The Cross Entropy Ant System for Network Path Management

POUL E. HEEGAARD, BJARNE E. HELVIK, OTTO J. WITTNER

Poul E. Heegaard is Associate Professor at NTNU and Senior Research Scientist in Telenor R&I

Bjarne E. Helvik is Professor at the Norwegian University of Science and Technology

Otto J. Wittner is a PostDoc at Q2S at the Norwegian University of Science and Technology

Finding paths between nodes is a basic enabling functionality in a communication network. At first glance, this may seem to be a trivial task. However, finding a path when no global information is available, is a challenge. Furthermore, paths should ensure an overall good utilisation of network resources, providing low delays and losses as well as the needed capacity between nodes. Paths should be altered as the network load and topology are changed, and paths should rapidly be recovered when network elements fail. The path management function has throughout the history of communication networks been designed to meet the prime requirement of the service provided by the network within what was technologically feasible. The future network will provide a multitude of services with, to some degree, conflicting requirements. At the same time inherent robustness and autonomy of network operation are of increasing importance. This invites new approaches relative to those used in the traditional communication network and the Internet. One such approach is to use swarm intelligence, where mobile agents explore, map and manage the network in a manner similar to the way insects, e.g. ants and bees, deal with their environment.

With the above in mind, we have developed a distributed, robust and adaptive swarm intelligence system for dealing with path management in communication networks. The system is called the Cross Entropy Ant System (CEAS), and is based on increasing the probability of finding a (near) optimal solution by an increasingly focused random search. As a background for the system, this paper gives a brief discussion on path finding challenges and trade-offs. Following up is a description of CEAS where its robustness and adaptivity are demonstrated on a variety of case studies using different management strategies, like: shared backup path protection (SBPP), p-cycles, resource search under QoS constraints and adaptive paths with stochastic routing. This paper also includes a description of a running implementation of CEAS based on small home routers. The implementation demonstrates and visualises the inner workings of the method.

1 Introduction

Being able to transfer addressed information between sources and destinations is the prime function of a communication network. Hence, how to find paths for the data flow between source and destinations through the network is one of the most salient issues and important functions in network architecture and operation. In this paper, the function is denoted *path finding*, irrespective of whether physically or virtually circuit switched paths (or circuits) are found, or stable routes for connectionless forwarding are obtained.

Throughout history, the path finding applied is a trade-off between requirements of the network service and available technology. The early POTS[1)] networks had a hierarchical routing scheme, which gradually evolved into a more non-hierarchical and adaptive schemes, see for instance [1]. In the early Internet, the prime objective was to have a routing scheme which was inherently robust and would find a path between source and destination irrespective of the current topology. This scheme has evolved to a range of routing protocols, see for instance [2]. The ability to deal with failures of network elements has always been an important issue, with schemes ranging from 1+1 protection of the physical circuits to elaborate end-to-end restoration schemes, as described in for instance [3,4].

A common characteristic of the state of art schemes, is that they apply some degree of preplanning, e.g. allocation of link weighs to links in OSPF[2)] and IS-IS[3)], introduction of operator policies in BGP[4)], planning of (G)MPLS[5)] shared protection paths. For a

1) *Plain Old Telephony Service*

2) *The Open Shortest Path First (OSPF) protocol is a hierarchical for routing in an Internet domain, using a link-state in the individual areas that make up the hierarchy. A computation based on Dijkstra's algorithm is used to calculate the shortest path tree inside each area. See IETF RFC 2328.*

3) *IS-IS is like OSPF a protocol for routing in an Internet domain, based on Dijkstra's algorithm, standardised as ISO10589. See IETF RFC 1195.*

4) *The Border Gateway Protocol (BGP) is the routing protocol between the domains (Autonomous Systems – AS) of the Internet. It is a path vector protocol and makes routing decisions based on path, network policies and/or rule-sets. See IETF RFC 4271.*

 ISSN 0085-7130

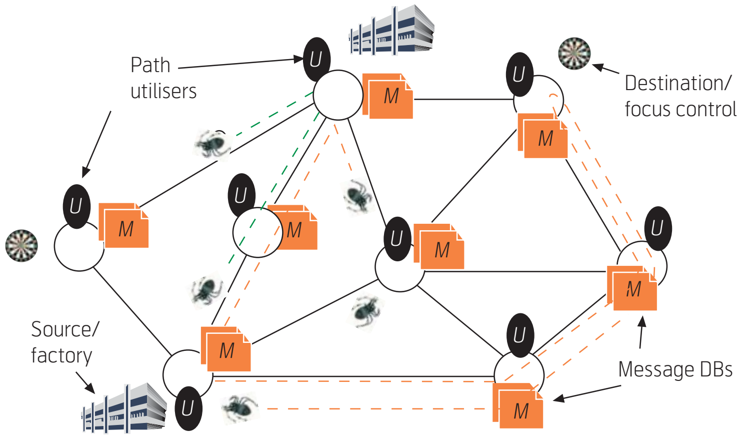


Figure 1 Virtual path management by ants

truly autonomous system, it is necessary that the path finding system itself discovers the network resources, changes in their operational status, their loading, etc. without reliance of a system external planning entity. This activity, as well as operation of the path selection should ideally be inherently robust and distributed.

Furthermore, vertically integrated networks, like for instance the PSTN[6)], were designed primarily to meet the requirements of one service. This is also reflected in the path management. The coming integrated multi-service network will transport services with highly differing QoS requirements with respect to *timeliness and performance* (delay, jitter, transfer rate, loss) as well as with respect to *trustworthiness* (availability, continuity, integrity, confidentiality, manageability). This poses a number of challenges on the path finding with respect to load sharing and QoS oriented routing [5], resilience differentiation [6], etc.

Current path finding schemes have problems in meeting these requirements, and there are ongoing research towards extending them or providing "add-on functionality". However, the self-management/ autonomy objective seems difficult to meet. For these reasons, emergent behaviour is investigated as a viable approach to routing and path finding in future networks. The approach adopted by us is inspired by the behaviour of ants. It is based on mobile agents[7)] representing ants swarming through the network. The idea is to let ants iteratively search for paths from a source to a destination node in a network. When a path is found the ant returns to the source on the reversed path and leaves markings, denoted pheromones (resembling the chemicals left by real ants during ant trail development), on every intermediate node. The strength of the pheromones depends on the quality of the path found. The subsequent searching ants stochastically select their next hops based on the current pheromone distribution. The overall process converges quickly toward a near optimal path. See Figure 1 for an illustration. The paper gives an outline of approach and summarises and discusses our findings.

The issue of path finding in communication networks and some background material on swarm based routing is elaborated further in Section 2. For dealing with path management in communication networks we have developed CEAS (cross-entropy ants system) which is based on Rubinstein's method for stochastic optimisation [7]. The theoretical background for CEAS and the principle of application are presented in Section 3. The CEAS basic technology is applied in a variety of studies using different management strategies. Section 4 presents four strategies; shared backup path protection (SBPP), p-cycles, resource search under QoS constraints, and adaptive paths with stochastic routing. For a proof of concept, and to gain experience with the implementation aspects of CEAS demonstrator systems have been made. This enables live, visualized demonstration of the inner workings of the CEAS. The current version consisting of interconnected small home routers, with a plug-and-play reconfigurable topology, are presented in Section 5. Some concluding remarks are given in Section 6.

2 Path Management

Paths between all source destination pairs in a communication network should be chosen such that an overall good utilisation of network resources is ensured, and hence high throughput, low loss and low latency achieved. At the same time the set of paths chosen must enable utilisation of the available spare capacity in the network in such a manner that a failure causes minimum disturbance of the directly affected traffic flows as well as other traffic flows in the network. The combinatorial optimisation aspects of this task are typically NP-hard, see for instance [8]. Nevertheless, considerable knowledge has been acquired for planning paths in networks. In addition

5) *(Generalized) Multi-Protocol Label Switching (G)MPLS are "Layer 2.5" protocols used to perform (virtual) circuit oriented switching in the Internet. See IETF RFC 3031 and 3945.*

6) *Public Switched Telephone Network*

7) *Mobile agents must be understood conceptually. Mobile agent technology may by used for implementation, but this has severe drawbacks with respect to security and performance. Hence, in our prototype realizations, the agents are realised by message passing between router kernels.*

to finding good paths, proper path management requires that: a) the set of operational paths should be continuously updated as the traffic load changes, b) new paths should become almost immediately available between communication nodes when established paths are affected by failures, and c) new or repaired network elements should be put into operation without unnecessary delays.

Insight and practical methods for obtaining paths *for connection oriented* networks by mathematical programming are available. For an overview, see the recently published book by Pióro and Medhi [9] and references therein. Several stochastic optimisation techniques which may be used to address these kinds of problems, have been proposed [10,11,12,7]. However, common to these are that they deal with path finding as an optimisation problem where the "solution engine" has a global overview of the problem and that the problem is unchanged until a solution is found. This differs from the requirement that path management should be truly distributed and adaptive. On the other hand, one should be aware that applying truly distributed decision-making typically yields solutions that are less fine tuned with respect to optimal resource utilisation.

Near immediate and robust fault handling advocates distributed local decision-making on how to deal with failures. This is reflected by the commonly applied protection switching schemes in today's telecommunication networks, e.g. in SDH and ATM [13,14]. Typically two (or more) disjoint paths are established, one serving as a backup for the other. Protection switching requires preplanning and is rather inflexible and not very efficient in utilising network resources.

If we turn to the *connectionless* domain; shortest path, distance vector and policy based routing as applied in the Internet, is distributed, has local decision-making and applies to some degree planning inherent in the network. See for instance [2]. However, routes (paths) are restored after a failure, which may incur a substantial delay before traffic flows along a route are fully reestablished. Furthermore, it is common that Internet operators use static link weights. This requires preplanning and lessens the adaptivity. In general, making plans that are able to cope efficiently with every combination of traffic load and network state is difficult, if at all possible.

There are two major "design axes" for management systems; a *spatial*, i.e. degree of centralisation-distribution, and a *temporal*, i.e. degree of preplanning. This is illustrated in Figure 2 where moments from the above discussion on planning for connection oriented and connectionless operation are indicated. In addition to spatial and temporal aspects, we have the degree of human involvement and control of the management. As illustrated in Figure 2, it is our research hypothesis that, relative to current approaches, self-managed path finding by emergent behaviour has the potential to provide combined advantages along both axes as well as minimise human involvement. Good resource utilisation currently obtained by centralisation and preplanning is potentially achievable, even combined with "real-time" adaptivity, inherent robustness of truly distributed schemes as well as continuity of service similar to what is today realised by preplanned "hardwired" protection. It is also an objective to overcome the trade-off, illustrated in Figure 3, between an efficient resource utilization with slow failure recoveries obtained by restoration tech-

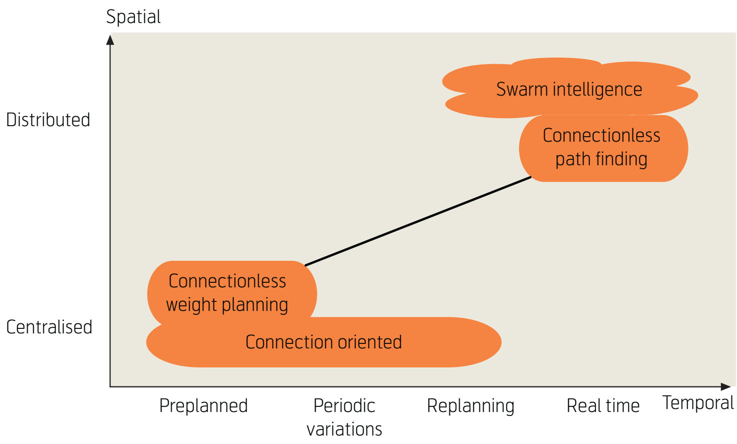


Figure 2 Illustration of the path planning activities connection oriented and connectionless networks, as well as networks managed by swarm intelligence

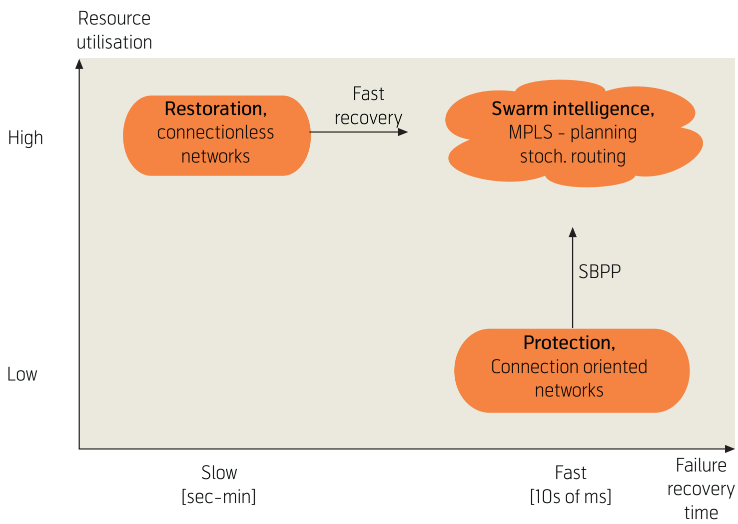


Figure 3 Approaches to achieving simultaneous fast failure recovery and high resource utilization in communication networks

ISSN 0085-7130 © Telenor ASA 2008

niques common in connectionless networks, and the short recovery times with better resource utilization obtained by protection schemes common in connection oriented networks. Overcoming this trade-off is an objective similar to what is the objective of the shared backup protection path (SBPP) techniques, e.g. [15,16,17,18], and fast recovery techniques, e.g. [19,20,21] currently heavily researched.

As mentioned above, self-management has the potential for autonomy, i.e. management without humans in the loop. A drawback is that in order to achieve autonomy, determinism is relaxed, both with respect to strict QoS guarantees and the overhead involved.

Research on swarm intelligence for path management has a ten year history. Schoonderwoerd & al. introduced the concept of using multiple agents with a behaviour inspired by ants to solve problems in telecommunication networks [22]. The concept has been pursued further by others, see for instance [23,24,25, 26,27,28] and references therein. Self-management by swarm intelligence is a candidate to meet the aforementioned requirements and to overcome some of the drawbacks of the current path and fault management strategies. In the next section we introduce the fundamentals of the technique investigated in this paper.

3 The Cross Entropy Ant System

In this chapter the virtual path management system by ants briefly introduced in Figure 1 is presented. Additional background is given in Section 3.1 before the Cross Entropy Ant System [CEAS] is introduced in Section 3.2. Performance and efficient use of network resources are important issues concerning the operation of CEAS itself. Hence, the basic CEAS is extended with mechanisms that improve its operation in these respects. These mechanisms are presented in Section 3.3. For readers interested in the inner workings of CEAS, three factboxes are added to provide more detailed insights.

3.1 Background

CEAS is based on two fundamentals; 1) the concept of emergent behaviour, and 2) the cross entropy method for stochastic optimization. Brief reviews of both fundamentals are given in the following.

Emergence is the behaviour of a system arising from a multiplicity of relatively simple interactions between system elements. This may be seen in nature, e.g. the behaviour of an insect colony emerging from the behaviour of swarms of insects. For an introduction, see for instance [29,30,31]. Emergence may be used to find solutions to optimization problems. An approach applying so-called 'swarm intelligence' is the Ant Colony Optimization (ACO) system [32]. An emergence analogy to path finding in networks is that mobile agents swarm randomly through the network from a source node to a destination node and communicate indirectly the quality of paths found by leaving messages, denoted *pheromones* (which also denotes chemical substances applied by insects to communicate), along their trail in a way similar to the foraging behaviour of ants. The use of swarm intelligence for path management was briefly reviewed at the end of Section 2.

The *Cross Entropy (CE) method for stochastic optimization* was first introduced by Rubinstein [7] and is applied for pheromone updates in CEAS. The basic notion of the method is that finding the best solution to a combinatorial optimization problem, e.g. a path in a network, by a random search has a very low probability when the search space is large, i.e. it is a *rare event*. Hence Rubinstein applies an *importance sampling* technique [33] where the random proportional rules are gradually and stepwise changed according to the importance (e.g. the cost) of the various paths found. The approach minimises the cross entropy between the random proportional rule matrices between two consecutive iterations considering the cost history. The CE-method is aimed at solving a wider range of discrete optimization problems (not only path finding). For a tutorial on the method, [34] is recommended.

With these fundamentals in mind, path management in arbitrary mesh networks is addressed. The structure of a network may formally be described as a bidirectional graph, see Fact Box 1. The overall objective is to, simultaneously and adaptively, find a set of paths between source – destination pairs in the network, minimizing the cost of the paths. Note that there may be additional requirements to the paths, for instance that paths should not contain loops/revisits

Fact Box 1 – Property 1. Fundamentals

System. A network may be represented by a bidirectional graph $G = (V,E)$, where V is the set of nodes (vertexes) and E is the set of links (edges). The links, $(i, j) \in E$, are specified by their end nodes i and j. See Figure 4 for an illustration. A path sample is denoted $\omega_{[s,d]} = \{(d, i_1), (i_1, i_2), \cdots, (i_{k-1},d)\}$, where $(i, j) \in E$ denotes the link connecting node i and j, and k is the number of hops in $\omega_{[s,d]} \in \Omega_{[s,d]}$. Here, $\Omega_{[s,d]}$ is the set of all feasible paths between s and d. The objective of path management is to find a path, or a set of paths, from source node s to destination node d, with a minimum *cost*.

Cost. The link cost is denoted $L((i, j))$. It may vary with traffic load and time, and be a function of the link attributes as well as the message database of node i. The cost of a path is additive, $L(\omega) = \sum_{\forall(i,j)\in\omega} L((i,j))$.

to nodes. Such requirements may also be more demanding. For instance, to require that a path is to form a Hamilton cycle visiting each node once[8], or require that corresponding working and protection paths must be disjoint.

The use of each link incurs a cost. The cost of a path should be the sum of the link costs along the path, cf. Fact Box 1. (Non-additive cost functions tend to create undesirable search spaces.) Hence the link cost will define the objective function of the swarm. For instance, if the link cost is the same for all links, paths with the minimum number of hops are sought; if the link cost is the measured short term average delay by using the link, the objective becomes finding the path with the shortest average end-to-end delay. Note that in the latter case, the link costs will change as the load of the network changes, hence the averaging interval applied will influence the reactivity and stability of the management network. The cost function may be compound and allow co-operation between swarms dealing with different tasks. An example: a penalty is introduced in the link cost for ants seeking a protection path if the link is likely to become a part of the working path for the same connection. Clearly the definition of the link cost plays a major role in the design of the emergent behaviour implemented by the management schemes presented in Section 4.

3.2 Cross Entropy Ant System (CEAS)

The Cross Entropy Ant System (CEAS) is illustrated in Figure 1. It was first introduced in [35] and is designed for robust and adaptive path management in communication networks. The source node of a path contains a factory that generates agents denoted (artificial) ants. Ants start their life cycles as *forward ants* searching for paths between a source and a destination node. At each step (intermediate node) along the path the next node is randomly chosen according to the random proportional rule in (1). The aggregated information in the random proportional rule is kept in *message databases* at each node as indicated in Figure 1. The paths and the behaviour of ants may also be governed by additional deterministic rules, e.g. paths without loops require that nodes are not revisited. At the destination node, the cost of a path is evaluated and a control variable, denoted *temperature*, is updated. The temperature indicates "the cooling level" of the search (cf. simulated annealing [10]), i.e. how close we are an optimal solution. From the destination *backward ants* return along the reversed path and update pheromone values in the message databases of each node visited. The better the path, the stronger the pheromone updates. The

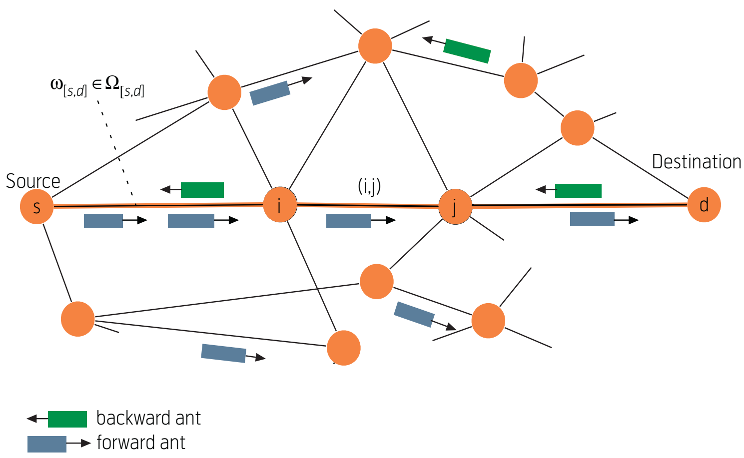


Figure 4 Graph representation of a network with forward and backward ants

corresponding change of the random proportional rule will guide future ants in their search for the same destination. A formal descriptions of the overall procedure is given in Fact Box 2.

The behaviour outlined above concerns *normal ants*. They follow the random proportional rule and maintain the paths with the best costs. In addition to normal ants, each factory generates a certain fraction of *explorer ants*. They perform random walks in the network to better detect new paths.

In Figure 1 it is shown that we may have more than one swarm searching for paths at the same time. In general, a CEAS may have an arbitrary number of species of ants as illustrated in Figure 5. Each of them deals with a single task, e.g. finding a good path

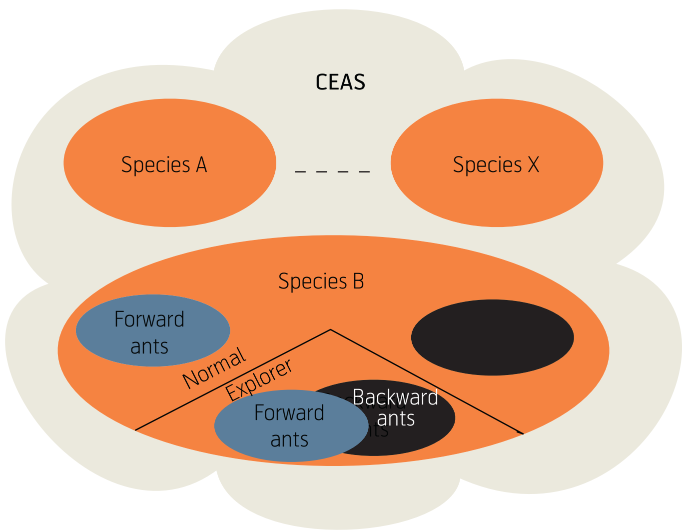


Figure 5 Classification of ants in CEAS

8) *Also known as a travelling salesman's tour. Finding a minimal such tour is an NP problem.*

ISSN 0085-7130 © Telenor ASA 2008

Fact Box 2 – Property 2. The Cross Entropy Ant System (CEAS) rules

Ant types. *Forward ants* are issued at the source s with rate λ_f, and are searching for the destination according to a random proportional rule. From the destination d, *backward ants* return on the reversed path and update the pheromone values and the random proportional rule in each node of the path. *Normal ants* follow the random proportional rule and maintain the paths with best cost, while *explorer ants* do random walks to detect new paths.

Random proportional rule. At visit t in node i, the normal forward ants sample their next hop according to a *random proportional rule* for link (i, j)[9)]

$$p_{ij,t} = \frac{\tau_{ij,t} \cdot I(j \notin U)}{\sum_{(i,l)\in E, l\notin U} \tau_{il,t}} \tag{1}$$

where $\tau_{ij,t}$ is the pheromone value of link $(i, j) \in E$ at update t, and U is a list of forbidden nodes according to deterministic rules associated with the ant. The random proportional rule matrix is denoted $p_t = [p_{ij,t}]_{||V||\times||V||}$.

Pheromone updating rule. The pheromone values are a function of the *entire history* of path cost values $L_t = \{L(\omega_1), \cdots, L(\omega_t)\}$ up to iteration update t. They are updated for *every path sample* according to (2), applying $H(L(\omega_t), \gamma_t)$, which is a *performance function* of the last cost value $L(\omega_t)$ and a control variable γ_t denoted the *temperature*.

$$\tau_{ij,t} = \sum_{k=1}^{t} I((i,j)\in\omega_k)\beta^{\sum_{x=k+1}^{t} I((i,\cdot)\in\omega_x)} H(L(\omega_k),\gamma_t) \tag{2}$$

The exponent of β is the number of ants that has updated node i at t since k, $\sum_{x=k+1}^{t} I((i,\cdot)\in\omega_x) \le t-k$ where $t - k$ is the total number of updates in the system at t since k.

Temperature update rule. The control variable γ_t, denoted the temperature, is determined by minimising γ_t subject to $H(L(\omega_t), \gamma_t) \ge \rho$ where ρ is a parameter (denoted *search focus*) close to 0 (typically 0.05 or less). In [7] a performance function recommended is $H(L_t, \gamma_t) = e^{-L_t/\gamma_t}$. To enable a continuous adjustment of γ at a small computational expense, an auto-regressive performance function $h_t(\gamma_t) = \beta h_{t-1}(\gamma_t) + (1 - \beta)H(L(\omega_t), \gamma_t)$ is applied. That gives

$$h_t(\gamma_t) \approx \frac{1-\beta}{1-\beta^t}\sum_{i=1}^{t}\beta^{t-i}e^{-\frac{L(\omega_t)}{\gamma_t}} \tag{3}$$

In [35] it is shown that the control variable γ_t is determined by minimising γ subject to $h(\gamma) \ge \rho$, which yields

$$\gamma_t = \left\{\gamma \mid \frac{1-\beta}{1-\beta^t}\sum_{i=1}^{t}\beta^{t-i}e^{-\frac{L(\omega_t)}{\gamma_t}} = \rho\right\} \tag{4}$$

To avoid excessive storage and processing demands, it is assumed that the changes in γ_t are small from one iteration to the next. This enables a (first order) Taylor expansion of (4), providing

$$\gamma_t = \frac{b_{t-1} + L(\omega_t)e^{-L(\omega_t)/\gamma_{t-1}}}{\left(1+\frac{L(\omega_t)}{\gamma_{t-1}}\right)e^{-L(\omega_t)/\gamma_{t-1}} + a_{t-1} - \rho\frac{1-\beta^t}{1-\beta}} \tag{5}$$

where $a_0 = b_0 = 0$ and $\gamma_0 = -L(\omega_0) / \ln \rho$, and

$$a_t = \beta\left(a_{t-1} + \left(1+\frac{L(\omega_t)}{\gamma_t}\right)e^{-\frac{L(\omega_t)}{\gamma_t}}\right)$$
$$b_t = \beta\left(b_{t-1} + L(\omega_t)e^{-\frac{L(\omega_t)}{\gamma_t}}\right)$$

The pheromone values in (2) are updated by the result of (5). Again, to reduce processing and storage requirements, (2) is reformulated by a (second order) Taylor expansion

$$\tau_{ij,t} \approx I((i,j)\in\omega_t)e^{-\frac{L(\omega_t)}{\gamma_t}} + A_{ij} + \begin{cases} -\frac{B_{ij}}{\gamma_t} + \frac{C_{ij}}{\gamma_t^2} & \frac{1}{\gamma_t} < \frac{B_{ij}}{2C_{ij}} \\ -\frac{B_{ij}^2}{4C_{ij}} & \text{otherwise} \end{cases} \tag{6}$$

where

$$\begin{aligned} A_{ij} &\leftarrow \beta\left(A_{ij} + I((i,j)\in\omega_t)e^{-\frac{L(\omega_t)}{\gamma_t}}\left(1+\frac{L(\omega_t)}{\gamma_t}\left(1+\frac{L(\omega_t)}{2\gamma_t}\right)\right)\right) \\ B_{ij} &\leftarrow \beta\left(B_{ij} + I((i,j)\in\omega_t)e^{-\frac{L(\omega_t)}{\gamma_t}}\left(L(\omega_t)+\frac{L(\omega_t)^2}{\gamma_t}\right)\right) \\ C_{ij} &\leftarrow \beta\left(C_{ij} + I((i,j)\in\omega_t)e^{-\frac{L(\omega_t)}{\gamma_t}}\left(\frac{L(\omega_t)^2}{2}\right)\right) \end{aligned} \tag{7}$$

The initial values of (7) are $A_{ij} = B_{ij} = C_{ij} = 0$ for all $(i, j) \in E$.

In spite of the seeming complexity of the above equations, they yield a compact ant implementation with minimal storage and processing demands.

9) *In the initialisation phase the ants explore G with a uniform random proportional rule* $p_{ij,t} = 1/(N_i - 1)$; $\forall i$. N_i *is the number of neighbours to node i.*

between a source and destination pair. Co-operation between the species can be used to

- reach the overall objectives for the system through several parallel tasks, and

- increase the performance of the system (details in Section 4.5).

Different species may be independent. This is for instance the case when the network has fixed link costs and the objective is to find the shortest paths, e.g. the minimum number of hops. Note that this is similar to the common routing schemes applied in today's Internet, i.e. OSPF, IS-IS, BGP. There may be an implicit communication and co-operation between species through the performance of the network. This is for instance the case when we use measured (short term average) delay as path cost. Use of CEAS based path management enables load balancing and allows a gradual shift of load between paths. By this instabilities, which may occur with dynamic link measures in distance vector routing schemes, are avoided. Ant species may also co-operate through explicit indirect communication. In such a case, pheromones left by "alien" species, e.g. A-ants from Figure 5, will be included in the link cost function of B-ants. This enables a co-ordinated planning among the species. How the various co-operation schemes may be used to achieve management objectives are exemplified in the application discussion of Section 4.

3.3 Performance Improvements

Experiments show that CEAS is very robust. Paths are found and maintained even with a large number of lost messages (ants). CEAS can find new solutions and adapt to changes quickly. Increased ant rates speed up adaptation as long as they are well below the network capacity. If rates are very high, the processing and transmission capacity consumed by the ants run in danger of disturbing the network production capacity (forwarding of packets) and in the worst case cause routing instabilities. If ant rates are low, CEAS may react too slowly and the system will have a transient period after a change in the network conditions where routing is suboptimal or even not operational. To control and reduce overhead in terms of memory, processing and bandwidth consumption, we have made several extensions to the original CEAS. Elitism has been introduced in our *elite selection* approach improving convergence and reducing overhead in general. Furthermore, the overhead (in terms of number of ants) is reduced by two complementary extensions; self-tuning of ant rates in the source of the path, and self-tuning of rates in intermediate nodes. Finally, overhead in terms of memory consumed in each node is reduced by an extension introducing *cooperation between ant species* coming from different sources but with the same destination and cost objectives. Cooperation also improves convergence rates.

Elite selection. *Elite CEAS* [36] is introduced to speed up convergence and to reduce the overhead in terms of number of updates. For each ant that reaches its destination, the cost of the path found by the ant is compared with an elite selection level. If the cost is below the level, a backward ant is returned towards the source node updating all pheromone values along the reversed path. Otherwise, the ant is discarded and no updates of the pheromone values take place. The elite selection level is self-tuned and gradually tightened as better and better solutions are found. It finally converges to the cost of the best (optimal) path.[10)]
In [38] the elite selection is applied only on normal ants and the results show a significant improvement in the performance compared to using elite selection on both normal and exploration ants.

Self-tuning ant rates. In [39,40] we propose two different, but complementary approaches to self-tune the ant generation rates in CEAS. The approaches are applicable to the adaptive path strategy variant of CEAS [41]. In short the approaches regulate the ant rate in the source of the path and in the intermediate nodes as a reaction to changes in network conditions. Results from simulation studies in [39,40] show that the overhead is significantly reduced without sacrificing performance in terms of convergence times.

Source rate. The source of a path generates ants at a given rate to search for a specified destination. If a (near) optimal path has been found and the network is stable, the source ant rate could be very low because a very limited number of ants is required to maintain and refresh the best path(s) in the network. If a change in the network occurs, the sending rate should be increased, and decreased again when the transient effect dissolves. To detect changes in the network conditions we propose to estimate and monitor the rates of sent (forward) and received (backward) ants in the source node. When a network is stable, nearly all ants follow a path with the same cost, and rates of forward and backward ants will be (almost) equal.

10) *An extension similar to Elite CEAS has also been introduced in ACO to realize the $\mathcal{MAX}$ - $\mathcal{MIN}$ Ant System ($\mathcal{MMAS}$) [37] where convergence is sped up by increasing the exploitation of the* best sample. *In $\mathcal{MMAS}$ however, a batch oriented approach is taken. Pheromone values are updated after* m *path samples have been collected, and updates are based on only the path with the best (either among the last* m *paths or all paths up to now) cost value.*

This is because all ants that find the destination will meet the elite selection criterion and return towards the source, updating pheromone values on the reversed path. However, when not all forward ants reach the destination, or when some ants that reach the destination are discarded by elite selection, the backward ant rate will be lower than the forward ant rate. Such a rate difference indicates that it is not known what the best paths are under the current network conditions. Hence, ant rates should be tuned in proportion to the difference between the estimated forward and backward rates, see (9). Note that self-tuning of forward ant rates in source nodes will require some detection and reaction time from a change in network conditions occurs to a corresponding increase in forward ant rate takes place. Required detection time depends on the network topology, and the location and nature of the link state event. To avoid that ant rates increase infinitely, an upper limit of the forward ant sending rate is defined equal to the initial exploration rate.

Node rate. The normal behaviour of a forward ant visiting a node is to look up relevant pheromone values and apply the random proportional rule to select which node to visit next. To reduce detection and reaction times relative to what is achievable by self-tuning in the source nodes, we have proposed to enable self-tuning of ant rates also in intermediate nodes. Broadening the search for new paths by sending more forward ants when network conditions have changed is in general desirable. However, a broadened search is particularly desirable in the domain that is directly affected by the changed network conditions. Hence, local rate tuning has a potential. Two detection mechanisms that depend only on local node information are introduced; detection of link status changes (carrier/no carrier) of this node, and detection of changes in rates of forward and backward ants passing through the node. If no changes are detected this implies normal ant behaviour. If a change is detected the forwarding rate is increased by replicating the forward ants which continue with normal behaviour. This will broaden the search for the destination and produce more alternative paths in a shorter time after a change in network conditions than is possible with self-tuning in the source nodes only. An ant replica is a copy of the original forward ant (including the path history from the source to current node). An ant replica continues to search for a path applying normal ant behavior, however it will never be replicated again. Only original ants can replicate and only original ants are monitored to estimate forward ant rates in each node. The number of ant replica generated in a node is regulated by a replication forward ant rate that is proportional to the difference in rates of forward and backward ants passing through the node, to the number of outgoing links of the node, and to how important and critical a link is in the random proportional rule. The latter is quantified by means of Shannon's *entropy measure*, $E = -\Sigma_{\forall x}\, p_x \log (p_x)$ [42].

Ant species cooperation. The original CEAS was designed for finding paths for end-to-end connection and the ant species identifiers (pheromone IDs) are specified according to source-destination pairs, i.e. the end nodes of a path[11)]. Such a design does not scale well. When a large number of paths need to be found, a large set of unique pheromone values need to be stored in most nodes in the network (minimum in all nodes that are part of the best paths for each of the source-destination pairs). The situation can be improved by letting ant species with partly common interests cooperate and share pheromone values. In this section we describe two such approaches.

Overlapping resource paths and profiles. In [44] a version of CEAS is designed where an ant's objective is to find a path to a resource type using a search profile that specifies a set of required resource attributes, e.g. capabilities, delay, content, and security requirements. This means that it differs from the path finding described above where only one (or at most two) attribute is considered. Each attribute in the search profile has a cost function related to it, with corresponding temperatures and pheromone values. A forward ant will apply a random proportional rule based on a combination of pheromone values, i.e. one pheromone value for each attribute in the search profile weighted by the corresponding temperature of each attribute. Backward ants will update pheromone values correspondingly. If the search profile of two ant species share attributes then they will both read and update the pheromone values for the shared attributes (as well as for attributes not shared). Hence, in each node a set of pheromone values will exist, which is the union of all attributes contained in the different search profiles of ants that have visited the node. A significant reduction in the number of unique pheromone values required to be managed in a node is now possible, compared to storing species specific sets of pheromone values. In addition, convergence times will be reduced since more ants (of several ant species) update the same pheromone value. See [44] for formal descriptions and performance studies.

Sub-path cooperation. Recall that the general objective, and overall problem, of path management is to find a path from a source to destination node with a

---

11) *In [43] ant species identifiers are extended with primary or backup path indexes.*

Fact Box 3 – Property 3. Performance improvement

Elite selection. Let n be the number of forward ants from s to d, and $t \leq n$ the number of backward ants. ω_n^* is the path of forward ant n, and γ_n^* is the *temperature* determined by (2) using the cost values observed by all n forward ants, $L_t^* = \{L(\omega_1^*), \cdots, L(\omega_n^*)\}$. In [36] the pheromone values and the backward ant index t are updated only when the *elite selection criterion,* $L(\omega_n^*) \leq -\gamma_n^* \ln \rho$, is true. Hence, the *update path sample* ω_t is

$$\left\{ \begin{array}{c} \omega_t = \omega_n^* \\ t \leftarrow t+1 \end{array} \mid L(\omega_n^*) \leq -\gamma_n^* \ln \rho \right\}$$

The temperature in (2) and pheromone values in (6) are using the update path samples ω_t.[12] The elite selection should be applied to *normal ants only*, and let all explorer ants lead to an update, see [38] for results.

Rate estimates. The forward and backward rates can be estimated by discretization of the time axis with granularity δ (and counting the number of ant arrivals in time intervals of size δ. A running average $\hat{\lambda}$ of these rate estimates is generated applying an auto-regressive formulation

$$\hat{\lambda}_{a,m} = \alpha \hat{\lambda}_{a,m-1} + (1-\alpha) \cdot \frac{N_{a,m}}{\delta}, \quad \hat{\lambda}_0 = 0, a \in \{f, b\} \tag{8}$$

where $N_{a,m} = \| \{T_a \mid (m-1)\,\delta \leq T_a < m\delta \vee T_a \in \mathbf{T}_a\} \|$ is the number of ant arrivals in time interval m, and $\mathbf{T}_a$ is the set of ant arrival events, where $a \in \{f,b\}$ (forward and backward ants). α is a memory factor that is tuned to capture the transient effects of the network.

Source rate self-tuning. The ant generation rate λ_f varies between $\lambda_0 \geq \lambda_f \geq \lambda_s$. The self-tuning rate introduced in [39,40] provides a low rate when $\lambda_f - \lambda_b$ is small (stable system), and a high rate when it is large (system not yet converged or network conditions changed):

$$\lambda_{f,m} \leftarrow \max\left(\lambda_s, \lambda_0\left(1 - \frac{\hat{\lambda}_{b,m}}{\lambda_{f,m-1}}\right)\right), \tag{9}$$

Node rate self-tuning. In [40] forward ants are replicated. The *replication rate* λ_r is self-tuned and proportional to the difference $\lambda_f^{(i)} - \lambda_b^{(i)}$ (where $\lambda_f^{(i)}$ and $\lambda_b^{(i)}$ are the forward and backward rates in node i), to the node out-degree ν_i, and to the knowledge about preferred links quantified by the Shannon's *entropy measure*, $E = -\Sigma_{\forall x}\, p_x \log(p_x)$ [42]. Hence,

$$\hat{\lambda}_{r,m}^{(i)} = \frac{E_m^{(i)}}{E_{\max}^{(i)}}\left(\hat{\lambda}_{f,m}^{(i)} - \hat{\lambda}_{b,m}^{(i)}\right)\nu_i = \nu_i \log(\nu_i)\left(\hat{\lambda}_{f,m}^{(i)} - \hat{\lambda}_{b,m}^{(i)}\right)\sum_{(i,j)\in E} p_{ij}\log(p_{ij}) \tag{10}$$

Maximum entropy $E_{\max}^{(i)} = \log(\nu_i)$ is when all links have the same (or no) pheromone values, i.e. uniformly distributed $p_{ij} = 1/\nu_i$.

Search profile cooperation. The objective of resource path management is to find a minimum cost path applying a chain of resources of a specific type, each with some given security r, capabilities f, content c and quality q. The random proportional rule, pheromone update rule, and temperature update rule in Fact Box 2 are no longer (s,d) specific, but instead applied individually for each attribute in the (r, f, c, q)-tuple. Note that each of the source profile attributes can be a set. If ant species i and j, as defined by their (r, f, c, q)-tuple, have (partly) overlapping tuples, $(r_i, f_i, c_i, q_i) \cap (r_j, f_j, c_j, q_j) \neq \emptyset$, they will (partly) use the same random proportional rules and update the same pheromone values of the attributes they have in common.

Sub-path cooperation. The objective of path management is to find a path from source node s to destination node d, with a minimum *cost*. In sub-path management the objective is to find a minimum cost path from a node i to the destination d. In node i the random proportional rule is (i,d) specific and can be shared and updated by all ant species that visit node i on their way to destination d.

minimum cost. Such problems may be split into subproblems [45], and hence the objective of subpath management can be to find a minimum cost path from an intermediate node to the destination. In such a case an intermediate node will apply a random proportional rule given for sub-paths from the node to a set of destinations. Hence, the pheromone values, and the corresponding random proportional rules, can now be shared and updated by all ant species visiting the same intermediate node and looking for the same destination. The sub-path concept was first introduced in [45], and improved and studied in detail in [38]. Simulation results from case studies applying a small and a medium sized network show significant savings in convergence time, memory storage and ant rates required.

[12] *Note that if the elite selection criterion is not met the iteration index* t *is not updated and no path is assigned to the update sample set.*

4 Applications of CEAS

The basic CEAS technology has demonstrated its applicability through a variety of studies of different path management strategies, including; shared backup path protection (SBPP), p-cycles, resource search under QoS constraints, adaptive paths with stochastic routing, and traffic engineering of MPLS. This section provides highlights from these studies. All studies have been conducted on models of small and medium sized networks with different sets of network dynamics. Studies are conducted by simulation using ns-2[13)] or Simula/DEMOS [46,47].

4.1 Disjoint Primary-Backup Paths with Performance Guarantees

In [48] a 1:1 protection scheme is adopted, i.e. every primary path is to have an independent and disjoint backup path ready for use if a link failure occurs in the primary path. Having dedicated capacity for backup paths implies 100 % redundancy in a network which is inefficient and expensive especially when failures are rare. Hence in [48] the capacity required by a backup path is sought shared with other (non-conflicting) backup paths, i.e. a shared backup path protection scheme (SBPP) is applied. Note that finding sets of such SBPP paths is complex and resembles proven NP-complete problems like “Path with Forbidden Pairs”, “Disjoint Connecting Paths” and “Shortest Weight-Constrained Path” [49].

The SBPP version of CEAS in [48] lets each primary path and each back-up path be dealt with by a separate species of ant. The different species are made to detest each other in accordance with primary/backup optimisation criteria, i.e.

- Backup ants search for paths which are disjoint with their corresponding primary paths.
- Backup ants having overlapping corresponding primary paths search for disjoint paths.
- All ants detest other ants which represent a load that in addition to their own load may incur an overload of a link.

A novel cost function is devised to make the above behaviours emerge. The difference between the available and required capacity of all potential paths over a link is summarised for each link in a path, which results in an estimate of expected loss along the path:

$$L(\omega_m^r) = \sum_{(i,j)\in\omega_m^r} \mathcal{S}\left[a_m + \sum_{\forall ns:(,i,j)\in\omega_n^s} P_{ij}^{ns} V_i^{ns} Q_{mr}^{ns} a_n - c_{ij}\right]$$

where S[...] is a shaping function applied to smooth the search space, c_{ij} is available capacity on link (i,j), a_m and a_n are required capacity of current (m) and competing (n) paths respectively, and $P_{ij}^{ns} V_i^{ns} Q_{mr}^{ns}$ is the total weight of the required capacity of competing traffic enforcing the detestation scheme. The index r is path rank, i.e. primary or backup. Results from case studies simulating SBPP CEAS show that near optimal sets of primary and backup paths can be found efficiently. See [48] for details.

4.2 Protection Cycles

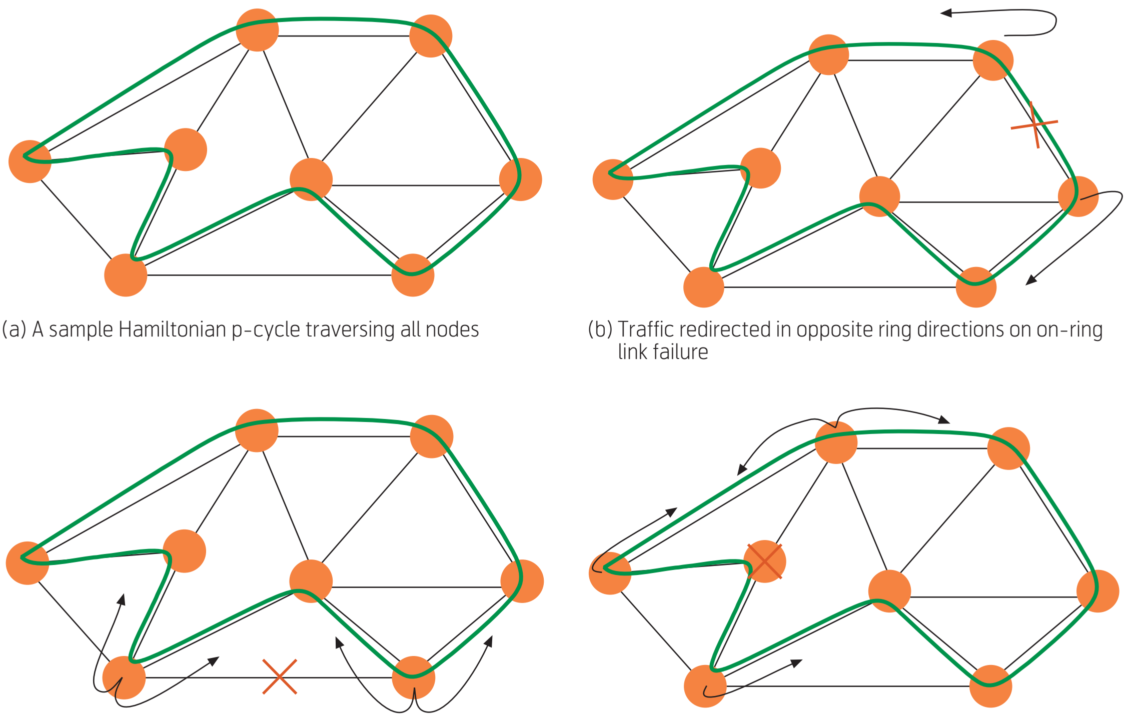
Protection cycles is a well know dependability measure and commonly applied in SDH networks with ring topologies. Applying protection cycles, known as *p-cycles* [50] in meshed networks has been shown to provide good protection against failures in both network links and nodes. Upcoming optical burst and packet switched networks will especially require node protection due to the expected longer down time of nodes compared to optical links.

A version of CEAS presented in [51] is capable of finding near optimal Hamiltonian cycles in meshed networks with respect to the amount of spare capacity on the links in the cycle. Hamiltonian cycles are good p-cycle candidates, enabling protection of all links and nodes in a network as may be seen in Figure 6. A new tabu memory as well as new cost functions are introduced in [51] to help CEAS find relevant cyclic paths. The capacity of the strongest “weakest” link of paths found is stored, shared and updated by all ants. A “weakest link” is the link with the lowest spare capacity in a path. Links with less spare capacity than the current strongest “weakest link” are tabued and hence avoided by ants during forward search. The cost functions take available capacity in both directions of links into account. See [51] for further details. Early results from simulations indicate that the new system has a promising ability to find good candidate p-cycles.

4.3 Path Management in Telenor’s IP Network

The (former) backbone topology of Telenor’s IP network has been applied in a series of simulation experiments with CEAS. The topology, as illustrated in Figure 7, consists of a core network with ten core routers (green) in a sparsely meshed topology, ring based edge networks with a total of 46 edge routers

13) *http://nsnam.isi.edu/nsnam/index.php/Main_Page*



(a) A sample Hamiltonian p-cycle traversing all nodes

(b) Traffic redirected in opposite ring directions on on-ring link failure

(c) Traffic redirected in all ring directions on stradding link failure

(d) Traffic redirected on (on-ring) node failure

Figure 6 p-cycle *protection in a meshed network*

(black), and a dual homing access network with 160 access routers (orange). The topology consists of approximately 350 links where the relative transmission capacities are 1, 1/4 and 1/16 for core, edge and access links, respectively. The service studied in this network is IP connectivity and in particular establishing virtual connections (VCs) with performance guarantees. CEAS is used to establish, maintain and monitor VCs between one or several node pairs connected to access routers (green nodes). In Figure 7 the topology is illustrated with an example VC that is established between node 74 and 164. The best route is indicated by thicker lines. A distribution of the number of hops of the shortest paths[14)] between any pair of access routes is included in a sub-figure. The average number of hops is 6.37 and the majority of paths have six or seven hops.

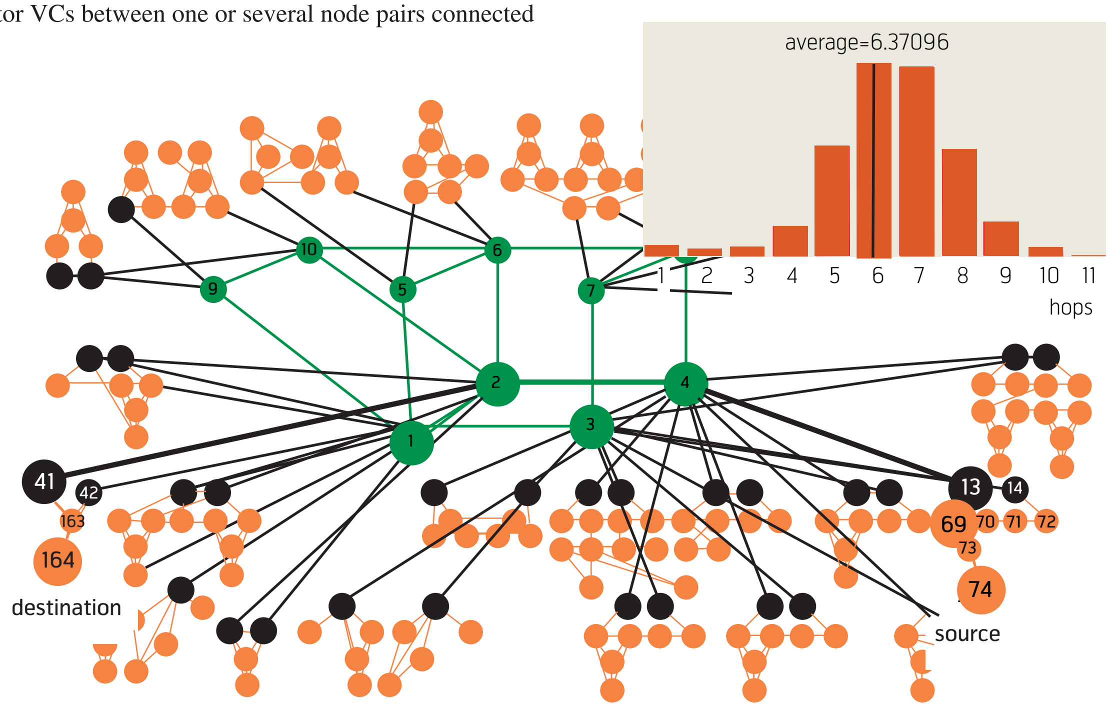


Figure 7 The simulated backbone network with 216 nodes and 350 links. The average number of hops between any pair of access nodes is 6.37. The shortest path between node 74 and 164 is given as an example

The illustrated network has be applied to demonstrate overhead reduction techniques described in Section 3.3, for studies of the effect of pheromone sharing in Section 4.5, and as a large scale example to demonstrate the use of CEAS for performance monitoring, see Section 4.6.

4.4 Load Sharing and Protection

In [52] two path management strategies are investigated. The *primary backup* scheme has as its prime objective to establish disjoint primary and backup (MPLS) paths for (all) source destination pairs. As described in Section 4.1, primary and backup paths are to be established such that backup-paths reuse network resources without preventing (due to overload) the scheme to provide continuity of service when a network element fails. The scheme's main advantage is its explicit knowledge of the immediately restorable traffic. However, its scalability with respect to increasing network sizes and advanced priority policies is not yet investigated. The other strategy investigated is the *adaptive path* scheme, which applies stochastic routing of paths for all source destination pairs in all nodes of the network. Such a scheme pro-actively provides alternative paths in case of failure. The scheme's main advantages are simplicity and fast adaption to major chances in the network. It lacks, however, the ability to give explicit indication of the fraction of traffic that will experience continuity of service. Differentiation or priority is also difficult to provide.

The network presented in Section 4.3 has been applied for investigating the operation of the *primary backup* and the *adaptive path* strategies under dynamic network conditions. Aspects of establishing and managing 10 separate paths are studied in detail. Paths are exposed to network link failures, drops of management information, and changes in offered traffic loads, see Table 1 for a summary of the simulated changes. The objective is to study the transient behaviour, i.e. the adaptivity and robustness, of this distributed management approach. For a more comprehensive discussion, see [52].

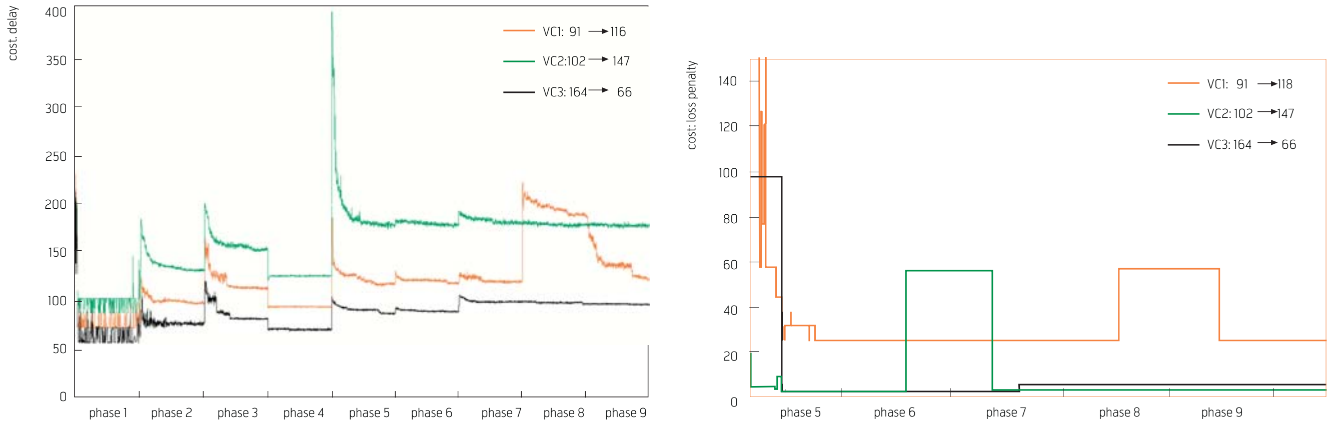
The results presented in Figure 8(a) shows the cost (e.g. delay) as a function of time. All results are based on ten simulation replications. Observe that the adaptive path strategy quickly switches to an alternative path on excessive load or link failure, and almost immediately back to the original path when load decreases or link is restored. In [52] it is also observed that the adaptive path strategy will distribute the load among paths with equal cost (e.g. delay) because a path is randomly selected according to the relative pheromone values that again are determined by the cost values.

Figure 8(b) shows the results from simulation of the primary-backup strategy for the same scenario. A switch-over from a disconnected operational path to an alternative path, either by protection switching (primary to backup) or by restoration (primary to a new primary), will cause an interruption of service. Observe for example the behaviour of VC2. After the core link failure at the beginning of phase 6, the primary path of VC2 is disconnected and VC2 is broken (regarded as down time). Explicit link failure notification will improve the path availability by making the protection switching mechanism more reactive.

Phase	Average load, ρ	Link events	Comments
-	0	-	Exploration phase
1	0	-	Initial topology
2	0.3	-	Increased load
3	0.6	-	Increased load
4	0.3	-	Decreased load
5	0.9	-	Sign. increase in load
6	0.9	Down [4,8], [6,8], [1,2]	Core links failed
7	0.9	Down [3,20], [1,42], [7,55], [3,22]	Edge links failed
8	0.9	Down [19,86]	Access link failed
9	0.9	Restored [19,86]	Access link restored

Table 1 Dynamic scenario for testing of adaptivity

14) *Determined by Dijkstra's algorithm assuming static link costs.*



(a) Adaptive path strategy

(b) Primary-backup strategy

Figure 8 Comparison of management strategies in dynamic environment

4.5 Path Finding with Partly Overlapping Paths

In contrast to existing resource localization mechanisms for peer-to-peer systems a version of CEAS is presented in [44] which determines the path quality and cost by considering all resources involved, including peering and (client and server) middelware and network resource. Two types of profiles with QoS parameters are introduced. A *user request profile* specifies requirement to different resource-types relevant for a search. During a search the user request profiles are matched against *resource profiles* provided by resources visited. A resource profile indicates capabilities of a resource. Profile matching results in QoS loss vectors, and a set of such vectors finally generates a path cost vector as well as an overall path cost. The path cost vector (together with a temperature vector) is applied by backward ants such that a pheromone value for each QoS parameter is updated in all visited resources.

The CEAS version of [44] enhances the scalability by enabling cooperation between ants when they, fully or partly, have overlapping user request profiles. The total number of relevant unique QoS parameters $|\Xi|$ will be limited, hence a limited number of unique pheromones is required. However the total number of possible unique profiles $N_{\bar{\xi}}$, will still be large since

$$N_{\bar{\xi}} = 2^{|\Xi|} - 1$$

e.g. to enable a total of N, = 10^{100} different profiles only $|\Xi| \approx 333$ unique pheromones are required.[15)]

Cooperative behavior due to overlapping profiles will also increase general performance since more ants update the same pheromones (especially in popular resources). Results from simulations are promising and show that a set of near optimal resource paths conforming to a set of different but overlapping user request profiles may be found with improved performance.

Scalability as a result of cooperation between ants when paths overlap may also be achieved for virtual connections (VC) between specific source and destinations. In [45] extensions and changes to CEAS required to enable such cooperation are described and initial simulations conducted. The extended CEAS lets all VCs with the same destination, typically with different sources, update the same pheromones in every shared node along the route to the destination. Cost is recorded, and corresponding pheromones are updated, *from the shared node to the destination* independent of the original source of the ant that visits this node on its way to its destination. Hence, ant species are identified by their shared node (the new source) and destination node, and not by their source (origin) and destination nodes as in the original CEAS strategy described in Section 3. As an illustration consider Figure 9 where two different ant species search for the same destination resource. They share the sub-path from node 5 to the destination resource and have different sub-paths from the blue nest to node 5 and from the red nest to node 5. The figure is taken from [38]. While pheromones are better utilised, more temperature calculations per backward ant are required since each sub-path in a path must be considered separately when costs and temperatures are handled.

The effect of pheromone sharing has been tested on small network topologies with some network dynamics, e.g. with link failures as introduced in Figure 10.

15) *Deterministic requirements and binary matching of QoS parameters (loss or no loss) are assumed. By introducing non-deterministic requirements, i.e. having weighted loss output from a parameter matching, the profile space becomes even richer/larger.*

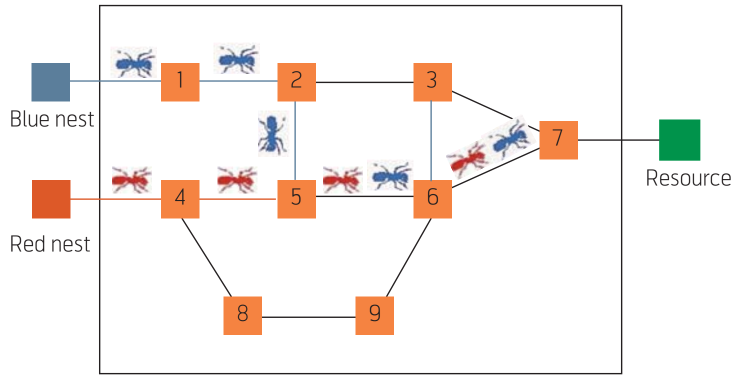


Figure 9 Two different ant species searching for the same destination (resource). In this example they share the sub-path from node 5 to the resource and have different sub-paths from the blue nest to node 5 and from the red nest to node 5. Figure obtained from [38]

The original source-destination and the new shared node-destination approaches are compared.

Results in Figure 10 show how the temperature for a VC changes as a function of iteration *t*. It can be observed that the convergence rate is improved without increasing the ant rate and with a significant reduction of the node storage demand with pheromone sharing compared to no sharing. It is also evident that the restoration time is significantly improved, and the failure detection is at least as good as in the shared strategy.

In [38] the shared pheromone strategy is studied in more detail and alternatives for cooperation are investigated. Several simulations have been conducted and demonstrate a significant effect of cooperation between ants with common interests. The simulation results using the Telenor example in Section 4.3 show that by introducing cooperation and reducing ant rates the number of temperature calculations may be kept at the same level as for original CEAS while overhead in terms of ant packets is reduced by 77 %.

4.6 Performance Monitoring of Path Quality

Monitoring of the quality of service is essential in the establishment and management of virtual connections. The ant system could be considered as a monitoring system. Several potential candidates for monitoring indices in an ant-based routing system are considered in [53] and the most promising with respect to detecting significant changes in the network conditions are (see summary in Table 2):

- Convergence index (temperature, or the elite limit that is a function of this);
- Cost value index (path delay, or loss ratio, available bandwidth);
- Pheromone values (in nodes).

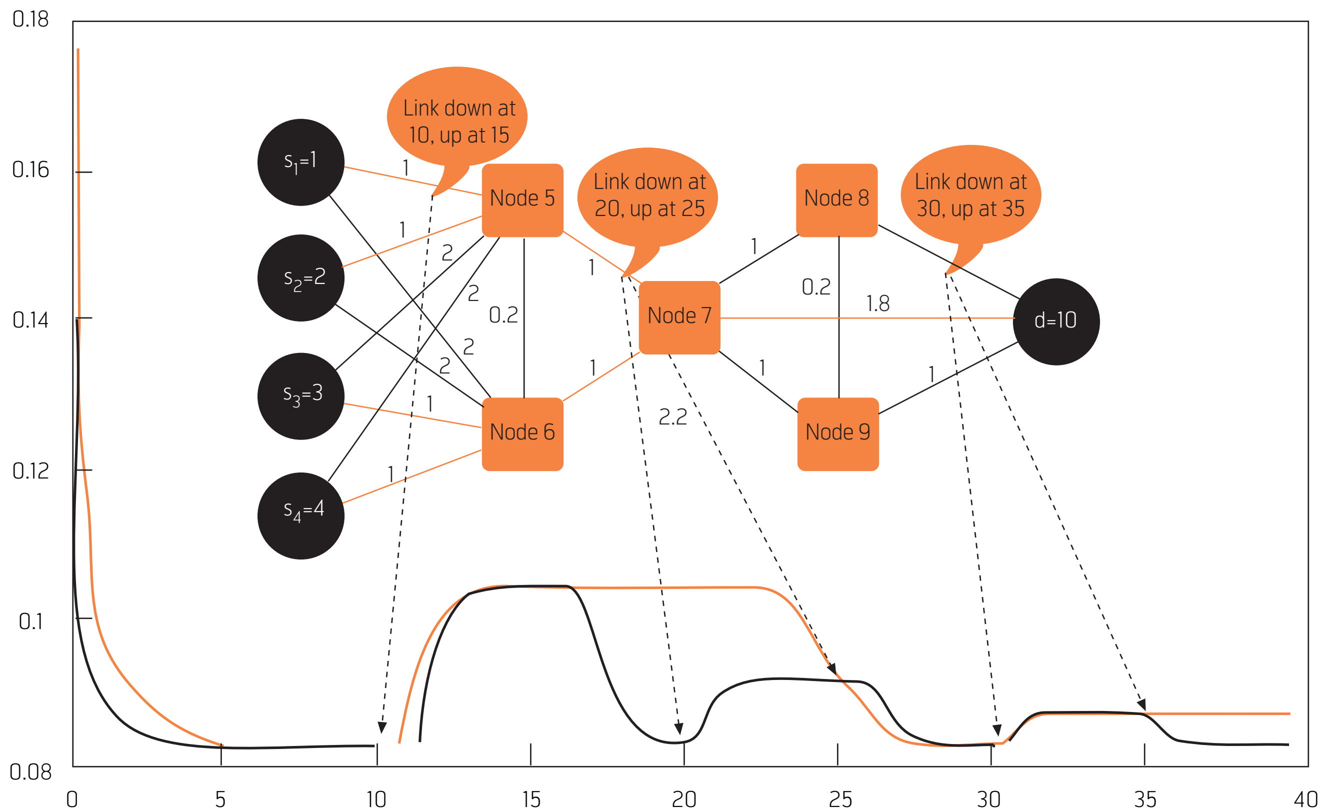


Figure 10 The temperature value for a VC as a function of iteration t in a simulation case with topology and network dynamics as indicated

ISSN 0085-7130 © Telenor ASA 2008

Metric	Observations	"Health"	Alarms
Ant route table	Deviation from data routing table	Misconfiguration in routing, interface overload	Significant deviation (in time or space)
Pheromone values	Increase by *x* % in *n* sec.	Node/link/path down	Check configuration
Convergence index	Decrease by *x* % in *n* sec.	New node/link/path discovered	A lower delay path for the VC exists
Cost value index	Average over *n* sec. decreased by x% last minute	After effect of change in network (still exploration)	None
Path probability	Close to max. for last minute	Stable networks	None

Table 2 Examples of use of CAS indices (from [53])

As an example, the cost (delay) of multi-VC connections is observed by simulation in *ns-2*. The simulator has implemented a Link state routing protocol that emulates the OSPF routing behaviour in an inter-domain and compares this with the CEAS behaviour. The network dynamics introduced in the simulations include node and link failure, and the variations in the cost and pheromone values in each node are observed.

In [54] it is observed that changes in the network topology are easily observed both by AntPing (probe packets routed by pheromones) and Ping (probe packets routed by link state tables) by use of time plot of cost and elite limit values. In Figure 11 is given an example. The plot includes observed cost values for the ants that return to the nest (AntPing: cost), the cost values for *all* ants reaching the destination (AntPing: costall), the elite limit that determines whether the VC should be updated or not (AntPing: elitelimit), and finally the one-way delay from the source to the destination observed by Ping packets. Observe that at time 50 a change occurs affecting the VC. At that point in time the change in delay of AntPing and Ping are different, hence they follow different paths. The reason is that the link state routing uses static cost values, while AntPing is sensitive to changes in link delays. If the cost metrics are not consistently set to reflect the (expected or observed) delays, the routing of AntPing and Ping might end up following different routes.

As an alternative to observations in end systems some information and indications of changes can be obtained by observing the pheromone values of the CEAS in the intermediate nodes. In Figure 12 the pheromone values for interface 2 and 3 in node 1 are

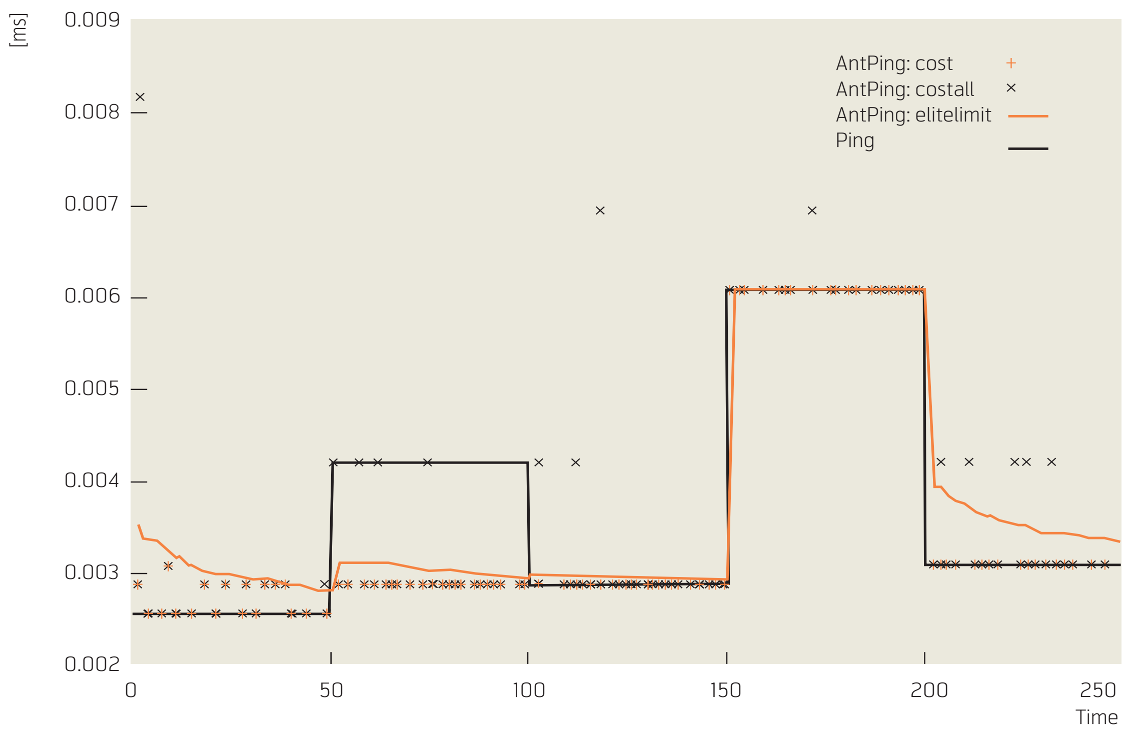


Figure 11 Cost and elite limit sample for typical time series

ISSN 0085-7130 © Telenor ASA 2008

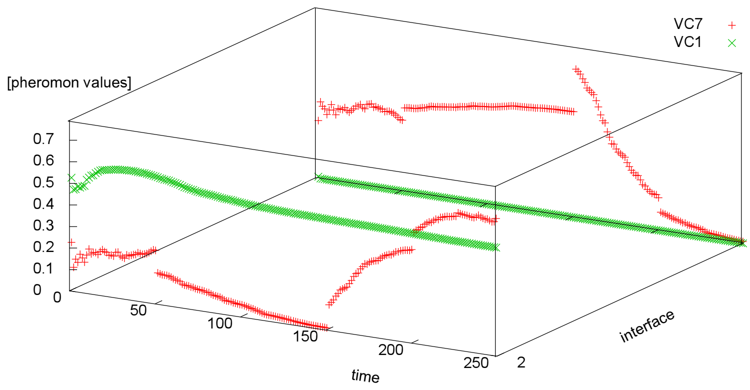


Figure 12 Pheromone values for VC3 and VC7 in node 1. It illustrates that VC7 changes preferred path at time 100 when link [2,4] fails. VC1 is not affected by the same dynamics, it has interface 2 as its preferred path for the entire period

plotted for two different VCs (denoted VC3 and VC7 in the figure). It can be observed that plotting all VCs in a node will visualize the *importance* (the number of VCs that have at least one preferred path through this node), *stability of node region* (the number of VCs that change the ant frequency and pheromone values) and *criticality* (the number of VCs with more than one preferred path through this node).

4.7 Establishing MPLS Label Switched Paths

In [55] the CEAS was applied for traffic engineering of Label Switched Paths setup by the Multi-protocol Label Switching (MPLS) protocol. CEAS is used to search, detect and monitor the best paths in the network and MPLS is applied to realize link disjoint primary and backup LSPs with specific QoS requirements. Control messages are specified to notify and change LSPs when something is changed in the network conditions.

Several strategies are studied for establishing virtual connections based on information from the underlying CEAS. The three most promising are discussed with respect to *speed of convergence* rather than globally optimal solution when network changes, and *stability* in solution (“route pinning”) rather than switching to (temporary) slightly better solution when network is stable. The three strategies are

- **Check Periodically (the CEAS pheromone values)**. This strategy checks periodically the CEAS pheromone values to determine the current best path. The *best* path is here the path with the current highest selection probability, i.e. the highest pheromone value of the underlying CEAS. The speed of convergence depends on the periodic check interval and might slow down the reaction to changes, but will on the other hand not cause too rapid changes (maximum one per interval).
- **Check When (the temperature is) Crossing Limit**. By monitoring the temperature and its variation it is possible to determine if the CEAS has stabilised. This strategy triggers MPLS (re)establishment when the variance of the temperature falls below (or crosses) a certain limit. The stability is only checked once and therefor reduces the number of unnecessary changes of the LSPs. It is however important to choose a correct stability threshold.
- **Check When (the temperature is) Above Limit**. A variant of the previous strategy. The absolute level of the variance of the temperature can be used to specify if the system is considered to be in a stable period or not. If the variance is below a certain limit the system is said to be stable, otherwise it is unstable. This strategy reacts fast to system changes, but a correct variance stability threshold must be set.

The latter two strategies are sensitive to the estimation of the variance of the temperature and to the specified limits.

A series of simulation in *ns2* is conducted. The *Check When (the temperature is) Above Limit* strategy performs best with respect to the bandwidth of the connections, but trigger more path changes compared to the *Check When (the temperature is) Crossing Limit* strategy. The *Check Periodically (the CEAS pheromone values)* strategy is easy to implement and requires no calculation and storage of CEAS indices like temperatures and its variance. It reacts slower to changes than the other two since it is necessary to wait till the next CEAS check is conducted.

5 Demonstrator System

Implementing a working prototype provides useful insights into the complexity of swarmbased methods in real routers, and reveals potential implementation challenges and performance bottlenecks which are hard to predict through simulations and analysis alone. Hence, CEAS has been prototyped, both to gain implementation insights and to provide a running system able to demonstrate CEAS routing principles and illustrate the inner workings of the method. Both technical implementations of CEAS described in this section are based on the *Click* Modular software router system [56]. Section 5.1 describes the first pioneering prototype implementation of CEAS which uses Click for packet forwarding and a Java-based Mobile Agent System called *Kaariboga* [57] for the routing process. An upgraded prototype implementation denoted *AntPing* is described in Section 5.2. AntPing provides improved performance by use of Click for both routing and forwarding. Section 5.2 explains how to use the AntPing to demonstrate CEAS and how a viewer can interact with the system.

5.1 Mobile Agent CEAS

The Mobile Agent CEAS was implemented as a "proof-of-concept" to gain experience with technical issues and effects that are hard to predict only through simulations. Maximising the performance of the working system was not a key consideration. The implementation was conducted as part of a Master assignment [58,59].

The system consists of two main components which interact to integrate the ants with the underlying network as shown in Figure 13. The software router is a customised version of the *Click modular router* system [56]. This implements the forwarding engine where data packets are forwarded according to the routing table that are updated by the ant system. The CEAS logic are implemented in a Java based Mobile Agent System framework called Kaariboga [57]. This ant-system receives ant packets detected by the kernel of the host machine. In case the ant packets are forward ants, they are routed stochastically according to the routing table, and otherwise the routing table is updated based on information in the ant packets and then forwarded to the next hop given in the header of the same packet.

The system was successfully implemented and tested in a small network. The test showed that the system is able to adapt to changes in traffic patterns and topology in the underlying network. Java and mobile agent systems are well suited for a rapid implementation of the system, however the implementation suffers from severe performance limitations even in a small-scale demo network.

5.2 AntPing

AntPing is also a prototype implementation of CEAS [54] developed as part of the final deliverable of the BISON project (IST-2001-38923). The main purpose was to learn more about the realization challenges of swarm intelligence on IP routers. The demonstrator visualises how ants are moving and dropped in the network. Animations are live and show how ants are searching and updating paths. Live plots of current and historical cost values of each virtual path are also provided as a function over time. The rest of this section includes a few details about the implementation and description about what it demonstrates.

Implementation. To achieve improved performance compared to the prototype system in Section 5.1, *AntPing* is implemented without use of the mobile agent system. Ants are no longer mobile agents but simple IP packets. AntPing extends the Click Modular software router system, and uses *hping3* [60] to generate and receive ant-packets from source to destination. The AntPing is running on home routers, with OpenWRT Linux [61], see [54] for more details. Figure 14 shows the functional blocks that extend Click (in the routers) and hping (in the end-systems). Figure 15 shows a picture from the lab. This implementation has moderate hardware and software requirements, which makes the demo inexpensive, flexible, and portable.

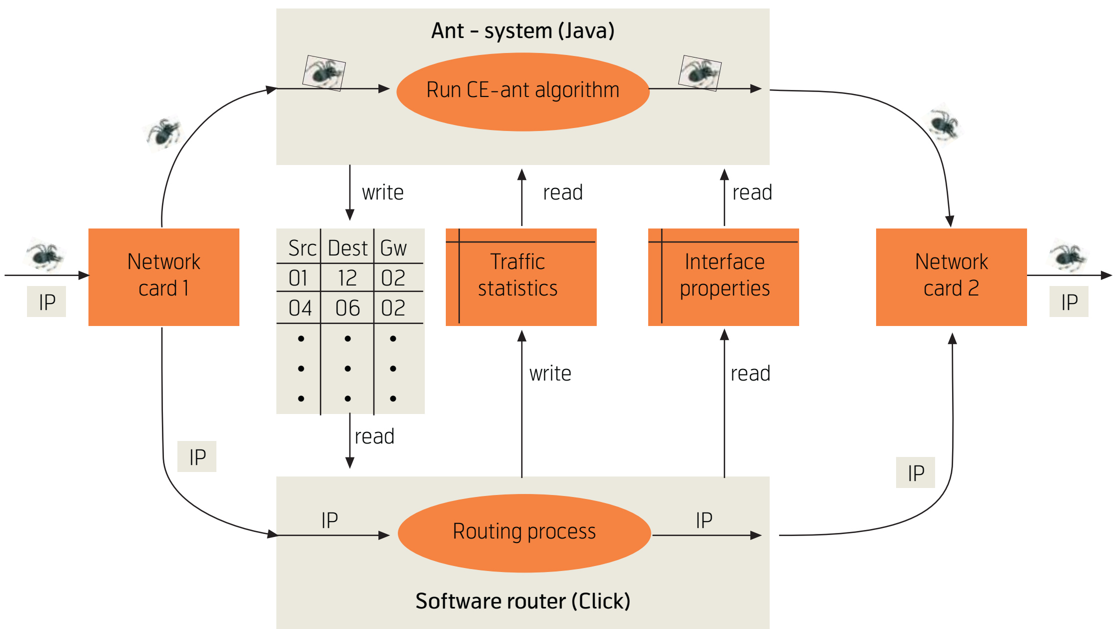


Figure 13 The main components of the hosts

ISSN 0085-7130 © Telenor ASA 2008

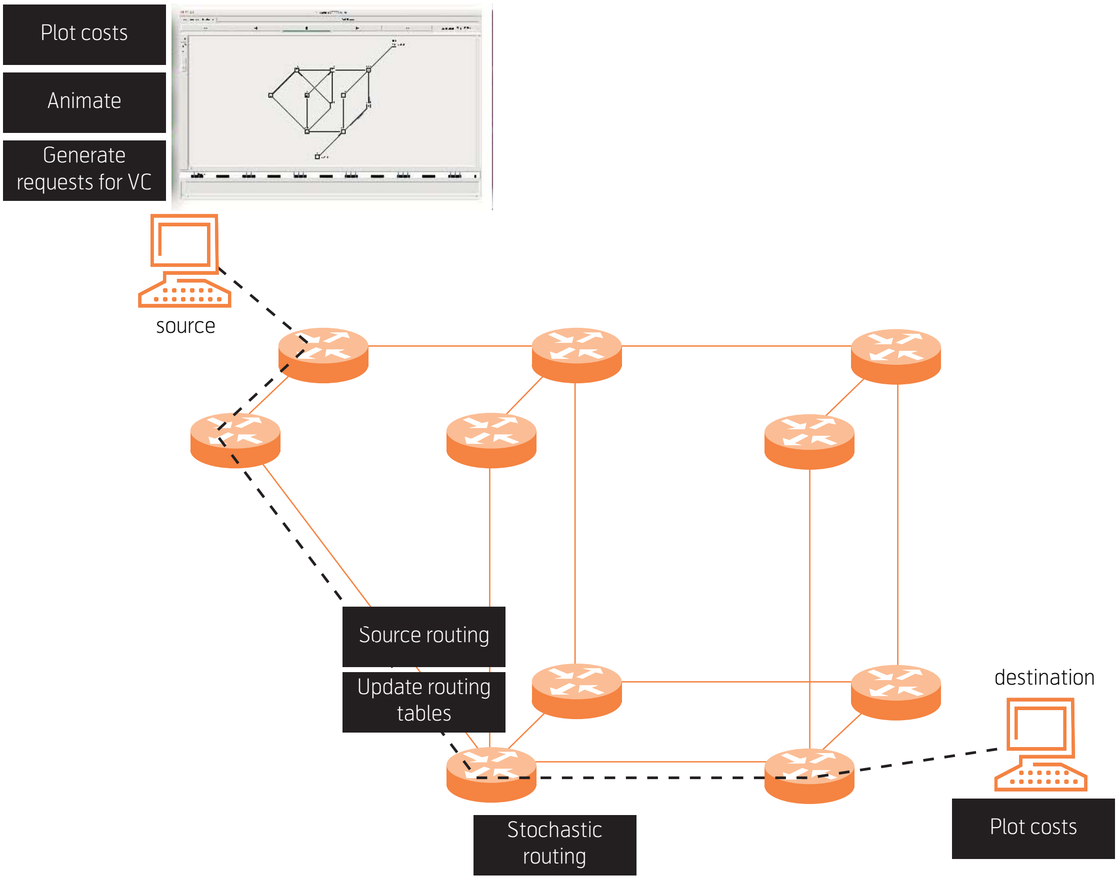


Figure 14 AntPing implementation

For the first version of this prototype the initial demo network topology is fixed and both sender and receiver processes are executed on external machines. In [62] the functionality of AntPing is extended to enable a self-configuring topology and to provide receiver processes in Click on every router. The latter implies that adding several destinations is possible without adding extra equipment.

Visualisation & interaction. The purpose of the demonstrator is to illustrate the inner workings of a swarm based method and to provide an interactive technical installation. The ant algorithm is animated live by use of Network Animator (*nam* [63]) showing how ants are moving and being dropped in the network, and how the topology is changing with link and node failures and restorations. The animation also shows ants that do not find the destination but are dropped because the Time-To-Live (TTL) is expired. Changes in cost values as a function over time of each virtual path are plotted live by use of *gnuplot* [64], both the cost of the current best path, and the cost of the last path found.

It is up to the users/audience to introduce network dynamics. They may unplug and re-plug cables between the nodes and/or the power supply to the nodes. Due to the extensions introduced in [62] new interfaces or links can also be added and several virtual connections can be established and monitored.

6 Closing Comments

Considering that finding paths between nodes is the basic and fundamental enabling functionality in a communication network, and that service handling in the future networks puts a wider range of requirements; an extension of the state-of-art in path management functionality is mandatory. Rather than pursuing "ad-hoc" improvements of current schemes or resorting to centralized management, we have addressed the problem by developing and applying the Cross Entropy Ant System (CEAS). This inherently robust, truly distributed and dynamically self-optimizing approach represents an important alternative paradigm for path management. The fact that CEAS has been applied to and is coping well with different relevant network management challenges, is a promising indication of its future.

There are, however, challenges ahead. We are confident that CEAS will operate in topologies typical for intradomain networks. Dealing with dynamic path management in the interdomain, as well as more comprehensive resource management tasks, requires additional functionality and improved insight into the

Figure 15 Setup from the lab (LinkSys WRT54GS (v4.0) routers)

rate of convergence and scalability issues. We intend to look further into these issues. With respect to the results presented in this paper, we will continue to look into how further improvements can be made to the system and merge our experience both with the simulated versions and the prototype implementations into an efficient design and implementation.

References

1 Ash, G R. *Dynamic Routing in Telecommunications Networks*. McGraw-Hill Professional, First ed., November 1, 1997. (ISBN 0-07-006414-8)

2 Huitema, C. *Routing in the Internet*. Prentice Hall PTR, 2 ed., November 1999.

3 Grover, W D. *Mesh-based Survivable Transport Networks : Options and Strategies for Optical, MPLS, SONET and ATM Networking*. Prentice Hall PTR, 2003.

4 Vasseur, J-P, Pickavet, M, Demeester, P. *Network Recovery : Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. The Morgan Kaufmann Series in Networking, Morgan Kaufmann, 2004.

5 Krunz, M, Matta, I (Eds.). Quality of service routing – Special issue. *IEEE Communications Magazine*, 40, Jun 2002.

6 Cholda, P, Mykkeltveit, A, Helvik, B, Wittner, O, Jajszczyk, A. A survey of resilience differentiation frameworks in communication networks. *Communications Surveys and Tutorials*, 9 (4), 2007.

7 Rubinstein, R Y. The Cross-Entropy Method for Combinatorial and Continuous Optimization. *Methodology and Computing in Applied Probability*, 127–190, 1999.

8 Ball, M O. *Handbooks in Operation Research and Management Science, Network Models*, 7. North Holland, 1995.

9 Pioro, M, Medhi, D. *Routing, Flow and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers, March 2004. (ISBN 0125571895)

10 Kirkpatrick, S, Gelatt, C D, Vecchi, M P. Optimization by Simulated Annealing. *Science*, 220, 671–680, 1983.

11 Glover, F, Laguna, M. *Tabu Search*. Kluwer Academic, 1997.

12 Goldberg, D. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1998.

13 ITU. *Types and characteristics of SDH network protection architectures*. Geneva, 1998. ITU-T G.841 (10/98).

14 ITU. *ATM protection switching*. Geneva, 1999. ITU-T I.630 (02/99).

15 Shen, G, Grover, W D. Survey and performance comparison of dynamic provisioning methods for optical shared backup path protection. *2nd International Conference on Broadband Networks*, 2005, 2, 1310–1319, 2005.

16 Jozsa, B G, Orincsay, D, Kern, A. Surviving multiple network failures using shared backup path protection. *Proceedings, Eighth IEEE International Symposium on Computers and Communication (ISCC 2003)*, 2, 1333–1340, 2003.

17 Zhou, L, Held, M, Sennhauser, U. Connection availability analysis of shared backup path-pro-

tected mesh networks. *Journal of Lightwave Technology*, 25, 1111–1119, 2007.

18 Mello, D A A, Pelegrini, J U, Ribeiro, R P, Schupke, D A, Waldman, H. Dynamic provisioning of shared-backup path protected connections with guaranteed availability requirements. *2nd International Conference on Broadband Networks 2005*, 2, 1320–1327, 2005.

19 Rai, S, Mukherjee, B, Deshpande, O. Ip resilience within an autonomous system: current approaches, challenges, and future directions. *IEEE Communications Magazine*, 43, 142–149, 2005.

20 Nelakuditi, S, Lee, S, Yu, Y, Zhang, Z-L, Chuah, C-N. Fast local rerouting for handling transient link failures. *IEEE/ACM Transactions on Networking*, 15, 359–372, 2007.

21 Menth, M, Martin, R, Hartmann, M, Spoerlein, U. *Efficiency of routing and resilience mechanisms*. Currently under submission, May 2007.

22 Schoonderwoerd, R, Holland, O, Bruten, J, Rothkrantz, L. Ant-based Load Balancing in Telecommunications Networks. *Adaptive Behavior*, 5 (2), 169–207, 1997.

23 Di Caro, G, Dorigo, M. AntNet: Distributed Stigmergetic Control for Communications Networks. *Journal of Artificial Intelligence Research*, 9, 317–365, 1998.

24 Wittner, O, Helvik, B E. Distributed soft policy enforcement by swarm intelligence; application to load sharing and protection. *Annals of Telecommunications*, 59, 10–24, 2004.

25 Wittner, O. *Emergent Behavior Based Implements for Distributed Network Management*. Trondheim, Norwegian University of Science and Technology (NTNU), Department of Telematics, November 2003. (PhD thesis)

26 Wedde, H F et al. Beeadhoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior. In: *Proceedings of the 2005 conference on Genetic and evolutionary computation (GECCO '05)*, New York, 2005, 153–160, ACM Press.

27 Di Caro, G, Gambardella, L M. AntHocNet: An Adaptive Nature-Inspired Algorithm for Routing in Mobile Ad Hoc Networks. *European Transactions on Telecommunications (ETT)* – Special Issue on Self Organization in Mobile Networking, 16 (5), 443–455, 2005.

28 Ducatelle, F, Gambardella, L M. Survivable routing in ip-over-wdm networks: An efficient and scalable local search algorithm. *Optical Switching and Networking*, 2 (2), 86–99, 2005.

29 Steels, L. Towards a theory of emergent functionality. In: Meyer, J-A, Wilson, S (eds.) *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior, SAB'90, Complex Adaptive Systems*, 451–461. Cambridge, MA, The MIT Press, 1990.

30 Damper, R I. Editorial for the special issue on 'emergent properties of complex systems': Emergence and levels of abstraction. *International Journal of Systems Science*, 31 (7), 811–818, 2000.

31 Engelbrecht, A P. *Fundamentals of computational swarm intelligence*. Wiley, 2005. (ISBN 13 978-0-470-09191-3)

32 Dorigo, M, Maniezzo, V, Colorni, A. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 26 (1), 29–41, 1996.

33 Heidelberger, P. Fast simulation of rare events in queueing and reliabili ty models. *ACM Transactions on modelling and Computer Simulation*, 5, 43–85, 1995.

34 De Boer, P-T, Kroese, D P, Mannor, S, Rubinstein, R Y. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134, 19–67, 2005.

35 Helvik, B E, Wittner, O. Using the Cross Entropy Method to Guide/Govern Mobile Agent's Path Finding in Networks. In: *Proceedings of 3rd International Workshop on Mobile Agents for Telecommunication Applications*, August 14–16, 2001, Springer Verlag.

36 Heegaard, P E, Wittner, O, Nicola, V F, Helvik, B E. Distributed asynchronous algorithm for cross-entropy-based combinatorial optimization. In: *Rare Event Simulation and Combinatorial Optimization (RESIM/COP 2004)*, Budapest, Hungary, September 7–8, 2004.

37 Stutzle, T, Hoos, H. MAX-MIN Ant System. *Journal of Future Generation Computer Systems*, 16, 889–914, 2000.

38 Kjeldsen, V. *Cooration through pheromone sharing in swarm routing*. Norwegian University of Science and Technology, 2007. (Master thesis)

39 Heegaard, P E, Wittner, O. Restoration performance vs. overhead in a swarm intelligence path management system. In: Dorigo, M, Gambardella, L M (eds.) *Proceedings of the Fifth International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS2006)*, LNCS, Brussels, Belgium, September 4–7, 2006, Springer.

40 Heegaard, P E, Wittner, O J. Self-tuned refresh rate in a swarm intelligence path management system. In: *Proceedings of the EuroNGI International Workshop on Self-Organizing Systems (IWSOS 2006)*, LNCS, University of Passau, Germany, September 18–20, 2006, Springer.

41 Heegaard, P E, Wittner, O, Helvik, B E. Self-managed virtual path management in dynamic networks. In: Babaoglu, O et al. (eds.) Self-* Properties in Complex Information Systems. *Lecture Notes in Computer Science*, LNCS 3460, 417–432, Springer, 2005. (ISSN 0302-9743)

42 Shannon, C E. A mathematical theory of communication. *Bell System Technical Journal*, 27, 379–423, 623–656, July, October 1948.

43 Wittner, O, Helvik, B E. *Simulating mobile agent based network management using network simulator.* Poster in Fourth International Symposium on Mobile Agent System (ASA/MA 2000), September 2000.

44 Wittner, O, Heegaard, P E, Helvik, B E. Scalable distributed discovery of resource paths in telecommunication networks using cooperative ant-like agents. In: *Proceedings of Congress on Evolutionary Computation, CEC2003*, Canberra, Australia, IEEE, December 2003.

45 Kjeldsen, V, Wittner, O J, Heegaard, P E. Distributed and Scalable Path Management by a System of Cooperating Ants. In: *Proceedings of the International Conference on Communications in Computing (CIC 2008)*, Las Vegas, USA, July 14-17, 2008.

46 Kirkerud, B. *Object-oriented programming with SIMULA*. Addison Wesley, 1989.

47 Birtwistle, G. *Demos – a system for discrete event modelling on simula*. 1997.

48 Wittner, O, Helvik, B E. Cross Entropy Guided Ant-like Agents Finding Dependable Primary/ Backup Path Patterns in Networks. In: *Proceedings of Congress on Evolutionary Computation (CEC2002)*, Honolulu, Hawaii, IEEE, May 12–17, 2002.

49 Garey, M R, Johnson, D S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

50 Stamatelakis, D, Grover, W. Rapid Span or Node Restoration in IP Networks using Virtual Protection Cycles. In: *Proceedings of 3rd Canadian Conferance on Broadband Research (CCBR'99)*, Ottawa, 7 November 1999.

51 Wittner, O, Helvik, B E, Nicola, V F. Internet failure protection using hamiltonian pcycles found by ant-like agents. *Journal of Network and System Management, Special issue on Self-Managing Systems and Networks*, 2005.

52 Heegaard, P E, Wittner, O, Helvik, B E. Self-managed virtual path management in dynamic networks. *Lecture Notes in Computer Science, Hot Topics*, 3460, 417–432, 2005.

53 Heegaard, P. *Performance monitoring of routing stability in dynamic networks*. Section 5 in Deliverable 05: Models for basic services in AHN, P2P and Grid networks in IST-FET Project BISON, December 2003. (IST-2001-38923)

54 Heegaard, P, Fuglem, I. *Demonstrator 1: Ant-based monitoring on software ip routers*. Tech. Rep., BISON, 2006. (IST-2001-38923)

55 Hesby, N, Heegaard, P E, Wittner, O. Robust connections in ip networks using primary and backup paths. In: *Proceedings of the 17th Nordic Teletraffic Seminar*, Fornebu, Norway, 25–27 August 2004.

56 Kohler, E et al. The click modular router. *ACM Transactions on Computer Systems*, 18, 263–297, 2000.

57 Struve, D. *Kaariboga mobile agents*. September 2003. http://www.projectory.de/kaariboga/, visited October 2003.

58 Mykkeltveit, A, Heegaard, P, Wittner, O. Realization of a distributed route management system on software routers. In: *Proceedings of Norsk Informatikkonferanse*, Stavanger, Norway, 29 Nov – 1 Dec 2004.

59 Mykkeltveit, A. *Realization of a distributed route management system by mobile agents*. Tech. Rep., Dept. of Telematics, NTNU, 2003.

60 Sanfilippo, S. *hping*. http://wiki.hping.org/ [online] – last checked 2007-09-05.

61 *Openwrt*. http://wiki.openwrt.org [online] – last checked 2007-09-05.

62 Kristiansen, T A. *Self-configured demonstrator of the Cross Entropy Ant System*. Norwegian University of Science and Technology, 2007. (Master thesis)

63 *Nam: Network animator*. http://www.isi.edu/nsnam/nam/ [online] – last checked 2007-09-05.

64 *Gnuplot*. http://www.gnuplot.info/ [online] – last checked 2007-09-05.

Poul E. Heegaard received his Siv.Ing. degree (M.S.E.E.) in 1989 from the Norwegian Institute of Technology (NTH), Trondheim, Norway and was awarded the degree Dr.Ing. (PhD) from NTH in 1998. Heegaard has since 2006 been Associate Professor at Department of Telematics, NTNU, where he is the coordinator of the Network Research area. In the academic year 2007/2008 he is a visiting professor at Duke University, Durham, NC, USA. Since 1999 he has been a Senior Research Scientist in Telenor R&I in Trondheim, where he currently holds a 20% position. He has previously been a Research Scientist and Senior Scientist at SINTEF Telecom and Informatics (1989-1999). His research interests are in the area of performance, dependability and survivability evaluation of communication systems. His special interests are rare event simulation techniques, and monitoring, routing and management in dynamic networks. His current research focus is on distributed, autonomous and adaptive management and routing in communication networks and services.

email: poul.heegaard@item.ntnu.no

Bjarne E. Helvik received his Siv.Ing. and Dr.Techn. degrees in 1975 and 1982, respectively. Since 1997 he has been Professor at the Norwegian University of Science and Technology (NTNU), Department of Telematics and is currently with the Norwegian Centre of Excellence (CoE) in Quantitative Quality of Service in Communication Systems (Q2S). He has previously held various positions with ELAB, SINTEF Telecom and Informatics, and as Adjunct Professor at NTH. Helvik's field of interests includes QoS, dependability modelling, measurements, analysis and simulation, fault-tolerant computing systems and survivable networks. His current research focus is on distributed, autonomous and adaptive fault-management in telecommunication systems, networks and services.

email: bjarne@q2s.ntnu.no

Otto J. Wittner received his Dr.Ing. degree from the Norwegian University of Science and Technology (NTNU) in 2003. He also joined the Norwegian Centre of Excellence (CoE) for Quantifiable Quality of Service in Communication systems (Q2S) in 2003 as a postdoc. His research interests focus on network control and management by emergent behavior principles, where he is looking into dependability issues, especially fault management.

email: wittner@q2s.ntnu.no