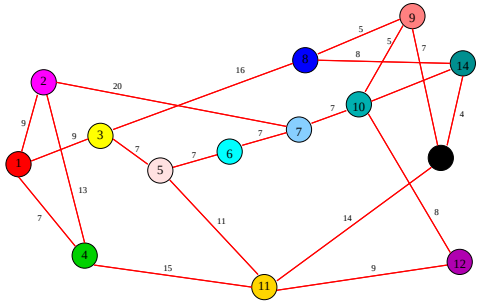
Extending AntNet for Best Effort Quality-of-Service Routing



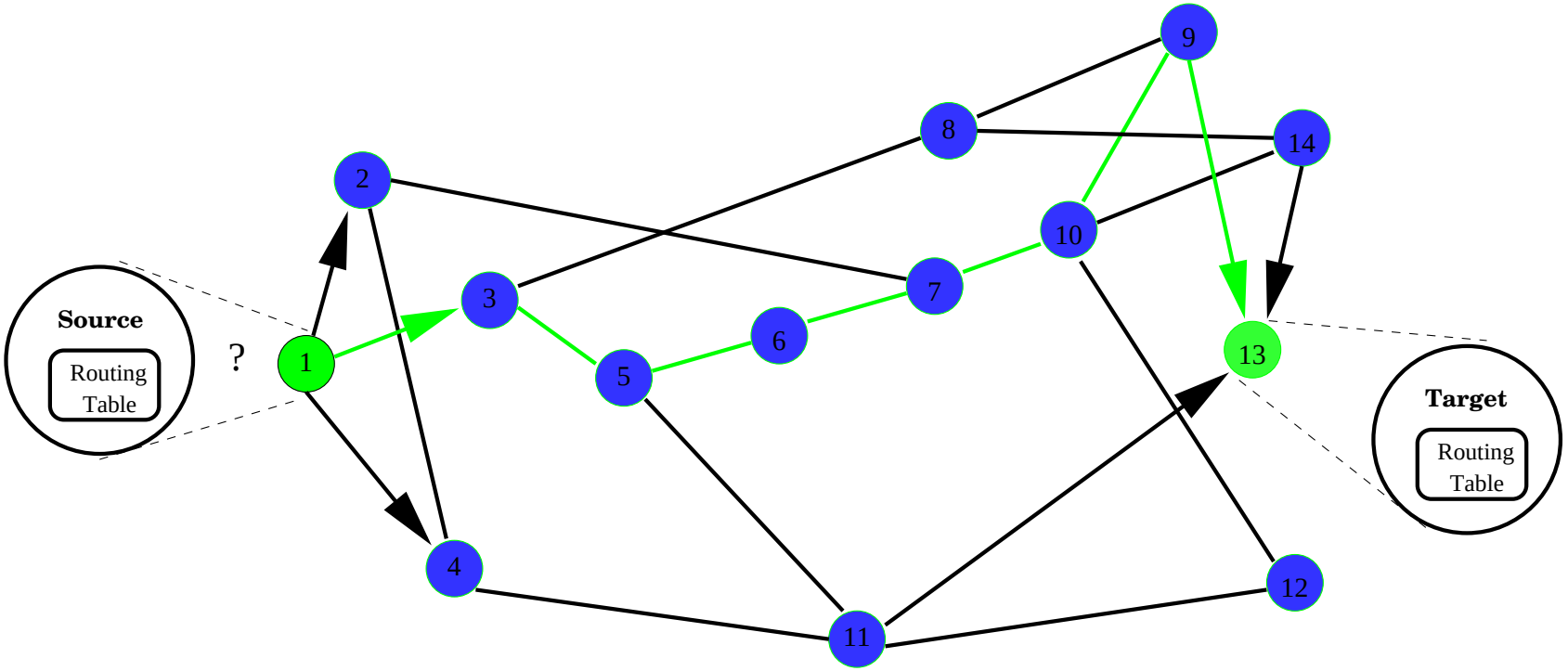
Gianni Di Caro and Marco Dorigo

IRIDIA, Université Libre de Bruxelles - Belgium

- Basic ideas behind ant colony optimization (ACO) and routing
- AntNet (1997): ACO for best-effort adaptive routing in datagram nets
- AntNet-FA (1998): ants can fly!
- AntNet-FS (1998): ants for Fair-share QoS networks
- Experimental results

Routing in Networks

Quo Vadis ?



- A Routing Algorithm is Distributed over all the network nodes.
- It has to select the best outgoing link to Forward data packets toward their destination nodes.
- It builds and uses Routing Tables.

Connection-Less Routing: Metrics and Goals

- **Measures of Performances**
  - Packet Delays Distribution (sec): quality of service
  - Throughput (bits/sec): quantity of service
  - Network Resource Utilization

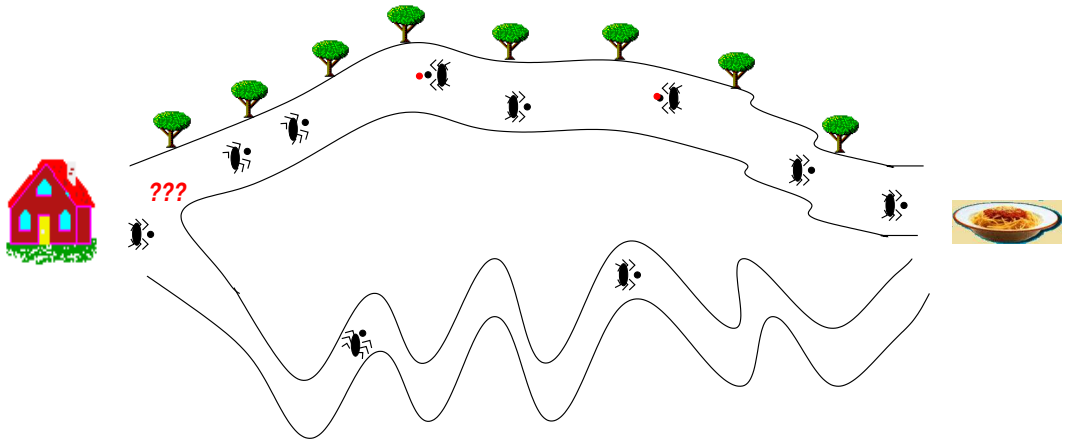
- **Goals of Good Routing**
  - Under high load: increase throughput for the same average delay
  - Under low load: decrease the average delay per packet

Ant Colony Optimization (ACO): Background

Got Pests? 

Ant Colony Optimization (ACO) techniques have been inspired by the following facts:

- It has been experimentally observed that many species of real ants are able to find shortest paths between their nest and sources of food.



- Shortest paths can emerge as the result of a collective behavior catalyzed by the use of:
  - a probabilistic local decision rule to move,
  - an indirect form of communication, called Stigmergy.

Pheromone and Stigmergy

- Ants deposit on the terrain's state they visit a chemical substance, called Pheromone that:
  - modify the way the terrain is locally perceived by following ants,
  - locally bias the ants' moving decisions.

- By locally reading/writing the pheromone trail:
  - the ants locally communicate in an indirect way mediated by the environment $\longrightarrow$ *Stigmergy*,
  - the whole colony shows an *Autocatalytic* behavior.

Ant Colony Optimization (ACO)

- The real ants' behavior inspired many algorithms for discrete optimization (1991 ... 1998) $\rightarrow$ ACO meta-heuristic.

- ACO algorithms add many components to the basic behavior of real ants to make the algorithm really effective.

- ACO algorithms are the best heuristics for:
  - Quadratic Assignment Problems (QAP)
  - Sequential Ordering Problem (SOP)

- Among the best heuristics for:
  - Traveling Salesman Problem (TSP)
  - Graph Coloring
  - Vehicle Routing Problem (VRP)

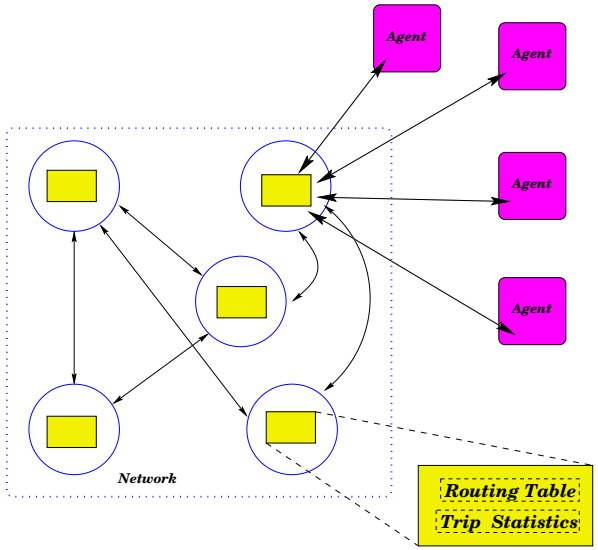
AntNet Meta-Description

The basic behavior of AntNet is very similar to the common behavior of many other ant algorithms (Dorigo, Di Caro, Gambardella, 1998).

- Every ant in a population of concurrent and asynchronous ants build (part of) a solution in an incremental way.
- Each ant applies a stochastic local search policy.
- While making "experiments" ants collect useful statistical information.
- The information is processed and locally deposited.
- The ant do not need to communicate directly. They communicate in a stigmergetic way through the information they locally read and write.
- The deposited information act like signals, modifying the way the optimization problem is perceived by subsequent ants, and triggering specific ant actions.

Collective Behavior

- The global routing problem cannot be efficiently solved by a single ant (intrinsically distributed problem).
- Good quality solutions emerge as the result of a collective process.
- There is no need for explicit coordination and synchronization.
- Agents communicate indirectly through Stigmergy mediate by the network.

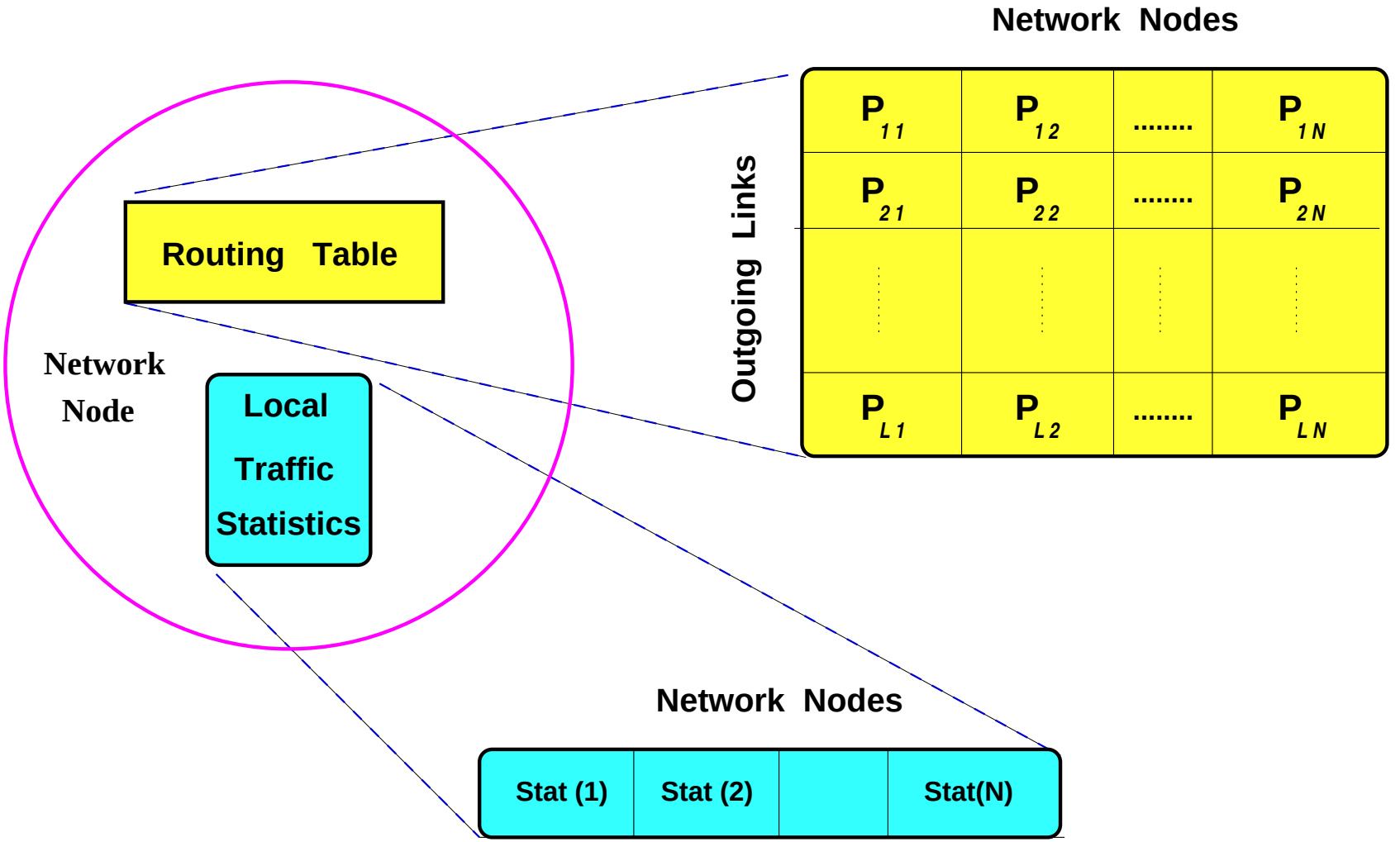


AntNet

- AntNet is a distributed algorithm for best-effort adaptive routing in connection-less networks (Internet!).

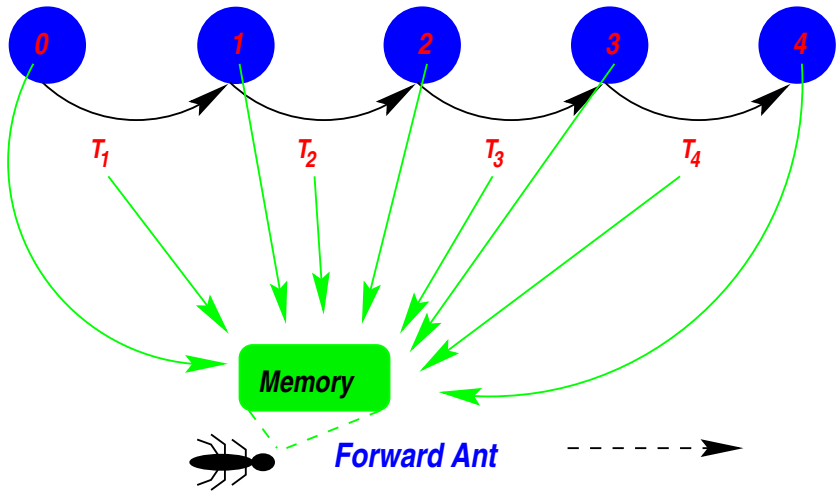
- The algorithm can be used for connection-oriented routing with suitable extensions $\rightarrow$ AntNet-FS under developing.

AntNet: Node Data Structures

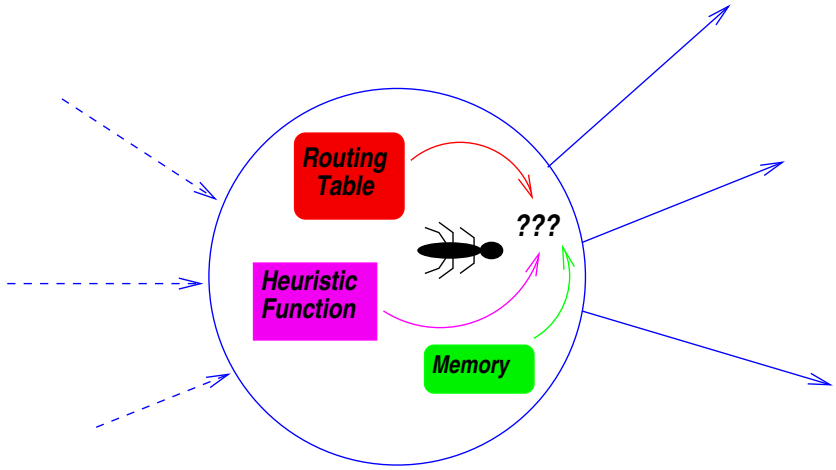


AntNet: Algorithm Description

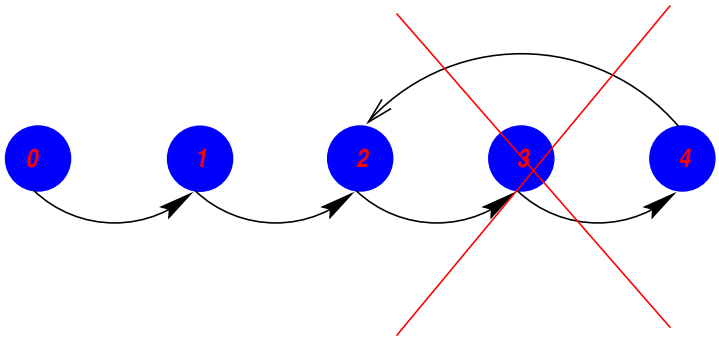
1. At regular intervals, from every network node, a mobile agent (Forward Ant) is launched, with a random destination matching traffic patterns.
2. The task of each ant is:
   - to discover a good path between its source and destination nodes,
   - to release on the visited node suitable information to direct the search process of future ants (variance reduction)
3. Each Forward Ant maintains a private memory of each visited node and of the elapsed times since its launching time.



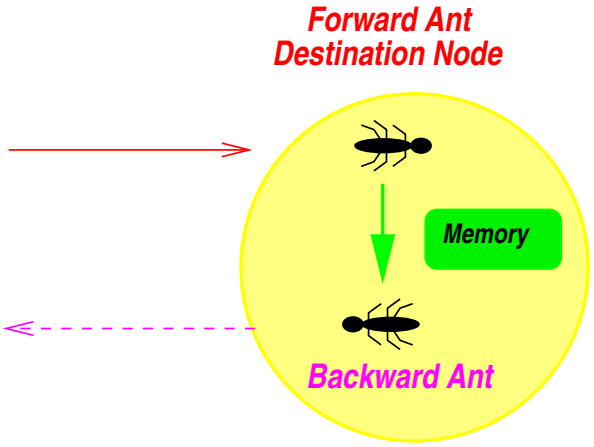
4. Next hop nodes are selected using:
   - probabilistic Routing Tables,
   - local heuristic based on the state of the local link queues,
   - memory of the past visited nodes



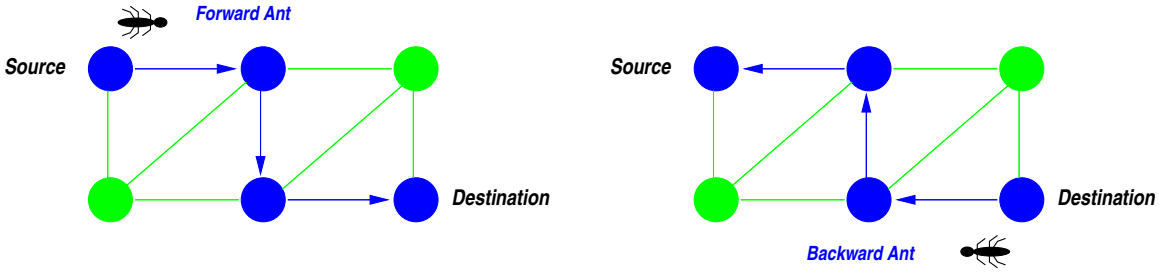
5. A simple backtracking procedure recover from cycles, that are deleted from the ant memory.



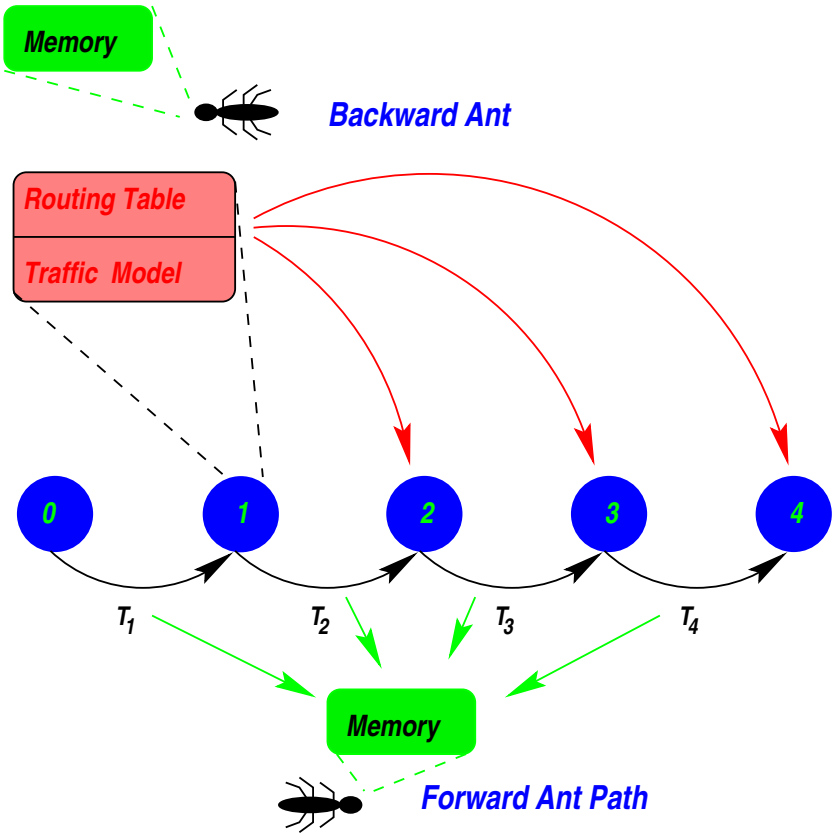
6. Reached the destination node, the Forward Ant generates a Backward Ant, transfers to it all its memory and dies.



7. The Backward Ant visits the same nodes of the Forward Ant, but in the opposite direction.



8. At each node the Backward Ant updates the local Routing Table and the Local Traffic Model using memorized information and the local model itself.



9. The amount of deposited "pheromone" depends on the evaluation of the path the ant discovered.

Forward Vs. Backward Ants

- The FORWARD ANTs share the same queues as data packets. In this way they collect information on the available paths and the traffic load:
  - Implicit information, through the rate of arrival to destination nodes.
  - Explicit information, storing the experienced trip times.
- The BACKWARD ANTs Back Propagate this information to the visited nodes as fast as possible (they have higher priority over data packets).

Main Components of AntNet

1. Frequency of ant generation at each node.
2. Selection of ant destination nodes.
3. Local heuristic for ant decisions.
4. Local ant decision policy.
5. Private ant memory.
6. Backtracking procedures.
7. Path retracing strategy.
8. Complexity of the local traffic models.
9. Implicit and explicit evaluation of the "solution" (path) the ant discovered.
10. Filtering of the ant collected information.
11. Routing table updating rule.
12. Routing table utilization by data packets.

Local Decision Policy

- Local heuristic for ant decisions:

$$l_n = 1 - \frac{q_n}{\sum_{n'=1}^{|\mathcal{N}_k|} q_{n'}}$$

$\mathcal{N}_k = \{neighbors(k)\}$,

q_n = length of the queue of the link connecting the node k with its neighbor $n \in \mathcal{N}_k$

- Local ant decision policy (at node k towards destination d):

$$P'_{nd} = \frac{P_{nd} + \alpha l_n}{1 + \alpha(|\mathcal{N}_k| - 1)}.$$

P_{nd} = Routing table entry, long-term memory
l_n = short-term prediction.

- The probabilistic decision rule is applied to all the still not visited neighbors. To all the neighbors in case all of them have been already visited.

Local Traffic Models

- A local estimate of the network status from the node point of view.
- Play an important role in the evaluation of the ant discovered paths and in the statistical filtering of the ant collected information.
- Simple parametric models.
- Moving exponential estimates of mean and dispersion for the observed ant trip times:

$$\begin{aligned} \mu_d &\leftarrow \mu_d + \eta(o_{k\to d} - \mu_d), \\ {\sigma_d}^2 &\leftarrow {\sigma_d}^2 + \eta((o_{k\to d} - \mu_d)^2 - {\sigma_d}^2) \end{aligned}$$

$o_{k\to d}$ is the new observed agent's trip time from node k to destination d.

Evaluation of the Forward Ant Path

- The path discovered by the ant is evaluated in an:
  - implicit way, through the time elapsing between path discovering and routing tables updates.

    implicit way, through the routing table updating policy, that allow to exploit the frequency the ants choose a path.
  - explicit way, through the evaluation of the ant trip time T.
- The implicit evaluation comes at zero cost from the distributed nature of the problem and from the choosed updating rule.
- The explicit evaluation is based on T as a measure of the goodness of the discovered path (physical length and traffic congestion).

Path Goodness Measure

- T cannot be scored on the basis of an absolute scale, but it is locally traffic-dependent.
- We associate to the trip time T a goodness measure R, $R \in]0,1]$ function of T and of the local model:

$$r = c_1 \left(\frac{W_{best}}{T} \right) + c_2 \left(\frac{I_{sup} - I_{inf}}{(I_{sup} - I_{inf}) + (T - I_{inf})} \right)$$

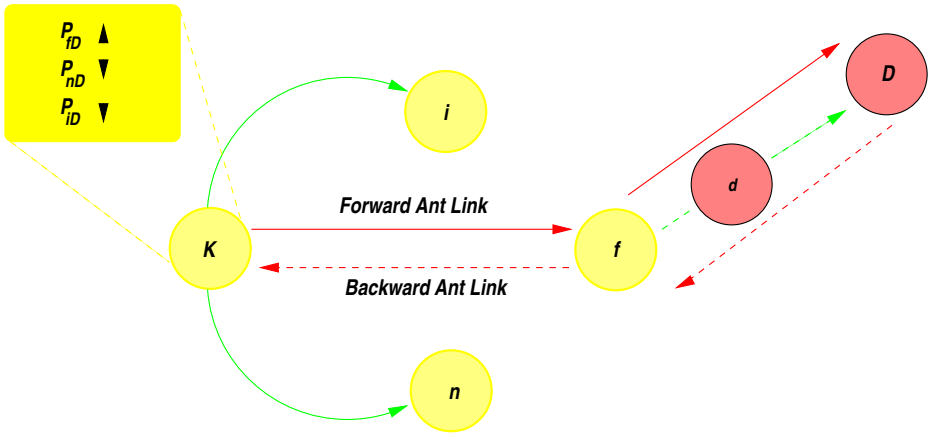
W_{best} is the best trip time experienced by the ants traveling toward the destination d, over the last observation window $\mathcal{W}$.

Is are estimated of confidence intervals for the expected T values.

- R is transformed by a squash function to allow the system to be more sensitive in rewarding good (high) values of R, while having the tendency to saturate the rewards for bad (near to zero) R values.
- The transformed goodness measure R is used to assign Reinforcements to the probability values in the Routing Tables.

Routing Table Updating Rule

■ The Probabilities of the Outgoing Links for node D (and d on the sub-paths) as destination are modified by the Reinforcement value R:



- The node f the Backward Ant comes from receives a positive reinforcement:

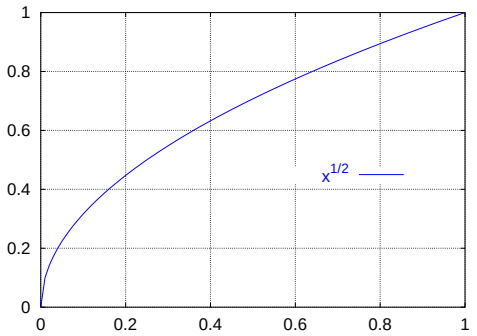
$$P_{Df} \leftarrow P_{Df} + R(1 - P_{Df})$$

- All the other neighbor nodes n receive, by probabilities normalization, a negative reinforcement:

$$P_{Dn} \leftarrow P_{Dn} - R \cdot P_{Dn}$$

Properties of Probabilistic Routing Tables

- Probabilistic Routing Tables give a built-in exploration method.
- Data packets also use the Routing Table in a probabilistic way, after transforming it by a power law:



- Small probability values are increased proportionally more than big probability values, favoring a quick exploitation of new and good discovered paths.
- All the ants increase in some measure the probability of the choosed path $\rightarrow$ cumulative reinforcement.

Algorithm Complexity Vs. Performance

- The algorithms takes into account many "details" and components.
- We experimentally observed (and we conjecture) that most of the algorithm components are really necessary to obtain state-of-the-art performance (similar situation as for Combinatorial Optimization).
- Anyway, we are working to make the algorithm more "simple".
- The computational overhead is almost negligible. Computations are carried out only by ants, they do not involve data packets.
- Traffic overhead can be kept negligible while obtaining state-of-the-art performance.

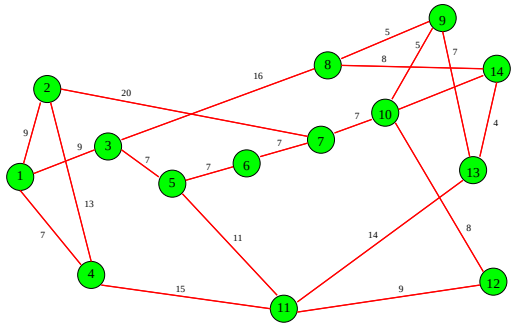
Experimental Setup

Setting of Realistic Conditions for:

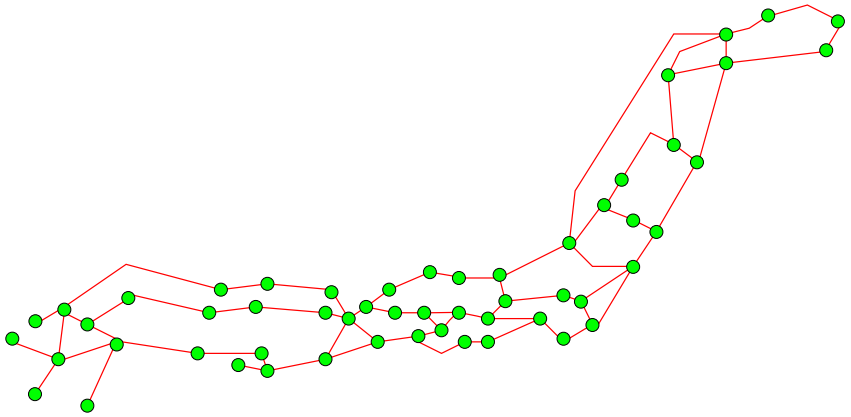
- Network topology and physical characteristics.
- Protocol for data transmission.
- Spatial and Temporal Traffic Patterns.
- Algorithms to compare the performances.

Networks

- NSFNET-T1, the old US backbone (1987). 14 Nodes, 21 Bi-directional Links (1.5 Mbit/s).

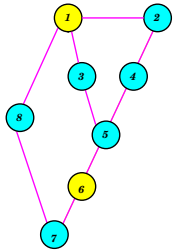


- NTT NET, Nippon Telegraph and Telephone corporation fiber-optic network. 57 Nodes (7 Junctions), 162 Bi-directional Links (6 Mbit/s).

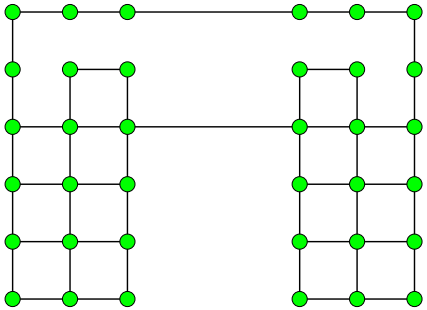


Networks

- Random Networks. Sets of randomly generated networks, 100 and 150 Nodes, mean connectivity degree ≈ 3, links at 10 Mbit/sec, 1-10 msec Transmission Delay.
- Simple Net. 8 Nodes, 9 Bi-directional Links (10 Mbit/sec, 1 msec Transmission Delay.)



- 6x6 Net. 36 Nodes, 99 Bi-directional Links (10 Mbit/sec, 1 msec Transmission Delay.) Regular topology network proposed by (Boyan & Littman, 1994).



Data Transmission Protocol

- Best-effort Datagram traffic.
- IP-like protocol.
- Packets can be discarded because of no buffer space.
- Failure situations are not taken into account.
- No arrival acknowledgement or error notification packets are generated.
- Simple Flow control mechanism: a static Production window limits the maximum number of produced but not still sent data packets.

Traffic Characteristics

- Inter-arrival times and Sizes for Sessions have Negative Exponential Distribution.
- Inter-arrival times and Sizes for Packets in each Session are generated following a Negative Exponential Distribution.
- Traffic Patterns are 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

1. OSPF: Minimum cost paths, current IGP Internet algorithm.

Adaptive - Link-state

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

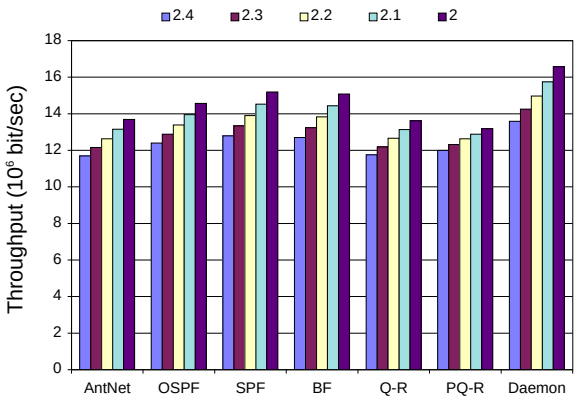
Adaptive - Vector-distance

3. BF: Asynchronous Bellman-Ford prototype, Adaptive link costs, old ARPANET algorithm.
4. Q-Routing: Asynchronous Bellman-Ford with online updates and Q-Learning-like updating rule.
5. PQ-Routing: Q-Routing with a system to learn a model of the link queues.

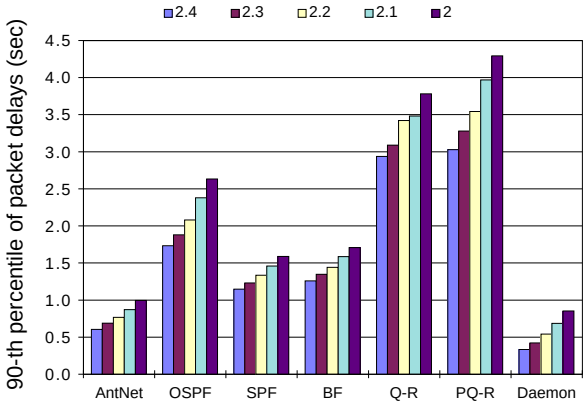
Ideal

6. Daemon: Access the state of all the net queues. Empirical bound on performances

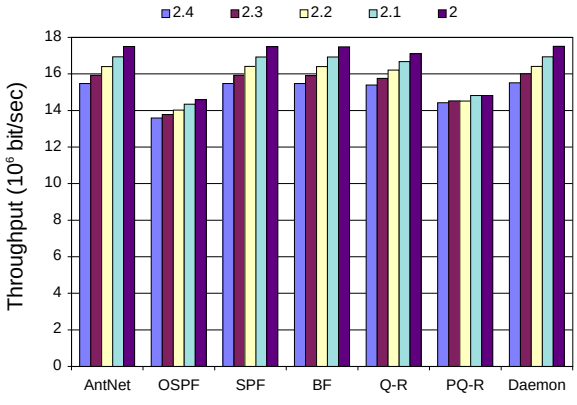
Results - NSFNET



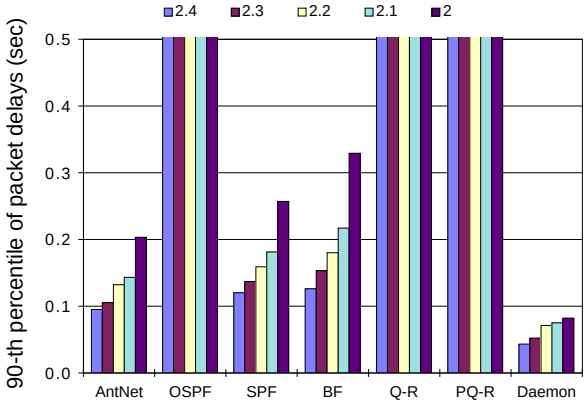
UP Load - Throughput



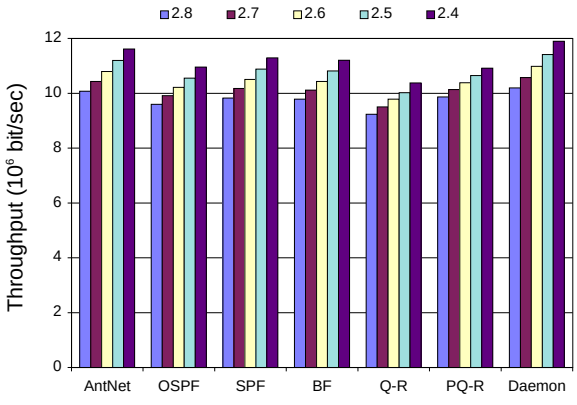
UP Load - Delay



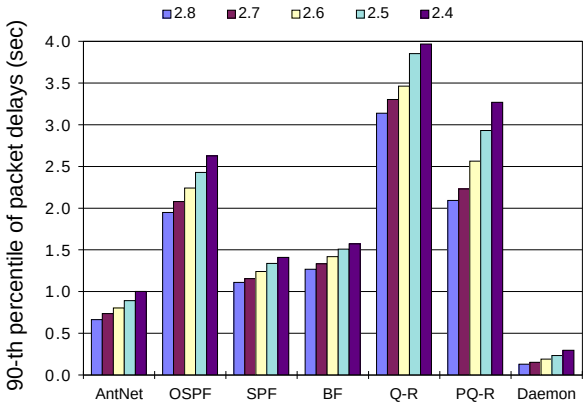
UPHS Load - Throughput



UPHS Load - Delay

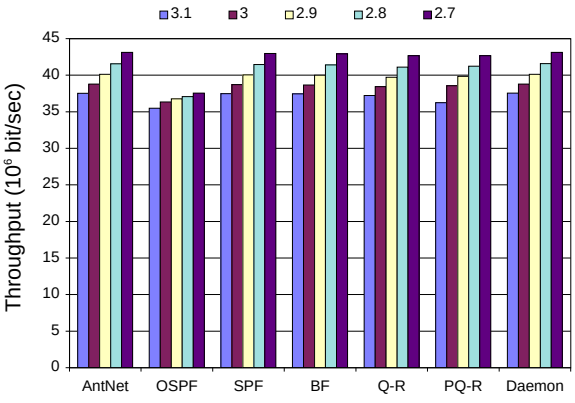


RP Load - Throughput

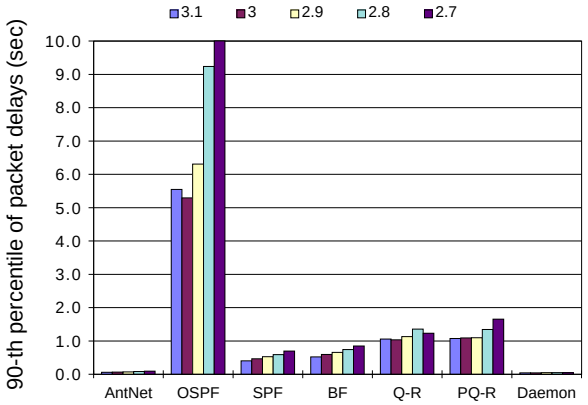


RP Load - Delay

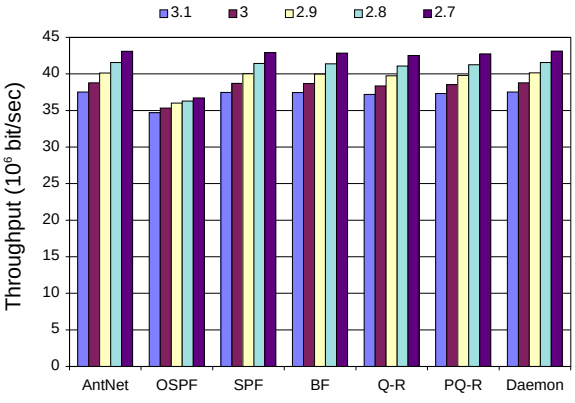
Results - NTTNET



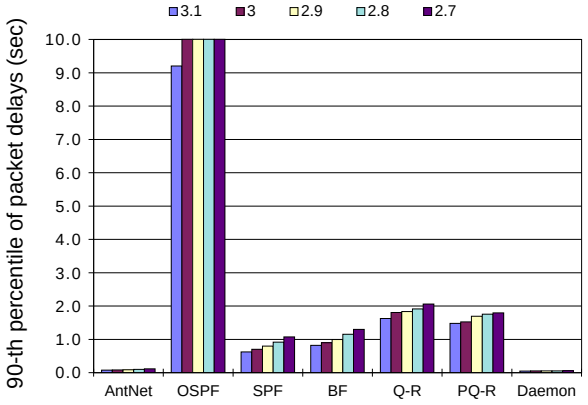
UP Load - Throughput



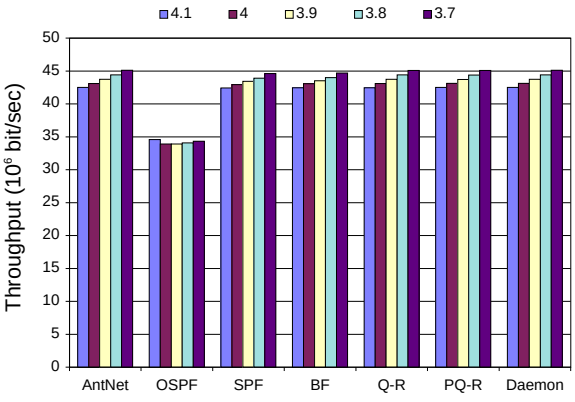
UP Load - Delay



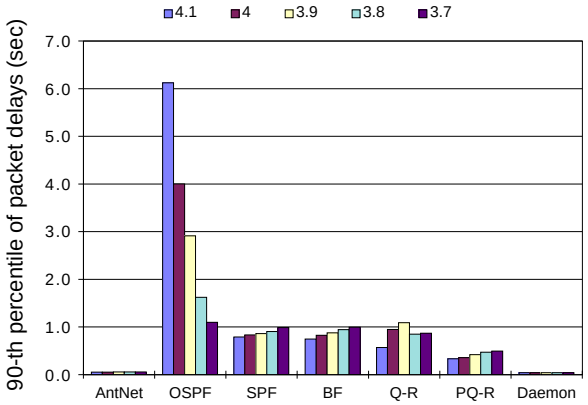
RP Load - Throughput



RP Load - Delay



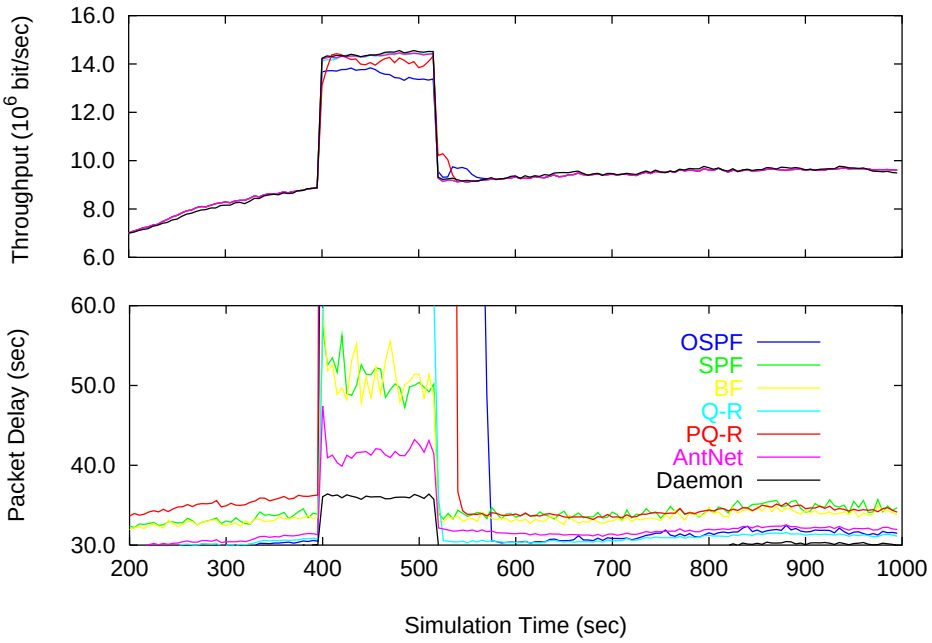
UPHS Load - Throughput



UPHS Load - Delay

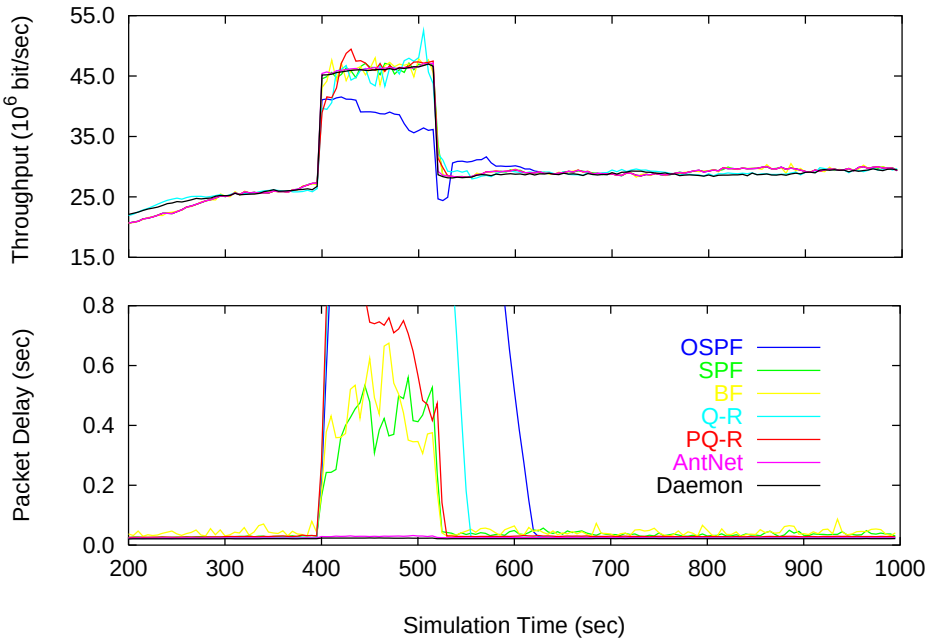
Results - Load Variation

NSFNET: UP Load Variation



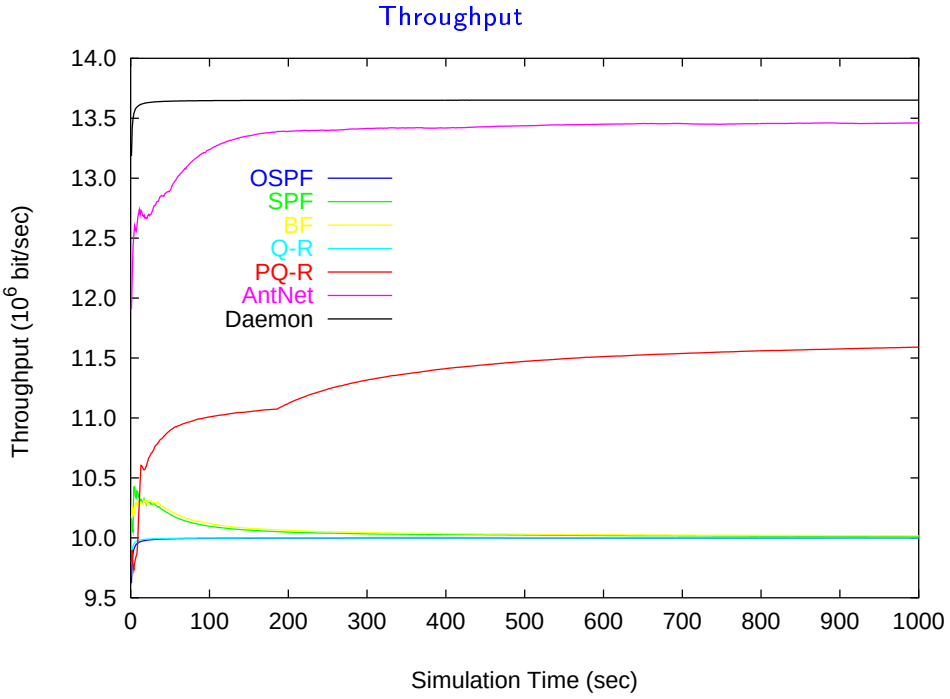
- Simulation Parameters: MSIA=3.0 sec, MPIA=0.3 sec, HS=4, MPIA-HS=0.04

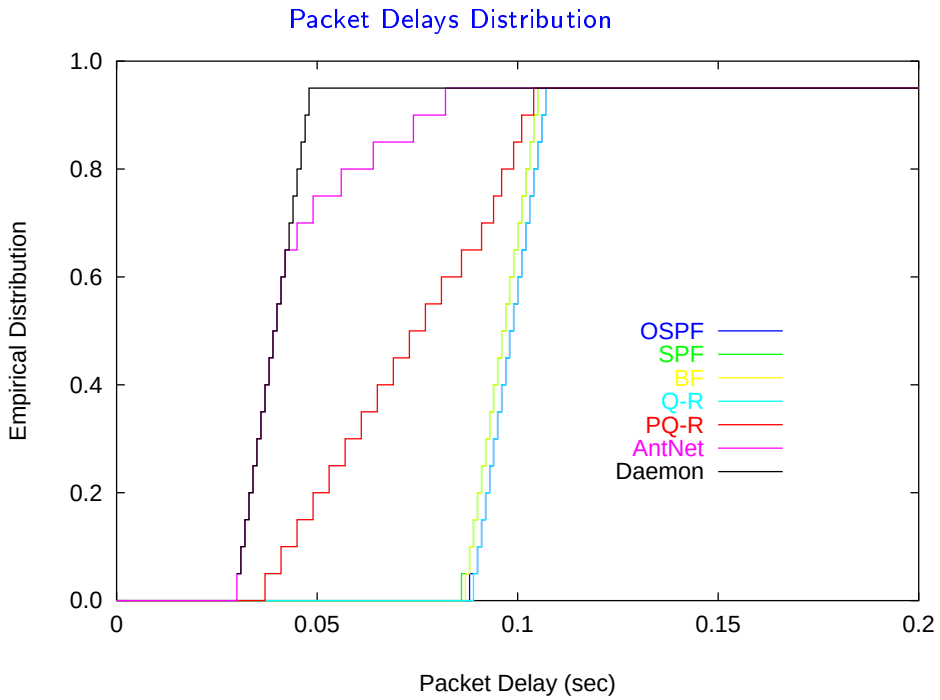
NTTnet: UP Load Variation



- Simulation Parameters: MSIA=4.0 sec, MPIA=0.3 sec, HS=4, MPIA-HS=0.05

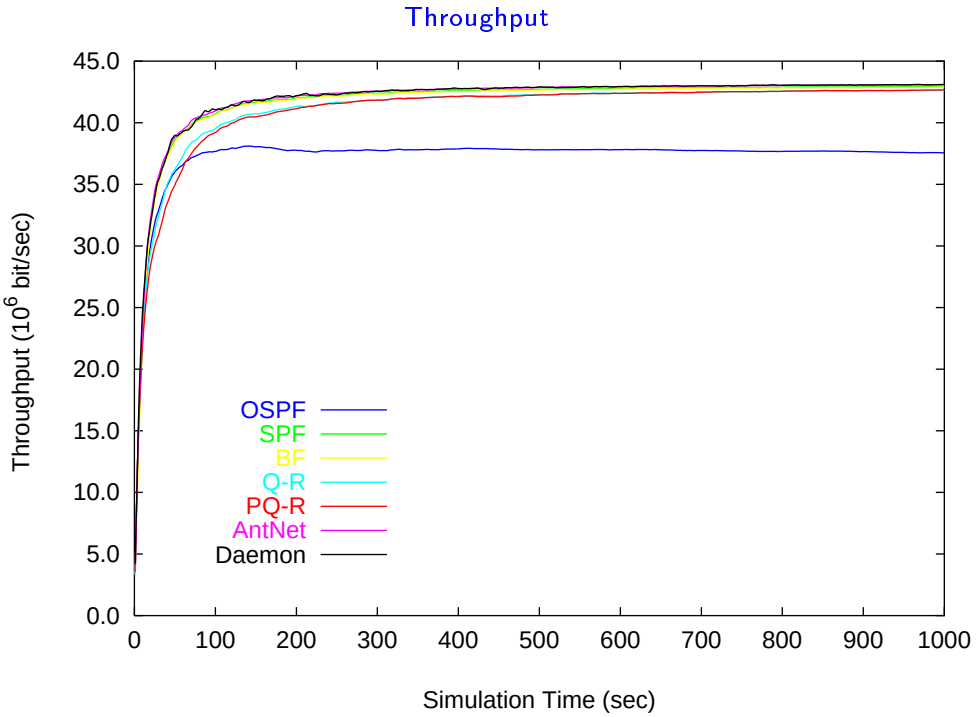
SimpleNet - CBR Load

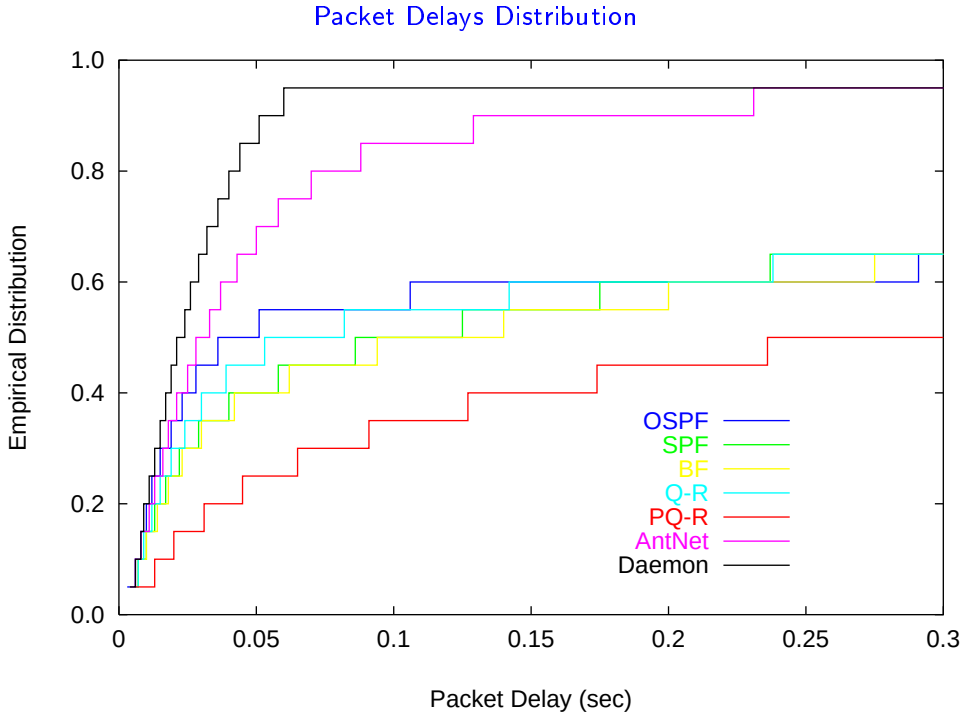




- - Simulation Parameters: CBR = 0.0003 sec

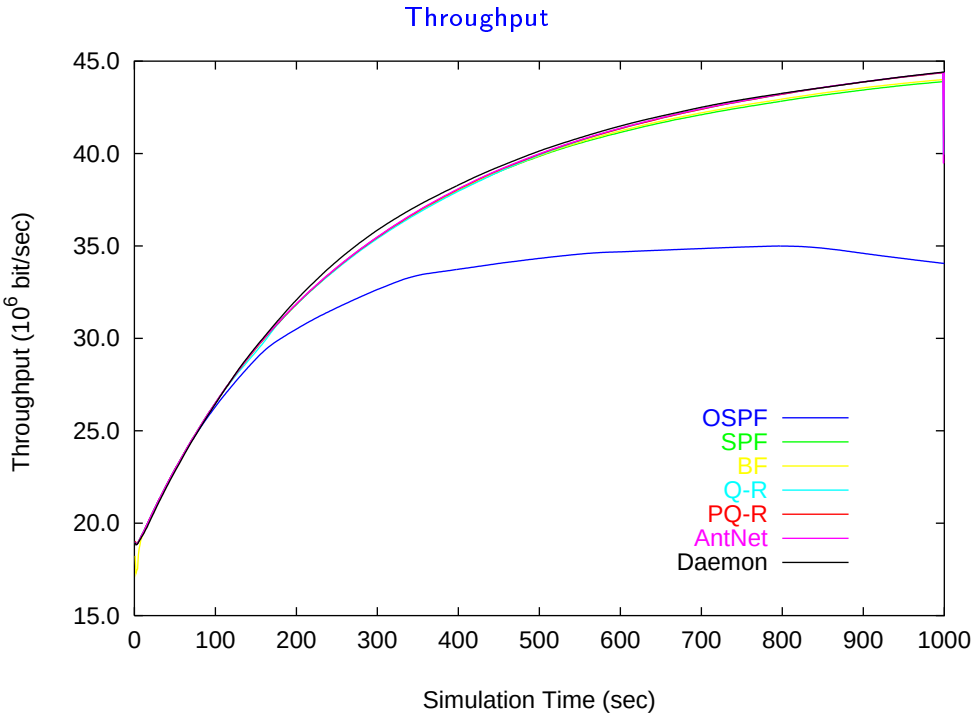
NTTnet - UP Load

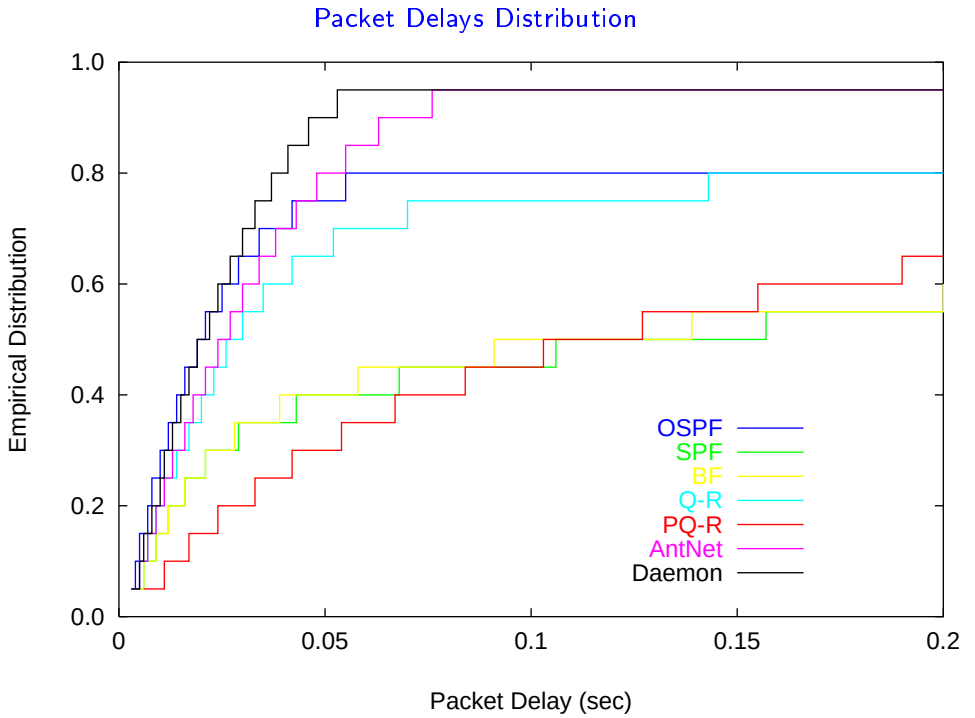




- - Simulation Parameters: MSIA = 2.7 sec, MPIA = 0.005 sec

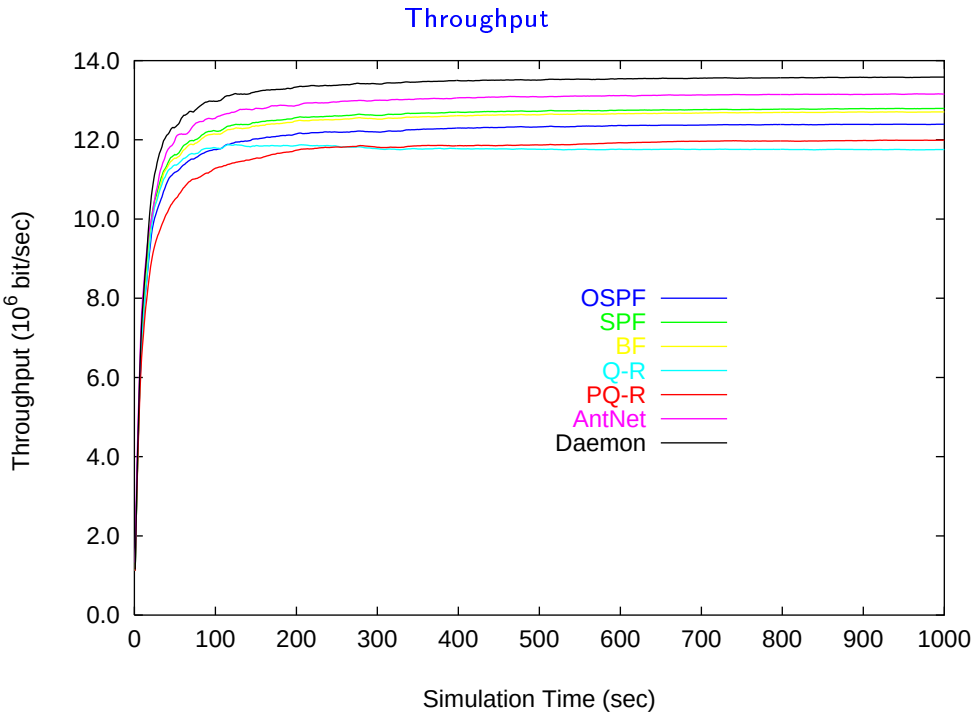
NTTnet - UPHS Load

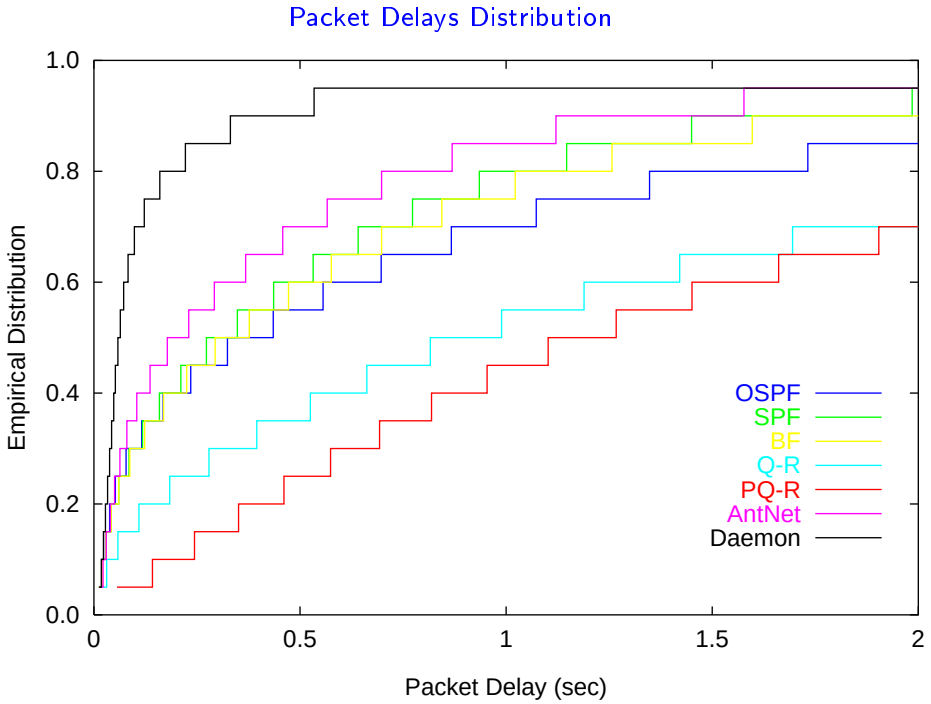




- - Simulation Parameters: MSIA=3.8 sec, MPIA=0.3 sec, HS=4, MPIA-HS=0.05 sec

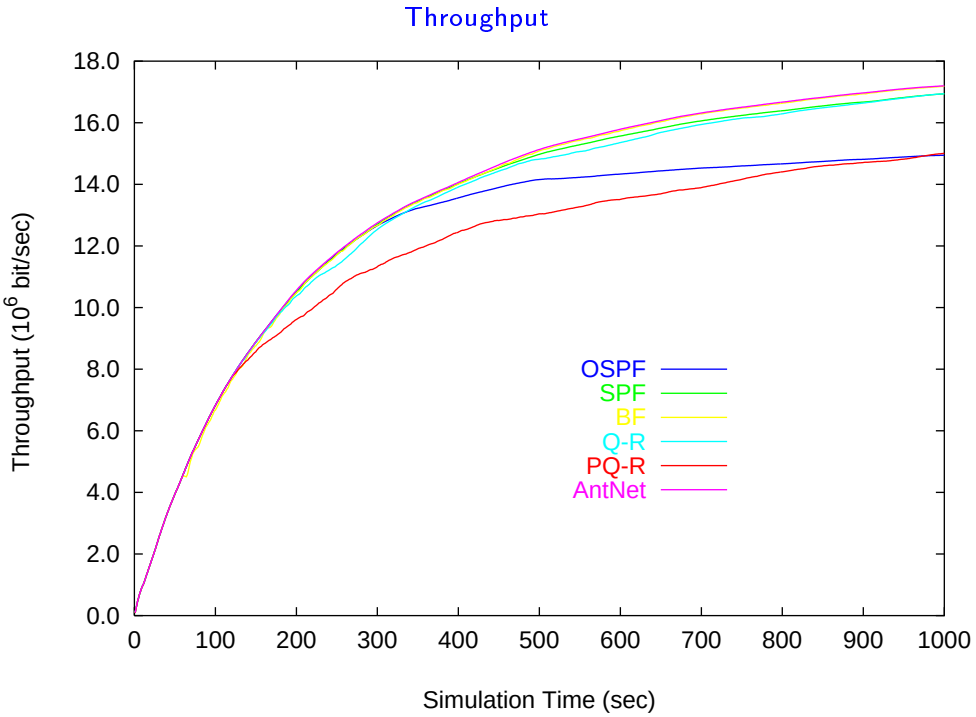
NSFNET - High Burstiness UP Load

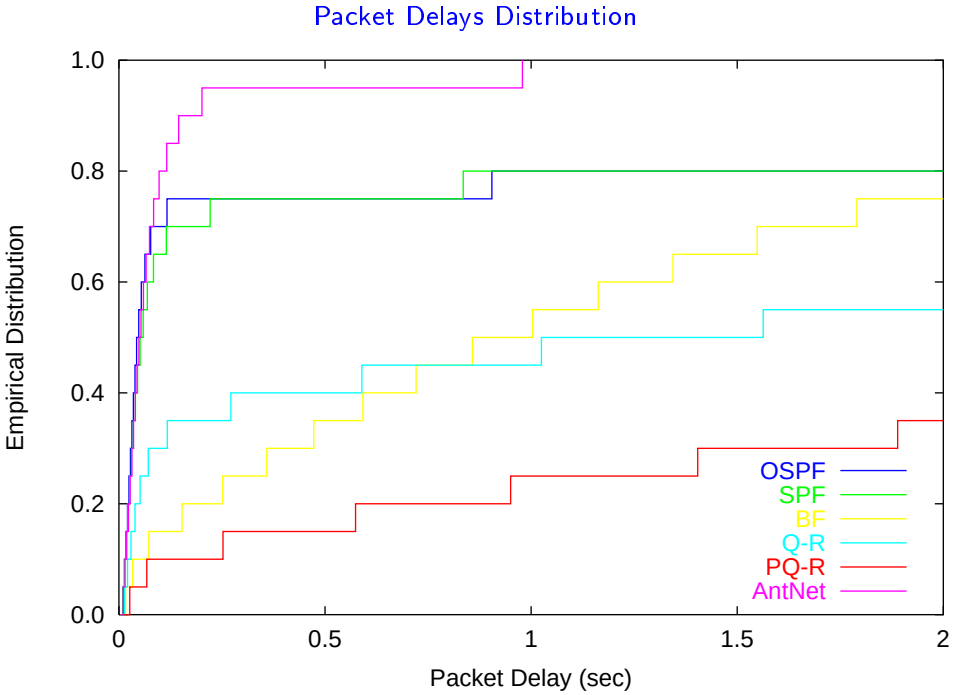




- - Simulation Parameters: MSIA=2.1 sec, MPIA=0.005 sec

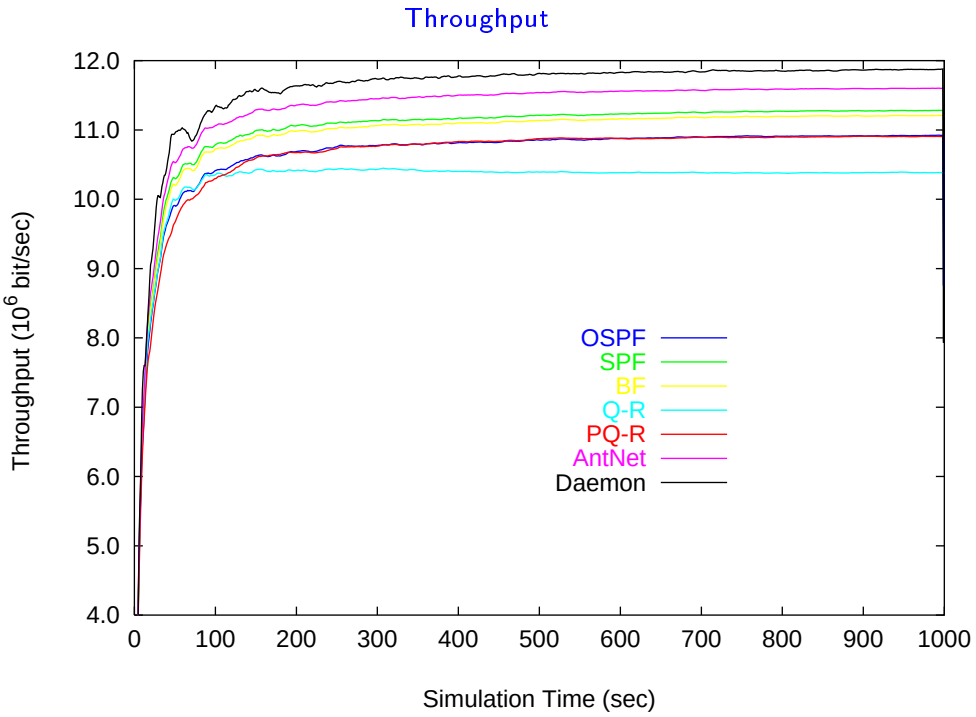
NSFNET - Low Burstiness UP Load

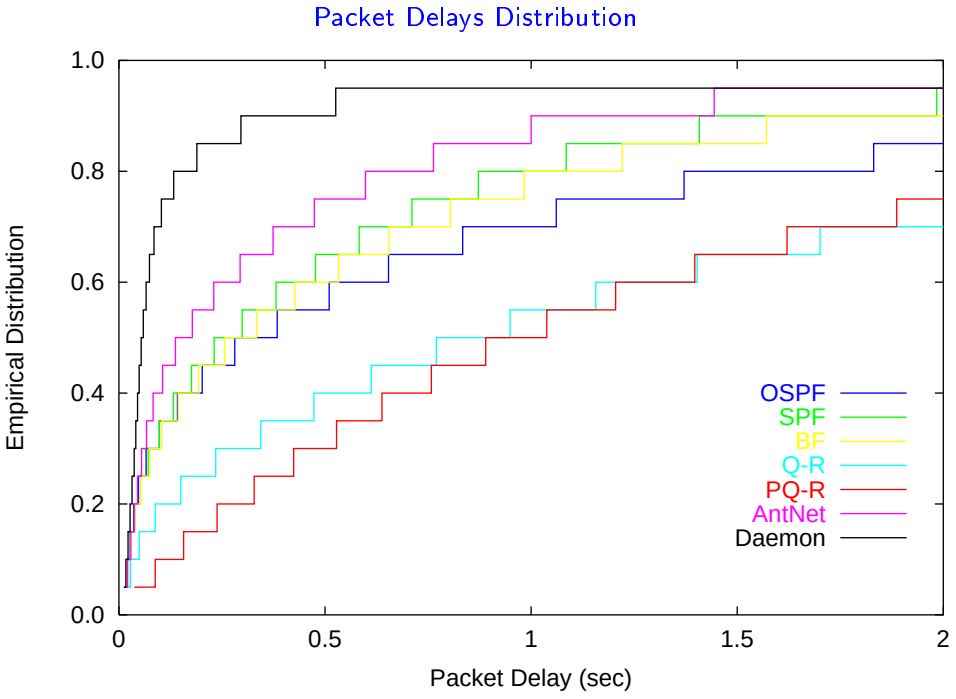




- - Simulation Parameters: MSIA=1.5 sec, MPIA=0.2 sec

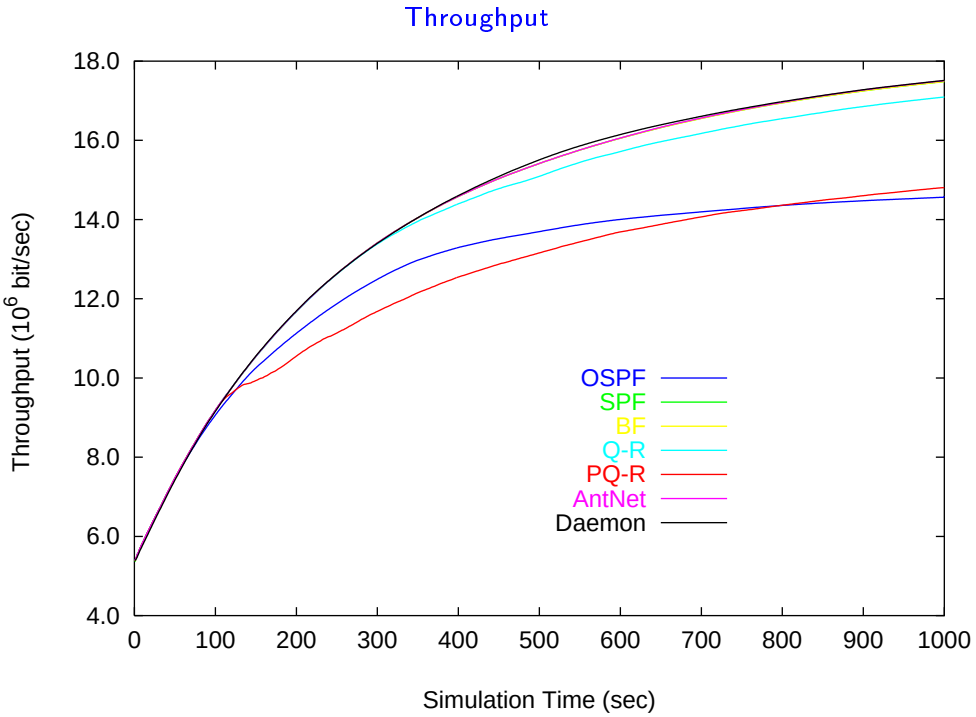
NSFNET - High Burstiness RP Load

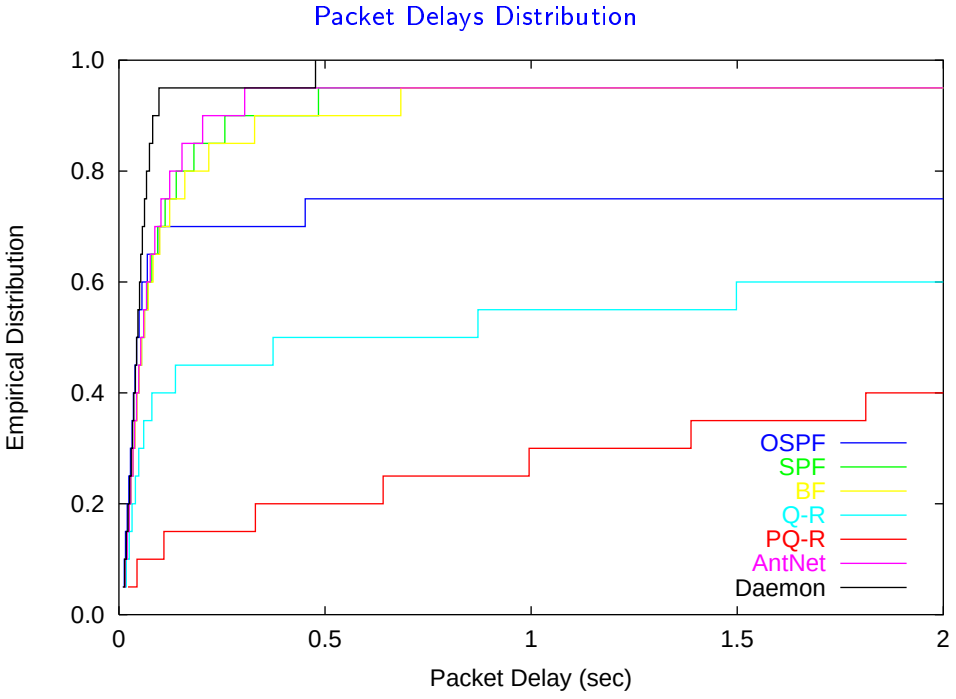




- - Simulation Parameters: MSIA=2.4 sec, MPIA=0.005 sec

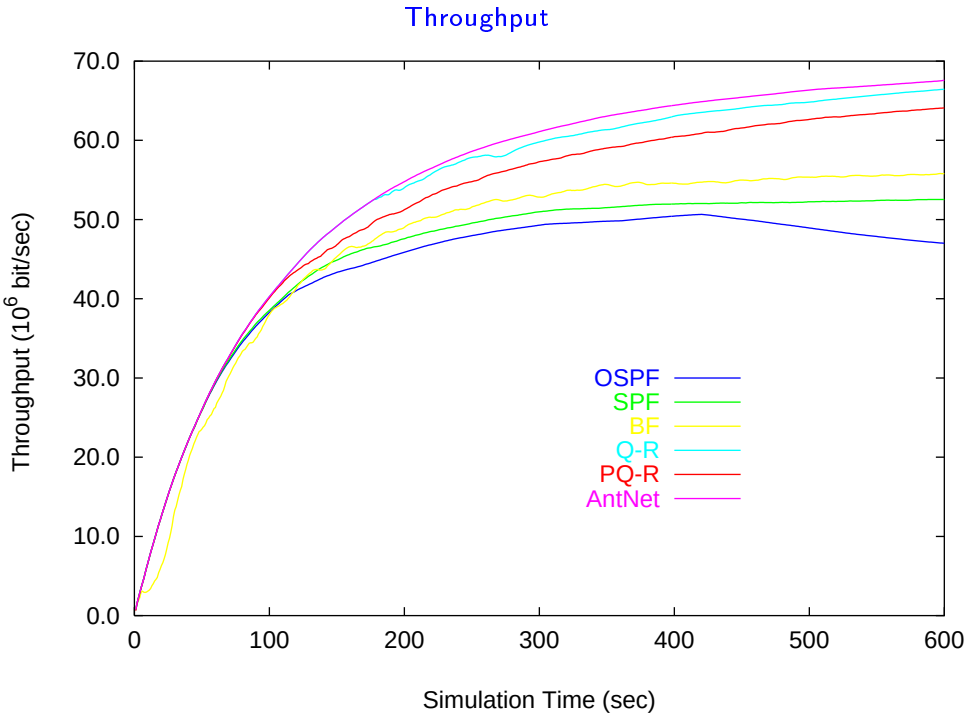
NSFNET - UPHS Load

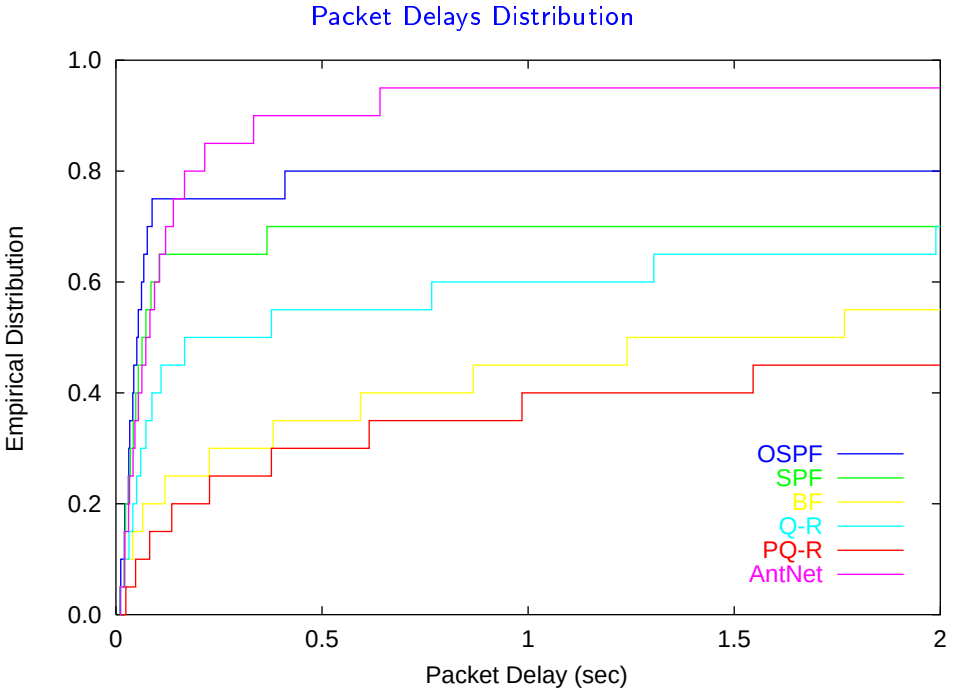




- - Simulation Parameters: MSIA=2.0 sec, MPIA=0.3 sec, HS=4, MPIA-HS=0.04

6x6net - Low Burstiness UP Load





- - Simulation Parameters: MSIA=1.0 sec, MPIA=0.1 sec

Routing Overhead

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

For all the considered algorithms the routing overhead was *negligible*.

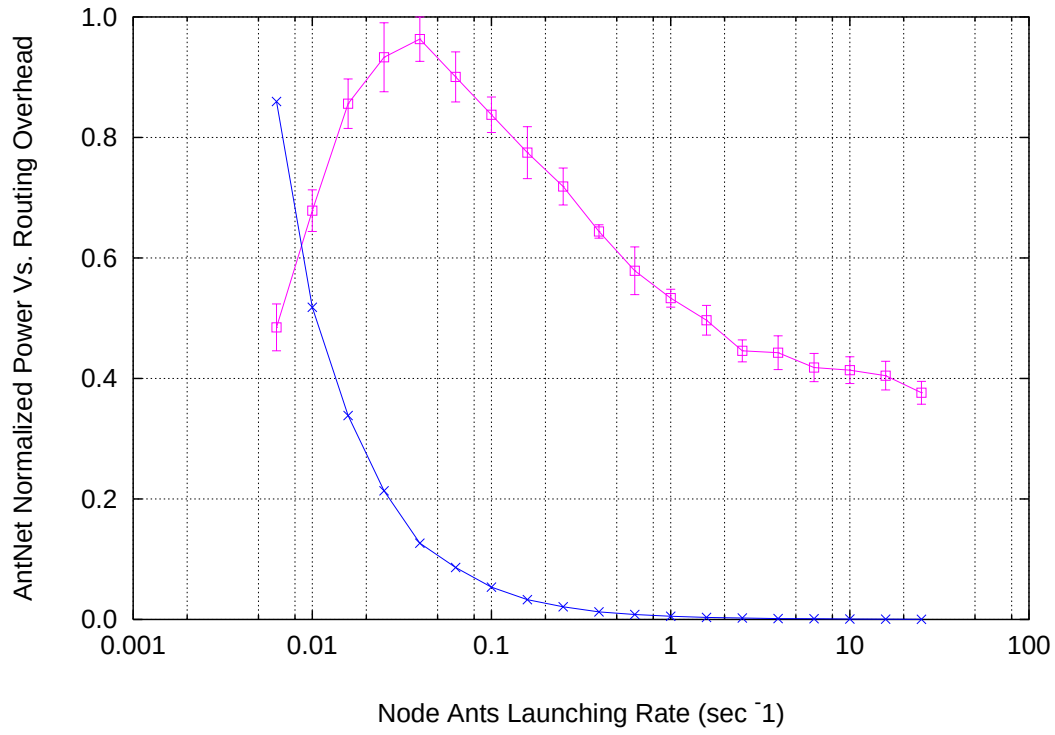
Routing Overhead (10^{-3}) for some of the realized experiments

	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

AntNet Power Vs. Routing Overhead

- Power = Ratio between the obtained Throughput and the 90-th Packet Delay Percentile.
- Routing Overhead = Fraction of the total network bandwidth resources used by the routing packets.

Normalized Power Vs. Routing Overhead
for increasing (per-node) rates of ants production.

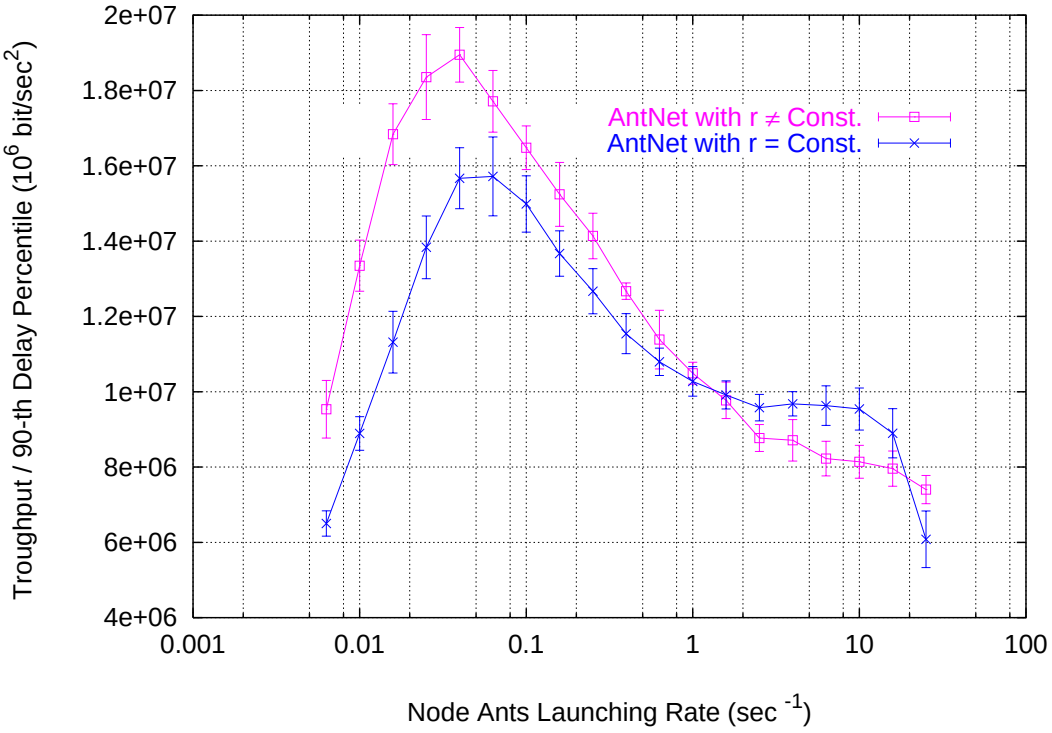


- Simulation Parameters: MSIA=2.4 sec, MPIA=0.005 sec

AntNet: Const. Vs. Non-Const. Reinf.

- Power curves for AntNet using non-constant vs. constant reinforcement values updating routing tables.
- Non-constant reinforcements give improvements ranging from few percent to more than 40%.

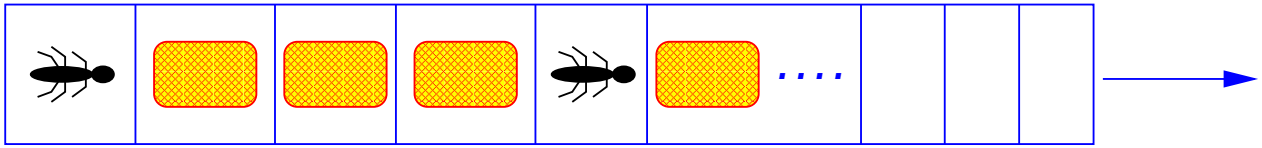
Constant Vs. Non-Constant reinforcements for increasing (per-node) rates of ants production.



- Simulation Parameters: MSIA=2.4 sec, MPIA=0.005 sec

Flying Ants

■ In AntNet Forward Ants behave like data packets, sharing the same link queues:



Link Queue

- In case of congestion or very long mean paths, this means that the collected information is propagated by Backward Ants with some not negligible delay.
- This delayed information could reflect no longer the current network status.
- This delayed behavior is not suitable to manage the setup phase in virtual-circuit networks.

■ We can give "wings" to Forward Ants to make the algorithm more reactive improving its performance → AntNet-FA.

AntNet-FA : A More Performing Algorithm

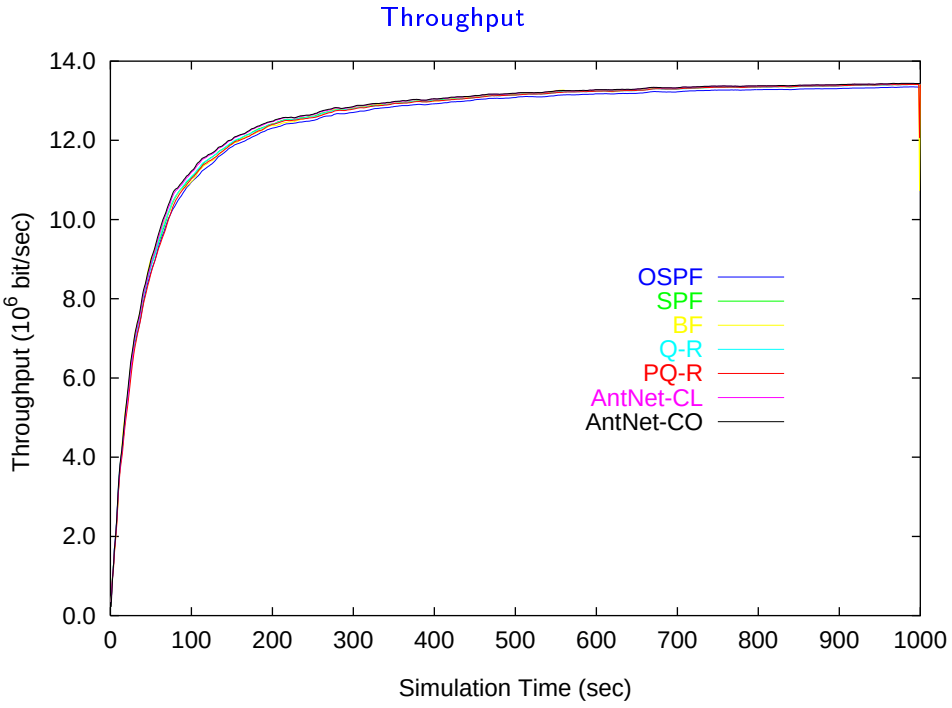
- AntNet-FA behaves like AntNet except for the following aspects:
  - Forward Ants make use of high-priority queues as Backward Ants.
  - They do not carry in the memory stack any information about their experienced trip times.
  - Backward Ants update the Routing Tables and the Local Models on the visited nodes using estimates of ants' trip times.
  - These estimates are computed at each node k, using a very simple local statistical model capturing the depletion dynamics of the queues of the local links:

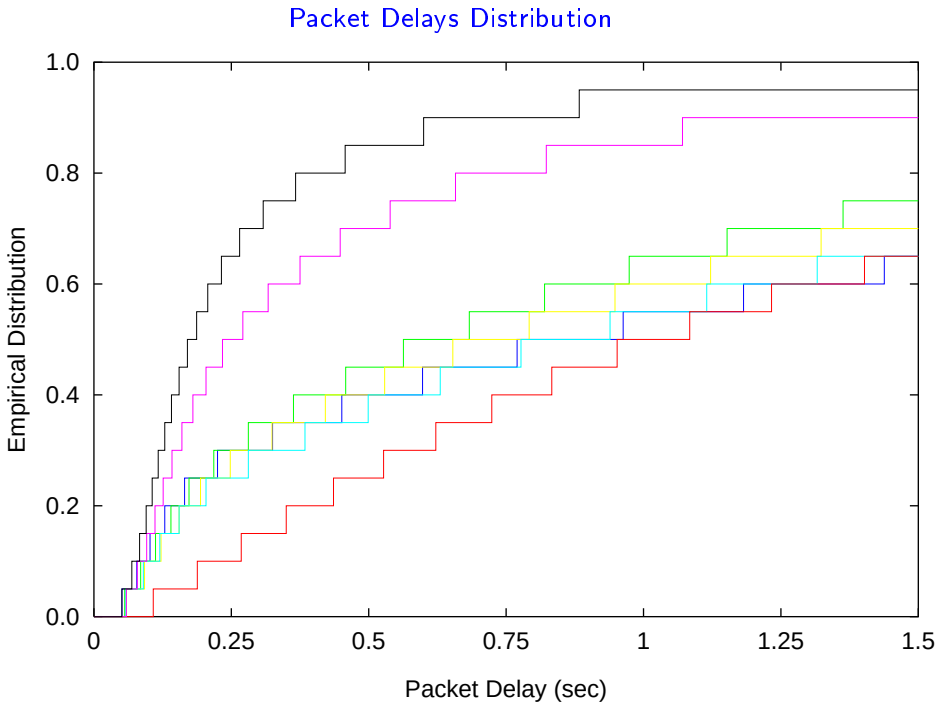
$$LinkCrossingTime = \frac{LinkQueueLenght + AntSize}{LinkBandwidth} + LinkPropagationTime$$

Characteristics of AntNet-FA

- Flying Forward Ants quickly discover a path and quickly the information about it is deposited by the Backward Ants.
- The used information is an estimate but it is much closer to the current instananeous state of the network.
- The simple estimates of the trip times are robust enough to evaluate the ant paths.
- AntNet-FA has been thought to support efficiently the setup phase in high-speed virtual circuit networks.
- It performs much better than the original AntNet when average paths become longer and longer and/or traffic conditions change quickly.

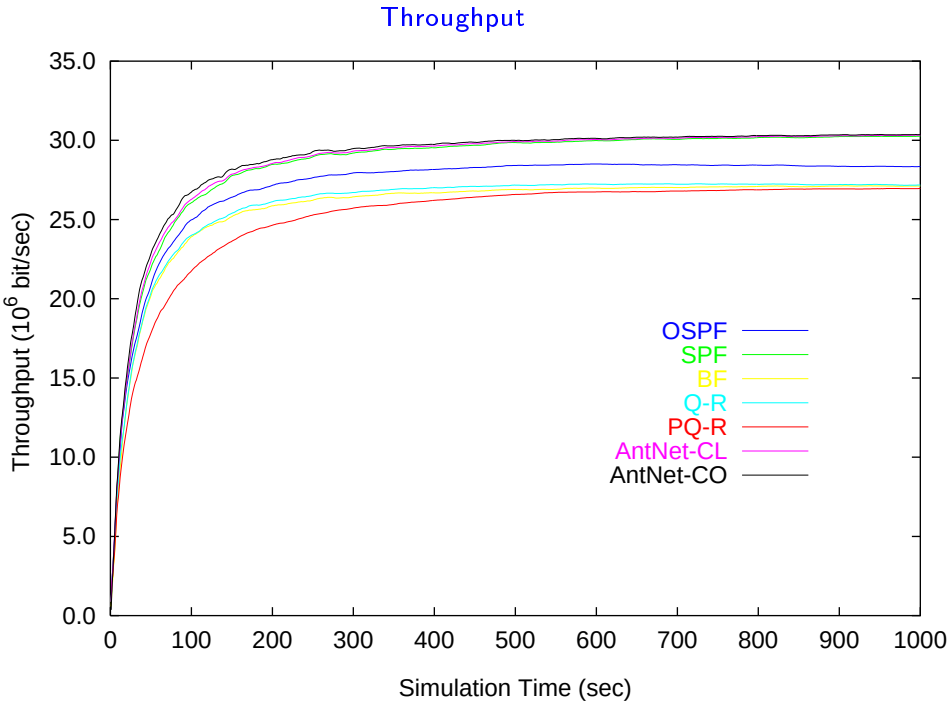
100-Nodes RandomNets - UP Load

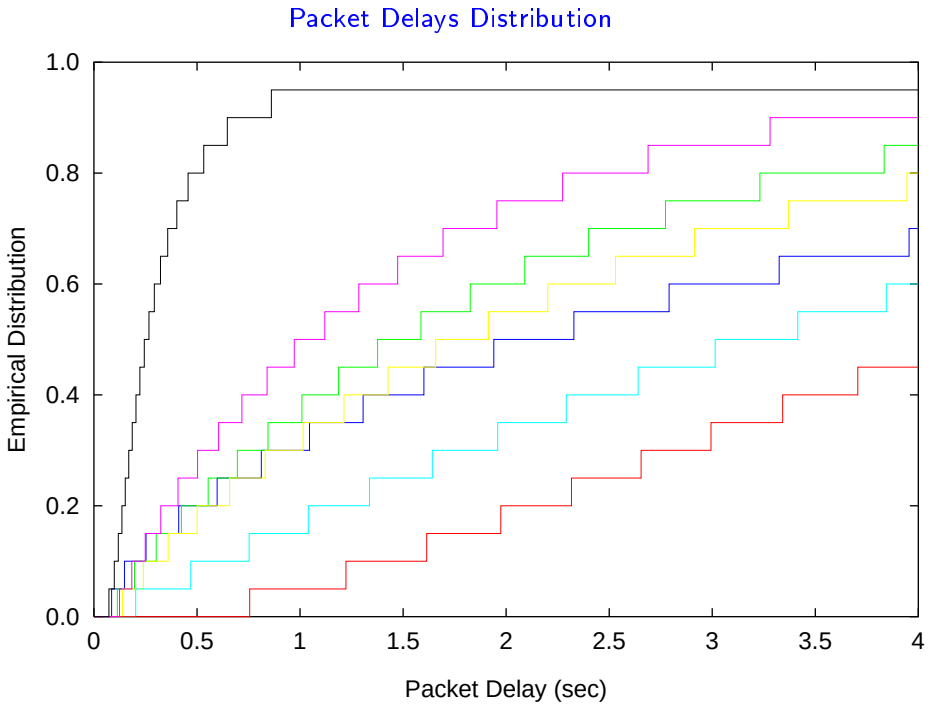




- - Simulation Parameters: MSIA = 15.0 sec, MPIA = 0.005 sec

150-Nodes RandomNets - RP Load





- - Simulation Parameters: MSIA = 10.0 sec, MPIA = 0.005 sec

Why AntNet Outperforms its Competitors ?

- AntNet is the only algorithm exploring the network concurrently with the data flow.
- Local information is organized in a less critical, much robust way than its competitors.
- Probabilistic routing tables give a built-in exploration mechanism and redistribute data traffic without generating counterproductive oscillations.
- Node local estimates are not directly propagate to other nodes, as done by all the competitors. AntNet is more robust with respect to locally wrong estimates.
- AntNet experimentally shown to be robust with respect the routing table updating frequency. This is a critical aspect for most of the competitors.

Current Limitations of the AntNet Approach

- More tests are necessary. In particular, failure conditions and packet error situations have to be considered.
- The behavior of the routing system has to be studied jointly with a congestion control algorithm, like the TCP.
- Probably the congestion control algorithm should be "adapted" to really well match the AntNet's behavior (AntNet-FS).
- Network managers do not like probabilistic routing tables.
- The AntNet multi-path routing can induce an additional computational overhead because of packet reordering.
- The ant frequency should be made adaptive, to follow traffic patterns while reducing, at the same time, routing overhead.

Extending AntNet-FA for Connection-Oriented Routing

- The AntNet and AntNet-FA structure is very flexible. They can be easily adapted for routing in a variety of connection-less and connection-oriented networks with different requirements.
- Flying Ants can be used by an appropriate setup procedure to establish virtual circuits.
- The AntNet routing table structure (as in distance-vector algorithms) does not allow source routing to assign a virtual circuit.
- A connection can be conveniently split over concurrent multiple paths.

AntNet-FA $\rightarrow$ AntNet-FS

(AntNet-FA for Fair-Share Connection-Oriented networks)

Routing in High-Speed Connection-Oriented Networks

- In modern connection-oriented high-speed networks (ATM) two types of integrated services are available and will coexist:
  - Quality-of-Service (QoS) services: need physical resource reservation (e.g., real-time audio/video streams).
  - Best-effort services: resources are allocated following a Fair-Share framework trying to maximize performance. No guarantees on the provided service. No physical resource reservation (e.g., web browsing, ftp, I/O for scientific computations, client-server exchanges).
- A state information is maintained at each node about all the virtual circuits using the node links.
- Algorithms for Fair-Share best-effort services can be very easily converted to manage the more strict requirements of QoS services.

AntNet-FS : Enhanced Flying Ants

- Two population of ants are used in AntNet-FS :
  - One of them behaves exactly the same as in AntNet-FA .
  - Ants in the other one:
    1. are triggered by the arrival of a new session,
    2. completely support the setup phase
    3. have the same basic functionalities as ants in AntNet-FA .

AntNet-FS : The Setup of a New Connection

- When a new connection arrives a Forward Setup Ant is launched towards the connection destination node.
- At each node IF more than a path shows an high desirability (high probability) the Setup Ant forks.
- A limit in the forking number is set to avoid a proliferation of Setup Ants that could follow the same or very superimposing paths.
- The first Setup Ant arriving at the destination node leaves on this node the information about:
  - The trip time T_{best} associate to its whole path.
  - The identifiers of the nodes on the path.
- After the first Setup Ant the connection virtually starts. The Backward Setup Ant gets from the routing components of each visited node the maximum link bandwidth the connection can have at the current time. The bandwidth is computed following a Fair-Share scheme.

- Node-by-node the Backward Setup Ant (i) computes the minimum Bw_{min} among all the so far read bandwidths and (ii) allocates a new virtual circuit with Bw_{min} "suggested" bandwidth.
- Forking Setup Ants arriving at the connection destination node after the first Setup Ant contracts the allocation of the path they discovered:
  - The path is not accepted IF: their trip time is poor with respect T_{best} OR their path superimpose too much already accepted path for the same connection.
  - Otherwise, the path is accepted and they behave like the first arrived Setup Ant.
- No ants resources are wasted: Setup Ants update Routing Tables and Local Models as any other ant.
- When a first Backward Setup Ant arrives back at the connection source node, if the total elapsed time is less than a predefined timeout threshold, the connection is activated. The connection flow is restricted by the max-min bandwidth Bw_{min}.
- Subsequent Backward Setup Ants split the

connection traffic among multiple paths and the connection flow is increased accordingly.

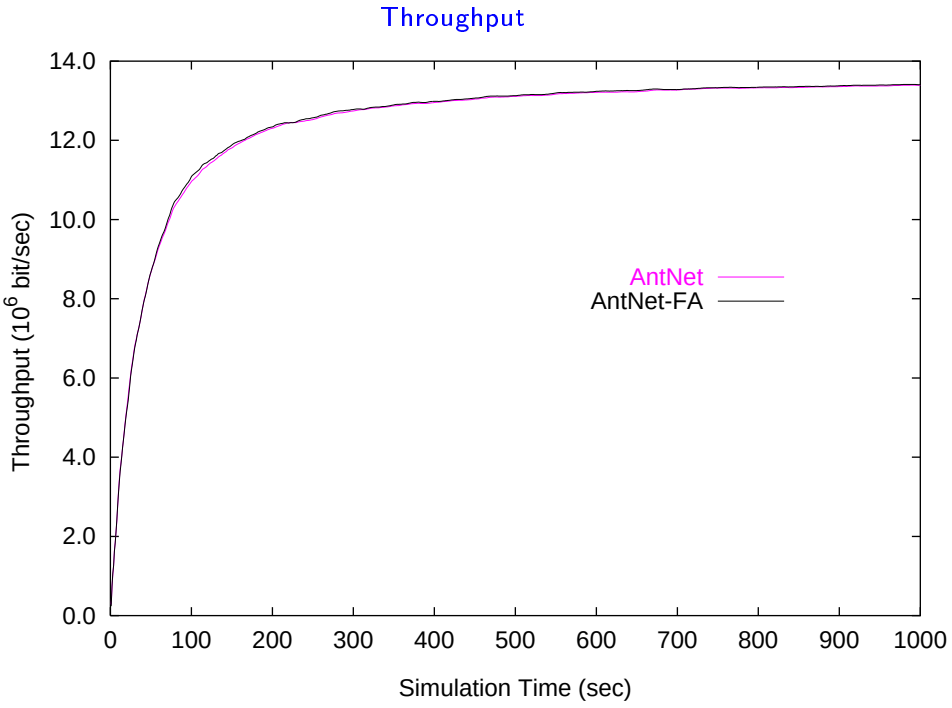
- At regular intervals the connection sends a control packet over the allocated paths to update the path-allowed bandwidth.
- "Normal" Flying Ants act in background as usual, and a Crunkback Ant can be used to explore the possibility to re-route the connection's paths in case of congestion or link/node failures.

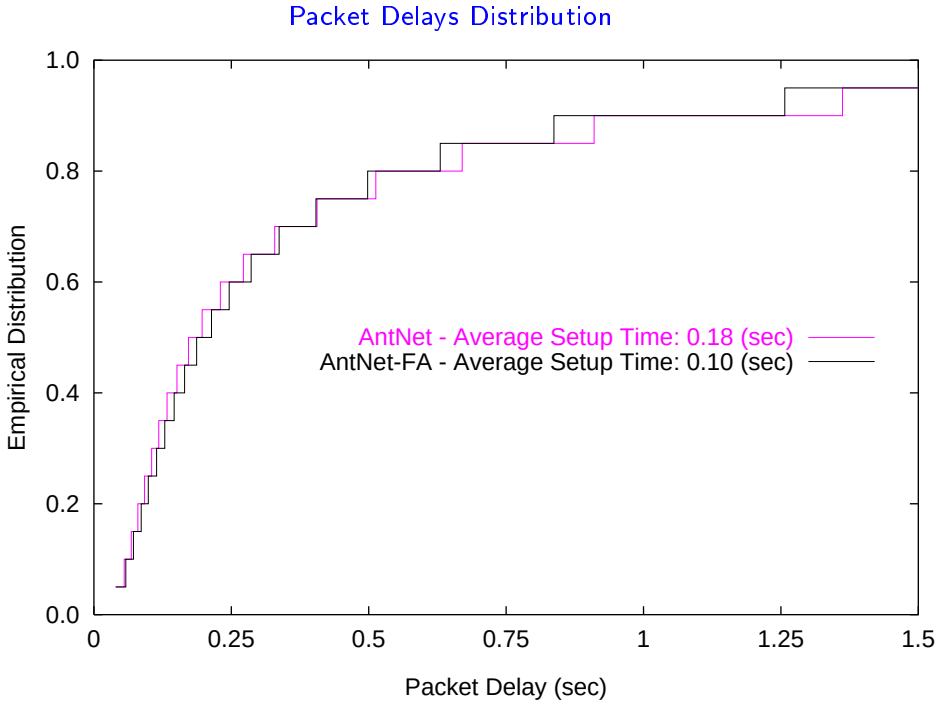
Some Characteristics of AntNet-FS

- Physically reserving resources and adapting the node routing component AntNet-FS can be used to provide QoS.
- AntNet-FS is more than a single routing system: is an adaptive routing and re-routing system AND a flow and congestion control system.
- Packet re-ordering is made easier by the use of virtual circuits and an appropriate use of the multiple paths can be done to reduce the re-ordering and error-control tasks.
- The system needs more work, study and testing.

What cannot be said should be left silent.
(Wittgenstein)

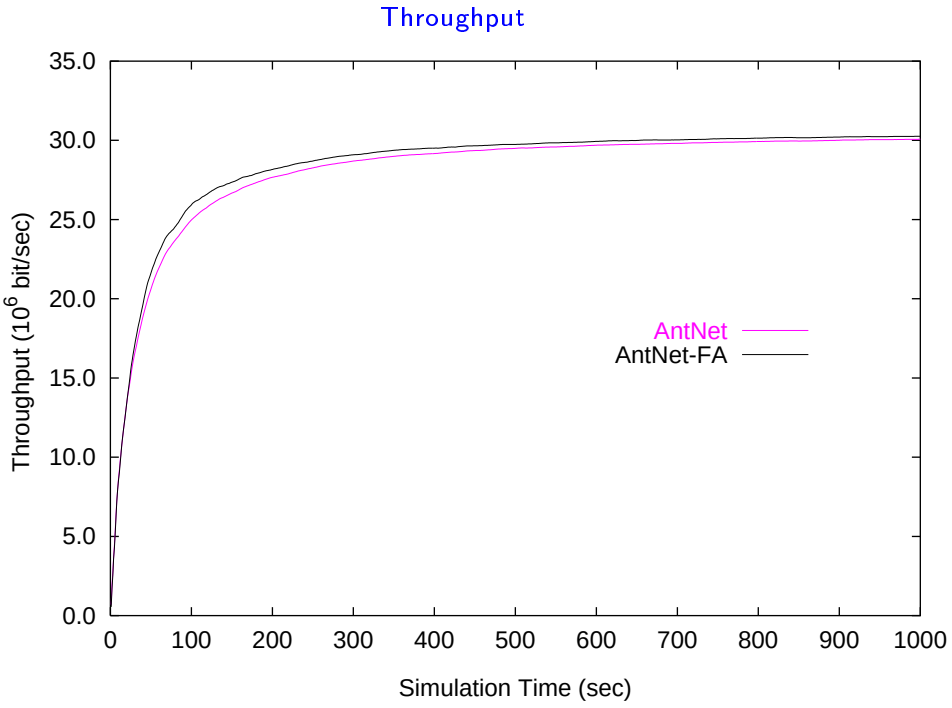
100-Nodes RandomNets - VC Traffic

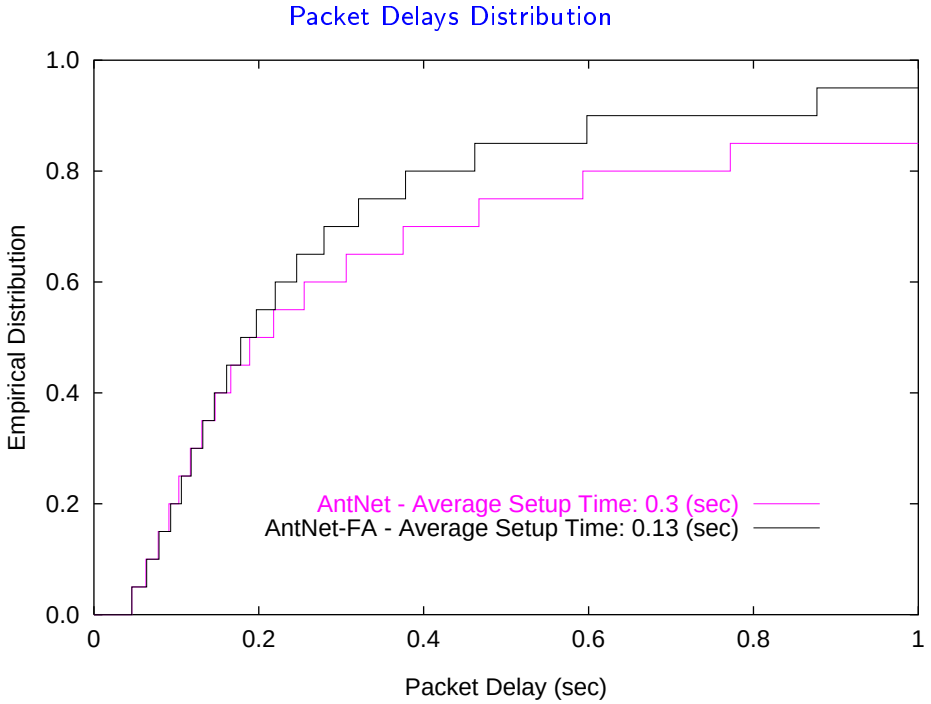




- - Simulation Parameters: MSIA = 15.0 sec, MPIA = 0.005 sec

150-Nodes RandomNets - VC Traffic





- - Simulation Parameters: MSIA = 10.0 sec, MPIA = 0.005 sec