Self-management of virtual paths in dynamic networks

Poul E. Heegaard[1], Otto Wittner[2], and Bjarne E.Helvik[2]

[1] Telenor R&D * and Department of Telematics,
Norwegian University of Science and Technology, Norway
poulh@item.ntnu.no
[2] Centre for Quantifiable Quality of Service in Communication Systems**
Norwegian University of Science and Technology, Trondheim, Norway
{bjarne,wittner}@q2s.ntnu.no

Abstract Virtual path management in dynamic networks poses a number of challenges related to combinatorial optimisation, fault and traffic handling. Ideally such management should react immediately on changes in the operational conditions, and be autonomous, inherently robust and distributed to ensure operational simplicity and network resilience. Swarm intelligence based self management is a candidate potentially able to fulfil these requirements. Swarm intelligence achieved by cross entropy (CE) ants is introduced, and two CE ants based path management approaches are presented. A case study of a nation wide communication infrastructure is performed to demonstrate their abilities to handle change in network traffic as well as failures and restoration of links.

Keywords Cross-entropy, Swarm intelligence, Ant-based optimisation, elite CE ants, Network management, Resilience,

1 Introduction

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 results in a 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 [1]. Nevertheless, considerable knowledge has been acquired for planning paths in networks [2]. Establishment of virtual path layouts and deployment of backup paths are issues discussed in this paper. Insight and practical methods

* This work was partially supported by the Future& Emerging Technologies unit of the European Commission through ProjectBISON (IST-2001-38923).

** Centre for Quantifiable Quality of Service in CommunicationSystems, Centre of Excellence appointed by The Research Council of Norway,funded by the Research Council, NTNU and UNINETT. http://www.ntnu.no/Q2S/

for obtaining such paths by mathematical programming are available. For an overview, see the recently published book by Pióro and Medhi [2] and references therein. Several stochastic optimisation techniques which may be used to address these kinds of problems, have been proposed [3,4,5,6]. 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 solution which are less fine tuned with respect to optimal resource utilisation.

In addition 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. 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 [7,8]. Typically two (or more) disjoint paths are established, one serving as a backup for the other. Protection switching requires preplanning, is rather inflexible and is not very efficient in utilising network resources. Shortest path, distance vector and policy based routing as applied in the Internet, is distributed, have local decision-making and applies to some degree planning inherent in the network, see for instance [9]. 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 not unusual 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.

Schoonderwoerd & al. introduced the concept of using multiple agents with a behaviour inspired by ants to solve problems in telecommunication networks [10]. The concept is known as *swarm intelligence* [11] and has been pursued further by others, see for instance [12,13,14] 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. This is elaborated further in Section 2. For dealing with path management in communication networks we have developed the CE ants (cross-entropy ants) which are based on Rubinstein's method for stochastic optimisation [6]. CE ants and their application in two path management approaches are presented in Section 3. The first approach, adaptive paths, presented for the first time in this paper, applies a stochastic routing scheme and is promising with respect to robust and adaptive forwarding. In Section 4, a case study demonstrates the adaptive abilities of the two CE ants based path management approaches. The approaches are confronted with a changing traffic load as well as failures and

restorations of links in the network. This demonstration is one of the original contributions of the paper. Some concluding remarks are given in Section 5.

2 Position on path management

Being able to transfer addressed information between sources and destinations is the prime function of a communication network, and hence, how to find a way for the information through the network is one of the most salient issues in network architecture and operation. How this has been done throughout history is a trade-off between requirements and available technology. We raise the question: is self-management by emergent behaviour a viable approach to path finding in future networks, meeting the requirement of an integrated multi-service transport network? Paths are in this context both explicit paths as virtual connections realized for instance by multiprotocol label switching (MPLS), [15], and implicit paths given by for instance open shortest path first (OSPF) routing tables, [9].

To substantiate our position on this question we first introduce the main objectives of routing/path management in networks, discuss basic architectural issues and outline two alternative algorithms and their rationale with respect to management. Details and the performance of these algorithms are presented later in the paper.

By path management we address medium length temporal characteristics of network, i.e., how to use the available network resources to establish paths that best meet the requirements of the offered traffic on a time scale in the range from 100 ms to some hours. Long term planning, involving installation and rearrangement of physical equipment, is outside the scope of path management together with the short term management issues dealing with the real time characteristics of the individual flows. The objective of path management may be summarised by the following obviously interrelated items.

Path finding ability This is the obvious objective; if a path exists between a source and a destination in a network, it should be found. There may be additional objectives to find multiple node and/or link disjoint paths for better resource utilisation and resilience.

Resource utilisation The network resources should be used efficiently. The exact interpretation of efficiency will depend on the QoS objectives of the network, e.g., be close to some optimal value with respect to traffic carried (or operator income) when constrained by QoS requirements. The efficiency should be maintained under changes in the loading of the network, in the topology and in the capacities of the network elements, i.e., *adaptivity* is mandatory for efficient resource utilisation.

Resilience If feasible, the dependability requirements of the service provided should be met in terms of availability, reliability and down time. In some services, the temporal aspects, e.g. continuity of service or negligible down times, are of great importance. A prerequisite for resilient services is the *resilience of the management system itself,* i.e., it should be robust to failures

of network elements as well as loss of management information due to failures and overload.

Priorities and fairness Common to most networks is that they should provide a fair service, i.e., all users (or traffic flows) having the same "status", should statistically receive the same QoS. Future networks will carry a variety of services with highly differing dependability requirements and importance (end-users' willingness to pay). In these networks, the management system has to implement priorities to deal with an offered load exceeding the capacity and with failures, while maintaining a generally high resource utilisation.

There are two major design axis for management systems, a spatial axis, i.e., degree of centralisation or distribution, and a temporal axis, i.e., the degree of preplanning. It would be too lengthy to go into detail on various solutions, so the discussion is limited to establishing the main pros and cons of the various options.

Centralisation has the advantage that decisions may be based on a global view, and hence, in theory, better decisions with respect to resource utilisation and priorities may be made. Its drawback is that it is vulnerable since centralisation yields a single point of failure, and since these systems have to rely on a potentially partially overloaded or failed network for extraction of measurements and status information, as well as for dissemination of control information. Centralised decision-making may also be slow due to a long decision cycle. *Distribution* tends to yield poorer resource utilisation, but has a potentially better resilience. preplanning is needed to be able to rapidly respond to network element failures, e.g. by protection switching, and may be used to deal with anticipated changes in the load, e.g. daily variations. For *long term preplanning*, it is impossible to plan for all eventualities making it necessary to plan for ranges of eventualities which may result in more costly solutions. On the other end of the scale, we have the *reactive* systems. Their drawback is the restoration time, i.e. the inability to provide continuity of service shortly after a failure. To avoid the extremes we advocate *short term planning*, dynamically making contingency plans for the current network state and load, or a *pro-active* preparation for a fast restoration, cf. Section 3.1.

In the public telecom networks, primarily designed for telephony, preplanned protection switching schemes (with a distributed implementation) is typically used to achieve fast fault management. This scheme is rather inflexible and costly in terms of spare equipment. It is combined with otherwise centralised management to obtain high resource utilisation and control of delays and loss. The Internet, which has its architecture governed by the requirement to survive a nuclear attack, has an inherent robustness in part achieved by distributed path finding. Resilience is the prime objective, while QoS is less focused. The major drawback of the Internet approach is the relative long time needed for restoration after failures, which results in missing ability to provide continuity of service when failures occur. Another issue is that fixed link costs are typically used in the routing algorithms, and hence, the ability to adapt the flows in

the network to changes in the load is restricted. MPLS and the generalised multiprotocol label switching (GMPLS) for management of underlaying circuit switched networks, have been introduced as means that may be used to overcome these drawbacks. However, the tendency is toward using (G)MPLS based on "off-line" centralised preplanning and thereby missing some of Internet's inherent robustness and adaptivity.

It is our research hypothesis that self-management of path finding by emergent behaviour has the potential to provide the advantages of both these approaches, inherent robustness and adaptivity, a good resource utilisation as well as continuity of service by protection-like schemes or pro-active path routing schemes. A drawback is that in order to achieve this, determinism is sacrificed. To support this hypothesis we have developed an emergent path finding algorithm based on the CE ants approach to stochastic optimisation [6], and performed extensive experiments on two variants of self-management algorithms with decreasing "designedness" in order to meet the continuity of service objective. The **primary backup** scheme has as its prime objective to establish disjoint primary and backup (MPLS) paths for (all) source destination pairs. The 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. Its main pro is the explicit knowledge of the immediately restorable traffic. Its scalability for increasing network sizes and complex priority schemes is not yet investigated. The other approach is the **adaptive path** scheme, which has stochastic paths for all source destination pairs in all nodes of the network. This approach will pro-actively provide alternative paths in case of failure. Its main pros 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. In order to substantiate our position, the remainder of the paper will present, compare and discuss these two schemes in the context of the objectives listed at the beginning of this section.

3 Cross Entropy Ants (CE ants)

The CE ant system which forms the foundation for the work presented in this paper, is a swarm intelligent system originally inspired by the foraging behaviour of ants, as outlined in the introduction. The overall idea is to have a number of simple ant-like mobile agents iteratively search for paths in a network. An ant, having found a path, backtracks and leaves markings, denoted *pheromones,* resembling the chemicals left by real ants during ant trail development. The strength of the change in markings depends on the quality of the path found. Hence, nodes hold distributions of pheromones pointing toward their neighbour nodes. A new ant in its searching phase visiting a node selects the next node to visit stochastically based on the pheromone distribution seen in the visited node. Using such trail marking ants, together with evaporation of pheromone, the overall process converges quickly toward having the majority of the ants follow

a single trail that tends to be a near optimal path. The behaviour of the cross entropy ants, to be presented in Sections 3.2 to 3.4, is in addition to mimicking ants in nature, founded in Rubinstein's method for stochastic optimisation [6]. Due to space limitation, the rest of this section presents only an outline of the CE ant system. For details readers are referred to [14].

The path management strategy implemented by the ants is governed by how the "quality of a path found" is determined. We denote this quality (or the lack of quality) *cost*. Traffic streams between pairs of nodes in the network is indexed by m. A path for this stream found by the t'th ant is denoted π_t^m. A link connecting two adjacent nodes i, j has a link cost L_{ij}. The link cost is chosen to a measure appropriate for the problem at hand. It may for instance be in terms of incurred delay by using the path, "fee" paid to the operator of the link, a penalty for using a scare resource like free capacity, etc., or a combination of such measures. The link cost may depend on the traffic stream to be carried and when the cost is observed. If this is the case the cost observed by the t'th ant is denoted $L_{t,ij}^m$. The cost function, L, of a path is the sum of the link costs, i.e.

$$L(\pi_t^m) = \sum_{ij \in \pi_t^m} L_{t,ij}^m \tag{1}$$

3.1 Management strategies

The management strategy should be reflected in the cost function determining the cost for the individual ants. Below two such strategies and their corresponding cost functions are presented.

Primary backup This strategy is designed to provide soft guarantees for retaining service under single link failures. This is done by finding pairs of mutually disjoint primary and back-up paths. Hence, the ants seeking primary paths and backup paths should detest each other. The capacity of the primary paths will be used in fault free operation, and ants finding primary paths should detest each other if using a common link would cause overload. The capacity on the back-up paths will be allocated and shared with other backup paths. Backup paths having primary paths with common links should also avoid using common links in the backup path that may be overloaded if the common primary link fails. The cost function should give high penalty to the primary backup paths where the accumulated traffic demand exceeds the capacity of at least one link. Hence, a penalty related to the approximate expected potential link overload, including the above mention detestation, is chosen as the link cost. The primary backup link cost expression relating to stream m_r,where r indicates rank of path ($r = 0 \Rightarrow$ primary and $r = 1 \Rightarrow$ backup), has the following structure:

$$L_{t,ij}^{m_r} = \mathcal{S}\left[a_m + \sum_{\forall n_s:\, ij \in \pi_t^{n_s}} P_{t,ij}^{n_s}\, V_{t,i}^{n_s}\, \mathcal{Q}_{t,m_r}^{n_s}\, a_n - c_{ij}\right] \tag{2}$$

where a_m, and a_n represent the load put by streams m and n, and c_{ij} capacity available on link ij. $P_{t,ij}^{n_s}$ is the probability that an ant n_s of rank s ($s = 0 \Rightarrow$ primary and $s = 1 \Rightarrow$ backup) for stream n will follow link ij when it visits node i, and $V_{t,i}^{n_s}$ is the probability that ant n_s visits node i. The factor $P_{t,ij}^{n_s} V_{t,i}^{n_s}$ indicates the likelihood, at the current state of the emerging process, that the actual link will be chosen as a path for stream n. These quantities are derived directly from pheromone levels in node i. $\mathcal{Q}_{t,m_r}^{n_s}$ is a weight function controlling the intensity of the detestation. $\mathcal{S}[...]$ is a smoothening function ensuring $L_{t,ij}^{m_r} > 0$. For details see [13].

Adaptive path This strategy is designed for fast restoration and adaptivity to both link failures and change in traffic loads. Hence, the cost function should be sensitive to the carried traffic, which makes a delay based link cost measure a natural choice. The link cost measure includes both the queueing and processing delay in a node, and the transmission delay of a link. The link cost measure for a link in a path for stream m is therefore $L_{t,ij}^{m} = d_{t,ij}^{m}$, where $d_{t,ij}^{m}$ is average delay induced by the link ij measured in the short time period between ant $t$and the succeeding ant traversing linkij.

3.2 The Cross Entropy Method

In [6] Rubinstein presents an algorithm for iteratively finding optimal solutions to hard combinatorial problems. It stems from the recognition of that finding the optimal solution by random selection is an extremely rare event. For instance, the probability of finding the shortest Hamiltonian cycle in a 26 node network is $\frac{1}{25!} \approx 10^{-26}$. Hence, a successive importance-sampling-like technique is used to increase the probabilities of finding good solutions. In our context, his approach may be regarded as a centralised search for a single best path in a network. For the sake of presentation, the cross entropy (CE) method is summarised with the above "ant terminology" with the modification that t is now interpreted as a batch of N ants rather than a single ant, cf. step 2 below. Hence Rubinstein's algorithm is batch oriented.

The total allocation of pheromones in a network is represented by a probability matrix P_t where an element $P_{t,ij}$ reflects the normalised intensity of pheromones pointing from node i toward node j. An ant's stochastic search for a sample path resembles a Markov Chain selection process based on P_t. By importance sampling in multiple iterations Rubinstein alters the transition matrix ($P_t \rightarrow P_{t+1}$) and increases, as mentioned, certain probabilities such that ants eventually find near optimal paths with high probabilities. Cross entropy is applied to ensure efficient alteration of the matrix. To speed up the process further, a performance function weights the path qualities such that high quality paths have greater influence on the alteration of the matrix, cf. step 2 below. Rubinstein's CE algorithm has 4 steps. The indexes m and r are omitted since a single path and single kind of ant is considered:

1. At the first iteration $t = 0$, select a start transition matrix $P_{t=0}$ (e.g. uniformly distributed).
2. Generate N paths from P_t. Calculate the minimum parameter γ_t, denoted *temperature*, to fulfil average path performance constraints, i.e.

$$\min \gamma_t \text{ s.t. } h(P_t, \gamma_t) = \frac{1}{N} \sum_{k=1}^{N} H(\pi_k, \gamma_t) > \rho \tag{3}$$

   where $H(\pi_k, \gamma_t) = \exp(-L(\pi_k)/\gamma_t)$ is the performance function returning the quality of path π_k. $L(\pi_k)$ is the cost of using path π_k as in Section 3.1, and $10^{-6} \leq \rho \leq 10^{-2}$ is a search focus parameter. The minimum solution for γ_t implies a certain reinforcement (dependent on ρ) of high quality paths and produces a minimum average $h(P_t, \gamma_t) > \rho$ over all path qualities in the current batch of N paths.
3. Using γ_t from step 2 and $H(\pi_k, \gamma_t)$ for $k = 1, 2..., N$, generate a new transition matrix P_{t+1} which maximises the "closeness" (i.e. minimises distance) to the optimal matrix, by solving

$$\max_{P_{t+1}} \frac{1}{N} \sum_{k=1}^{N} H(\pi_k, \gamma_t) \sum_{ij \in \pi_k} \ln P_{t+1,ij} \tag{4}$$

   where $P_{t+1,ij}$ is the transition probability from node i to j at iteration $t+1$. The solution of (4) is shown in [6] to be

$$P_{t+1,ij} = \frac{\sum_{k=1}^{N} I(\{i,j\} \in \pi_k) H(\pi_k, \gamma_t)}{\sum_{l=1}^{N} I(\{i\} \in \pi_l) H(\pi_l, \gamma_t)} \tag{5}$$

   where $I(X) = 1$ if $X = true$, 0 otherwise. (5) results in a minimised cross entropy between P_t and P_{t+1}, and ensures an optimal shift in probabilities with respect to γ_t and the performance function.
4. Repeat steps 2-3 until $H(\widehat{\pi}, \gamma_t) \approx H(\widehat{\pi}, \gamma_{t+1})$ where $\widehat{\pi}$ is the best path found.

3.3 Distributed Cross Entropy Method

In [16] a distributed and asynchronous version of Rubinstein's CE algorithm is developed, today known as *CE ants*. By a few approximations, (5) and (3) may be replaced by autoregressive counterparts based on

$$P_{t+1,ij} = \frac{\sum_{k=1}^{t} I(\{i,j\} \in \pi_k) \beta^{t-k} H(\pi_k, \gamma_t)}{\sum_{l=1}^{t} I(\{i\} \in \pi_l) \beta^{t-l} H(\pi_l, \gamma_t)} \tag{6}$$

and

$$\min \gamma_t \text{ s.t. } h_t^{'}(\gamma_t) > \rho \tag{7}$$

where

$$h_t^{'}(\gamma_t) = h_{t-1}^{'}(\gamma_t)\beta + (1-\beta)H(\pi_t, \gamma_t) = \frac{1-\beta}{1-\beta^t} \sum_{k=1}^{t} \beta^{t-k} H(\pi_k, \gamma_t)$$

and where $\beta \in \langle 0, 1\rangle$ (typically close to 1) controls the history of paths remembered by the system (i.e. replaces N in step 2). See [16] for details on the auto-regression. Step 2 and 3 in the algorithm can now be performed immediately after a single new path π_t is found (i.e. t again represents the t'the ant), and a new probability matrix P_{t+1} can be generated. Hence CE ants may be viewed as an algorithm where search ants evaluate a path found (and calculate γ_t by (7)) right after they reach their destination node, and then immediately return to their source node backtracking along the path. During backtracking, pheromones are placed by updating the relevant probabilities in the transition matrix, i.e applying $H(\pi_t, \gamma_t)$ through (6).

Due to the compact autoregressive schemas applied in a CE ant system, the system becomes both computationally efficient, requires limited amounts of memory and is simple to implement.

3.4 Elite CE ants

In [17] elitism is introduced in the CE ants system. The new system, denoted *elite CE ants,* performs significantly better in terms of the number on path traversals required to converge toward a near optimal path. The kind of contribution an ant makes depends on the cost of the path it has traversed relative to the cost of paths found by other ants. All ants contribute in updating the temperature γ_t as in (7). However, a limited set of ants, denoted the *elite set*, updates a different temperature γ_t^*. Only ants belonging to the elite set backtrack their paths and update pheromones applying $H(\pi_k, \gamma_t^*)$ in (6), and hence, reducing the total number of backtracking traversals and pheromone updates.

The criterion for determining if an ant is in the elite set is based on the fact that the best solutions in the CE ants method relates to ρ through $e^{-L(\pi_t)/\gamma_t} > \rho$, cf. step 2 in Section 3.2. The elite criterium of (8) is a rearrangement of this relationship. An ant is considered an elite ant if the cost of the path found by the ant satisfies

$$L(\pi_t) < -\gamma_t \ln \rho \tag{8}$$

Note that the temperate γ_t updated by *all* ants is applied in (8). Hence, when removing parts of the search space which enables elite ants to find their paths, e.g. by a link breakdown in the best path found, the temperature γ_t will increase and allow ants with higher path costs to perform pheromone updates. Hence dynamic network conditions are handled. Note also that the elite criterium does not introduce any additional parameters. It is self-tuning.

4 Case studies of a national-wide Internet topology

In order to demonstrate the effect of the swarm-based path management approaches, case studies are carried out based on a topology extracted from a national-wide Norwegian Internet provider. In this section the simulation cases are described, and results are given from the studies of adaptivity and robustness of the primary-backup and adaptive path strategies. A few comments on

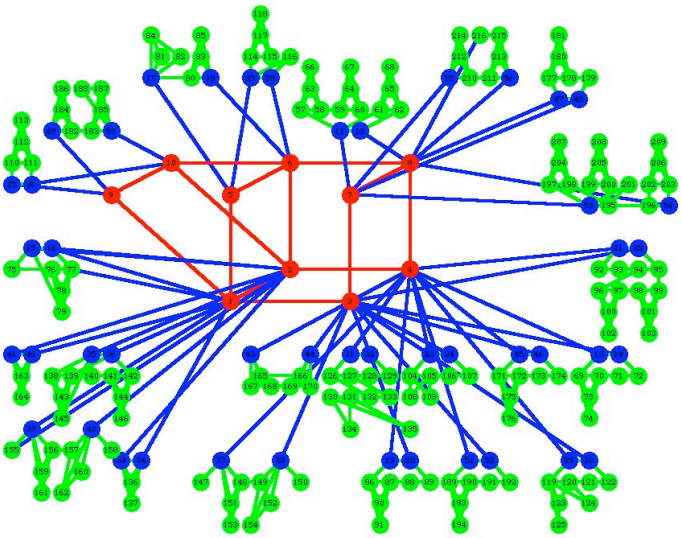

Figure 1. The network topology in case studies

the efficiency, i.e. the management overhead relative to its reactiveness, are also included. As previously mentioned, relative to more traditional centralised path finding schemes under static conditions, we expect to loose some "performance", but not necessary too much [16].

4.1 The simulation case description

The network in Figure 1 consists of a core network with 10 core nodes in a sparsely meshed topology, ring based edge networks with a total of 46 edge nodes, and dual homing access network with 160 access nodes. The relative transmission capacities are 1, 1/4 and 1/16 for core, edge and access links, respectively. In the processing delay in the nodes only includes the variable queueing delay as a function of the load level.

The path management of 10 separate paths is studied in details. The paths are exposed to network link failures, drops of management information, and changes in offered traffic loads. The terminal nodes, i.e. the ingress and egress nodes, of the 10 paths are all access nodes. Each path is routed through an edge and the core network. The average load, ρ, is the link utilisation of every link of the paths through the network. The traffic is routed according to the (multi) paths provided by the management algorithm. This traffic represents the background traffic and is added to study how the algorithm reacts to load variations. In order to stress the algorithm and create instabilities, the load changes are in

significant steps, see Table 1. All results in the following are from 10 simulation replications.

The objective is to study the transient behaviour, i.e. the adaptivity and robustness, of this distributed management approach. Hence, the dynamism simulated are specific, and abrupt, changes in the network environments. The main observations from these experiments are given in the following.

4.2 Adaptivity

In order to test the adaptivity of the path management approach, a scenario with changes in traffic pattern and load level, and changes in structure (link failures and restorations) is defined. The details of the 9 phases of the scenario are given in Table 1.

Adaptive path strategy The results presented in Figure 2 are the average cost values from 10 simulation replications over the 9 phases. The results are from 3 of the 10 paths, selected from the paths that are affected by at least one of the changes in network conditions given in Table 1. There are three main observations from the series of simulation experiments.

1. The adaptive path strategy will switch to an alternative path almost immediately. This will in some cases cause a transient decrease in the quality (e.g. delay) but not necessary an interruption of the transport service. As an example, follow the path VC1. When a core link fails (phase 5 to 6), a sudden increase in the cost value of VC1 is observed because the preferred path is no longer available. An alternative path is immediately available. The elite CE ants continue to search for better paths. In this experiment the alternative eventually found in phase 6 has the same cost value as the best in phase 5. If a prescribed upper bound on the delay of the transport service relying on the VC, the service will not be conform with the requirements

Table 1. Dynamic scenario for testing of adaptivity

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

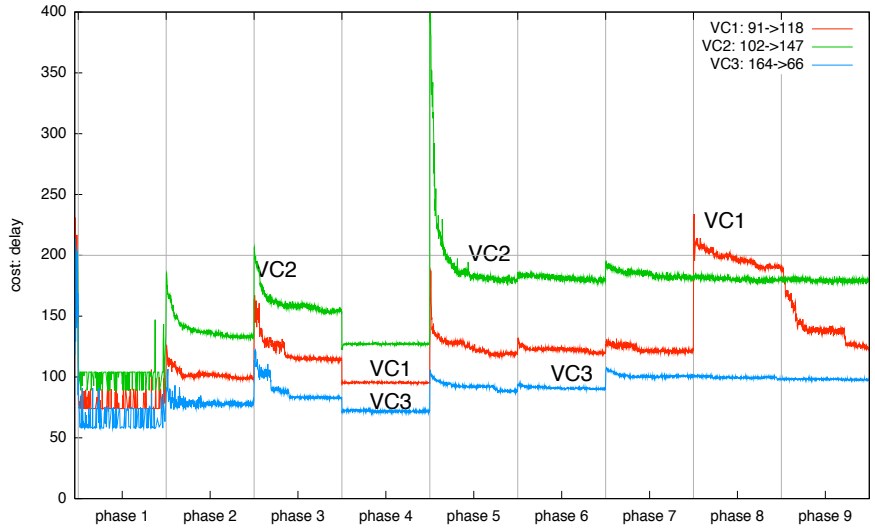

Figure 2. Adaptive path strategy: adaptivity in dynamic environment

and hence unavailable. E.g. if this delay bound is 200 ms (see horizontal grid line in Figure 2), the VC2 will at start of phase 3 and 5 experience a short interruption of the transport service. Measuring the unavailability, U, as the relative time with delay above 200, this gives $U_{VC1} = 0.036$, $U_{VC2} = 0.022$, and $U_{VC3} = 0.0$ for this simulation experiment. This unavailability can be reduced by increasing the number of updating messages per time unit, but this will increase the overhead of the management function.

2. If the increase in traffic load causes an overload on a link, the load sharing property of the adaptive path strategy will resolve this rather quickly. E.g. the sudden increase in traffic load of VC2 from 30 to 90% (phase 4 to 5) will cause a sudden peak in the cost value because one of the access links is overloaded. But, after a while (during phase 5), a new and good solution is found.

Primary backup strategy Figure 3 shows the results from the most illustrative simulation out of 10 replications applying the primary backup strategy for the scenario describe above. The cost of the operational paths for three selected VCs during the phases 5-9 are plotted. An operational path is either a primary or a backup path. The cost value function for the primary backup strategy is not sensitive to the carried load and hence the phases 1-5 from Table 1 are indistinguishable and represented by phase 5 in Figure 3. The cost value indicated at the y-axis is the loss penalty, as specified in Section 3.1. Note that the loss

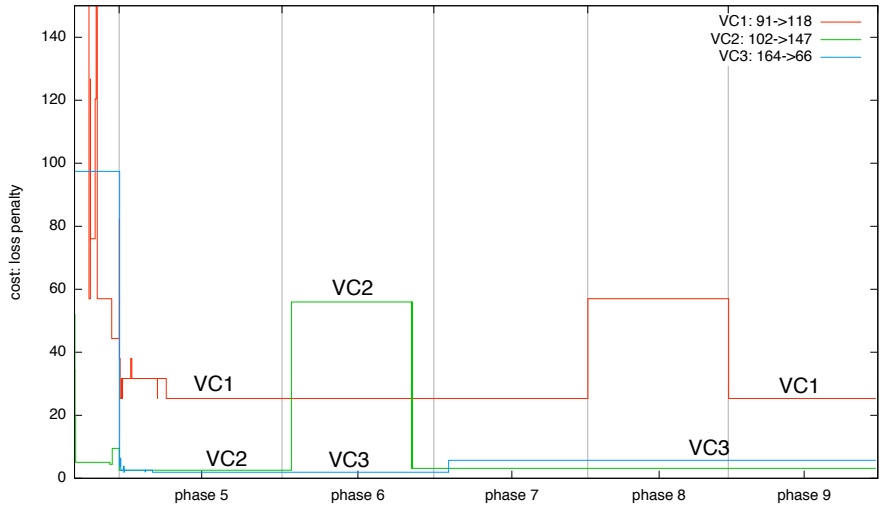

Figure 3. Primary backup strategy: adaptivity in dynamic environment

penalty is greater than 0 even if no traffic is lost. This is due to smoothening function in (2). Two lessons learned from the experiments are emphasised.

1. 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*. E.g. observe 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*). After a short period, the backup path takes over and is made operational. The backup path, which has a higher cost value, is operational until a new good primary path is found (primary is restored) at the end of phase 6. For VC1, the failed primary path at the beginning of phase 6 is quickly restored to a new primary path of equal cost (hence no shift in the curve in Figure 3), i.e. the restoration mechanism reacts faster than the protection switching mechanism. Again, from phase 7 to 8, the operational primary path becomes unavailable, but is very quickly restored to a new primary path (space between the vertical start-line of phase 8 and the cost curve for VC1 is almost not observable). Again restoration is faster than protection switching. The reason is that the nodes contains (in pheromone values) alternative primary paths that are almost immediately available, at least quicker than switching to the protection, or backup, path. The cost value is increased because the new best path needs extra hops to establish a path from the ingress to the egress node. Also for VC3 restoration works faster than protection switching, however as observed in the beginning of

phase 7 a more significant (and visible) delay is experienced, i.e. a down time, before a new primary path is found.

2. *Explicit link failure notification* will improve the path availability by making the protection switching mechanism more reactive. In the current implementation, no explicit notification of link failure is given to the ingress node of the path. The switch-over from primary to backup is triggered by a significant increase in the elite selection criterion from E.g.. (8) in Section 3.4. This type of "ant driven" failure reporting is robust, but may be inefficient because more than one ant is required to trigger and update. Even so, down times for VC1, 2 and 3 are short. The unavailabilities are $U_{VC1} = 0.0003$, $U_{VC2} = 0.012$, and $U_{VC3} = 0.018$.

4.3 Robustness

To test the robustness of the strategies two critical kinds of events are studied. First, *loss of information packages* (i.e. ants), and secondly, *loss of information* (i.e. pheromone values) in a node N_c along a specific primary path. As in previous experiments, we have studied the performance of the 10 paths using both the adaptive and primary backup path strategies.

In the first series of experiments, the management information was lost in all phases of the experiments, also in the exploration and transient phases. The strategies performed as if the number of ants where reduced and therefore the convergence rate was reduced, which is a desired and very robust behaviour indeed. The second series of experiments introduced failures after a path is established. These results for both loss of information packages and loss of information in a node are reported in this section.

Loss of information packages The ants are dropped on a specific interface that is a part of the preferred path for ingress node 194 and egress node 84. One of the interfaces of this path drops packets with a probability p_d. When $p_d = 1$ this is similar to a link failure and the method reacts as described in previous section. When $p_d < 1$ at least some of the searching (forward) and updating (backward) ants will get through and the pheromone values are updated. For the adaptive path strategy, if a single best path exists, it will remain the best even with $p_d > 0$. The reason is that the cost function does not reflect this performance degradation. However, when there are several paths with the same best value, the paths with packet loss will be updated less frequent than the paths without failure, and hence their pheromones will evaporate relative to the paths with less, or no, loss of ants.

Loss of information The second simulated failure mode is deleting all the pheromone values, i.e. removing all routing information in a specific node. This means that all interfaces of this node are affected. The specific node studied is a core node with 9 edges (interfaces). This node holds routing information about the preferred paths for 2 out of 10 VCs. When the routing information is removed, it means that an ant (and the data traffic) will be routed randomly according to a uniform distribution over the 9 available

interfaces. The probability of deleting pheromones is $p_f = 0.05$, which corresponds to that on average every 20th ants will meet an empty routing tables in this node. The main observation is that the best paths are retained and that loosing all routing information in one single node only causes minor problems. After very few ants (less than the average 20 in between node failures) the routing table is restored. This is because the neighbour nodes contain sufficient information to avoid an extensive exploration to re-establish the routing tables again. The adaptive strategy is more robust than the primary backup strategy because no explicit resource reservation and establishment of path is necessary. The primary backup strategy will suffer from the same problems as standard MPLS LSP management with respect to loss of soft state establishment (LSP) messages.

Based on the experiments in this section, it seems that both methods are robust to random loss of information packages (ants) and to loss of routing information (pheromones). In both cases, the paths are retained or restored quickly, without loss of consistency. As a general comment, the adaptive strategy seems robust to the random loss of any management information. This strategy is less sensitive to loss of specific control packets like the routing updates messaged, or LSP establishment messages you find in primary backup path strategy and in MPLS. The adaptive path strategy relies on small but redundant pieces of information. However, this redundancy comes with a price, and good and adaptive rules for managing the overhead must carefully be looked into.

5 Concluding Remarks

In this paper, we look at virtual path management in dynamic networks that poses a number of challenges related to combinatorial optimisation, fault and traffic handling. We claim that swarm intelligence based self-management is a promising candidate which reacts immediately to changes in the operational conditions, is autonomous, inherently robust and distributed, all necessary conditions to achieve operational simplicity and network resilience. Swarm intelligence achieved by elite CE ants is introduced and two path management strategies based on these are presented, denoted *adaptive path* and *primary backup*. A case study of a nation-wide communication infrastructure is presented to demonstrate the ability to handle change in network traffic as well as failures and restoration of links. The adaptive path strategy is designed to react quickly to loss and overload of resources. This reaction is demonstrated through the case study, in addition to a slower observable reaction when resources become available or underloaded. The latter is dependent on the number of ants used, i.e. the overhead. Note, however, that it is acceptable to operate on a sub-optimal solution for a short period as long as the prescribed QoS requirements are fulfilled. The primary backup is designed to guarantee, by establishing link disjoint primary and backup paths, that sufficient bandwidth is available if an arbitrary link fails. The case study demonstrates that fast switch-over to backup paths as well as fast

restoration of primary paths is possible. The case study also demonstrated that both methods are robust to loss of management state and updating information.

Further work includes continued work on the principles of applying emergent behaviour for managing QoS in networks, as well as dealing with engineering issues for introduction of these principles in operational networks.

References

1. M. O. Ball, *Handbooks in Operation Research and Management Science, Network Models*, vol. 7. North Holland, 1995.
2. M. Pióro and D. Medhi, *Routing, Flow and Capacity Design in Communication and Computer Networks*. ISBN 0125571895, Morgan Kaufmann Publishers, March 2004.
3. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science 220*, pp. 671–680, 1983.
4. F. Glover, *Tabu Search*. Kluwer, 1996.
5. D. Goldberg, *Genetic Algorithms in Search, Optimization and MachineLearn ing*. Addison Wesley, 1998.
6. R. Y. Rubinstein, "The Cross-Entropy Method for Combinatorial and Continuous Optimization," *Methodology and Computing in Applied Probability*, pp. 127–190, 1999.
7. ITU-T G.841 (10/98), "Types and characteristics of SDH network protection architectures," 1998.
8. ITU-T I.630 (02/99), "ATM protection switching," 1999.
9. C. Huitema, *Routing in the Internet*. Prentice Hall PTR, 2 ed., November 1999.
10. R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, "Ant-based Load Balancing in Telecommunications Networks," *Adaptive Behavior*, vol. 5, no. 2, pp. 169–207, 1997.
11. E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artifical Systems*. Oxford University Press, 1999.
12. G. D. Caro and M. Dorigo, "AntNet: Distributed Stigmergetic Control for Communications Networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, Dec 1998.
13. O. Wittner and B. E. Helvik, "Distributed soft policy enforcement by swarm intelligence; application to loadsharing and protection," *Annals of Telecommunications*, vol. 59, pp. 10–24, Jan/Feb 2004.
14. O. Wittner, *Emergent Behavior Based Implements for Distributed Network Management*. PhD thesis, Norwegian University of Science and Technology, NTNU, Department of Telematics, November 2003.
15. E. Rosen, A. Viswanathan, and R. Callon, "RFC3031: Multiprotocol Label Switching Architecture." IEFT, January 2001.
16. B. E. Helvik and O. Wittner, "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*, Springer Verlag, August 14-16 2001.
17. P. E. Heegaard, O. Wittner, V. F. Nicola, and B. E. Helvik, "Distributed asynchronous algorithm for cross-entropy-based combinatorial optimization," in *Rare Event Simulation & Combinatorial Optimization [RESIM2004]*, (Budapest, Hungary), September 7-8 2004.