Optimizing routing schemes for fast moving users in MST-based networks

Filip De Greve, Frederic Van Quickenborne,
Filip De Turck, Ingrid Moerman and Piet Demeester

Ghent University - IBBT - IMEC - Department of Information Technology
Gaston Crommenlaan 8, 9050 Gent, Belgium.
Email: {Filip.DeGreve,Frederic.VanQuickenborne}@intec.ugent.be

Abstract. With the currently emerging trials for best-effort internet solutions on trains, solutions are required for delivering multimedia services to fast moving users. Research has already been devoted to dimensioning Ethernet aggregation networks, taking user movement into account while neglecting the experienced network performance. This paper extends this design for resilient networks and aims at minimizing packet loss and packet reordering in the dimensioning and routing process. For deployment in an Ethernet network which supports Multiple Spanning Trees (MSTs), effective path aggregation methods are proposed for finding a minimal set of spanning trees and these are thoroughly evaluated for different scenarios. Moreover the spanning tree assignment problem with predefined backup conditions is studied.

Keywords: Ethernet, user mobility, spanning trees

1 Introduction

1.1 Motivation

The challenge telecom operators are facing is to examine how multimedia applications can be provided to users in fast moving vehicles. While satellite systems were the first best-effort solutions on the market, on-roof antenna architectures with WiFi/WiMax base stations located near the railroad track are recently gaining interest. We proposed such a WiMax-based architecture in [1] but more research is required for combining the performance that mobile hosts experience along their journey with a cost-effective design. The network must deliver packets to the correct point of attachment while minimizing the effects of packet loss (PL) and packet reordering (PR) that are caused when switching paths between access gateways. A lot of research has been performed in order to achieve seamless connectivity. At network layer methods have been proposed in order to improve performance degradation in case of packet reordering due to path delay variations or packet loss due to motion across wireless cell boundaries: e.g. triangular routing or bi-casting [2]. However to the authors' knowledge no work intends to relate the fast moving aspect of users with the cost-effective design of the network infrastructure. If this work is combined with the above mentioned techniques, it will lead to lower buffer capacities, less forwarding overhead or improved real-time behavior.

1.2 Contribution

We present a design technique for resilient Ethernet networks which tries to minimize PL and PR in the dimensioning and routing process. Because the flow routing is optimized, we still have to find a way of enforcing these paths in a multiple spanning tree (MST) environment [3]. In a MST network the available paths must be placed on one of the active Spanning Tree Instances (STIs). Classic spanning tree problems try to find optimal routes within a single spanning tree: e.g. resource-efficient transport of multi-cast traffic [4] [5] but the problem of finding multiple STs is rarely studied. Heuristic techniques are presented which aim to find a minimal set of trees covering the set of paths calculated in the design phase. In [6] we presented a similar approach (see Fig. 1) without taking network reliability into account. In this paper we will extend this study: design of networks that are resilient against single network failures requires more resources and the path assignment problem on a minimal set of STIs is more difficult because recovery is based on ST and not on a path-per-path basis.

1.3 The system architecture

Our FAMOUS (= FAst MOving USers) network architecture is illustrated in Fig. 2. There is a single Service Gateway (SGW) and a number of Access Gateways (AGWs) which are placed along the railroad tracks. Service guarantees can be assured by making on-time resource reservations. Data connections of fast moving users will be mapped on VLAN tunnels that are responsible for the delivery to the correct AGW in the aggregation network. The VLANs are fixed end-to-end tunnels, automatically installed with GVRP (GARP VLAN Registration Protocol). At their due time, tunnel reservations are registered for the aggregated flow, but only shortly before the flow will be effectively using the VLAN tunnel. When reservations are no longer required, they are immediately released. In this way the system will guarantee that the current and the next hop tunnel are able to maintain the service level and that capacity bottlenecks are prevented. For further details on the system architecture we kindly refer to [1].

This paper is organized as follows: first the influence of path switching and the mathematical model for cost-effective network design while taking user movement into account are presented. In Sec. 4 the network costs are evaluated. Next path aggregation techniques are proposed for obtaining a minimal set of trees. In the final section these techniques are extended for backup scenarios.

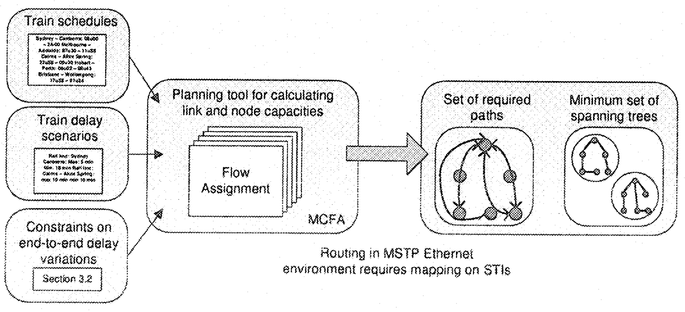


Fig. 1. Design approach of Ethernet environment for fast moving users.

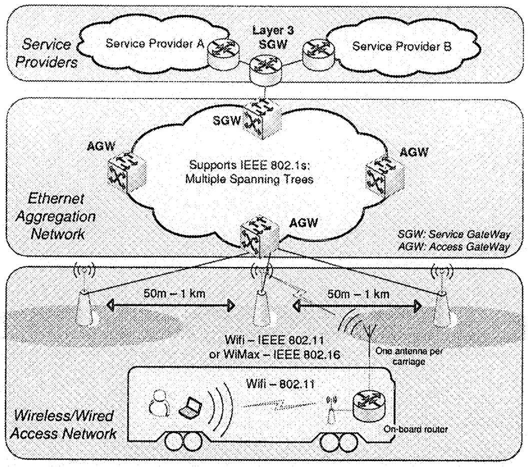


Fig. 2. Schematic representation of the network architecture - specifically designed to cope with requirements of train passengers.

2 Path switching

2.1 Influence on TCP performance

First we studied the PL and PR which occurred at aggregation network level when switching between paths with different end-to-end delays on a Click/Linux test bed [7]. For CBR flows, the PL curves can be approximated as follows:

$$PL = \Delta \cdot BW/S. \tag{1}$$

with BW = the flow bandwidth and S = the packet size and Δ = decrease in e2e delay. Note that no pause in communication is assumed at aggregation network level: the losses represent theoretical minima that can be achieved depending on the performance of the implemented handover mechanism. Still, the only parameter in (1) that can be tuned at network design level, is the delay variation Δ. For non-reliable real-time applications, it are indeed the variations in e2e delays that are perceived as critical for human interaction. However, this assumption is not true for all application types. On Fig. 3 the influence of delay variations and connectivity disruptions during path switching on the TCP performance is evaluated. We assume that the handover period when the train's connection is moving from one AGW to another AGW, is short compared to the connection time to a single AGW. It shows that TCP throughput will not degrade severely due to delay variations in a train scenario environment but that absolute end-to-end delays are more important. Delay variations of several tens of milliseconds are not considered harmful for TCP sessions. The time-averaged end-to-end delays during the connection are more important for the TCP throughput performance. Due to the fact that we assume a low congested network operation by making resource reservations, absolute end-to-end delays will not form a bottleneck on the TCP performance. For the remainder of the paper we will further assume applications which are sensitive to e2e variations such as real-time traffic.

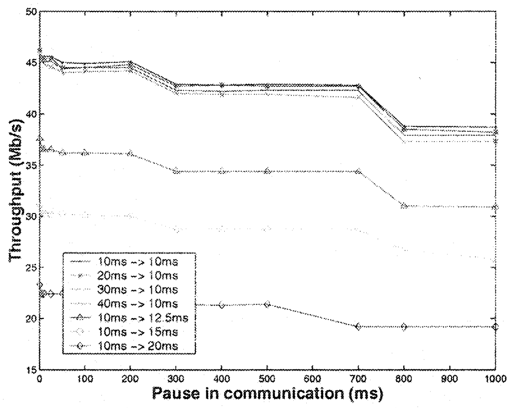


Fig. 3. Influence of end-to-end delay variations on TCP throughput.

2.2 Motivation of routing schemes for reducing PL and PR.

In this section we will motive that PL and PR can be reduced in the design phase. When the source-destination path is altered, packets that are on their way on the old path, run the risk to be reordered or lost. It is clear that packets might be lost in the *down* direction (from SGW to AGW) while packets might be reordered in the *up* direction (from AGW to SGW), not vice versa. The amount of affected packets depends on the implemented trigger mechanism that correlates the train positioning with the tunnel switching. For the topologies of Fig. 4 we measured the PL and PR when switching tunnels from node 3 to node 4 for a 10 Mb/s flow with a packet size of 128 bytes. We illustrated the principle for a 5ms delay per link. In a second phase we activated the impairment node 2 (=20ms additional delay). Tunnel switching was always triggered 3ms after the successful delivery of all packets destined for node 3. The results are depicted in Table 1. The first network has equal e2e path delays but due to the fact that tunnel switching cannot occur at both ends of the tunnel at the exact moment, PL is to be expected. The second network does not have equal e2e path delays but still has similar results for PL; PR however is worse. The third network has increasing e2e path delays, therefore the tunnel switching mechanism has a bigger margin for making a lossless swap decision. However, a network design with ever increasing path delays remains not preferable and unscalable. This fact reveals

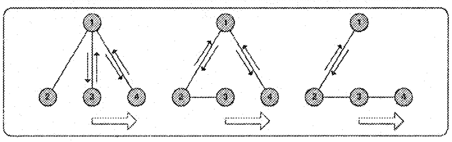


Fig. 4. Three example network topologies.

Table 1. Packet loss (PL) and packet reordering (PR) due to path switching for the topologies of Fig. 4.

	Network 1	Network 2	Network 3
PL ↓	2	2	0
PR ↑	0	53	0
With impairment node 2			
PL ↓	12	12	0
PR ↑	208	265	0

that PL or PR can of course never fully be eliminated in the design phase but that intelligent techniques are required in order moderate the e2e path variations at the moment of tunnel switching. Now, which schemes were used to design the network of Fig. 4 ? Network 1 can be designed by applying shortest path routing: this network has a good delay performance but is typically not the cheapest solution. Network 2 is a more cost-effective solution which has similar characteristics. It has only limited delay variations and was designed with limited hop count variations (see Sec. 3.2). Network 3 was designed with the shared routing constraints (see Sec. 3.2). Routing schemes which promote shared routing, have the main advantage that impairment nodes or congested nodes (typically situated closer to the aggregation point) in the shared part have less impact on the applications' performance.

3 Motion-aware Capacity & Flow Assignment Problem

3.1 Theoretical Model

In previous work we proposed a theoretical model for calculating the dynamic tunnels that meet the traffic demands and for optimizing the utilization of the network resources: a Motion-aware Capacity and Flow Assignment (MCFA) problem. In this paper we add reliability constraints to the ILP formulation.

Decision variables The following variables give information about the number of fibres and the usage of every link l and the number of line cards available in each node:

$$u_l = \begin{cases} 1, & \text{if link } l \text{ is used} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$x_l^v = \# \text{ of fibres with link speed } C_v \text{ on link } l. \tag{3}$$

$$z_n^{vw} = \# \text{ of Ethernet cards with } O_{vw} \text{ interfaces of speed } C_v \text{ on node } n. \tag{4}$$

Each card has a specific cost, c_{vw}: v indicates the speed type and defines the speed C_v of the interfaces while w indicates the interface type of the card and defines the number of interfaces O_{vw} available on the card.

Next we define a set of possible paths $P_{ik} = \{p_{ikq}\}$ in which index q is used to indicate the different considered paths between AGW i and SGW k. They are calculated by taking M candidate paths between the two end nodes. We define a variable to indicate which path $p \in \bigcup_{i,k} P_{ik}$ is used:

$$y_{pijk} = \begin{cases} \text{1 if path } p \text{ is used between AGW i} \\ \text{and SGW } k \text{ for flow } j \\ 0, \text{ otherwise} \end{cases} . \tag{5}$$

Node capacity constraints Every node needs enough interfaces with appropriate link speeds to provide the links that are connected to it:

$$\forall n, \forall v : \sum_{w=1}^{|w_{max}|} z_n^{vw} \cdot O_{vw} \geq \sum_{l \text{ incident to node } n} x_l^v \tag{6}$$

Link Capacity constraints This constraint sets the capacity of each link and imposes that the traffic that is transported over a link does not exceed the capacity of that particular link at any time. Traffic demands are defined as $d_{ijk}(t)$: the demand at time t between AGW i and SGW k for flow j.

$$\forall l : \sum_k \sum_i \sum_j \sum_p y_{pijk \cdot \delta_{pl}^{ik} \cdot d_{ijk}(t)} \leq \sum_v x_l^v \cdot C_v \tag{7}$$

$$\delta_{pl}^{ik} = \begin{cases} 1, \text{ if path } p \text{ uses link } l \text{ to get to} \\ \text{destination k from source } i \\ 0, \text{ otherwise} \end{cases} \tag{8}$$

Path Activation constraints This constraint takes care of the fact that we only need a single path for a specific flow j for every SGW-AGW pair:

$$\forall i, \forall j, \forall k : \sum_p y_{pijk} = 1 \tag{9}$$

Reliability constraints Reliability is taken into account by providing a second dedicated backup path, disjunct from the working path. Assume you have N_T trains running on the railroad track, redundancy can then be modelled by taking two flows per train: a *working* flow (2j') and a *backup* flow (2j'+1) with the condition that flows (2j') and (2j'+1) use disjunct paths. Assume that between every AGW i and every SGW k M possible paths are available. Suppose that for every path p_{ikq}, element of P_{ik}, n paths ($0 \leq n \leq M$) of P_{ik} are not fully node and link disjoint. The collection of these paths is denoted as $P_{P_{ik}}^{ND}$. Working flows and backup flows for the same train must be routed on disjunct paths. To assure a correct dimensioning in case of single failures, the following constraints must be added to the problem formulation:

$$\begin{array}{l} \forall j' : 0 \ldots N_T \\ \forall i, \forall k, \forall p_{ikq} \in P_{ik} : p_{ik1}, p_{ik2}, \ldots, p_{ikn} \in P_{P_{ik}}^{ND} : \\ y_{p_{ikq} i 2j' k} + y_{p_{ik1} i (2j'+1) k} \leq 1 \\ y_{p_{ikq} i 2j' k} + y_{p_{ik2} i (2j'+1) k} \leq 1 \\ \ldots \\ y_{p_{ikq} i 2j' k} + y_{p_{ikn} i (2j'+1) k} \leq 1 \end{array} \tag{10}$$

Objective function o

$$
\begin{aligned}
o = \alpha \sum_{l} u_l \cdot c_l + \beta \sum_{n} \sum_{v} \sum_{w=1}^{|w_{max}|} z_n^{vw} \cdot c_{vw} \\
+ \gamma \sum_{k} \sum_{i} \sum_{j} \sum_{p} y_{pijk} \cdot HopCount_p
\end{aligned}
\tag{11}
$$

The first term represents the cost to install fibers, the second represents the cost to install network equipment with sufficient interface cards and the final term represent the sum of hop counts of all required SGW-AGW tunnels (= tiebreaker term).

3.2 Routing schemes

In this section routing schemes are proposed which improve the performance of fast moving users. These schemes will lead to extra constraints that have to be added to the MCFA (Motion-aware Capacity Flow Assignment) problem as previously presented.

Limited hop count variations routing (LHCV): The experienced hop count variation along the trajectory is limited: less than or equal to HC_{diff}. The following constraints are added for the path hop counts HC_p:

$$
\begin{aligned}
&\forall i, \forall j, \forall k, \forall p \in P_{ik}, \forall q \in P_{i_{next}k} : \\
&\sum_{p} y_{pijk} \cdot HC_{pijk} \\
&- \sum_{q} y_{qi_{next}jk} \cdot HC_{qi_{next}jk} \leq HC_{diff}
\end{aligned}
\tag{12}
$$

Shared routing: Previous routing schemes still have the disadvantage that different physical routes are chosen with inherent different congestion levels. Therefore an approach is developed which favors route sharing in the design. One of the following constraints is added for every connection:

- (i) The current path of the connection must share at least the first $HC_{previous} - K$ nodes with the previous path of the connection
- (ii) The previous path of the connection must share at least the first $HC_{current} - K$ nodes with the current path of the connection.

with HC_{path} as the path hop count used for a connection and the parameter K for tuning the routing's strictness.

Incremental routing: This is a special case of the previous routing scheme with non-decreasing path lengths. It has the advantage that PR is unlikely to occur.

4 Reliability constraints vs. routing schemes

In this subsection the network cost of single rail line networks will be optimized in terms of required networks cost with respect to PL and PR constraints and reliability constraints. We considered several scenarios with up to 4 trains, varying traffic loads per train ranging from 500 Mb/s up to 1.5 Gb/s, time schedule variations and low

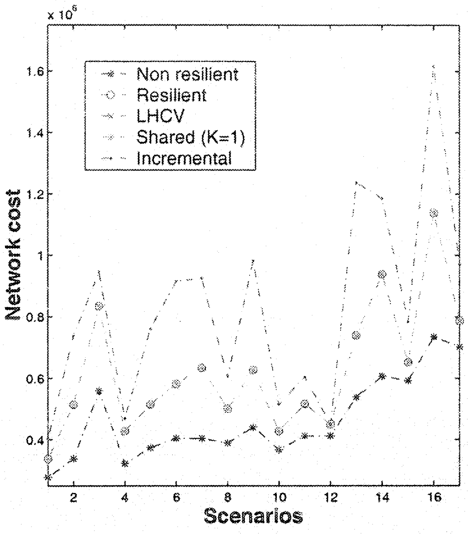


Fig. 5. Influence of routing schemes on the network cost and the set of required paths for single rail line scenarios, ordered from left to right for increasing number of AGWs.

Table 2. Averaged amount of required tunnels for the scenarios of Fig. 5.

	Without resilience	Resilient	LHCV	Shared (K=1)	Incremental
Number of tunnels	10.67	15.5	14.4	10.3	8.5

installation cost of fibers - or dominant routing cost - in order to focus on the moving user aspect ($\beta \gg \alpha$, $\beta \gg \gamma$ in equation 11). The results with M (= number of parallel candidate paths between every 2 nodes) high enough to find optimal solutions are presented in Fig. 5. Figure 5 shows up to 60% additional costs for making the network resilient (35% on the average). As can be expected incremental routing turns out to be very resource consuming which makes it not suitable for large-scale telecom networks. Surprisingly in most cases addition of the constraints for LHCV and shared routing do not further increase the cost of the network solution. This can be explained by the effect that the network is ideally provisioned with knowledge of the moving trains trajectory: examination of the network solutions for ideal routing, reveal that the hop count variations between two consecutive working tunnels (connecting to two adjacent AGWs) for a specific tunnel are often already small. And similar: in case of LHCV routing in many cases two consecutive tunnels of a working flow will already share parts of their route. However while the network costs are equal, different paths are selected to satisfy the additional constraints for the working flows as is represented in Table 2. Obviously adding resilience constraints increases the required amount of tunnels. The most tunnels are required for resilient networks without additional constraints. Adding more routing constraints leads automatically to a more limited set of available paths which is shown by the decreasing amount of required tunnels if the routing scheme becomes more severe: therefore the most severe routing scheme (i.e. incremental routing) uses the least paths.

Routing constraints ensure that hop count variations are confined during the entire journey of the users. Of course non-ideally designed networks will experience a lower throughput with such routing schemes. We can conclude that by adding simple constraints - according to one of the presented routing schemes - to the MCFA problem, lower delay variations can be attained during path switching. In case of dominant routing cost this can be achieved without increasing the network cost significantly.

5 Path aggregation technique for calculating a minimal set of spanning tree instances (STIs)

Deployment of routing schemes in Ethernet networks requires mapping of the required routes as calculated for the MCFA problem on VLANs and STIs. The IEEE 802.1s standard does not specify details of mapping VLANs on one or more STIs. The amount of VLANs that need to be configured is determined by the number of AGWs (Access Gateways) and the number of parallel paths to each AGW. Within aggregation network scope, the amount of required VLANs will clearly remain under the 4096 upper limit. For the ST assignment we present a path aggregation (PA) algorithm that forms a minimal set of STIs. We compare our heuristic with a *Random Assignment* (RA) which aggregates paths randomly in loop-free sets of trees and with a PA heuristic that tries to merge paths which share a common feature e.g. pair of edges. This heuristic was introduced in [8] and we will refer to this method as the *Shared Assignment* (SA).

Algorithm 5.1: SUB-PATH ASSIGNMENT(*STIs*)

```
Set of all paths : $\Phi \leftarrow (\mathrm{p}_0, ..., \mathrm{p}_\mathrm{m})$
$\Phi \leftarrow order(\Phi)$
comment: in descending order of path length
Set of all path pairs : $\mathrm{PP} \leftarrow \{(\mathrm{p}_0, \mathrm{p}_1), ..., (\mathrm{p}_{\mathrm{m}-1}, \mathrm{p}_\mathrm{m})\}$
Set of path pairs taking part of same STI : $\mathrm{PP}_\mathrm{same} = \emptyset$
Set of all spanning trees : $\mathrm{T} = \emptyset$
while $(\exists\{p_k, p_l\} \in PP)$ true
do
    comment: Sub-path substitution step
    Remove$\{\mathrm{p}_\mathrm{k}, \mathrm{p}_\mathrm{l}\}$fromPP.
    if $p_k \subseteq p_l$
        Add $\{\mathrm{p}_\mathrm{l}, \mathrm{p}_\mathrm{k}\}$ to $\mathrm{PP}_\mathrm{same}$.
        Remove $\mathrm{p}_\mathrm{k}$ from $\Phi$.
    if $p_l \subseteq p_k$
        Add $\{\mathrm{p}_\mathrm{k}, \mathrm{p}_\mathrm{l}\}$ to $\mathrm{PP}_\mathrm{same}$.
        Remove $\mathrm{p}_\mathrm{l}$ from $\Phi$.
while $(\exists p \in \Phi)$ true
do
    Remove p from $\Phi$
    if $(\exists \mathrm{t} \in \mathrm{T}$ :checkloop$(\mathrm{p}, \mathrm{t}) ==$ true )
      then Merge p with t and
    $\forall$ q with $(\mathrm{p}, \mathrm{q}) \in \mathrm{PP}_\mathrm{same}$ : Merge q with t.
      else Add p to new tree in T and
    $\forall$ q with $(\mathrm{p}, \mathrm{q}) \in \mathrm{PP}_\mathrm{same}$ : Merge q with new tree.
```

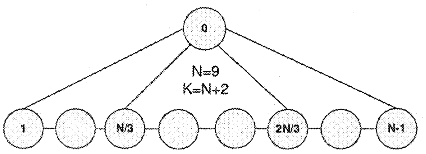


Fig. 6. Aggregation test network with amount of AGW-SGW links limited to 4.

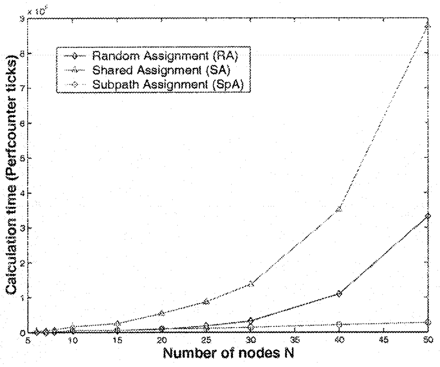


Fig. 7. Calculation times for the PA problem for aggregation networks such as depicted in Fig. 6.

For k=1

SpA	SA	RA
1	1	1

For k=2

SpA	SA	RA
2	3	4

For k=3

SpA	SA	RA
3	3	5

For k=4

SpA	SA	RA
4	4	6

Fig. 8. Number of STIs for aggregation networks (depicted in Fig. 6) and with k-shortest paths active to every AGW.

Joining paths with shared features is a good idea but provides no absolute guarantee to obtain a minimal set. For an aggregation network that supports fast moving users, typically a lot of the paths will be sub-paths of other paths. With this knowledge, the idea is to aggregate those paths on the same STI - *Sub-path Assignment* (SpA) - and this is in fact a more relevant condition for obtaining a minimal set. The pseudo-code of the SpA algorithm is presented in Algorithm 5.1.

The remainder of this section is dedicated to the evaluation of the heuristic's performance in large test sets. In [6] we already presented performance tests but we allowed the same path to be selected more than once. This clearly favored SpA because SpA specifically makes sure that these paths are assigned to the same tree. In all the tests of this paper we didn't allow the same path to be selected twice for tree assignment. It will be shown that under these conditions SpA remains the most favorable heuristic method. First we examine the calculation times[1] for aggregation networks with various rail lengths and the amount of SGW-AGW links limited to four (as illustrated in Fig. 6). The calculation times and the minimal number of STIs are presented in Fig. 7 and Fig. 8. For this simple topology and for k parallel paths ($1 \leq k \leq 4$) to every AGW, SA and SpA find similar solutions but the SpA takes less time, even for larger problems it takes less time than RA due to the fact that less loop checks are required. However RA itself doesn't succeed in finding the optimal solution. This result confirms

---

[1] In the simulation a single perfcounter tick equals 0.3μ sec

the assumption that SpA finds the best non-optimal solution in the shortest time for aggregation network scenarios. The loop check calculations are performed every time paths from the ordered set Φ are incrementally assigned to a specific tree. Paths are expressed as an ordered vector of nodes. A tree is represented as a subset of loop-free paths from Φ which are part of the tree. We call this the Path Composition representation. This is actually quite similar as the Predecessor representation [9]. Additionally the set of nodes is maintained which form disconnected subtrees, part of the actual tree. This enables loop avoidance to be verified easily if paths are added incrementally. Intermediate tree solutions are not necessarily STs (i.e. covering all nodes) or are not necessarily fully connected (during the process a tree may consist of two disconnected subtrees). Before every assignment to a tree it is verified that the path and subtree it will be connected to, do not share a pair of nodes unless all nodes on the path between this pair of nodes are also shared. This loop check is $O(N^2)$. Checking loops will remain to be effortsome unless alternative tree notations such as Prüfer numbers [9] are used. However these representations lack locality and heritability. In other words very alike tree representations do not consist of similar substructures. This would require a lot of transformations between trees and Prüfer numbers (typically O(NlogN)) because it is crucial to verify that all paths of set Φ are covered by the set of STIs.

Characteristic Vector representations of trees which can be used in ILP-based formulations lack the probability of representing an actual tree. If the network contains N nodes and K edges, a tree T can be presented as a vector E = e_k, k=1 ... K and e_k is one if edge k is part of T and zero otherwise. In a fully meshed network $2^{N(N-1)/2}$ possible values exist for E. In [10] it is shown that a complete graph of N nodes contains N^{N-2} spanning trees. This means that the chance of having an E value which represents a tree is $\frac{2^{\frac{N(N-1)}{2}}}{N^{N-2}} = 2^{-N(N/2-log_2 N+2(log_2 N)/N-0.5)}$. For N=10 this chance equals 2.84E-06. Standard predecessor tree notations represent a tree with a probability 1/N. Moreover we have to represent a multiple tree structure (containing up to $| \Phi |$ trees). The chance of representing $|\Phi|$ valid trees is $2^{-|\Phi|N(N/2-log_2 N+2*(log_2 N)/N-0.5)}$ for the characteristic vector notation. For $|\Phi|$=5 and N=10 this equals 7.35E-10. It is clear that there are too many representations which are not trees. In our representation only $|\Phi|^{|\Phi|}$ representations are possible. For $|\Phi|$=10 and N=10 there are 5.5E+308 times more Characteristic Vector representations than Path Composition representations. This clearly shows the inefficiency of Characteristic Vector representations.

What about the algorithm's performance in other scenarios? In Tables 3 and 4 the comparison is made for grid networks (4x4, 5x5, 6x6 dimensions) and for mesh networks with a randomly assigned route pattern. The route pattern is chosen from a set of shortest paths between the node pairs. This makes it able to define a path density λ which indicates how many paths are selected. λ=1 means that all paths between the node pairs are selected for the PA problem: for 4x4, 5x5 and 6x6 grids this corresponds respectively to 120, 300 and 630 paths. As can be derived, SpA performs better for mesh networks: up to 14.7% less trees for λ=1 compared to SA. For low λ values SpA and SA are quite similar because the chance of finding paths with similar features starts to decrease. RA doesn't succeed in finding good solutions. For the symmetric full mesh networks, SpA and SA find the same solution and RA solutions remain competitive because under these conditions SpA and SA don't succeed in finding paths with similar features at all: all paths are direct links and no links are selected twice. While both heuristics are not designed to operate properly in these conditions, the calculated solutions approximate the optimal solution. The optimal solution was derived by experimental validation. For a full mesh topology and λ=1 the following

Table 3. Grid networks: amount of STIs and calculation times for different path aggregation techniques.

Grid		SpA		SA		RA	
Size	λ	STIs	t	STIs	t	STIs	t
(4x4)	0.04	1.6	1193	1.6	19920	1.66	966
	0.08	2.14	2681	2.2	32381	2.48	1077
	0.25	4.06	12624	4.1	76791	5.06	3922
	0.42	5.3	27522	5.4	139788	7.36	8827
	0.83	7.36	85733	7.94	3.1e5	11.88	27270
	1	7.78	1.1e5	9	3.8e5	13.22	36513
(5x5)	0.03	2.16	3132	2.2	78905	2.3	1411
	0.17	5.4	31892	5.52	3.7e5	7.08	12928
	0.33	7.84	99998	8.24	7.6e5	11.62	40264
	0.67	11.08	3.6e5	12.4	1.7e6	18.84	140757
	1	13.02	7.1e5	15.18	2.8e6	24.68	2.9e5
(6x6)	0.02	2.36	2922	2.26	1.8e5	2.42	1723
	0.06	4.94	27349	5.08	6.5e5	6.02	16843
	0.16	8.44	1.2e5	8.76	1.7e6	11.82	61576
	0.32	12.36	4.0e5	12.9	3.6e6	18.76	2.0e5
	0.63	17.06	1.4e6	18.68	8.7e6	30.36	7.1e5
	1	20.23	3.4e6	23.73	1.7e7	40.86	1.6e6

Table 4. Mesh networks (λ=1): amount of STIs and calculation times for different path aggregation techniques.

Mesh	SpA		SA		RA		Exact
N	STIs	t	STIs	t	STIs	t	STIs
3	2	1333	2	1796	2	1187	2
4	2.16	1380	2.16	3110	2.2	916	2
5	3	3085	3	6496	3.02	549	3
6	3.6	3677	3.6	17633	3.66	1063	3
7	4.04	5714	4.04	37898	4.06	1153	4
8	4.68	9363	4.68	76424	4.8	1949	4
10	5.92	21449	5.92	3e5	5.92	2970	5
15	8.28	90940	8.28	3.5e6	8.28	10852	8
20	10.92	2.8e5	10.92	2.11E7	11	31489	10

formula for the minimal amount of STs can be derived:

$$Number\ of\ trees = \begin{cases} \frac{N}{2}, & \text{if N even.} \\ \frac{N+1}{2}, & \text{if N uneven.} \end{cases} \tag{13}$$

Due to the fact that SpA can't profit from the amount of shared paths that are typically found in the case of aggregation networks, SpA is no longer faster than RA in these scenarios. We can conclude that the rather simple SpA heuristic manages well to tackle the essence of the path aggregation problem.

6 Path aggregation technique for calculating a minimal set of STIs with predefined backup conditions

Classic PA techniques which aim to identify the most efficient spanning tree often do not take into account that this trees can reconfigure after network failures. Therefore we try to solve the more difficult PA problem for ST-based recovery with predefined backup routes. We will compare the amount of STIs with a centralized backup system (which doesn't rely on ST-based recovery such as [8]): the paths Φ are the working paths and if failures are detected by the central management system, the affected working paths are switched to predefined backup paths. This will be referred to as *Centralized Backup PA*. We will compare both techniques on their STI usage. The additional STIs that are required to fulfill the backup constraints are also examined: intuitively it is clear that backup conditions might double the amount of required trees in worst-case scenarios (or even more). We will verify this assumption for randomly generated traffic patterns and various network topologies. In Algorithm 6.1 an heuristic is presented for finding a set of STIs which aggregate paths with predefined backup paths for ST-based recovery. It is based on the SA algorithm instead of SpA because the sub-path substitution step of SpA can no longer be applied under backup conditions. Therefore the heuristic is referred to as *Backup Shared Assignment* (Backup SA).

Algorithm 6.1: BACKUP SHARED ASSIGNMENT(*STIs*)

```
Set of all paths : $\Phi \leftarrow (p_0, ..., p_m)$
Set of all backup paths : $\Gamma \leftarrow (b_0, ..., b_m)$
$\Phi \leftarrow order(\Phi)$
comment: in descending order of path length
Set of all edge pairs : $EP \leftarrow \{(e_0, e_1), ..., (e_{K-1}, e_K)\}$
$EP \leftarrow orderfrequency(EP, \Phi)$
comment: in descending order of their frequency of appearance in members of $\Phi$.
Set of all spanning trees : $T = \emptyset$
while ($\exists\{e_k, e_l\} \in EP$) true
  do while ($\exists p \in \Phi$ and $\{e_k, e_l\} \subset p$) true
       do Remove p from $\Phi$
          if ($\exists t \in T$ : ((checkloop(p, t) == true )
          and (checkloop(p, b, t) == true for all failures scenarios.)))
            then Merge p with t.
            else Add p to T.
```

In Fig. 9 the calculation times for aggregation networks are depicted. As shown in Sec. 5 the problem is quite easy to solve for the shortest paths, therefore this time the shortest or second shortest path are randomly selected as working path. The path density λ is varied by changing the network size and keeping the number of paths equal. For larger network sizes (or smaller λ) the calculation time of all heuristics will rise. However it is clear that backup PA techniques increase more because in every step the PA problem has to be solved for all possible link or node failures. This means Max(O(N);O(K)) PA problems more have to be calculated compared to non-resilient PA techniques at every step. Figure 10 shows clearly that in aggregation networks

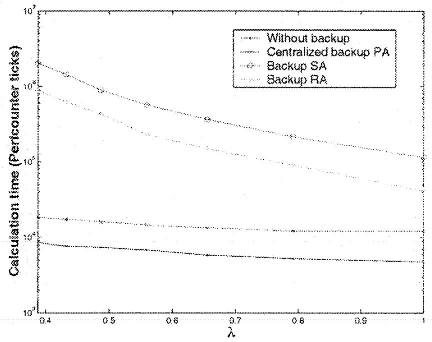


Fig. 9. PA calculation times for the aggregation networks with predefined backup paths and fixed amount of paths.

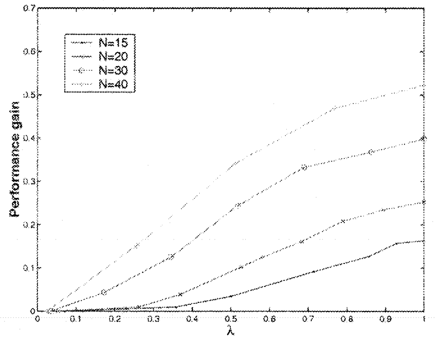


Fig. 10. The reduction of the required amount of STIs of Backup SA compared to Backup RA for aggregation networks.

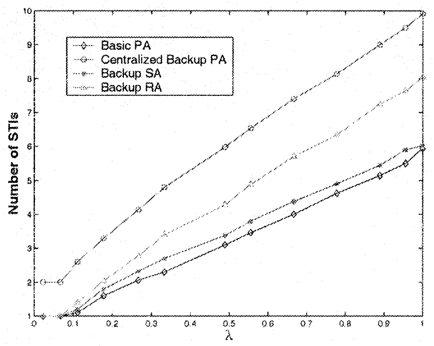


Fig. 11. Path aggregation for mesh networks with predefined backup routes.

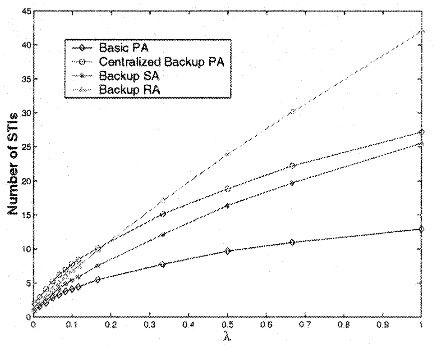


Fig. 12. Path aggregation for grid networks with predefined backup routes.

the Backup SA heuristic outperforms the Backup RA: for increasing network size and λ the gain increases. Compared to the centralized backup PA up to 20% more trees are required in aggregation networks. Compared to non-resilient PA the amount of supplementary trees increases with the network size: from 44% to 76% for λ=1. This would indicate that centralized backup PA requires less trees than Backup SA. However the calculations for mesh and grid topologies don't confirm this finding.

Calculations for a mesh network with N=10 and a 5x5 grid network are presented in Fig. 11 and Fig. 12. Similar as in aggregation networks the performance gain of Backup SA compared to Backup RA increases for increasing λ and increasing network size. In comparison to previous results the Centralized Backup PA requires more STIs compared to Backup SA for both mesh and grid networks: 100% to 61% additional STIs for the mesh network and 100% to 6% additional STIs for the grid network. This means that Centralized Backup PA doesn't necessarily require less STIs than ST-based recovery mechanisms. Therefore we repeated the tests for uncut aggregation networks with N-1 AGW-SGW links. Wether we selected the shortest paths, the 2nd, the 3rd shortest paths or combinations as working paths, in all cases the Centralized Backup PA method required up to 38% more STIs compared to Backup SA. Therefore we can conclude that in most cases ST-based recovery will require less trees because Centralized Backup requires separately configured trees for working and backup paths. The number of supplementary trees which are necessary to fulfill the backup conditions can also be derived: for the mesh network barely up to 10% extra STIs are required and for the grid the amount of trees is almost doubled. This strengthens the assumption that backup conditions can double the amount of required trees in worst-case scenarios but our calculations prove that in many cases far less additional trees are required.

7 Conclusions

We presented a cost-effective design technique for resilient aggregation networks which minimizes packet loss and packet reordering in the dimensioning process. It is shown that the additional resource usage of the presented routing schemes remains limited for aggregation networks. We presented efficient techniques for mapping the required paths on a minimal set of STIs. Evaluation of these methods in different scenarios proved that SpA performed well for aggregating working paths. Evaluations on path aggregation techniques with predefined backup conditions proved that ST-based recovery is truly a flexible way of providing efficient network recovery and that expensive and vulnerable centralized backup systems require rather more STIs for similar backup routes because they require separately configured trees for working and backup paths.

References

1. De Greve F., Van Quickenborne F., et al., *FAMOUS: A Network Architecture for Delivering Multimedia Services to FAstMOving USers*, Special Issue of the Wireless Personal Communications Journal Vol. 33, Numbers 3-4, pp. 281 - 304, 2005.
2. Campbell A.T., Gomez J. and Valko A., *An overview of cellular IP*, IEEE Wireless Communications and Networking Conference, pp. 606-611, 1996.
3. IEEE, *IEEE Standard for Local and Metropolitan Area Networks: Multiple Spanning Trees*, 2002.
4. Li Y., Ooi W.T., *Distributed construction of resource-efficient overlay tree by approximating MST*, The proceedings of ICME 2004, pp. 1507-1510, June 2004.
5. Dziong Z., Jia M., Mason L.G., *ATM Workshop Proceedings, 1998 IEEE*, pp. 186-194, 1998.
6. De Greve F., Van Quickenborne F., et al., *Cost-effective Ethernet routing schemes for dynamic environments*, IEEE Globecom, Dec 2005.
7. Kohler E. , Morris R., Chen B., Jannotti J. and Kaashoek M. F., *"The Click modular router", ACM Transactions on computer systems*, Volume 18 number 3, pp. 236-297, 2000.
8. Sharma S., Gopalan K., et al., *Viking: A multi-spanning tree Ethernet architecture for metropolitan area and cluster networks*, IEEE Infocom 2004, Mar 2004.
9. Palmer, C., *Handbook of Evolutionary Computation*, Oxford University Press, 1997.
10. A. Gibbons, *Algorithmic Graph Theory*, Cambridge University Press, 1985.

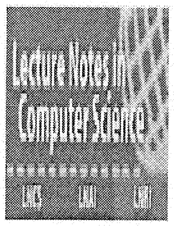
SPRINGER | FEEDBACK | FAQ | JOBS & CAREERS

SEARCH FOR RETURN Articles GO ADVANCED SEARCH

ABOUT | BROWSE | FAVORITES | ALERT | ORDERS LOG OFF

> Home / Publication / Volume /

Chapter



Lecture Notes in Computer Science
Publisher: Springer Berlin / Heidelberg
ISSN: 0302-9743
Subject: Computer Science
Volume 3883 / 2006
Title: Wireless Systems and Network Architectures in Next Generation Internet: Second International Workshop of the EURO-NGI Network of Excellence, Villa Vigoni, Italy, July 13-15, 2005, Revised Selected Papers
Editors: Matteo Cesana, Luigi Fratta
ISBN: 3-540-34025-4
DOI: 10.1007/11750673
Chapter: pp. 4 - 20
DOI: 10.1007/11750673_2

Previous chapter
Next chapter

Export Citation: RIS | Text

Linking Options

Send this article to
an email address
Go

Quick Search
Search within this publication...
For:
● Search Title/Abstract Only
○ Search Author
○ Search Fulltext
○ Search DOI
Search

Wireless Connectivity Solutions: From Wireless LANs to Ad Hoc and Sensor Networks

Optimizing Routing Schemes for Fast Moving Users in MST-Based Networks

Filip De Greve[1], **Frederic Van Quickenborne**[1], **Filip De Turck**[1], **Ingrid Moerman**[1] **and Piet Demeester**[1]

(1) Ghent University - IBBT - IMEC - Department of Information Technology, Gaston Crommenlaan 8, 9050 Gent, Belgium

Abstract

With the currently emerging trials for best-effort internet solutions on trains, solutions are required for delivering multimedia services to fast moving users. Research has already been devoted to dimensioning Ethernet aggregation networks, taking user movement into account while neglecting the experienced network performance. This paper extends this design for resilient networks and aims at minimizing packet loss and packet reordering in the dimensioning and routing process. For deployment in an Ethernet network which supports Multiple Spanning Trees (MSTs), effective path aggregation methods are proposed for finding a minimal set of spanning trees and these are thoroughly evaluated for different scenarios. Moreover the spanning tree assignment problem with predefined backup conditions is studied.

Keywords: Ethernet, user mobility, spanning trees.

Full Text Secured
The full text of this article is secured to subscribers. To gain access, you may:
- Add this item to your shopping cart for purchase later. Add to Shopping Cart
- Purchase this item now. Purchase Item Now
- Log in to verify access. Log In

Filip De Greve
Email:
Filip.DeGreve@intec.ugent.be

Frederic Van Quickenborne
Email:
Frederic.VanQuickenborne@intec.ugent.be

The references of this article are secured to subscribers.

Frequently asked questions | General information on journals and books