The Ant Colony Optimization (ACO) Metaheuristic: a Swarm Intelligence Framework for Complex Optimization Tasks

Part IV: ACO for Routing in Networks

Gianni Di Caro

gianni@idsia.ch

IDSIA, USI/SUPSI, Lugano (CH)

Part IV:

ACO for routing in networks

Road map

- General structure and characteristics of ACO algorithms for routing problems in telecommunication networks
- *AntNet & AntNet-FA [Di Caro & Dorigo, 1998]: ACO for best-effort routing in wired IP networks*
  - Description of the algorithms and discussion of properties
  - Results of extensive simulation experiments and comparison with several state-of-the-art routing algorithms
- *AntHocNet [Di Caro, Ducatelle & Gambardella, 2004]: ACO for best-effort routing in wireless mobile ad hoc networks*
  - Characteristics and challenges of mobile ad hoc networks
  - Description of the algorithm and discussion of its properties
  - Results of extensive simulation experiments in realistic settings, comparison with state-of-the-art

ACO well matches network problems

- *Straightforward mapping:*
  - Decision points → Network nodes
  - Decision variables to learn (pheromone) → Next hops (outlinks)
  - Each node is like a single colony acting independently and socially
- The *intrisically distributed* nature of ACO (autonomous multiple agents, distributed decision policy, and stigmergic communications) can be fully exploited
- Networks are extremely dynamic environments calling for *adaptive learning and control systems*
- Ants repeatedly construct solutions, realizing a *parallel Monte Carlo simulation system* which can be “cheaply” implemented online in networks (but not in other real-world systems!)
- *Adaptivity, self-organization, and robustness,* are focal aspects in Traffic Engineering and Autonomic Communications

ACO-routing algorithms

- A number of routing algorithms have been designed after ACO and applied with success to several adaptive routing problems in wired and wireless networks

- Basic mechanisms in typical ACO-routing algorithms:
  - *Ant-like agents are proactively generated* at the nodes to find/check paths toward assigned destinations *[probing packets]*
  - Ants move hop-by-hop according to a *exploratory routing policy* based on the *local* routing information *[Monte Carlo path sampling]*
  - After reaching their destination, ants retrace their path and update nodes' routing information according to the *quality of the path*
  - Routing information = *statistical estimates* of the time-to-go to the destination maintained in *pheromone arrays [distance vector]*
  - *Data are probabilistically spread over the paths* according to their estimated quality as stored in the pheromone variables

Properties of ACO-routing algorithms

- *Nice properties:* good adaptivity, robust to failures, multiple path routing, automatic load balancing, good scalability

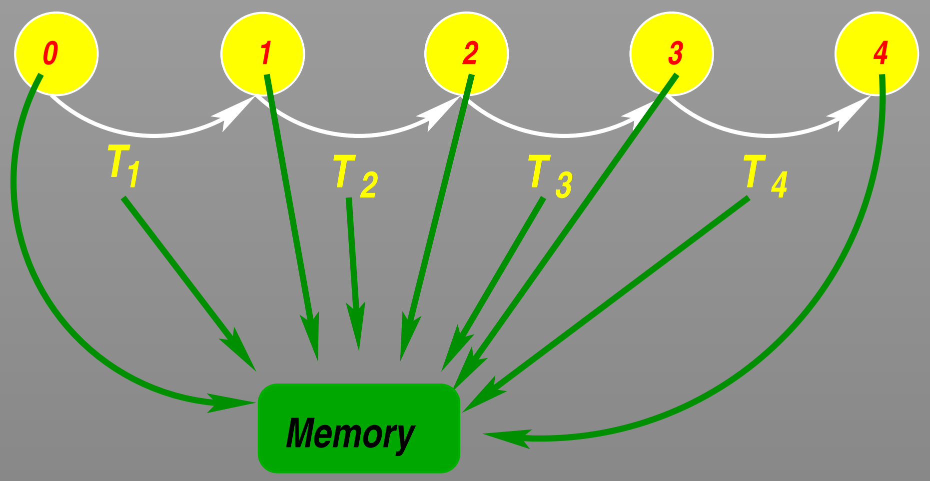
- *Bad properties:* to properly track all the changes in the network the frequency of proactive generation should be high and this might be a problem with scarce bandwidth

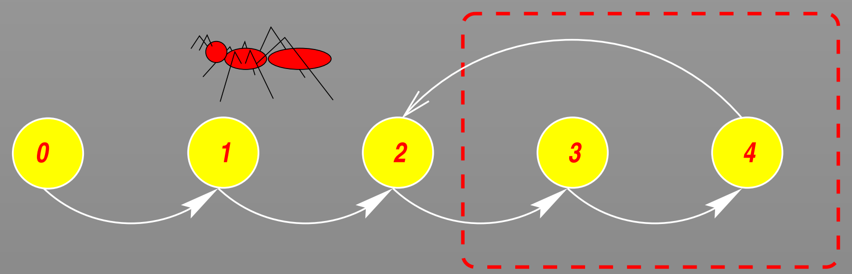
AntNet & AntNet-FA

- First ACO algorithms for *datagram networks* [Di Caro & Dorigo, 1997, 1998] (Schoonderwerd et al. applied ACO to telephone-like networks in 1996)
- General architecture: straightforward application of ACO
- *Careful design of each component*
- *State-of-the-art performance*
- *Reference algorithms* for a lot of subsequent algorithms
- AntNet-FA is an improvement of AntNet

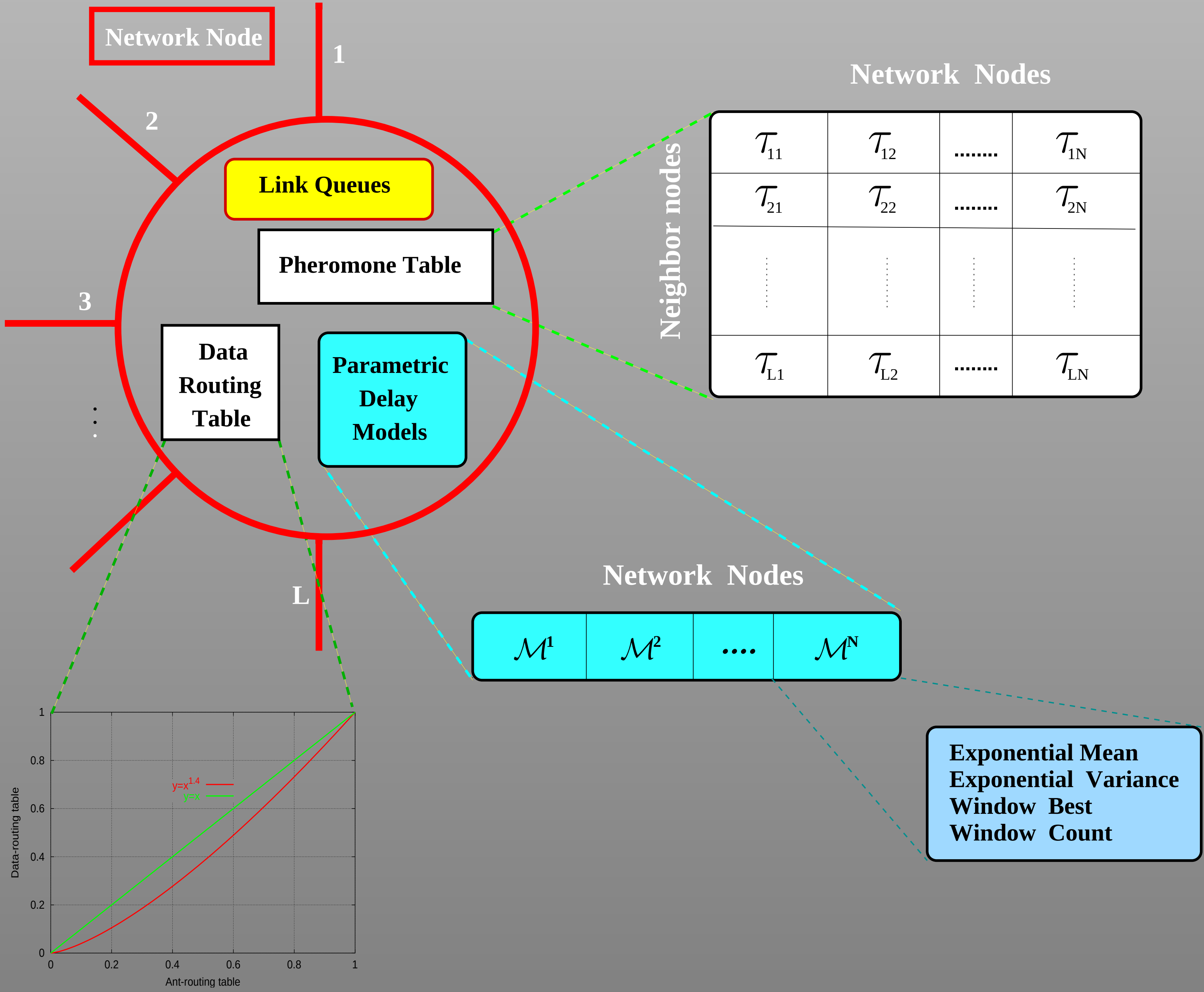
AntNet: algorithm description (1)

- *Proactive generation of Forward Ants*
- An ant faithfully *simulate a data packet*
  - *Discover/sample a good path* (agents explore. . .)
  - *Update routing information* (. . . data packets exploit)
- Forward ants maintain a private *memory* of each visited node and of the time of the visit (loops are removed)





AntNet: algorithm description (2)

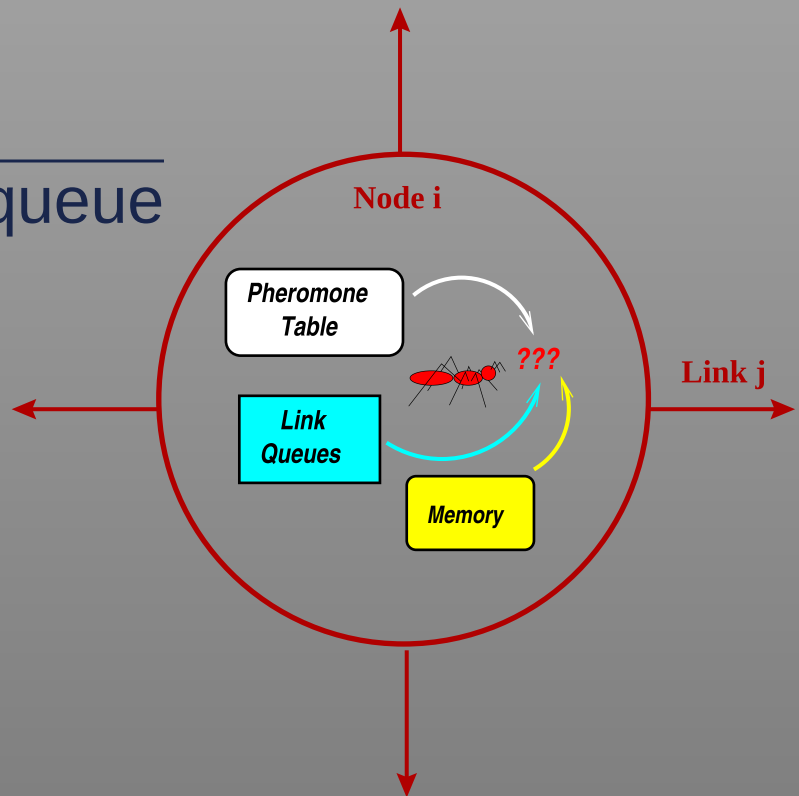


AntNet: algorithm description (3)

- *Next hop nodes* are selected according to a stochastic decision policy π_ϵ parametrized by:
  - Pheromone variables $\boldsymbol{\tau}_{ij}$ = *Measures of end-to-end delay*
  - Heuristic variables $\boldsymbol{\eta}_i$ = *Status of local link queues*
  - *Memory* of the nodes visited so far

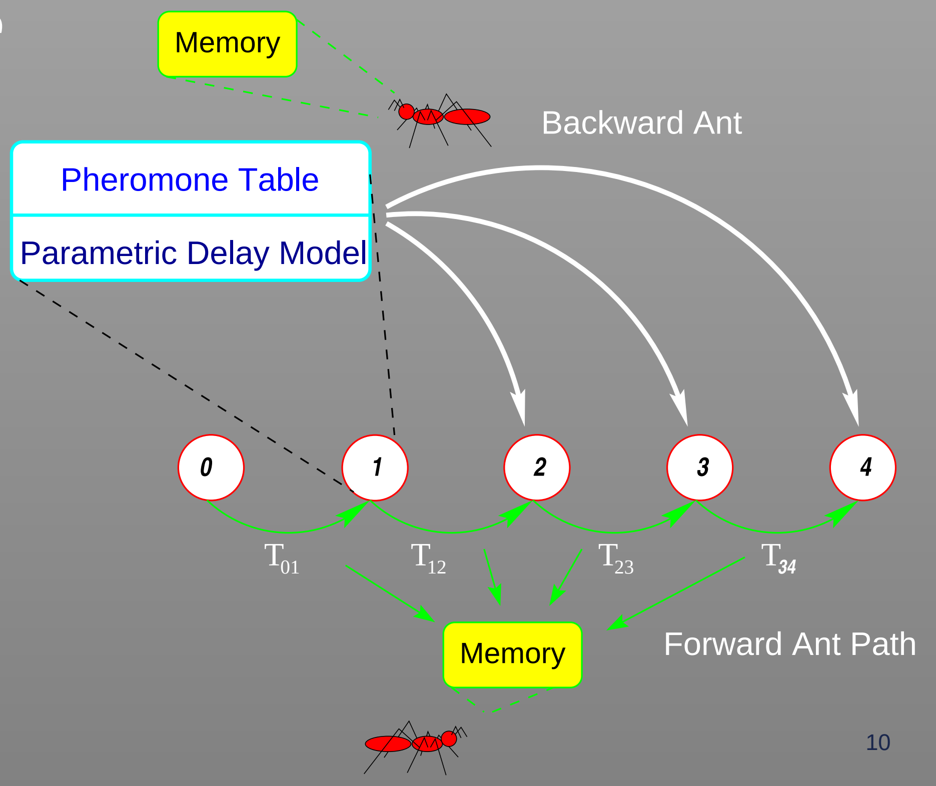
$$\eta_{ij}(t) \propto \frac{1}{\text{expected waiting time at } j\text{'s queue}}$$

$$\pi^d_\epsilon(i, j;\ t) \propto \alpha\tau^d_{ij}(t) + (1-\alpha)\eta_{ij}(t)$$



AntNet: algorithm description (4)

- Reached destination d, the Forward Ant turns into a *Backward Ant* and *retraces the path* back to the source node
- At each node i the Backward Ant, coming from neighbor j:
  - *Updates the Parametric Delay Model* $\mathcal{M}_i^d$
  - *Evaluates the path*: $r_{ij}^d = J(T_{i \to d}, \mathcal{M}_i^d)$
  - *Updates the pheromone table and the routing table with* r_{ij}^d
  - Also intermediate nodes
  - *Pure Monte Carlo updating*



Path evaluation and updating: formulas

- Non-stationarity and incomplete state information (aliases) make necessary the use of local adaptive models to provide an advisory evaluation:

$$r_{ij}^d = c_1 \left(\frac{W_{best}^d}{T_{i \to d}} \right) + c_2 F(T_{i \to d}, I_{sup}^d, I_{inf}^d), \qquad r_{ij}^d \in [0,1]$$

- The pheromone variables associated to the node j the Backward Ant comes from receives a *positive reinforcement*:

$$\tau_{ij}^d \leftarrow \tau_{ij}^d + r_{ij}^d \alpha (1 - \tau_{ij}^d)$$

- All the other possible next hops $n \in \mathcal{N}(i)$ receive, by probability normalization, a *negative reinforcement*:

$$\tau_{in}^d \leftarrow \tau_{in}^d - r \alpha \tau_{in}^d$$

IDSIA

AntNet-FA: improving AntNet

- In AntNet the path trip time $T_{i \to j}$ is the actual time experienced by the ant
- In AntNet-FA Forward ants make use of high priority queues (they fly!)
- The trip time $T_{i \to d}$ is calculated *during the backward journey, estimating the current waiting time at the link* j:

$$T_{i \to j} = d_j + \frac{q_j}{b_j}$$

- $T_{i \to j}$ is an *up-to-date* estimate of the time to hop from $i \to j$

What is the outcome of the ant actions?

- *Proactive exploration and route adaptation*
- At each node a *bundle* of datagram paths are available
- Each choice has a *goodness value* (pheromone) which is online adapted to the traffic patterns ↓
- *Data are spread stochastically (multi-path routing)*
- The less good paths are *backup paths*
- Automatic *load balancing*
- *Robust* wrt ant failures and to parameter setting
- No global propagation of local estimates
- Active non-local information gathering
- Shortcomings: TCP, short-lived loops, topological adaptivity

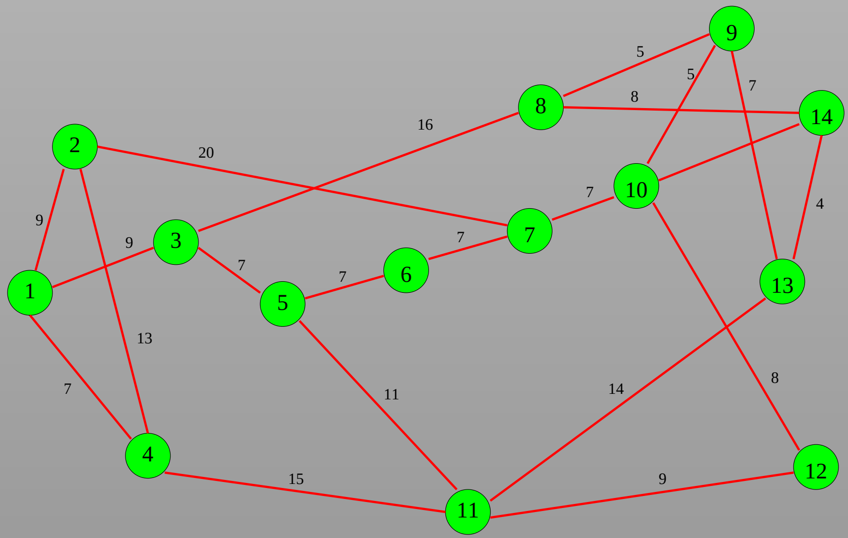
IDSIA

Experimental setup for AntNet(-FA)

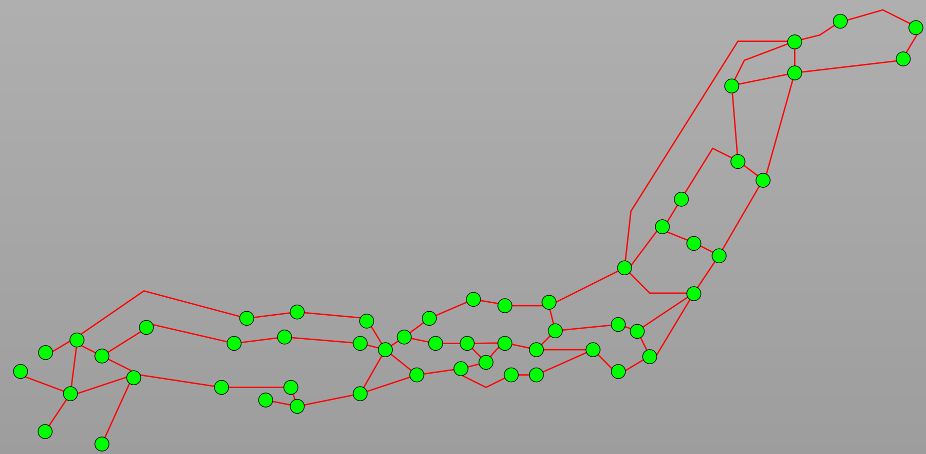
- Extensive simulation studies
- Realistic experimental setup for:
  - Network topology and physical characteristics
  - Protocol for data transmission
  - Spatial and Temporal Traffic Patterns
  - Algorithms to compare the performances

Networks

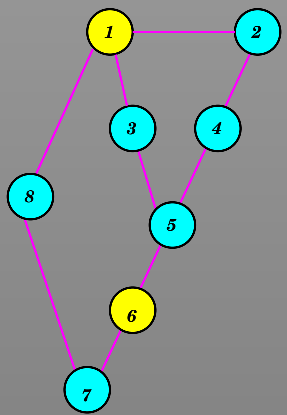
NSFNET-T1 (14, 21, 1.5) - US backbone (1987)



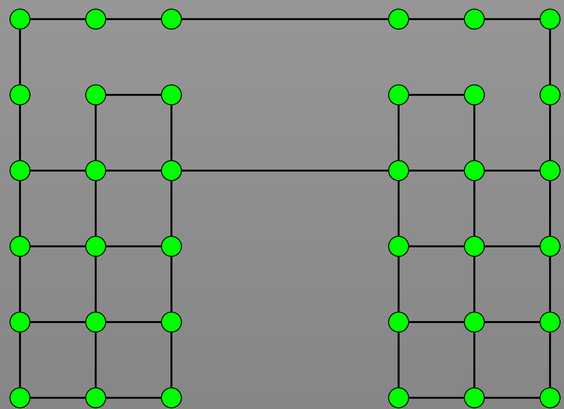
NTTnet (57, 162, 6) - NTT network (?)



Simple Net (8, 9, 10)



6x6 Net (36, 99, 10) - (Boyan & Littman, 1994)



Random Nets (100/150, ≈ 3, 10)

Traffic patterns

Data Transmission Protocol

- Best-effort Datagram traffic
- IP-like protocol
- Discarding packet for no buffer space
- Failure situations not considered
- No arrival acknowledgment or error notification packets
- Simple Flow control mechanism based on a static production window

Data sessions

- Negative exp distribution for sessions' inter-arrival times, global size, and packet sizes
- Traffic patterns obtained by the combination of three basic traffic types:
  - Poisson (Spatially Uniform (UP) and Random (RP))
  - Constant Bit Rate (CBR)
  - Hot Spots (HS)

Algorithms used for comparison

Static - Link-state

- *OSPF*: Minimum cost paths, current IGP Internet algorithm

Adaptive - Link-state

- *SPF*: Link-state prototype, Adaptive link costs, last ARPANET algorithm

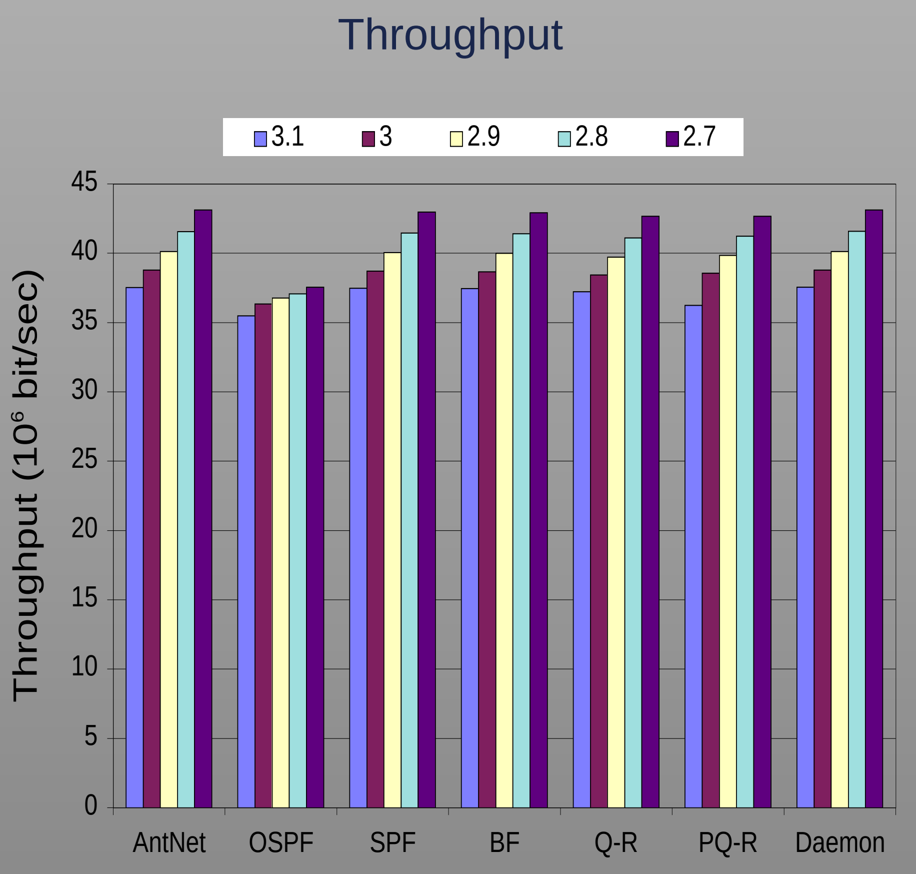
Adaptive - Distance-Vector

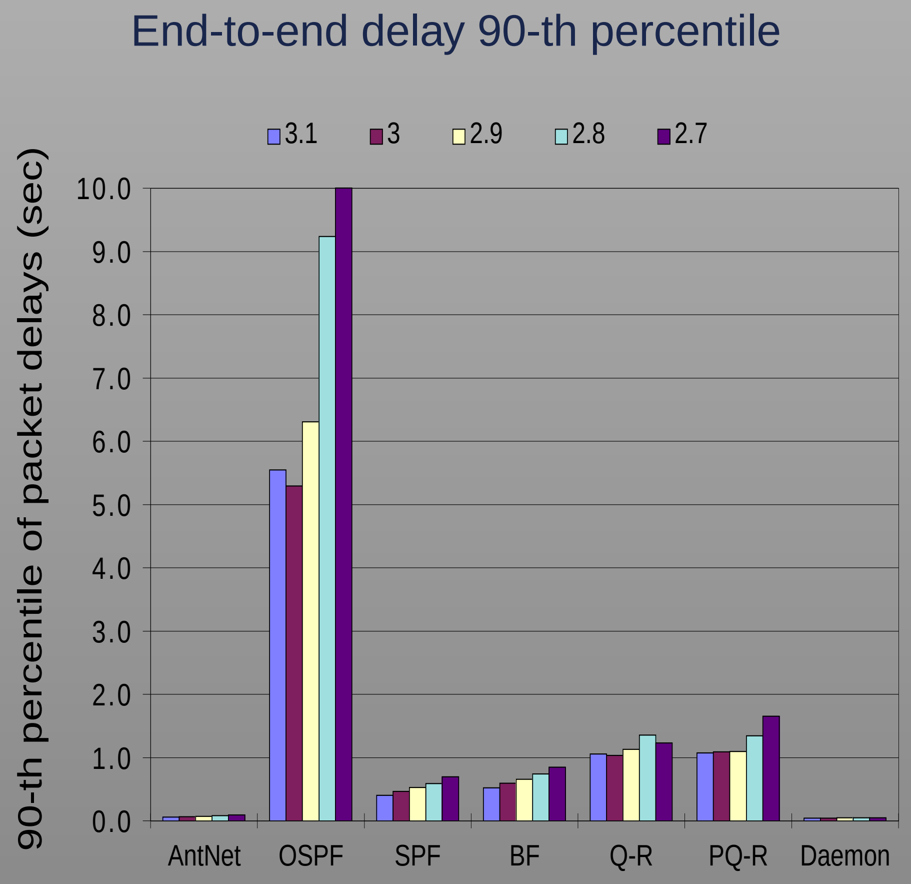
- *BF*: Asynchronous Bellman-Ford prototype, Adaptive link costs, ARPANET
- *Q-Routing*: Asynchronous Bellman-Ford with online updates and Q-Learning-like rule
- *PQ-Routing*: Q-Routing with a system to learn a model of the link queues

Ideal

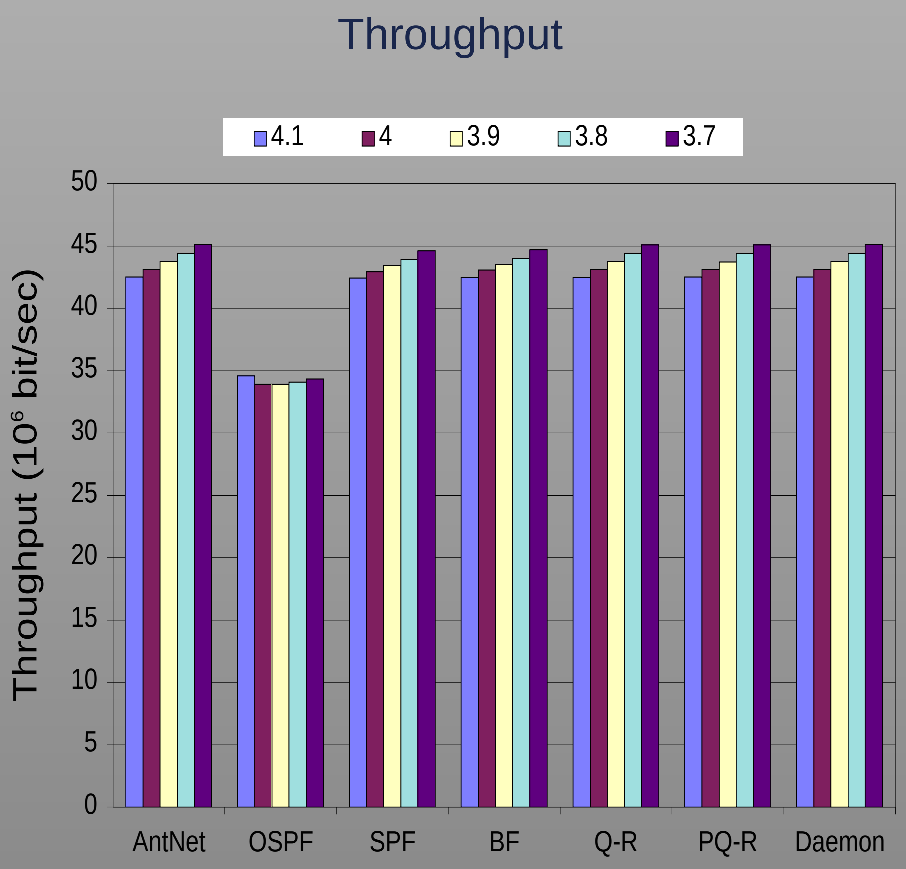
- *Daemon*: Access the state of all the net queues, empirical bound on performance

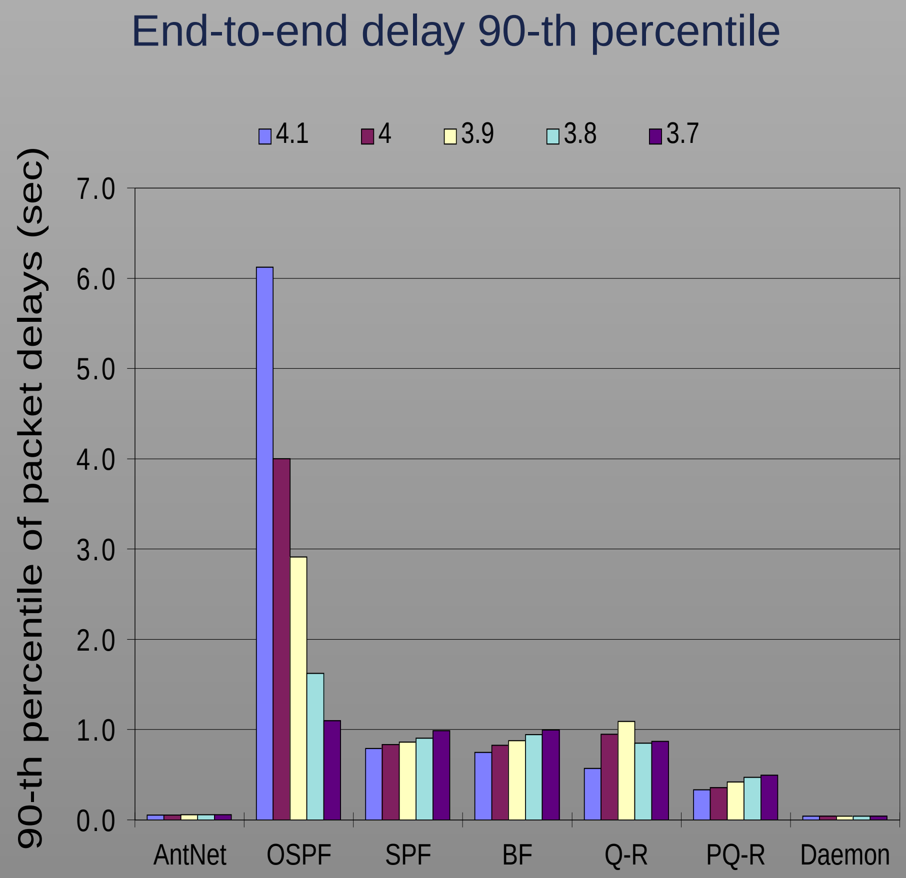
Results - NTTnet UP Load



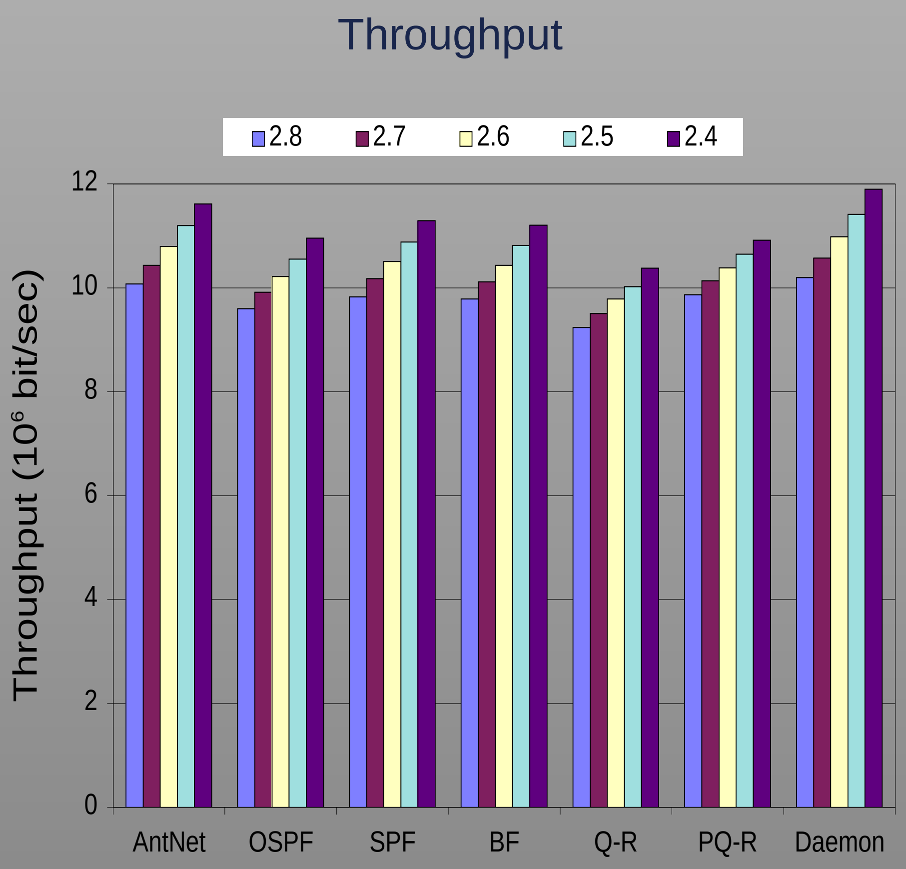


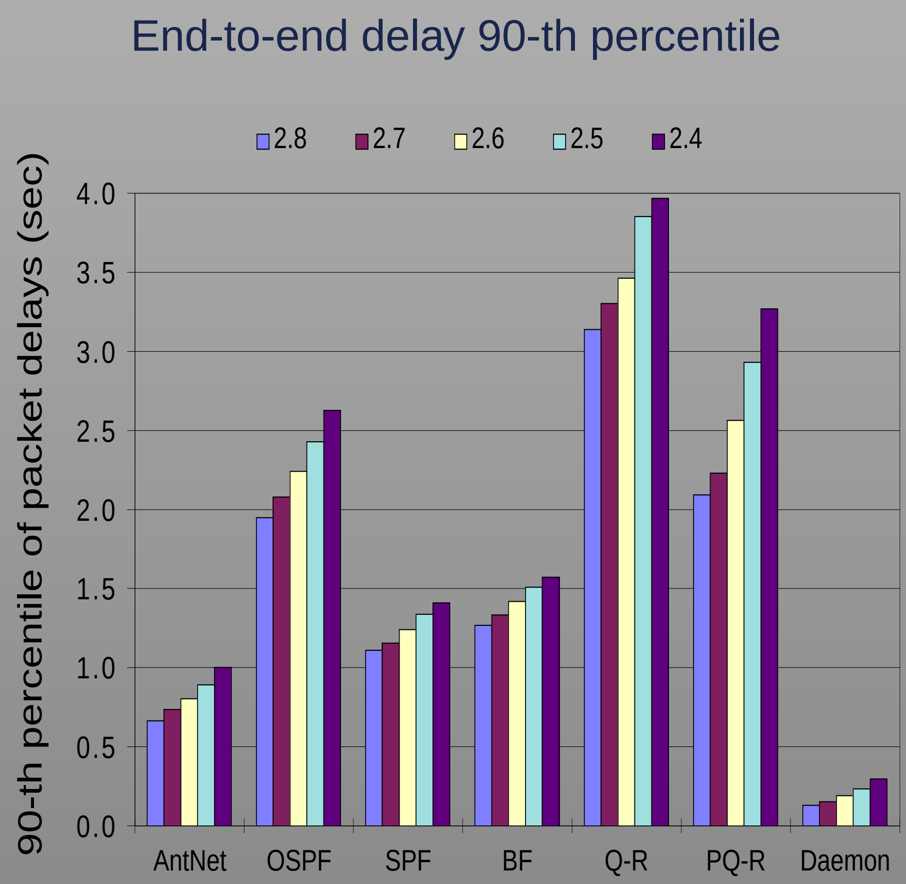
Results - NTTnet UPHS Load





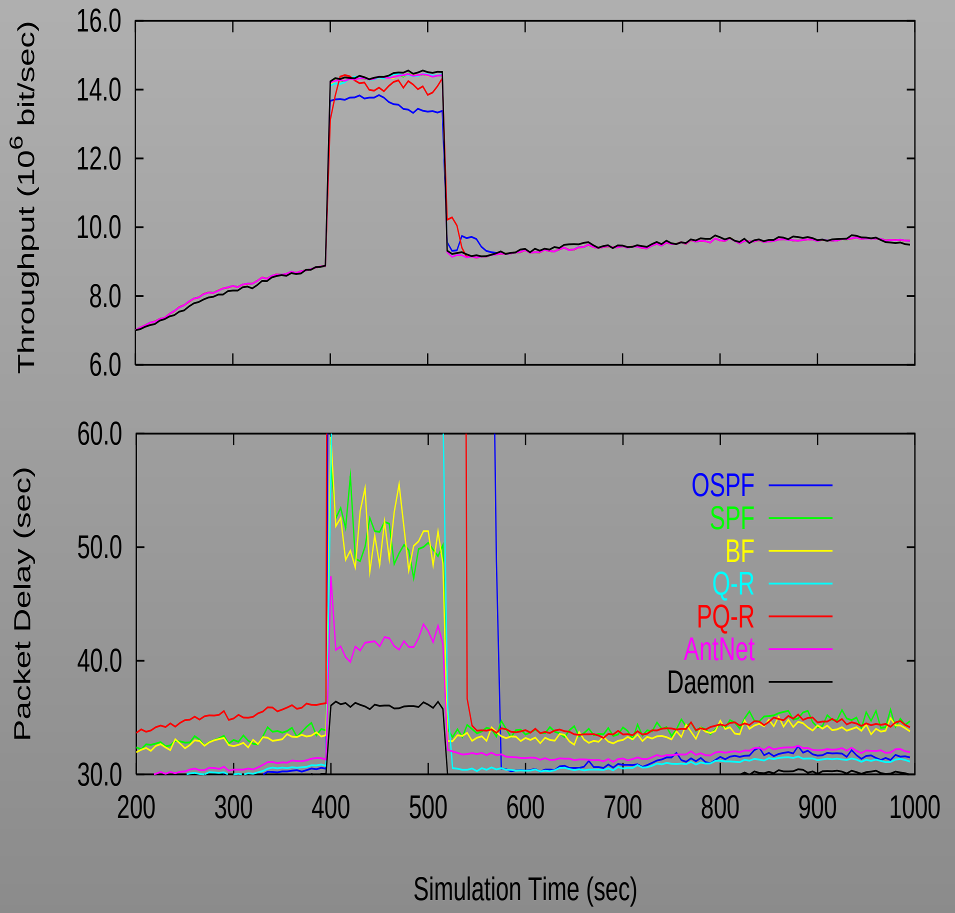
Results - NSFNET RP Load





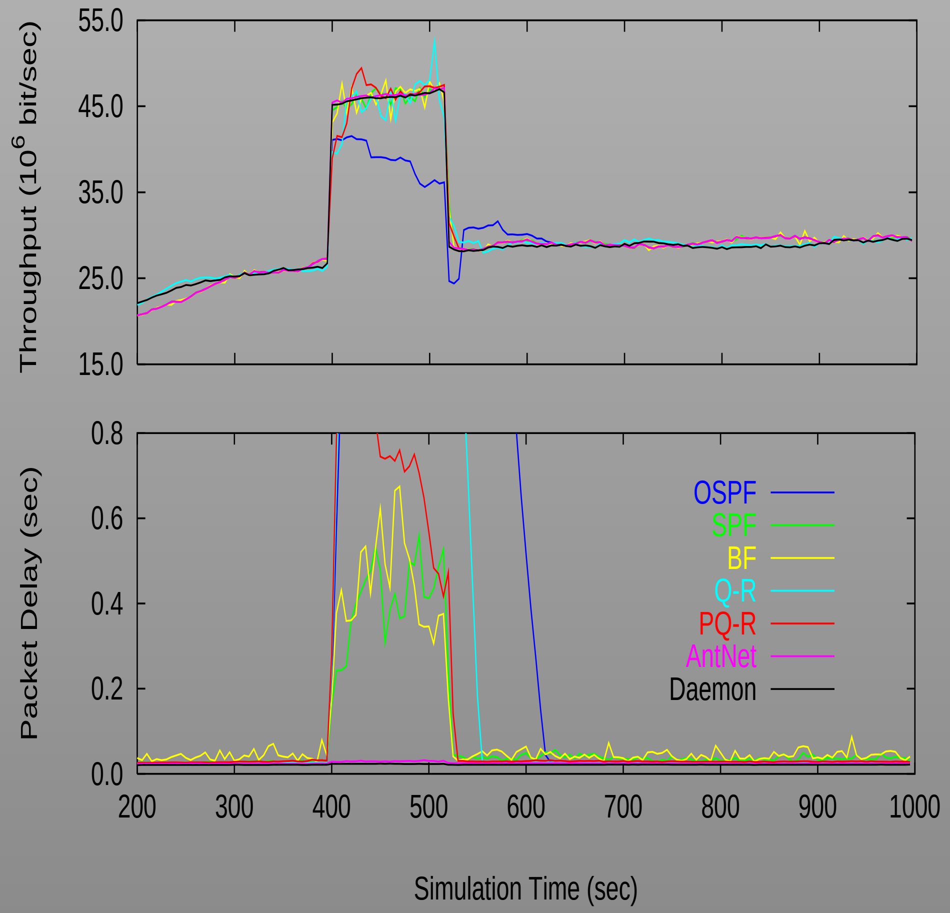
Results - Load Variation

NSFNET: UP Load Variation



MSIA=3.0, MPIA=0.3, HS=4, MPIA-HS=0.04

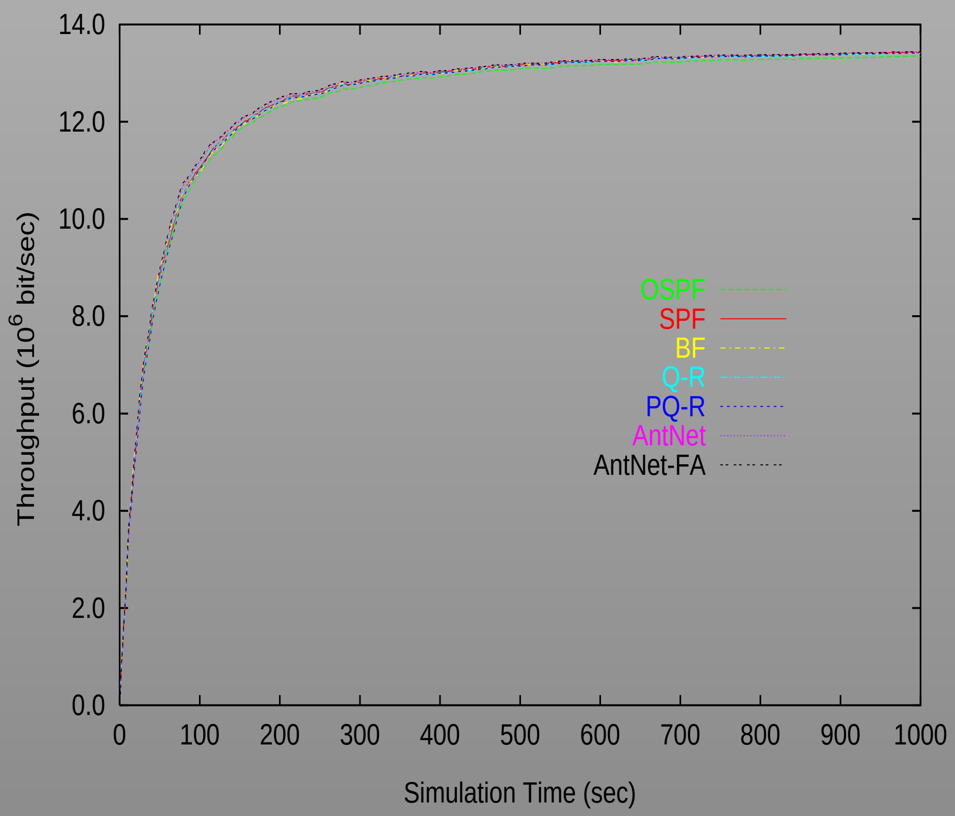
NTTnet: UP Load Variation



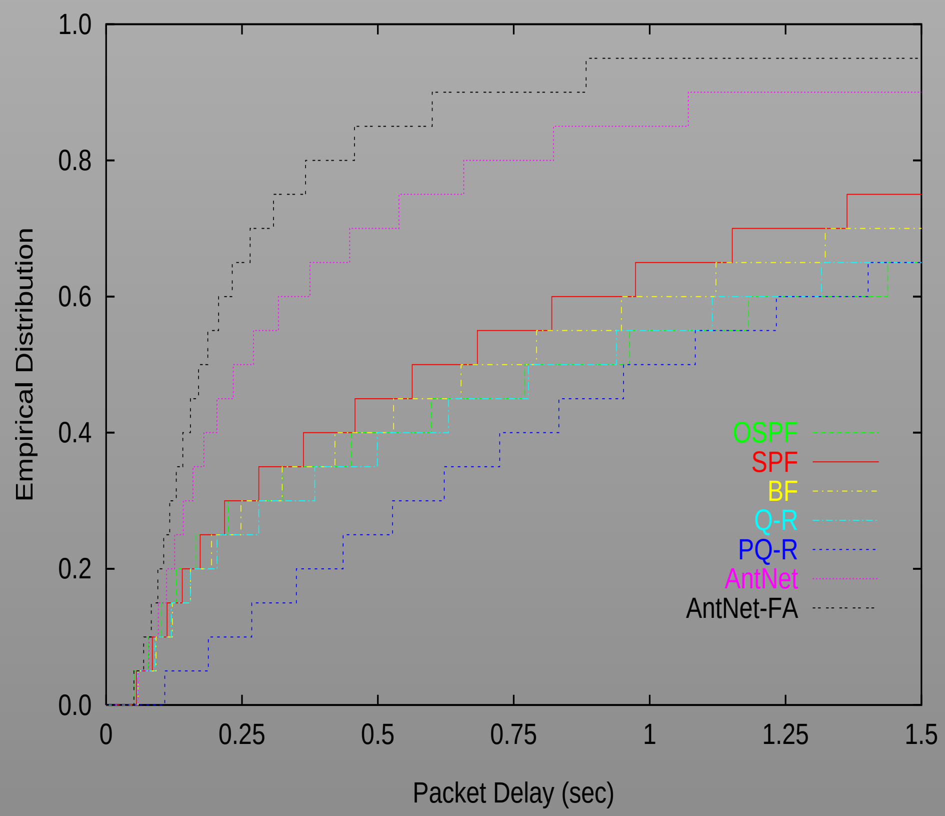
MSIA=4.0, MPIA=0.3, HS=4, MPIA-HS=0.05

100-Nodes RandomNets - UP Load

Throughput



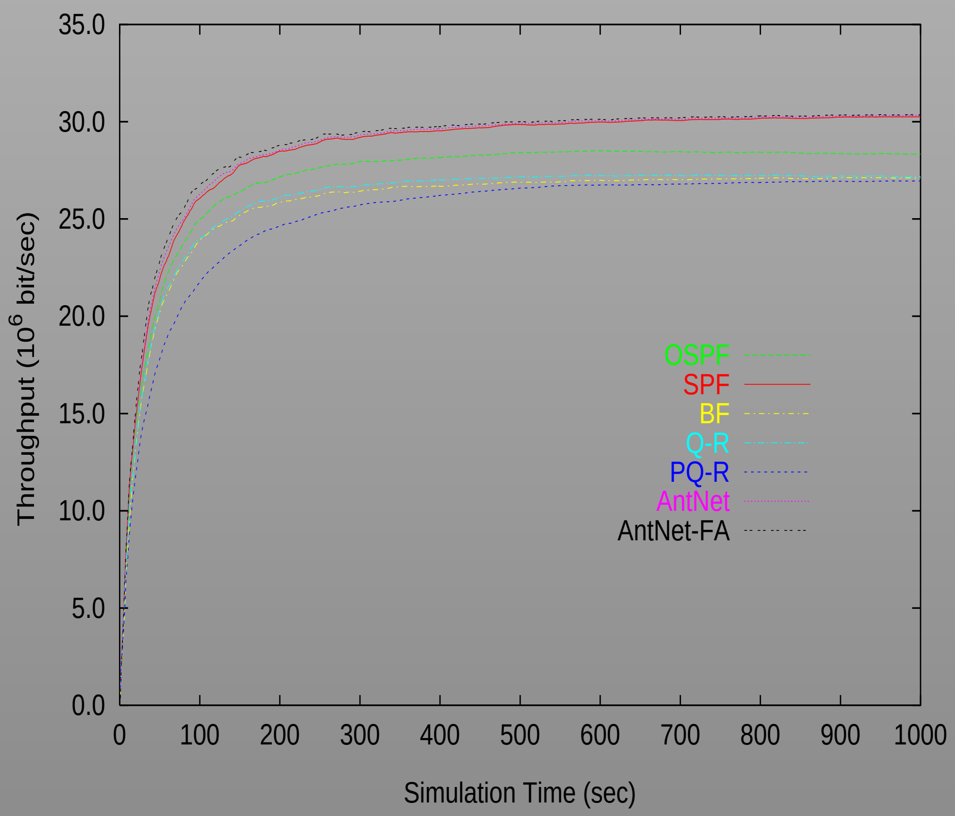
Delay distribution



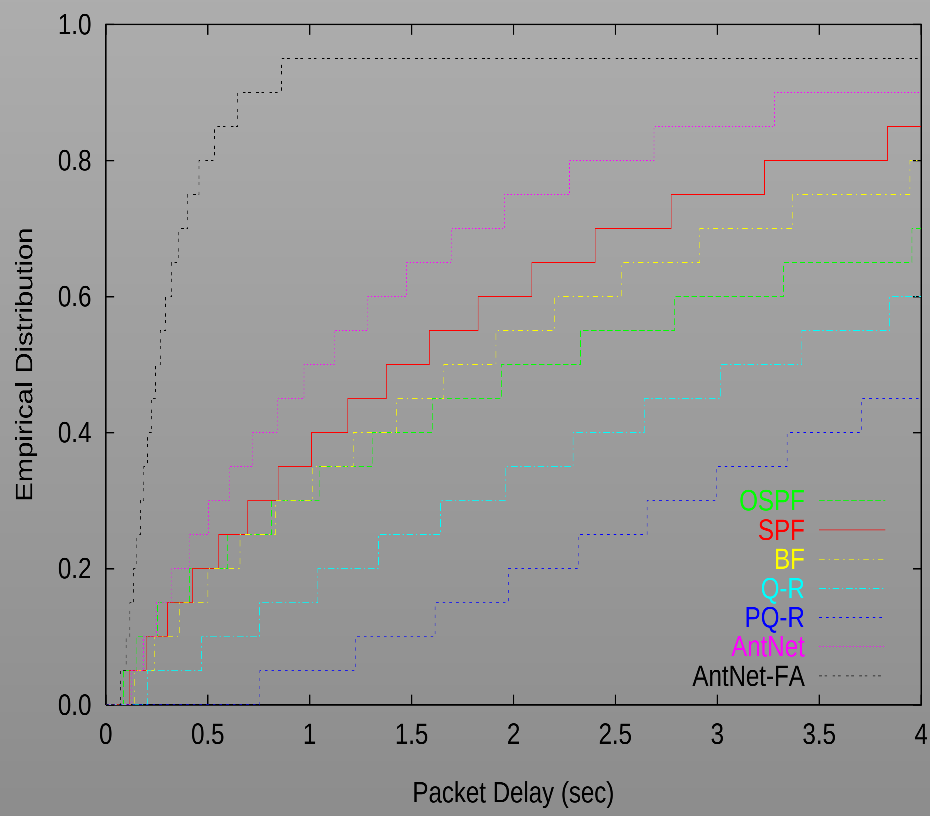
(MSIA = 15.0, MPIA = 0.005)

150-Nodes RandomNets - RP Load

Throughput



Delay distribution



(MSIA = 10.0, MPIA = 0.005)

Routing Overhead

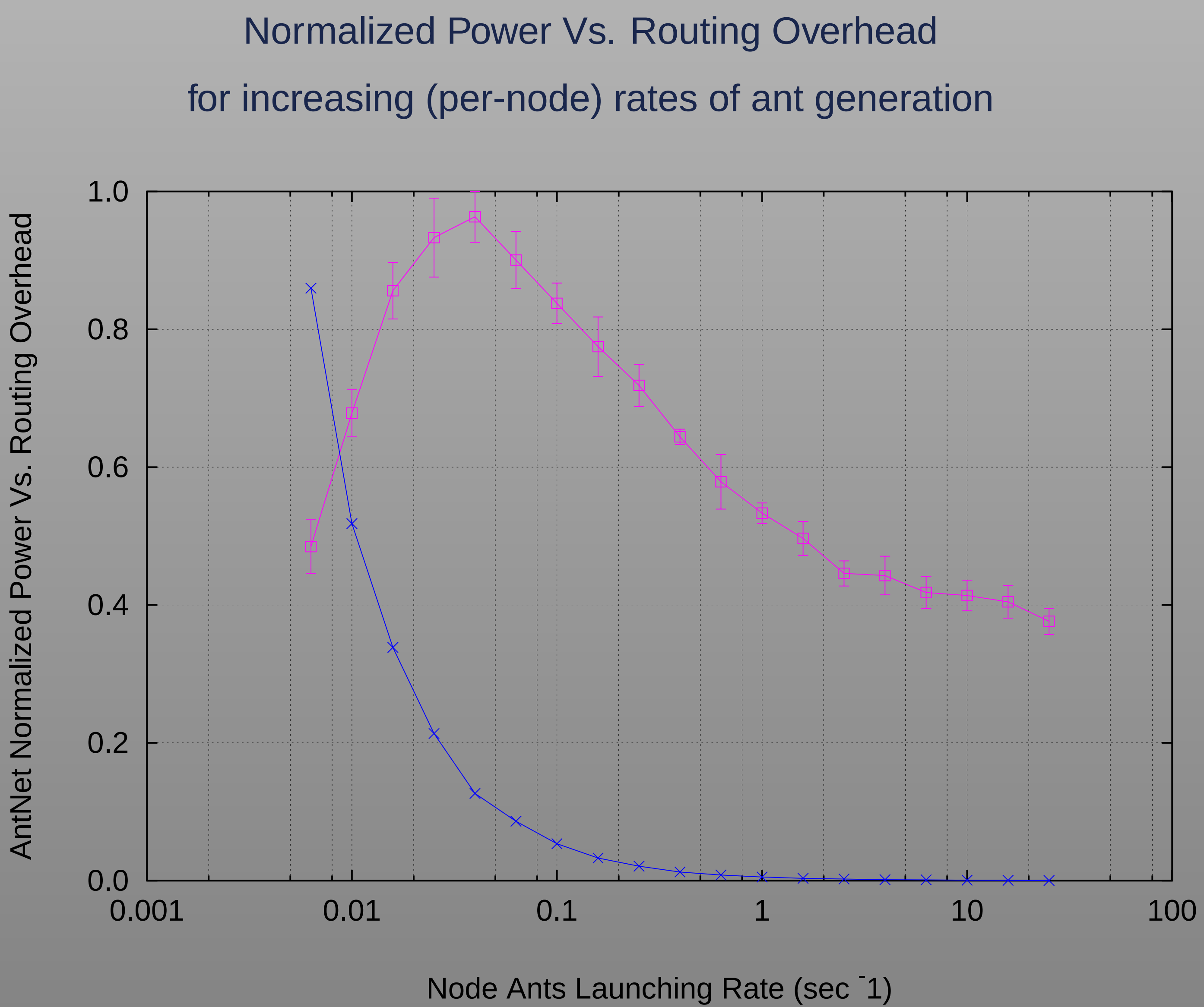
Routing Overhead (10^{-3}) for some of the realized experiments

	AntNet	OSPF	SPF	BF	Q-R	PQ-R	Daemon
NSF - UP	2.39	0.15	0.86	1.17	6.96	9.93	0.00
NSF - RP	2.60	0.16	1.07	1.17	5.26	7.74	0.00
NSF - UPHS	1.63	0.15	1.14	1.17	7.66	8.46	0.00
NTT - UP	2.85	0.14	3.68	1.39	3.72	6.77	0.00
NTT - UPHS	3.81	0.15	4.56	1.39	3.09	4.81	0.00

Routing Overhead = Ratio between the generated routing traffic and the total available bandwidth

For all the considered algorithms the routing overhead is quite low

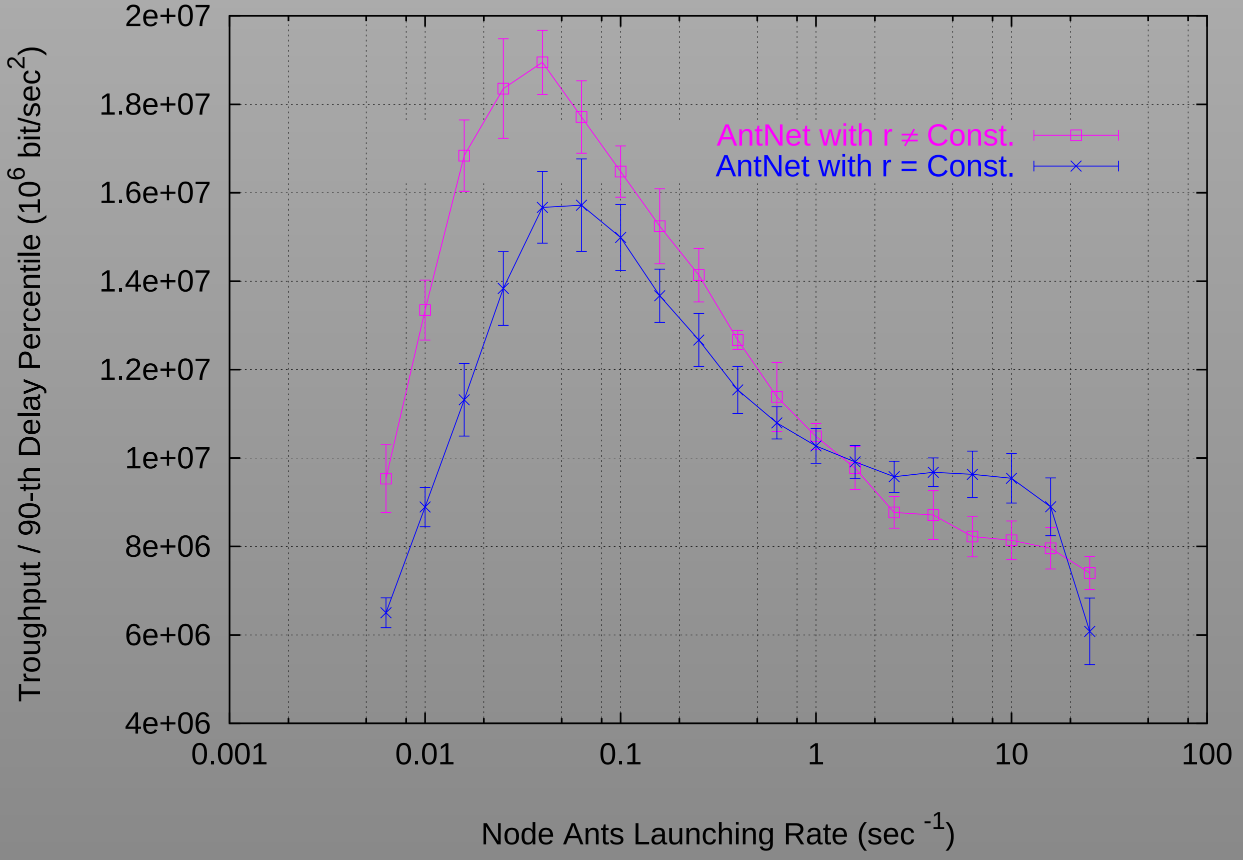
AntNet Power Vs. Routing Overhead



AntNet: Const. Vs. Non-Const. Reinforcements

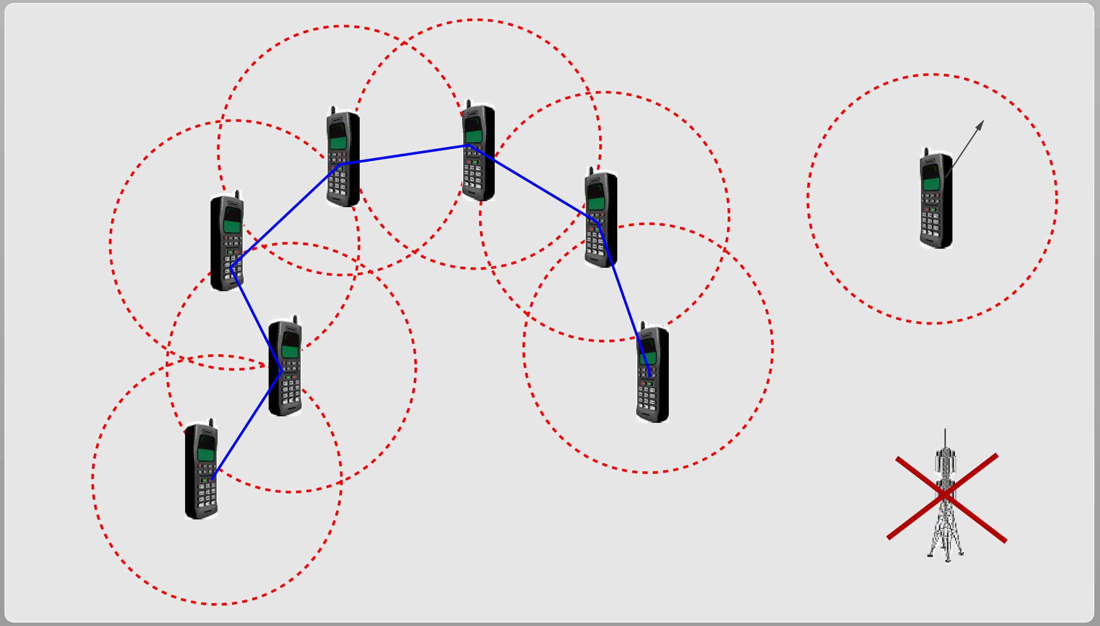
Constant Vs. Non-Constant reinforcements

for increasing (per-node) rates of ants production.



Non-constant reinforcements give improvements ranging from few percent to more than 40%

Mobile Ad Hoc Networks (MANETs): Definition



- No fixed infrastructure or centralized control
- Nodes can move, join, and leave the network at any time
- Nodes can communicate each other via wireless interfaces
- All nodes are equal and must serve as routers for each other
- Data packets are forwarded in a multi-hop fashion

MANETs: Applications and Challenges

- Bring connectivity in infrastructureless areas
- A fundamental building block towards flexible, pervasive, ubiquitous, and seamless networking

- *Constant topology and traffic variations*
- *Access control to the shared wireless medium reduces the effective available bandwidth (e.g., IEEE 802.11)*
- *Noise and transmission errors*

Routing in MANETs: Requirements

- Work in fully distributed and localized way (*Self-organizing*)
- *Adapt* to the continual topological and traffic variations
- *Robust* to errors and losses
- *Scalable* to number of nodes and traffic load (limited overhead)

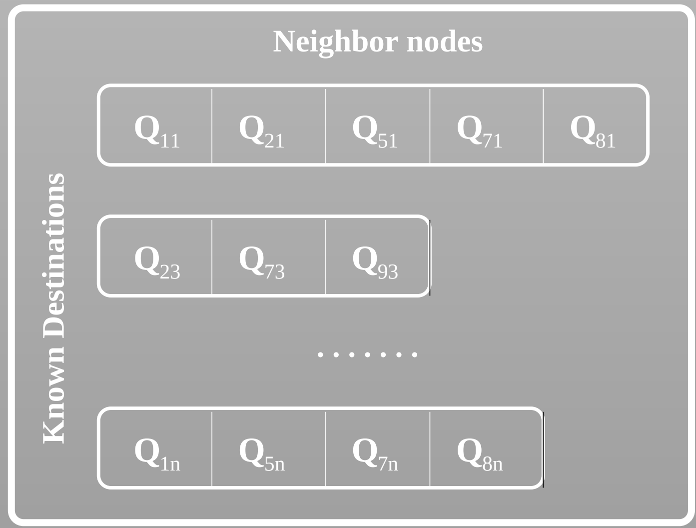
A solution based on *Swarm Intelligence* looks as a natural choice!

IDSIA

AntHocNet: Hybrid, Monte Carlo+Bootstrapping

- AntHocNet is a hybrid ACO-routing algorithm:
  - *Reactive:* paths are only setup at the starting of a session
  - *Proactive:* during the course of a session paths are *maintained* and *improved* proactively
- AntHocNet makes use of two learning mechanisms:
  - *Monte Carlo sampling and updating of full paths with ants*
  - *Local exchange and boostrapping of routing information*

AntHocNet: Routing (Pheromone) Tables



- Routing tables have an entry for each *known destination* and each *known neighbor*
- Each entry is a *pheromone variable* indicating the estimated *goodness/quality* Q_{nd} of a routing decision n for a destination d
- $Q_{nd} = F($*end-to-end delay, number of hops*$)$
- The task of ant agents is to *build/update* pheromone variables, which are used for routing according to a *stochastic policy*

AntHocNet: Reactive Path Setup

- *At the start of a session*, reactive forward ants are sent from the source to the destination
- They are *broadcast*, or *follow pheromone* whenever possible
- Ants are *filtered* at intermediate nodes to balance number of paths and overhead

AntHocNet: Stochastic Data Routing

- *Data packets are routed stochastically* according to the pheromone values: links with high pheromone have higher selection probability

- This way, data follow paths with *lower delay* and *lower number of hops*, avoiding areas of high congestion

- If pheromone values are kept up-do-date, this leads to *automatic load balancing*

AntHocNet: Dealing with Link Failures

1. *Discovered* via missed pheromone messages or failed unicast
2. Broadcast message to *notify the change* in the routing table
3. Try *local repair* if no alternative is available for data packet
4. Local repair ants travel to destination like reactive ants but with *limited broadcasting*
5. If local repair fails the packet is discarded and a notification is broadcast, otherwise the communication can keep going without the need to notify the source

Simulation Scenarios

All test scenarios are obtained by varying the parameters of the following *base scenario:*

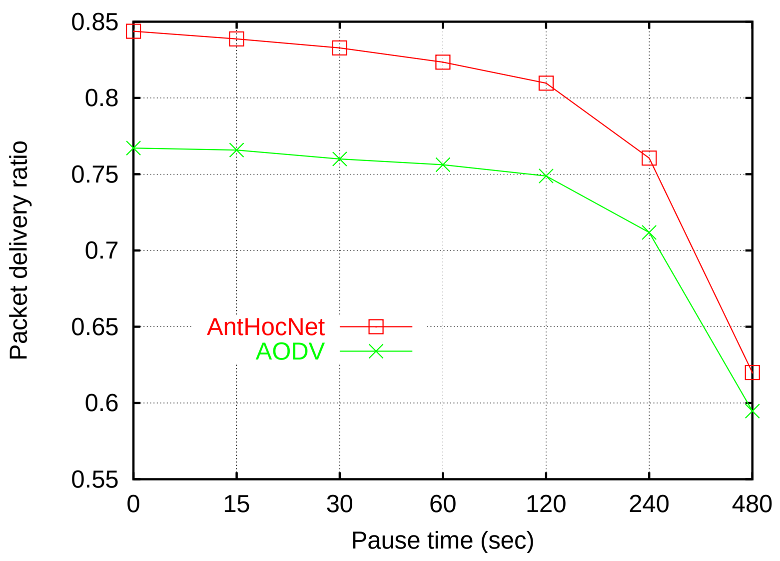
- *Area:* Flat, open space, rectangular of $3000 \times 1000\ m^2$
- *Mobility:* Random Waypoint (RWP), speed [0,20] m/s, pause time 30s
- *Data traffic:* starting from 20 Constant Bit Rate sources (CBR), 64 bytes/s, UDP at transport layer
- *Radio propagation:* two-ray and free space path loss, no fading
- *MAC:* single channel IEEE 802.11b DCF with 2 Mbps

Evaluation Methodology

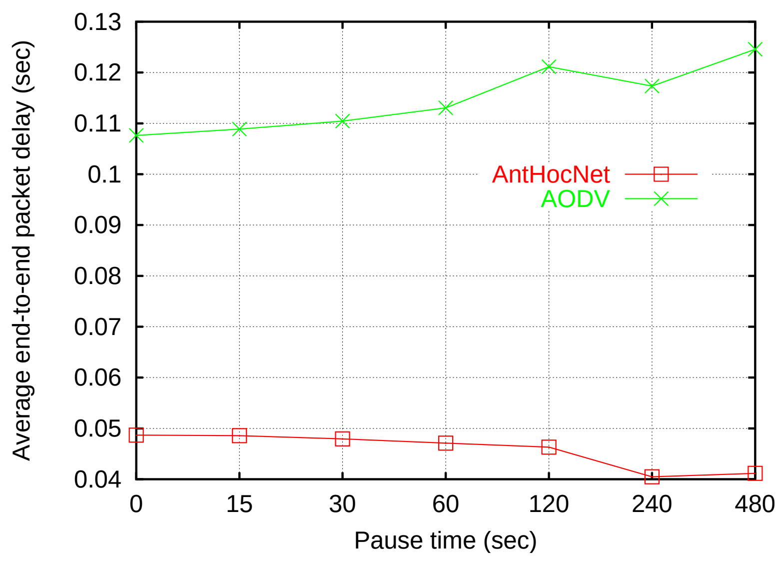
- *Measures of effectiveness:*
  - Delivery ratio
  - Average end-to-end delay
  - Average delay jitter
- *Measure of efficiency:*
  - Routing overhead: control packets vs. correctly delivered data packets
- *“Swarm” properties:*
  - Scalability: wrt to number of nodes and data sessions
  - Short- and Long-term Adaptivity:
- *Benchmark algorithm:*
  - AODV, a state-of-the art algorithm and common standard for comparison
- *Simulation environment:*
  - QualNet, a realistic packet-level commercial simulator

Mobility (Pause Time)

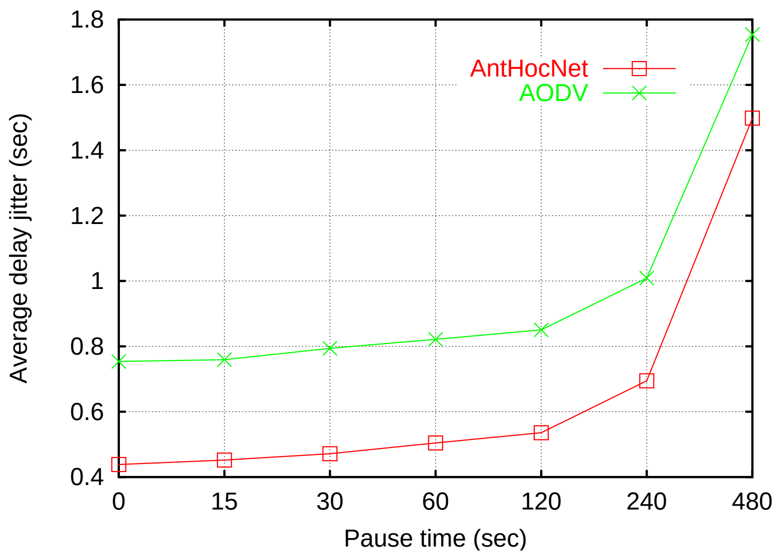
Delivery Ratio



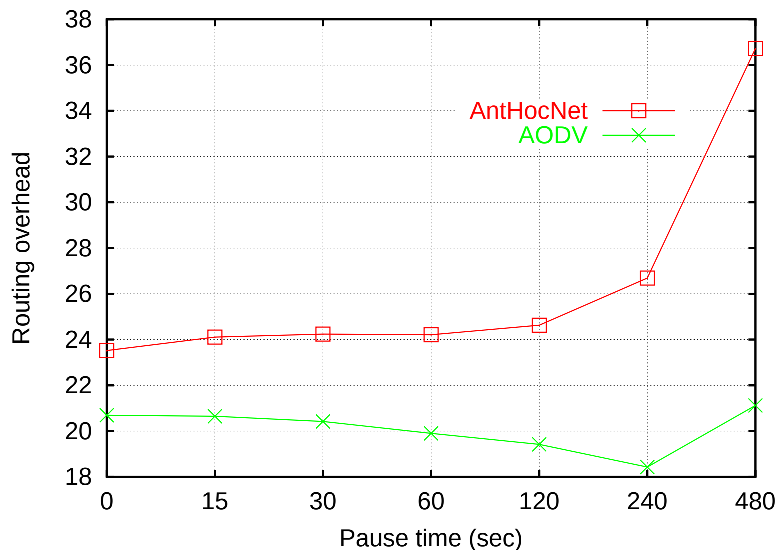
Average end–to–end delay



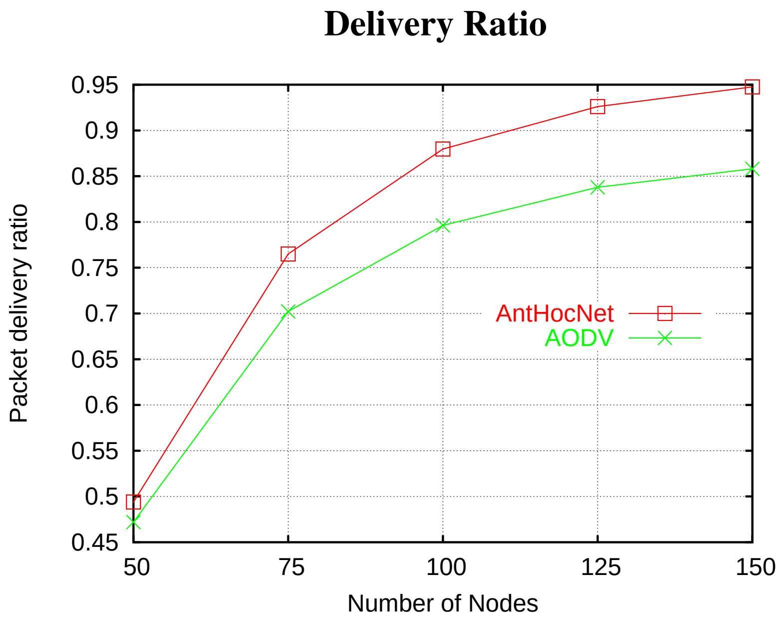
Average delay jitter

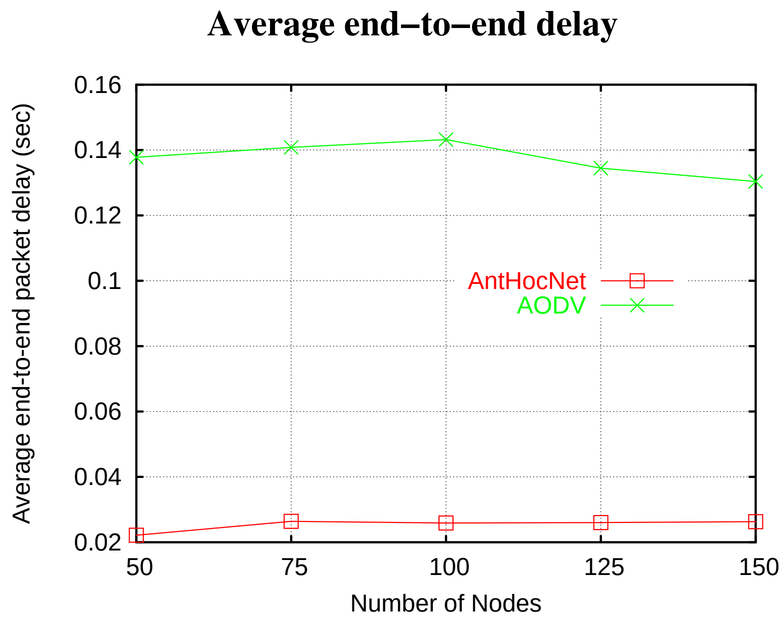


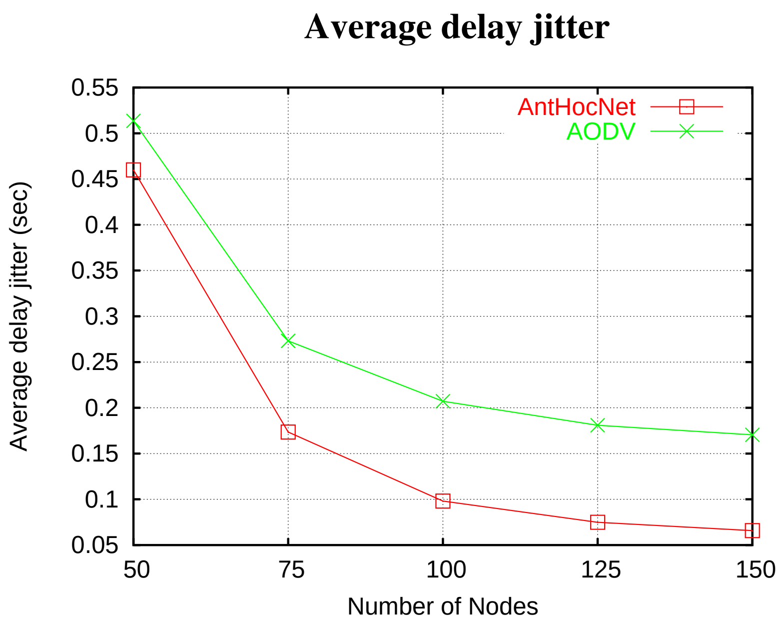
Routing overhead

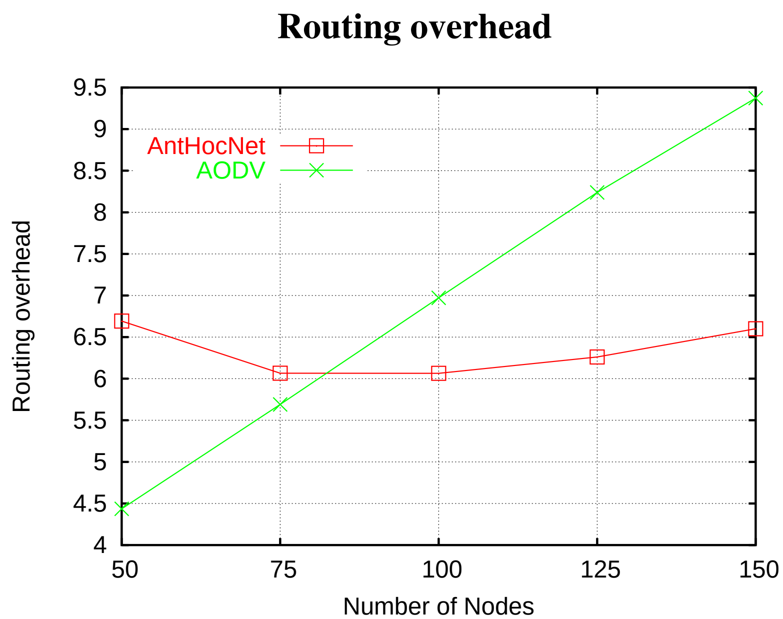


Node Density (Number of Nodes)



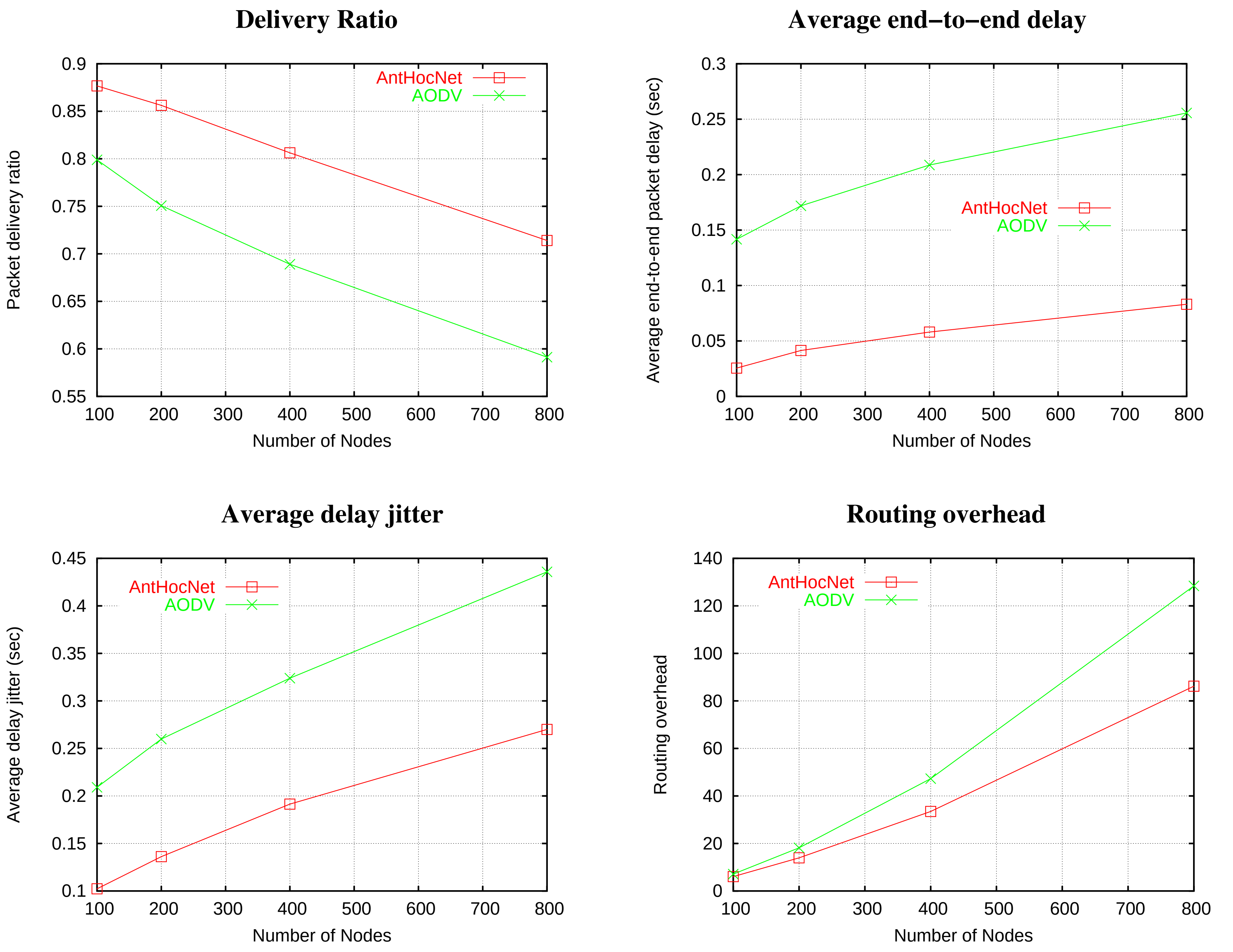




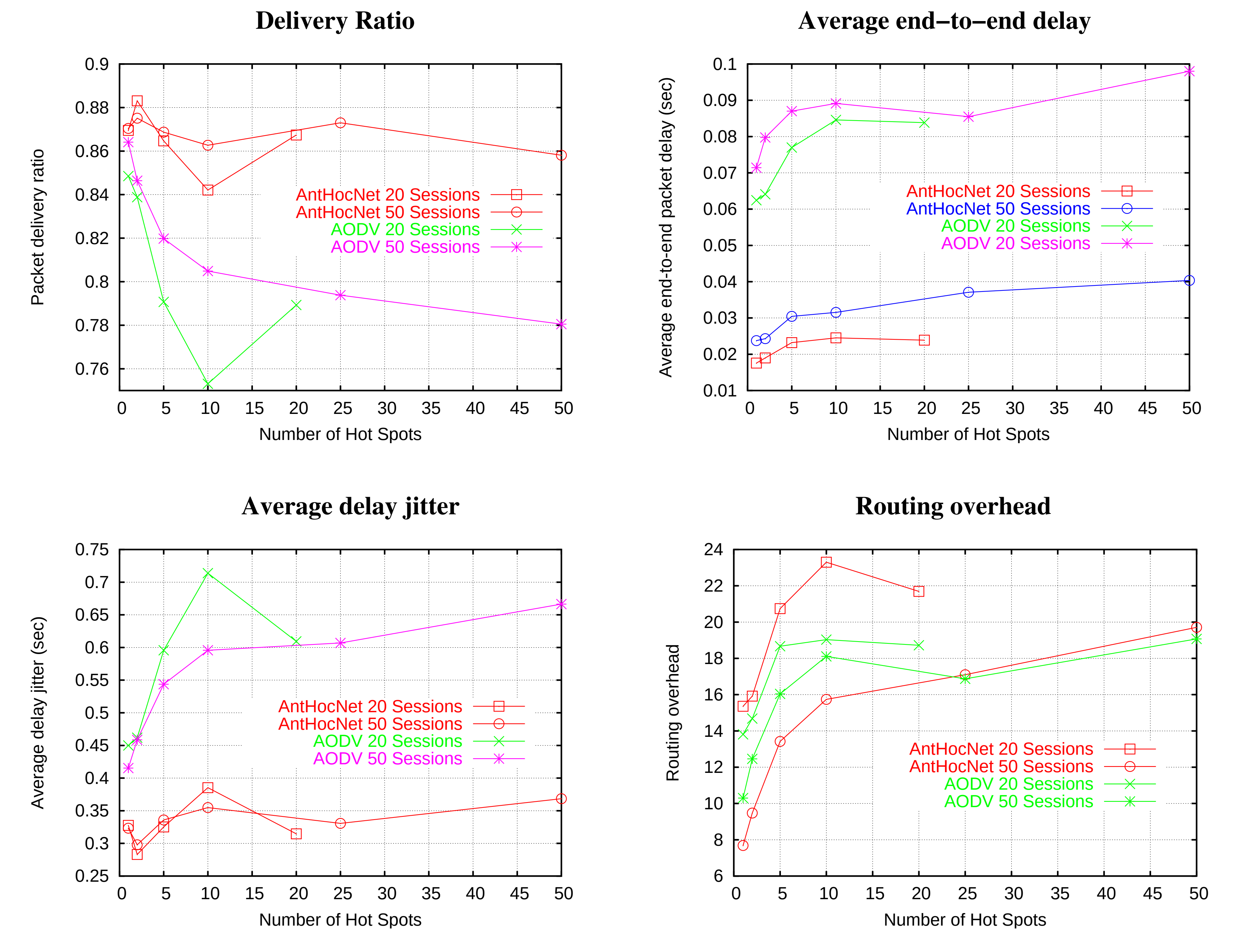


IDSIA

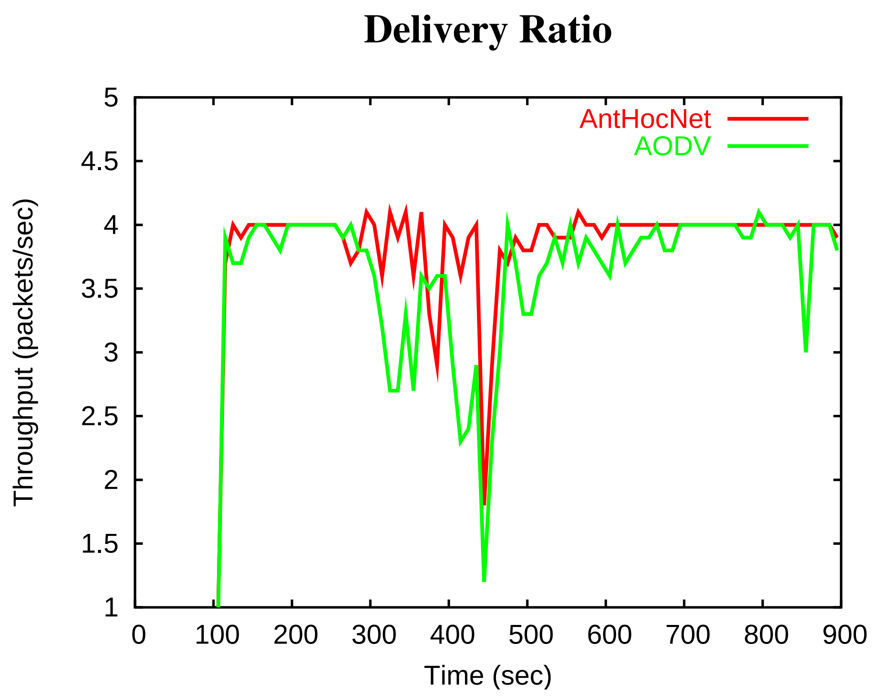
Scaling (Number of Nodes)

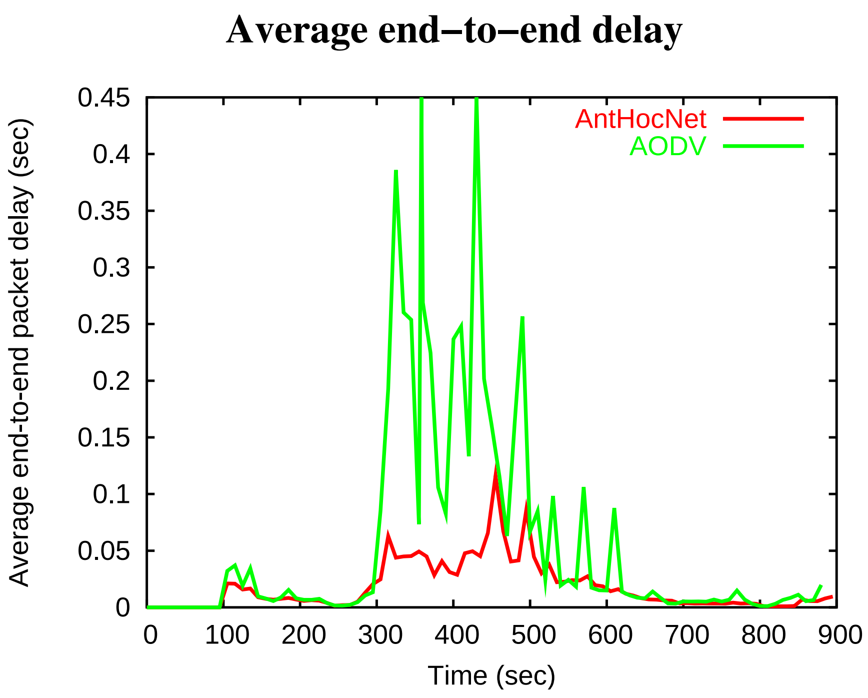


Scaling (Number of Hot Spots)



Short-term Adaptivity (Traffic Burst)





Basic references

- G. Di Caro and M. Dorigo, "AntNet: Distributed Stigmergetic Control for Communications Networks", Journal of Artificial Intelligence Research (JAIR), Vol. 9, pages 317-365, 1998
- G. Di Caro and M. Dorigo, "Two Ant Colony Algorithms for Best-Effort Routing in Datagram Networks", Proceedings of the Tenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98), Las Vegas, Nevada, IASTED/ACTA Press, pages 541-546, 1998
- G. Di Caro, "Ant Colony Optimization and its Application to Adaptive Routing in Telecommunication Networks", Ph.D. thesis, Faculté des Sciences Appliquées, Université Libre de Bruxelles, Brussels, Belgium, 2004
- G. Di Caro, F. Ducatelle and L.M. Gambardella, "AntHocNet: an adaptive Nature-inspired algorithm for routing in mobile ad hoc networks", European Transactions on Telecommunications, Vol. 16(5), 2006 (to appear)
- F. Ducatelle, G. Di Caro and L.M. Gambardella, "Using ant agents to combine reactive and proactive strategies for routing in mobile ad hoc networks", International Journal of Computational Intelligence and Applications, 2006 (to appear)

You can download these papers, as well as other papers related to AntNet and AntHocNet, from: `http://www.idsia.ch/~gianni/my_publications.html`.

These papers also contain extensive pointers and discussions to the quite vast literature related to ACO-routing implementations

The End

Thanks for listening!