Efficient Retrieval of User Contents in MANETs

IEEE INFOCOM 2007

Presented by Hanse Huan-Chun Tseng(曾煥鈞)

2008.1.31

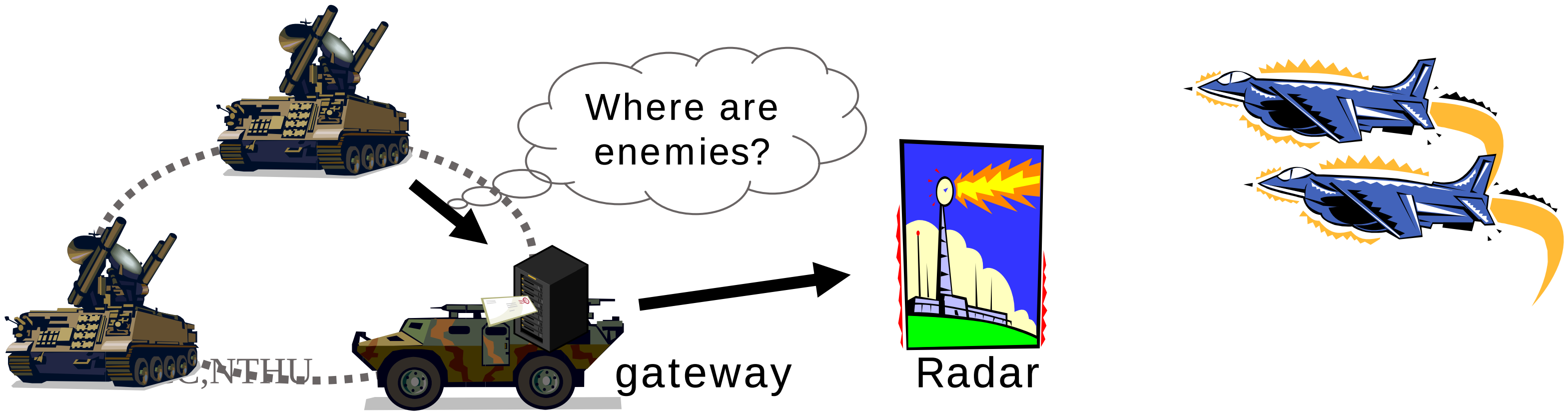
outline

- 1.Introduction
- 2.System Assumptions
- 3.Eureka
- 4.Simulation
- 5.Conclusion

Introduction

- MANET environment
  - Wireless nodes are free to move, join, leave the network-factors
  - Highly-dynamic network system

- MANET applications often require to use resource/services at a gateway or at another user devices



Introduction

- MANET as a peer-to-peer network
  - User nodes not only require **content delivery** but also act as **content providers**

- A visible application of peer-to-peer MANET is in the field of vehicular networks
  - Car-to-Car Peer-to-Peer (C2P2)

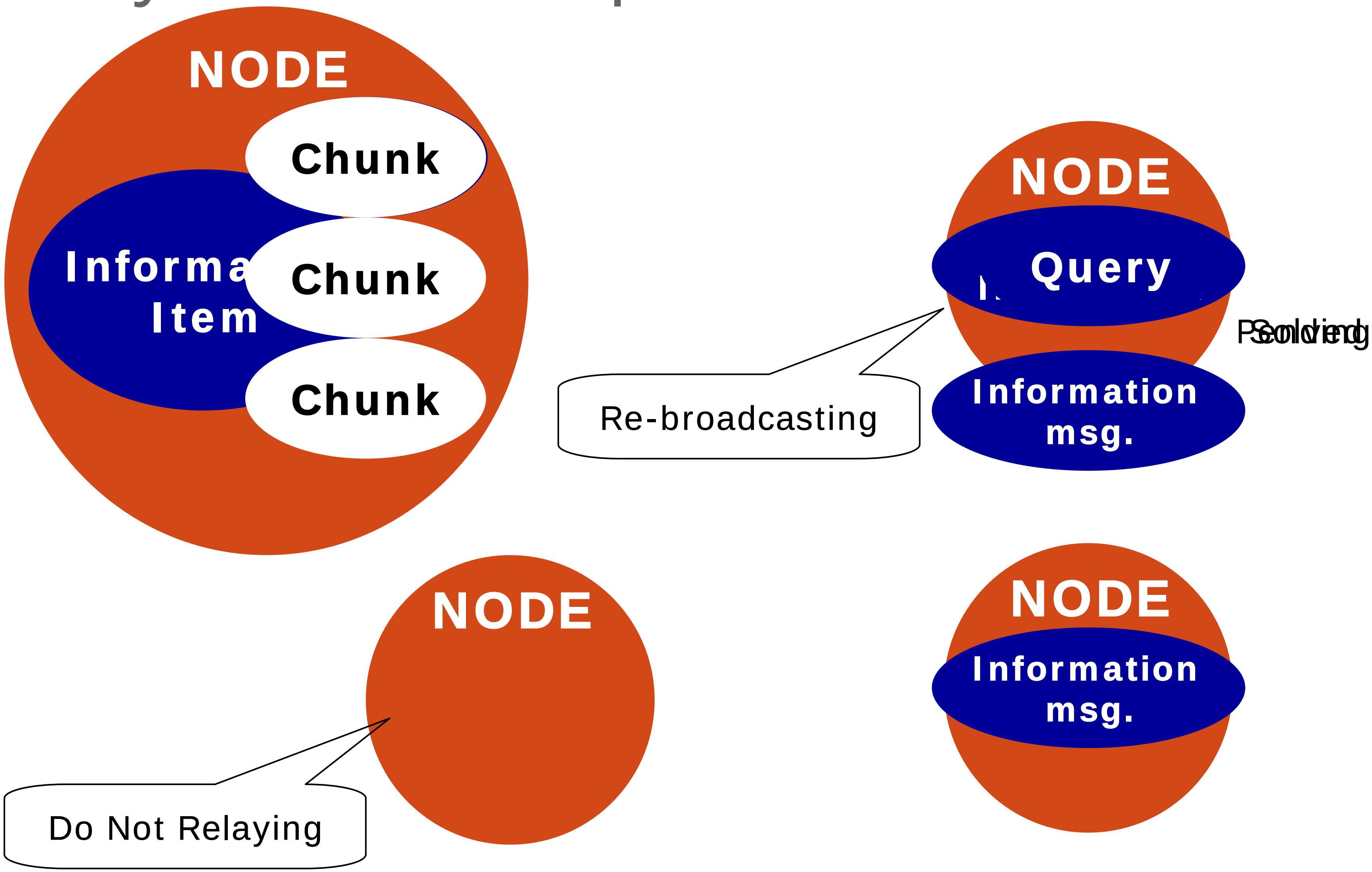
Introduction

- Solution: Eureka
  - Purpose
    - Reduce query/messages overhead from broadcast storms in a cooperative MANET environment
  - Concept
    - Exploit the information density
    - Methodology to identify regions of the network where the required information is more likely to be stored and steer queries there

System Assumptions

- Basic Features
  - Finite set of information items: each item has an identifier $0 \leq i \leq N$ and it is divided into chunks
  - Nodes query other nodes for missing information item with rate λ. Each node generates queries for as many as *C* chunks of the missing items
  - Queries are broadcast and relayed by other nodes
  - Nodes with all, or part, of the requested information chunks reply to the query. Chunks are returned through an application-driven unicast path
  - Information chunks retrieved by the requesting node are locally cached and then dropped at a rate of μ chunks per second

System Assumptions



System Assumptions

- Message exchange
  - Queries
    - Broadcast by requesting and relaying nodes
    - Before relaying a query, a node stores its routing details (ID, src add, last add) and set status for the missing chunk to "pending"
  - Responses
    - MAC-layer unicast transmission by relaying nodes.
    - Next-hop chosen by routing tracks stored at relay node applications
    - Promiscuous mode are allows overhearing nodes to toggle the status of pending queries (pending-> solved)
      - Avoiding the relay of duplicated information msg.

Eureka

- Eureka and Information density
  - Provide each node in the network with an estimate of information density in its proximity
  - Difference between local and other nodes' estimates are used to decide whether a query must be forwarded
  - Thus, queries travel toward areas where it is more likely that information is found

Eureka

- Information density estimation
  - Fully distributed process, Run by all nodes
  - Information Density Function $\delta_i(x,y)$
    - Spatial density of information item *i* , cached at nodes around point(x,y)
    - We measure the information density in *copies/m²(in case of uni-dimensional topologies such as a* highway scenario, we consider $\delta_i(x)$ *measured in copies/m).*

^

- Information density estimation
  - Two contributions
    - $S^{l}_{i,j}$: locally-computed density sample from generated, overheard and received information messages of item *i at step j*
    - $S^{d}_{i,j}$: density sample computed from advertised samples by neighboring nodes
  - A Moving Average(MA) filter weights the contributions and returns the density estimate $\delta_i(x,y)$ ^
  - It is not important to us that the estimates match the absolute values of the actual density

Eureka

- Three techniques to counter the drawbacks of flooding
  1. Queries are issued with a Time-To-Live(TTL)
  2. Relay nodes delay query forwarding by a Query Lag Time, in the hope that a response is returned by neighboring nodes

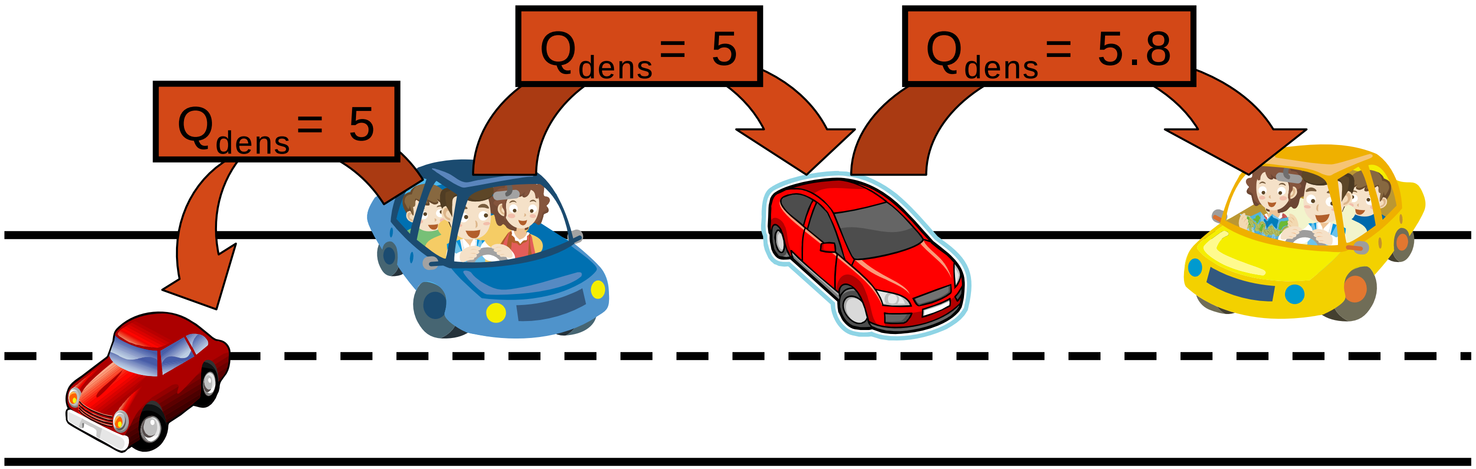
Mitigated Flooding

Eureka

- Three techniques to counter the drawbacks of flooding
  1. Queries are issued with a Time-To-Live(TTL)
  2. Relay nodes delay query forwarding by a Query Lag Time, in the hope that a response is returned by neighboring nodes
  3. Queries include information density as seen by issuing node; relay nodes forward query only if they “see” higher information density

Eureka

- Three techniques To counter the drawbacks of flooding
  3. Queries include information density as seen by issuing node; relay nodes forward query only if they "see" higher information density



Density : 3.5 **Density : 5** **Density : 5.8**

It updates its density estimate
But, does ***not*** forward the query

It updates its density estimate
and forward the query

Information Density Estimations

- *A. Local information density sample*
  - $S^{1}_{i,j}$: locally-computed density sample from generated, overheard and received information messages of item *i at step j*

$$1-\frac{h_Q-1}{TTL}$$

- h_Q is equal to the number of hops covered by the query
- Range between $\frac{1}{TTL}$ to 1

Information Density Estimations

- B. *Distributed information density sample*

$$s^d_{i,j}(n) = \frac{\sum_{m \in \mathcal{M}_{i,j}(n)} s^l_{i,j}(m)}{|\mathcal{M}_{i,j}(n)|}$$

  - $M_{ij}(n)$ *is the set of neighbor nodes which advertised* their local sample to node *n*, *for information item i and* sampling step *j*

Information Density Estimations

- *C. Overall information density sample*

$$s_{i,j}(n) = \frac{s^{l}_{i,j}(n) + s^{d}_{i,j}(n)}{2}$$

- *D. Filtering and information density estimate—using MA filter*

$$\hat{\delta}_{i,j}(n) = \sum_{k=1}^{W-1} \left(1 + \alpha^{W} - \alpha^{W-k}\right) s_{i,j-k}(n) + \left(1 + \alpha^{W} - \alpha\right) \sum_{k=W}^{j} \alpha^{k-W+1} s_{i,j-k}(n) \quad (1)$$

$$\hat{\delta}_{i,j}(n) = \sum_{k=1}^{W-1} \left(1 + \alpha^{W} - \alpha^{W-k}\right) s_{i,j-k}(n) + \left(1 + \alpha^{W} - \alpha\right) \sum_{k=W}^{j} \alpha^{k-W+1} s_{i,j-k}(n) \quad (1)$$

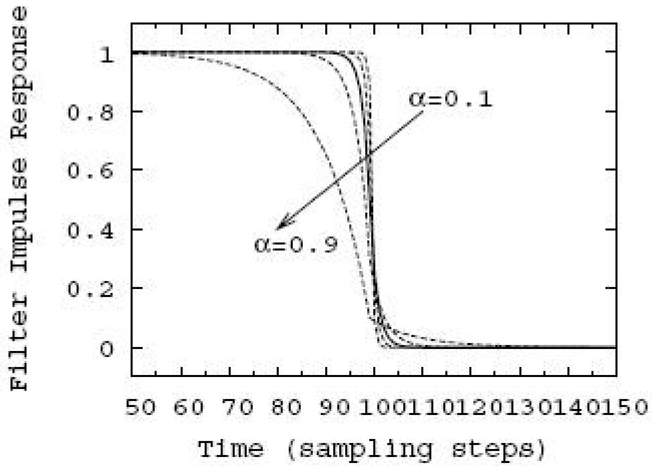


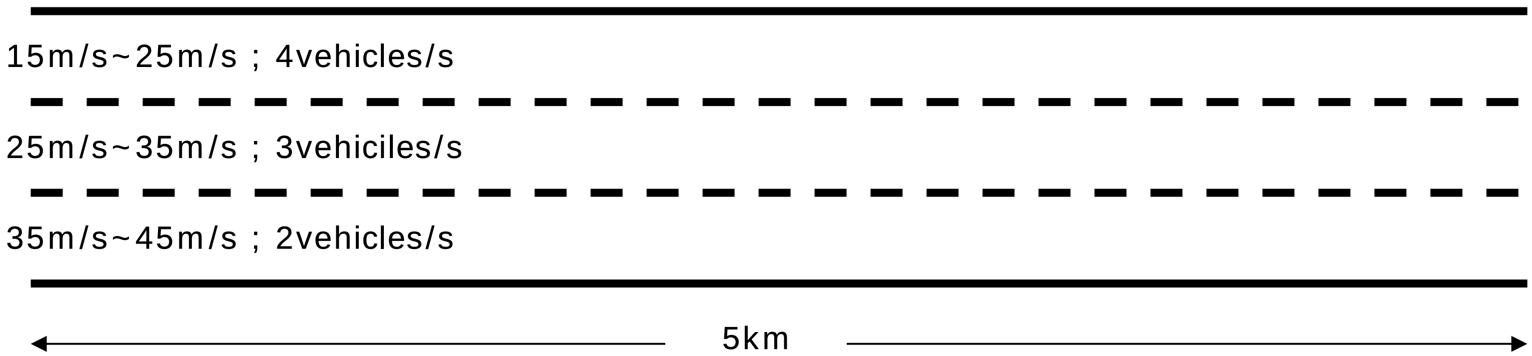
Fig. 2. MA filter impulse response, with $W = 100$ and varying α

Simulation

- Performance evaluation
  - Simulation with ns-2
  - Scenario parameter
    - Information set cardinality, N=[1,25]
    - Each information item divided into up to 30 chunks
    - Queries generation rate, λ=[1,6] queries/ms
    - Cached information drop rate, μ=[5,50] drops/ms
    - Queries Time-to-Live, TTL=10
  - Topologies
    - Highway
    - Urban
  - Nodes have a 802.11 MAC layer, radio range : 100m

Simulation

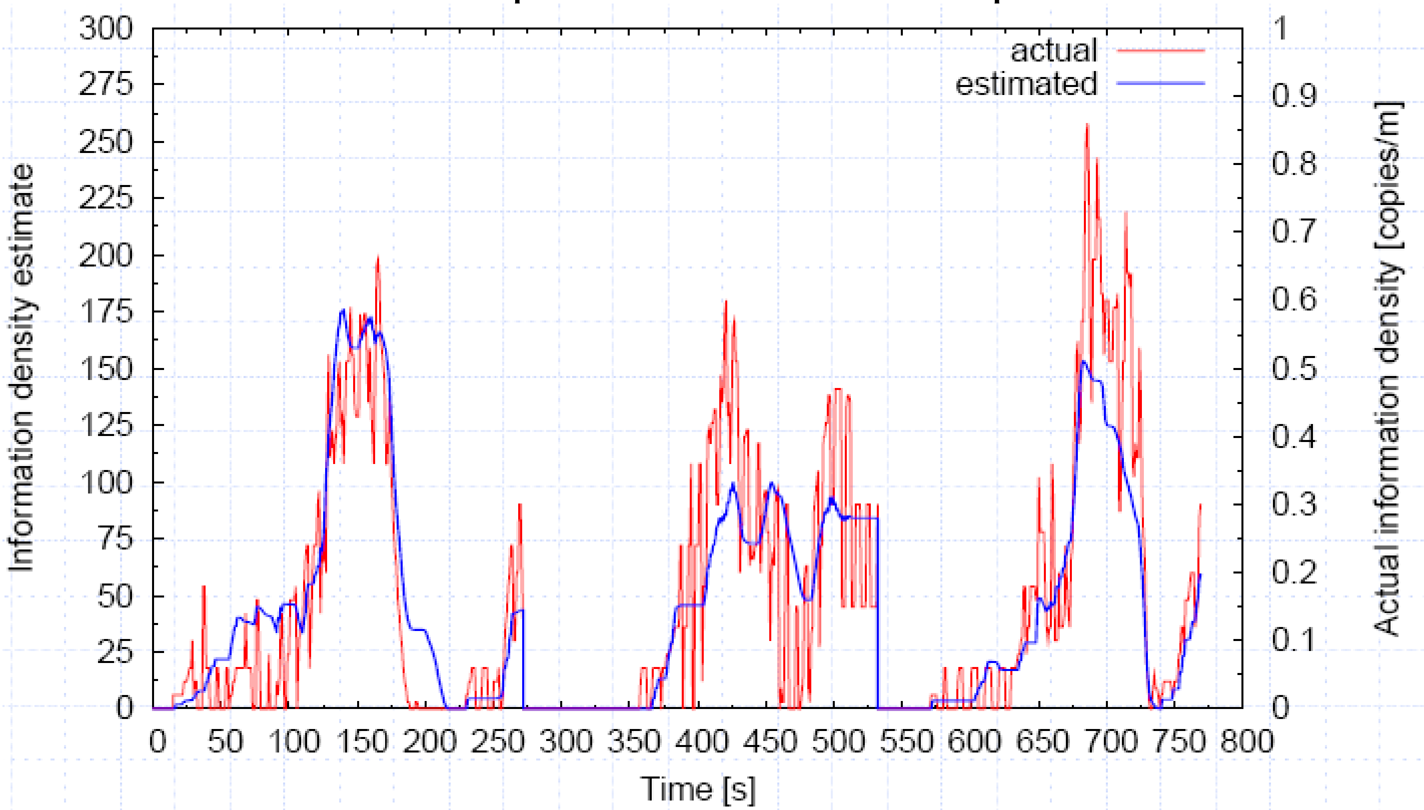
- Road Scenario : Highway
  - 5km straight, unidirectional road with three parallel lanes at different speeds
  - Lane changes allowed when overtaking
  - Speeds ranging from 15m/s to 45m/s
  - An “information gateway” node is located halfway along the road



Simulation

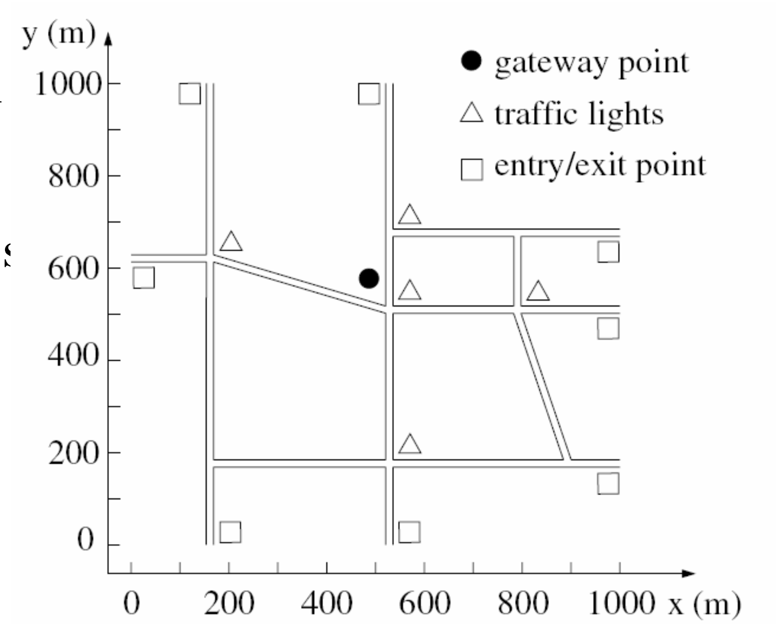
- Estimation Accuracy – Highway

λ= 3queries/ms ; μ= 25 queries/ms



Simulation

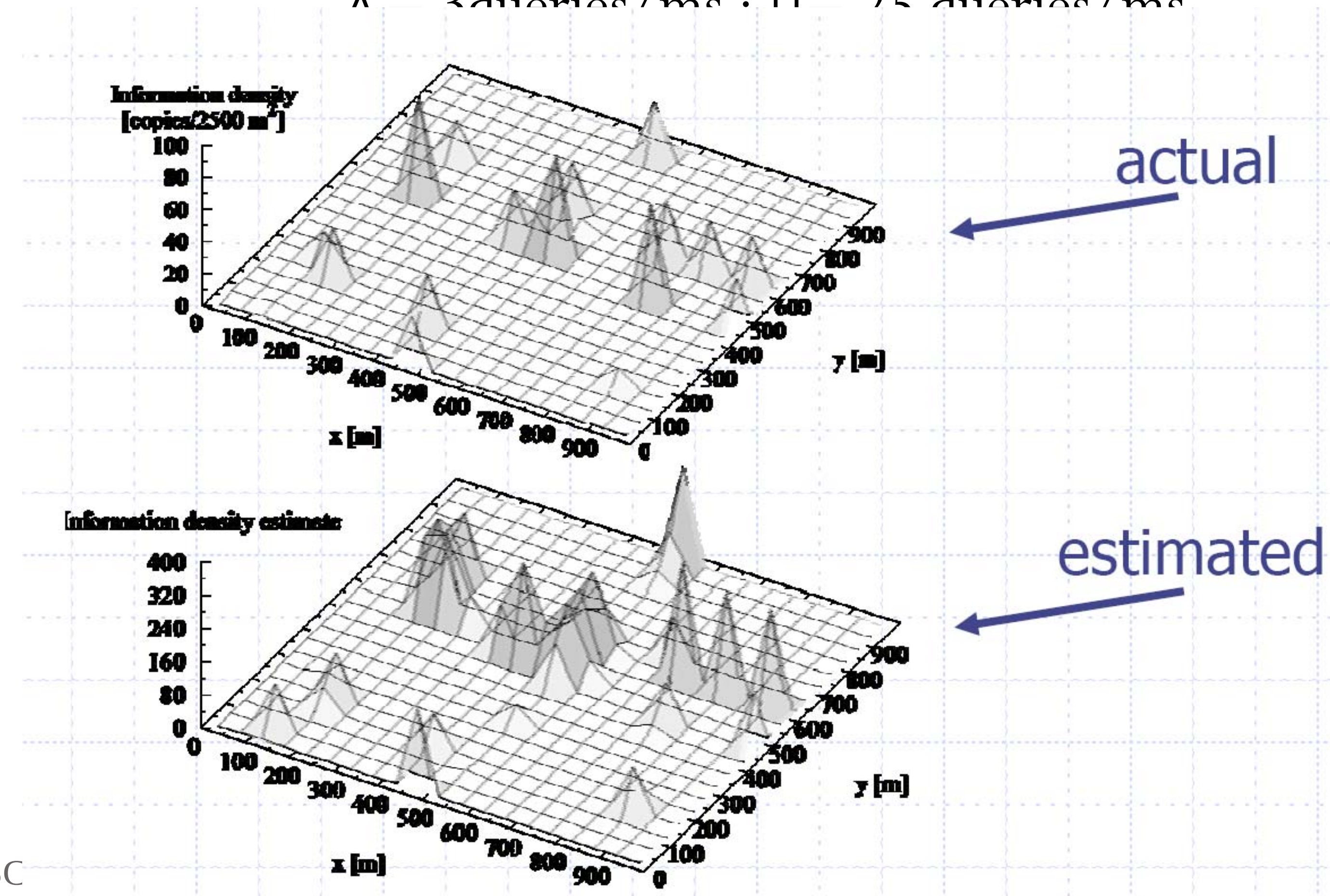
- Road Scenario : Urban
  - **CanuMobiSim**
  - City sections with traffic lights and stop signs at intersection.
  - Vehicles enter/exit at random from entry/exit points
  - Speeds raging from 10m/s to 20m/s
  - On average, 70 vehicles at 5.8m/s
  - One “information gateway”

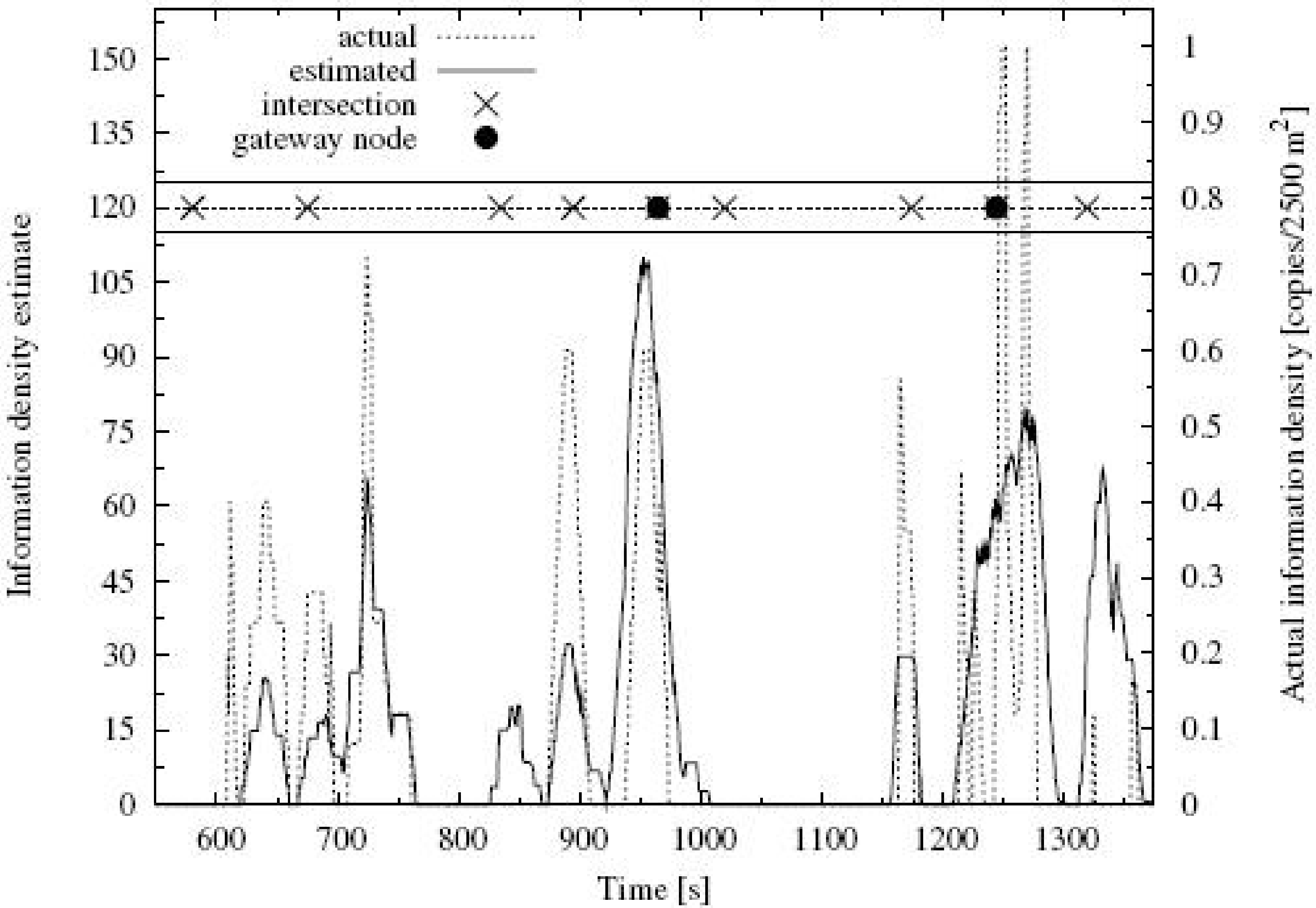


Simulation

- Estimation Accuracy – Urban

$\lambda = 3$queries/ms ; $\mu = 25$ queries/ms




actual
estimated
intersection
gateway node
Information density estimate
Actual information density [copies/2500 m²]
Time [s]
150
135
120
105
90
75
60
45
30
15
0
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
600
700
800
900
1000
1100
1200
1300

Conclusion

- Eureka provides effective overhead reduction
- User use it to direct queries toward areas where the requested information is denser
- Broadcast storms are prevented and congestion is reduced

Thank You !