BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
DEPTARTMENT OF COMPUTER SCIENCE AND INFORMATION THEORY

RESILIENCE AND QUALITY OF SERVICE ASSURANCE METHODS IN ETHERNET AND UTRAN NETWORKS

János Farkas

Ph.D. Dissertation Summary

Supervised by

Dr. László Györfi
Department of Computer Science and Information Theory
Budapest University of Technology and Economics
Dr. Csaba Antal
Ericsson Hungary

Budapest, Hungary
2011

1 Introduction

Its simplicity and the high bandwidth provided at low cost made Ethernet an attractive technology in various network deployments. Since its invention in the 1970s, Ethernet has proved that it can adapt to evolving requirements. It was originally developed to provide connectivity in local area networks (LAN) and has become the de facto standard for enterprise networking. Ethernet is now evolving from the enterprise to the carrier. Nevertheless, as a LAN technology, Ethernet did not offer the resilience that is required in carrier grade service networks to provide quality of service guarantees to the customers.

One of the most important carrier grade requirements is resilience. Carriers have got used to the failover performance and robustness of Synchronous Optical Networks (SONET) and Synchronous Digital Hierarchy (SDH) networks, hence they expect similar performance from a packet network too. Spanning Tree Protocol (STP) provides failure handling in Ethernet networks from its early ages, which does not meet carrier requirements as its convergence time is in the order of ten seconds. The Rapid Spanning Tree Protocol (RSTP) [1] and the Multiple Spanning Tree Protocol (MSTP) [2] provides significantly faster failover by design, nonetheless, they cannot assure the 50 milliseconds (ms) carrier grade failover. RSTP and MSTP are distance vector protocols, therefore, the count-to-infinity problem [3] may appear during restoration from a failure. Other fault handling methods specified for Ethernet [4, 5, 6] are specifically designed for ring topologies thus cannot be applied in networks having arbitrary topology. Another approach is to rely on a centralised entity for fault handling as proposed by Sharama [7], which may decelerate failover and may be a single point of failure. Therefore, new algorithms and protocols were needed to meet carrier grade requirements in Ethernet networks and make it applicable even in metro scale networks.

A recent development in Ethernet networks is the introduction of a link state control protocol instead of the formerly used distance vector protocols. Thus network utilisation and transmission bandwidth can be increased compared to that of spanning tree protocols. The ISO/OSI Intermediate System to Intermediate System (IS-IS) [8] routing protocol is a suitable basis for defining the new control for Ethernet networks because it supports the handling of MAC addresses and it is defined based on Type, Length and Value (TLV) structures. There are two recent standards addressing this problem space: the IEEE 802.1aq Shortest Path Bridging (SPB) [9] and the IETF Transparent Interconnection of Lots of Links (TRILL) [10, 11], both rely on IS-IS. As SPB is specified by IEEE 802.1, it preserves the 802.1 architecture and it is compatible with all other 802.1 standards. Therefore, SPB uses standard 802 frame formats; it has Operations, Administration and Maintenance (OAM), the scalability and data centre solutions specified by 802.1, etc. As opposed to this, TRILL specifies

a new frame format, i.e. introduces a new data plane which implies the need for new hardware. Furthermore, TRILL is limited to customer Ethernet services [10] because it is specified assuming IEEE 802.1Q-2005 [2] thus it is not compatible with any amendments to [2], i.e. with IEEE 802.1 standards published after 2005. Therefore, TRILL has no OAM, it has scalability issues and it is not compatible with recent data centre standards, which is summarised e.g. by Eastlake [12], who is the chair of the TRILL Working Group. Due to the above differences, SPB is applicable in a much wider space of networking scenarios. Nevertheless, the introduction of link state control in Ethernet networks implies serious problems to be solved, e.g. loop prevention. Therefore, SPB has to implement extensions to IS-IS in order be able to use it for the control of Ethernet networks.

The spreading use of connectionless networks, e.g. the Internet Protocol (IP) and Ethernet, as transport for connection oriented services raise the need for Quality of Service (QoS) solutions. IP is often used in the transport of a Radio Access Network (RAN). The QoS requirements (delay, loss and jitter) of the real-time traffic of Universal Mobile Telecommunication System (UMTS) Terrestrial Radio Access Network (UTRAN) are stringent. For example the total UTRAN delay for voice traffic has to be below 7 ms. UTRAN has to implement Connection Admission Control (CAC) in order to be able to meet the QoS requirements. The CAC has to take into account the characteristics of UTRAN in order to be able to decide whether or not a new connection can be admitted without disrupting ongoing connections. The CAC has to be adapted to the applied transport technology in order to utilise network resources.

2 Research Objectives

The objective of this dissertation is to find algorithms and protocols that make Ethernet an applicable transport technology for metro scale networks keeping its advantages that made it attractive for campus and enterprise. A further aim is to define algorithms that help to provide QoS assurances in UTRAN deployed on IP transport. The following bullets summarise the objectives of the Theses.

- Define and evaluate methods and algorithms to ensure that an Ethernet network comprised of standard IEEE 802.1Q-2005 bridges in its core is able to provide the 50 ms carrier grade failover. That is, the new methods or algorithms can only be implemented in edge nodes or in a management system. Based on the physical topology of the network, forwarding paths have to be determined such that they are able to tolerate at least a single link or node failure. Furthermore, a method for detecting and handling failure events has to be also provided. (Thesis 1)

- Propose and evaluate algorithms extending existing link state protocols thus making them applicable for the control of Ethernet networks. The main goal is to provide loop prevention for a link state protocol controlling an Ethernet network. (Thesis 2)

- Define a Connection Admission Control algorithm for UTRAN using IP as transport network, such that the CAC is able to take advantage of Weighted Fair Queuing scheduling implemented in transport network nodes. Furthermore, define a CAC algorithm taking into account finer granularity than traffic aggregates thus able to guarantee QoS for voice flows. (Thesis 3)

Beyond the research objectives, I aimed to contribute to IEEE 802.1 standardisation with my results related to Shortest Path Bridging.

3 New Results

3.1 Resilient Ethernet

Carrier grade networks require fast failure handling, the requirement for the failover time is 50 ms. The spreading of Carrier Ethernet implies the need for enhancements to networks comprised of IEEE 802.1Q-2005 standard bridges as they cannot guarantee the required failover time. I have proposed and then evaluated enhancement techniques in order to provide 50 ms failover time in a bridge network.

Thesis 1 *I have defined a resilient Ethernet architecture for networks comprising standard IEEE 802.1Q-2005 bridges in their core and implementing the proposed new functionality in their edge nodes. I have shown that the proposed architecture is able to meet the 50 milliseconds carrier grade failover requirement.*

I proposed the resilient Ethernet architecture illustrated in Figure 1, where the core of the network comprises manageable IEEE 802.1Q-2005 standard Ethernet bridges (B1-B4), which support Virtual LANs (VLAN). Multiple predefined trees are statically set up across the network to serve as either primary or alternative paths thus provide connectivity. Each of the trees (T1-T3) is identified by a VLAN Identifier (VID). The trees are fault tolerant, i.e. they provide connectivity despite of the breakdown of a network component. In the event of a failure, the edge nodes have to stop forwarding frames to the affected trees and redirect traffic to unharmed trees. That is, the new functionality is only added to the edge nodes (EB1-EB4).

The edge nodes have to implement a fault handling mechanism, which includes the monitoring of the availability of the trees and traffic redirection in case of a failure

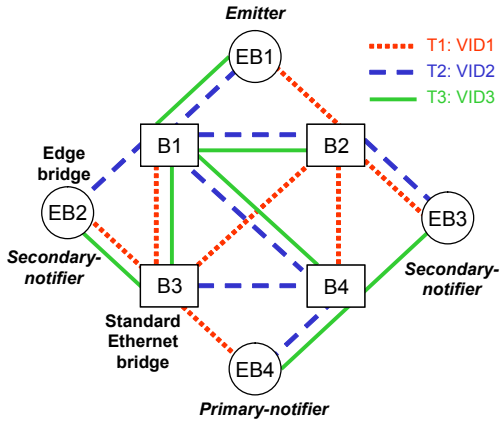

Figure 1: An example for the resilient Ethernet network

event. The trees have to be designed such that at least one tree survives the failure event against which the network is aimed to be protected. An algorithm is needed to calculate the fault tolerant trees, which requires accurate knowledge of the physical topology of the network.

Thesis 1.1 *I have defined a lightweight distributed fault handling protocol to be implemented in the edge nodes of the resilient Ethernet network and I have shown by means of measurements that the proposed protocol provides the 50 ms carrier grade failover. [J3, C7, P16]*

My proposed Failure Handling Protocol (FHP) is implemented in the edge nodes of the network as illustrated in Figure 1. FHP relies on a few broadcast messages to detect failures and to provide fast reaction to them. The three broadcast messages and the corresponding roles of the edge nodes are the following:

- *Keep Alive (KA)* messages are broadcasted periodically by one or more edge nodes referred to as *emitter* over each VLAN-tree according to a predefined time interval T_{KA}. If the *KA* messages are received by all other edge nodes, then the VLAN-tree is alive and operational.
- *Failure* message is issued by an edge node having *notifier* role when a *KA* message does not arrive over a VLAN-tree within a pre-defined *detection interval* T_{DI}. Thus the *notifier* informs all the other edge nodes on the breakdown of the given VLAN-tree.
- *Repaired* message is issued by the *notifier* that detected the failure when a *KA* message arrives over a previously failed VLAN-tree. Thus the *notifier* informs all the other edge nodes about the reparation of the broken VLAN-tree.

An example for the edge node roles is indicated in Figure 1. In order to avoid broadcast storms, *primary* and *secondary notifier*s are distinguished. The T_{DI} of *primary notifier*s is smaller than that of *secondary notifier*s, which is the only difference between the two types of *notifier*s. The operation of the protocol is specified by the flowcharts in Figure 2 for the two types of node roles.

Failover time is an important performance indicator of resilience approaches. The upper bound of the failover time of my proposed architecture is

$$T_F \leq T_{KA} + T_{DI} + T_{tr} + T_{pr}, \tag{1}$$

where T_{tr} and T_{pr} are the worst-case end-to-end transmission and packet processing delays in the network, respectively. Assuming that $RTT \sim 2(T_{tr} + T_{pr})$ and $T_{DI} = RTT$, the failover time is $T_F \leq T_{KA} + 1.5 \cdot RTT$, where RTT is the Round Trip Time. That is, the failover time depends on the size of the network. Besides the network specific delays, the failover time can be controlled by T_{KA}, of which smallest value is 3 ms in practice. Further guidance for the configuration of the protocol is given in [D].

The operation of the protocol has been verified in a prototype implementation. The measurements confirmed that the failover time can be maintained below 50 ms.

In order to be able to use the FHP, a proper connectivity structure is needed, which can be provided by tree topologies in case of Ethernet networks. Nonetheless, even determining the number of trees needed to handle a single link failure is NP-complete as proven by Čičić [13, 14].

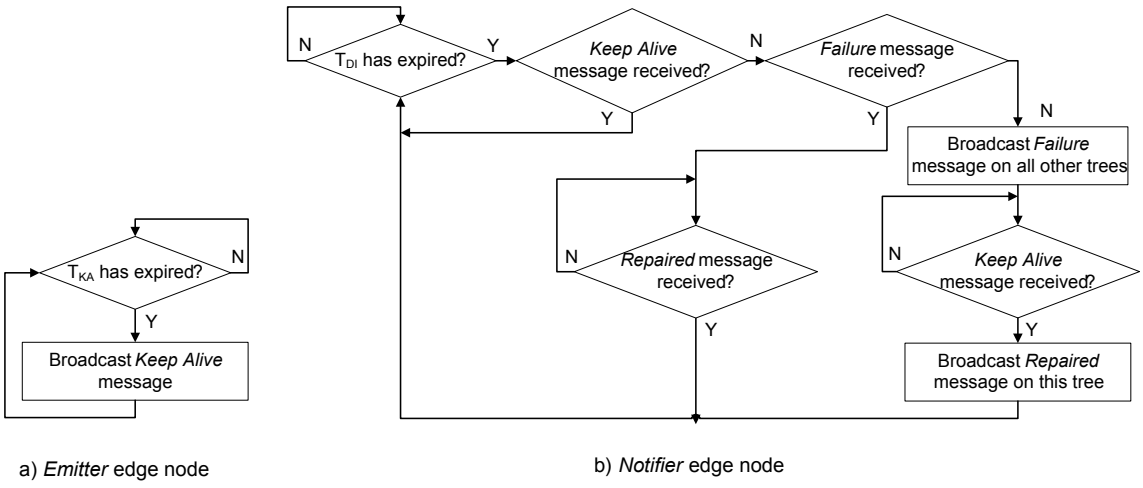

Figure 2: Operation of FHP

Thesis 1.2 *I have shown that at least* $k = \left\lceil \frac{l}{l-n+1} \right\rceil$ *spanning trees are needed in order to provide protection against a single link failure in a topology comprised of* n *nodes and* l *links. [C8]*

Remark: The lower bound is closer to the results provided by tree computation algorithms than the upper bound.

The resilient architecture relies on redirecting the traffic from one tree to another if a tree breaks down because of a failure event. In order to be able to perform traffic redirection in case of a link failure, there has to be at least one tree that survives the failure event thus provides connectivity. That is, for each link, there has to be a spanning tree not containing that particular link. Determining the minimal number of spanning trees meeting this requirement is an NP-complete problem.

Let n denote the number of nodes and l denote the number of links comprising topology $G = (N, L)$, where $|N| = n$ and $|L| = l$. The lower bound for the number of spanning trees necessary for link protection can be then calculated as:

$$k = \left\lceil \frac{l}{l-n+1} \right\rceil . \tag{2}$$

An upper bound on the number of trees needed in order to provide tolerance against a single link failure was given later by Čičić in [15], which also gives results of a heuristic algorithm. Table 1 provides a comparison of the lower bound provided by Equation (2) and the results published by Čičić [15] for random generated 16-node topologies. The table also shows results provided by Algorithm 1.3-1, which is described in the next thesis. The results provided by the algorithms are not integer in the table because they were obtained as the average of several runs of the algorithms. The upper bound provided by Čičić [15] is the Largest Minimal Cycle (LMC) in the topology, thus the table provides the average LMC of the different 16-node topologies.

As the table shows, both algorithms provide results closer to the lower bound than to the upper bound. Further results published by Čičić [15] show even larger gap between the upper bound and heuristic results. Thus, the lower bound provides more accurate picture on the characteristics of the topology than the upper bound.

Table 1: Number of trees required for handling a single link failure in 16-node networks

Average node degree	4	4.4	4.8
Upper bound of Čičić [15]	4.6	4.5	4.2
Average of heuristic algorithm of Čičić [15]	2.6	2.5	2.2
Average of Algorithm 1.3-1	2.4	2.18	2.16
Lower bound: Equation (2)	2	2	2

Besides having an estimate on the number of required spanning trees, the spanning trees themselves have to be determined.

Thesis 1.3 *I have defined an algorithm that determines spanning trees for a given input topology in order to provide protection against any single link or node failure, which is based on heuristics. I showed by means of extensive simulations on random topologies that the difference between the number of trees generated by my algorithm for link protection and the lower bound was 1 for the majority of the up to 50-node topologies evaluated. I have also defined an accurate physical topology discovery algorithm for heterogeneous Ethernet networks that provides the input necessary for spanning tree computation; and I have evaluated its operation by means of measurements. [C3, C8, P11, P15]*

The VLAN-tree topologies used for frame forwarding have to be fault tolerant in order to be able to handle failure events. The aim is to have at least a spanning tree that remains complete despite of the breakdown of a single network element, i.e. a node or a link. Thus, the requirements for the trees are the following for the two types of failures:

R1 *Link failure* – For each link, there has to be a spanning tree that does not include that particular link.

R2 *Node failure* – For each node, there has to be a spanning tree where that particular node is a leaf, i.e. its degree is one.

If these constraints are fulfilled, then there is at least one tree for each failure that is not affected, thus able to provide the forwarding among network nodes.

My algorithm computes the VLAN-trees that meet the above requirements. The construction of the VLAN-trees is split into two phases according to the two types of failures aimed to be handled. The algorithm aims to be forward looking as much as possible in order to minimise the number of trees both for link and node failures. Despite Phase 1 algorithm only addresses link failures, the trees are constructed in order to be able to handle node failures as well or at least do not deteriorate the computation of node protection trees if possible. Furthermore, in each step the algorithm makes a decision taking into account potential further steps. In order to implement the forward looking behaviour, a number of attributes are taken into account, which makes the description space consuming. For the detailed description of the algorithm please refer to [D]. A high level description of the algorithm is as follows:

Algorithm 1.3

Phase 1 – Algorithm 1.3-1: *Determining spanning trees for link protection*

Step 1.: Select the Central node, which is the highest degree node.
Step 2.: Construct the first tree from the Central node such that include all possible links but reserve a link for the second tree at each node if possible. Thus, the first tree becomes a star-like topology originated from the Central node, furthermore, allowing to form a disjoint tree if it is possible.
Step 3.: Construct further trees until link failure handling criterion R1 is met. The further trees are also constructed from the Central node. Nodes not yet leaf in any former tree are connected by a single link if possible thus aiding tree construction for node protection. Nonetheless, the key goal that the algorithm addressees in this step is to avoid including a link in the tree under construction if that link is included in all former trees. If it is not possible, then another tree is needed.

Phase 2 – Algorithm 1.3-2: *Determining spanning trees for node protection*

Step 4.: Construct further trees until node failure handling criterion R2 is met. The tree under construction is determined such that the branching points are the nodes that are leaf in a former tree. The nodes that are not yet leaf in any tree are connected by a single link to the tree under construction. If it is not possible, i.e. a non-leaf node becomes a branching point in order to make the tree spanning, then another tree is needed.

A key characteristic of the algorithm is that it minimises the number of the trees providing fault tolerance, which is illustrated e.g. in Table 1. Algorithm 1.3-1 always provided either the ceil or the floor of the average value shown in the table. Algorithm 1.3 was also evaluated by means of extensive simulations on random topologies comprising 5 to 50 nodes having average degree from 2.5 to 5. The results showed that Algorithm 1.3-1 provided as many trees for link protection as the lower bound or only a single additional one for the topologies where the average degree of the nodes was 2.8 or larger. Note, that a two-connected topology with the average degree of 2 is a ring topology.

Accurate knowledge of the physical topology is essential for determining the VLAN-trees. Therefore I defined an algorithm that is able to discover the entire physical topology among Ethernet bridges from multiple vendors even if they do not implement standard features in support of topology discovery. The algorithm requires that the manageable bridges implement some basic standards: STP or RSTP, VLAN tagging, Simple Network Management Protocol (SNMP) [16], Bridge MIB [17], MIB-II [18] and Interface MIB [19]. The algorithm consists of the following steps:

Algorithm 1.4

Step 1 – LLDP discovery: Discovery of the topology segment that implements the Link Layer Discovery Protocol (LLDP) [20], which is the standardized support for topology discovery.
Step 2 – Node discovery: After the LLDP discovery, the Network Management System (NMS) implementing the topology discovery algorithm issues a broadcast ping message to the sub-network broadcast address and waits for the replies in order to discover the manageable nodes that do not support standard topology discovery.
Step 3 – Spanning tree discovery: Based on the ping messages and replies, the links comprising the spanning tree that carries management traffic among non-LLDP bridges are determined by the NMS.
Step 4 – Inactive link discovery: The links span among non-LLDP bridges and not included in the spanning tree are then finally determined. The NMS logs-in the non-LLDP bridges and turns down the ports not yet included in the physical topology database. The NMS then receives an SNMP message from both ends of these inactive links. The topology database then becomes complete with the inactive links. Note that turning down and up a link that is not part of the spanning tree neither causes re-convergence in the spanning tree protocol nor disturbs user traffic.

The measurements on various, up to 12-node mesh topologies in a test network comprising bridges of five vendors showed that the algorithm is accurate. For the detailed description of the algorithm and the measurements please refer to [D].

3.2 Enhancements to Shortest Path Bridging

Frames are often forwarded along a roundabout path in Ethernet networks controlled by a spanning tree protocol. Therefore, the IEEE 802.1aq Shortest Path Bridging (SPB) [9] standard is introducing a link state protocol for the control of Ethernet networks and aims to make frame forwarding more efficient by using the shortest path. For the support of multicast forwarding, SPB implements shortest path forwarding by means of source rooted Shortest Path Trees (SPT), i.e. each bridge has its own SPT for frame transmission. Nevertheless, the existing link state protocols require extensions in order to be applicable in SPB, e.g. because they do not provide loop prevention, which is crucial in Ethernet networks.

Thesis 2 *I have shown that a loop prevention mechanism is needed in link state controlled Ethernet, then I have defined loop prevention algorithms that can be implemented as extensions to IS-IS and thus applicable in the control protocol of the IEEE 802.1aq Shortest Path Bridging architecture. I have proved that the proposed algorithms prevent the appearance of loops. I have evaluated the effects of the proposed*

algorithms on network convergence time by means of extensive simulations in realistic topologies.

Loop free operation at all times is an absolute requirement in Ethernet networks. Therefore, SPB has to incorporate a mechanism that prevents loops irrespective of the number of link state updates in progress and the order of their arrival and inclusion in the link state computation at a bridge. Note that there have been some loop prevention mechanisms proposed for IP Fast Re-Route (IPFRR). However, as summarised by Shand [21] none of these is a pure control protocol approach being able to handle multiple topology changes in a reasonable time. Thus they are not applicable in SPB.

The application of a loop mitigation mechanism was proposed for SPB in order to treat transient loops. Reverse Path Forwarding Check (RPFC) can be applied to audit the port of arrival of a frame in order to ensure that it arrives on the port lying on the shortest path from the source. Note that RPFC is referred to as ingress check in the SPB specification [9].

Thesis 2.1 *I have shown that Reverse Path Forwarding Check fails to prevent the appearance of forwarding loops. [S7]*

Remark: The counter example I provided is referred to as Farkas loop e.g. by Allan et al. [22].

All stable forwarding topologies are loop free both in Ethernet and IP networks, however, loops may occur during topology transients. Figure 3 shows an example topology transient, where a loop occurs despite of RPFC, i.e. ingress check

The figure shows only part of a topology, there might be further nodes connected to B, C, D or E. Nevertheless, the figure shows all the links connecting the nodes depicted in the figure and also indicates the cost of each link. The solid line links are active in the SPT of node A but the dash dot line links are inactivated, e.g. by

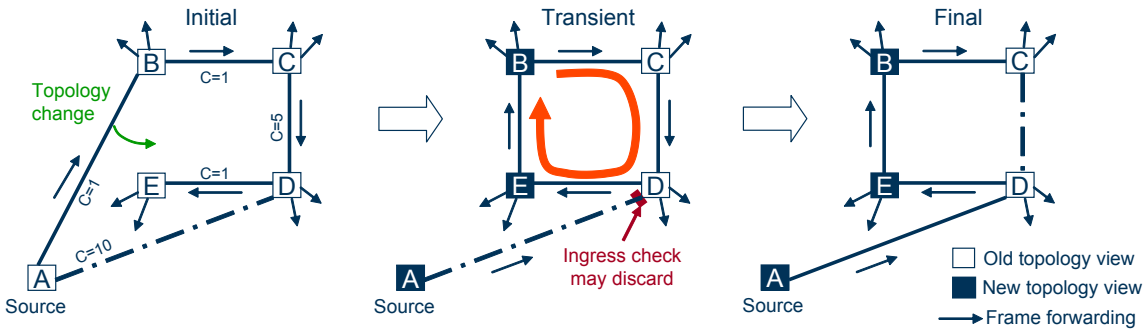

Figure 3: Farkas loop

frame discarding implemented by RPFC. The little arrows show the direction of frame forwarding on the SPT of node A.

There is a topology change in the initial stage of the example: the physical connection between A and B is cut and a new physical connection appears between B and E at the same time. The initial and the final topologies are loop free as illustrated in the figure. The link between A and D is not used in the initial topology; and the link between C and D is unused in the final one. However, a loop is formed during the transient if nodes A, B and E are aware of the change thus have an updated view on the topology but nodes C and D have an outdated view. The loop appears even if RPFC is applied. As a consequence of the loop, multiple copies of a multicast or broadcast frame may be spread towards other nodes connected to B, C, D or E as shown by the little arrows. Note that the Time To Live (TTL) field applied in IP packets is a weaker loop handling technique than RPFC, because TTL does not prevent the appearance of loops but it provides means to live with them.

As the existing methods do not provide satisfactory solution for loop prevention in link state controlled Ethernet networks, there is a need for a new and efficient mechanism.

Thesis 2.2 *I have defined the Neighbour Synchronisation loop prevention algorithm and I have shown that it ensures loop free operation in link state protocol controlled networks if neighbours having mismatch in their topology view drop packets instead of forwarding them to each other. [J2, S6, P7]*

Remark: Neighbour Synchronisation is an add-on to an existing link state protocol for providing loop prevention.

If a link state protocol is used for the control of the network, then transient loops may appear because of network nodes having different views on the physical topology after a topology change. Therefore, I have proposed the Neighbour Synchronisation algorithm for preventing transient loops, where neighbour nodes implement a handshake mechanism in order to make sure that they have the same view on the topology. Furthermore, they do not exchange data frames in case of a topology mismatch. The Neighbour Synchronisation fits very well into standard link state protocols, e.g. IS-IS. The verification of matching topology databases can be implemented by exchanging a digest on the topology database.

The topology database synchronisation is performed between neighbours and it is independent of the synchronisation between another pair of nodes. Therefore, different ports of the same node may have different synchronisation states. The state of a port of a node can be modelled by the two-state state machine illustrated in Figure 4. The names of the states reflect whether or not the node is synchronised with its neighbour connected through the given port.

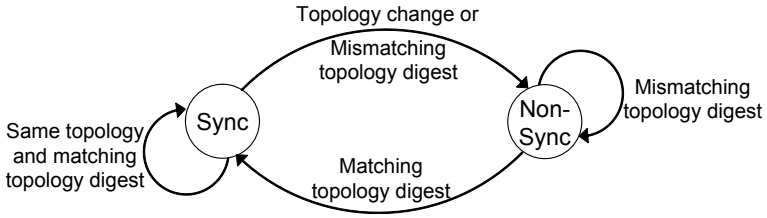

Figure 4: Node port state machine

The port is in Sync if the topology digest of the peering node is the same as the own topology digest, otherwise it is in Non-Sync. The port remains in Sync until the node is not notified about any topology change or the peering node does not send a topology digest differing from the locally stored one. If any of these two happens, then the state of the port changes to Non-Sync. The port remains in Non-Sync as long as the two neighbours have mismatching digests. As soon as they have a matching digest again, the port moves to the Sync state. If the state of the port is Non-Sync, then it blocks data communication to its neighbour connected through the given port. Data communication only operates if the port is in Sync state.

I showed by an indirect proof that the Neighbour Synchronisation algorithm prevents loops, please refer to [D] for the details of the proof. The basis of loop prevention is that the Neighbour Synchronisation mechanism ensures that there is no trespassing between different topologies, which is illustrated in Figure 5. Node A belongs to topology k because it only received the LSPs describing topology k. As opposed to this, node B belongs to topology k+1 as it has received one or more LSP on topology information which differs from topology k. As the topology view of A and B are different, the link between them is blocked by the Neighbour Synchronisation.

If a packet to be forwarded to B received by A on topology k, then A may perform two actions on the packet. A either drops the packet or stores it in a buffer and forwards it to B when they belong to the same topology again. If buffering is applied, then a packet may pass through a topology update, e.g. from topology k to k+1.

If a packet is able to pass from one topology to another, then it may be forwarded multiple times through the same node.

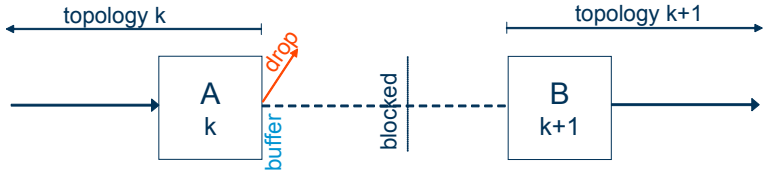

Figure 5: Topology separation

Thesis 2.3 *I have shown that the Neighbour Synchronisation algorithm provides optimal loop prevention even if buffering is applied instead of packet dropping because it ensures that each packet gets to its destination such that it is transmitted by a node at most k+1 times if there are no more than k topology changes in the network.*

If packets are buffered instead of dropping meanwhile the neighbours are not in synch and forwarded as soon as they have the same topology view, then the same packet may be transmitted multiple times by the same node due to a very unlikely series and coincidence of events, for which an example is given in [D]. An optimal loop prevention algorithm minimises the number of times a packet transmitted through a node even in such unlikely cases.

Definition 2.1. The state of a packet towards its destination is defined by (X, D), where
– X is the number of topology changes in the network minus the number of topology updates the packet has passed through and
– D is the distance to the destination in the number of hops remaining to the destination within the current topology.
Lexicographic order can be applied for the states:

$$(X_1, D_1) > (X_2, D_2) \equiv (X_1 > X_2) \vee (X_1 = X_2 \wedge D_1 > D_2). \tag{3}$$

That is, state 1 is greater than state 2 if the packet in state 2 has been passed through more topology updates than the packet in state 1 or if the packets in both states are within the same topology and the distance in state 2 is smaller than the distance in state 1.

Neighbour Synchronisation ensures that the state of a packet is always strictly decreasing, i.e. either X or D decreases in the state of a packet following a former one, otherwise the two states are the same. The reason for this is that no packet can be exchanged between nodes being in different topologies as shown in Figure 5. Nodes only exchange data packets if they have the same topology view, i.e. they are part of the same topology. Within a particular topology, each packet is always forwarded along a tree and transmitted by a node only once, which decreases the distance to the destination at each hop.

Due to buffering, a packet may move from a topology to a more recent one if the node storing the packet is updated to a more recent topology. If there are k topology changes in the network, then a packet may be passed to an updated forwarding topology at most k times. Thus, a packet may be forwarded at most along k+1 topologies. Therefore, the Neighbour Synchronisation method ensures that a packet is forwarded at most k+1 times by a node, thus it provides optimal loop prevention even in case of strange and unlikely constellation of events.

Suspending packet exchange between neighbours adds some delay to network convergence time after a change in the topology.

Thesis 2.4 *I have shown by means of extensive simulations on real and artificial network topologies that the Neighbour Synchronisation loop prevention algorithm only increases the convergence time within the range of milliseconds. Measurement results in a test implementation with standard parameter settings showed that the effect of Neighbour Synchronisation on network convergence is negligible compared to the convergence time. [S1]*

The operation of the Neighbour Synchronisation algorithm was analysed in a simulator developed in OMNeT++ 4.0 [23], which is a discrete event object oriented C++ simulator environment. The bridge architecture was implemented according to the IEEE 802.1Q [2] specification in the INET framework of OMNeT++. IS-IS was then implemented as a Higher Layer Entity of the bridge architecture along the ISO specification [8]. The Neighbour Synchronization handshake was then implemented using IS-IS Hello PDUs.

The simulation analysis was performed on six topologies: the 22-node AT&T [24], the 37-node COST266 [25] reference topology, a 50-node German backbone network Germany50 [24] and an artificially constructed topology comprised of multiple rings thus referred to as Rings. In addition, 100-node (R100) and 150-node (R150) random topologies were used. IS-IS parameters were fine tuned to eliminate artificial delays, e.g. the delay between becoming aware of a topology change and starting the Dijkstra computation. Thus, it was possible to detect the effect of the Neighbour Synchronisation.

Table 2 shows the average of hundred simulation results for each scenario. Comparing convergence time results with and without the Neighbour Synchronisation, it can be seen that the difference is roughly 1 ms in case of a link failure. The difference varies more in case of a node failure, the additional delay to the average value is between 1 ms and 10 ms. That is the Neighbour Synchronisation algorithm does not deteriorate network convergence.

The operation of the Neighbour Synchronisation algorithm was also evaluated by means of measurements in a prototype network comprised of six Debian GNU/Linux PCs. The prototype uses the ISIS routing daemon called *isisd* from the *quagga* open source routing suite [26]. The Neighbour Synchronisation algorithm was implemented in the *quagga* daemon. All the IS-IS parameters were set to their smallest value allowed by the standard [8] during the measurements.

The convergence time was investigated with and without the Neighbour Synchronization loop prevention algorithm. The convergence time was measured from the

Table 2: Average convergence time after a failure event [ms]

Link failure

	AT&T	COST266	Germany50	Rings	R100	R150
without Nbr Sync	7.980	18.632	35.432	77.331	163.648	394.811
with Nbr Sync	9.008	19.615	36.019	78.336	164.734	395.758

Node failure

	AT&T	COST266	Germany50	Rings	R100	R150
without Nbr Sync	9.339	21.044	37.629	86.904	164.490	395.140
with Nbr Sync	10.202	29.319	44.809	95.342	165.333	396.746

reception of the first notification of the link failure until the last FIB update associated with the topology change has been completed in the network.

The average convergence time of twenty measurement results without loop prevention was 2.03 seconds. Neighbour Synchronisation increased the average convergence time to 2.1 seconds, i.e. the difference is two orders smaller than the convergence time itself. That is, Neighbour Synchronisation did not increase the convergence time significantly in case of standard IS-IS parameter settings.

Both the simulation and measurement based evaluations assured that Neighbour Synchronisation eliminates the loops that occur without loop prevention. Furthermore, Neighbour Synchronisation does not deteriorate network convergence.

Thesis 2.5 *I have defined Root Controlled Bridging (RCB) for the control of SPB and I have shown that RCB prevents loops and reduces the computational complexity to* $\mathcal{O}\left(|L| + |N| \cdot \log|N|\right)$ *from* $\mathcal{O}\left(|N|\left(|L| + |N| \cdot \log|N|\right)\right)$*, which is the complexity of alternative solutions for topologies* $G(N, L)$ *comprised of* $|N|$ *nodes and* $|L|$ *links. [C2, S7, P9, P10]*

Remark: RCB is an extension to an existing link state protocol, which improves computational complexity besides providing loop prevention.

The SPB solution applying standard IS-IS without any special path computation enhancement is referred to as Basic IS-IS in the following. In Basic IS-IS, each node has to compute the source rooted SPTs of all other nodes besides its own one in order to be able to implement proper frame forwarding, i.e. they have to perform an All Pairs Shortest Path computation. Thus the computational complexity of Basic IS-IS is $\mathcal{O}\left(|N|\left(|L| + |N| \cdot \log|N|\right)\right)$.

I proposed a new approach for the control of SPB networks referred to as Root Controlled Bridging (RCB), which is an extension to IS-IS for the control of SPB

networks. In RCB, bridges maintain the link state database as specified in IS-IS, nonetheless, each bridge only computes its own tree and controls the set-up or update of its own tree, i.e. each tree is controlled by its Root Bridge. Thus, the architecture is distributed and robust as SPTs are controlled independently of each other. The control of each tree is centralised, which has significant advantages in reducing computation. As each RCB bridge only computes a single SPT, the computation complexity is reduced to $\mathcal{O}(|L| + |N| \cdot \log |N|)$. Note that if a bridge goes down, then its SPT becomes unnecessary.

The operation of RCB only differs from standard IS-IS when a forwarding tree is computed and set up in the network. The operation of these processes is described by the flowcharts in Figure 6. As Figure 6(a) shows, The Root Bridge computes its new SPT if there is a change in the topology. If there is a change in the SPT as well, then the new tree has to be set up in the network. If that is the case, then the Root Bridge sets its discarding ports, then its forwarding ports. A discarding port drops all frames it receives. After that the Root Bridge advertises its SPT to the rest of the bridges in Tree Advertisement (TA) messages. TA can be implemented in a new TLV, thus it is a standard compliant extension to IS-IS.

Figure 6(b) shows the set-up or update of a tree by non-root bridges after the reception of the TA message. An important feature of TA message propagation is that a TA message is only forwarded along the tree that it describes from the root towards the leaves, i.e. it is not flooded. Another key feature of the SPT update process is that discarding ports are always set before forwarding ports and further distribution of the TA message, as also illustrated in Figure 6. This ensures loop free operation, hence, no additional mechanism is needed for loop prevention in RCB.

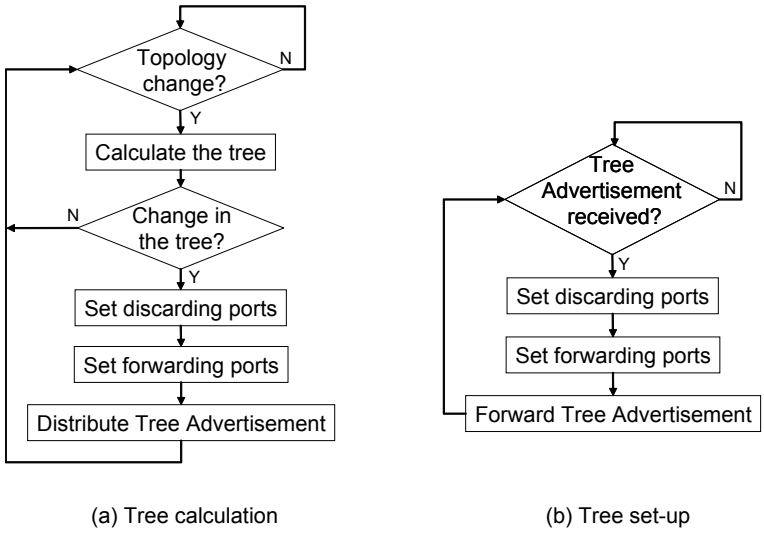

Figure 6: Tree computation and set-up

I gave an indirect proof showing that the RCB update mechanism prevents loops, please refer to [D] for the proof.

RCB reduces computation complexity at the price of messaging, which may influence network convergence time.

Thesis 2.6 *I have shown by means of extensive simulations over various topologies and parameter settings that RCB provides faster network convergence than alternative link state solutions in mesh topologies larger than 200 nodes. [C1]*

I have evaluated and compared the convergence time of Basic IS-IS and RCB for different topology types of various sizes, both after link and node failure events. In addition, an MSTP based control for SPB was also included in the evaluation, which was the first one proposed when standardisation started; it is referred to as MSPT-SPB. Thus the results also show a comparison of distance vector and link state protocols. Three types of topologies were investigated. The Rings topology consists of a central ring and sub-rings connected to it. In addition to the Rings, lightly and heavily meshed topologies were applied. The number of nodes comprising the networks varied from 50 to 280. The convergence time after a link failure in the Light-mesh topology is shown in Figure 7. Please refer to [D] for further results and detailed parameter settings.

Sparse topologies such as rings are not that favourable for solutions intense in control messaging, e.g. MSTP-SPB or RCB, because the control information has to travel on long paths. Thus, the convergence time of the Basic IS-IS approach is smaller than that of the RCB for the Rings topology.

Nevertheless, SPB is advantageous to be used in more meshed topologies where using the shortest path improves frame forwarding efficiency. The convergence time of Basic IS-IS is considerably affected by the network size as both the number of nodes

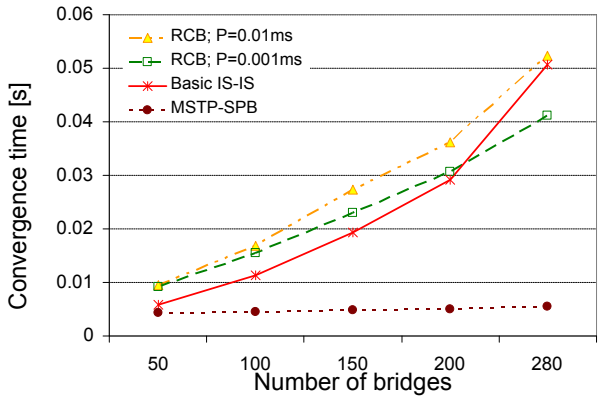

Figure 7: Link failure in the Light-mesh topology

and links influences its computational complexity, which becomes the dominant factor as network size increases. Therefore, RCB outperforms Basic IS-IS over a certain network size in the mesh topologies. In case of a link failure in a light mesh topology the convergence time of RCB with the most realistic TA message parsing setting (P = 0.001 ms) becomes smaller than that of the basic IS-IS if the network comprises more than 200 nodes, which is illustrated in Figure 7.

The performance analysis showed that the convergence time of the Basic IS-IS approach is sensitive to the size of the network in case of mesh topologies due to its computational complexity. Root Controlled Bridging, which reduces the computational complexity at the price of more control messages, converges faster than Basic IS-IS as the size of mesh topologies increases. The more meshed and the larger the topology, RCB converges faster than the alternative Basic IS-IS approach.

3.3 Connection Admission Control for UTRAN

UMTS Terrestrial Radio Access Networks (UTRAN) have to implement Connection Admission Control (CAC) in order to be able to meet the stringent QoS requirements of admitted connections. On the other hand, the CAC should allow the utilisation of the available network resources.

Thesis 3 *I have defined Connection Admission Control algorithms for the Iub interface of UTRAN that are able to take into account flow level characteristics of user traffic and improve the utilisation of network resources by taking advantage of the Weighted Fair Queueing scheduling applied in transport network nodes.*

The traffic of the Iub interface of UTRAN can be modelled by independent ON-OFF modulated periodic sources. Traffic sources belonging to the same traffic class i are described by the $\{T_i, b_i, \alpha_i\}$ parameter set. The Transmission Time Interval (TTI) is denoted by T_i, which is the deterministic packet inter-arrival time if the source was always in ON state. The packet size is denoted by b_i and α_i is the activity factor, which describes the ON-OFF behaviour. The QoS requirements of traffic class i are described statistically by the $\{d_i, \varepsilon_i\}$ parameters, where d_i is the delay requirement and ε_i is the allowed packet drop rate. Flow level QoS may be described by the additional parameter δ_i, which is the probability of the violation of the ε_i packet drop rate. The UTRAN system has K traffic classes and variable N_i is the number of actually ongoing connections in class i.

A QoS requirement is violated if a buffer is *overload*ed, i.e. its input rate exceeds its service rate, which is referred to as *burst level QoS violation* because it is caused by the burst of sources in ON state. A QoS requirement may also be violated due to large packet *delay* caused by temporary packet congestion even if the input rate of a buffer

is smaller than its service rate, which is referred to as *packet scale QoS violation* or *delay violation*. A model that handles these two types of violations separately was proposed and verified by Malomsoky in [27, 28]. That is, the packet drop QoS requirement can be split into two:

$\varepsilon_i = \varepsilon_i^{burst} + \varepsilon_i^{packet}$.

Thus, burst scale and packet scale effects can be analysed separately. Furthermore, a CAC algorithm can check burst and packet scale QoS violations independently of each other before admitting a new connection.

A CAC algorithm may take advantage of the knowledge on the system and traffic characteristics if they are available. A model for the traffic of UTRAN Iub interface is given above. The $n \cdot D/D/1$ queuing system described in detail by Roberts [29] can be extended in order to describe the operation of UTRAN Iub. Therefore, a model based CAC algorithm can be applied for the Iub interface of UTRAN.

As delay requirements are small compared to burst level dynamics, it is enough to model the burst level operation by bufferless multiplexing. As opposed to this, if a delay requirement is violated due to packet scale operation, then it can be assumed that the buffer does not overflow. Therefore, the system can be modelled as if the buffer was infinite when investigating packet scale QoS violations.

A model based CAC is described in the following.

Thesis 3.1 *I have defined a Connection Admission Control algorithm that takes advantage of the Weighted Fair Queueing scheduling applied in an IP UTRAN network. I have shown by means of simulations that my algorithm improves bandwidth utilisation at small link capacities, e.g. by* 46% *in case of a* $2 \cdot E1$ *link. [C10, P18]*

For checking the packet scale QoS violation, I proposed to approximate the operation of the CB-WFQ system with a set of Strict Priority (SP) systems that operate separate from each other. My proposal is referred to as Separated Strict Priority. Approximation is required due to the complexity of the CB-WFQ system.

A traffic mix $\underline{N} = N_1, N_2, \ldots, N_K$, which gives the number of ongoing flows of each traffic class in a K-class system, does not violate the packet scale QoS requirement if it is below the packet scale constraint surface in the space span by the number of connections of different traffic classes.

Applying the proposed Separated Strict Priority model, the complex packet scale constraint surface of a traffic class in a CB-WFQ system can be approximated by a set of hyperplanes.

At an admission request, the CAC algorithm can check QoS violation performing checks on the $\underline{N}_{new}$ that had been formed if the new connection was admitted. Traffic mix $\underline{N}_{new}$ does not cause delay violation for class i if $\underline{N}_{new}$ is below any of the hyperplanes of class i, which is tested by checking

$$\max_{Y} \left(F_{ii}^{Y} + 1 - \sum_{j \in Y} \frac{F_{jj}^{Y}}{F_{ji}^{Y}} \cdot N_j \right) > 0, \tag{4}$$

where Y denotes the index set of the buffers; F_{ji}^{Y}, F_{ii}^{Y} and F_{jj}^{Y} are used to determine a hyperplane, of which computation is described below.

If traffic mix $\underline{N}_{new}$ passes the check of Equation (4) for each traffic class, then admitting the new connection request resulting in $\underline{N}_{new}$ does not cause packet scale QoS violation.

As described above, the aim is to propose such equivalent systems to the original CB-WFQ system that allows to express the packet scale constraint region of class i served in queue k. Figure 8 illustrates my proposed approximation model in case of a three-queue system. Three types of queues are distinguished: the observed queue denoted by k, saturated and lightly loaded queues. That is the queues are partitioned into three sets: $\{k\} \cup A \cup B$, where A involves the lightly loaded and B involves the saturated queues. Class $1 \in B$ and Class $3 \in A$ in the example shown in Figure 8.

A lower bound for the service rate S_k for queue k in a WFQ system can be determined as

$$S_k \geq \frac{c_k}{c_k + \sum_{j \in B} c_j} \left(C - \sum_{j \in A} R_j \right), \tag{5}$$

where R_j is the input rate for queue j.

In order to provide an equivalent system, the saturated buffers (with index set B) are first separated from the scheduler. Thus the reduced system includes buffers in index set A and the observed buffer k. The service rate of the reduced system is

$$C' = \frac{c_k}{c_k + \sum_{j \in B} c_j} C, \tag{6}$$

which is an upper bound to the service rate of queue k.

The service order of packets depends on the actual value of the minimum bandwidth assignments (c_i). To give a worst-case approximation for queueing delay in

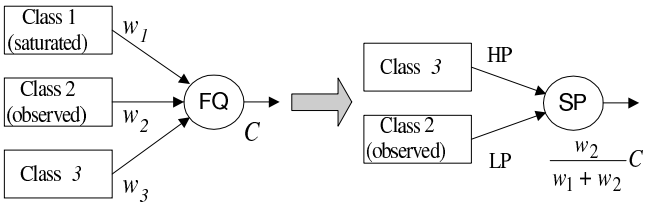

Figure 8: Separated SP model

buffer k, packets in buffers A are assumed to be served as if they had priority over the packets in buffer k.

The packet size has to be also adjusted for buffers in A in order to achieve proper operation in the reduced system. These packets are served at the linkrate of the original system, which appears for a buffer k packet in the reduced system as if the size of the packets in any buffer in A were reduced to

$$b'_i = \frac{c_k}{c_k + \sum_{j \in B} c_j} b_i; \ \forall i \in A. \tag{7}$$

As a result, the reduced system operates as a Strict Priority scheduler with parameters C' and b'. Depending on which buffers are considered to be saturated, 2^{L-1} different reduced systems can be distinguished, where L is the number of real-time buffers of UTRAN Iub. Different reduced systems give good approximations under different traffic conditions, and a combination of them gives a conservative approximation for the service rate at any traffic mixes. Therefore, queueing delay of buffer k packets in the model is an upper bound for their queueing delay in the original system.

By using the Separated Strict Priority model, the approximation problem of packet scale constraint surface in the CB-WFQ system is reduced to the problem of approximating the packet scale constraint surface of low priority (LP) classes in multi-class strict priority systems. As shown by Malomsoky [28], the delay constraint surface of a LP class can be approximated by a single hyperplane and this approximation is conservative.

Denote Y the index set of queues and K_{SP} the number of classes in the SP system. F_{ji}^{Y} is the maximal number of class j sessions in the SP system if delay requirement of a single class i session should be met and all other classes are empty. Formally,

$$F_{ji}^{Y} = \max \left\{ N_j \mid \sum_{n_j=0}^{N_j} \mathbb{P}\left[\mathcal{D}_i^{n_j} > d_i\right] \cdot \Pi(n_j) \leq \varepsilon_i^{packet} \right\}, \tag{8}$$

where $\mathbb{P}\left[\mathcal{D}_i^{n_j} > d\right]$ is the delay distribution of a class i session if the number of class j sessions in the system is n_j. $\Pi(n_j)$ is the probability that n_j sessions are active, which can be calculated using the multi-dimensional binomial distribution.

The equation of an approximating hyperplane of the delay constraint of class i can be then written as

$$\sum_{j \in Y} \frac{F_{jj}^{Y}}{F_{ji}^{Y}} \cdot N_j = F_{ii}^{Y} + 1. \tag{9}$$

The packet scale constraint is then met if Equation (4) is valid for each traffic class as described above.

The burst scale QoS violation has to be also checked, for which I proposed a Gaussian approximation. Applying the central limit theorem, the distribution of the number of active sessions in a buffer can be approximated by a normal distribution if the number of sources grows. Then, the check for burst scale QoS violation can be performed using the following closed form approximation:

$$1 - \Phi\left(C, \tilde{R}_k, \tilde{V}_k\right) + \frac{\tilde{V}_k}{\tilde{R}_k} \cdot \varphi\left(C, \tilde{R}_k, \tilde{V}_k\right) \leq \varepsilon_k^{burst}, \tag{10}$$

where $\varphi(x, \mu, \sigma^2)$ and $\Phi(x, \mu, \sigma^2)$ are the density and cumulative distribution functions of the normal distribution with mean μ and variance σ^2, respectively. Furthermore, $\tilde{R}_k$ and $\tilde{V}_k$ are the mean and variance of the input rate in buffer k, which are corrected according to the CB-WFQ operation as described in detail in [D].

I have evaluated the performance of the proposed CAC algorithm by means of simulations and compared its bandwidth efficiency to the Separated FIFO algorithm, which was the only alternative to Separated Strict Priority. Separated FIFO applies a FIFO approximation for WFQ, i.e. the capacity is split proportional to the weight setting, thus the guaranteed rate is always reserved for each queue. At small link capacities, e.g. at 2·E1, the capacity need of Separated FIFO is 46% larger than that of the proposed Separated Strict Priority CAC algorithm. The difference decreases as the link capacity increases because the packet scale operation becomes less dominant.

Thesis 3.2 *I have given a Connection Admission Control algorithm that guarantees QoS at flow level such that Gaussian approximation is applied for the bufferless multiplexing model and I have shown that the packet loss violation probability can be approximated as the quantile of a normal distribution. [C9]*

If QoS is aimed to be guaranteed at flow level instead of aggregates, then the QoS definition of [C9] should be applied. The QoS requirement for aggregates, e.g. for a traffic class, is typically determined by the delay requirement d and packet drop requirement ε. However, besides the number of admitted flows, the fraction of dropped packets also depends on the activity factor α of ongoing flows. The flow level QoS requirement proposed by [C9] is that the δ probability of the violation of the ε packet drop rate requirement should be kept below a required value.

Bufferless multiplexing is applied to evaluate burst level operation, which is also applied when taking into account the flow level QoS requirement. The multiplexer serves $Z = C \cdot T/b$ packets in a source period, and multiplexes N ON-OFF sources, of which activity factors are iid random variables.

Meeting the QoS requirement depends on the number of sources in state ON at the same time, which is a sum of Bernoulli random variables. Applying the central

limit theorem, we get

$$\delta = \mathbb{P}\left[\frac{Z - \sum_{i=1}^{N} \alpha_i}{\sqrt{\sum_{i=1}^{N} \alpha_i(1-\alpha_i)}} \leq \Phi^{-1}(1-\varepsilon)\right] \tag{11}$$

for the QoS measure δ, where $\delta = 1$ indicates QoS violation.[1]

The direct calculation of δ is very complicated even for small values of N, which motivates the application of an accurate and fast approximation of δ.

I propose the following method for the calculation of δ in case of a general distribution of the activity factor. In order to calculate δ, we need to determine

$$\mathcal{Y} = \frac{Z - \sum_{i=1}^{N} \alpha_i}{\sqrt{\sum_{i=1}^{N} \alpha_i(1-\alpha_i)}}. \tag{12}$$

I applied the following steps to derive δ:

1. I have shown that $\mathcal{Y}$ is normally distributed independently of the distribution of activity factors;
2. I have determined the mean and the variance of $\mathcal{Y}$ using the first moments of the activity factor. The first and the second moments of $\mathcal{Y}$ can be calculated as infinite Taylor:

$$\begin{aligned}
\mathbb{E}[\mathcal{Y}] &= A(\mu_1) + C(\mu_1)N\sigma^2 + E(\mu_1)N\mathbb{E}\left[(\alpha-\mu_1)^3\right] + \dots \\
\mathbb{E}\left[\mathcal{Y}^2\right] &= A^2(\mu_1) + N\sigma^2 \times \\
&\quad \left[B^2(\mu_1) + 2A(\mu_1)C(\mu_1)\right] + N\mathbb{E}\left[(\alpha-\mu_1)^3\right] \times \\
&\quad [2A(\mu_1)E(\mu_1) + 2B(\mu_1)C(\mu_1)] + \dots
\end{aligned}$$

where μ_i is the i-th moment, σ^2 is the variance of the activity factor and

$$\begin{aligned}
A(a) &= \frac{Z - Na}{\sqrt{Na(1-a)}} \\
B(a) &= -\frac{1}{2}\frac{(Z-Na)(1-2a)}{[Na(1-a)]^{3/2}} - \frac{1}{\sqrt{Na(1-a)}} \\
C(a) &= \frac{1}{2}\frac{Z-Na}{[Na(1-a)]^{3/2}} + \frac{3}{8}\frac{(Z-Na)(1-2a)^2}{[Na(1-a)]^{5/2}} + \\
&\quad \frac{1}{2}\frac{1-2a}{[Na(1-a)]^{3/2}}
\end{aligned}$$

[1] $\Phi(x)$ and $\Phi^{-1}(x)$ denote the standard normal distribution function and its inverse, respectively.

$$E(a) = -\frac{9}{4}\frac{(1-2a)^2}{(Na(1-a))^{5/2}} - \frac{15}{8}\frac{(Z-Na)(1-2a)^3}{(Na(1-a))^{7/2}} - 3\frac{1}{(Na(1-a))^{3/2}} - \frac{9}{2}\frac{(Z-Na)(1-2a)}{(Na(1-a))^{5/2}}.$$

3. I have then calculated $\delta = \mathbb{P}\left[\mathcal{Y} \geq \Phi^{-1}(1-\varepsilon)\right]$.

The above results can be applied in an admission control algorithm that takes into account flow level characteristics as described in detail in [C9]. The probability of drop rate violation δ_N for N flows, can be calculated based on the results described above as follows:

$$\delta_N = \Phi\left(\Phi^{-1}\left(1-\varepsilon\right);\ \mu = \mathbb{E}\left[\mathcal{Y}\right],\ \sigma^2 = \mathbb{E}\left[\mathcal{Y}^2\right] - \mathbb{E}\left[\mathcal{Y}\right]^2\right), \tag{13}$$

where the mean and variance of $\mathcal{Y}$ should be determined as specified in Step 2 above. If $\delta_N < \delta$, then the CAC decision is "Admit", otherwise it is "Reject".

The numerical properties of the proposed algorithm were evaluated for uniform activity factor distribution and for the GSM speech activity distribution measured by Westholm [30], which showed that the proposed algorithm is accurate. Please refer to [D] for the details of the evaluation and for the proof of Equation (13).

4 Methodology

Existing or already specified telecommunication network systems are generally evaluated by means of measurements, simulations or analytical formalism, which provide deep understanding of our networks. Nonetheless, these evaluation methods on their own are not necessarily suitable to define the innovation involved in the evolution of telecommunication networks. Advancement requires the specification of new architectures, protocols and algorithms, which then can be evaluated by the widely applied evaluation methods.

A new Ethernet architecture has been defined in Thesis 1 in order to meet the resilience demands, which includes the definition of protocol components and algorithms. The operation and performance of these protocols and algorithms were verified by simulations and measurements on a prototype implementation. The performance of the architecture was evaluated by means of measurements.

Protocol extensions were proposed in Thesis 2 in order to accommodate the link state principle being introduced in Ethernet networks. The operation of the Neighbour Synchronisation loop prevention technique was verified in a prototype implementation by means of measurements. Furthermore, the characteristics of the proposals were determined by means of packet level simulations.

New algorithms for Connection Admission Control have been defined for the Iub interface of UTRAN networks in Thesis 3. These algorithms apply analytic methods and they were verified by means of simulations.

5 Application of the Results

Protection for point-to-point and point-to-multipoint Ethernet services is provided by the Provider Backbone Bridge Traffic Engineering (PBB-TE) [31] standard, which applies similar principles to the resilient architecture defined by Thesis 1. Furthermore, protection for multipoint services in an SPB controlled Ethernet network can be implemented along the principles of Thesis 1 as described in detail in [P1]. Patent application [P16] is submitted on the failure handling protocol of Thesis 1.1. The tree computation algorithm defined by Thesis 1.3 is covered by patent application [P15] and it can be also applied within the multi-topology routing framework proposed by Menth [32] and Čičić [13] for IP resilience. Patent application [P11] was submitted on the topology discovery algorithm of Thesis 1.3. Furthermore, the prototype implementation of Thesis 1 was the winner at an Ericsson-wide prototype competition in 2006.

Thesis 2.2 has become the basic principle of the standard loop prevention solution for SPB [9]. Furthermore, Thesis 2.2 can be also applied as a loop prevention method in the IP Fast Re-Route framework. Patent application [P7] was submitted on Thesis 2.2. Patent applications [P5, P8, P9, P10] are related to Thesis 2.5.

The CAC algorithm specified by Thesis 3.1 can be applied in IP UTRAN networks. The CAC algorithm of Thesis 3.1 was submitted in patent application [P18]. A dimensioning algorithm, e.g. the one specified in [P13], may also take into account the CAC algorithm implemented in transport network nodes.

Acknowledgements

I express my deepest gratitude to all people who have contributed to this work.
First, I thank to my supervisors Dr. László Györfi and Dr. Csaba Antal for their help and support during my Ph. D. studies and research.
I thank to all my colleagues at Traffic Lab for the inspirative atmosphere and for the joint work, especially to Dr. Csaba Antal. I am also grateful to all my co-authors for the joint research.
I also thank Dr. András Császár, Dr. Sándor Rácz and Dr. Gábor Rétvári for their helpful suggestions and review.
I thank for all the support I have got from Ericsson Traffic Lab and HSN Lab.
Above all, I thank Beatrix and to my parents for their love and support.

References

[1] IEEE Std. 802.1D-2004, "IEEE Standard for local and metropolitan area networks: Media Access Control (MAC) bridges," 2004.

[2] IEEE Std. 802.1Q-2005, "IEEE standard for local and metropolitan area networks: Virtual bridged local area networks," 2005.

[3] A. Myers, T. S. E. Ng and H. Zhang, "Rethinking the service model: scaling Ethernet to a million nodes," *Third Workshop on Hot Topics in Networks (HotNets-III)*, San Diego, November, 2004.

[4] ITU-T Std. G.8032, "Ethernet ring protection switching," March 2010.

[5] IEEE Std. 802.17, "Resilient packet ring," 2004.

[6] IETF RFC 3619, "Ethernet automatic protection switching," October 2003.

[7] S. Sharama, K. Gopalan, S. Nanda, and T. Chiueh, "Viking: A multi-spanning-tree Ethernet architecture for metropolitan area and cluster networks," in *Proceedings of IEEE InfoCom 2004*, March 2004.

[8] ISO/IEC 10589, "Information technology – Telecommunications and information exchange between systems – Intermediate system to intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)," 2nd ed., 2002.

[9] IEEE Draft Std. 802.1aq D3.6, "IEEE draft standard for local and metropolitan area networks: Virtual bridged local area networks – Amendment 9: Shortest path bridging," February 2011.

[10] IETF RFC 5556, "Transparent Interconnection of Lots of Links (TRILL): Problem and applicability statement," May 2009.

[11] R. Perlman et al. , "RBridges: base protocol specification," Internet Draft, March 2010.
https://tools.ietf.org/html/draft-ietf-trill-rbridge-protocol-16

[12] D. Eastlake, "Future work for TRILL 2," Technical presentation at IETF79, November 2010.
http://www6.ietf.org/proceedings/79/slides/trill-7/trill-7_files/trill-7.pptx

[13] T. Cicic, "On basic properties of fault-tolerant multi-topology routing," *Computer Networks Journal*, Elsevier, Vol. 52, pp. 3325-3341, 2008.

[14] T. Cicic et al., "Relaxed multiple routing configurations: IP fast reroute for single and correlated failures," *IEEE Transactions on Network and Service Management*, Vol. 6/1, pp. 1-14, March 2009.

[15] T. Cicic, "An upper bound on the state requirements of link-fault tolerant multi-topology routing," in *Proceedings of ICC 2006: IEEE International Conference on Communications*, Vol. 2, pp. 1026-1031, Istanbul, June 2006.

[16] IETF RFC 1157, "A Simple Network Management Protocol (SNMP)," May 1990.

[17] IETF RFC 1493, "Definitions of Managed Objects for Bridges," July 1993.

[18] IETF RFC 1213, "Management Information Base for Network Management of TCP/IP based internets: MIB-II," March 1991.

[19] IETF RFC 2233, "The Interfaces Group MIB using SMIv2," November 1997.

[20] IEEE Std. 802.1AB, "IEEE standard for local and metropolitan area networks: Station and media access control connectivity discovery," 2005.

[21] M. Shand and S. Bryant, "A Framework for Loop-free convergence," Internet Draft, October 2009.
http://tools.ietf.org/html/draft-ietf-rtgwg-lf-conv-frmwk-07

[22] D. Allan, N. Bragg, J. Chiabaut and D. Fedyk, "802.1aq link state protocol, SPBB multicast loop prevention," Technical presentation, July 2008.
http://www.ieee802.org/1/files/public/docs2008/aq-fedyk-Loop-Prevention-0708-v01.pdf

[23] OMNeT++
http://www.omnetpp.org

[24] Survivable fixed telecommunication Network Design library (SNDlib),
http://sndlib.zib.de

[25] M. L. Garcia-Osma, "TID scenarios for advanced resilience," Technical Report of The NOBEL Project, Work Package 2, Activity A.2.1, Advanced Resilience Study Group, September 2005.

[26] GNU Quagga routing software
http://www.quagga.net

[27] S. Malomsoky, S. Rácz and S. Nádas, "Connection admission control in UMTS radio access networks," *Computer Communications – Special Issue on 3G Wireless and Beyond for Computer Communication*, Vol. 26, pp. 1907-1917, November, 2003.

[28] S. Malomsoky, "Resource management problems in packet switched networks," *Ph.D. dissertation*, Budapest University of Technology and Economics, 2003.

[29] J. W. Roberts eds., "Methods for the performance evaluation and design of broadband multiservice networks," Part III, Traffic models and queuing analysis, The COST 242 Final Report,1996.

[30] T. Westholm and B. Olin, "A model for GSM speech," in *Proceedings of 2000 Symposium on Performance Evaluation of Computer and Telecommunication Systems*, pp. 458-62, July 2001.

[31] IEEE Std. 802.1Qay, "IEEE standard for local and metropolitan area networks: Virtual bridged local area networks – Amendment 10: Provider backbone bridge - traffic engineering," 2009.

[32] M. Menth and R. Martin, "Network resilience through multi-topology routing, in *Proceedings of DRCN 2005: Design of Reliable Communication Networks*, pp. 515-522, Ischia, October 2005.

Publications

[D] J. Farkas "Resilience and quality of service assurance methods in access and metro networks," *Ph.D. dissertation*, submitted to Budapest University of Technology and Economics, 2011.

Journal Papers

[J1] D Allan, **J. Farkas** and S. Mansfield, "Intelligent load balancing for shortest path bridging," *IEEE Communications Magazine*, 2011.

[J2] D. Allan, P. Ashwood-Smith, N. Bragg, **J. Farkas**, D. Fedyk, M. Ouellete, M. Seaman and P. Unbehagen, "Shortest path bridging: Efficient control of larger Ethernet networks," *IEEE Communications Magazine*, October 2010.

[J3] **J. Farkas**, A. Paradisi and C. Antal, "Low-cost survivable Ethernet architecture over fiber," *Journal of Optical Networking*, Vol. 5, Issue 5, pp. 398-409, April 2006.

[J4] T. Kumli and **J. Farkas**, "Real-time simulation of public switched telephony networks," *Infocommuniations Journal*, Vol. XLIX, in Hungarian, Budapest, November 1998.

[J5] B. P. Gerő, **J. Farkas**, S. Kini, P. Saltsidis and A. Takács, "Upgrading the metro Ethernet network," *submitted for review to IEEE Communications Magazine*

Conference Papers

[C1] **J. Farkas** and Z. Arató, "Performance analysis of shortest path bridging control protocols," in *Proceedings of GlobeCom 2009: IEEE Global Communications Conference*, Honolulu, December 2009.

[C2] **J. Farkas** and R. Pallos, "Root controlled bridging: A scalable control protocol for shortest path bridging," in *Proceedings of Networks 2008: 13th International Telecommunications Network Strategy and Planning Symposium*, Budapest, September 2008.

[C3] **J. Farkas**, V.G. Oliviera, M.R. Salvador and G.C. Santos, "Automatic discovery of physical topology in heterogeneous multi-vendor Ethernet networks," in *Proceedings of ICC 2008: IEEE International Conference on Communications*, pp. 2055-2060, Beijing, May 2008.

[C4] **J. Farkas**, V.G. Oliviera, M.R. Salvador and G.C. Santos, "Automatic discovery of physical topology in Ethernet networks," in *Proceedings of AINA 2008: Advanced Information Networking and Applications*, pp. 848-854, Okinawa, March 2008.

[C5] R. Pallos, **J. Farkas**, I. Moldován and C. Lukovszki, "Performance of rapid spanning tree protocol in access and metro networks," in *Proceedings of AccessNets 2007: Second International Conference on Access Networks*, Ottawa, August 2007.

[C6] C. Lukovszki, I. Moldován, A. Kern, **J. Farkas**, W. Zhao and Z. Ghebretensaé, "Standard-based physical and active topology discovery in Ethernet-based aggregation networks," in *Proceedings of NOC 2007: 12th European Conference on Networks and Optical Communications*, Stockholm, June 2007.

[C7] **J. Farkas**, C. Antal, L. Westberg, A. Paradisi, T.R. Tronco and V.G. Oliviera, "Fast failure handling in Ethernet networks," in *Proceedings of ICC 2006: IEEE International Conference on Communications*, Vol. 2, pp. 841-846, Istanbul, June 2006.

[C8] **J. Farkas**, C. Antal, G. Tóth and L. Westberg, "Distributed resilient architecture for Ethernet networks," in *Proceedings of DRCN 2005: Design of Reliable Communication Networks*, pp. 515-522, Ischia, October 2005.

[C9] S. Rácz, T. Jakabfy, **J. Farkas** and C. Antal, "Connection admission control for flow level QoS in bufferless models," in *Proceedings of InfoCom 2005: 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1273-1282, Miami, March 2005.

[C10] G. Mátéfi, **J. Farkas** and C. Antal, "Towards efficient call admission control in IP UTRAN," in *Proceedings of ITC 18: 18th International Teletraffic Congress*, pp. 238-253, Berlin, September 2003.

Patent Applications

[P1] **J. Farkas**, "Method and arrangement for multipoint service protection in Ethernet networks," International Patent Application, WO/2011/038750, 2011.

[P2] **J. Farkas**, "System, network management system and method for avoiding a count-to-infinity problem," International Patent Application, WO/2011/058450, 2011.

[P3] D. Jocha and **J. Farkas**, "Loss measurement for multicast data delivery," International Patent Application, WO/2011/003478, 2011.

[P4] **J. Farkas**, R. Pallos, G. Kapitány, S. Plósz and D. Horváth "Port table flushing in Ethernet networks," International Patent Application, WO/2010/086022, 2010.

[P5] **J. Farkas**, C. Antal and A. Takács "Multiple tree registration protocol," International Patent Application, WO/2010/007467, 2010.

[P6] **J. Farkas**, C. Antal and A. Takács "Method and apparatus for Ethernet protection with local re-routing," International Patent Application, WO/2009/115480, 2009.

[P7] **J. Farkas**, "Method and apparatus for link-state handshake for loop prevention," International Patent Application, WO/2009/112929, 2009.

[P8] **J. Farkas**, C. Antal, A. Takács and P. Saltsidis, "Ethernet spanning tree provision," International Patent Application, WO/2008/125144, 2008.

[P9] **J. Farkas**, C. Antal, A. Takács and P. Saltsidis, "Method and apparatus for network tree management," International Patent Application, WO/2008/087547, 2008.

[P10] **J. Farkas**, C. Antal, A. Takács and P. Saltsidis, "Method, bridge and computer network for calculating a spanning tree based on link state advertisements," International Patent Application, WO/2008/087543, 2008.

[P11] **J. Farkas**, V.G. Oliveira and M.R. Salvador, "Method of discovering physical topology of a telecommunications network," International Patent Application, WO/2008/076052, 2008.

[P12] **J. Farkas**, W. Zhao, "Method for fault localisation in multiple spanning tree based architectures," International Patent Application, WO/2008/095538, 2008.

[P13] **J. Farkas**, S. Nádas, C. Antal, S. Rácz, S. Malomsoky and U. Rosberg, "Dimensioning link capacity in a packet switched telecommunications network," International Patent Application, WO/2008/120077, 2008.

[P14] P. Lundh, C. Faronius, **J. Farkas**, S. Rácz and S. Nádas, "Enhanced flow control in a cellular telephony system," International Patent Application, WO/2008/066430, 2008.

[P15] **J. Farkas**, and G. Tóth, "Centralised calculation of minimum number of multiple spanning trees," International Patent Application, WO/2007/043919, 2007.

[P16] **J. Farkas**, C. Antal and L. Westberg, "Method and arrangement for failure handling in a network," International Patent Application, WO/2006/135282, 2006.

[P17] G. Mátéfi, **J. Farkas** and T. Éltető, "Method and device for audience monitoring on multicast capable networks," United States Patent Application, US 2006/0294259 A1, 2006.

[P18] G. Mátéfi, **J. Farkas** and C. Antal, "Connection admission control system and method for interpreting signalling messages and controlling traffic load in Internet protocol differentiated services networks," International Patent Application, WO/2005/022851, 2005.

Standardisation Contributions

[S1] **J. Farkas**, "Notes on IS-IS network convergence," Technical presentation, November 2010.
http://www.ieee802.org/1/files/public/docs2010/new-farkas-convergence-1110.pdf

[S2] Clause 28.9 of the IEEE Std. 802.1aq D3.2, "IEEE Draft Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks - Amendment 9: Shortest Path Bridging," October 2010.

[S3] **J. Farkas**, "CFM in 802.1aq SPB," Technical presentation, July 2010.
http://www.ieee802.org/1/files/public/docs2008/aq-farkas-CFM-in-802.1aq-0908.pdf,
http://www.ieee802.org/1/files/public/docs2008/aq-farkas-proposal-for-CFM-in-SPB.pdf

[S4] Clauses 28.7 and 28.8 of the IEEE Std. 802.1aq D3.0, "IEEE Draft Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks - Amendment 9: Shortest Path Bridging," June 2010.

[S5] Clause 27.7 of the IEEE Std. 802.1aq D2.0, "IEEE Draft Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks - Amendment 9: Shortest Path Bridging," June 2009.

[S6] **J. Farkas**, "802.1aq: Link-state handshake for loop prevention," Technical presentation, March 2008.
http://www.ieee802.org/1/files/public/docs2008/aq-farkas-link-state-handshake-0308.pdf

[S7] **J. Farkas**, "802.1aq: link-state protocol and loop prevention," Technical presentation, November 2007.
http://www.ieee802.org/1/files/public/docs2007/aq-farkas-loop-prevention-1107-v02.pdf