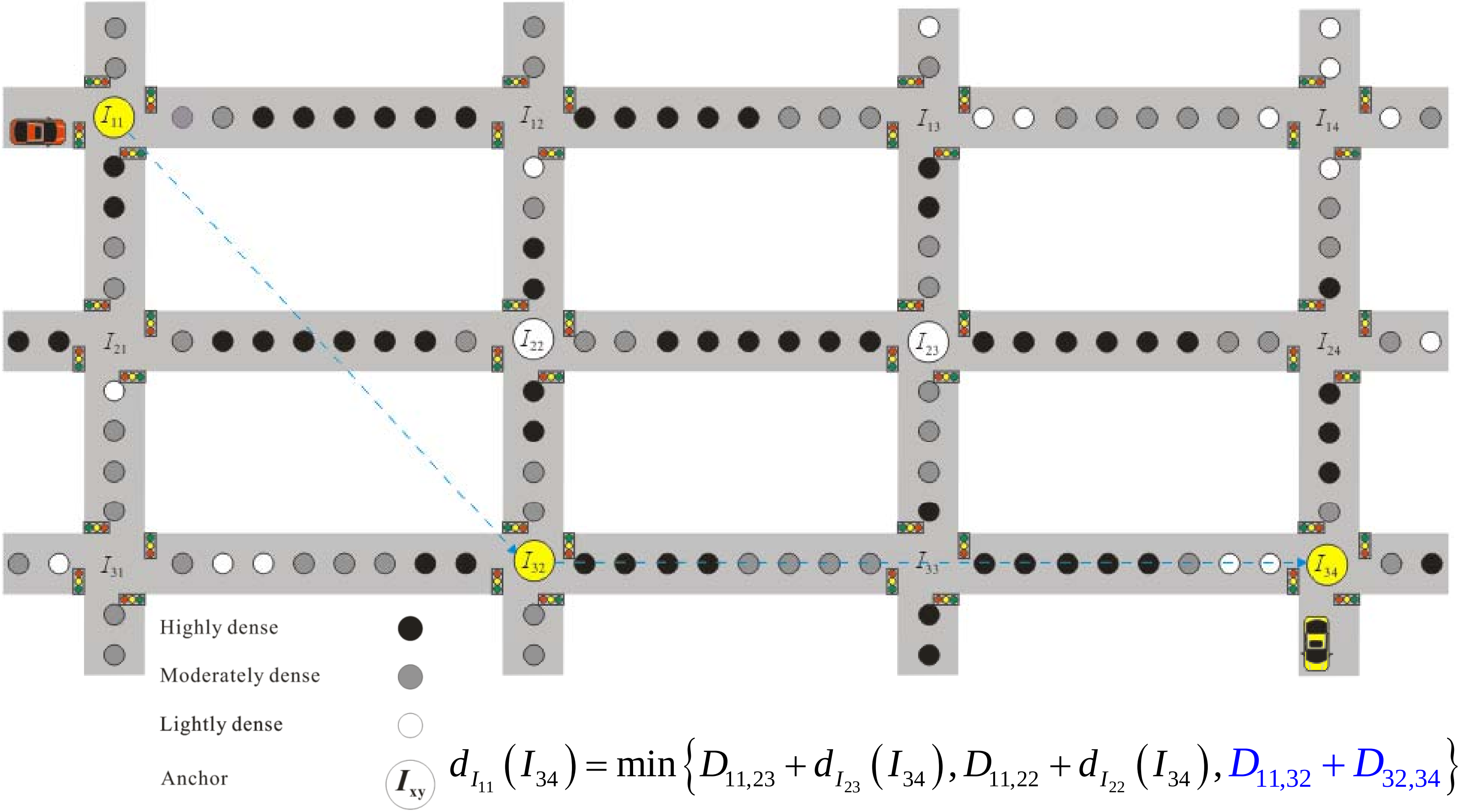
Moderately dense ◍

Lightly dense ○

Anchor I_{xy}

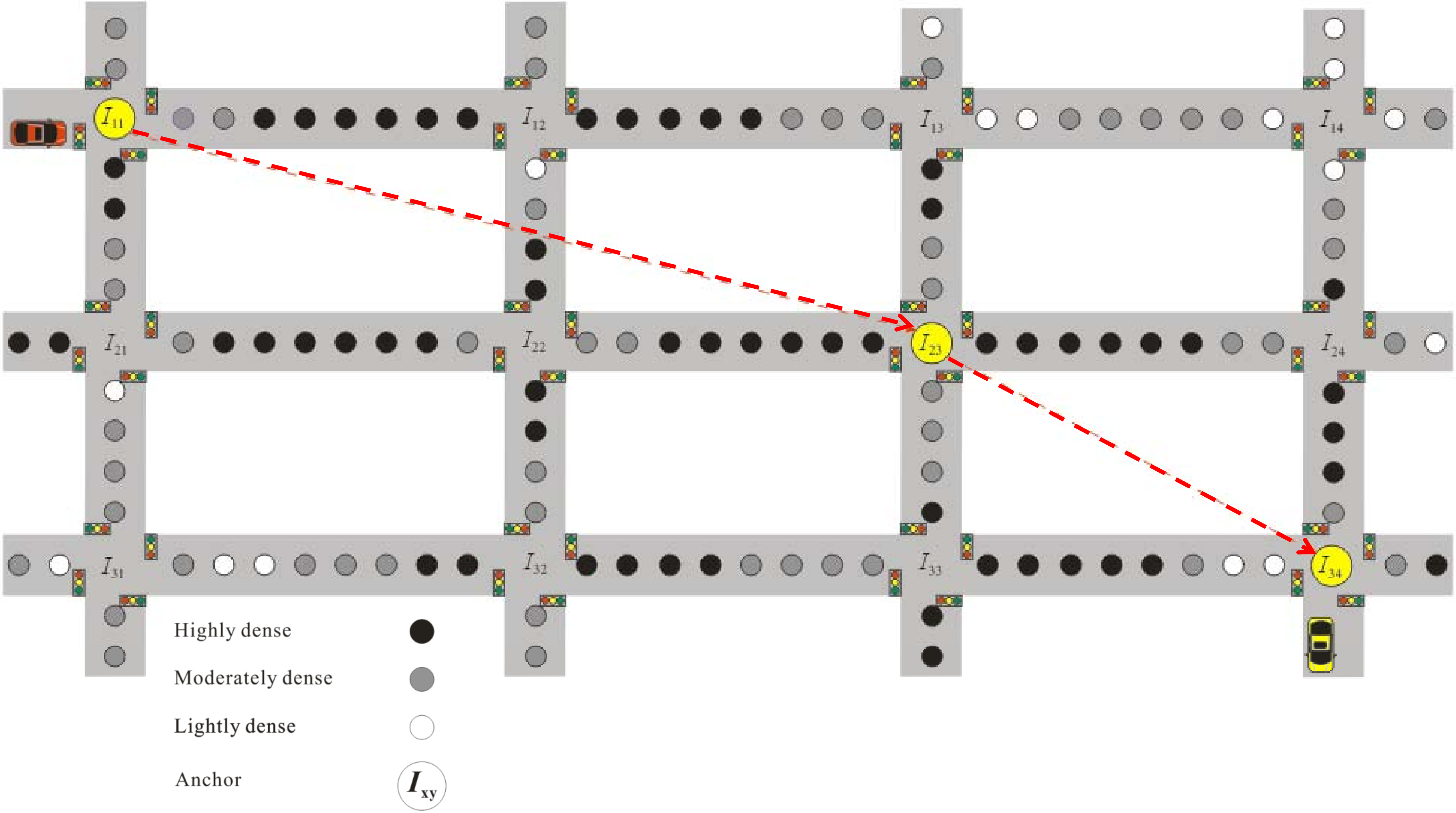
$$d_{I_{11}}\left(I_{34}\right) = \min\left\{D_{11,22} + d_{I_{22}}\left(I_{34}\right), D_{11,22} + D_{22,34}, D_{11,32} + d_{I_{32}}\left(I_{34}\right)\right\}$$

Example of DIR-Algorithm B

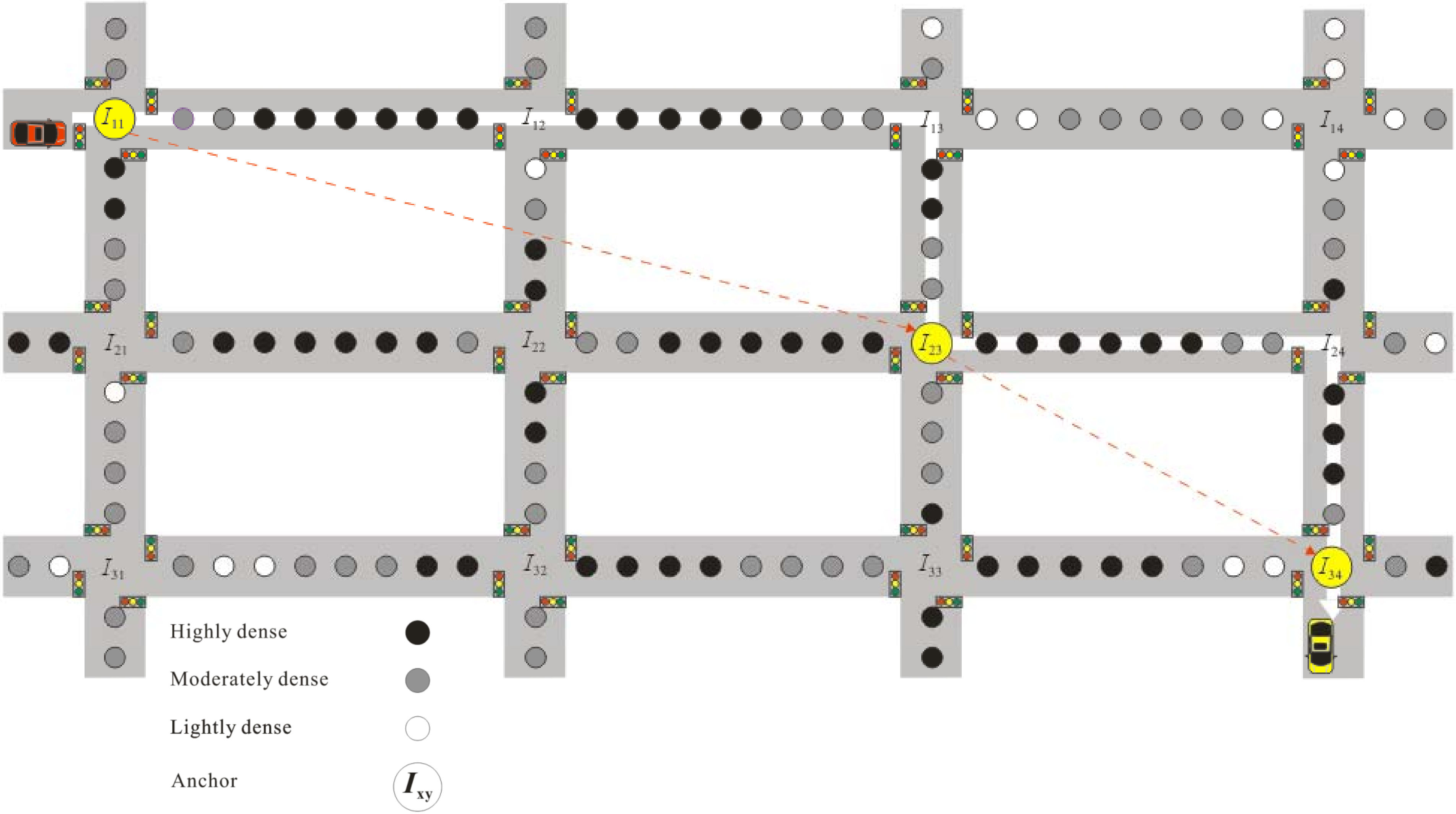


$$d_{I_{11}}\left(I_{34}\right) = \min\left\{D_{11,23} + d_{I_{23}}\left(I_{34}\right), D_{11,22} + d_{I_{22}}\left(I_{34}\right), D_{11,32} + D_{32,34}\right\}$$

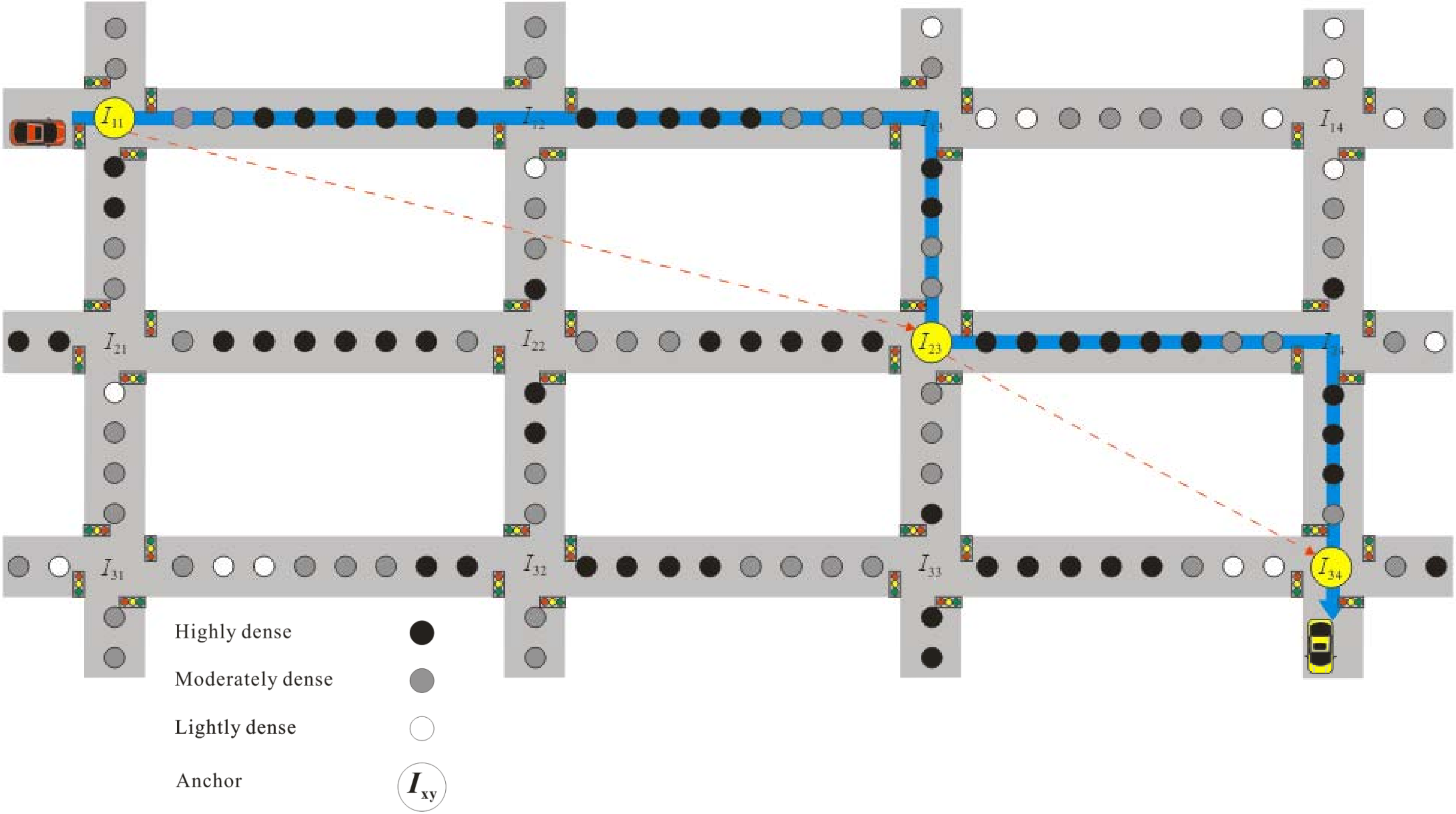
Example of DIR-Algorithm B



Example of DIR-Algorithm B



Example of DIR-Algorithm B



Data Forwarding

- The data forwarding operation is formally given as follows
  - S1：Vehicle in $I_{x_iy_i}$ can keep the most accurate traffic information to re-calculates $D_{x_iy_i,x_jy_j}$

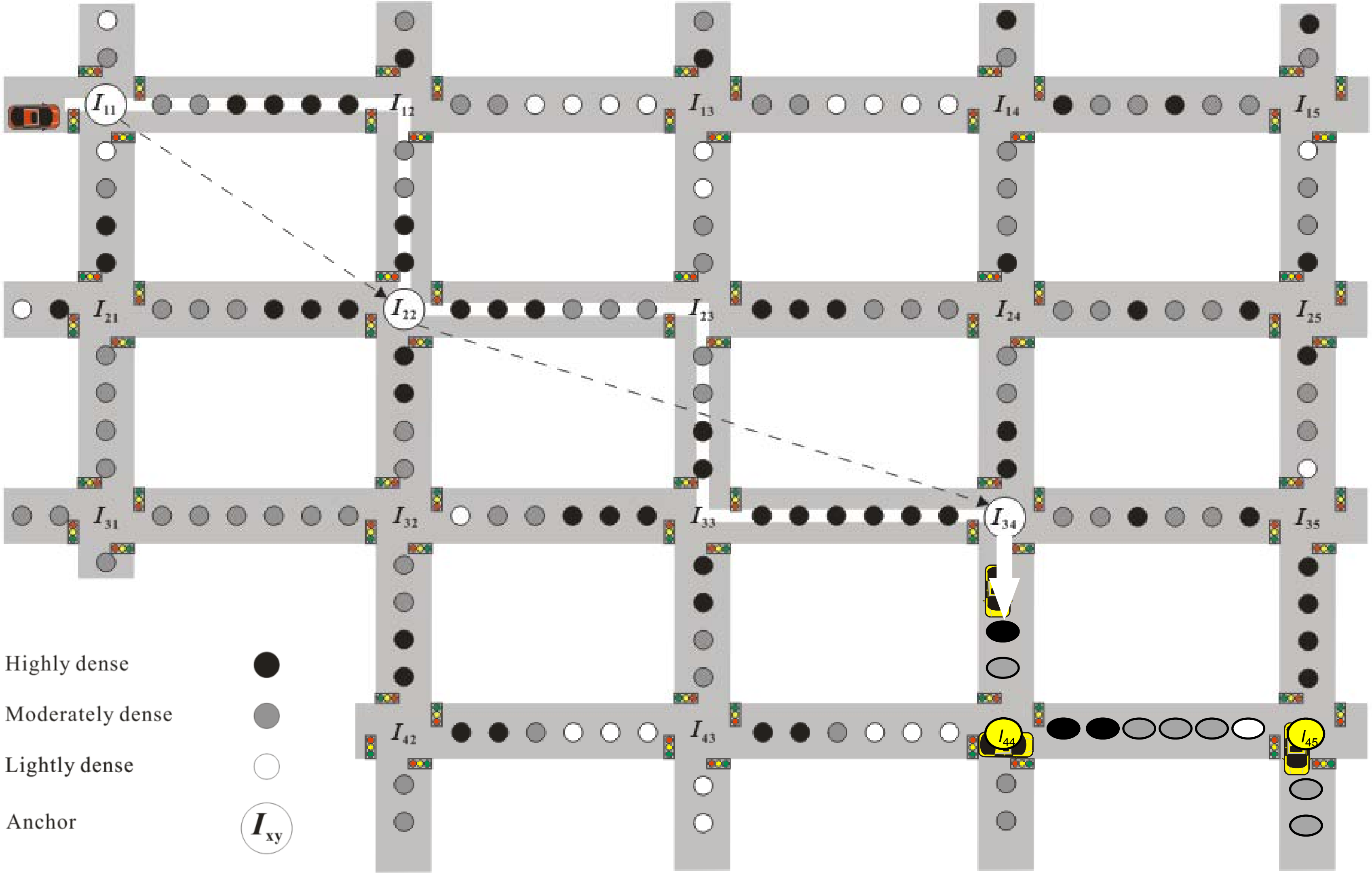
$$\left\{\begin{array}{c} |x_i - x_j| = 1 \cap |y_i - y_j| = 1 \text{ or} \\ |x_i - x_j| = 2 \cap |y_i - y_j| = 1 \text{ or} \\ |x_i - x_j| = 1 \cap |y_i - y_j| = 2, \text{ as follows.} \end{array}\right.$$

  - S2：$|x_i-x_j|=1\cap|y_i-y_j|=1$, two different sub-path
  - S3：$|x_i-x_j|=2\cap|y_i-y_j|=1$, three different sub-path
  - S4：$|x_i-x_j|=1\cap|y_i-y_j|=2$, three different sub-path

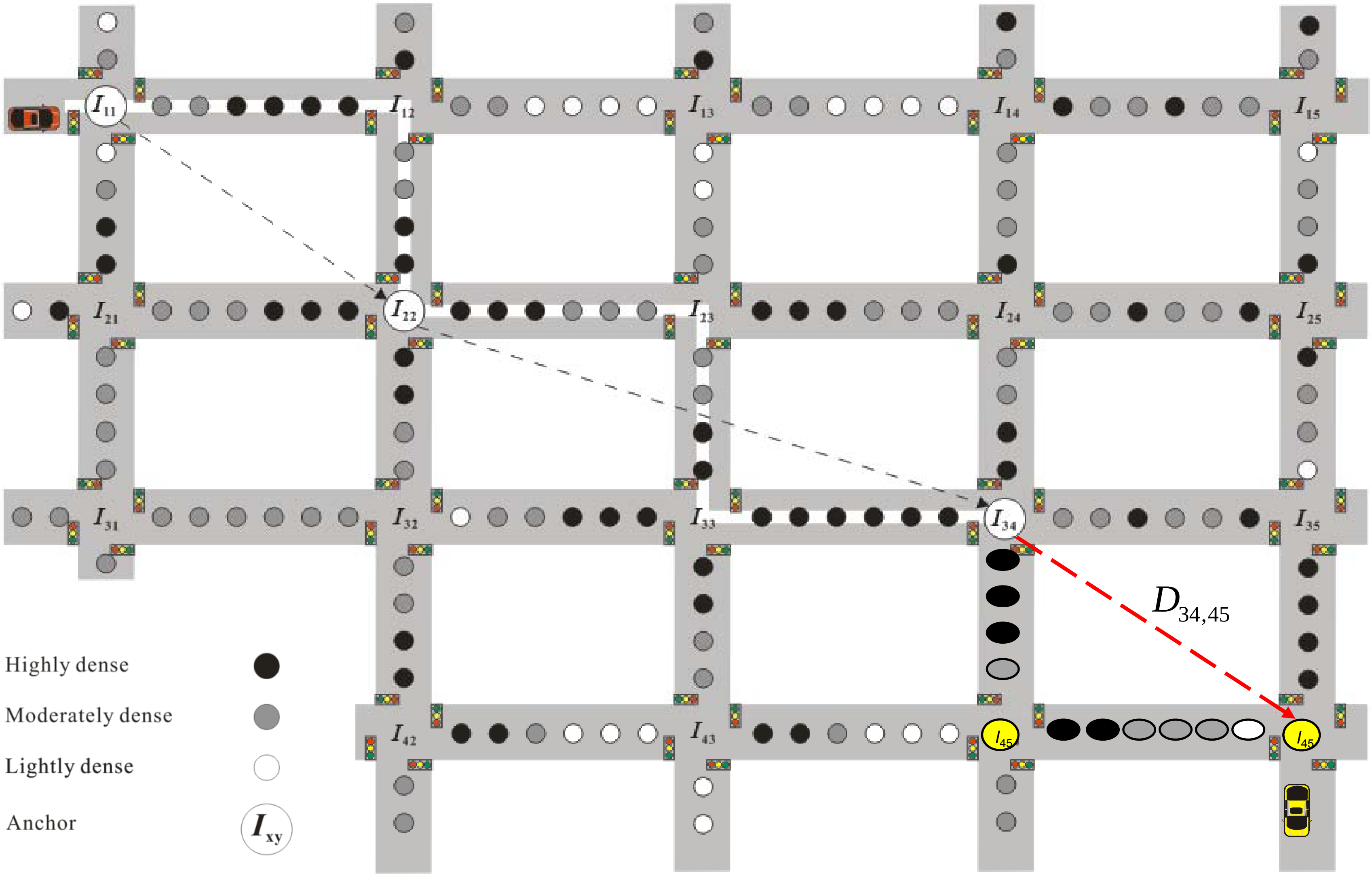
Path Maintenance

- Source and destination are fixed
  - The data forward is done based on the constructed *DIL*=[DI_1, DI_2,·····, DI_n] in the data forwarding phase
- Source and destination are mobile
  - S1：The destination is moving and far away the $DI_{current\_list}$ in the current *DIL*
    - Appended into *DIL*
  - S2：The destination is moving and near to last $DI_{current\_list}$ in the current *DIL*
    - Constructed new *DIL'*

Example of Destination is Far Away

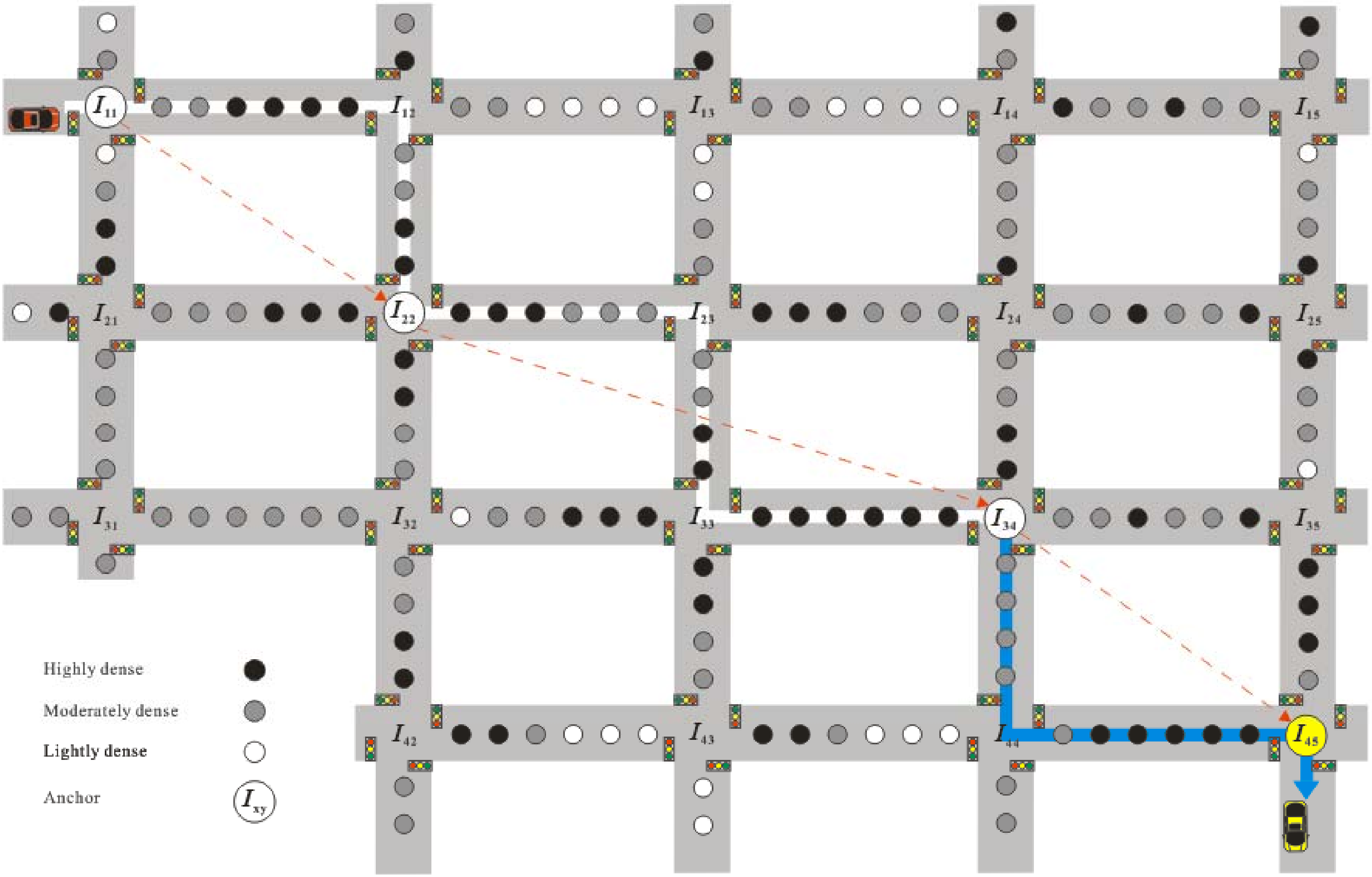


Example of Destination is Far Away

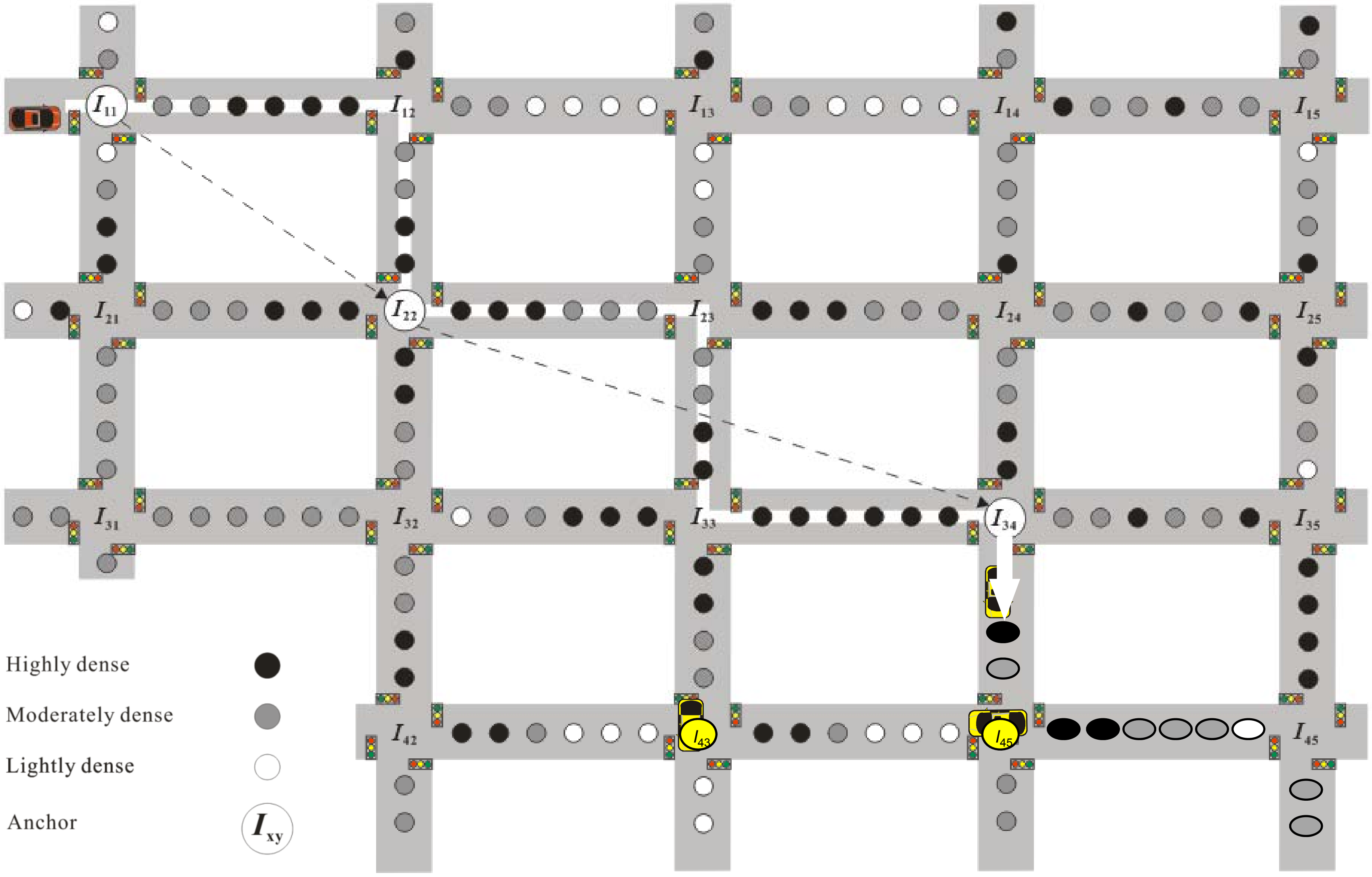


$$D_{34,45} = \min\left(d_{34,44} + d_{44,45}, d_{34,35} + d_{35,45}\right)$$

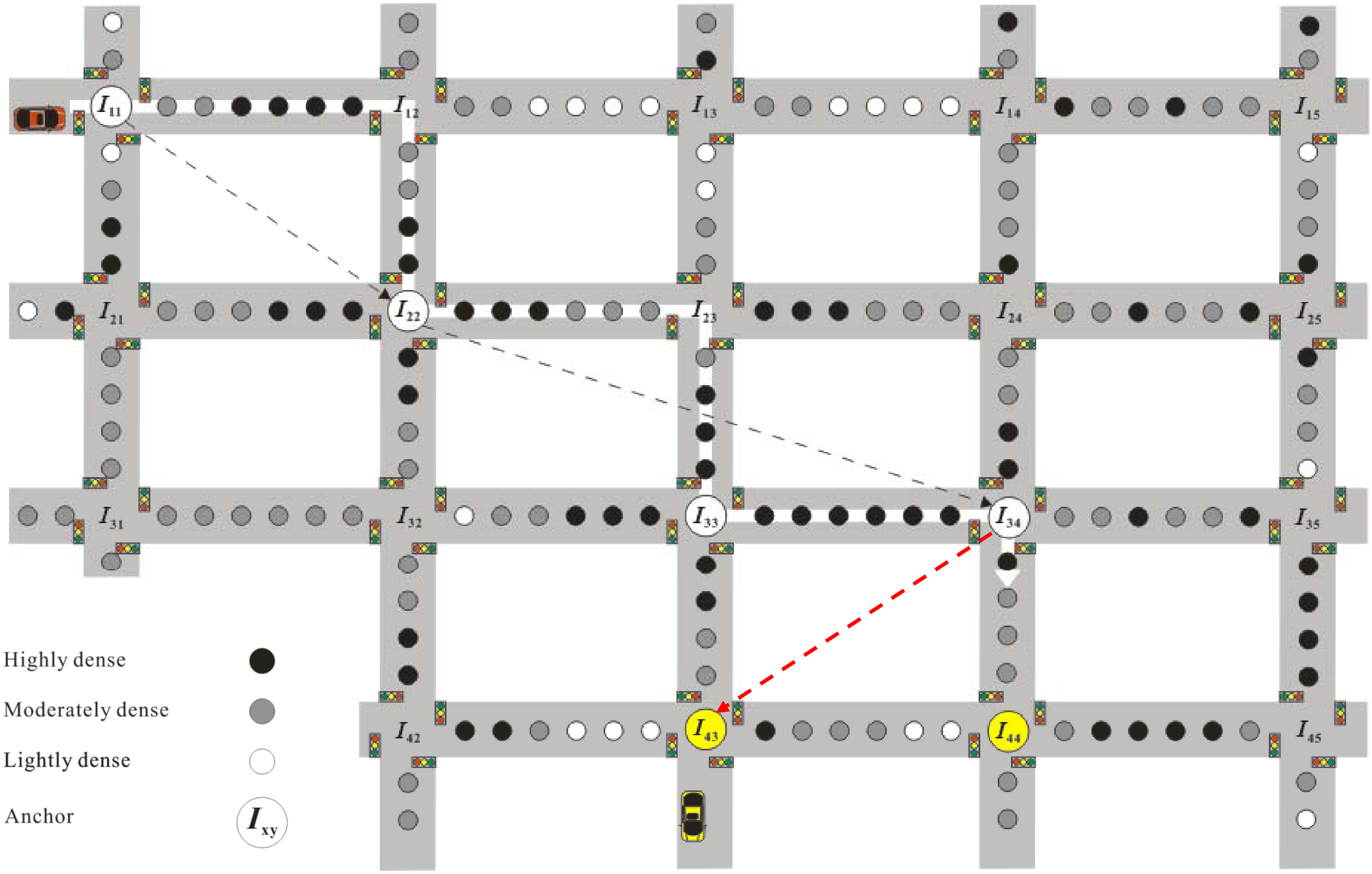
Example of Destination is Far Away



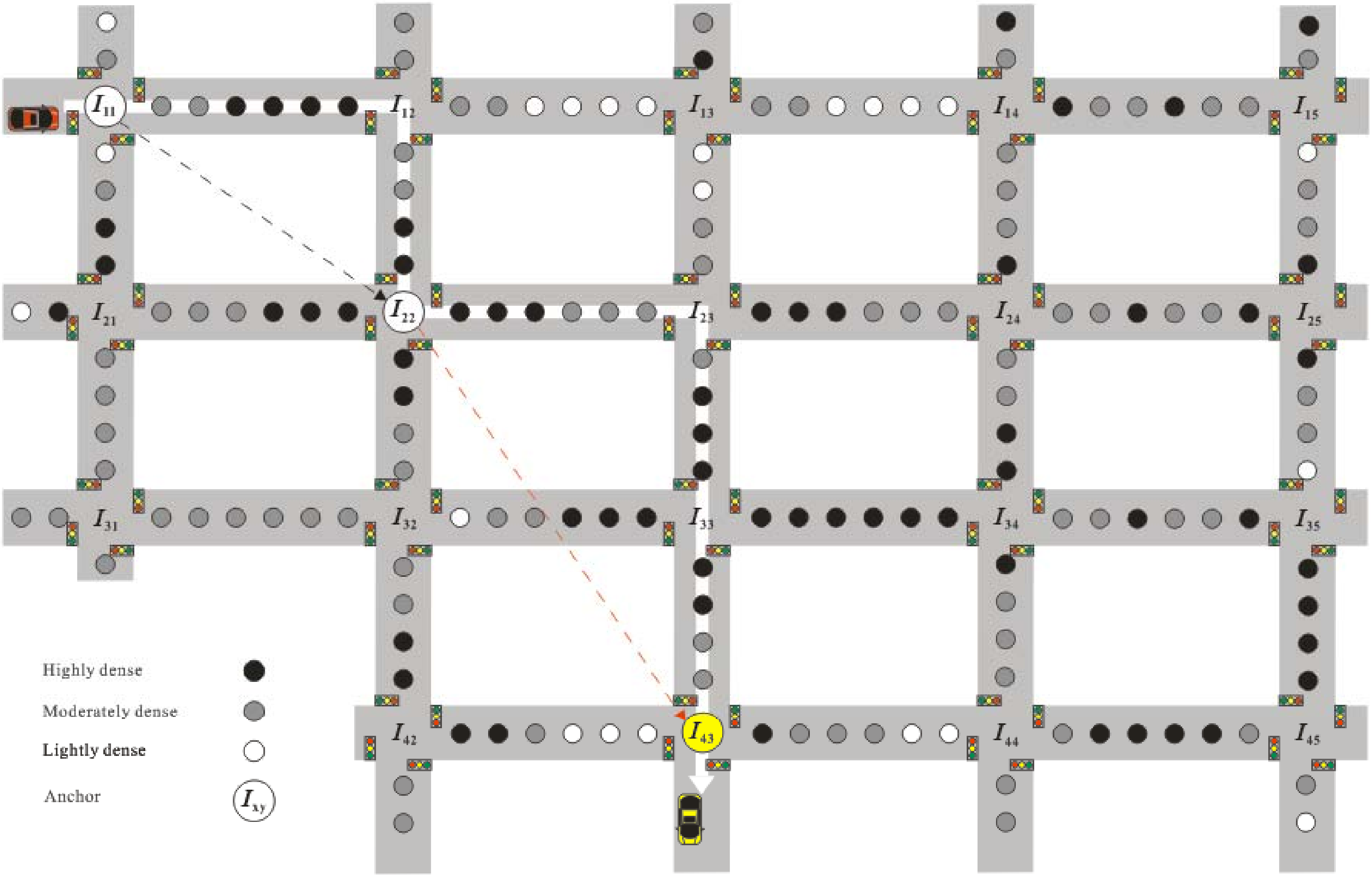
Example of Destination is Closed to Source



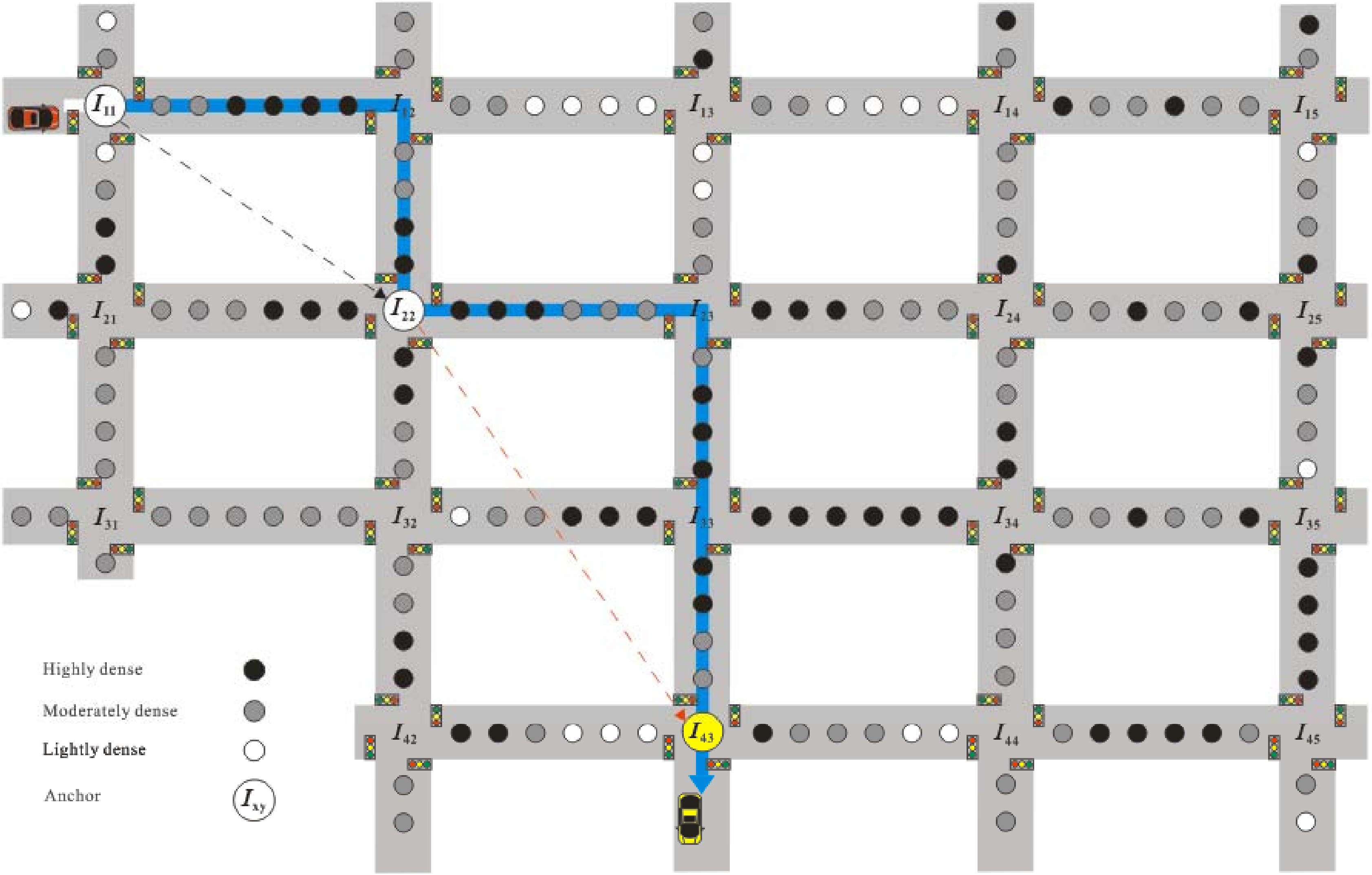
Example of Destination is Closed to Source



Example of Destination is Closed to Source



Example of Destination is Closed to Source

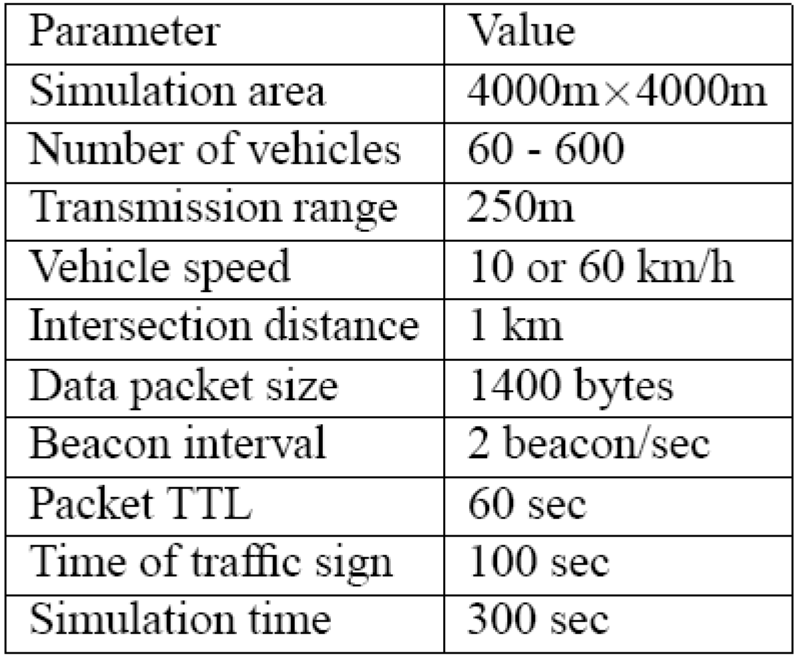


Performance Analysis

- Time complexity analysis
- Simulation results

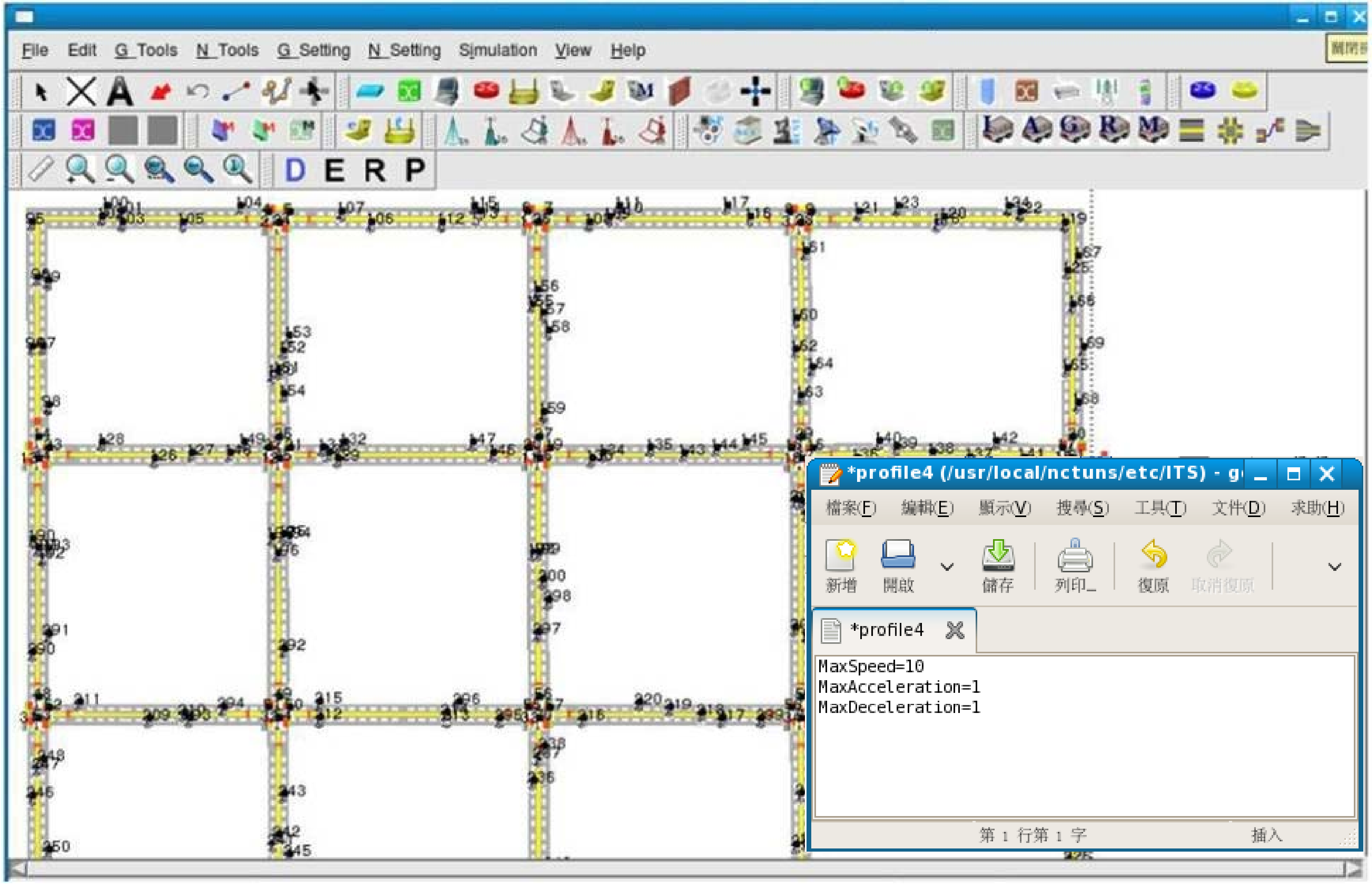
Simulation Results

- All protocol are implemented by using NCTUns 4.0 for the following protocols.
    - CAR
    - DIR_A
    - DIR_B
- System parameters



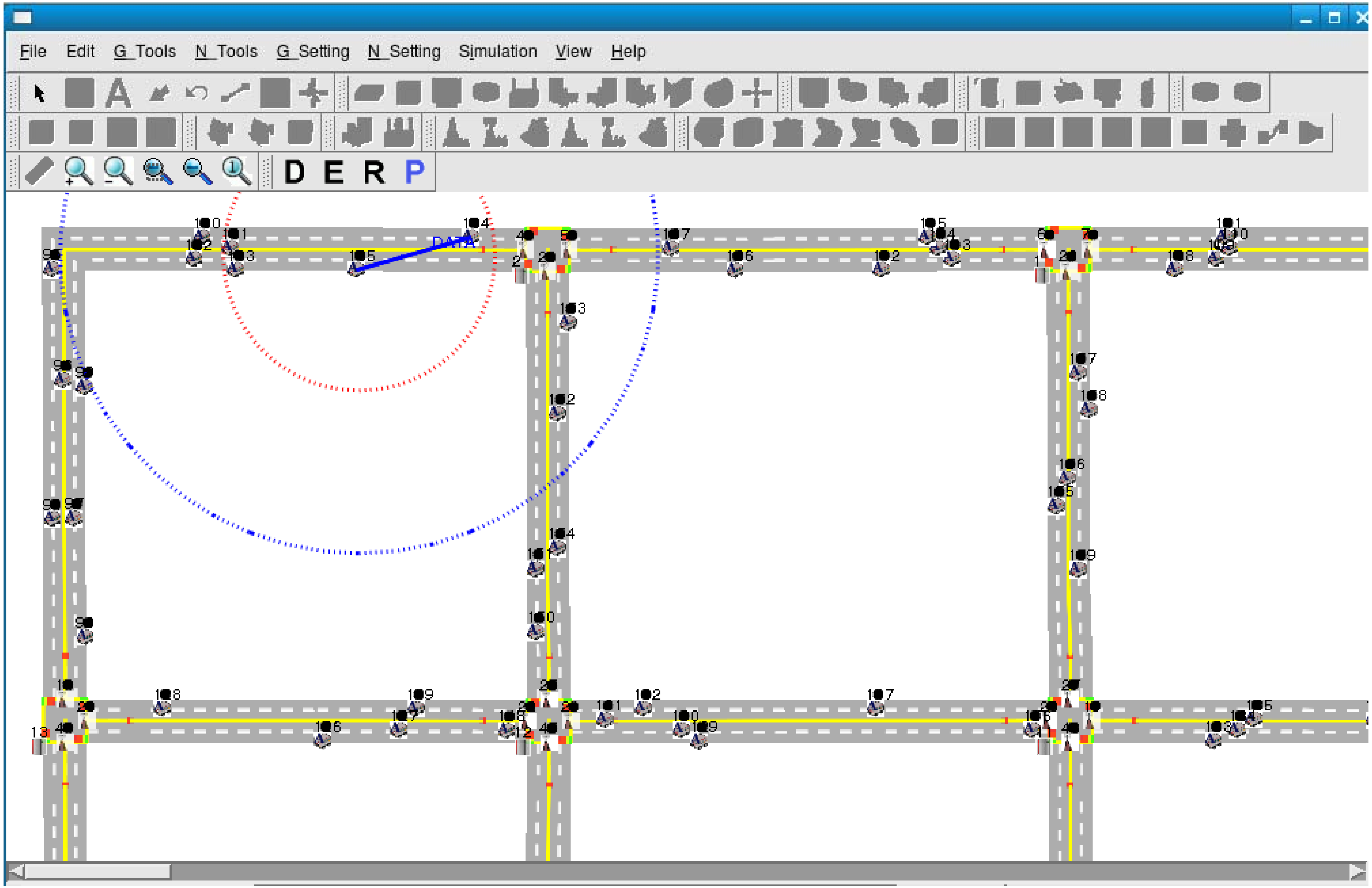
Parameter	Value
Simulation area	4000m×4000m
Number of vehicles	60 - 600
Transmission range	250m
Vehicle speed	10 or 60 km/h
Intersection distance	1 km
Data packet size	1400 bytes
Beacon interval	2 beacon/sec
Packet TTL	60 sec
Time of traffic sign	100 sec
Simulation time	300 sec

Simulation Tool



```
MaxSpeed=10
MaxAcceleration=1
MaxDeceleration=1
```

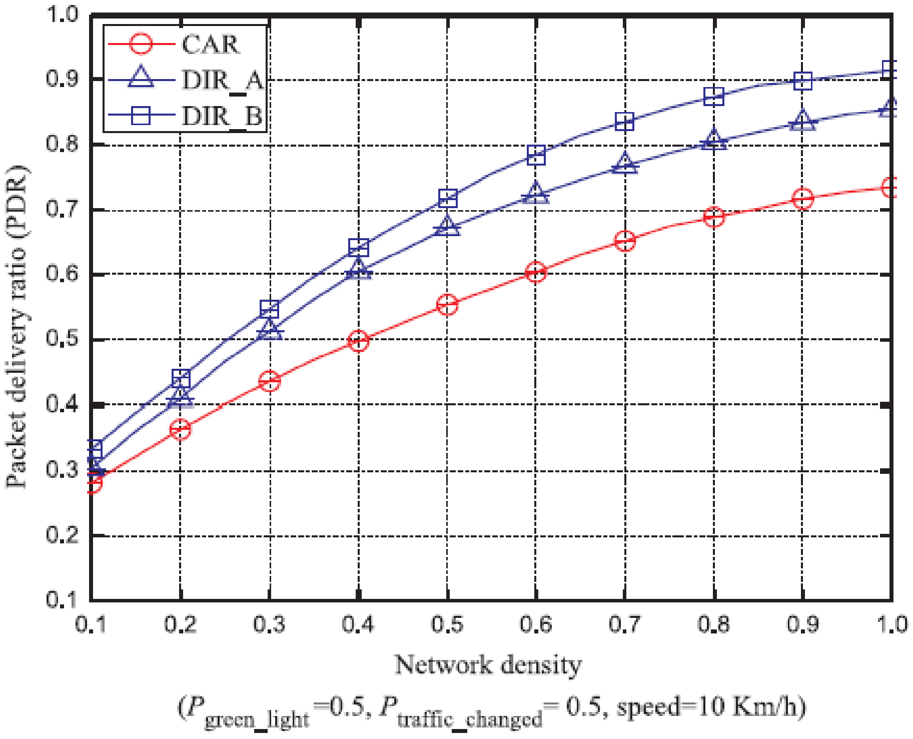
Simulation Tool (Cont.)



Performance Metrics

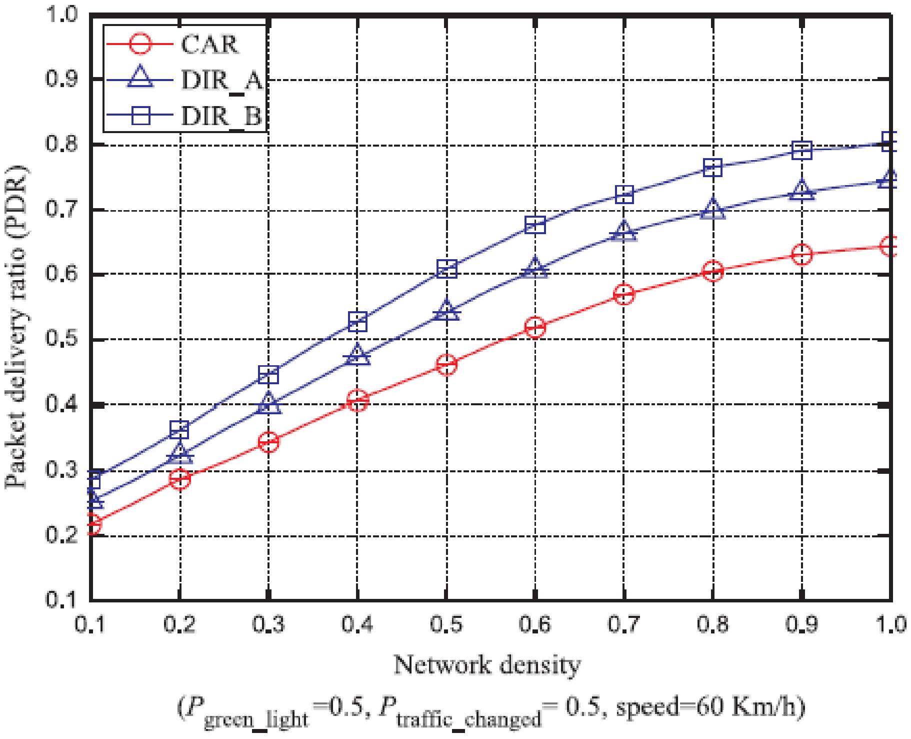
- Packet delivery ratio (PDR)：
  - Total number of packets successfully received by destination vehicle divided by the total number of packets sent by the source vehicle.
- Packet delivery delay (PDD)：
  - Average time cost of data packet traveled from the source to the destination.
- Message overhead (MO)：
  - Total number of packets that source vehicle transmit.
- Throughput (TP)：
  - Total number of data packets the destination vehicle received per second.

Packet Delivery Ratio (PDR) vs. Network density



(P_{green_light}=0.5, $P_{traffic\_changed}$= 0.5, speed=10 Km/h)

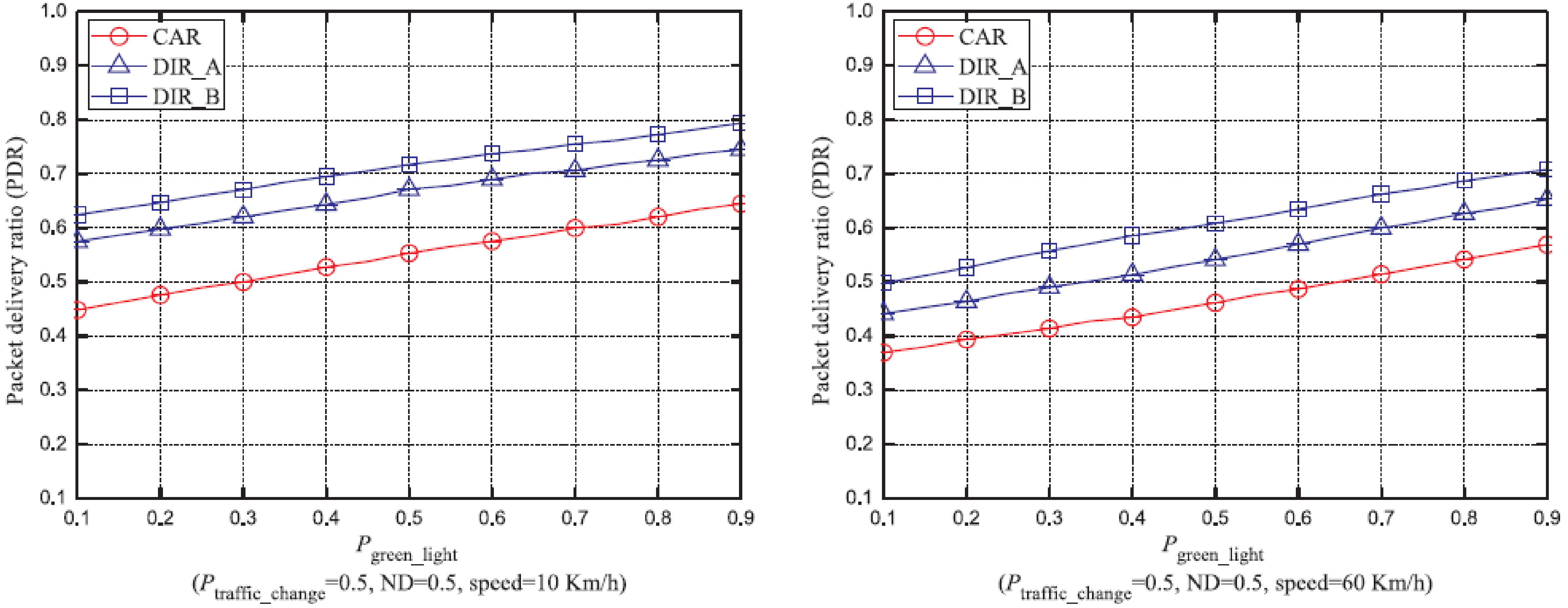
(a)



(P_{green_light}=0.5, $P_{traffic\_changed}$= 0.5, speed=60 Km/h)

(b)

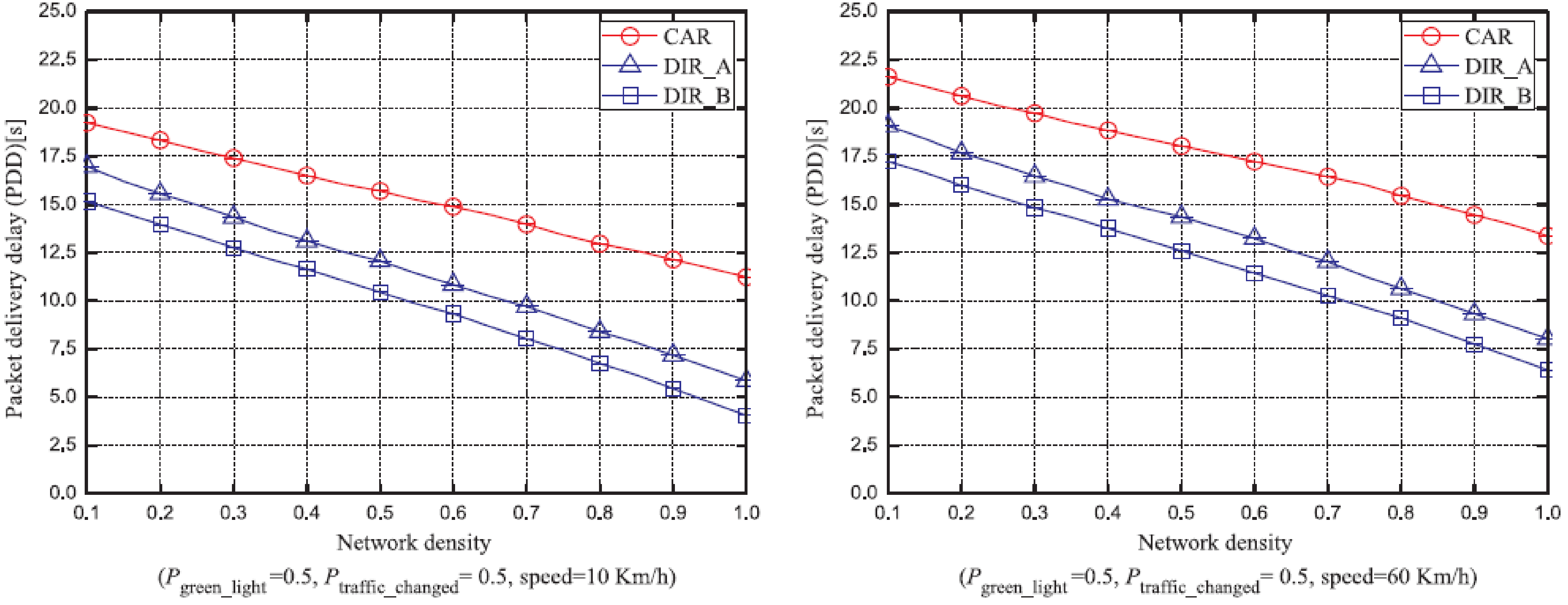
Packet Delivery Ratio (PDR) vs. P_{green_light}



(a)

(b)

Packet Delivery Delay (PDD) vs. Network density

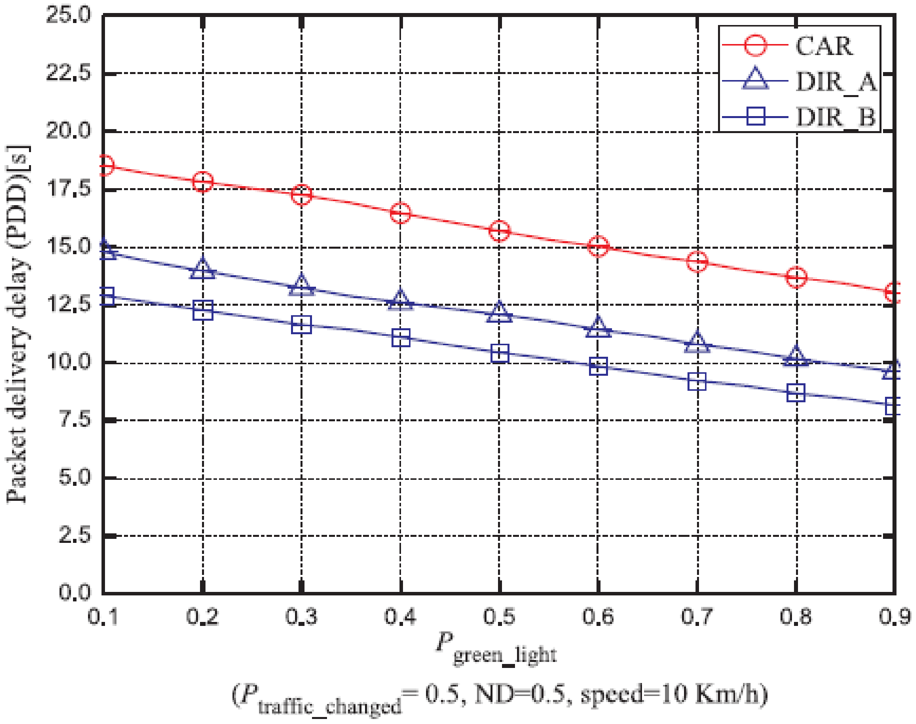


(P_{green_light}=0.5, $P_{traffic\_changed}$= 0.5, speed=10 Km/h)

(P_{green_light}=0.5, $P_{traffic\_changed}$= 0.5, speed=60 Km/h)

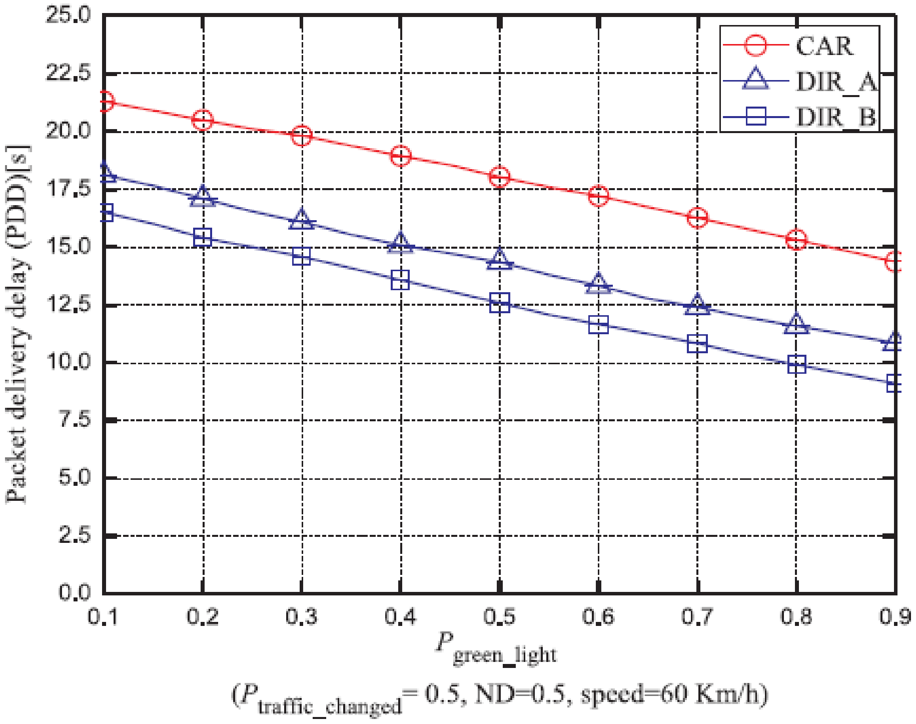
(a) (b)

Packet Delivery Delay (PDD) vs. P_{green_light}



($P_{traffic\_changed}$= 0.5, ND=0.5, speed=10 Km/h)

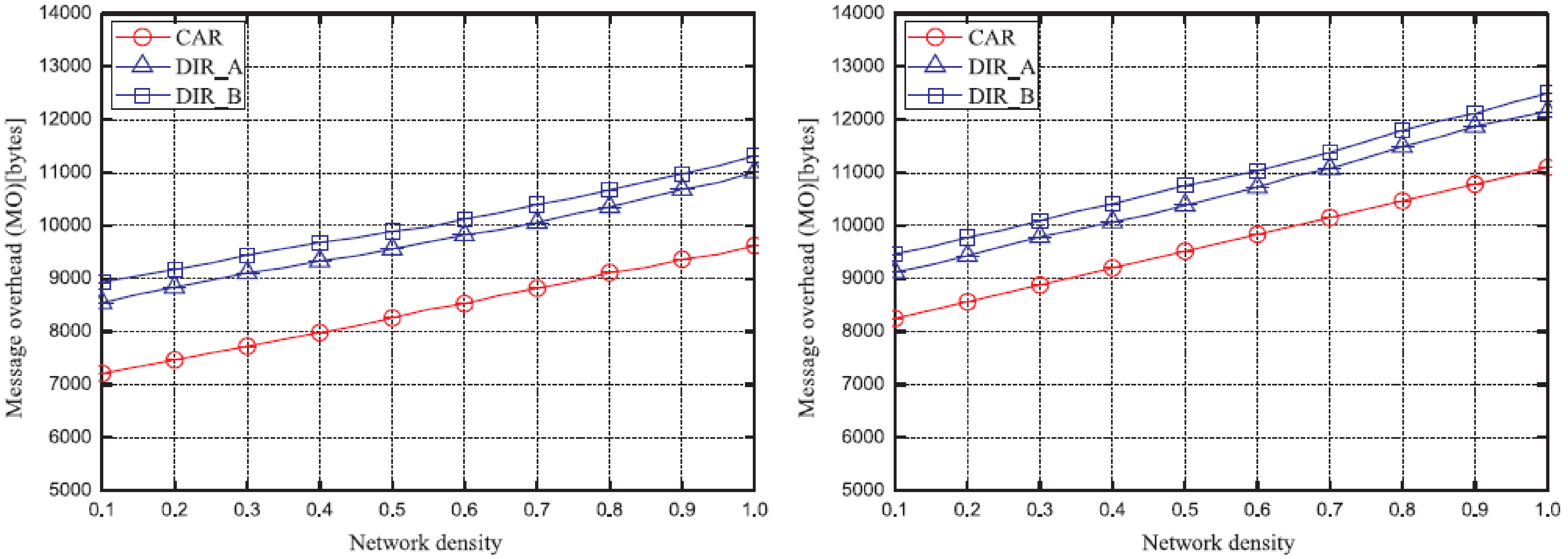
(a)



($P_{traffic\_changed}$= 0.5, ND=0.5, speed=60 Km/h)

(b)

Message Overhead (MO) vs. Network density



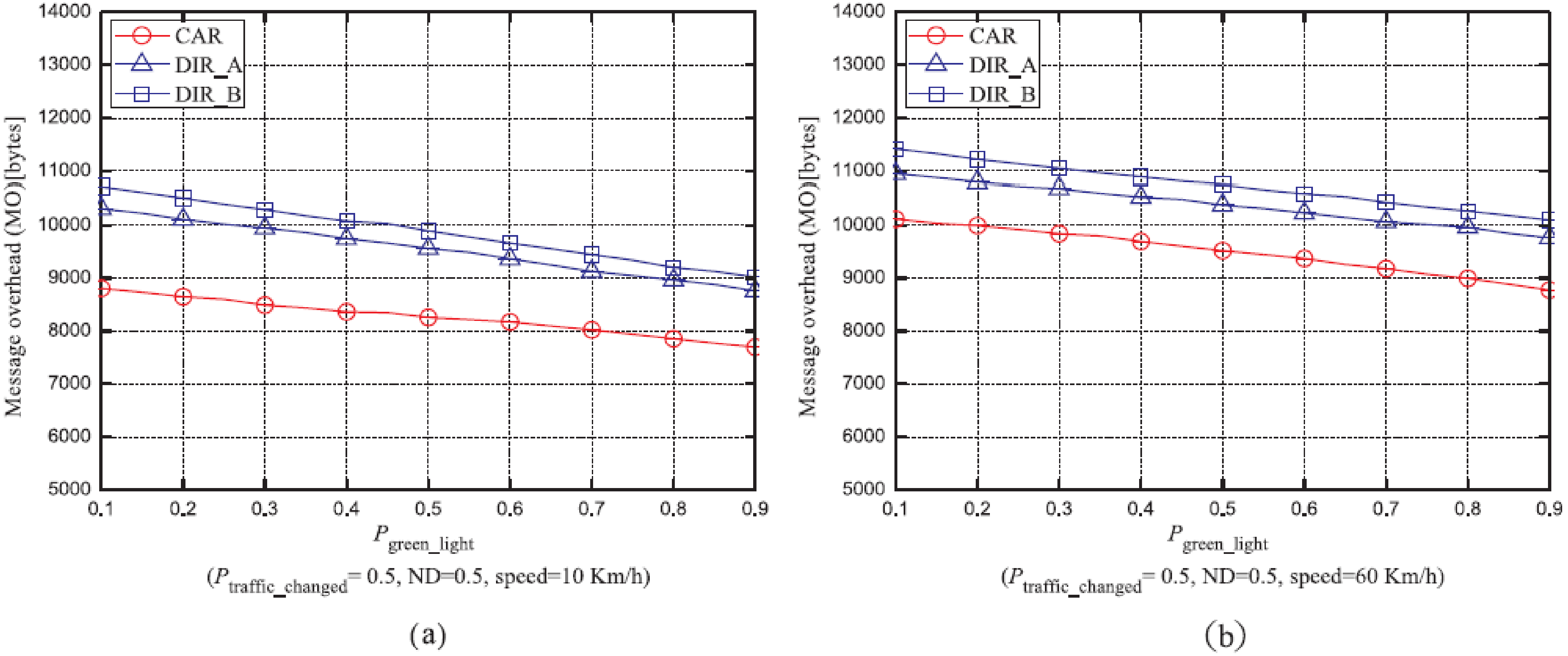
(P_{green_light}=0.5, $P_{traffic\_changed}$= 0.5, speed=10 Km/h)

(a)

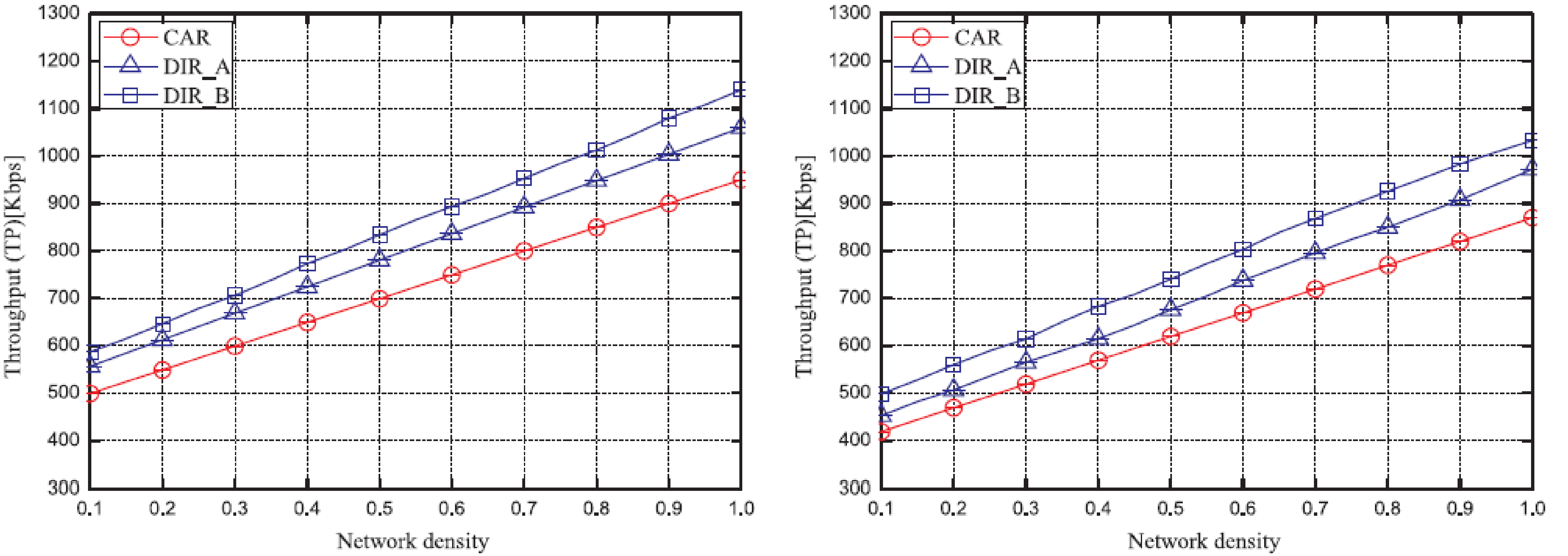
(P_{green_light}=0.5, $P_{traffic\_changed}$= 0.5, speed=60 Km/h)

(b)

Message Overhead (MO) vs. P_{green_light}



Throughput (TP) vs. Network density



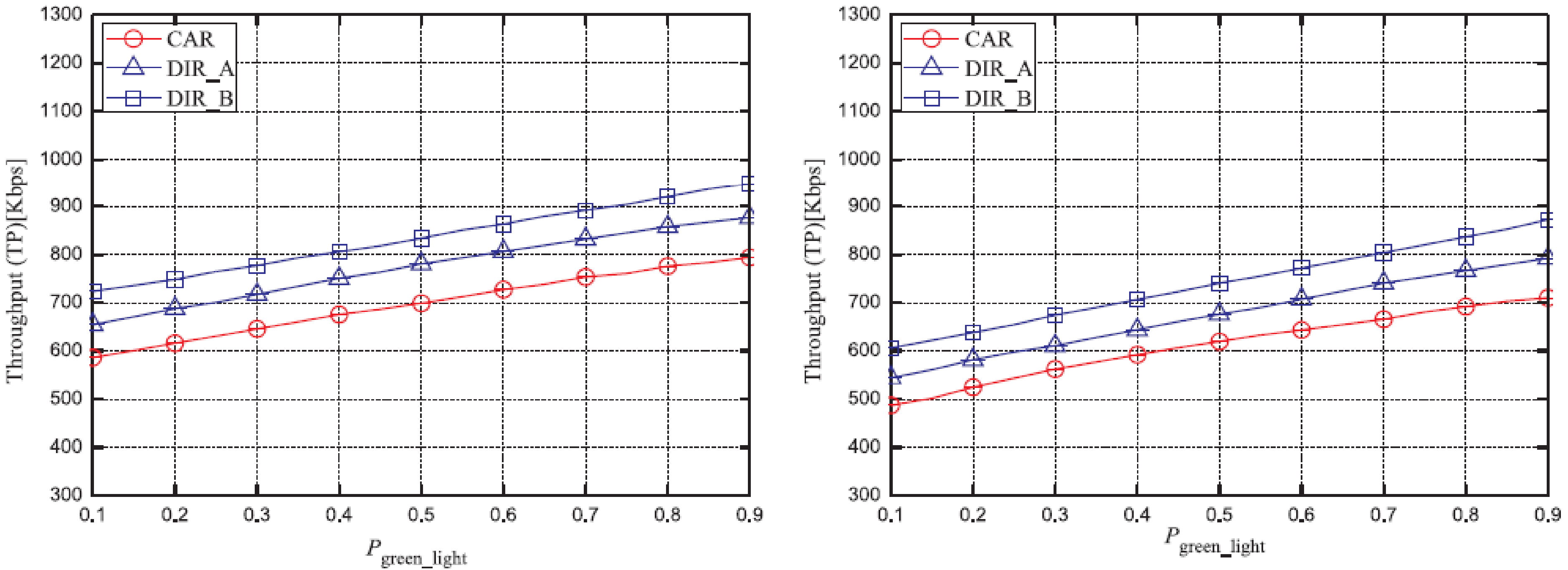
(P$_{\text{green\_light}}$=0.5, P$_{\text{traffic\_changed}}$= 0.5, speed=10 Km/h)

(a)

(P$_{\text{green\_light}}$=0.5, P$_{\text{traffic\_changed}}$= 0.5, speed=60 Km/h)

(b)

Throughput (TP) vs. P_{green_light}



($P_{traffic\_changed}$= 0.5, ND=0.5, speed=10 Km/h)

(a)

($P_{traffic\_changed}$= 0.5, ND=0.5, speed=60 Km/h)

(b)

Conclusion

- This work develops a diagonal-intersection-based routing (DIR) protocol for city environment to significantly improve the packet delivery ratio and the packet delivery delay.
  - The DIR protocol has the auto-adjustability capability to maintain a least delay sub-path between the source to the destination.
- Performance analysis shows that DIR has better results of packet delivery ratio, packet delivery delay, and throughput.
- Future work is to develop
  - A diagonal-intersection-based multicast routing protocol
  - A diagonal-intersection-based delay-bounded routing protocol

Delay-Bounded Routing in Vehicular Ad-hoc Networks

Antonios Skordylis and Niki Trigoni

ACM MobiHoc 2008

Section Outline

- Motivation
- Delay-Bounded Greedy Forwarding (D-Greedy)
- Delay-Bounded Minimum Cost Forwarding (D-MinCost)
- Simulation Result
- Conclusion

Abstract

- The multitude of vehicular applications calls for routing schemes that satisfy **user-defined delay** requirements while at the same time maintaining a low level of channel utilization to allow their coexistence.
- This paper focuses on the development of **carry-and-forward** schemes that attempt to **deliver data from vehicles to fixed infrastructure nodes** in an urban setting.
- The proposed algorithms leverage local or global knowledge of traffic statistics to carefully alternate between the Data Muling and Multihop Forwarding strategies, in order to **minimize communication overhead** while adhering to delay constraints imposed by the application.

Introduction

- Recent trends in Intelligent Transportation Systems show that an increasing number of vehicles will be equipped with wireless transceivers that will enable them to communicate with each other and form a special class of wireless networks, known as *vehicular ad hoc networks or VANETs*.
- Network resources will be shared by applications that provide internet access to passengers, propagate advertisements about nearby places of interest, provide the driver with safety information (e.g. emergency braking) and so on.

Cont.

- We classify VANET-based applications into two categories:
  1. those that require broadcasting of information from one vehicle to many nearby vehicles, e.g. for collision avoidance
  2. those that require the propagation of information hop-by hop to a single destination point or area, e.g. sending an advertisement from an attraction site to a busy intersection, or sending an emergency message from an accident site to the closest roadside unit that is connected to a fixed network.

Cont.

- The focus of this paper is the second class of applications; our motivating example is the *ambient traffic sensor application* wherein vehicles are equipped with sensors that detect accidents, road faults and traffic congestion.
- On detection of an interesting event, vehicles attempt to notify the city's traffic monitoring center, by sending the information to one of the stationary roadside units dispersed in the city. These are referred to as *access points (APs)* and act as gateways to stream traffic information through a fixed network to the outside world.

Cont.

- We note that messages may have very different priorities, and thus delay thresholds until they are delivered to one of the APs.
- For example, information about a serious accident has higher priority than information about a road fault.
- The former must be delivered to one of the APs much faster than the latter, since it calls for immediate assistance from fire, hospital or police departments. It is therefore vital that packet forwarding algorithms are designed to prioritize packets based on their urgency and deliver them within user defined delays.

Goal

- The goal is to design algorithms that try to optimize **bandwidth utilization**, by being frugal in wireless packet transmissions. To do so, we plan to leverage knowledge of traffic information on different parts of the city; our proposed algorithms are traffic-informed and they adapt their behavior depending on the traffic density and the average vehicle speed on different road segments.
- We can therefore argue that in order to bring vehicular networks to their full potential, we must try to satisfy application requirements for bounded delays in packet delivery, whilst trying to minimize the utilization of the wireless medium.
- The key to achieve this goal is to take into consideration statistics of vehicle density and speed in various parts of the city.

Contribution

1. We define the problem of timely and bandwidth efficient data dissemination from vehicles acting as data sources to one of several access points dispersed in the city, given statistical information about road traffic. We carefully study the tradeoff between the competing requirements for timely data delivery and low bandwidth utilization.
2. We propose two novel algorithms, D-Greedy and DMinCost, that exploit traffic information to forward packets to the most convenient access point. D-Greedy exploits only local traffic information, whereas DMinCost leverages traffic information about the entire city. Unlike existing vehicular-assisted forwarding algorithms [16], D-Greedy and D-MinCost do not try to minimize delay of packet delivery. Their goal is to minimize the number of packet transmissions required to satisfy packet-specific delay thresholds.

Cont.

3. In our extensive simulation study, we evaluate the performance of the proposed algorithms in terms of packet delivery ratio and bandwidth utilization, and compare them with the epidemic protocol proposed in [13] and the MinDelay protocol inspired by the VADD protocols [16]. Our experiments are conducted using realistic vehicle traces on a real city map.

MODEL/Assumption

- Upon sensing an interesting event, the vehicle produces a message containing the event description and all event specific information, the message generation time t_g and a time-to-live value λ. The message is considered to be successfully delivered, if it arrives at one of the access points before time $t_g + \lambda$. We will refer to λ as the message *delay threshold* in the rest of this paper.

Objective

- Our objective is thus to devise carry-and-forward algorithms that leverage knowledge of traffic statistics in an urban setting to enable timely delivery of messages from vehicles to stationary access points, whilst minimizing wireless transmissions and optimizing bandwidth utilization.

PROPOSED ALGORITHMS

- We present two novel routing algorithms for VANETs, *Delay-bounded Greedy Forwarding (D-Greedy)* and *Delay-bounded Min-Cost Forwarding (D-MinCost)*.

- The goal of algorithms is to deliver messages originating in vehicles to an access point with bounded delay while **minimizing the number of wireless transmissions**.

Cont.

- D-MinCost requires knowledge of ***global traffic conditions***, i.e. statistical information about the speed and density of cars on every road segment of the city.
  - In this work we do not study the precise process of maintaining a fairly accurate set of urban traffic statistics but rather assume that, when in the vicinity of access point, vehicles can update the preloaded street map with the latest statistical information.
- D-Greedy, on the other hand, requires no such knowledge. It only relies on **local information**, i.e. vehicle speed, to make forwarding decisions.

Cont.

- Our algorithms intend to minimize the number of transmissions while forwarding a message to an access point within the message-specific delay threshold.
- Two forwarding strategies

a) *Multihop Forwarding*, which refers to the aggressive forwarding of messages to vehicles that are better positioned to deliver them to an access point.

b) *Data Muling*, which refers to buffering messages in local memory and carrying them at the vehicle’s speed.

Cont.

- For the **Multihop Forwarding strategy** to be a feasible option, traffic needs to be dense enough so that better positioned vehicles exist within communication range.
- The **Data Muling strategy** is a feasible option as long as the current vehicle is traveling on the path selected by the routing algorithm.
- The novelty of our proposed algorithms lies in their **careful alternation** between the *Multihop Forwarding* and *Data Muling* strategies to achieve a good tradeoff between delay and communication cost.
- This is in stark contrast with the previously proposed VADD protocols, which aim at **minimizing message delay**, and thus always prefer Multihop Forwarding to Data Muling when the former is possible.

Cont.

- An additional difference from existing work is that our algorithms treat each buffered message in a different way depending on its remaining delay budget; the same vehicle may decide to adopt the Multihop Forwarding strategy for one message and Data Muling for another.

Delay-bounded Greedy Forwarding (D-Greedy)

- The D-Greedy algorithm defines a forwarding strategy that assumes no knowledge of traffic information beyond node speed, which can be derived locally from the available location information.
- D-Greedy assumes that the best path to an access point is the shortest one.
  - i.e. the path that minimizes the sum of the lengths of the edges on the directed graph *G* that abstracts the street map.
- When multiple APs exist, the algorithm selects the closest one, i.e. the one on the shortest path beginning at the vehicle’s location.

Cont.

- Each vehicle maintains a neighbor list by periodically broadcasting beacons.
- A beacon message contains the unique vehicle identifier (*id*) and the length of the shortest path between the vehicle's current location and the location of the closest access point (*distToAP*).
- ***distToAP*** is computed by running a single invocation of Dijkstra on *G* just before broadcasting a beacon. As soon as a vehicle senses an event and generates a new message, the message is assigned a **delay threshold value (*TTL*)** and is considered to be useful only if delivered before *TTL* has elapsed.

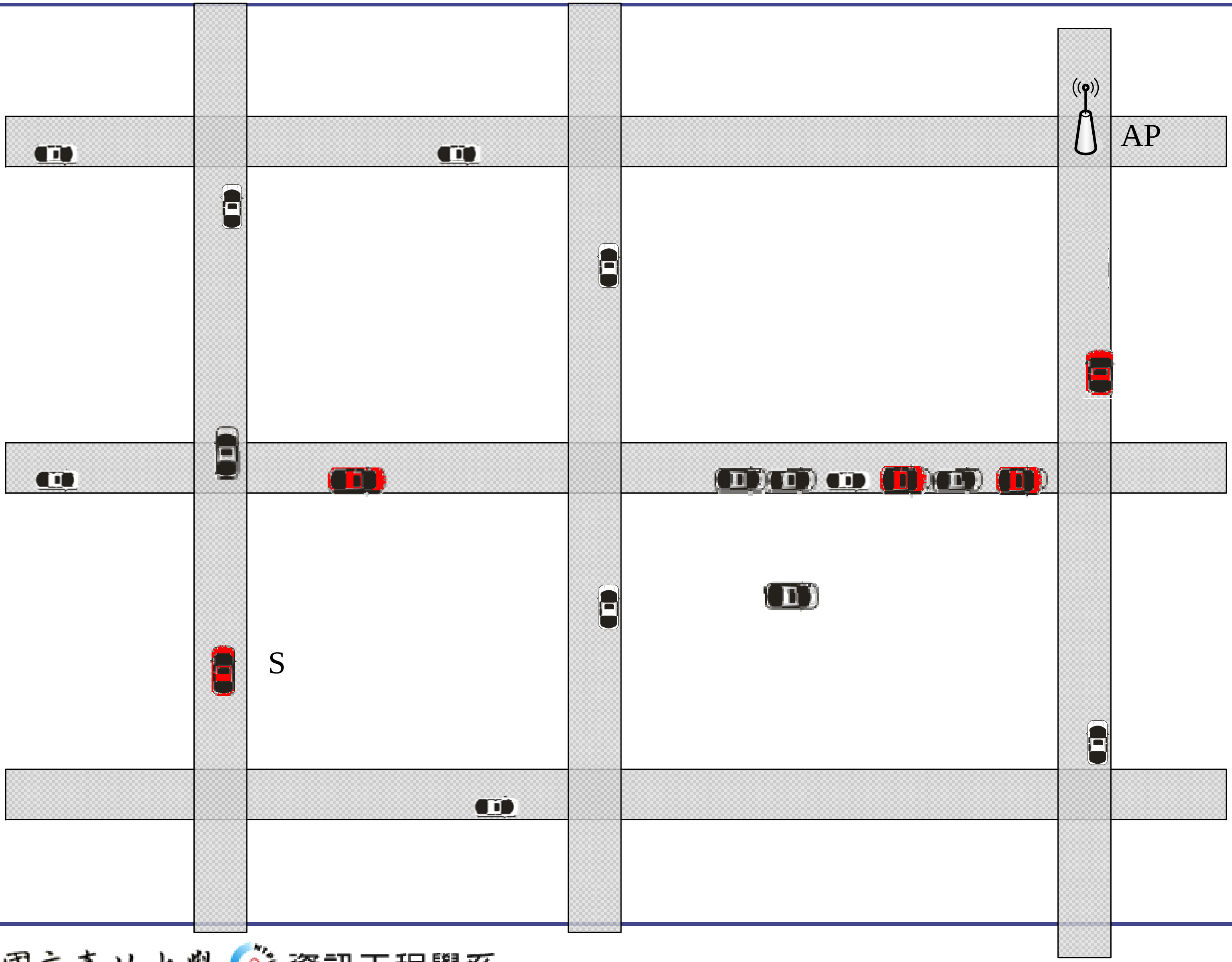
Greedy Strategy Selection

- Vehicles periodically iterate through their buffers and make greedy decisions about the strategy that will be used for forwarding each message to the closest AP.
- The decision depends on the **remaining delay budget (*TTL*)** until the message expires as well as on the distance to the closest AP (*distToAP*).
- Since global traffic information is not available, D-Greedy assumes that the remaining message delay budget can be uniformly distributed among the edges that compose the shortest path to the AP.
- *Each edge on the path is allocated a delay budget that is proportional to its length*.

Cont.

- The algorithm periodically monitors the forwarding progress of each message; as long as the actual time spent by the carrying vehicle that travels along an edge does not exceed the delay allocated to that edge, the Data Muling strategy is selected for the message. Otherwise, the algorithm assigns the Multihop Forwarding strategy to the message.

Delay-bounded Greedy Forwarding



Delay-Bounded Greedy Forwarding

- Let ***distToInt*** be the remaining length, until the next intersection, of the current street segment on which the vehicle is traveling.
- ***distToAP*** denotes the current shortest-path distance from the closest AP.
- ***u*** the average speed of the vehicle calculated during a *k*-second historical window.
- D-Greedy computes the available delay budget *Del* for forwarding the message along the current edge up to the next intersection as follows:

$$Del = TTL \times \frac{distToInt}{distToAP}$$

- D-Greedy calculates the expected delay if the Data Muling strategy were to be used to carry the message to the next intersection

$$Del_{DM} = \frac{distToInt}{u}$$

Cont.

- If $Del_{DM} \leq Del$ then the algorithm opts for the Data Muling strategy.
- Otherwise, the Multihop Forwarding strategy is chosen.
  - In this case, the message is forwarded to the neighboring vehicle in range that is closest to the AP (Figure 2) and it is deleted from the node's buffer.

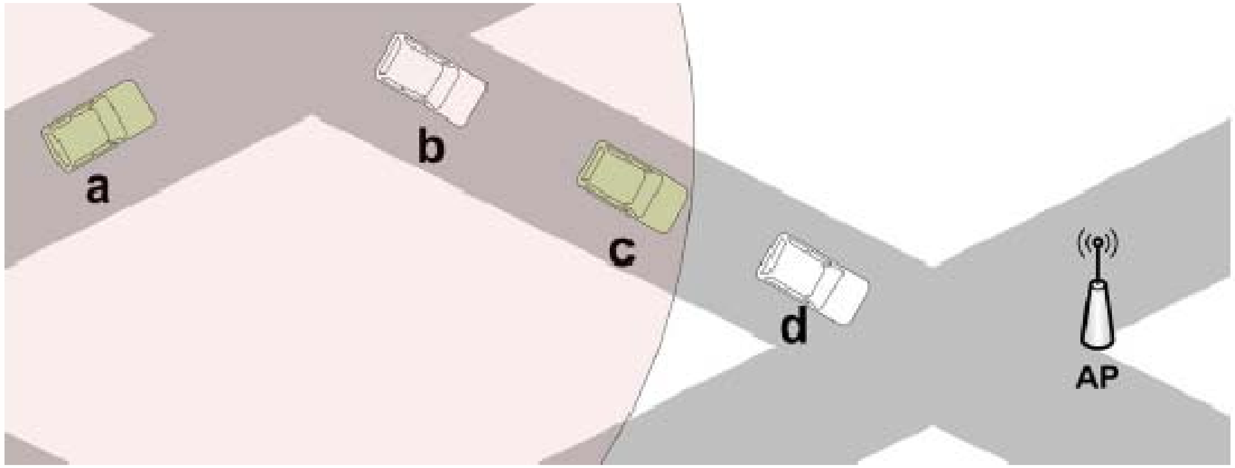


Figure 2: Node a will choose to forward the message to node c, the closest node to the AP among those in range.

Cont.

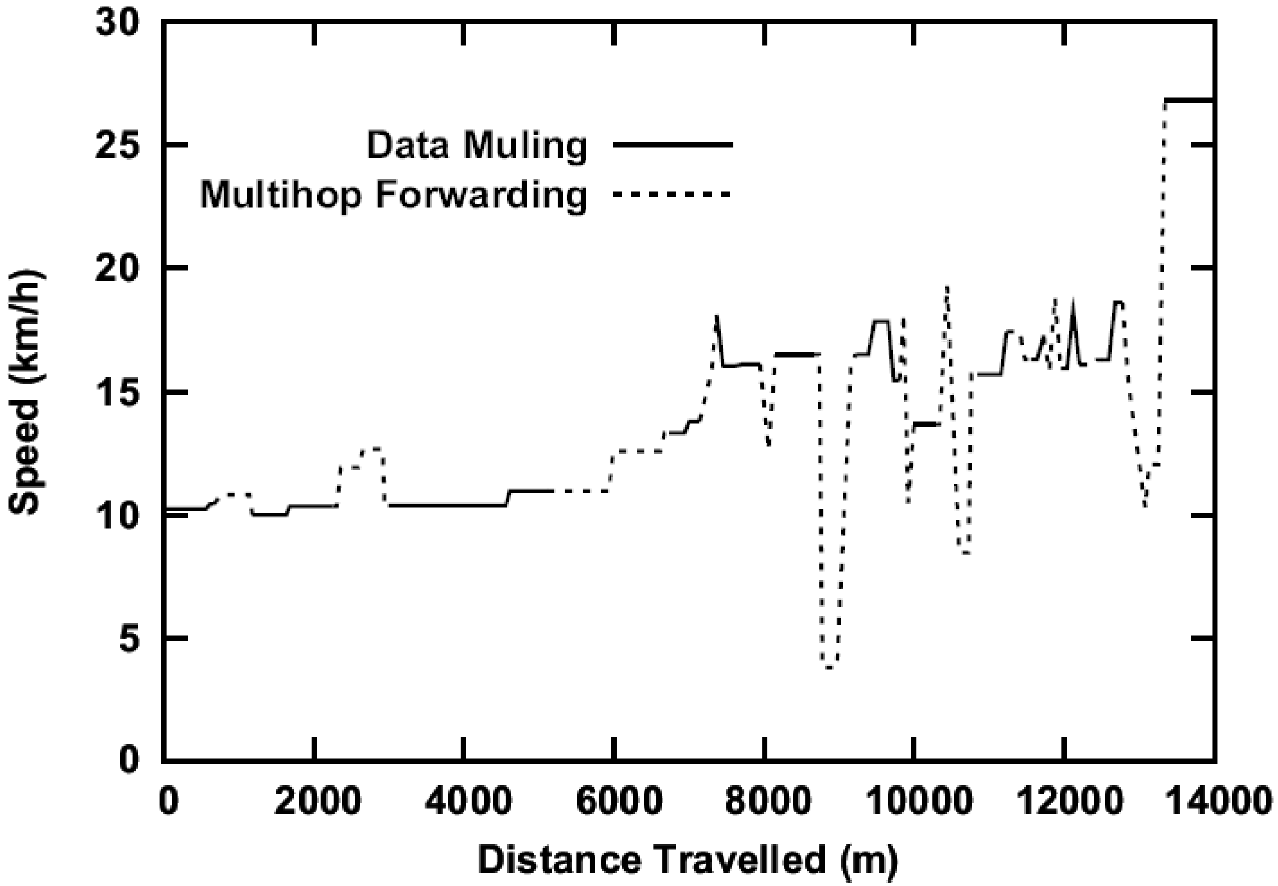
- There are two extreme cases in which a vehicle does not apply the selected forwarding strategy for the message.
  - When there is no better-positioned neighbor node to forward the message than the current node, messages that were originally assigned to use the Multihop Forwarding strategy switch to Data Muling.
  - Similarly, if the carrying vehicle is moving away from the closest AP, messages that were originally assigned to use the Data Muling strategy switch to the Multihop Forwarding strategy until a vehicle traveling towards the AP is found.

Cont.

- Figure 3 shows the strategy selection of D-Greedy in action.
- Observe that when the message is being carried by a vehicle with high speed, it is propagated with the Data Muling strategy, whereas when a vehicle with low speed carries the message, it is propagated with the Multihop Forwarding strategy.
- Data Muling is allowed at lower speeds during the early lifetime of a message because the algorithm **overestimates the delay allocated at each edge**, since it assumes the message will follow the shortest path to the AP.
- As the message progresses through the network, the delay budget tightens and only high-speed carriers are allowed to perform Data Muling.

Fig. 3

- **Correlation between node speed and forwarding strategy**

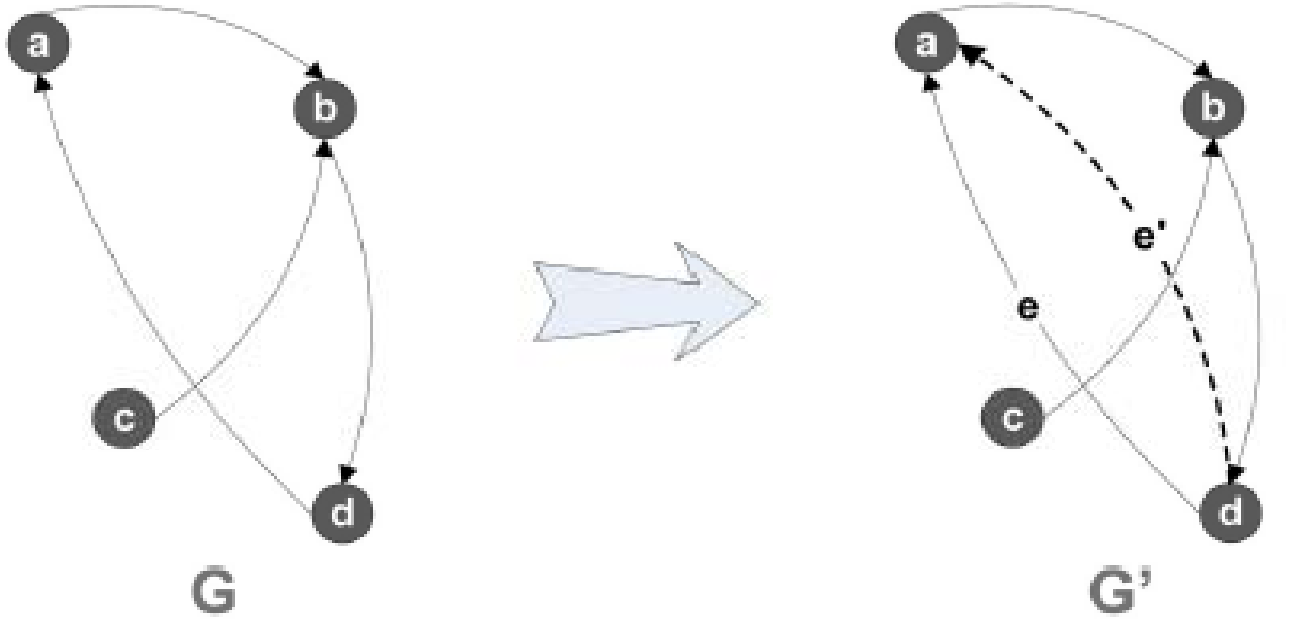


Delay-Bounded Minimum Cost Forwarding (D-MinCost)

- Our second proposed algorithm leverages the knowledge of global traffic statistics, i.e. estimated values of average vehicle speed *u* and density *d* for all edges of the street graph *G*.
- Based on this information, D-MinCost computes bandwidth-efficient delay-constrained paths for every message in the node’s buffer.

Graph extension

- Recall that in the graph that abstracts the street map, **edges** represent road segments and **vertices** represent road intersections.
- We would like to annotate each edge with two metrics:
  1) cost (*C*), representing the **number of message transmissions** along the edge
  2) delay (*Del*), denoting the time required to forward a message along the edge.



Cont.

- However, the cost and delay of forwarding a message along an edge depends on whether we are using the Data Muling strategy or the Multihop Forwarding strategy.

Delay-Bounded Minimum Cost Forwarding (D-MinCost)

- For edges associated with the Data Muling strategy:

$$Del_{DM} = \frac{\ell}{\overline{u}}, \; C_{DM} = 1$$

- where l denotes the length of the edge and $\overline{u}$ the average vehicle speed along that edge.
- We fix the communication cost of the Data Muling strategy to 1 message transmission regardless of the segment length.
- The reason is simple: the vehicle carries the message along the entire road segment, and in the worst case, transmits it only once upon reaching the intersection.

Delay-Bounded Minimum Cost Forwarding (D-MinCost)

- For edges associated with the Multihop Forwarding strategy, we must first check whether multihop is feasible on the road segment.
- For wireless communication range *R*, **Multihop Forwarding** is an available option if $\ell > R$ and $\overline{d} \geq \frac{\ell}{R}$, where *d* is the average vehicle density for the edge in question.

$$C_{MH} = \frac{\ell}{R}, \; Del_{MH} = C_{MH} \times q$$

- ***q*** denotes the time required for the node to check its neighbor list and identify the best next hop.

Path Selection

- After annotating the edges of the extended graph *G* with their corresponding delays and costs, the next step is to choose the minimum cost path, such that the **total delay of the path** does not exceed the **message delay budget**.
- By doing so, we will have not only selected the sequence of edges through which the message should be forwarded, but also the strategy that vehicles must adopt at each edge for the particular message.

Cont.

- The *delay-constrained least-cost routing problem* is known to be NP-complete [6] and various heuristics have been proposed in the literature.
- D-MinCost utilizes one such heuristic, the *Delay Scaling Algorithm (DSA)* [7], in order to efficiently compute delay-constrained least cost paths from the vehicle's location to all access points on the network.

Cont.

- By computing these least cost paths we are able to identify:
  - The access point that can be reached with the least cost.
  - The exact min-cost path to that access point.
  - The strategy that should be followed at each edge of the path in order to adhere to the message's remaining delay budget.
- D-MinCost maintains a neighbor list at each node through periodic beacon broadcasts, similarly to D-Greedy. When a message *p* is generated at the node, the algorithm applies the DSA heuristic on the extended graph *G* for message *p* with delay budget *TTL*. The next intersection *I* is used as the location of the message.

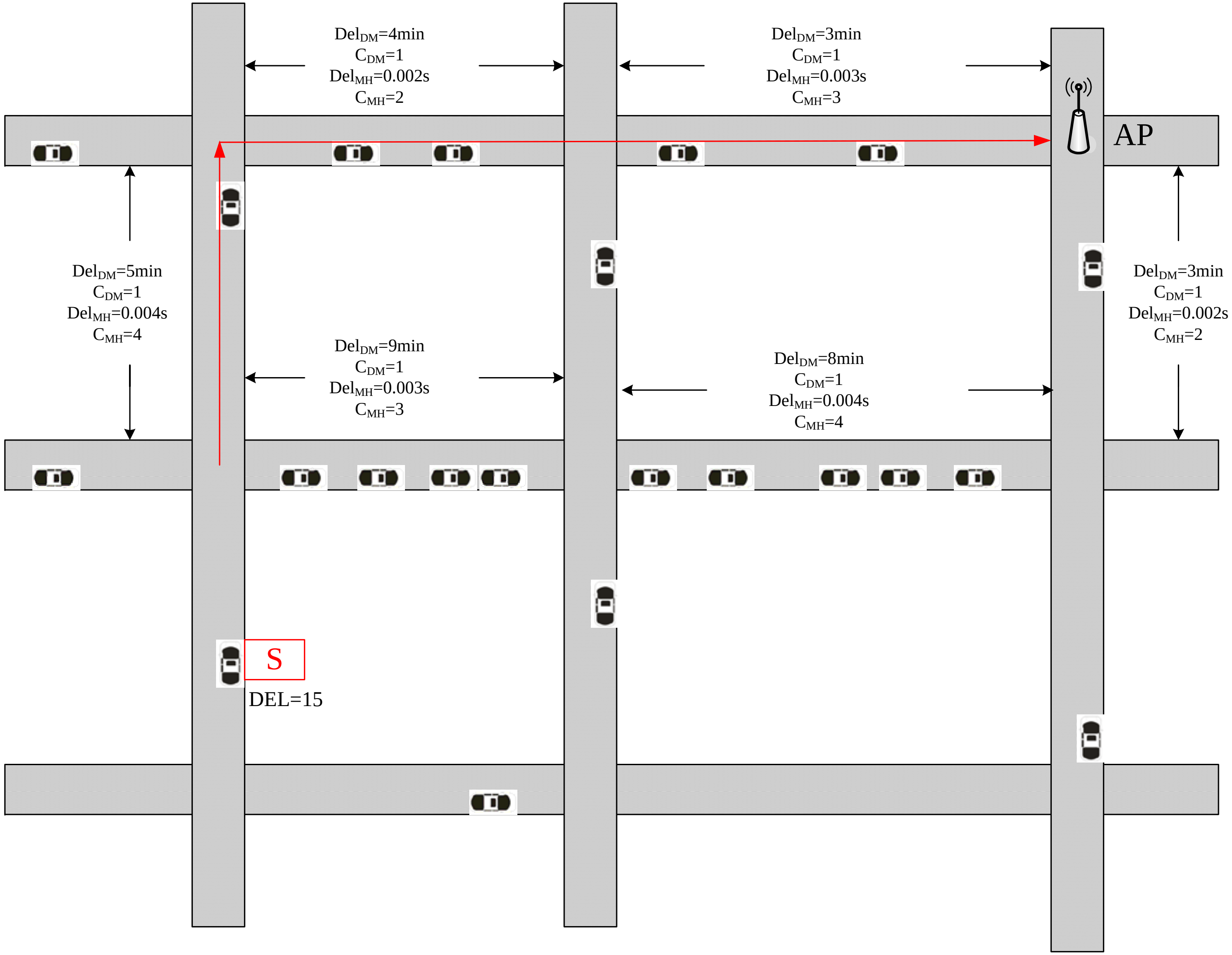
Cont.

- From the paths returned by *DSA*(*I*,*TTL*), D-MinCost selects the minimum cost path that leads to an access point and encodes it in the message header.
- If the first edge of the path suggests the use of Data Muling, the vehicle carries the message until the next intersection *I*.
- Otherwise, the message is forwarded to the neighboring vehicle in range that is closest to *I*.
- Upon successful message reception, the neighbor returns an acknowledgment so that the sending node can remove the message from its buffer.

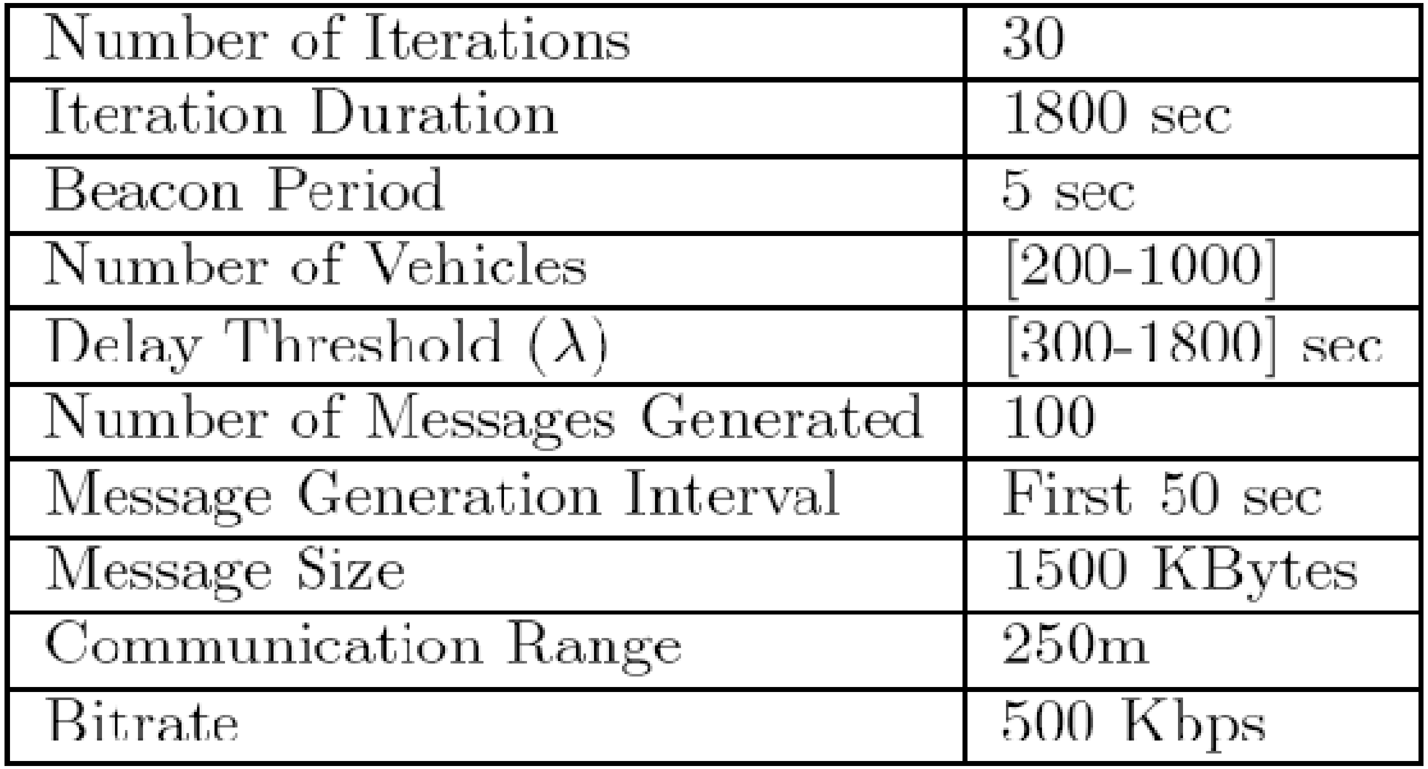
Cont.

- Subsequently, the new message carrier will obey the strategy encoded in the message header together with the suggested path.
- The message path will be recomputed at the next intersection by its carrier only if it is not feasible to follow the suggested edge and its associated strategy.
- This can happen if, for instance, there are no available vehicles on the recommended edge1.
- In this case the edge is removed from graph *G* and the DSA heuristic is reinvoked on the resulting graph in order to compute an alternative min-cost path.

Delay-Bounded Minimum Cost Forwarding (D-MinCost)



Simulation Setup



Number of Iterations	30
Iteration Duration	1800 sec
Beacon Period	5 sec
Number of Vehicles	[200-1000]
Delay Threshold (λ)	[300-1800] sec
Number of Messages Generated	100
Message Generation Interval	First 50 sec
Message Size	1500 KBytes
Communication Range	250m
Bitrate	500 Kbps

Delivery Ratio

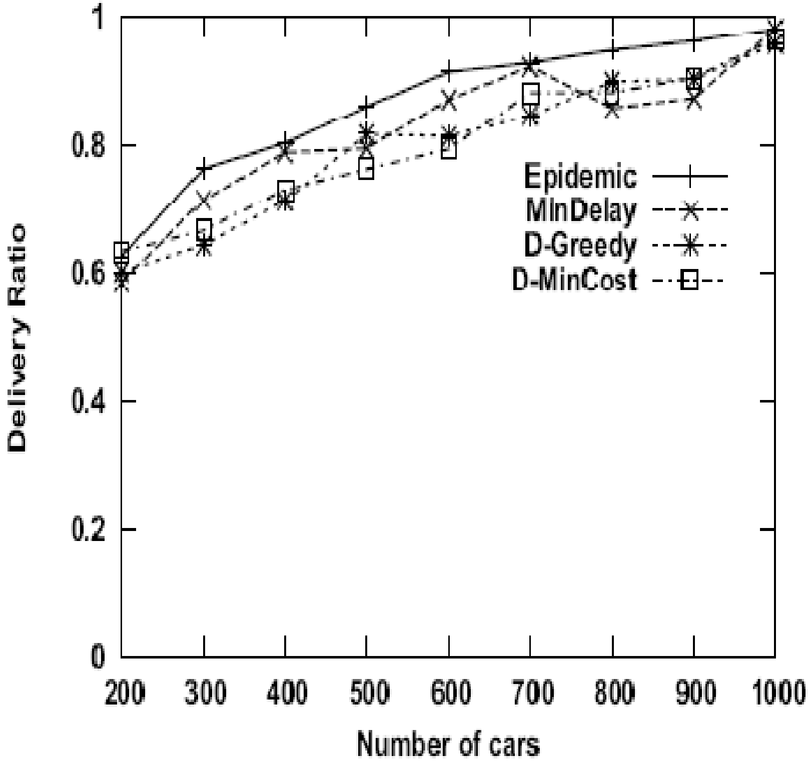


Figure 7: Message Delivery Ratio varying the number of cars (λ=1200sec)

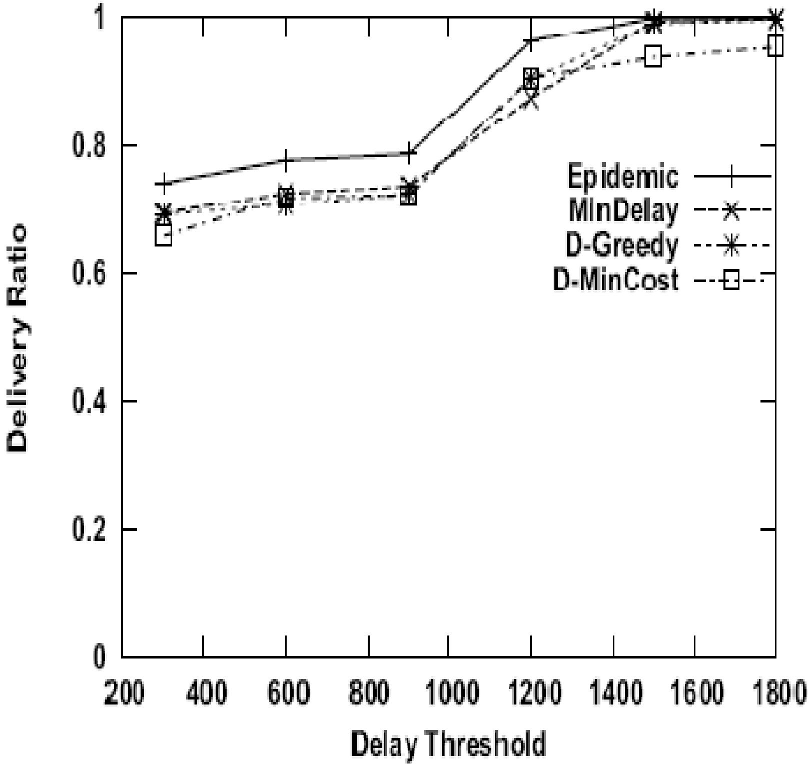


Figure 8: Message Delivery Ratio varying λ (number of cars=900)

Transmitted Bytes

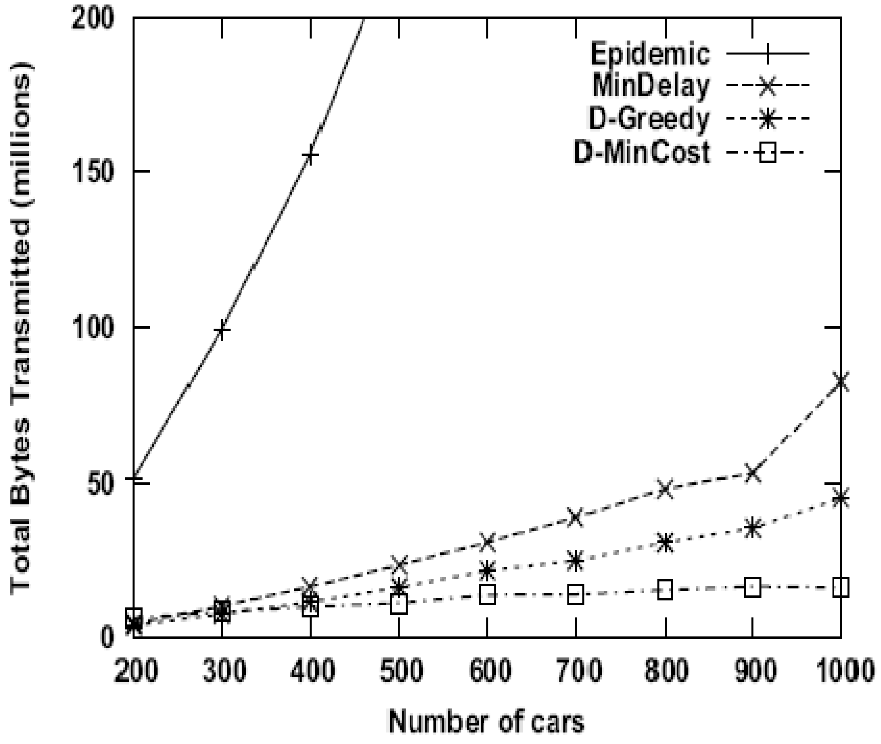


Figure 9: The total number of bytes sent for different car densities (λ=1200sec)

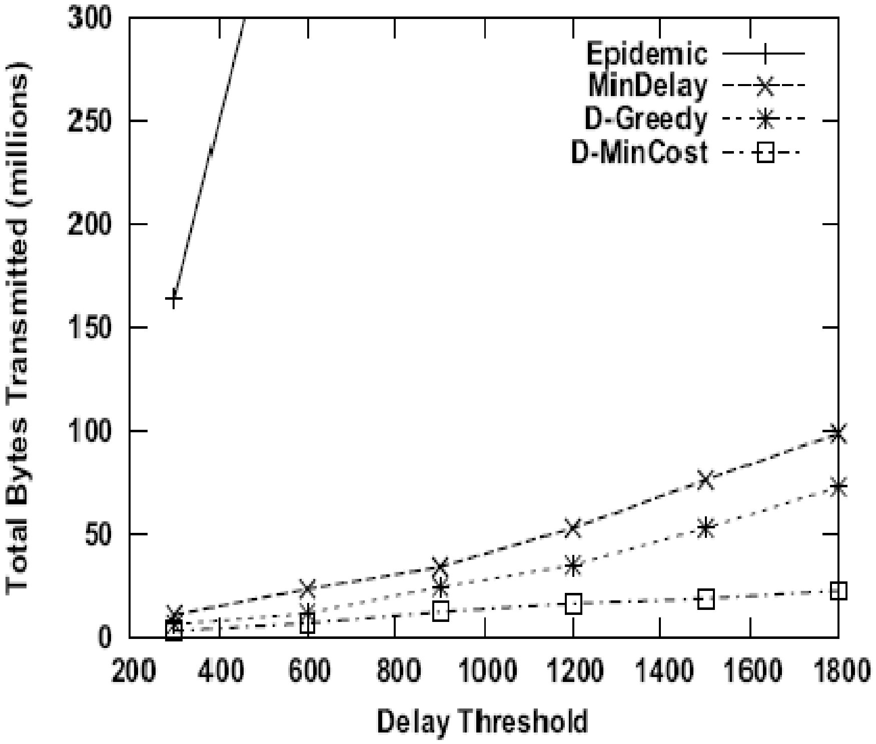


Figure 10: The total number of bytes sent for different values of λ (number of cars=900)

Effect of λ

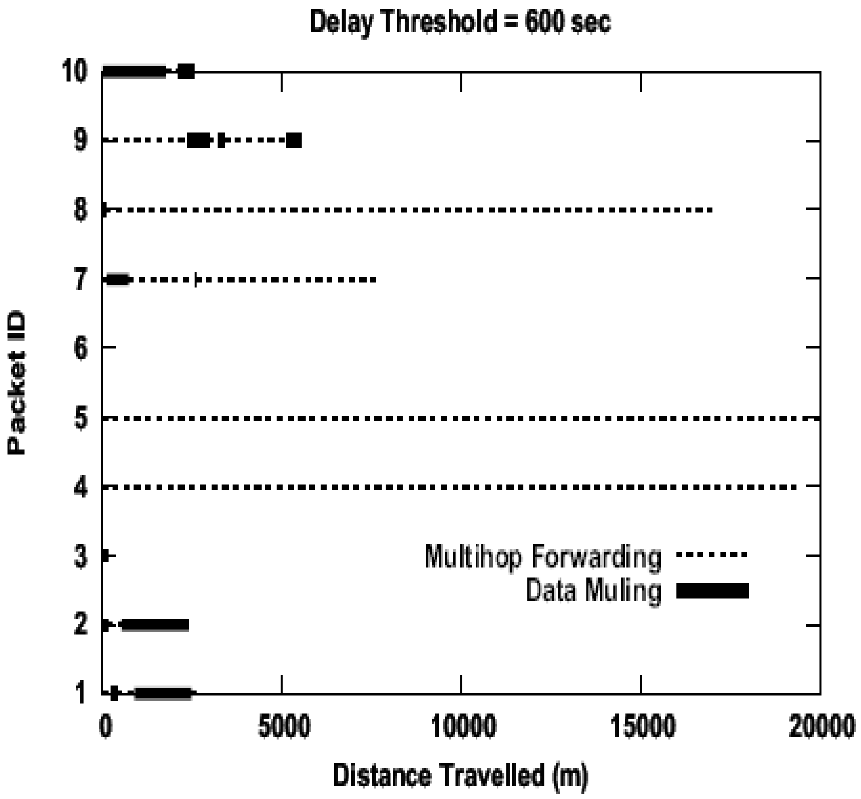


Figure 13: Strategy chosen for low $\lambda = 600s$

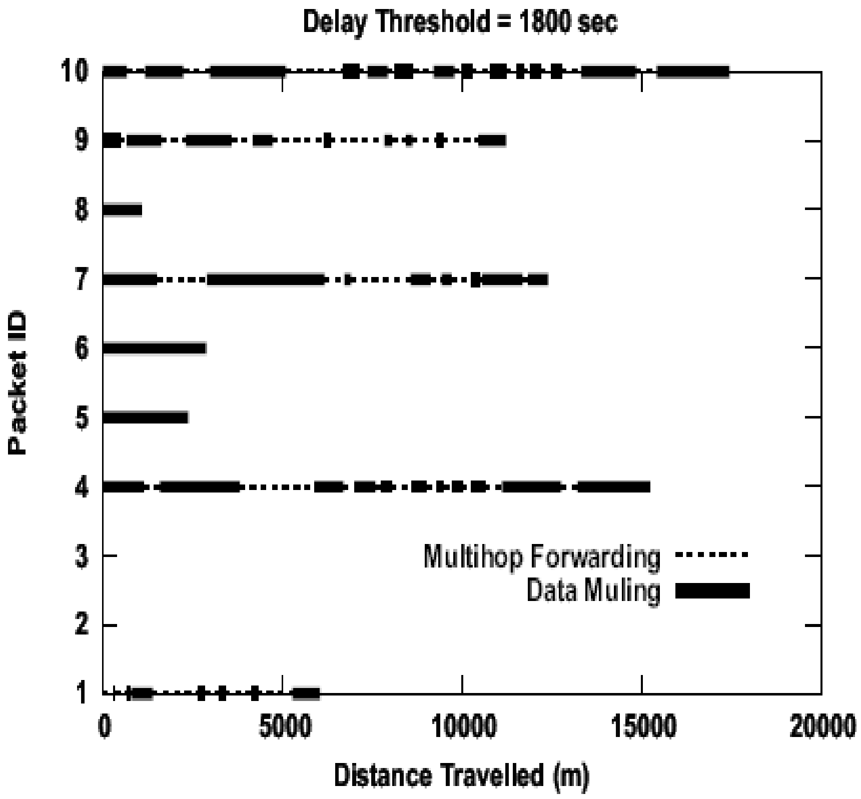


Figure 14: Strategy chosen for high $\lambda = 1800s$

Conclusions

- The cost savings of D-Greedy and D-MinCost are derived from carefully alternating between the Multihop Forwarding and Data Muling strategies while maintaining delivery delay below the required threshold.
- It would be interesting to extend D-MinCost to allow for dynamic updates to a node’s traffic statistics by utilizing the knowledge of neighboring nodes; a node could then obtain very accurate traffic information for the surrounding area, allowing for more efficient path computation and strategy selection.

Homework#10:

1. What is geographic routing in city scenarios ?
2. What is the VADD delay model in VADD Protocol ?
3. What is the CAR routing protocol ?
4. What is the advantage of DIR Protocol ?
5. What's the delay bounded routing in VANETs ?