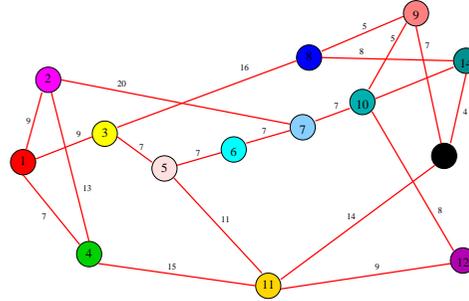


Extending AntNet for Best Effort Quality-of-Service Routing

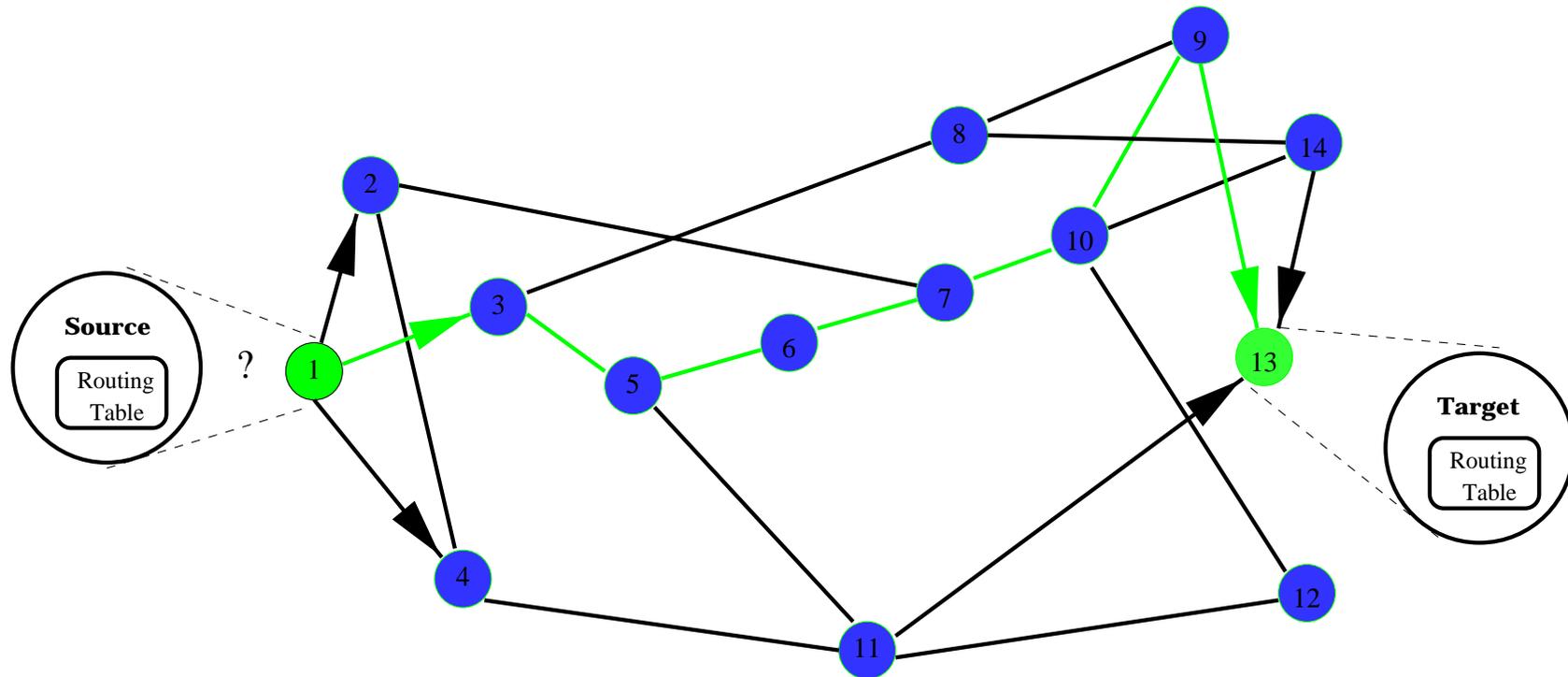


Gianni Di Caro and Marco Dorigo
IRIDIA, Université Libre de Bruxelles - Belgium

- ❖ 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

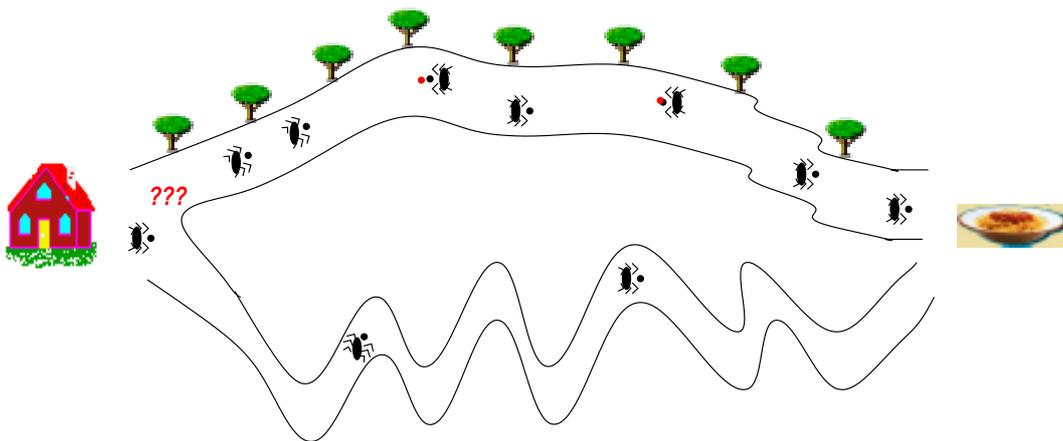
Ant Colony Optimization (ACO): Background

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 → *Stigmergy*,
 - ◆ the whole colony shows an *Autocatalytic* behavior.

Ant Colony Optimization (ACO)

■ The real ants' behavior inspired many algorithms for discrete optimization (1991 ... 1998) →
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)

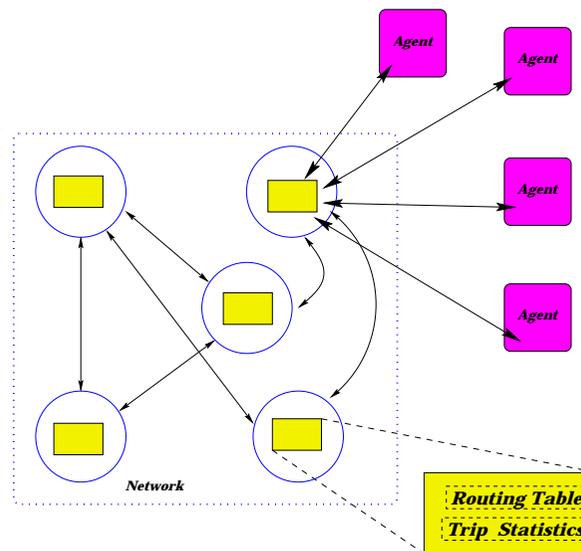
AntNet Meta-Description

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

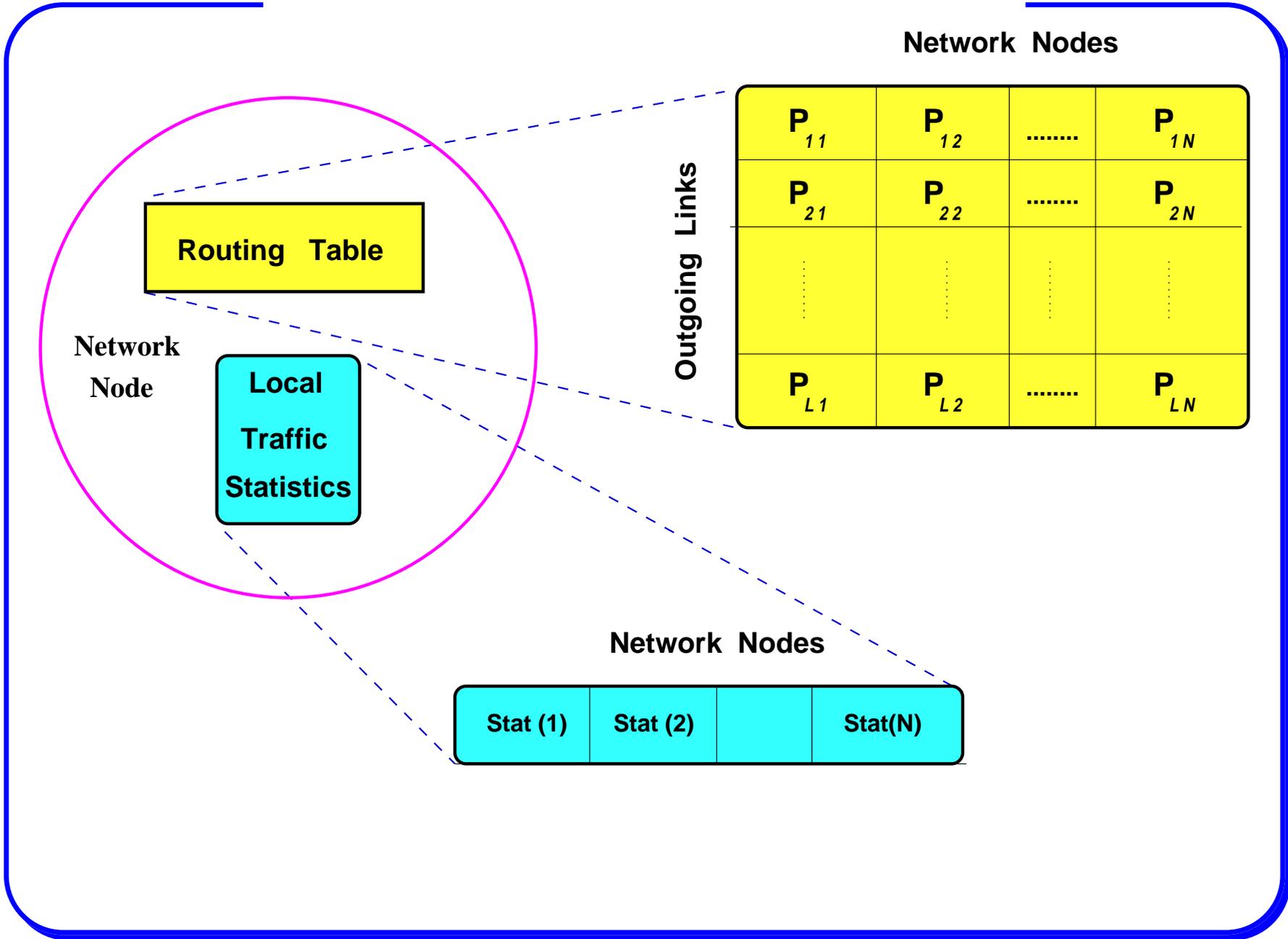
- ❖ The global routing problem cannot be efficiently solved by a single ant (**intrinsically distributed problem**).
- ❖ Good quality solutions emerge as the result of a **collective process**.
- ❖ There is **no need** for explicit **coordination** and **synchronization**.
- ❖ Agents communicate indirectly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



AntNet

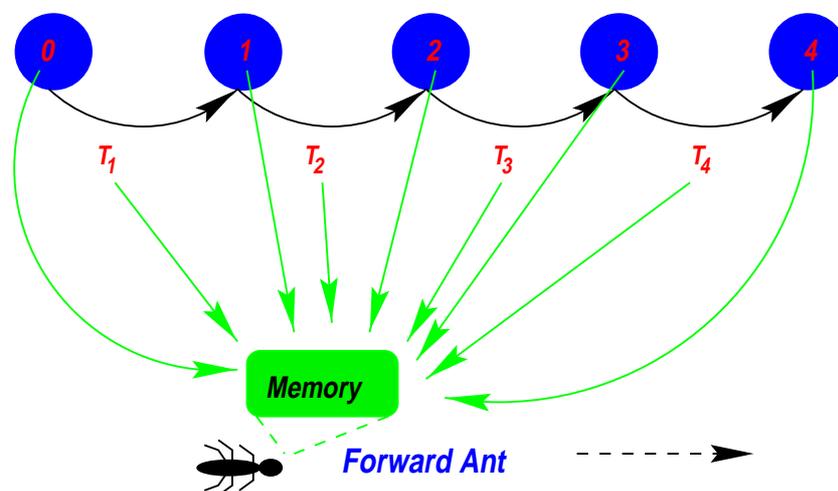
- AntNet is a distributed algorithm for best-effort adaptive routing in connection-less networks (Internet!).
- The algorithm can be used for connection-oriented routing with suitable extensions → AntNet-FS under developing.

AntNet: Node Data Structures



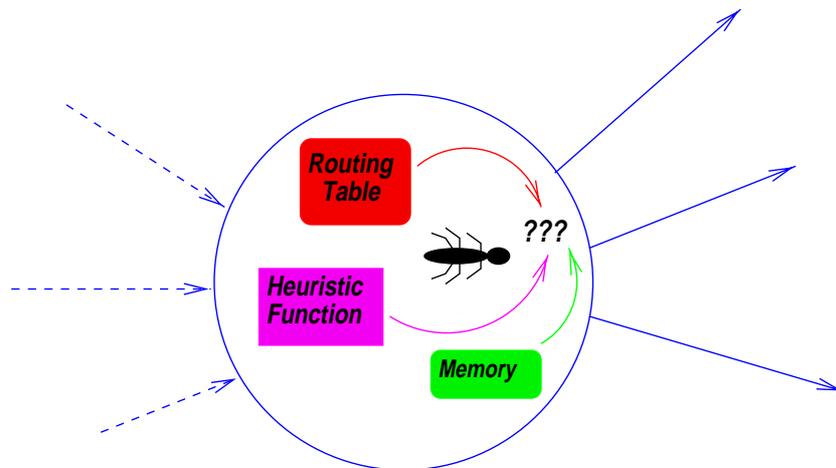
AntNet: Algorithm Description

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.

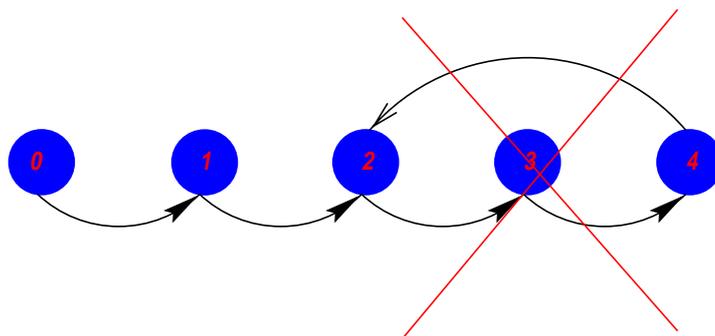


4. Next hop nodes are selected using:

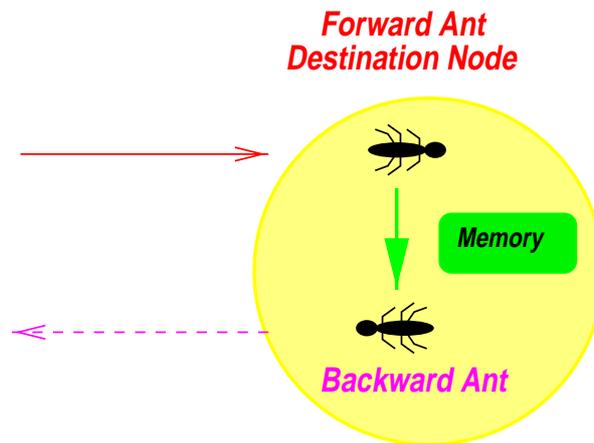
- ❖ probabilistic Routing Tables,
- ❖ local heuristic based on the state of the local link queues,
- ❖ memory of the past visited nodes



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



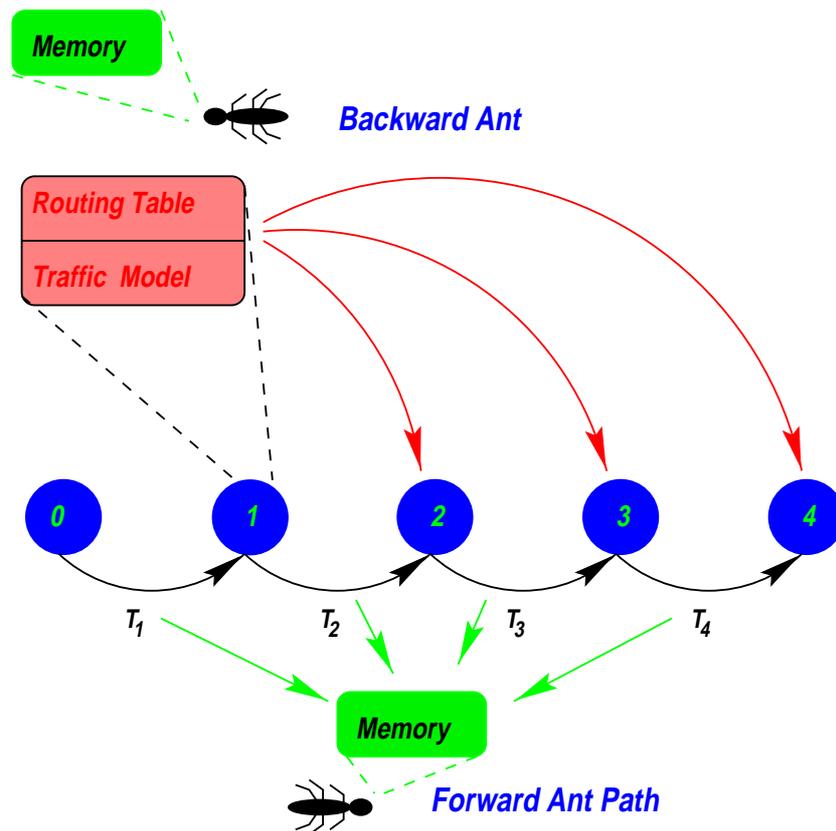
6. Reached the destination node, the Forward Ant generates a **Backward Ant**, transfers to it all its memory and dies.



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:

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

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

$q_n =$ 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:

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

$o_{k \rightarrow d}$ 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:

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

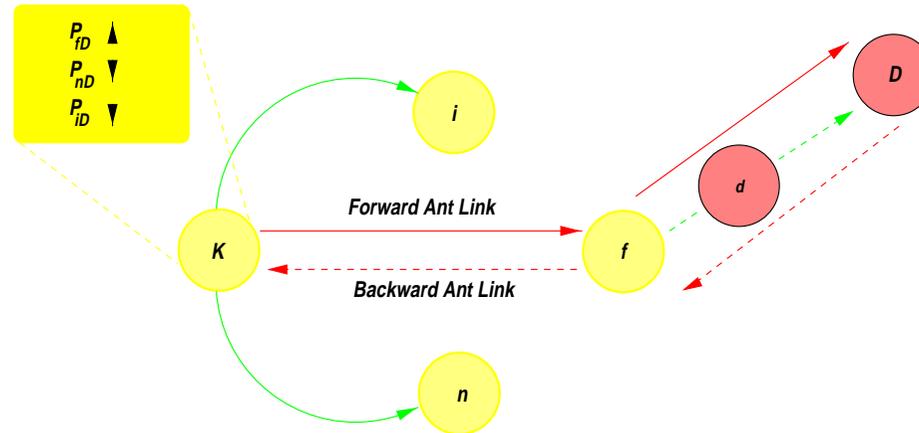
W_{best} is the best trip time experienced by the ants traveling toward the destination d , over the last observation window \mathcal{W} .

I s 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**:

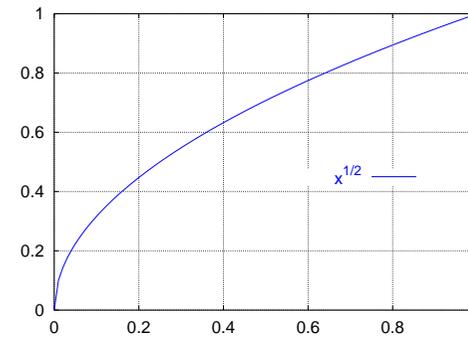
$$P_{Df} \leftarrow P_{Df} + R(1 - P_{Df})$$

- ❖ 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 → **cumulative reinforcement**.

Algorithm Complexity Vs. Performance

- ❖ The algorithm 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.