BeeJamA

Honey Bee Inspired Traffic Jam Avoidance

Highly Dynamic and Adaptive Traffic Congestion Avoidance in Real-Time Inspired by Honey Bee Behavior

H. F. Wedde, S. Lehnhoff, B. van Bonn, Z. Bay, S. Becker, S. Böttcher, C. Brunner, A. Büscher, T. Fürst, A. M. Lazarescu, E. Rotaru, S. Senge, B. Steinbach, F. Yilmaz, T. Zimmermann

08.12.2007

Fraunhofer Institute Material Flow and Logistics

Content

- Introduction
- Distributed Traffic Model
  - Quality Rating Function
  - Distributed Traffic Jam Avoidance (The *BeeJamA* Algorithm)
- Simulation Studies
  - Experimental Results
- Conclusion
- Future Work

Introduction

- Traffic congestion – a highly dynamic *distributed* problem in metropolitan areas world-wide.
- Serious and complex problems regarding the timely arrival of goods or persons.
- Expertise in developing distributed routing algorithms from Nature (*BeeHive, BeeAdHoc*)
  - Dynamic multipath routing algorithms
  - High adaptability / flexibility
  - High throughput
  - High fault tolerance
- Distributed *on-line* traffic control model →***BeeJamA***

Distributed Traffic Model

- Individual hop-to-hop routing
- Probabilistic next hop determination
- Regionally responsible ***navigators***
  - Management of local routing tables
  - Real-time response
- Layered routing model
  - ***Area-Layer***: nodes correspond to intersections, edges correspond to roads
  - ***Net-Layer***: nodes correspond to areas, edges represent roads connecting adjacent areas

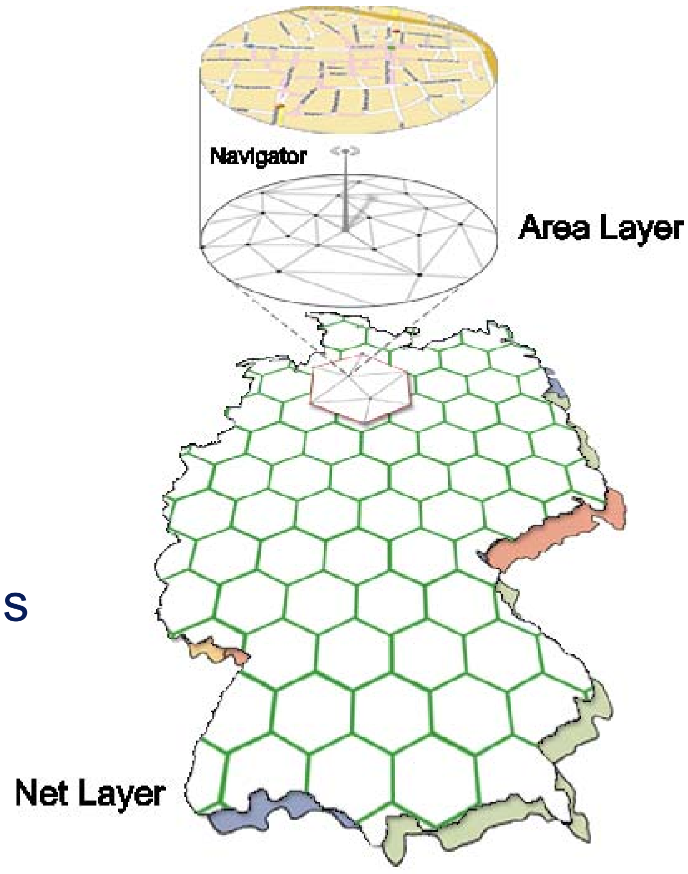
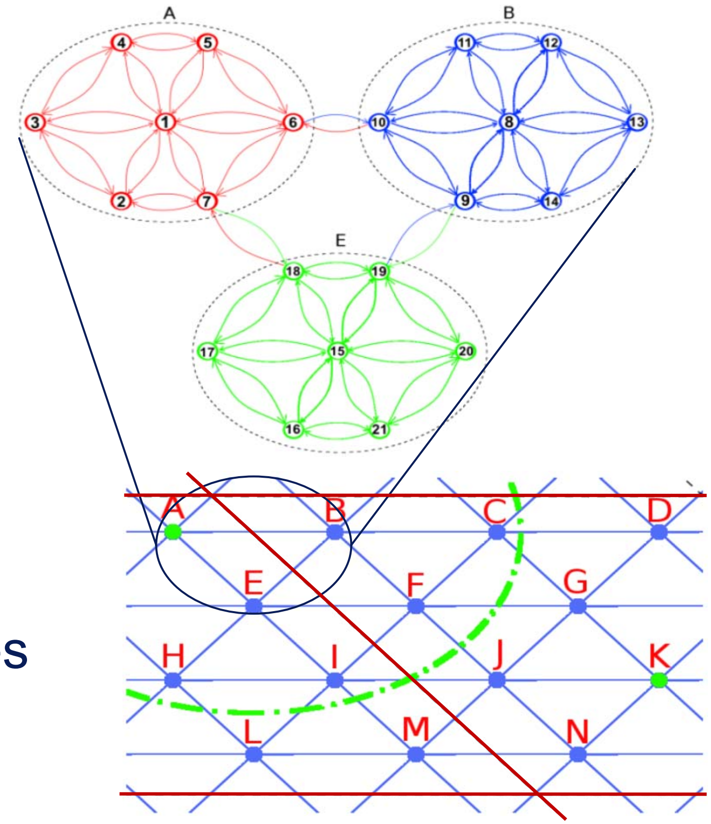


Figure 1: Routing Layers

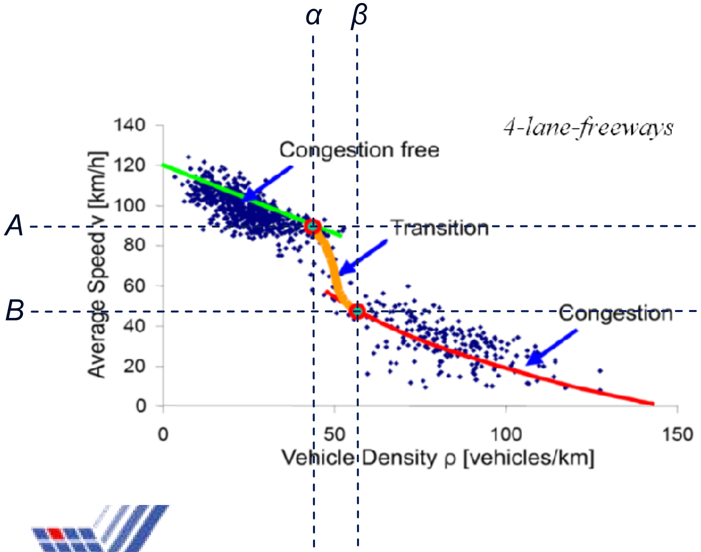
Distributed Traffic Model (Partitioning)

- ***Net-Layer***: partitioning into *Foraging Zones* (→IFZ_{Net}-table) and *Foraging Regions* (→IFR_{Net}-table)
  - FZ(*N*): all nodes (areas) within a limited hop range of node (area) *N*
  - FR: fixed partitions with a single *Representative Node* (area)
- ***Area-Layer***: partitioning into areas
  - Size must allow for sufficient routing alternatives
  - Real-time constraints must be met
  - FZ: each node's FZ consists of all nodes within its area (→IFZ_{Area}-table)
  - *Border-nodes* connect adjacent areas



Quality Rating Function

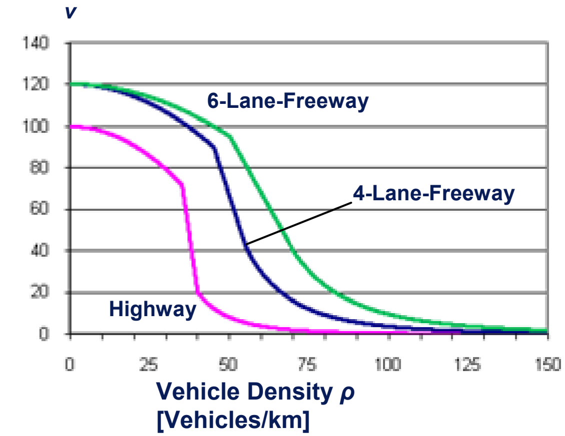
- Routing tables reflect estimated travel times
- Extensive empirical studies in traffic congestion development
- Functional dependency between vehicle speed and density on a road section



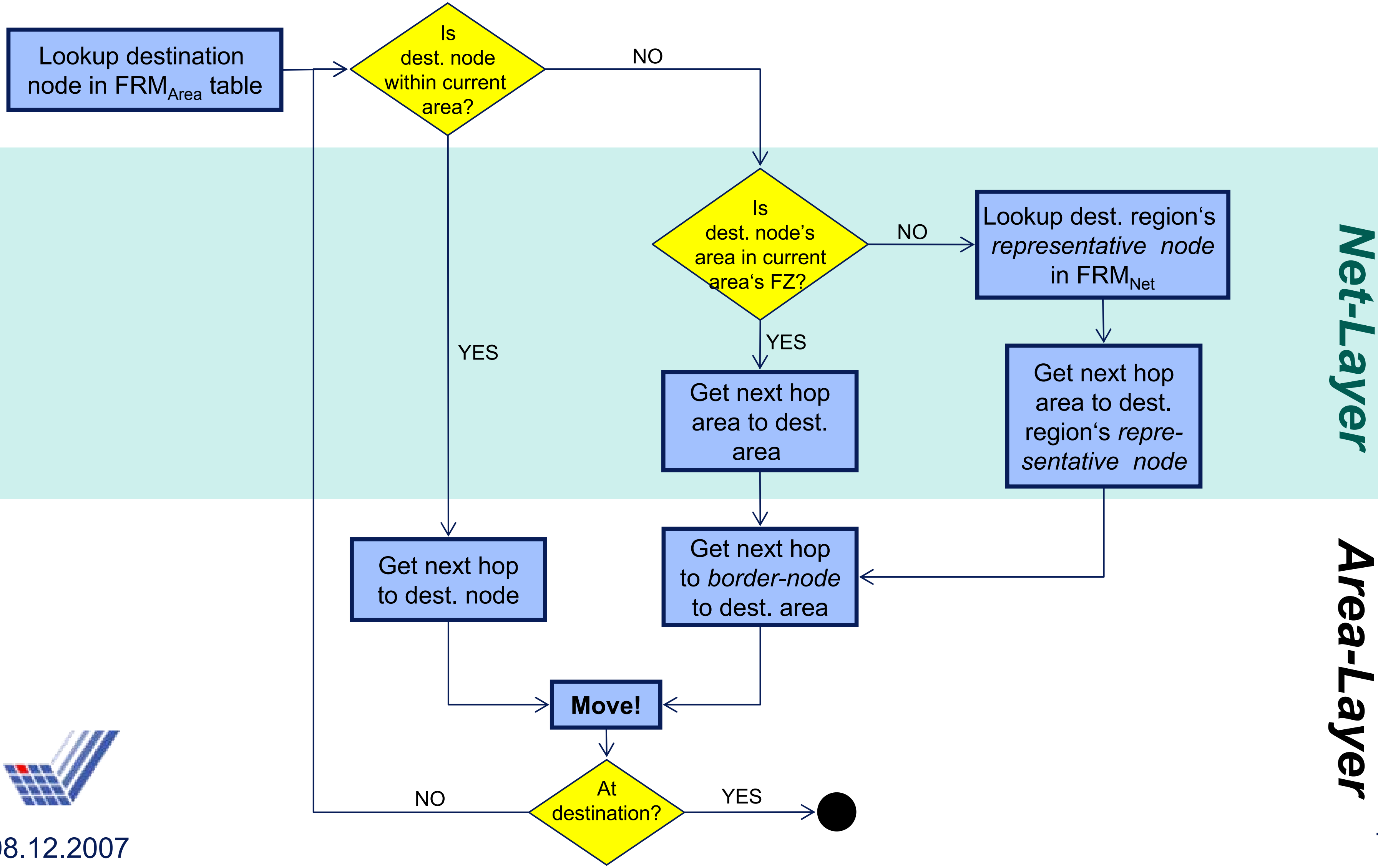
$0<\rho\leq\alpha$: $$v_{Edge} = \frac{(A - v_{\max})}{\alpha^2}\rho^2 + v_{\max}$$

$\alpha<\rho\leq\beta$: $$v_{Edge} = \frac{(\rho-\alpha)(B-A)}{\beta-\alpha} + A$$

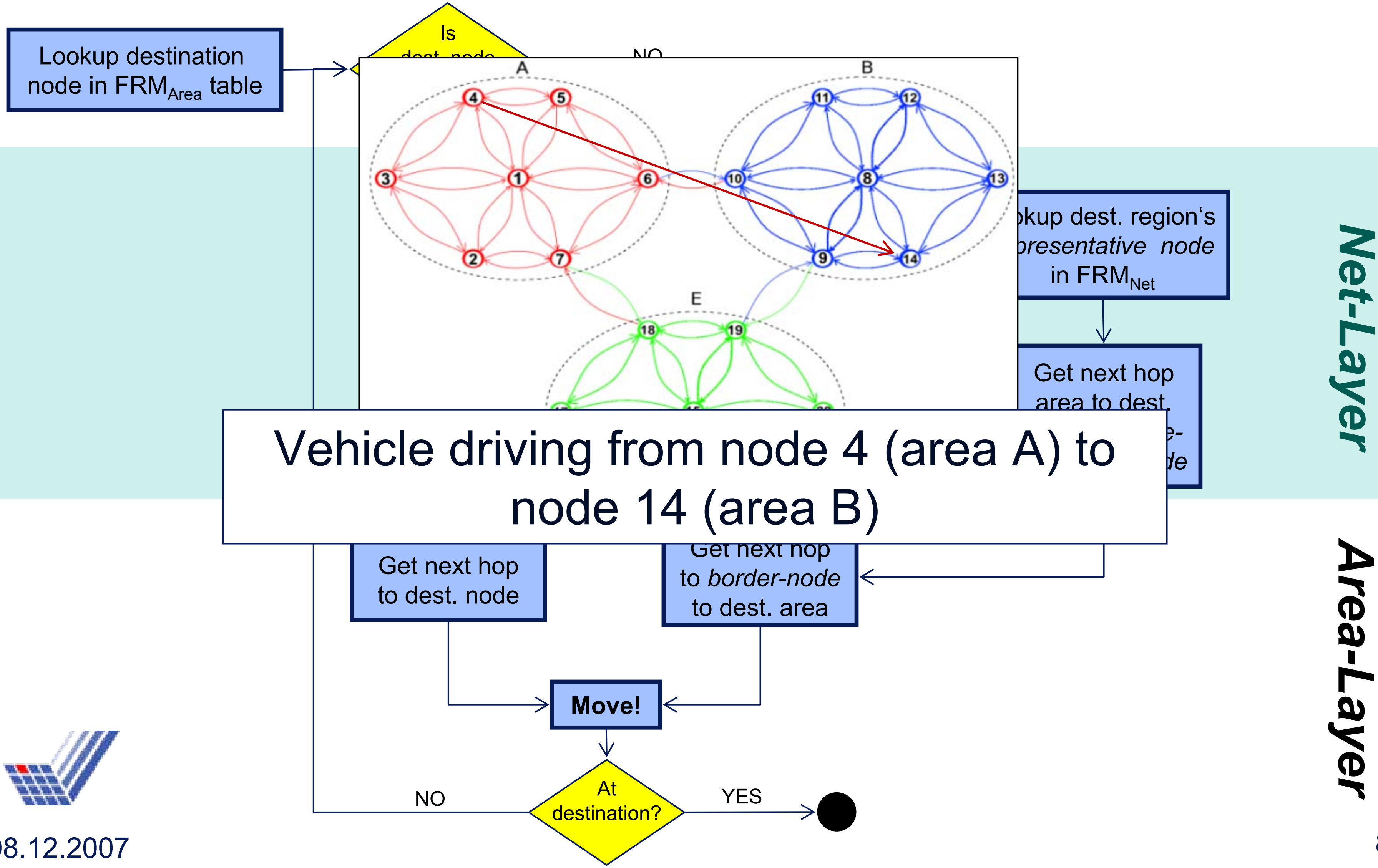
$\rho>\beta$: $$v_{Edge} = \frac{\beta^4}{\rho^4}B$$



The *BeeJamA* Algorithm



The *BeeJamA* Algorithm



Vehicle driving from node 4 (area A) to node 14 (area B)

4→14

The *BeeJamA* Algorithm

Lookup destination node in FRM_{Area} table

Is dest. node within current area?

NO

YES

FRM_{Area}	1	…	7	8	…	***14***	15	…
	A	…	A	B	…	***B***	E	…

Get next hop to dest. node

Move

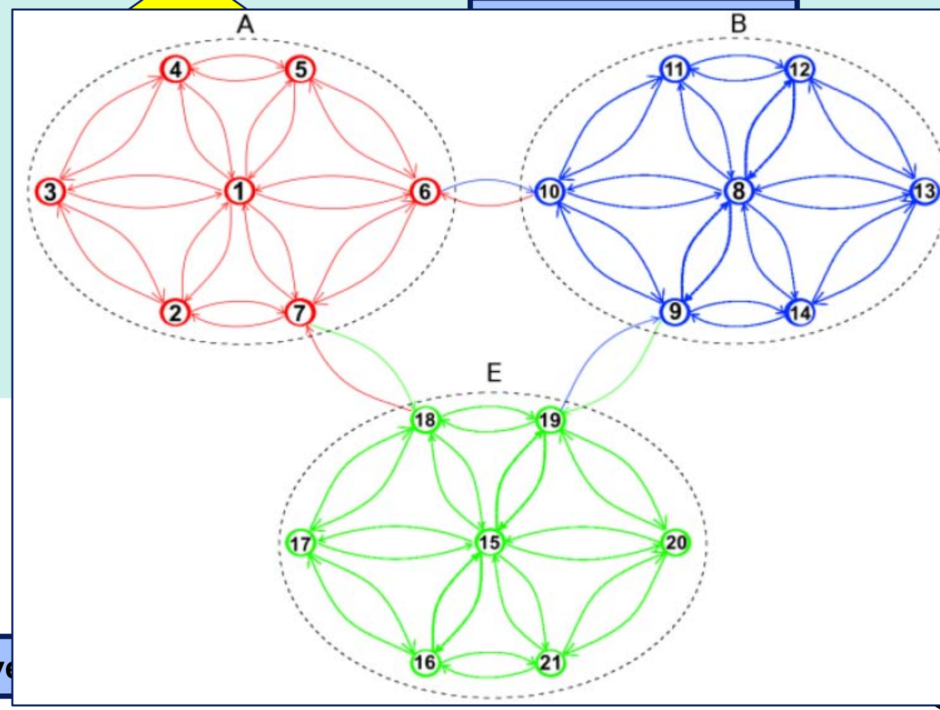
At destination?

NO

YES

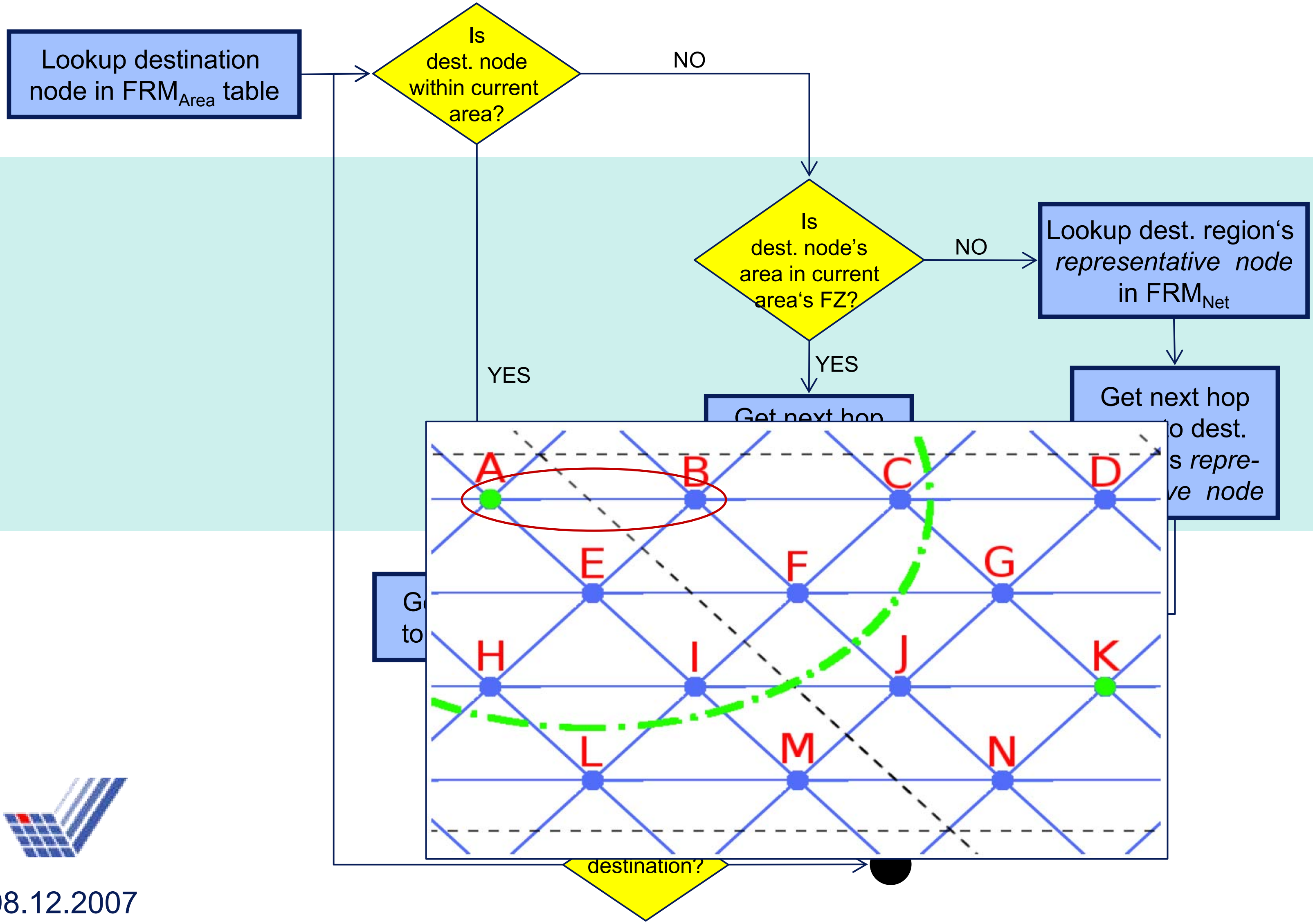
Net-Layer

Area-Layer



A→B

The *BeeJamA* Algorithm



Net-Layer

Area-Layer

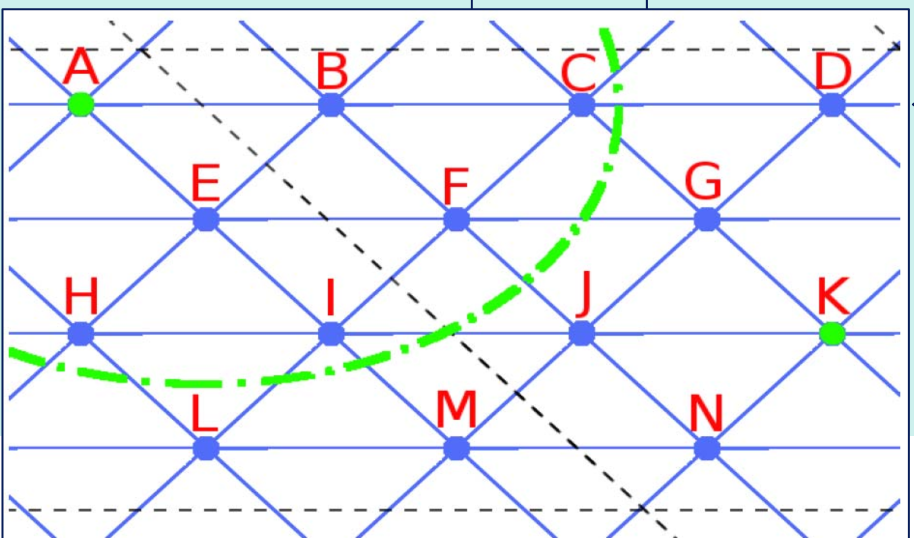
A→B

The *BeeJamA* Algorithm

Lookup destination node in FRM_{Area} table

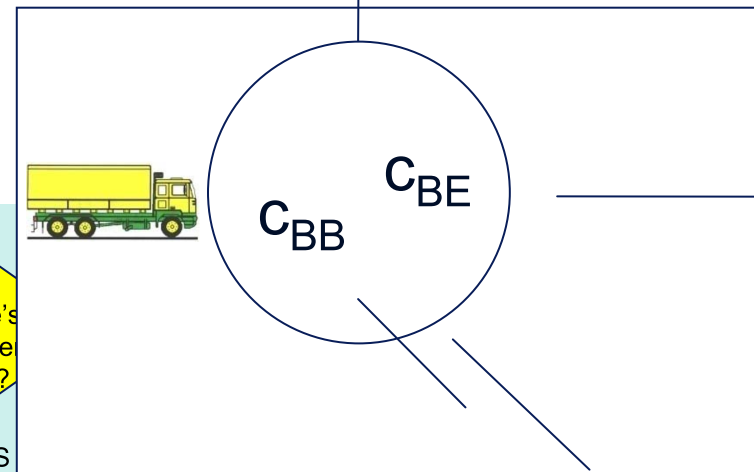
Is dest. node within current area?

NO



Is dest. node's area in current area's FZ?

YES



Get next hop area to dest. area

Get next hop area to dest. region's *repre-sentative node*

Get next hop to *border-node* to dest. area

to dest. node

IFZ_{Net} (A)	***B***	C	E	F	H	I
B	***c_{BB}***	c_{CB}	c_{EB}	c_{FB}	c_{HB}	c_{IB}
E	***c_{BE}***	c_{CE}	c_{EE}	c_{FE}	c_{HE}	c_{IE}

At destination?

NO

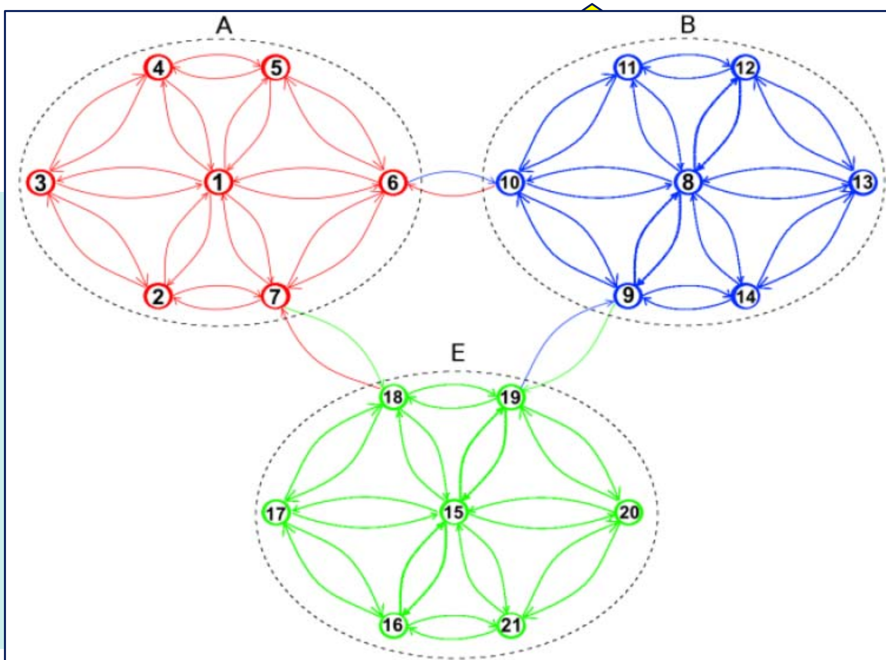
YES

ayer

Area-Layer

4→B (*via* 6)

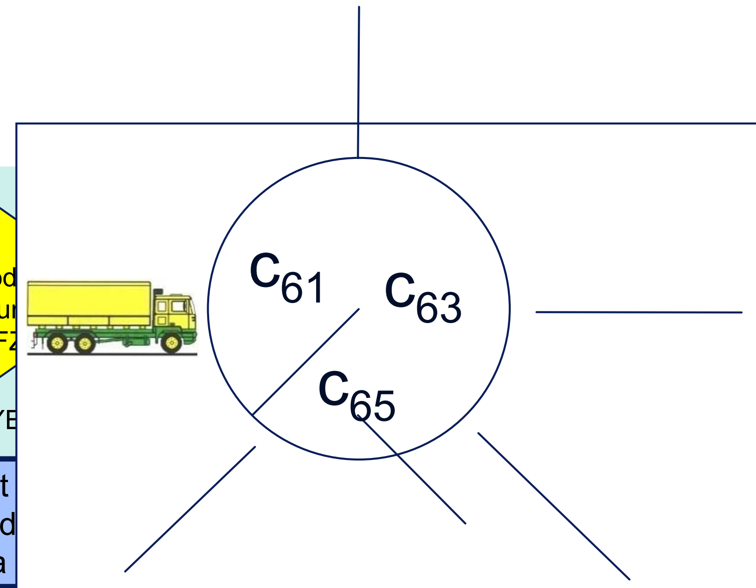
The *BeeJamA* Algorithm



Is dest. nod area in cur area's FZ

YE

Get next area to d area



Get next hop to dest. node

Get next hop to *border-node* to dest. area

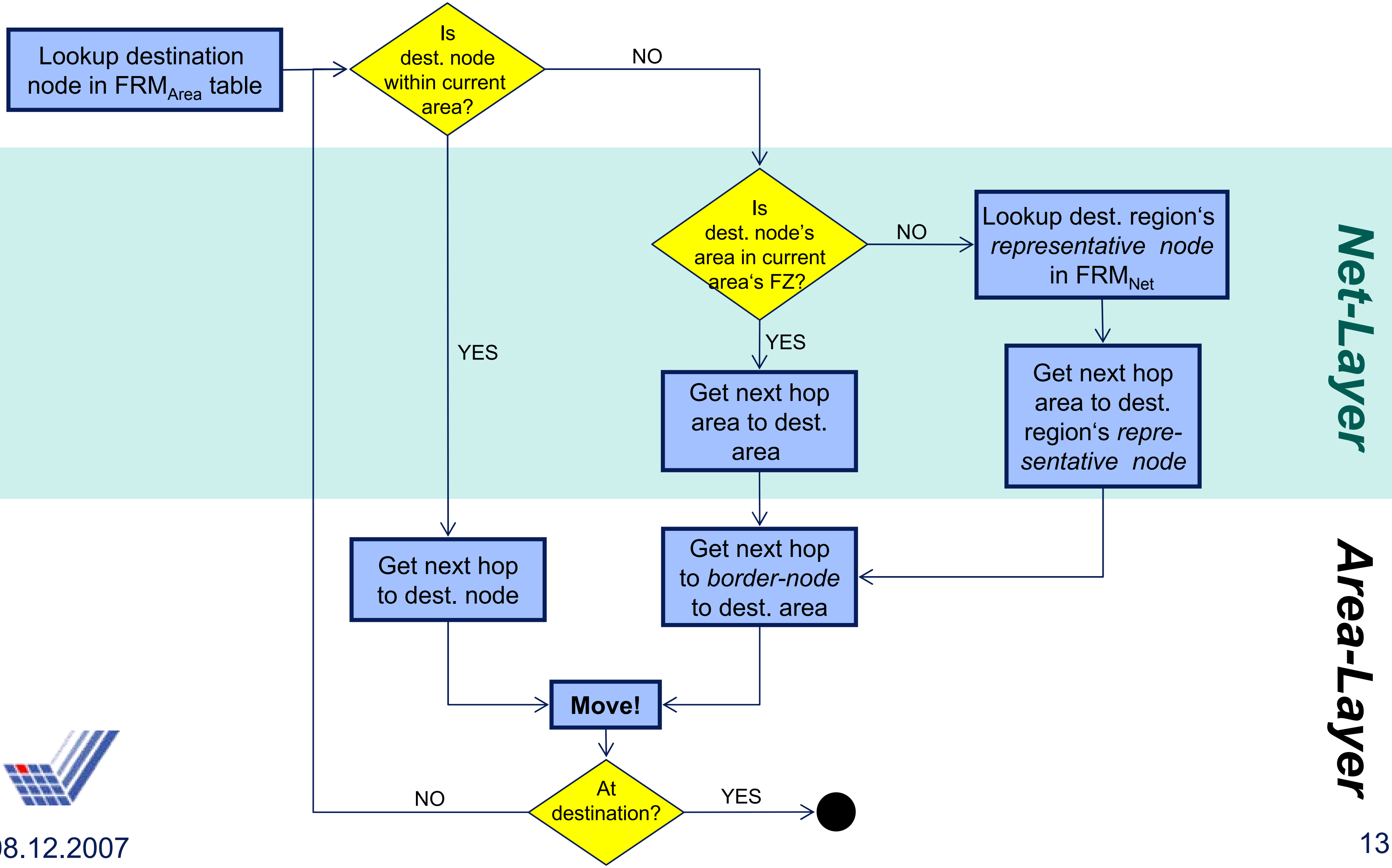
NO

IFZ_{Area}	1	2	3	4	5	***6***	7
1	c_{11}	c_{21}	c_{31}	c_{41}	c_{51}	$\boldsymbol{c_{61}}$	c_{71}
3	c_{13}	c_{23}	c_{33}	c_{43}	c_{53}	$\boldsymbol{c_{63}}$	c_{73}
5	c_{15}	c_{25}	c_{35}	c_{45}	c_{55}	$\boldsymbol{c_{65}}$	c_{75}

Area-Layer

1→14

The *BeeJamA* Algorithm



Simulation Studies

- Traffic Simulator
  - Realistic traffic model (Nagel/Schreckenberg)
  - Cellular automaton based
  - Commercially available navigational data *(AND)*
  - Section of the German Ruhr District

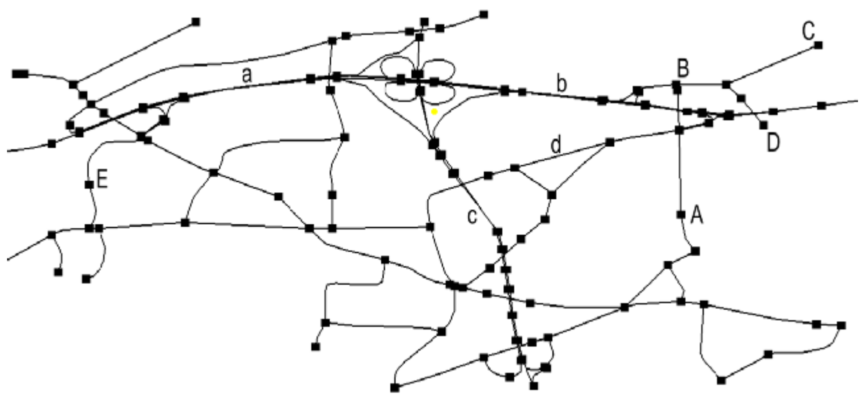


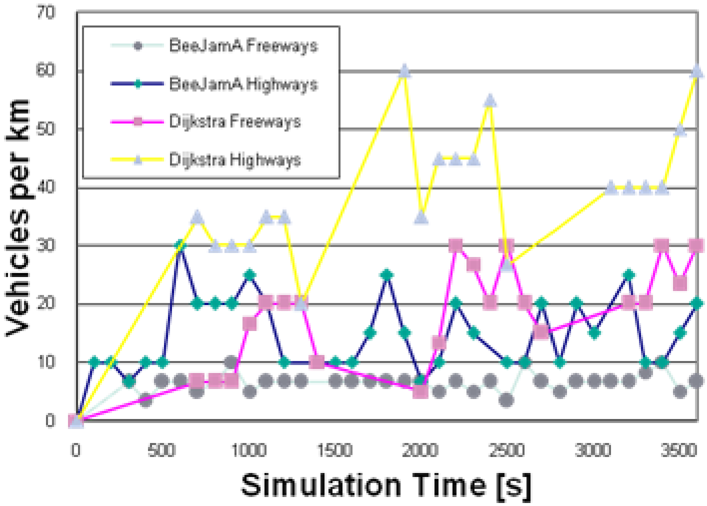
Figure 3: Realistic Section of the Ruhr Destrict

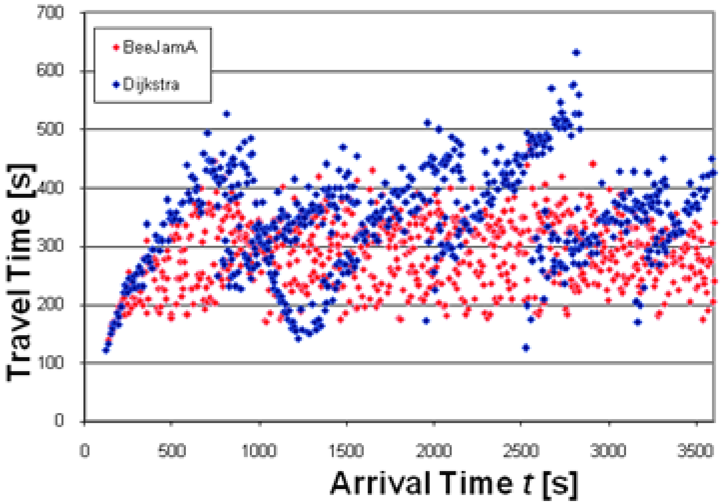
Source Nodes	A, B, C, D
Destination Node	E
New Vehicles per Second	4 (*1 per Node*)
Simulation Time	3600 seconds
Dijkstra Update Interval	600 seconds
Tempo Limits	135 km/h (Freeways), 85 km/h (Highways)
Max. Speed for Vehicles	135 km/h
Vehicular Density Limits	
Highways	α=35, β=40 [vehicles/km], A=50, B=10 [km/h]
4-Lane-Freeways	α=40, β=55 [vehicles/km], A=70, B=30 [km/h]
6-Lane-Freeways	α=45, β=65 [vehicles/km], A=75, B=35 [km/h]

Table 1: Experimental Setup

Experimental Results

- Traffic congestion avoidance with respect to individual travel times
  - BeeJamA routing against Dijkstra-based fastest path routing (10min update interval)





Conclusion

- We developed our own simulator, due to the lack of software for dynamic, distributed routing algorithms in vehicular traffic networks.
- Distributed layered traffic model based on *BeeHive/BeeAdHoc* routing.
- Dynamic cost model: Minimal travel times to destinations, congestion avoidance as objectives.
- Although there is no global information the navigators work very efficiently.
- *Demonstration available after the session*

Future Work

- Distributed on-line simulator
  - Realistic communication modeling
  - Dynamic rule sets
  - OpenStreetMap integration
  - Optimal off-line clustering of navigation areas
- Hardware implementation and evaluation
  - OpenMoko handsets
  - Deadline granularity 1 sec.

- Online version of the simulator:
  - *http://ls3-www.cs.uni-dortmund.de/en/projekte/bees/beejama.html*

Thank you!