Some Methods to Increase Network Resilience and Recovery Performance

Teză destinată obţinerii
titlului ştiinţific de doctor inginer
la
Universitatea "Politehnica" din Timişoara
în domeniul INGINERIE ELECTRONICĂ
ŞI TELECOMUNICAŢII
de către

Ing. Florin-Josef Lătărețu

Conducător ştiinţific:	prof.univ.dr.ing. Corneliu Ioan Toma
Referenţi ştiinţifici:	prof.univ.dr.ing. Mircea Petrescu
	prof.univ.dr.ing. Gavril Toderean
	prof.univ.dr.ing. Miranda Monica Nafornită

Ziua susţinerii tezei: 03.12.2010

Seriile Teze de doctorat ale UPT sunt:

1. Automatică
2. Chimie
3. Energetică
4. Ingineria Chimică
5. Inginerie Civilă
6. Inginerie Electrică
7. Inginerie Electronică şi Telecomunicaţii
8. Inginerie Industrială
9. Inginerie Mecanică
10. Ştiinţa Calculatoarelor
11. Ştiinţa şi Ingineria Materialelor
12. Ingineria sistemelor

Universitatea „Politehnica" din Timişoara a iniţiat seriile de mai sus în scopul diseminării expertizei, cunoştinţelor şi rezultatelor cercetărilor întreprinse în cadrul şcolii doctorale a universităţii. Seriile conţin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susţinute în universitate începând cu 1 octombrie 2006.

România, 300159 Timişoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Teza de doctorat a fost elaborată pe parcursul activităţii mele în cadrul Departamentului de Comunicații al Universităţii „Politehnica" din Timişoara.

Lucrarea de faţă este o abordare nouă a ceea ce in termenul de specialitate este cunoscut sub numele de *network resilience,* adică aptitudinea reţelei de a recupera dintr-o situaţie de defecţiune fiind în măsură să asigure serviciile in continuare, fără intrerupere. Una din problemele actuale de interes in reţelele moderne automatizate optice este prelucrarea automată a defectiunilor fără interventia operatorului, aşa încît reţeaua să revină la funcţionalitatea iniţială deplină in timp cît mai scurt.

Teza îşi propune să găsească mijloace care sa îmbunătăţească performanţa recuperării si rezilienţa reţelelor in contextul ASON. In particular m-am preocupat în acest sens cu protocolul RSVP care joacă un rol central in semnalizare.

Cercetările efectuate au condus la realizarea unei metode care să accelereze stabilirea căilor *Label Switched Path (*LSP), de care se poate beneficia in regim normal dar şi de recuperare. Această metodă este in curs de patentare in S.U.A. O alta metodă propusă îmbunătăţeşte protecţia pachetelor 1+1 prin monitorizarea căilor LSP redundante in scopul detecţiei degradărilor în faza incipientă. Această metodă a obţinut brevetul de patent.

În încheiere doresc să aduc mulţumiri deosebite conducătorului de doctorat prof.dr.ing.Corneliu I. Toma, care prin supraveghere constantă şi multe sfaturi utile a contribuit la realizarea prezentei lucrări.

Timişoara, 10. 2010 Florin Lătăreţu

Lătăreţu, Florin-Josef

Some Methods To Increase Network Resilience And Recovery Performance

Teze de doctorat ale UPT, Seria 7, Nr. 28, Editura Politehnica, 2010, 128 pagini, 42 figuri.

ISSN: 1842-7014

ISBN: 978-606-554-203-7

Cuvinte cheie: network resilience, network recovery, network reliability, control plane, signaling protocols, RSVP, MPLS Packet Protection.

Rezumat:
Teza de doctorat este dedicată unei probleme de continuu interes in reţelele moderne de telecomunicaţii, şi anume, prelucrarea automată si recuperarea situaţilor de defecţiune fără intervenţia operatorului. Problema constă in faptul că reţelele continuă să crească in mărime şi complexitate astfel încît personalul operator poate fi depăşit de o astfel de situaţie. Impactul economic şi financiar este considerabil. De aceea preocuparea mea s-a concentrat pe găsirea unor metode care să imbunătăţească performanţa recuperării si rezilienţa reţelelor (*network resilience*) in contextul reţelelor optice comutate automat (Automatically Switched Optical Network - ASON). In particular m-am preocupat in acest sens cu protocolul RSVP care joacă un rol central in semnalizarea planului de control. Ca rezultat am propus o metodă care sa accelereze stabilierea căilor *Label Switched Path* LSP, de care se poate beneficia in regim normal dar si de recuperare. Această metodă este mult mai eficientă decît metoda tradiţională pentru că pentru că foloseşte capacitea de procesare distribuită. Timpul pentru setup LSP este micşorat considerabil şi devine mai puţin dependent de lungimea conexiunii LSP. Pentru metoda de setup in parallel am propus două alternative teoretice de sincronizare: sincronizare secvenţială in nodurile intermediare şi sincronizare finală in nodul ultim. De asemenea am schiţat implementarea acestor metode ca o extindere a protocolului RSVP. Această implementare are faţă de propuneri similare, avantajul că e mai generică, nu necesită messaje noi şi poate fi folosită intr-o reţea neomogenă, in care nu toate elementele au aderat la versiunea extinsă a protocolului.

Deoarece cea mai eficientă recuperare este prevenirea, am propus de asemenea imbunătăţiri ale metodei consacrate *1+1 MPLS Packet Protection,* luând în considerare cerinţele specifice ale consumatorului, cât şi aspectele dinamice. In acest sens propun ciţiva parametri noi in măsură să detecteze din timp degradarea perfomanţei. In particular mărimea *SlidingWindow* îşi găseşte o semnificaţie extinsă faţă cea din standard. De asemenea schiţez o generalizare a acestei metode care să poate fi folosită pentru supravegherea latenţei diferitelor *subflowuri,* îmbunătăţind astfel resilienţa conexiunilor multipath TCP.

Table of Content

List of Abbreviations 8
1. Motivation 9
1.1. General Consideration on Recovery Performance and Resilience 9
1.2. Document Structure 13
1.3. Personal Publications 14
2. Network Recovery and Reliability 16
2.1. Definitions, Taxonomy 16
2.1.1. Network Classification 16
2.1.2. Network Reliability 17
2.1.3. Recovery Process 19
2.1.3.1. Recovery Cycle 19
2.1.3.2. Reversion Cycle 20
2.1.4. Criteria of Recovery Mechanisms 21
2.1.5. Characteristics of Recovery Mechanisms 22
2.2. IP Routing 25
2.2.1. Recovery Cycle 26
2.2.1.1. Fault Detection and Characterization 26
2.2.1.2. Hold-Off Time 27
2.2.1.3. Fault Notification 27
2.2.1.4. Recovery Operation 27
2.2.1.5. Traffic Recovery 28
2.2.2. Non Time Aspects 28
2.2.2.1. Traffic load balancing 28
2.2.2.2. Bandwidth Efficiency 29
2.2.2.3. Scalability 29
2.2.2.4. QoS during failure 29
2.3. MPLS Traffic Engineering 30
2.3.1. Mode of Operation & Protocol Extensions 31
2.3.1.1. MPLS TE Global Default Restoration 31
2.3.1.2. MPLS TE global protection 32
2.3.1.3. MPLS TE local protection 32
2.3.1.4. 1+1 Packet Protection 33
2.3.2. Recovery Cycle 34
2.3.2.1. Fault Detection 34
2.3.2.2. Hold-Off Time 35
2.3.2.3. Fault Notification 35
2.3.2.4. Recovery Operation 36
2.3.3. Non Time Aspects 36
2.3.3.1. Scalability 36
2.3.3.2. Bandwidth Efficiency 37
2.3.4. Comparison of Recovery Alternatives 38
2.3.4.1. Default Global Restoration 38
2.3.4.2. Global Path protection 38
2.3.4.3. Local Protection 39
2.3.4.4. 1+1 Packet Protection 39
2.4. Reliability Concepts Introduced By RSVP 40
2.4.1. Functional Specification of Resource Reservation Protocol RFC 2205 . 40

2.4.1.1. Soft State 41
2.4.1.2. Local Repair 41
2.4.1.3. Refresh Period and State Lifetime 42
2.4.2. RSVP Refresh Overhead Reduction Extensions RFC 2961 43
2.4.3. RSVP-TE: Extensions to RSVP for LSP Tunnels RFC 3209 45
2.4.3.1. LSP Tunnel 46
2.4.3.2. Abstract Node 47
2.4.3.3. Explicitly Routed LSP 47
2.4.3.4. Hello Extension 48
2.4.4. GMPLS Signaling RSVP-TE Extensions RFC3473 49
2.4.4.1. Notify Message 49
2.4.4.2. Control Channel Separation 50
2.4.4.3. Fault Handling 50
2.4.4.4. Procedures for the Restarting node 52
2.4.4.5. Procedures for the Neighbor of a Restarting node 53
2.4.5. GMPLS RSVP-TE Usage and Extensions for ASON-IETF RFC 3474 54
2.4.5.1. Support For Behaviors during Control Plane Failures 54
2.4.5.2. Supporting Objects 55
2.5. Conclusion 56
3. Method for fast source routed connection setup 59
3.1. Background 59
3.2. Basic Idea of the Proposal 61
3.3. Detailed Description 62
3.3.1. Parallel Setup With Sequential Synchronization 62
3.3.2. Parallel Setup With Final Synchronization 64
3.4. Benefits & validation 65
3.5. Implementation in the RSVP context 68
3.5.1. Sequential Synchronisation 68
3.5.2. Final Synchronisation 70
3.5.3. Conclusions 71
4. Improved 1+1 MPLS Packet Protection By Preventive Detection Of Quality Degradation 73
4.1. Background 73
4.2. Basic Ideas 76
4.3. Detailed Description 77
4.4. An example 79
4.5. Contributions 85
4.6. Future Work 85
4.7. Latency Control and Congestion Prevention 88
5. Some Recent Proposals for End-to-End Recovery 91
5.1. Inter-Domain Label Switching Traffic Engineering 91
5.2. GMPLS End-2-End Recovery 93
5.3. GMPLS Segment Recovery 107
6. Conclusions - Personal Contributions – Future directions 111
7. References 117

List of Figures

Fig. 1 Transport Plane, Management and Control Plane 9
Fig. 2 Multi domain Automatically Switched Networks (ASONs) 10
Fig. 3 Recovery Cycle (from RFC3469 [53]) 20
Fig. 4 Reversion Cycle (from RFC3469 [53]) 21
Fig. 5 Path Restoration 31
Fig. 6 Path Protection 32
Fig. 7 Local Protection 33
Fig. 8 1+1 Packet Protection 34
Fig. 9 Sequential Setup Request 59
Fig. 10 Sequential Setup Response 60
Fig. 11 Parallel Setup with sequential synchronization 62
Fig. 12 Parallel Setup With Final Synchronization 64
Fig. 13 Sequential Setup in a meshed topology 66
Fig. 14 Sequential Synchronization in a meshed 67
Fig. 15 Final Synchronization in a meshed topology 67
Fig. 16 Sequential Synchronization: Modified behavior 68
Fig. 17 PrePath message sent by A to I2 69
Fig. 18 Final Synchronization: Modified behavior 70
Fig. 19 PostPath message sent by I2 to Z 71
Fig. 20 Mechanism of Packet 1+1 Protection - ITU-T G.7712 73
Fig. 21 Sliding window mechanism for wrap around 74
Fig. 22 Relation between Sliding and Delay Window 75
Fig. 23 Short Sliding Window (compared with Delay Window) 76
Fig. 24 Initial state 79
Fig. 25 No packets lost at t6 80
Fig. 26 Loosing 2 packets on LSPlead at t(24+6) 81
Fig. 27 Loosing 1 packet on LSPtrail at t(24+10) 82
Fig. 28 Loosing 2 packets on LSPtrail at t(24+9) 83
Fig. 29 CSW must be increased because CDW exceeded at t(24+12) 84
Fig. 30 Signaling methods for multi domains 92
Fig. 31 1+1 Unidirectional Recovery 95
Fig. 32 1+1 Bidirectional Recovery 96
Fig. 33 1:1 Protection with Extra-Traffic 97
Fig. 34 End-to-end Rerouting (Restoration) without Extra-Traffic 99
Fig. 35 End-to-end Rerouting (Restoration) with Extra-Traffic 100
Fig. 36 Shared-Meshed Restoration 101
Fig. 37 LSP Preemption 103
Fig. 38 (Full) LSP Rerouting 104
Fig. 39 Reversion for 1+1 Bidirectional Protection 105
Fig. 40 Segment Topology 107
Fig. 41 Explicit Control of LSP Segment Recovery: Path message processing 109
Fig. 42 Explicit Control of LSP Segment Recovery: Resv message 110

List of Abbreviations

ASON	Automatic Switched Optical Network
BGP	Border Gateway Protocol
CP	Control Plane
CPE	Customer Premises Edge
CSPF	Constraint Shortest Path First
E-NNI	Exterior Network to Network Interface
ERO	Explicit Route Object
FRR	Fast Reroute
GMPLS	Generalized Multiprotocol Label Switching
IGP	Interior Gateway Protocol
I-NNI	Interior Network to Network Interface
IS-IS	Intermediate System to Intermediate System routing protocol
ITU	International Telecommunication Union
LSA	Link State Advertisement
LSP	Label Switched Path
LSR	Label Switched Router
MP	Merge Point
MPLS	Multiprotocol Label Switching
NHOP	Next_Hop Bypass tunnel: to bypass a single link
NNHOP	Next_Hop Bypass tunnel: to bypass a single node
NVM	Non volatile Memory
OIF	Optical Interworking Forum
OSPF	Open Shortest Path First
OTN	Optical Transport Network
PDU	Protocol Data Unit
PLR	Point of Local Repair
QoS	Quality of Service
RHE	Router Head End
RRO	Record Route Object
RSVP	Resource Reservation Protocol
RTE	Router Tail End
SERO	Secondary Explicit Route
SRRO	Secondary Record Route
SLA	Service Level Agreement
SPF	Shortest Path First
SRLG	Shared Risk Link Group
TE	Traffic Engineering
TP	Transport Plane
UNI	User Network Interface
VoIP	voice over IP
VPN	virtual Private Network

1. Motivation

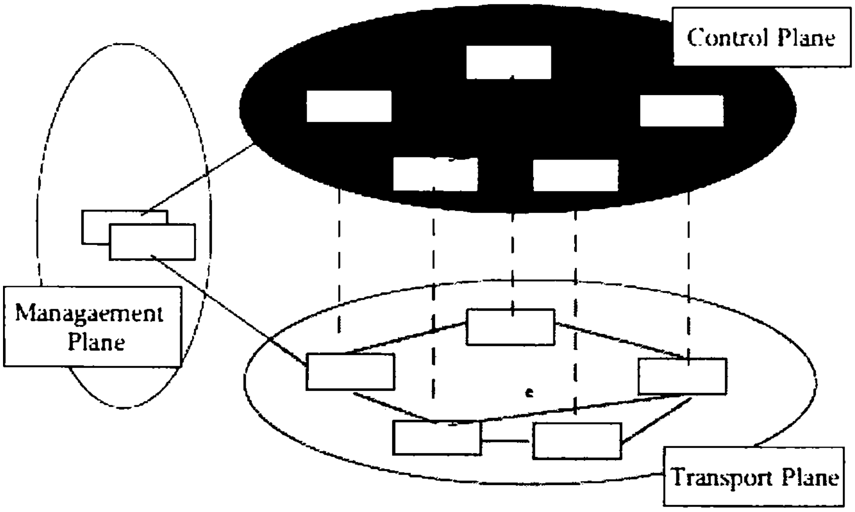
This introductory chapter is structured as follows: The first section presents the context of my preoccupation with network resilience and recovery performance and underlines the actuality of these themes. The second section describes the document structure. The final section summarizes my related publications.

1.1. General Consideration on Recovery Performance and Resilience

The data transported over the networks is permanently increasing at a high rate. At the same time the networks are rapidly increasing in size and complexity.

Modern transport networks, which provides the actual transfer of the user information requires besides the usual *management plane* which is in charge for the traditional so called "FCAPS" functional areas (fault, configuration, accounting, performance and security management) also a *control plane* responding to the need for intelligent control capabilities, which can be provided in an automated manner, independent from the management plane.

Fig. 1 Transport Plane, Management and Control Plane

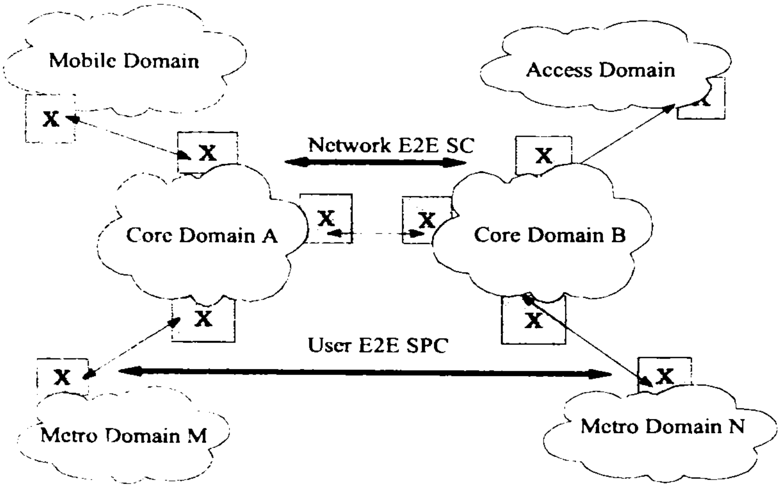


Given the size and complexity of modern networks, operators may be overstressed, so there is also an ongoing trend towards automation. For the optical networks the standardization bodies (ITU-T, OIF) agreed on the Automatically Switched Optical Network (ASON) (see [1], [2]).

The ASON is the reference architecture for the optical control plane describing the key components and their interaction in a multi-vendor environment.
The purpose of the Automatic Switched Optical Network control plane is to provide means for:

1. Fast and efficient configuration of connections within a transport layer network. The control plane supports connection set-up and tear down as a result of:
   - user request as Switched Connection (SC) or
   - management request Soft Permanent Connection (SPC). (See Fig. below)
2. Reconfiguration or modification of the existing connections
3. Autonomously re-establishment of failed connections (e.g. by restoration). The corresponding state information of the connection (e.g. fault and signal quality) is detected and exchanged by the transport plane (e.g. via overhead or OAM messages) and/or via the management plane (including the DCN).

Fig. 2 Multi domain Automatically Switched Networks (ASONs)



Mobile, Access, Metro Domains which are possibly managed and/or controlled by different authorities are inter-connected as shown in the next figure. The key concepts in this context are the Switched Connection (SC) and the Soft Permanent Connection (SPC) mentioned before.
Switched Connection is a connection that is established, as a result of a request from the end-user, between connection end points using a signaling/control plane.
Soft Permanent Connection is a user-to-user connection where by the user-to-network portion of the end-to-end connection is established by the network management system as a permanent connection (PC). The network portion of the end-to-end (E2E) connection is established as a switched connection.

For both cases· dynamic exchange of signaling information between signaling elements within the control plane(s) is required.

In addition a multitude of new services is emerging so that not only quantity but also quality issues are becoming the essential differentiation factors. One basic motivation was to find ways to handle network failures (node, link) automatically, without the operator contribution, so that the network can recover from failures in the shortest time reaching again its full functionality.

Therefore my current work focused on the recovery performance and resilience aspects of the networks. The initial context was the ASON control plane, however most of the statements may apply also for the WSON, some of them also in the mobile context. This is particularly valid for my proposed methods [10] and [11].

The ASON architecture was further specified and aligned with some IETF protocols by the Optical Interworking Forum (OIF) resulting in standardized UNI, E-ENNI and I-NNI interfaces. The OIF framework is based on the GMPLS protocol suite which extends the MPLS technology for circuit switching as well as for non-IP based systems. The protocol suite contains:

- Generalized RSVP-TE (alternatively CR-LDP) for signaling
- OSPF with TE extensions for inter-area routing
- ISIS with TE extensions for intra-area routing
- LMP for link management and discovery functions

I spend particular attention to the RSVP protocol which plays a central role for the signaling inside the GMPLS protocol family.

GMPLS and RSVP in particular initiated a paradigm shift and continues to have a major technological impact for this and next generation networking.

ITU-T G.8080 requires in general: "*A well-designed control plane architecture should give service providers control of their network, while **providing fast and reliable call set-up**.*

*The **control plane itself** should be **reliable, scalable and efficient**. It should be sufficiently generic to support different technologies, differing business needs and different distribution of functions by vendors (i.e. different packaging of the control plane components)."* [1]

In particular the OIF requires for the SCN in its Design Guide [3] predictable performance in the light of varying network situations. Here some of the significant requirements:

R-1 Established Calls and Connections MUST not be impacted by an SCN failure.
R-2 Calls and Connections that are actively being restored MUST have priority over New Calls and Connections being presented to the Network.

R-3 Control Plane communications MUST avoid overload under failure/overload conditions, by ensuring that critical messages shall not get locked out and control messages shall not overwhelm the control plane operations.

R-4 Routing updates and individual types of signaling messages MUST be assigned priority levels, so that control can be exercised during failure/overload conditions to manage potential signaling storms and avoid catastrophic control plane failures in the network.

R-5 Under failure/overload conditions alarm network management messages, critical topology updates, and signaling for connection restoration MUST have priority over other control, signaling and management messages.

Despite its acceptance and some implementations in the field, the research and development activities continues moving from basic aspects of GMPLS architecture to aspects of **performance improvement** and looking for ways to increase **resilience** as a key pre-requisite for deployment on a wide scale.

This was the **motivation to address in my PhD thesis** some methods to improve the performance of network recovery and to increase network resilience in general.

My contributions are focusing on two aspects designated to improve the *resilience* - as the ability to recover from faults and to provide uninterrupted service - of the control plane:

- Performance aspect by presenting a method to speed up the setup of Label Switched Path (LSP). This method is particularly effective for some restoration schemes, so that not only regular setup is improved but also the recovery performance.
- Reliability aspects by presenting a method to enhance the existing MPLS 1+1 Packet Protection by monitoring redundant LSPs for the purpose of early detection of quality degradations.

Because the heterogeneity of the approaches, I felt the necessity to summarize in a systematical manner, as a theoretical preparation, the current stadium of available information on network recovery, reliability and resilience.

My practical activities started with detailed network measurements which have been summarized in a lab report [5]. In this report I analyzed the performance of the signaling in a meshed SDH/SONET network. The outcome was that efficient signaling via IP messages is basically possible. At the same time I identified some opportunities for improving performance.

As a consequence I elaborated the "Method for fast source routed connection setup", which applies not only to regular setup but also to the recovery procedure.

Given the fact that the most effective recovery is the prevention, I'm proposing enhancements for the traditional 1+1 MPLS Packet Protection.

1.2. **Document Structure**

In the chapter **Network Recovery and Reliability** I present an overview of the recovery aspects (protection, restoration) and reliability in the context of the current and emerging network technologies. My intention is to summarize the current available information which is Vasseur's "Network Recovery" [12] (at this moment the key reference on this topic) and some relevant RFCs (comp. Bibliography).

- First the taxonomy and the relevant definitions are introduced.
- Then IP Routing and MPLS Traffic Engineering are analyzed from the perspective of the recovery cycle. Both aspects: performance and non-time aspects (e.g. scalability) are considered.
- In section 4 I'm analyzing in detail the reliability key concepts introduced by the different RFCs.
- I'm showing that there is room for some improvements, as described below:

The next chapter is dedicated to my proposal for a **Method for fast source routed connection setup**. With this new method I address performance deficiencies of the conventional setup methods. This mechanism employs a parallel approach to network connection setup that utilizes existing distributed processing potential to minimize the dependency of connection setup time on connection path length. This way performance (in particular recovery performance) and consequently network reliability may be considerably increased.

Chapter 5 describes my second proposal, the **Improved 1+1 MPLS Packet Protection By Preventive Detection Of Quality Degradation.** I'm proposing a mechanism which extends the traditional MPLS 1+1 Packet Protection as described in ITU-TG.7712 by integrating specific application needs and taking into account the dynamics introduced by the supporting LSP.

In particular I'm introducing adequate parameters to measure the quality of the LSP to detect degradation and to initiate counter measurements. The mechanism can be used in any network which supports MPLS 1+1 Packet Protection. It is in particular recommended for time critical applications. The mechanism offers significant improvements in terms of reliability and restoration performance compared to the traditional MPLS Packet 1+1 Protection as described by ITU-T G.7712.

The next chapter is dedicated to the presentation of **Some Recent Proposals for End-to-End Recovery** as:

- inter-domain LSP Traffic Engineering,
- domain end-to-end recovery,
- LSP segment recovery.

In addition I'm analyzing the applicability of my proposed methods.

In the final chapter I'm summarizing the personal contributions.

1.3. **Personal Publications**

Previous research on shared protection focused on schemes, which minimize the use of real-time message exchange between network elements [13]-[17]. Some proposals even do restoration without message protocols at all, but have to compromise on some of the carrier-grade requirements. While this direction was certainly exciting I took the challenge to evaluate other options for fast and efficient message based signaling between real NEs, motivated by the fact that also BLSR/MS-SPRING protection uses some message based protocol over the K bytes. The results have been published in:

Efficient Signaling for Fast Restoration in Meshed Sonet/SDH Networks: a Lab-Report Florin Lataretu, Walter Rothkegel, Dieter Stoll, Lucent Technologies, Nürnberg, **ITG-Fachtagung Photonische Netze, Mai 2004**

The conclusion was that fast meshed restoration can be efficiently implemented even if the related signal protocols have a comparatively large overhead.

Since there was not too much information available at that time I started to analyze the suitability of RSVP for the ASON signaling procedure in "Stadiul actual si de perspectiva in sistemele moderne de comunicatii. Protocoale de rezervare a resurselor"[6].

In my second presentation: "Optimizari in sistemele moderne de comunicatii. Recuperarea si fiabilitatea retelelor" [7], I focused on the reliability and resilience aspects of the signaling protocols in the control plane.

My next step was the optimization of the signaling protocol which resulted in the following proposal to speed up the connection setup in the context of optical networks:

Method for Fast Source Routed Connection Setup, Inventor: Florin-Josef Lataretu, Pub. No.: US 2006/0034288 A1

The methods described in this patent application are suitable for multiple purposes, not only for the intra-domain (I-NNI) but also in the context of E-ENNI restoration. In this context, I'm proposing in **"Fast Source Routed Connection Setup - Proposal for a RSVP Implementation"** (submitted for TELCOR 2010 [9b]) an implementation of the mentioned method based on the current RSVP standard and I'm comparing it with a recent similar proposal.

In my third presentation "Simulari si rezultate privind optimizarile in sistemele moderne de comunicatii. Recuperarea si fiabilitatea retelelor" [8], I showed the benefit of applying the Fast Source Routed Connection Setup in the context of the the recent Segment and End to End Recovery proposals (comp. [19], [20]).

The next step was to find methods to improve the network resilience. As mentioned there was not too much literature available on this subjects except some IEEE publications on resilience ([21], [22]), network reliability ([23], [24]), network recovery ([30]) or network survivability ([26], [27]) In the mean time some additional material was published ([28], [32], [33], [34], [NREC3], [31], [29], [25]).

Following the idea that the most effective restoration is the prevention of the recovery, I chose a different approach and improved the packet 1+1 protection in order to allow the preventive detection of quality degradation, which resulted in:

Method for improved packet 1+1 protection, Inventor: Florin-Josef Lataretu, Patent No.: US 7,525,903 B2, April 2009

This method may be adapted and extended in order to improve the resilience of the emerging multipath technology. I made a corresponding proposal in the article **"Improving the Resilience of Multipath TCP by Latency Supervision"** [9] which was accepted for publication by the IADIS Applied Computing conference.

2. Network Recovery and Reliability

This chapter presents a overview of the reliability aspects and recovery (protection, restoration) in the context of the current and emerging network technologies. It summarizes the current state of available information in a systematical manner.

In particular the focus is on following protocols: IP Routing, MPLS and RSVP. There is no dedicated chapter for the recovery in the SDH layer since the corresponding mechanisms are already in place and undisputed.

This chapter is structured as follows:

First section introduces the taxonomy and the relevant definitions.

Section 2 and 3 analyzes IP Routing and MPLS Traffic Engineering from the perspective of the recovery cycle introduced in chapter 1.

IP Routing is already a traditional technique, although also in this area the research work is ongoing e.g. for SRLG SPF aware algorithms.

The main activities in the network communities (including standardization bodies) concentrate on the MPLS Traffic Engineering (MPLS and RSVP).

Network recovery aspects are partially included in the basic IETF RFCs related to RSVPs as mentioned in my first PhD presentations [6]. In the section 4 the key concepts introduced by the RSVP RFCs are analyzed in more details.

This chapter is mainly based on the Vasseur's "Network Recovery" [12] - at the moment still the key reference on this topic - and on the relevant RFC related to RSVP: [36], [37], [38], [39]. In the mean time some additional RFCs on this subject have been published ([40], [41], [42], [43]). For definitions and fundamentals concepts from the RFCs quotation marks have occasionally been left over for the sake of better readability.

2.1. Definitions, Taxonomy

2.1.1. Network Classification

Following network characteristics must be considered in general for the recovery process.

- Switching technology
    - **Circuit switching**: The information is transported through the network via circuits (e.g. path with a fixed bandwidth).
    - **Packet switching:** The information is split up in packets, which are sent one by one through the network. Packet switching is more efficient from the bandwidth usage perspective (statistical multiplexing) but requires more operation in the network nodes.

- Switching techniques
    - **Connection oriented:** requires an end to end connection to be established in advance of each communication session.
    - **Connectionless:** in general for shared multiple access approaches (e.g. Ethernet)

      Hybrid forms are also possible (see [44]).
- Traffic characteristics:
    - **symmetrical:** requires same bandwidth in each direction
    - **asymmetrical traffic**
    - **unidirectional**
    - **bidirectional traffic:** the route from a node A to a node Z is the same as from the node Z to the node A.
- Topology
    - **ring networks:** a set of nodes form a closed loop
    - **meshed networks.** Meshed may be seen as a collection of rings (comp. P-cycles approach).

  The topology of the data (user) plane (see below) is in general different from the topology of the control plane and from the management plane.

    - ***Data (user) plane*** transfers user information (payload). Every network layer has its own user plane.
    - ***Control plane*** handles signaling for the connection setup, supervision and tear down by transferring the control information through the network routing tables. By its nature it works in a distributed way.
    - ***Management plane*** consists of two parts:
        - *Layer management* for each layer
        - *Plane management* to ensure correct coordination between the different layers.

      Management plane is usually operating in a centralized way. However this may be a bottleneck in some critical situation, for instance when many network elements have to report abnormal conditions to the central manage instance.

2.1.2. Network Reliability

Related definitions:

Reliability: = probability of a network element to be fully operational *during a certain time frame* [E800]. Other sources defines the reliability as the probability that a system will *continue to perform satisfactorily* for a given period of time [47].

Availability: = probability of a network element to be operational *at one particular point in time.*

The availability (A) of a network element is usually computed as

$$A = 1 - MTTR/MTBF$$

where MTBF (*mean time between failures*) is the average length of the time interval that elapses between two subsequent failures, and MTTR (*mean time to repair*) is the average time needed to repair that network element.

More interesting is rather the availability of a path, which is in the product of the availability of the corresponding network elements and links, if their probability are independent. Since this independence is in general not given, because a NE failure leads to failures of related links, the overall path availability is less than the calculated product.

Still this does not reflect the fact that the traffic of a failed path may be rerouted without significant quality impact for the user. Therefore the next definitions.

Integrity: = ability of a network to provide the desired Quality of Service (QoS) to the services not only in a failure free network but also when network congestion or network failure occur.

Survivability: = ability of network to recover the traffic in the event of a failure, causing a few or no consequences for the user [48].

Survivability is a subset of the integrity. In practice the degree of survivability is used to denote the extend to which a network is able to recover.

From this perspective it becomes evident how important it is to have the complete picture of the fail and repair process - see below.

Resilience: = the ability to recover from faults and to provide uninterrupted service.

The related failure terminology [49] is as follows:

NE **defect**: = a decrease in the ability of a NE to performed a required function. Note that this implies the existence and utilization of measuring instruments which must be able to quantify this ability instead of a simple operational or not-operational condition.

NE **failure**: = the termination of the ability of a NE to performed a required function.

NE **fault**: = the inability of a NE to performed a required function.

The reliability requirements of communication networks depends on the type of users (safety critical, business critical, low cost, basic level users) and on the type of services transported through the network. The interesting criteria for these ones are the need for recovery (in general always present) and the *delay sensitivity,* which may set upper time limits for the recovery process.

The ***Service-Level Agreements*** (SLA) is a contract reflecting the reliability expectations between service provider and its customer: usually a minimal availability (e.g. 99,99) and a maximal down-time.

Measures to increase the reliability:

1. Prevent failures as much as possible e.g. by physical and administrative measurements, safer design, increased testing, etc.
2. Use the *dual homing* principle e.g. by redundant links to a critical node.
3. Use of SRLG (Shared Risk Link Group) group of resources affected by the same failure.
4. Use *network recovery* or *resilience scheme:* In case of failure the traffic is redirected form the working path to a recovery path. In order to have alternative routes, the so-called *single point of failure* must be avoided by design.

2.1.3. **Recovery Process**

2.1.3.1. Recovery Cycle

The different phases of this cycle are shown in the figure below.

- Failure detection Time

  Time between failure occurrence and its detection in the adjacent nodes. Depends for instance on the frequency of the signals sent, speed of failure detection in the lower layer and notification to the upper layers involved in the recovery, time to collect additional status information to be correlated in order to get an exact fault status information.

- Hold-Off Time

  Time period which have to expire before the fault notification is sent. This may allow the lower layer to repair the fault. In addition it may suppress the notification in case of a resource is toggling between operational and fault state. May be static or dynamic, as function of the failure frequency (*dampening*).

- Fault Notification Time

  Time between sending the notification messages and receiving them at the nodes involved in the recovery actions.

- Recovery Operation Time

  Time between the first and the last recovery action. This could include the exchange of messages with other nodes involved in recovery.

- Traffic Recovery Time

  Interval between the time when the traffic starts to use the recovery path and the time when the traffic is completely recovered. Is influenced by the propagation delay along the recovery path.

- Overall Recovery Time

  Time between failure occurrence and complete traffic recovery

```
   --Network Impairment
 |    --Fault Detected
 |    |    --Start of Notification
 |    |    |    -- Start of Recovery Operation
 |    |    |    |    --Recovery Operation Complete
 |    |    |    |    |    --Path Traffic Recovered
 |    |    |    |    |    |
 |    |    |    |    |    |
 v    v    v    v    v    v
---------------------------------------------------------
 | T1 | T2 | T3 | T4 | T5 |

T1   Fault Detection Time
T2   Fault Hold-off Time
T3   Fault Notification Time
T4   Recovery Operation Time
T5   Traffic Recovery Time
```

Fig. 3 Recovery Cycle (from RFC3469 [53])

2.1.3.2. Reversion Cycle

The new routes of the traffic after recovery may be less ideal than before the recovery (e.g. longer or congested path). Therefore either a subsequent dynamic rerouting protocol is initiated to optimize the network resource usage or the traffic is switched back to the initial working path once the failure is completely recovered. The different phases of the reversion cycle are shown in the Figure below.

- Fault Clearing Time

  Time between the failure repair and its detection at the upper layer involved in the reversion process.

- Hold-off Timer

  Time interval between detection and sending the repaired notification. Avoids toggling failures. May be static or dynamic, as function of the failure frequency (*dampening*).

- Fault Repaired Notification Time

  Time between sending the notification messages and receiving them at the nodes involved in the reversion actions.

- Reversion Operation Time

  Time between the first and the last reversion action. This could include the exchange of messages with other nodes involved.

- Traffic Reversion Time

Interval between the time when the traffic starts to re-use the working path and the time when the traffic is completely reversed. Is influenced by the propagation delay along the path.
In contrast to the recovery cycle which is initiated by some unforeseen events, the reversion cycle may be planned in advance. Well-controlled switched back with minimal disruption of the traffic is preferred instead of minimized overall recovery time.

```
  --Network Impairment Repaired
  |   --Fault Cleared
  |   |   --Path Available
  |   |   |   --Start of Reversion Operation
  |   |   |   |   --Reversion Operation Complete
  |   |   |   |   |   --Traffic Restored on
  |   |   |   |   |   |         Preferred Path
  |   |   |   |   |   |
  v   v   v   v   v   v
---------------------------------------------------------
  | T7 | T8 | T9 | T10| T11|
```

T7 Fault Clearing Time
T8 Clear Hold-Off Time
T9 Clear Notification Time
T10 Reversion Operation Time
T11 Traffic Reversion Time

Fig. 4 Reversion Cycle (from RFC3469 [53])

2.1.4. Criteria of Recovery Mechanisms

- **Scope of Failure Coverage**

The recovery schemes may be designed to cover particular failure scenarios: single link, single node, double link failure, SRLG failures.
The recovery schemes may be designed for a specified percentage of coverage: E.g. recovery of some percentage of the traffic volume, 100% coverage of intermediate node failure.

- **Recovery Time** is the time between a network failure and the point at which the traffic re-starts to flow through the recovery path.
  Short recovery times are preferred in general, however sometimes it may be a trade of with the quality of the recovered path (e.g. signal quality)
- **Backup Capacity Requirements** may be a function of different recovery schemes, layer at which the recovery mechanism operates, algorithm selecting the recovery path, traffic characteristics.

- **Guaranteed Bandwidth**. Some recovery mechanisms inherently guarantee that the full bandwidth of the affected traffic will be rerouted, some others cannot offer this guarantees for all situations.

- **Reordering and Duplication.** Switching from the working to the protecting path may result in reordering of packets (because different delays of the paths) or duplicate elimination. Such activities may have negative impact (complexity, time penalties) on the destination node.

- **Additive Latency and Jitter.** Longer recovery path may increase latency to the traffic. Some services may be sensitive to fluctuation of the delay for the data of the same traffic flow (jitter).

- **State overhead**. Is in general a function of the number of the recovery path. Different recovery mechanisms may have specific needs, leading to more or less over overhead.

- **Scalability**. Performance of the recovery mechanism is in general a function of the network size (nodes, links) and of the traffic transported over the network. Usually related to the state overhead (see before). Additional aspects influenced by network grows: recovery time, required backup capacity (see before).

Scalability is one of the most important criteria for a recovery scheme since it addresses the inherent grow of existing networks.

- **Signaling Requirements**. The recovery mechanism have specific signaling needs (e.g. large number of signaling message) resulting in specific resource requirements (CPU, bandwidth).

- **Stability** addresses the trade-of between quick reaction to recovery events (achieved by small values for the related time parameters) and potential instabilities e.g. caused by toggling link failures.

- **Notion of Recovery Class**. Some recovery mechanisms allows a specific handling (with different costs) for different classes of traffic (e.g. identified by their QoS).

2.1.5. Characteristics of Recovery Mechanisms

- **Backup Capacity: Dedicated versus Shared**
  A dedicated backup resource is one-to-one related to a particular working path. A shared resource is used by many paths (one-to-many). This is the preferred option if the probability for simultaneous faults is low. Benefit: more efficient resource usage.

- **Recovery Paths: Pre-planned versus Dynamic**

In the pre-planned version, the recovery path is calculated in advance for all accounted failure scenarios. In the dynamic version the calculation is "on the fly", therefore it is slower. However it has the advantage of possibly handling also unaccounted failures. This option is typically used with the shared backup capacity (see above).

- **Protection versus Restoration**

For the *protection* the recovery path are preplanned and fully signaled before a failure occurs. In case of *restoration*, the recovery path can be either pre-planned or dynamically allocated, but when the failure occurs, additional signaling will be needed to establish the restoration path. Protection is faster then restoration. However restoration can be more flexible and requires less backup capacity because its shared nature.

Protection variants:

- **1+1 (Dedicated Protection)**

  One dedicated protection path protects exactly one working path and the normal traffic is permanently duplicated at the recovery head end (RHE) on both paths. The RTE (recovery tail end) selects the signal with the highest quality. Alternative it selects always the working path, unless a signal defect is detected. Very efficient recovery time but very expensive bandwidth usage.

- **1:1 (Dedicated Protection with Extra Traffic)**

  One dedicated protection path protects exactly one working path, but in failure free condition the traffic is transmitted over only one path at the time. This leaves the opportunity to use the protection path for the transport of extra traffic, which is preempted in case of failure.

- **1:N Protection (Shared Recovery with Extra Traffic)**

  A specific recovery entity is dedicated to the protection of up to N (explicitly) identified working entities. In failure-free conditions, the recovery entity can be used for extra traffic.

- **M:N Protection (M<=N)**

  A set of M specific recovery entities protects a set of up to N specific working entities. The two sets are explicitly identified. Extra traffic can be transported over the M recovery entities when available.

- **Global versus Local Recovery**
  - In *local recovery* only the affected network elements are by passed. The RHE and the RTE are chosen as close to the failed network as possible. In case of a link failure a link disjoint recovery path is set up between the nodes adjacent to the failure. In case of a node failure the local recovery path is established between every two neighbor nodes of the failing node.
  - In *global recovery* the complete working path between source and destination is bypassed by the recovery path. The RHE and the RTE coincide with the source and destination of the working path.

Local recovery is usually faster because fault detection is faster (RHE and RTE are closer to failure). However the local perspective is limited, so the recovery path may be longer, sub-optimal, for instance crossing a particular link twice ("back hauling"). Global recovery is by nature network-wide optimizing. Therefore it requires in general less backup capacity, can cover failures in two successive nodes along the working path. However it may generate more state overhead.
Intermediate options applies for sub-portions of the working path. Compare "segment recovery (G-MPLS networks), subnet connection protection (SNCP, in the OTN networks).

- **Control of Recovery Mechanism**
    - *Centralized recovery mechanism* depend on a central controller which has to determine which recovery actions to take: determine where and when the failure occurred, get network-wide state information, issue (switching) commands to reconfigure the network.
    - *Decentralized or distributed recovery mechanism* operates without the intervention of a central control system. Instead local intelligence is available in the network elements, which autonomously initiate and steer the recovery. [12] argues that the global view may be missing. However this is not the case in general: IP routing has a global view despite its distributed nature. Typical example is the control plane in IP and G-MPLS networks.
    - Combination is also possible: centralized path computation and distributed failure detection and recovery decision.
      Distributed mechanism are in general more complex. However they scale better.
      Centralized mechanism have a better global view of the network, therefore possibly more efficient in required capacity. They allow in general also operator interaction in case of unaccounted catastrophes. However they are vulnerable by nature.

- **Ring Networks versus Mesh Networks**

  If failures occur in a ring topology, the traffic is rerouted along the other side of the ring. The recovery is performed in a ring by ring basis. Rings with common nodes need special attention/handling (single point of failure).
  Meshed topologies do not underlie such restrictions imposed by the routing pattern of the recovery path.

- **Connection-Oriented versus Connectionless**

  Relevant for the setup of the recovery path, depending on the nature of the network technology. Connection-oriented networks requires a connection setup in advance to the failure detection.

- **Revertive versus Non Revertive Model**

  Some recovery mechanism switch back from recovery path to the working path once the fault is completely repaired. This may be the preferred option to maximize the resource utilization along the recovery path (extra traffic). However the switch-back operation may have also negative effects (e.g. temporary hits).

- **Single Layer versus Multilayer Recovery**

  Realistic networks are today multi layered (e.g. IP-over-OTN). Each layer may perform its recovery mechanism (e.g. IP restoration and one-to-one optical protection) for its specific detected failure. However:

  - failure may affect different layers
  - some failure cannot be resolved by a recovery mechanism in the same layer.

  Therefore and in order to coordinate the recovery activities of the single layers a multilayer interworking must be considered.

  The ***sequential approach*** imposes a chronological order on the recovery mechanism is order to avoid racing conditions: A failure should successively resolved in different layers. Usually this is implemented by so called hold-off timers. An alternative implementation is based on a recovery token signal which is passed (after some time) from the server layer to the client layer.

  The ***integrated approach*** combines the mechanism of each layer in one integrated recovery scheme. This implies an adequate insight view on the layers in order to decide when and at which layer to take the appropriate actions. However this approach is very challenging because the related complexity.

2.2. IP Routing

IP routing is in fact a restoration protocol. It relies on the concept of routing algorithm with following characteristics:

- distributed: Any node computes the shortest path from itself to every other node in the network.
- dynamic: Routes are re-computed as soon as significant events occurs - as opposed to static routes.
- adaptive: May take in to account certain dynamic network state conditions e.g. link load, experienced delay.

Routing protocols are classified in two categories:

- Distance vector routing protocols - restricted relevance in field.
- Link state routing protocols: Each router is responsible for originating a link state protocol (LSP) data unit (PDU) that describes its local topology. Link state PDUs are disseminated throughout the network via a reliable flooding mechanism. Any router is able to build a complete map of the network based on the collection of all the link state PDUs, which is called link state database.

Link state with routing protocols with field relevance: ISIS and OSPF (also called in general Interior Gateway Protocols - IGP). Additional variants are available offering extensions for Traffic Engineering (TE). For instance ISIS-TE may carry router address information (type 1) or link information (type 2) in order to allow node disjoint, link or SRLG disjoint routing or LSP setup (see later).

Relevant characteristics:

- relative fast convergence time after network failures.
- network stability in case of network resource oscillation.

2.2.1. **Recovery Cycle**

2.2.1.1. Fault Detection and Characterization

Link failures results in loss of IP connectivity for that link. They may be caused by:

- Fiber cut
- optical equipment failures
- SONET/SDH equipment failures,
- router interface failure.

Node failures results in loss of IP connectivity for all links. They may be caused by:

- Power supply outage
- Route Processor failure
- Software Failures
- Planned Node Failure: e.g. for HW and SW upgrades

Failure detection is possible by:

- Lower Layers Failure Notification

  Optical and SONET/SDH layers may provide very fast link failure notification (less 10 ms). They do not work if the IP neighborhood is not identical to physical neighborhood (e.g. layer 2 switches in-between). Therefore additional protocols are proposed as for example Multiaccess Reachability Protocol (MARP) which allows a router to be notified of the local failure between the IP neighbor and the switch.

- Hello based mechanisms

  Periodical sending of hello message. When one router stops receiving hello messages it concludes on a failure. The corresponding time intervals are configurable. Examples of Hello based mechanisms

- IGP Hellos - Either ISIS or OSPF Hello mechanism
  They have a scalability impact: High frequency exchange may impact routing activity.

- Bidirectional Forwarding Detection (BFD)
  Is independent on a routing protocol and designed to require low processing overhead - see [50].

2.2.1.2. Hold-Off Time

Time interval after failure detection which the IP layer should wait for in case recovery actions of the lower layer (e.g. optical, SDH layer) are expected.

The hold-off timer may be dynamically computed when dampening techniques are used. Dampening is used as a counter measurement when facing flapping resources in order to avoid instabilities.

The basic idea is to consider an interface down as soon as it fails but to postpone the operational "up" state for a certain period of time. This period may be fixed ("up-state timer" algorithm) or calculated as a function of some accumulated penalties experienced by crossing thresholds (interface dampening using an exponential decay algorithm).

For the LSA propagation as well as for the SPF trigger exponential back-off algorithm are recommended as dampening mechanism: The timer for declaring a link down is doubled after each consecutive fault until a certain maximum Z, and it is reset to the initial value if no fault is detected during 2 x Z.

For details on exponential back-off algorithms see also section Reliability Concepts Introduced By RSVP.

2.2.1.3. Fault Notification

Each node having detected a failure sends a fault indication signal (FIS) throughout the network. For OSPF this is a new LSA which is stored by the receiver NE and flooded to the next neighbors.

Aspects of the LSA flooding:

- reliability
  LSA sent to a neighbor must be acknowledged, otherwise it is retransmitted.
- two-ways connectivity check
  When a link fails the related routers will report the loss of adjacency by a new LSA flooded into the network. For any network element, one received LSA is sufficient to consider the link down. In order to consider an link "up" both related LSA are necessary.
- triggers and frequency
  New LSAs are originated when: local connectivity changes, local IP prefix change, refresh according to a specific timer, configuration changes (e.g. link metric changes).

2.2.1.4. Recovery Operation

The receiver of the new LSA must compute a new routing table according to the new information. Usually the Dykstra algorithm is used to compute the shortest path between two nodes taking into account the link metrics.

New extensions to the ISIS [52] allow to assign multiple metrics to each link. This supports the concept of multi topology routing in which multiple topologies can be derived from a single physical network.

Notice that this may be used to distinguish the data plane from control plane, which are in general different. Therefore failures (and recovery) of the control plane do not necessarily impact the traffic of the data plane. See next section.

2.2.1.5. Traffic Recovery

Separation of Control and Data Plane allows the implementation of so called NonStop Forwarding (NSF) procedures. This procedure is also known as graceful restart: a router which detects the failure of its routing processor (RP) still continues to forward traffic based on the last state of its routing table while a stand by RP takes control. The restarting period is the resynchronization time of the control plane.

The backup RP sends a notification to each of its neighbors indicating the begin of the restarting mode. OSPF uses an opaque grace LSA with a local link flooding scope. For the duration of the grace period the neighbors:

- continue to advertise the restarting router in its LSA and
- keep forwarding traffic assuming that control plane failure is not affecting the data plane. This opposed to common IGP strategy, which computes path around the failed node (assuming data plane failures, power supply failure).

NSF may be used in the:

- EDGE node e.g. when there is a single link from the customer premises edge (CPE) to the edge router.
- core node, if there is a redundant router and alternate path are not usable for extra traffic.

IGP timer have to be short for fast convergence but should be larger than the restarting NSF procedure. If this condition is violated a so called false-positive condition is resulting. However this is not critical in general and can be solved by IP routing.

Notice that this idea of NSF is refined in the RSVP self-refreshment procedure.

2.2.2. **Non Time Aspects**

2.2.2.1. Traffic load balancing

In case there are N equal cost path computed by the routing algorithm there are two modes of operations for load balancing.

Per-packet load balancing: = packets are distributed among the N paths in a round-robin fashion. Negative effect: reordering of the microflows is necessary at the receiver.

Per-destination (per session) load balancing: = packets belonging to the same flow (session) always follow the same path. Positive effect: No reordering needed. Negative effect: Since the same hashing mechanism on the IP address fields is performed on every node, a kind of polarization effect for a certain path is introduced.

2.2.2.2. Bandwidth Efficiency

Regular IP routing does not deal with QoS, so the only objective may be traffic load balancing (see before). Some kind of QoS objectives during failure conditions may be achieved by IGP (OSPF, ISIS) metric manipulations e.g. in order to minimize the maximum link utilization.

Traffic Engineering (TE) Extensions may help for efficient network utilization will be discussed in the next chapter.

2.2.2.3. Scalability

For a link failure the number of flooded LSA in a full meshed network is O(n2): LSA is sent to the n neighbors which in turn sent it to their n neighbors. For a node failure the number of flooded LSAs is O(n3) since the node failure affects every of the related n links.

2.2.2.4. QoS during failure

In order to be able to handle large amounts of LSA without significant delay, potentially competing with regular data packets some QoS mechanism are in place:

1. Packet marking: each differentiated services (DS) header field of an IP packet is marked at the origination with a particular DiffServ code point (DSCP) value (also known as "coloring" packets).
2. Packet scheduling: A router can use the "color" in order to provide the appropriate QoS treatment to the packet when sending it out:

- Queue the packet based on its color.
- Use congestion avoidance mechanism like random early detection (RED). RED performs selective packet discard upon queue congestion. Because TCP react to packet loss by reducing the sending rate it is desirable to avoid corresponding oscillations effects. Therefore RED uses probabilistic dropping of packets already when a certain queue size threshold exceeds. When the average queue size exceeds a maximum all packets are dropped.

Notice that packet marking applies only to OSPF, because IS-IS uses directly connectionless network services. For IS-IS on congestion selective dropping of routing messages at the incoming site may be applied instead.

Notice that the congestion avoidance for the IGP packets is a good example of cooperation between two layer (IGP and TCP) for the purpose of increased network reliability. The 1+1 packet protection (see next chapter) will continue this idea of multilayer cooperation.

2.3. MPLS Traffic Engineering

This section handles both the MPLS layer, also so-called 2 ½ layer, which acts on the control plane (CP) and being responsible for routing decisions using layer 3 (see previous chapter) as well as RSVP which acts on layer 4.

The main functionality of the MPLS TE is:

1. Configuration of TE LSP on head-end LSR.
   Attributes include: address of tail-end LSR, required bandwidth, required protection/restoration, affinities.
2. Topology and resource information distribution
   ... which is needed for path computation at the head-end. This information is distributed by each node via link state routing protocol (OSPF or IS-IS) with TE extensions that reflects link characteristics and reservation states (including bandwidth availability).
3. TE LSP computation
   Every LSR uses information available in its topology and resource database to compute a constraint shortest path (via CSPF algorithms) according to a set of requirements for the LSP to be setup.
4. TE LSP setup
   The head-end LSR signals the LSP by means of RSVP-TE signaling protocol. LSP are maintained (refreshed) and torn down if not needed anymore.
5. Packet forwarding
   Once the LSP is set up, the head-end starts forwarding packets along the LSP. This packets are marked with a label so intermediate routers do not need to make any routing decision (label switching).

For further details compare [44], [51].

Motivation for deploying MPLS Traffic Engineering:

1. Better network resources utilization: bandwidth optimization, traffic load balance: MPLS TE overcomes the classic "fish problem" which may occur in IP networks (congestion on some parts of the network, while others still offer spare capacities) because routing decisions are made not only based on the final destination IP address and fix link metrics, but also on additional constraints: bandwidth, affinities, further routing constrains.
2. Strict QoS guarantees: In case of networks with a single class of service (CoS), MPLS TE allows the operator to reduce the average and maximum link utilization. Hence the probability of delays caused by traffic queuing is reduced, resulting in better QoS. In case of networks with a multiple class of services various mechanisms like marking, queuing and congestion avoidance in the data plane must be in place. MPLS TE allows the control of the proportion of high – medium - and low-priority traffic. Strict QoS guarantees are necessary in particular for sensitive traffic flows: voice, video, circuit emulation (e.g. VPN).

3. Fast recovery: MPLS TE supports different recovery mechanisms with fast convergence along with additional guarantees like QoS protection during recovery (see next section).

For further details compare [44] and [54].

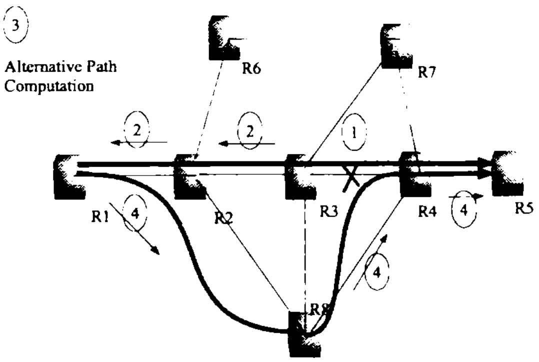
2.3.1. **Mode of Operation & Protocol Extensions**

Three basic mechanisms can be used as recovery mechanism: Global Default Restoration, Global Protection and Local Protection. In addition the 1+1 Packet Protection is to be considered.

2.3.1.1. MPLS TE Global Default Restoration

The MPLS TE Global Default Restoration is usually also referred to as "path restoration".
Principle: Failures are notified to the head-end by means of RSVP (explicitly) or by routing protocol updates (loss of routing adjacencies). The head-end LSR (ingress node) recomputes (global recovery) and re-signals the LSP along an alternate path.

Fig. 5 Path Restoration



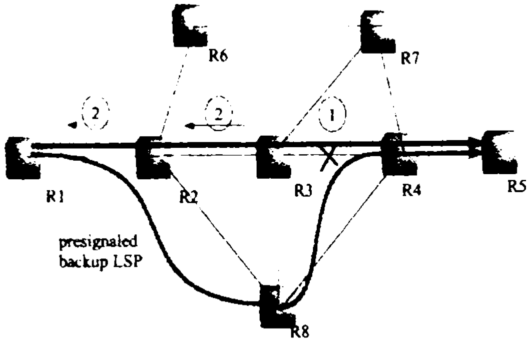
Following steps illustrate the Path Restoration procedure (compare numbers in the figure above):

- Step1: Node R3 detects the link failure to the node R4.
- Step2: The link failure is indicated to the ingress node. This may be either by means of RSVP signaling (Path Error) or by means of the routing protocol (IGP).
- Step3: Ingress node computes an alternative LSP, which of course will not use the failed link R3-R4.
- Step4: The new LSP is RSVP signaled towards R5 (Path). This LSP may be used as soon as R1 receives the corresponding Resv back from R5.

2.3.1.2. MPLS TE global protection

Usually also referred to as path protection.

Fig. 6 Path Protection



In principle this is a global 1:1 protection recovery mechanism. The failures are notified to the ingress node by means of RSVP (or by the IGP routing protocol). The ingress node switches to the already computed and signaled backup LSP.

Following steps illustrate the Path Protection procedure (compare numbers in the figure above):

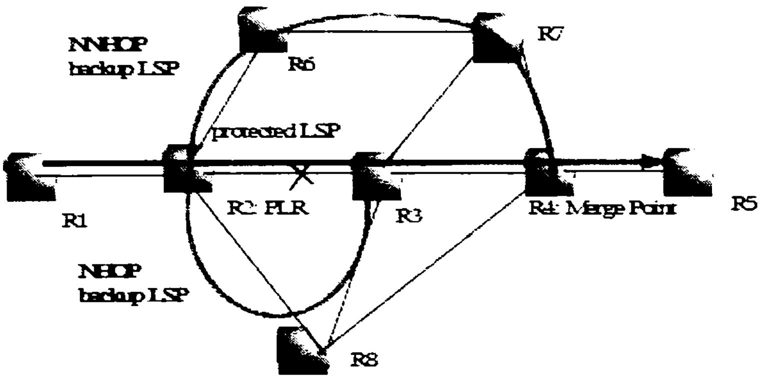
- Step1: Node R3 detects the link failure to the node R4.
- Step2: The link failure is indicated to the ingress node. This may be either by means of RSVP signaling (Path Error) or by means of the routing protocol (IGP).
- Step3: Ingress node switches to the pre-signaled backup LSP, which of course does not use the failed link R3-R4.

2.3.1.3. MPLS TE local protection

The MPLS TE local protection is usually also referred to as Fast Reroute.

Principle: This is a local protection recovery mechanism. The failure is notified to the node immediately upstream to the failure (point of local repair, PLR). All affected LSP are locally rerouted away from the failure on a backup tunnel. The protected LSP is signaled with a specific attribute set in the Path message which indicates it as "fast rerouteble". The backup tunnel is pre-allocated and signaled before failure.

Fig. 7 Local Protection



The backup tunnel may terminate:

- on the PLR next hop (NHOP backup tunnel). In this case it can protect only against a link failure (link R2-R3).
- on the neighbor of the PLR's neighbor (NNHOP backup tunnel). In this case it can protect also against a node failure (R3).

Two different techniques may be applied:

- facility backup (bypass): = a single backup LSP is used to protect all the fast-reroutable LSPs from a link or node failure. If the bandwidth offered by the LSP is not sufficient additional LSP may be used. The mechanism uses the MPLS stacking property.
- one-to-one backup (detour) creates a separate backup LSP for each protected TE LSP at each hop.

Merging rules can reduce the number of detour LSP.

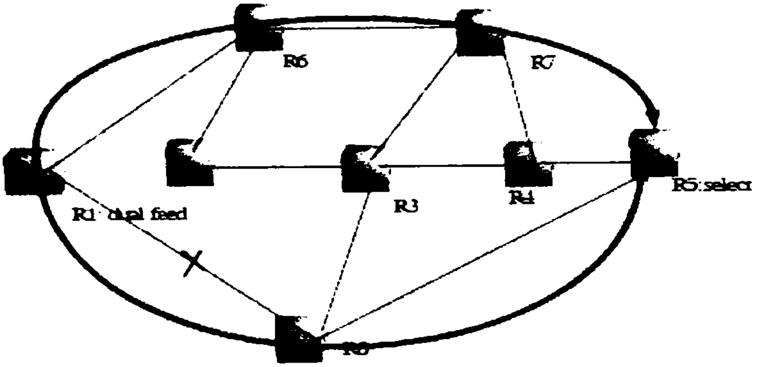
2.3.1.4. 1+1 Packet Protection

Principle: Permanent bridge of traffic over two diversely routed TE LSPs. At the ingress node packages are dual fed. At the egress node the select operation takes the first arrived packet and drops the copy package. This is a single-ended protocol since the decision is taken by the tail end without any signaling exchange.

There are two different approaches:

1. switch and switch back is explicitly triggered by the failure.
2. With an additional sequence number in the shim header no trigger necessary since there is an implicit switch on the tail LSP (see ITU-7712).

Fig. 8 1+1 Packet Protection



Although this mechanism is simple and efficient in terms of recovery times, it has some major drawbacks (see Comparison of Recovery Alternatives). Therefore according to [12] this mechanism has never been implemented.

2.3.2. **Recovery Cycle**

2.3.2.1. Fault Detection

Faults may be detected by updates of the IGP routing database (see previous chapter) These updates suffers in general some delay because network propagation time, lower frequency of the related IS-IS timer, additional hold-off time.

An additional mean to detect faults is via the "RSVP hello protocol extension " mechanism introduced by RFC 3209. RSVP Hellos are exchanged periodically between the neighbors on a LSP. If no RSVP Hellos have been received during a certain configurable time period, the RSVP Hello adjacency is considered down.

The RSVP Hello monitors a LSP, however for scalability reasons only one RSVP Hello adjacency should be activated per set of LSP traversing the same interface. Unnumbered interfaces towards a certain peer node should share the same RSVP Hello adjacency. It is also recommended for numbered interfaces towards a certain peer node that they should share the same RSVP Hello adjacency. However in this case it is necessary to pay attention to the different identifications of the node.

RFC3473 recommends separation of control channels from the actual data channels. A RSVP Hello failure may indicate a failure limited to the CP. If this failure is not confirmed as a failure of the Data Plane (DP) (e.g. by a corresponding SDH link failure) it is not necessary to switch the data flow. Instead recovery of the CP is sufficient. If CP and DP are using the same physical link, RSVP Hello failures without DP failure are rare in general, however possible e.g. caused by an overloaded CP.

Distinction of node vs. link failure is essential for local protection in order to choose between NHOP backup tunnel vs. NNHOP tunnel. For this purpose a control Hello sent over the backup tunnel is proposed. The distinction node vs. link failure is provided by an extension of the RSVP Hello mechanism (see [37]).

In general any of the nodes traversed by a LSP may fail. For the global path restoration and protection as well as for the local protection the failure must be detect by the upstream node. For the 1+1 Packet Protection failure detection is by nature only on the tail end.

2.3.2.2. Hold-Off Time

The general rule for the configuration of the hold-off timer is to avoid racing condition between the recovery mechanisms on different layers.
Therefore in most cases a bottom-up timer-based approach is adopted: The upper layer waits a certain amount of time to give the lower layer a chance to recover - before starting it's own recovery. For instance following layer synchronization could be recommended:

- MPLS waits for the optical layer, which recovers (in general) faster.
- MPLS recovers itself before ISIS convergence is reached.
- TCP and its application should usually wait for ISIS recovery.

Notice that in general RSVP does not "wait" for ISIS to recover since the fault detection of ISIS is relatively slow (see previous chapter). However in the case of 1+1 packet protection (see next presentation) recalculation of the protecting LSP can be triggered by an update of the routing TE DB.

In general a multilayer recovery approach can offer appropriate means for the synchronization between individual layer. However such approaches are currently under research investigation.

2.3.2.3. Fault Notification

The FIS (Fault Indication Signal) can be propagated upstream to a node capable to reroute as:

- RSVP message: Path Error or Notification.
- IGP update. The loss of routing adjacency is usually detected and signaled less fast than the PathError.

The destination of the FIS may be the head end (in this case global recovery of the LSP) or an intermediate node (local recovery). Therefore propagation time for the local protection mechanisms is in general less than for global one.
In addition to the physical propagation delay (around 5 ms per 1000 Km fiber) also queuing and propagation delay caused by overload situations must be considered. Therefore it is indicated to prioritize the RSVP messages over application message. In particular RSVP messages should not be deliberately dropped - if it is possible to

distinguish. Because RSVP still could get lost, several RFCs have added means for a reliable messaging mechanism (see next chapter): Srefresh, message IDs with ACKnowledgements, exponential back-off procedures.

1+1 packet protection does not rely on a failure indication signal.

2.3.2.4. Recovery Operation

SPF calculation time is in general a function of network size and CPU power.

For CSPF, additional constraints must be considered. They may lead to a increased complexity (up to NP completeness) of the algorithms and related delays. CSPF calculation time is also a function of number of affected LSP since this operation must be done for each of them.

For the global path and 1+1 packet protection the CSPF calculation and the signaling time for the new path are not relevant since they are performed in advance. The disadvantage is that actual bandwidth distribution is not reflected.

Signaling of the new path is a function of the LSP length. In particular the operation time at each node must be considered.

The RFC 3473 contains a detailed description of the procedure to be performed by the restarting node and by its neighbors (see next section)

The OIF-E-NNI contains a detailed description to recover from control channel and control plane failures. (see [55], [4]).

2.3.3. **Non Time Aspects**

2.3.3.1. Scalability

The main indicator for the scalability is the number of primary and backup LSPs.

For global path protection, global path restoration and for the 1+1 packet protection the number of backup LSP is equal to the number primary LSP. In a network of N nodes the maximal number of primary LSP per node is N-1. The maximal number of LSP per network is N*(N-1). Intermediate nodes may be traversed in the worst case (e.g. star topology or minimal meshing, ring topology) by maximum 2*N*(N-1) LSPs.

For local protection if L links and N nodes with connectivity C are protected by facility backup, the number of backup LSP is: L + N*C*(C-1).

For the total number of backup LSP an additional factor K*S must be taken into account. It reflects the fact that a number of splits (S) may be necessary since one backup tunnel usually cannot offer the required bandwidth. The factor K reflects the number of recovery classes for the case of backup up LSP dedicated per class of recovery (e.g. sensitive traffic VoIP).

For local protection by one-to-one backup in a network with an average diameter D the total number of backup LSP is N*(N-1)*D. In addition a factor M reflecting the number of meshes (e.g. distinct meshes for voice and data) must be taken into account.

The scalability impacts:

- Memory consumption: Each LSP requires some memory (dynamic, NVM) mainly for the RSVP states, but also for related LSP attributes, in case they must be re-synchronized with the neighbor nodes during recovery.
- CPU load consumed for the state refresh. Is directly dependent on the number of LSP and on the load per LSP. In particular an intermediate node has to deal with a large number of traversing LSP (see above).
  In order to improve scalability per LSP several proposals have been included in order to reduce the processing load for the state refresh: Srefresh messages, aggregated acknowledge mechanism based on message Ids, dedicated liveliness mechanisms (Hellos) (see dedicated RSVP chapter RSVP Refresh Overhead Reduction Extensions RFC 2961).
- Recovery time Is directly dependent on the number of LSP to be recovered.
  Notice that also the neighbor of a restarting node may be overstressed by state re-synchronization during the recovery. For this purpose a pacing mechanism is included in the RFC3473 (see dedicated RSVP chapter GMPLS Signaling RSVP-TE Extensions RFC3473).

2.3.3.2. Bandwidth Efficiency

Bandwidth efficiency denotes the ability to share backup bandwidth depends on the recovery mechanisms.

In the context of global path protection it would be possible to share the bandwidth between backup path protecting independent resources. However this introduces a additional criteria, which is increasing even more the complexity of the CSPF calculation. In practice this would require to perform the path computation by an off-line tool. Distributed computing would require significant extensions in the CP (signaling and routing)

The complexity objection is also valid for the One-to-One backup local protection.

For the facility backup it is possible to achieve a certain degree of bandwidth sharing depending on the network topology (connectivity, elements to be protected). [12] indicates a value up to 5 for a single failure assumption.

Global path restoration does not reserve backup capacity in advance. In this context also RFC3209 [37] must be mentioned, which describes the make-before-break procedure used to avoid double bandwidth accounting when switching to the backup LSP on global path restoration.

1+1 packet protection is in principle not designed to share backup bandwidth since the packets transported over the trailing LSP must be instantly available.

2.3.4. **Comparison of Recovery Alternatives**

2.3.4.1. Default Global Restoration

Advantages

- Does not require additional configuration of the backup path. The corresponding TE LSP must not be established in advance. Instead it is computed (using a CSPF algorithm) and established very close to the time when needed.

Drawbacks

- The slowest recovery mechanism, because it implies following specifics:
- FIS propagation to the head-end node.
- Dynamic path computation. Depending on the complexity of the network and of the TE constraints (diversity, bandwidth utilization) this may be time consuming.
- TE LSP signaling after failure occurrence.
- Lack of predictability.

There is no guarantee that a TE LSP can be rerouted: Either because CSPF cannot always find a route (in this case some constraints may be relaxed for a new calculation run) or because LSP signaling may fail because an unstable network.

2.3.4.2. Global Path protection

Advantages

- The configuration effort depends on the number of LSP to be protected. If this number is limited, also the provisioning effort is limited. In particular for a network topology with many nodes and links this is the preferred solution versus the local protection solution.
- Predictability, because the protecting path is signaled in advance. The path is in general deterministic if centralized computed and provisioned.

Drawbacks

- Scalability impact since for each working path an additional protection path must be setup in advance, regardless if needed or not.
- Slow, because the failure notification must reach the head-end before switching the traffic.
- May require an off line tool for the centralized computation, in particular if bandwidth guarantee is required.

2.3.4.3. Local Protection

Advantages

- Fast recovery time since a local protection mechanism.
- Propagation and jitter delay does not change significantly.
- Relative good scalability for the facility backup method since the number of backup tunnels depends on the number of nodes to be protected, actually the number of next-next-hops.

Notice that this may not hold if this protection scheme apply to every node.

Drawbacks

- Requires configuration and setup of a potentially large number of LSP.
- Might be complex to troubleshoot
- Poor scalability for the one-to-one backup method since for each fast reroutable LSP a separate diversely routed LSP (detour LSP) terminating at the tail-end must be setup at each hop. Therefore the number of backup LSP is a function of the number of the LSP and of the network diameter.

2.3.4.4. 1+1 Packet Protection

Advantages

- very simple mechanism: single-ended protocol because switching decision is made by a single entity, the tail-end.
- very efficient since it does not require configuration effort. The calculation of working and protecting LSP is performed locally on the head-end node.
- Does not require neither signaling upon failure occurrence.
- The fastest, since it switches "instantly"[1] to the protecting packet.

Drawbacks

- Double bandwidth consumption
- The failure discovery at the tail-end may require some hardware changes.

In chapter 5 I'm elaborating on a solution which mitigates the mentioned drawbacks, taking the full advantage of the efficiency of this recovery technique.

[1] Actually there is some small latency corresponding to the delay of the trailing LSP.

2.4. **Reliability Concepts Introduced By RSVP**

In the following the reliability concepts (including recovery) and the related support as introduced by RSVP are described in detail.
The chapter is structured by the significant RFCs in order to allow a quick reference to the correspondent standards. This offers an inside view on the evolution of the concepts and may offer the explanation for historical grown terminology. Here an overview on the reliability concepts introduced by the following relevant RFCs:

- RFC 2205 introduces the Functional Specification of RSVP, the Protocol Mechanism and Reservation Model and Soft State as a fundamental concept with respect to network recovery
- RFC 2961 improves reliability by Srefresh and Bundle messages.
- RFC 3209 extends RSVP to be used as Signaling Protocol to establish LSP Tunnels. Such tunnels may be subject of Traffic Engineering (TE) over MPLS with beneficial implication for network recovery.
- RFC 3473 introduces some RSVP-TE additional mechanism: rapid failure notification, rapid convergence on state removal, fault handling.
- RFC 3474 proposes additional extensions to these signaling protocols to support the recovery of transport plane in an ASON network.

Some of the RFC statements are directly included inside quotation marks if they are subsequently commented or if it essential to be followed exactly.

2.4.1. **Functional Specification of Resource Reservation Protocol RFC 2205**

The RFC 2205 describes the RSVP Protocol Mechanisms with the fundamental RSVP messages and some key concepts like the *Soft State*.
The RSVP protocol is used by hosts and routers to establish and maintain a state in which it is able to provide the requested service. Therefore it is involved by definition in the recovery process.

For this purpose, RSVP introduces the "soft" state; That is, RSVP sends periodic refresh messages to maintain the state along the reserved path(s). In the absence of refresh messages, the state automatically times out and is deleted.

The purpose of the soft state is to support:

- graceful support for dynamic membership changes and
- automatic adaptation to routing changes.

Both may be necessary actions in the context of a recovery action.

Recovery from different failures, reliability and resilience aspects are not handled explicitly in this RFC. However the section 3.6 handles the so called "**Local Repair**" following route changes and the section 3.7 handles the "**Time Parameters**" related to the soft state. From the recovery perspective both are of interest in the context of the failure detection. Here some details.

2.4.1.1. Soft State

RSVP introduces the so called *soft state* approach in order to manage the reservation state in routers and hosts. The soft state is created and periodically refreshed by Path and Resv messages. States may be deleted by an explicit teardown message.

Two specific timeout are essential to this concept:

- At the expiration of each "refresh timeout" period and after a state change, RSVP scans its states to build and forward Path and Resv refresh messages to the neighbor hops.
- At the expiration of a so called "cleanup timeout" interval, the state is deleted if unused, means no matching refresh messages arrived before.

The state maintained by RSVP is dynamic:

- Route changes, will generate new Path (and Resv) message along the new routes. The 'old' states along the 'old' routes will timeout.
- On any other change (e.g. on QoS request) on the given routes, the host starts sending revised Path and/or Resv messages, which is resulting in an appropriate adjustment in the RSVP state in all nodes along the path.

2.4.1.2. Local Repair

Section 3.6 introduces the so called "*Local Repair*" as a consequence of route changes: "*When a route changes, the next Path or Resv refresh message will establish path or reservation state (respectively) along the new route. To provide fast adaptation to routing changes without the overhead of short refresh periods, the local routing protocol module can notify the RSVP process of route changes for particular destinations. The RSVP process should use this information to trigger a quick refresh of state for these destinations, using the new route.*

The specific rules are as follows:

- *When routing detects a change of the set of outgoing interfaces for destination G* [N.B. caused by routing changes], *RSVP should update the path state, wait for a short period W, and then send Path refreshes for all sessions G/* (i.e., for any session with destination G, regardless of destination port). The short wait period before sending Path refreshes is to allow the routing protocol to settle, and the value for W should be chosen accordingly. Currently W = 2 sec is suggested; however, this value should be configurable per interface.*
- *When a Path message arrives with a Previous Hop address that differs from the one stored in the path state, RSVP should send immediate Resv refreshes to that PHOP."*

This shows the limitations at this stage: RSVP must rely on the routing protocol since neither own RSVP Hello mechanism nor the concept of Label Switched Path (LSP) was not yet available. The time W suggested for the adaption to the new routes may be inadequate for time critical applications.

Notice that the behavior described before assumes implicitly the transport plane and control plane are identical. If they are different, more specific considerations are necessary. For instance the mechanism with the improved 1+1 Packet Protection proposed by me in chapter 4 offers a faster and more reliable solution.

2.4.1.3. Refresh Period and State Lifetime

According to this RFC, RSVP originally sent its messages as IP datagrams with no reliability enhancement. Therefore it was necessary to introduce periodic transmission of refresh messages by hosts and routers, which is expected to handle the occasional loss of an RSVP message. *"If the effective cleanup time out is set to K times the refresh time out period, then RSVP can tolerate K-1 successive RSVP packet losses without falsely deleting state."* In addition a general requirement is formulated: *"The network traffic control mechanism should be statically configured to grant some minimal bandwidth for RSVP messages to protect them from congestion losses."*, which is not easy to achieve in practice.

Section 3.7 recommends that if RSVP is experiencing noticeable packets losses when crossing a congested non-RSVP cloud, a larger value should be used for the timeout factor K. Actually this is valid as well for inside the RSVP cloud. In particular depending on the reliability of the link layer either the timeout itself or the timeout factor K could be adapted.

Two time parameters are defined to be relevant to each element of RSVP path or reservation state in a node:

- The *refresh period R* between generation of successive refreshes for the state by the neighbor node. This parameter is part of the TIME_VALUES object which is included in the Path/Resv message.
- The *local state's lifetime L* which uses the refresh period R to evaluate the stored states.

Following consideration are made in chapter 3.7 for the configuration of this parameter:

1. *"Floyd and Jacobson [57] have shown that periodic messages generated by independent network nodes can become synchronized. This can lead to disruption in network services as the periodic messages contend with other network traffic for link and forwarding resources. Since RSVP sends periodic refresh messages, it must avoid message synchronization and ensure that any synchronization that may occur is not stable. For this reason, the* ***refresh timer should be randomly set*** *to a value in the range [0.5R, 1.5R]."*

   This recommendation addresses the burst of messages flooded over the network. However it is not of significant relevance with the summary refresh messages introduced by RFC2961

2. *"To avoid premature loss of state, L must satisfy* ***L >= (K + 0.5)*1.5*R****, where K is a small integer. Then in the worst case, K-1 successive messages may be lost without state being deleted. To compute a lifetime L for a collection of state with different R values R0, R1, .., replace R by max(Ri). Currently K = 3 is suggested as the default. However, it may be necessary to set a larger K value for hops with high loss rate. K may be set either by manual configuration per interface, or by some adaptive technique that has not yet been specified."*

   Such adaptive techniques may be offered in the context of my proposal for the "Improved 1+1 Packet Protection" (see chapter 4).

3. The refresh time R is carried as part of the ***TIME_VALUES*** **object** in each Path and Resv message. This allows the receiving node to compute the *lifetime* L of the corresponding RSVP states.

4. The default value of the refresh time ***Rdef*** should be configurable per interface. The suggested default value is 30 sec. However *"The refresh time R is chosen locally by each node. If the node does not implement local repair of reservations disrupted by route changes, a smaller R speeds up adaptation to routing changes, while increasing the RSVP overhead. With local repair, a router can be more relaxed about R since the periodic refresh becomes only a backstop robustness mechanism. A node may therefore adjust the effective R dynamically to control the amount of overhead due to refresh messages."*

   A *relaxed* R can be considered also when instead of the local repair for instance the [Improved] 1+1 Packet Protection is available.

5. If the refresh time is changed dynamically (in order to reduce the overhead on congestion), then its increasing rate should be limited: *"Specifically, the ratio of two successive values R2/R1 must not exceed 1 +* ***Slew.Max****. Currently, Slew.Max is 0.30. With K = 3, one packet may be lost without state timeout while R is increasing 30 percent per refresh cycle."*

6. However *"a node may temporarily send refreshes more often than R after a state change (including initial state establishment)."*
   This may improve the robustness for the price of an increased amount of messages. This dilemma is addressed by the RFC2961 (see next)

7. *"The values of Rdef, K, and Slew.Max used in an implementation should be easily modifiable per interface, as experience may lead to different values. The possibility of dynamically adapting K and/or Slew.Max in response to measured loss rates is for future study."*
   Such adaptive techniques may be offered in the context of my proposal for the "Improved 1+1 Packet Protection" (see chapter 4).

2.4.2. RSVP Refresh Overhead Reduction Extensions RFC 2961

RFC 2961 addresses following two aspects which are essential for the network recovery:

- **Reliability**: Standard RSVP [35] recovers from a lost message via RSVP refresh messages. The refresh intervals of the nodes experiencing the lost should be short in order to increase reliability by short fault detection times.
- **Scaling**: The resource requirements increase proportionally with the number of sessions. Each session requires the generation, transmission, reception and processing of RSVP Path and Resv messages per refresh period. A large number of sessions requires a large number of resources. In order to reduce the resource utilization refresh intervals should be increased. This way also overload situation as they typically happens in the network recovery context may be reduced.

The time requirements for the refresh intervals are divergent. RFC 2961 solves this contradiction by describing a number of mechanisms that can be used to:

- support reliable RSVP message delivery on a per hop basis ...
- ... also offering an improved scaling performance.

The related key concepts introduced are the following:

1. **Distinction between trigger and refresh messages**

   Trigger messages advertise state or other information which was not previously transmitted e.g. new states, new routes. Trigger messages use ACK_(acknowledge) desired flag for improving reliability. Refresh messages (Path, Resv) represent previously advertised states and contain exactly the same objects and same information as a previously advertised and are sent over the same path. Refresh messages do not use a (positive) acknowledge mechanism.

2. **Summary Refresh Message**

   The Summary refresh message enables refreshing state without the transmission of whole refresh messages, while maintaining RSVP's ability to indicate when state is lost and to adjust to changes in routing. All matching states are updated as if normal RSVP trigger messages were received. If matching states are not found then the sender is notified with a NACK (not acknowledged). A flag in the common RSVP header indicates the Refresh (overhead) reduction capability.

3. **Exponential Back-Off Procedures**
   Messages awaiting acknowledgment should be retransmitted according to the following parameter:

   - **Rapid retransmission interval** *Rf,* initial retransmission interval, for unacknowledged messages. If first sent message is not acknowledged, the sending node will schedule a retransmission after *Rf* seconds. The value of *Rf* could be in the range of the round trip time (RTT) between a sending and a receiving node.
   - **Rapid retry limit** *Rl,* maximum number of times a message will be transmitted without being acknowledged.
   - **Increment value** *Delta*; The ratio of two successive retransmission intervals is (1 + Delta).

The successive retransmit timeout *Rk* is initialized to *Rf* and in increased according to the formula below, until the limit *Rl* is reached.

$$Rk = Rk * (1 + Delta);$$

This way the amount of exchanged messages is reduced avoiding escalation of overload as it typically occurs in network recovery scenarios.

In addition some new objects are introduced in order to increase reliability and to reduce message load:

- MESSAGE_ID: The Message Identifier, uniquely identifies a message in the context of the generator's IP address.
- MESSAGE_ID_ACK and MESSAGE_ID_NACK objects are used for fast detection of message loss. They may be sent piggy-backed in unrelated RSVP messages. They may be grouped as a list in a single ACKnowledge message.
- EPOCH, indicates that message identifier sequence was reset, e.g. when a node reboots. This is necessary for the receiving node in order to distinguish between lost (out of sequence) messages and a restart situation of the sending node.

2.4.3. **RSVP-TE: Extensions to RSVP for LSP Tunnels RFC 3209**

This RFC proposes several additional objects that extend RSVP allowing:

- Establishment of explicitly routed label switched paths using RSVP as a signaling protocol. The result is the instantiation of label-switched path (LSP) - also called tunnel, which can be automatically routed away from network failures, congestion, and bottlenecks ("smooth rerouting" of LSPs).
- Preemption and loop detection.
- Establishment of multiple parallel LSP tunnels between the head and tail end node, so that the traffic between these nodes could be mapped onto these LSP tunnels according to local policy.
- Enhanced management and diagnostics of LSP tunnels.
- Rapid node failure detection via a new HELLO message.
- Explicit routing capability by incorporating a new *EXPLICIT_ROUTE* object into the RSVP Path messages. This object contains the concatenation of hops which constitutes the explicitly routed path. This path can be either predetermined by administrative specification or dynamically computed, taking into consideration QoS, traffic engineering, policy requirements and the current network state. Explicit routing can be used to optimize the utilization of network resources and enhance traffic oriented performance characteristics increasing this way the network reliability.

The signaling protocol model supports in particular the specification of an explicit path as a sequence of strict and loose routes possibly combined with abstract nodes (defined as a group of nodes which do not expose their topology).

Resource allocation (e.g. bandwidth) on LSP tunnels established via RSVP is usually intended but not mandatory. Such LSPs without resource reservations can be used, to carry best effort traffic and/or to implement different fall-back and recovery policies under fault conditions: dedicated and shared backup capacity, dedicated protection with and without extra traffic, pre-planned and dynamic recovery path.

Following concepts supporting the reliability are introduced:

2.4.3.1. LSP Tunnel

An LSP (Label Switched Path) which is used to tunnel below normal IP routing and/or filtering mechanisms. "Tunnel" reflects the fact that the traffic through it is opaque to intermediate nodes along the label switched path.

Following capabilities are supported with the extensions proposed by this RFC:

1. **Establish LSP tunnels with or without QoS requirements**
   The ingress node sends the RSVP Path message with a session type of *LSP_TUNNEL_IPv4 and* with a *LABEL_REQUEST* object, which is indicating that a label binding for this path is requested. If the ingress node has information about routes which are satisfying the resource requirements and policy criteria, it may include the *EXPLICIT_ROUTE* object, as a sequence of abstract nodes.

2. **Dynamically reroute an established LSP tunnel.**
   "If, after a session has been successfully established, the sender node discovers a better route, the sender can dynamically reroute the session by simply changing the EXPLICIT_ROUTE object." However in practice some complementary actions are necessary, like teardown of the previous LSP tunnel.

3. **Observe the actual route traversed by an established LSP tunnel.**
   This may be accomplished by adding a *RECORD_ROUTE (RRO)* object to the Path and Resv message, so that the ingress node can receive information about the actual route that the LSP tunnel traverses. The *RRO* collects path information hop-by-hop. The collected information may also be used for loop detection. *"The sender node can also use this object to request notification from the network concerning changes to the original routing path."*

4. **Identify and diagnose LSP tunnels.**
   For this purpose a *SESSION_ATTRIBUTE* object can be added to Path messages. It contains additional control information, such as setup and hold priorities, resource affinities, and local-protection. "*For instance, in the traffic engineering application, it is very useful to use the Path message as a means of verifying that bandwidth exists at a particular priority along an entire path before preempting any lower priority reservations."*

5. **Preempt an established LSP tunnel under administrative policy control.** Preemption becomes necessary when there are not sufficient resources available. Preemption may be controlled via the *setup and hold priorities* along with *SENDER_TSPEC* and *POLICY_DATA* objects contained in Path messages. *"If a Path message is allowed to progress when there are insufficient resources, then there is a danger that lower priority reservations downstream of this point will unnecessarily be preempted in a futile attempt to service this request."*

6. **Perform downstream-on-demand label allocation, distribution, and binding.**
   This is performed by the *LABEL_REQUEST* object requesting the intermediate and receiver nodes to provide a label binding for the session. *As* a response the egress node (and the following intermediate node) includes the *LABEL* object in the RSVP Resv message, which is sent back upstream towards the sender, following the path state created by the Path message, in reverse order. The received label is used for the outgoing traffic associated with the LSP tunnel. The label sent upstream is used to identify the incoming traffic associated with the LSP tunnel. The "Incoming Label Map" (ILM), which is used to map incoming labeled packets to a "Next Hop Label Forwarding Entry" (NHLFE) is accordingly updated.

2.4.3.2. Abstract Node

An *Abstract Node* is a group of nodes whose internal topology is opaque to the ingress node of the LSP. The abstract node concept improves scalability. An abstract node is said to be simple if it contains only one physical node. Local Repair activities are in general limited inside the abstract node remaining hidden for the outside network. An abstract node may stay for a whole E-NNI domain [4]. Recovery activities inside the domain may remain hidden for the adjacent E-NNI domains.

2.4.3.3. Explicitly Routed LSP

The explicitly routed LSP is a LSP whose path is established by other means than conventional IP routing. This allows the establishment of any of the different types of protection and restoration path. In particular following procedures are supported:

1. **Make Before Break**: This is a concept of smooth, adapting rerouting traffic by establishing first a new LSP tunnel and transferring traffic from the old LSP tunnel onto the new LSP tunnel before tearing down the old LSP tunnel. To support make-before-break in a smooth fashion, it is necessary that resources used by the old LSP tunnel should not be released before traffic is transitioned to the new LSP tunnel. On the other site reservations on common links should not be counted twice because this might cause Admission Control to reject the new LSP tunnel.
   A smooth transitions in routing and bandwidth may be achieved by the combination of the LSP_TUNNEL SESSION object (which narrows the scope of the RSVP session to the included TE Tunnel ID) and the SE (shared explicit) reservation style. During the reroute (or bandwidth-increase operation - see next), the tunnel ingress needs to appear as two different senders to the RSVP

session. This is achieved by including a distinct "LSP ID" inside the SENDER_TEMPLATE and FILTER_SPEC objects. To initiate rerouting, the ingress node sends a new Path message including the original SESSION object with the old Tunnel_ID and with its IPv4 address in the Extended_Tunnel_ID but with a new LSP ID, a new SENDER_TEMPLATE and a new ERO along the new path.
The egress node responds with a Resv message with an SE flow descriptor formatted as:

<FLOWSPEC>
<old_FILTER_SPEC><old_LABEL_OBJECT>
<new_FILTER_SPEC><new_LABEL_OBJECT>

When the ingress node receives the Resv Message(s), it may begin using the new route. It SHOULD send a PathTear message for the old route.
The make-before-break procedure applies for global (path) restoration - compare chapter 2.3.

2. **Bandwidth Increasing Procedure**
This procedure describes how to setup a tunnel that is capable of maintaining resource reservations while it is attempting to increase its bandwidth. The problem is again to avoid double counting of resources along links which are in common for the old and new LSP tunnels, since the new Path message indicates the final (increased) bandwidth. The solution is to have indicate in the new Path message the fact that only a delta between the new and old bandwidth is needed. This is achieved by changing the SENDER_TEMPLATE and the FILTER_SPEC by the inclusion of the new "LSP ID", while the LSP_TUNNEL SESSION object remains the same.

3. **Establishment of LSP pairs** (working and protecting) to support global (path) protection or 1+1 packet protection mechanism. Compare chapter 2.3

2.4.3.4. Hello Extension

The RSVP Hello extension is introduced to allow RSVP nodes to detect when a neighboring node is not reachable. In principle such a situation is handled the same as a link layer communication failure. *"This mechanism is intended to be used when notification of link layer failures is not available and unnumbered links are not used, or when the failure detection mechanisms provided by the link layer are not sufficient for timely node failure detection. It should be noted that node failure detection is not the same as a link failure detection mechanism, particularly in the case of multiple parallel unnumbered links".* It should be mentioned that in general the failures on the link layer may detected in general more faster by means of the lower layer. Ideally failure detected by the Hello extension should be correlated with link layer communication failure.

The Hello extension consists of a Hello message, which may include either a HELLO REQUEST object or a HELLO ACK object. Each neighbor can autonomously issue a "HELLO REQUEST". Each request must be answered in time by an acknowledgment. Otherwise a failure is assumed. Failure detection intervals may be configured independently - possibly different - on either site.

The RSVP Hello Extensions enables RSVP nodes to detect and to distinguish between node failure and failures of the control channel (link) by the following mechanism: Hello messages includes a Source Instance and a Destination Instance object. If the sender restarts (reboots) then the value of the Source Instance must be different then the previous one. The Destination Instance contains the most recently received Source Instance as received from the neighbor. If nothing ever received, then it is set to 0. Neighbor failure detection is accomplished by collecting and storing a neighbor's "instance" value. If a change in value is seen or if the neighbor is not properly reporting the locally advertised value, then a reset of the neighbor is assumed.

If the links between the nodes are unnumbered, then the Hellos are exchanged between the nodes, on an arbitrary link - not on every link -, as a matter of routing. Therefore it is recommended to use in addition specific link failure mechanism instead of this Hello extension. This allows to differentiate between the necessity of resynchronization between the nodes vs. link repair.
If the links are numbered, then the Hellos must be exchanged on each of the numbered links.

2.4.4. GMPLS Signaling RSVP-TE Extensions RFC3473

This document introduces following recovery related features:

- rapid failure notification
- control channel separation
- restart capability

2.4.4.1. Notify Message

The Notify message informs non-adjacent nodes about LSP related events. It provides a generalized notification mechanism.

Notify messages are generated on demand, means only after a Notify Request object has been received. They differ from the previously defined error messages (i.e., PathErr and ResvErr messages) in that they are "targeted" to a node other than the immediate upstream or downstream neighbor.

The reliable delivery of the Notify message is achieved by using an Ack Message [36] to acknowledge the receipt of a Notify Message.

The notify message is the preferred mean for signaling fault notification because its direct (opposed to the hop-by-hop) addressing. This may be beneficial for instance in the context of path restoration to reduce the fault notification time. In particular they are suitable to be used to to implement the ForwardAck and the PartialAck messages (which are described in chapter 3).

2.4.4.2. Control Channel Separation

A control channel is said to be *separate* if it is not in-band with a transport channel or more general: there is not a one-to-one association of a control channel to a transport channel. This RFC provides following protocol specific objects and procedures to support the separation of the control channels:

First of all the transport channel has to be identified. The RSVP_HOP object which was already specified by RFC2205 must be refined by the introduction of the new IF_ID RSVP_HOP subobject. This object used by the sender of the Path message by setting the outgoing interface according to its choice. Of course this choice must be consistent with the corresponding ERO. This object is mandatory for unidirectional connections.
For bidirectional connections the sender may decide the interfaces for both directions, normally it is a common downstream and upstream data channel. The new IF_ID RSVP_HOP subobject is also used in the Resv message to indicate the actual downstream node's usage of that interface.

Notice that the separation of control and data channel has an significant impact on the reliability and recovery considerations since failures of control channel does not necessarily imply failure of the data channel and vice-versa. The separation of the control channels are essential for differentiated fault handling procedure (see next section).

2.4.4.3. Fault Handling

Following two types of control communication faults may be distinguished:

- **Nodal faults**, relates to the case where a node losses its control state (e.g., after a restart) but does not loose its data forwarding state, means the transport plane is not affected.
- **Control channel faults**, relates to the case where control communication is lost between two nodes.

The handling of both fault types is supported by the ***Restart_Cap*** object defined below and requires the use of Hello messages. The Restart_Cap object MUST NOT be sent when there is no mechanism to detect data channel failures independent of control channel failures.

Notice that the fault handling as suggested in this RFC relies on the separation of the Control Plane and Data (Transport) Plane which implies at least the separation of the control channel (see previous section).

The Restart_Cap Object is carried in Hello messages. It contains:

- **Restart Time**: "*SHOULD be set to the sum of the time it takes the sender of the object to restart its RSVP-TE component (to the point where it can exchange RSVP Hello with its neighbors) and the communication channel that is used for RSVP communication. A value of 0xffffffff indicates that the restart of the sender's control plane may occur over an indeterminate interval and that the operation of its data plane is unaffected by control plane failures.*" As mentioned before this "method" actually requires a strict separation of the control from the transport plane.

- **Recovery Time**: "*The period of time, in milliseconds, that the sender desires for the recipient to re-synchronize RSVP and MPLS* [N.B. transport] *forwarding state with the sender after the re-establishment of Hello synchronization. A value of zero (0) indicates that MPLS forwarding state was not preserved across a particular reboot.*"

These values should be set by the sending node to the specific values and recorded by the receiving node in order to be used by the subsequent *State Recovery,* see below.

The Hello Processing is modified to support State Recovery as follows: "*When a node determines that RSVP communication with a neighbor has been lost, and the node previously learned that the neighbor supports state recovery, the node SHOULD wait at least the amount of time indicated by the Restart Time indicated by the neighbor before invoking procedures related to communication loss. A node MAY wait a different amount of time based on local policy or configuration information.*"

During this waiting period, the node should behave as if it continues to receive periodic RSVP refresh messages from the neighbor*:*

- Preserve the RSVP states (of course also the transport states) for the LSPs established along the links with the neighbor node.
- Continue to send Hello messages with a Dst_Instance value set to zero (0), whereas the Src_Instance should be unchanged.
- Supress Refreshing of Resv and Path state

However, opposed to the regular: "*The node MAY inform upstream nodes of the communication loss via a PathErr and/or upstream Notify message with "Control Channel Degraded State" indication. If such notification has been sent, then upon restoration of the control channel the node MUST inform other nodes of the restoration via a PathErr and/or upstream Notify message with "Control Channel Active State" indication. (Specific error codes have been assigned by IANA.)*".

"*The node MAY clear RSVP and forwarding state for the LSPs that are in the process of being established when their refresh timers expire.*"

When a new Hello message is received from the neighbor, the node must determine based on the Src_Instance received, if the fault was limited to the control channel or was a nodal fault: *A* value different from the value that was received from the neighbor prior to the fault, indicates a restart. If the fault was limited to the control channel then the same Src_Instance is expected.

The different situation are handled as follows:

- **Control Channel Faults** "*In the case of control channel faults, the node SHOULD refresh all state shared with the neighbor. Summary Refreshes [36] with the ACK_Desired flag set SHOULD be used, if supported. Note that if a large number of messages are need, some pacing should be applied. All states SHOULD be refreshed within the Recovery time advertised by the neighbor.*"

- **Nodal Faults** Recovering from nodal faults uses one new object (Recovery Label having a format identical to a Generalized Label) and some other existing protocol messages and objects (see next section)

2.4.4.4. Procedures for the Restarting node

The first decision to be taken after a node restarted its control plane, depends on the check whether it was able to preserve its transport state. The recovery procedure relies mainly on the received Path message.

If the transport state was not preserved, then the node must send the Hello message with the Recovery Time set to 0. The recovery procedure is finished from the local perspective. However control states may be recovered from the neighbor of the restarting node (see next section)

If the transport state was preserved, the node must initiate the recovery of the control states. After the expiry of the Recovery Time (advertised in the Hello message - see before) all control states which are not resynchronized should be removed. Notice that this behavior is reconsidered by RFC3474 (see below) and overruled by the later OIF standards [4], which mandates an interaction with the management plane.

If the restarting node maintains its transport state on a per neighbor basis (dedicated, per interface labels are used on point-to-point interfaces) but it determines that a neighbor does not support state recovery then the Recovery Procedure with that neighbor is considered completed.

If the upstream neighbor node supports the recovery procedure, it will send specific (see below) Path messages during the Recovery Period. The restarting node first checks if it has an RSVP state associated. Notice that the condition to identify the Path state is not clearly specified: It may be the same message ID, or extended over all objects of the Path message). In the simplest case the Path state is found and refreshed. Otherwise, if the Path state is not found:

- If the message does not carry a Recovery Label object, the node treats this as a setup for a new LSP.
- If the message carries a Recovery Label object, the node searches its transport plane (table) for an entry whose is equal to the label carried in the Recovery_Label object.
    - If such an transport entry is not found, then the node treats this as a setup for a new LSP.
    - If such an entry is found in the transport table, the corresponding Path state is created, *"the entry is bound to the LSP associated with the message, and related forwarding state should be considered as valid and refreshed. Normal Path message processing should also be conducted. When sending the corresponding outgoing Path message the node SHOULD include a Suggested_Label object with a label value matching the outgoing label from the now restored forwarding entry. The outgoing interface SHOULD also be selected based on the forwarding entry. In the special case where a restarting node also has a restarting downstream neighbor, a Recovery_Label object should be used instead of a Suggested_Label object."*

- "*Additionally, for bidirectional LSPs, the node extracts the label from the UPSTREAM_LABEL object carried in the received Path message, and searches its MPLS forwarding table for an entry whose outgoing label is equal to the label carried in the object (in the case of link bundling, this may also involve first identifying the appropriate incoming component link).*"
  If such an entry is not found the Path message is considered as a setup for a new LSP. Otherwise (such an entry is found) "*the entry is bound to the LSP associated with the Path message, and the entry should be considered to be re-synchronized. In addition, if the node is not the tail-end of the LSP, the corresponding outgoing Path messages is sent with the incoming label from that entry carried in the UPSTREAM_LABEL object.*"

Notice that this way not only the incoming label is resynchronized for the unidirectional connection but also the outgoing label is resynchronized for the bidirectional connection.

Resv messages are processed normally during the recovery period except that when a forwarding entry is recovered, no new label or resource allocation is required. When the Resv message is not matching a Path state it should be silently discarded instead of generating a ResvErr (as during the normal operational processing).

2.4.4.5. Procedures for the Neighbor of a Restarting node

The following procedure applies for the neighbor node after reestablishing the communication with the restarting node which preserved its transport data (non-zero Recovery Time) within its Restart Time. Notice that setting a Restart Time value of 0xffffffff (which indicates an infinite Restart Time interval) may be the preferred option since the actual end of the neighbor's restart time is indicated by the reception of a corresponding Hello message.

- The upstream neighbor node must refresh all the Path states shared with the restarting node by including a Recovery_Label object with the label value received in the most recently corresponding Resv message. All Path messages must be sent within approximately 1/2 of the Recovery time advertised by the restarted neighbor. This is in order to allow the possibility of retries. For the purpose of pacing: "*If there are many LSP going through the restarting node, the neighbor node should avoid sending Path messages in a short time interval, as to avoid unnecessary stressing the restarting node's CPU. Instead, it should spread the messages across 1/2 the Recovery Time interval.*"
- The downstream neighbor node must refresh all the Resv states shared with the restarting node but only after corresponding Path message is received. In the mean time normal Resv and Summary Refresh messages should be suppressed.

2.4.5. GMPLS RSVP-TE Usage and Extensions for ASON-IETF RFC 3474

This RFC includes a subsection dedicated to the support of recovery from control plane (CP) failures in the scope of the ASON model.

2.4.5.1. Support For Behaviors during Control Plane Failures

The restart mechanisms as described in [38] is necessary to recover from control plane failures in the context of the GMPLS model. However, in the scope of the ASON model, additional procedures are necessary in order to support the following control plane behaviors (compare also [59]):

- The control plane should persistently store call and connection state information (see section below) for the following purposes:
    - Local recovery of the states of calls/connections from failure of the signaling controller.
    - verification of the neighbor calls/connections states. Notice that if the node maintains its state on a per neighbor basis and if during Hello synchronization it turns out that a neighbor does not support state recovery, the recovery procedure may be immediately considered completed.
- If the control plane node detects failure on all control channels between a pair of nodes, then it should request an external controller (e.g., the management system) for further instructions: e.g. remain in the *self-refresh mode (i.e., preservation)* for the local call/connection states or release local states for certain connections. Notice that this a recommendation for the behavior at the failure detection time. In general if self-refresh mode is entered (as a default), then there is a good chance to perform recovery in a distributed way without the interaction with the management system. An interaction with the management system is necessary to solve re-synchronization problems - see next.
- If the control plane node detects that one (or more) connections cannot be re-synchronized with its neighbor (e.g., due to different states for the call/connection) it should request an external controller (e.g., the management system) for further instructions. Notice that in general, if the situation is not clear, it is a good practice to maintain local connection state unless the management plane decides otherwise.
- If the control plane node (after recovering from node failure) loses information on forwarding adjacencies it should request an external controller (e.g. management plane) for information to recover the forwarding adjacency information.

Notice that in general the restart mechanism is designed to allow an automatic, non-centralized recovery. However this RFC explicitly mandates (SHOULD) for requests of the control plane towards an external controller (e.g. management system) for further instruction in case of the mentioned exceptions: failure of all control channels between a pair of nodes

- one or more connection cannot be re-synchronized with the neighbor after node recovery.
- forwarding adjacencies are lost after node recovery.

2.4.5.2. Supporting Objects

The following objects are relevant for the recovery aspect:

- *Soft Permanent Connection* (SPC) is a connection established by the management system - as opposed to the switched connection. The label association of the permanent ingress segment with the switched segment at the switched connection ingress node is a local policy matter. Support via SPC_LABEL (same format and structure as the EGRESS_LABEL)
- *Call* is a special purpose connection that requires a different subset of information to be carried by the messages. This information is processed by the call controller (as opposed to the connection controller) for the purpose of setting up a call/connection association. The call/connection separation is part of the call model (see [59]). Every call (during steady state) may have one (or more) associated connections.
  A special case is the zero connection call which may be used to :
  - indicate transient state during a break-before-make restoration event, or
  - setting up the user end-point relationship prior to connection setup, in particular to setup the protection path.

Support via the CALL_ID Object which may be part of the Path, Resv, PathTear, PathErr, and Notify message. It is optional for GMPLS but mandatory for ASON compliant networks. May be operator specific or globally unique.

- *Call Capability* is introduced and used to specify the capabilities supported for a call. It is implemented by the CALL_OPS. May be carried by the Path, Resv, PathTear, PathErr, and Notify messages. Contains two flags indicating:
  - call without connection, "call-only" call
  - synchronizing a call (for restart mechanism).
  - Notice that call synchronization implies the synchronization of all participating connections which increases the complexity of the recovery behavior.

2.5. Conclusion

Network resilience and the related categories: network recovery, reliability, survivability, continues to be a very interesting subject for the research community. Compare for instance the recent FP7 EU project ResumeNet ([58]). A recent publication (comp. [56], April 2010) gives as explanation the fact that today's internet cannot be considered as resilient, despite the perception that it operates apparently well. The main contribution of this paper is good overview on existing and emerging network resilience technologies.

For the **existing technologies** there is not too much systematical documentation available: Some basic literature covers the MPLS principles (comp. [44]). Vasseur offers a good overview on the Network Recovery [54]. Background theory on networks and their reliability of systems may be found in [61], [62].

For the **emerging technologies** there is a increasing number of IETF RFCs and drafts which is a good indicator for the actuality (compare some of the RFC4ddd and RFC5d1dd). However -typically for the RFCs- the available information is spread across different documents, which are referencing each other, in some cases, apparently even contradicting each other. This is particularly valid for the performance and reliability aspects.

The **classical performance engineering** of telecommunication and information systems is handled in [47]. Some chapters are dedicated to the network performance from the point of view of traffic overflow. Despite a short reference on "Alternative routing strategies" (section 6.11.3) and on "Network delay and routing" (section 7.6) there is no special consideration of the network recovery aspect. Chapter 8 is dedicated to the "Introduction to reliability", addressing the trade-off reliability-costs, different ways to increase reliability including redundant network components and improved maintenance procedures. However they address mainly the hardware part: e.g. redundant hardware units, inspection by maintenance technicians.

The method of "1+1 MPLS Packet Protection By Preventive Detection Of Quality Degradation" which I'm proposing in chapter 4 is addressing these principles on software by replicating only packets (with the same hardware basis) and by delegating the maintenance from the technician to the software. A further going basic idea is that of a fault tolerant software.

Modern telecommunication networks are introducing additional, specific aspects of reliability, see [6], [7], [8]. In particular it is required that the network as a whole should be able to guaranty a certain availability, serviceability. This may be offered by autonomous recovery actions in the network.

There is a lot of IEEE publications dedicated to tne subject network recovery and reliability. Here some examples:

A recent proposal [32] suggests the usage of a new recovery scheme called Multiple Routing Configurations (MRC) for a fast IP network recovery. MRC is based on keeping additional routing information which may have an impact on the scalability.

Another proposal to speed up recovery below the benchmark of 50 milliseconds is presented in [33]. This fast re-route technique is to be applied upon BGP peering link failures. The proposal is based on the precomputation of protection tunnels for each of the interdomain links. Overflows on the protection tunnel could be an issue since node or link failures tends to result into congestion situation.

Therefore a proposal was made to decouple path failure detection from the congestion control [34]. The solution is expected to improve the Stream Control Transmission Protocol (SCTP) failovers and may be interesting in the mobility context. Instead, active path monitoring using unreliable heartbeats is proposed. The solution is somehow related to my proposal for monitoring redundant LSPs for the purpose of early detecting quality degradation and may be tuned by the applications to meet their own requirements. There are also some pure theoretical considerations on survivability [25], [26], [27].

My contribution on the subject of network resilience concentrates on two aspects:

1. How to improve the performance of path setup. This direction is motivated by the performance penalties expected for network restoration as mentioned in the literature [12].

In this context I started with some lab studies experimenting with different signaling methods on real network elements [5].
The initial motivation was the fact that restoration schemes offers some advantages compared to protection schemes with dedicated capacity resources (e.g. reduced bandwidth needs, more flexible since signaling based), but have not been deployed at that time because their slow performance.
The outcome was that the control HW of the current generation of network elements is able to provide acceptable restoration performance. We found out that restoration time of 11ms and less can be achieved in a small network of 5NEs. From the scaling behavior it could be predicted that for larger networks the restoration time can be kept under the 50ms benchmark [48].

However fast and reliable traffic restoration is a permanent challenge, therefore it was indicated to think about additional means that could speed it up. One of the ideas is to improve the signaling by a optimized path setup algorithms, which is using the inherent existing parallel processing capacity. Recall that the path setup is a time critical activity necessary for the activation of a (possibly pre-calculated) connection restoration plan. This subject is covered in the next chapter 3 and resulted in an US patent application.
In the mean time (2008) a similar proposal was made under the title "A Fast and Efficient Segmented Signalling Protocol for GMPLS/WDM Optical Networks" [63]).
At that time only a few studies could be found on the distributed control protocols in the context of optical networks: Compare [70] for a survey of the various distributed signaling protocols. Examples in the WDM context are [64] and [65] which are proposing the forward reservation protocols (FRP) and the backward reservation protocol (BRP). FRP is reserving the wavelength on different candidate links during the forward traversing. This results in an over-reservation since only one wavelength is finally used. The other wavelength are released at the end of the reservation process, still they are not available in the mean time, increasing the reservation conflict.

This drawback is overcome with the backward reservation protocol, which collects the available wavelengths on the forward traversal. The destination node selects one from this set and makes the reservation on the backward traversal. The drawback of

the BRP is the possibly out-dated information whereas the drawback of FRP is the excessive reservation. Some variants suggested in [66] – [68] tries to mitigate these drawbacks by proposing different dropping/holding policies. [69] proposed that the intermediate nodes start the reservation before the connection request reaches the destination node. This solution does not solve the excessive reservation on the intermediate nodes, and it does not reduce the setup time since the reservations must be confirmed by the final node faster.

In order to solve this deficiency [63] proposed a novel segmented signaling (SSP) based on the concept of intermediate destinations. A special case of this protocols is called "parallel reservation protocol" (PRP), where the destination node sends the RESV_INFO message (with the actual wavelength reservation) to all the intermediate nodes along the route, whereas in the SSP the destination node sends the RESV_INFO message only to some specific intermediate node (ID), determined by dividing the route in smaller segments on which RSVP runs in parallel.

Here a summary of of the specific differences between these proposals:

- The [63] proposal applies on WDM Networks whereas my proposal may apply to any kind of GMPLS signaling (in particular WDM, TDM) which provides source routing.
- The [63] optimization addresses segments while my optimization addresses any intermediate nodes for the parallelization. Therefore there is a additional potential for efficiency gain for the setup time.
- The [63] suggests the modification of the RSVP protocol by defining some new control message types (RESV_INFO, RESV_SUCCESS, FAIL_INFO) while my implementation proposal is based on the existing control message.
- The [63] indicates parallelization only for the Resv message while my proposal may also include the Path message and the Confirmation message.
- In addition my proposal offers two different variants for sequential and for final synchronization.

The other basic idea was that the best resilience may be actually achieved by preventing recovery, therefore the next question/motivation:

2. How to a avoid expensive and time critical recovery actions by preventive measurements?

This subject is covered in the next chapter 4 and resulted in an US patent.

3. Method for fast source routed connection setup

3.1. Background

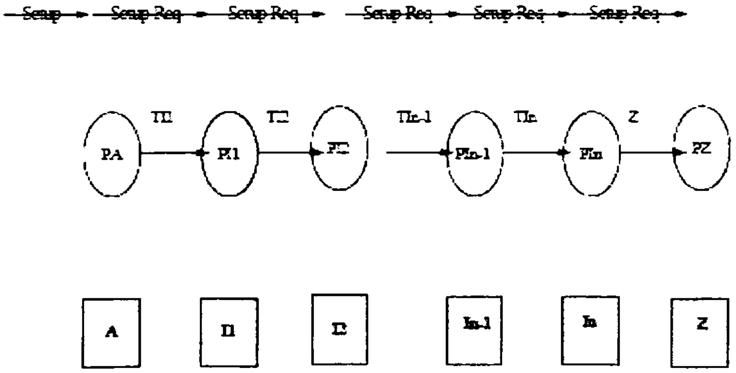
Automatically Switched Optical Networks (ASON) uses as a central concept the connection as a concatenation of links and subnetwork connections that allows the user to transport its data between an ingress and an egress node [1]. Thus a connection path through the network is a sequence of nodes starting with a begin node (ingress, usually denoted "A" node) and terminating at an end node (egress, usually denoted "Z" node). In general one or more intermediate nodes (in the following denoted "I" node) are traversed. Compare Fig. 9 Sequential Setup Request.

There are numerous methods for provisioning a connection path through a network, including manual path provisioning and automated path provisioning. ITU-T G.8080 describes three basic types of algorithms, which are to be used for automatic path provisioning: hierarchical routing, step-by-step routing, and source routing. Source routing is essentially defined as a hop-by-hop interaction between the nodes on the connection path.

A typical example of such a setup protocol implementation is the Resource Reservation Protocol (RSVP). For details see [35], [36], [37], [38].

In the current source routed connection establishment procedures, traditionally the sequential setup is employed to establish connections across a network. When a connection is to be set up in a network, a setup request message is generated at the starting A node and transmitted from there to a first intermediate node in a sequence of nodes - well known at the A node - that will be traversed by the connection.

Fig. 9 Sequential Setup Request



The A node processes the setup request (*Path* message in RSVP implementation) in the time *PpA* and after processing, forwards the message to the first intermediate node I1. The transmission of this message takes some time *TpI1*.

The first intermediate node processes the setup request message received from the begin node (*PpI1*) and then forwards the setup request message to the next intermediate node in the sequence (*TpI2*), and so on. This process continues until the final intermediate node in the sequence transmits the connection request message to the end node in the sequence.
As the connection request message follows the traversed intermediate nodes along the connection path between the begin node and the end node, specific actions are performed at each of the intermediate nodes and the end node that require a certain amount of processing time (*PpIi*). Such actions include receiving and reading the message, allocating the bandwidth required by the connection path, performing checks, and the like.

Using this method of serial transmission and processing, the connection setup time is calculated as the sum of the transmission times required to transmit the connection request message from one node in the sequence to the next node in the sequence, and the processing time that each node in the sequence spends processing the connection request message.

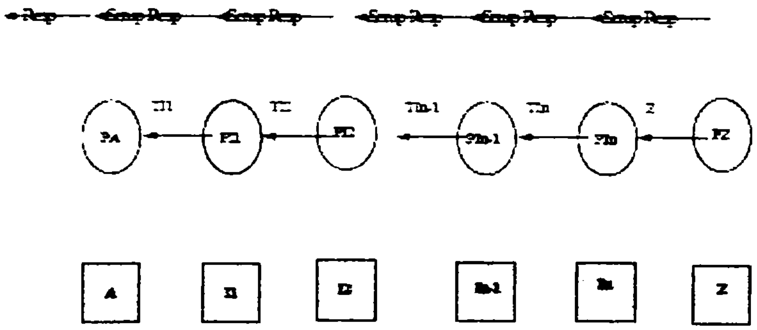
The total time to perform a setup request is:

$$SetupTimeP_{sequential} = PpA + \sum_{i=1}^{N} (TpIi + PpIi) + TpZ + PpZ \qquad (3.1)$$

where i=1...N are the intermediate nodes to A (head-end) and Z (tail-end)

Therefore, in this method, and other serial methods of establishing connections across a network having a plurality of nodes, the connection setup time is a direct function of the connection path length *N* (number of nodes in the sequence).

Fig. 10 Sequential Setup Response



After the setup request has reached the end node, there is typically a handshake between the end node and the begin node in which the success (or failure) of the connection establishment is communicated from the end node to the begin node. In the RSVP implementation this is the *Resv* message which is also traversing hop by hop the intermediate nodes up to the begin node consuming for the transmission a certain time (*TrIi*). On each hop a specific processing time (*PrIi*) is needed.

The total time to perform a setup response is:

$$SetupTimeR_{sequential} = PrA + \sum_{i=1}^{N} (TrIi + PrIi) + TrZ + PrZ \tag{3.2}$$

where $i=1...N$ are the intermediate nodes.

Notice that depending on the implementation, some of the time consuming activities (e.g. interaction with different controller for resource allocation) may be performed either on the Setup Request (Path) or on the Setup Response (Resv). In the following the generic term SetupTime will be used for both Request and Response part of the procedures and time calculation. However it should be noticed that the for the total calculation the time consumed as well as the time saved with the alternative are to be accumulated.

As the size of communication networks continues to grow, the connection paths provisioned through a network tend to traverse increasing numbers of nodes, significantly increasing the time required to establish a connection path through the network.

So, the problem is that some critical time restrictions cannot be fulfilled if the path length exceeds a certain maximum. Therefore, a faster method of performing source routed connection path provisioning is desirable.

3.2. **Basic Idea of the Proposal**

I proposed the method for fast source routed connection setup in the context of a mesh network topology. However, the methodology can readily be applied to other network topologies.

With this new method, I address performance deficiencies by providing a method for establishing a connection between a begin node and an end node by the parallel transmission and processing of connection request messages, thereby minimizing the dependency of the connection setup time on connection path length. *"Specifically, the method comprises transmitting respective connection request messages to each of a plurality of nodes scheduled to form a connection path, wherein the connection setup request (and/or the setup response) messages are adapted to cause the plurality of nodes to initiate the formation of respective portions of the connection path in a substantially contemporaneous manner."* [11]. The present mechanism employs a parallel approach to network connection setup that utilizes existing distributed processing potential to minimize the dependency of connection setup time on connection path length.

In parallel connection setup, rather than transmitting a single connection request message from a begin node to a first intermediate node (and so on, hop-by-hop, until the connection request message reaches the end node), a begin node transmits respective connection request messages to each of a plurality of nodes elected to form a connection path. The respective connection request messages are adapted to cause the plurality of nodes to initiate the formation of the respective portions of the connection path. In the context of source routing it is assumed that the sequence of nodes to be traversed (A, Ii, Z) is known. In the RSVP protocol this sequence is actually included in the Path message via some specific objects Explicit Route Object (ERO), Record Route Object (RRO).

For the parallel connection setup I'm proposing two alternative of synchronization:

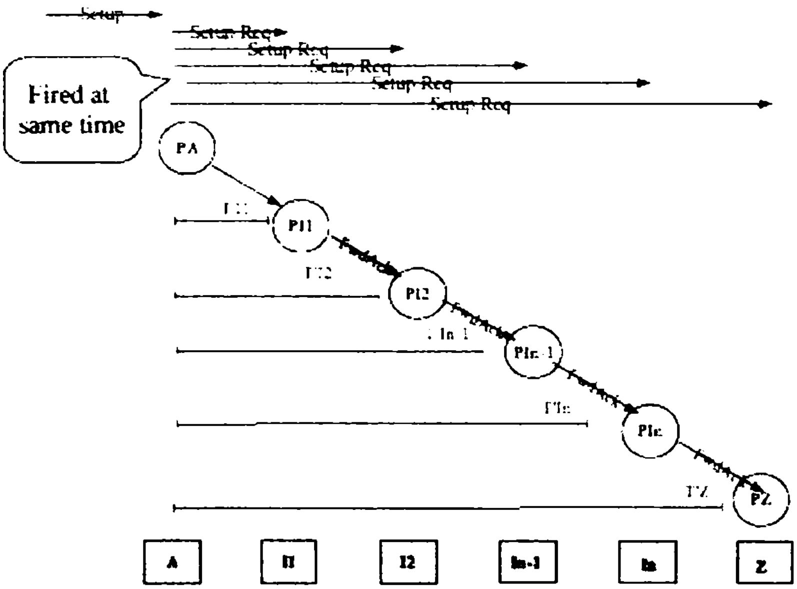
a) sequential synchronization on intermediate nodes and

b) final synchronization in the end node

3.3. Detailed Description

3.3.1. Parallel Setup With Sequential Synchronization

Fig. 11 Parallel Setup with sequential synchronization



In the variant of sequential synchronization, each intermediate node operates as a partial synchronization point for a previous intermediate node in the sequence of nodes scheduled to form a connection path. Each intermediate node transmits a Forward Acknowledge (*FwdAckIi*) message to a next intermediate node (Ii) in the sequence. The end node (Z) decides on completion (usually initiating the Setup Response) as soon as the corresponding Forward Acknowledge message have been received from its previous node I_N (compare Fig. 11 Parallel Setup with sequential synchronization),

Usually
$$T'z < \sum_{i=2}^{N} FwdAckI_i \tag{3.3}$$

Therefore the setup message reaches the node Z (after *T'z*) before the Forward Acknowledge from its upstream node *FwdAck*$_Z$. So that node Z can perform its processing already before it received the FwdAck$_Z$.

In order to estimate the time savings, I make for simplification following assumptions:

- Equal processing times for all nodes *PpA* = *PpIi* = *PpZ*
- Processing time *PIi* is in general greater than the transmission time *T'Ii*. This is in general a valid assumption, since the processing time covers resource allocations, usually related to additional communication between different controller in the node. In particular, in overload situations it is rather the processing time which is affected.

So, the usual temporal sequence of events is *PpA*, *FwdAck*$_2$, *FwdAck*$_i$, ..., *FwdAck*$_Z$.

The SetupTime for the sequential Synchronization is:

$$SetupTime_{sequentialSynchronisation} = PpA + \sum_{i=2}^{N} FwdAckI_i + FwdAck_Z \tag{3.4}$$

where *i*=2...*N* are the intermediate nodes

This reflects that the setup time is independent of the processing time in the intermediate nodes.

Assuming for simplicity equal transmission times for the Forward Acknowledgement:

$$SetupTime_{sequentialSynchronisation} = PpA + N * FwdAckI_1 \tag{3.5}$$

The direct transmission time *T'* is slightly increasing towards Z, therefore *T'i+1* > *T'i* However it remains comparable with *FwdAckI*, the transmission time of the ForwardAcknowledge. So, the difference between the sequential setup and the parallel setup with sequential synchronization results by comparing (3.1) with (3.5):

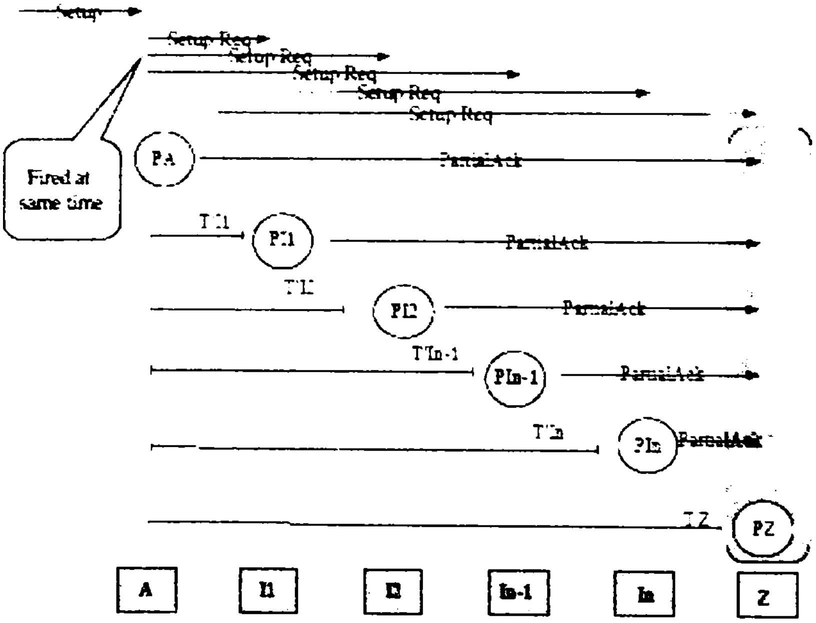
$$SetupTime_{sequentielSetup} - SetupTime_{sequentielSynchronisation} \geq \sum_{i=1}^{N} PpI_i + PpZ \tag{3.6}$$

In other words the benefit of the sequential synchronization is the result of the enforced parallelization of the setup processing in the intermediate and final nodes, saving the processing time in these nodes. The estimated result was validated by experimental tests.

3.3.2. Parallel Setup With Final Synchronization

This option of the parallel setup may be used if there is no need for intermediate synchronization points. Again the begin node A starts sending the Setup Requests to all intermediate nodes as well to the end node Z. The intermediate nodes send Partial Acknowledge (*PartAck*) - as soon as they complete their processing. Using parallel synchronization, the end node operates as a central synchronization point for each of the intermediate nodes in the sequence of nodes elected to be traversed by the connection. The end node (Z) decides on completion, usually initiating the Setup Response, as soon as all Partial Acknowledge message have been received. Compare following figure.

Fig. 12 Parallel Setup With Final Synchronization



In case of final synchronization the Setup time is at worst:

$$SetupTime_{FinalSynchronisation} \leq max(T'I_i) + max(PI_i) + max(PartialAckI_i) \quad (3.7)$$

where *i=A,* 1...*N, Z* (ingress, intermediate nodes, egress node)

Again compared with the sequential Setup the parallel Setup with Final Synchronization saved N processing times *PIi.*

3.4. **Benefits & validation**

The direct advantage of the parallel setup is of course the reduced setup time. This is of benefit for path setups which are time critical for instance in the context of Path Restoration. The implicit positive effect is the increased reliability because the improved performance on network recovery and better scalability because the reduced dependency on the path length N.

See below for some examples.

Assuming for simplicity that:

- transmission time is increasing linearly with the hop distance:

$$T'I_i = i * T'I_1 \qquad (3.8)$$

$$PartialAckI_i = (N - i) * T'I_1 \qquad (3.9)$$

- equal processing times *PIi*

$$PI_1 = \ldots = PI_i = \ldots = PI_n \qquad (3.10)$$

we obtain the simplified formula

$$SetupTime_{FinalSynchronisation} = PA + N * T'I_1 \qquad (3.11)$$

Notice that the transmission time to the first intermediate node are the same for the original sequential and for the parallel setup: $TI_1 = T'I_1$ (3.12)

The SetupTime for sequential synchronization and for final synchronization are equal:

$$SetupTime_{SequentiellSynchronisation} = SetupTime_{FinalSynchronisation} = PA + N * TI_1 \qquad (3.13)$$

This is valid for the worst case of a ring topology with hop by hop access to each intermediate node.

In the best case in a completely meshed topology each node is directly accessible, therefore:

$$T'I_i = PartialAckI_i = T'I_1 \qquad (3.14)$$

So the best case Setup Time for final synchronization is:

$$SetupTime_{FinalSynchronisationBestCase} = PA + T'I_1 \qquad (3.15)$$

This shows that scalability is again increased since the strict dependency on the path length is minimized and in the best case even eliminated.

In conclusion the reduction of the Setup Time compared to the classic sequential Setup is in the following range:

$$N * PI_1 \leq ReductionSetupTime \leq N * PI_i + N * TI_i \qquad (3.16)$$

The conclusion is that the maximal reduction of the setup time may be achieved with the Final Synchronization in a well meshed topology.

The performance improvement was validated by simulation tests. As a practical example processing time let us assume:

- the processing time in an intermediate node *PIi* = 30ms,
- transmission time: *TIi* = 10ms,
- path of length *N* = 9 (intermediate nodes)

The Setup Time is reduced from the classical sequential way:

$$SetupTime_{SequentialClassic} = 30ms + 9*30ms + 9*10ms + 10ms + 30ms = 430ms$$

to 1/3 of its value for the sequential synchronization method.

$$SetupTime_{SequentialSynchronisation} = 30ms + 9*10ms + 10ms = 130ms$$

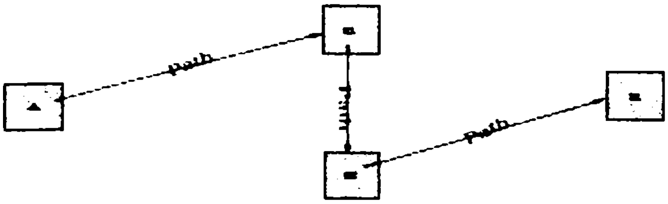
An additional reduction with maximum of 1/3 of the Setup Time for the sequential synchronization may be obtained with the final synchronization method in the context of a completely meshed topology.

$$SetupTime_{FinalSynchronisationBestCase} = 30ms + 10ms = 40ms$$

In this example the Setup Time with the method of final synchronization was reduced to 1/10 compared with the classic method of sequential setup.

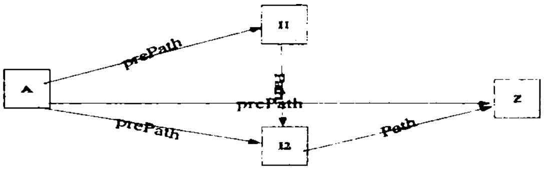
Please notice that for a complete setup (as mentioned before - see Background) the nodes are traversed two times: for the setup request and for the setup response. So the time budgets are in general doubled as well as the corresponding savings. However if a processing *PIi* does need results obtained from the previous processing, then it can start only after receiving the *FwdAckIi*. This may be the procedure for some resource allocations performed either on the setup request or setup response. In this case the benefit of the parallel setup will apply only once, for the direction opposite to the constraint resource allocation.

Fig. 13 Sequential Setup in a meshed topology



The negative side effect of the parallel setup is the fact that some additional messages are generated. For instance in the case of the sequential setup the A node generates a number of additional Setup Requests corresponding to the number of intermediate nodes. Apparently this is reducing the scalability. Looking closer it turns out that this affects mainly the ingress node. However even so, the load would be in general distributed over more than one outgoing interfaces.

Fig. 14 Sequential Synchronization in a meshed topology

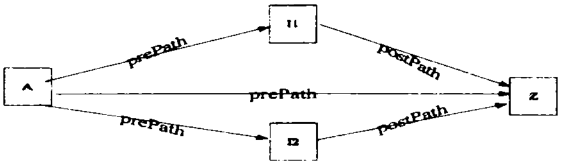


The good meshed topology in Fig. 14 shows the situation where the A nodes has Control Plane (CP) adjacency to all related nodes. Notice that in this example the transport link A-Z is already busy or failed (recovery scenario), so that the path must be established on the alternate route I1, I2, Z. This illustrates how the additional messages are actually distributed over different CP links, minimizing the scalability issue. The procedure assumes direct IP addressing.

In case of a multicast network (LAN) an additional optimization may be considered by using an IP multicast address. However this would imply a common message content for all destinations. The additional load on an intermediate node (e.g. node I2) is just one message.

The principle of synchronization points may be used also in case when setup decision is dependent on conditions available from the an external source (e.g. central management system).

Fig. 15 Final Synchronization in a meshed topology



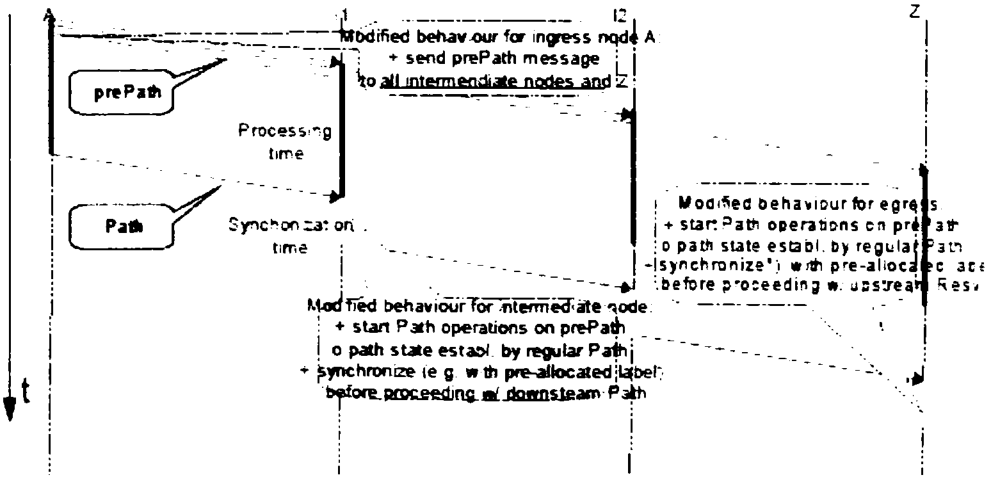
3.5. Implementation in the RSVP context

Whereas it is obvious that the abstract Setup Request translates in the RSVP context to the *Path* message and that the Setup Response translates to the *Resv* message, there are no specific RSVP messages for the abstract *FwdAck* and *PartialAck*.
The first approach would be to implement them as RSVP Notify message, which is different from the common RSVP messages in that it is "targeted" to another node than the immediate up- or downstream neighbor. This would fit for instance for the *PartialAck*, which is directly addressed to the final node. However this solution would violate RFC3473 which states that a notify should be sent to a node only if it was explicitly requested by a corresponding notify request.

3.5.1. Sequential Synchronisation

Looking closer to the specific activities behind the sequential synchronization, it turns out that the Setup Req messages, which are fired simultaneously to all nodes implied in the path are essentially a kind of preliminary - lets call them "*prePath*"-messages, suitable for some preparating operations like validation checks. Instead, the abstract FwdAck message could be directly implemented as regular Path traveling the LSP hop-by-hop in the traditional way. The *prePath* message is to be sent to the end node and to all intermediate nodes, except the first intermediate node, which is informed by a regular Path. For the *prePath* message, it would not be necessary to set the complete Explicit Routing Object (ERO), since these messages end in the intermediate nodes. However, the presence of the ERO may provide additional information about the expected resource allocation.

Fig. 16 Sequential Synchronization: Modified behavior


A
I1
I2
Z
Modified behaviour for ingress node A
+ send prePath message
to all intermendiate nodes and Z
prePath
Processing time
Path
Synchonization time
Modified behaviour for egress
+ start Path operations on prePath
o path state establ. by regular Path
-(synchronize") with pre-allocated label
before proceeding w. upstream Resv
Modified behaviour for intermediate node
+ start Path operations on prePath
o path state establ. by regular Path
+ synchronize (e.g. with pre-allocated label)
before proceeding w/ downsteam Path
t

The *prePath* message has the same content as the traditional Path, except it is addressed directly to an intermediate nodes via corresponding IP destination address. In order to avoid confusions with the regular Path, the MessageID is omitted, which in turn prevents the A node from receiving a bulk of acknowledges from every intermediate node. The RSVP_HOP can be set to the A node, thus giving an implicit indication (by comparing with the ERO object) for the special treatment. The explicit indication of the special semantic of the prePath is possible by introducing a new bit: "PREliminary" to the ADMIN_STATUS object, in addition to the existing R, T, A, D bits specified by RFC3473. This would be the natural way, however if there objections because the reserved part of this object, the session name as part of the Session_Attribute may be used instead.

```
IP Dest Addr = I2.IPaddress
//NO [<MESSAGE_ID>](=>no ack)

<SESSION>
   IP tunnel end = Z.IPaddress
   TunnelID
   ExtTunnelID = A.IPaddress
<RSVP_HOP> = A
//[<EXPLICIT_ROUTE>]= I1,I2,Z    opt.
<LABEL_REQUEST>
[<SESSION_ATTRIBUTE>]
  Session name = prePath
<sender descriptor>
   <SENDER_TEMPLATE> ...
   <SENDER_TSPEC> ...
   [ <ADSPEC> ...
```

Fig. 17 PrePath message sent by A to I2

The procedure of a node receiving the *prePath* message is as follows: The RSVP Path state is established as normal. In addition preliminary checks may be performed on the related resources and on positive result they may be reserved (e.g. time slots, database records). The actual allocation as well as related database commitment may occur after the regular Path message is received from the previous intermediate node. If no regular Path is received in the following refreshing time, the Path state and the related pre-reserved resources are freed. This behavior is actually given by the traditional RSVP soft state behavior. The association of the regular Path with the prePath message is by using the same TunnelID in the Session object.

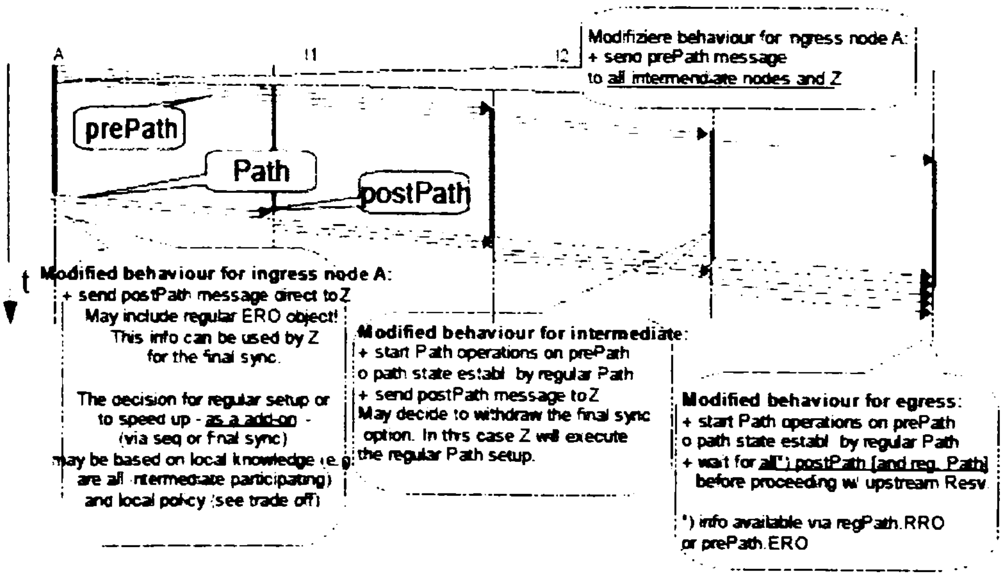
Notice the path setup shows now the characteristic of a two phased transaction: preparation phase followed by the commit or a roll-back phase. In general most of the initiated Path Setups will be probably successful since they have been planned and initiated on the A node based on the initial availability of the related resources. However if some conditions changed in the mean time the roll-back is implicitly supported by the regular RSVP procedure with the soft state approach. In addition the A node will initiate the Path Teardown in case of an outstanding Resv message.

Similar considerations can be made for the setup Response, which may use a special preliminary Resv - lets call it "preResv" message, possibly triggering some preliminary label operations. PreResv is directly IP addressed to all intermediate nodes and to the A node, whereas the abstract *FwdAck* message is implemented as regular Resv message, traveling the LSP hop-by-hop in the traditional way.

3.5.2. **Final Synchronisation**

The final synchronization is in particular interesting for such cases where the intermediate nodes are restricting some of the setup conditions (e.g. suggested label set, resource affinity) potentially leading to a negative result (PathError), which can be concluded more faster. Also for the positive case, the setup time is reduced since the Z node can proceed with the Resv rigth after receiving all *PartialAcks,* even before regular Path was received. The *PartialAkcs* can be implemented as Path with some modifications: It is IP addressed directly to the egress node, without the Message_ID.

Fig. 18 Final Synchronization: Modified behavior


Modifiziere behaviour for ngress node A:
+ seno prePath message
to all intermendiate nodes and Z
A
I1
I2
prePath
Path
postPath
t
Modified behaviour for ingress node A:
+ send postPath message direct to Z
May include regular ERO object!
This info can be used by Z
for the final sync.
The decision for regular setup or
to speed up - as a add-on -
(via seq or final sync)
may be based on local knowledge (e.g.
are all intermediate participating)
and local policy (see trade off)
Modified behaviour for intermediate:
+ start Path operations on prePath
o path state establ by regular Path
+ send postPath message to Z
May decide to withdraw the final sync
option. In this case Z will execute
the regular Path setup.
Modified behaviour for egress:
+ start Path operations on prePath
o path state establ by regular Path
+ wait for all*) postPath [and reg. Path]
before proceeding w/ upstream Resv
*) info available via regPath.RRO
or prePath.ERO

Its special semantic could be indicated indirectly e.g. by comparing RSVP_HOP, ERO, and Record Route Object (RRO). The preferred option is the explicit one by setting a new ADMIN_STATUS.PostBit

Alternatively the session name could be used for this purpose. The modified behavior is illustrated in the figure above. The A node sends the prePath message to Z and to all intermediate nodes. The intermediate nodes can proceed with the postPath message as soon as they have decided on a positive result of the preliminary Path operations. The Z node should wait for all postPath messages before proceeding with the upstream Resv message.

IP Dest Addr = Z.IPaddress

```
//NO [<MESSAGE_ID>](=>no ack)
<SESSION>
   IP tunnel end = Z.IPaddress
   TunnelID
   ExtTunnelID = A.IPaddress
<RSVP_HOP> = A or missing
//NO [<EXPLICIT_ROUTE>]
<LABEL_REQUEST>
[<SESSION_ATTRIBUTE>]
  Session name = postPath
<sender descriptor>
   <SENDER_TEMPLATE> ...
   <SENDER_TSPEC> ...
   [ <ADSPEC> ]...
   [ <RECORD_ROUTE> ] = A
```

Fig. 19 PostPath message sent by I2 to Z

3.5.3. **Conclusions**

The proposed RSVP implementation is evolutionary and backward compatible, so it can be applied also to networks containing nodes handling traditional and enhanced RSVP setup messages. Compared with a similar proposal [63], it is more generic and avoids the introduction of new RSVP messages. The preferred option for extending the semantic of the RSVP message is by introduction of a new ADMIN_STATUS bit. However also an implicit indication of the new semantic would be possible.

The ingress node may decide on any of these alternative procedures, since they do not require a divergent behavior in any of the implied nodes. In general the *Final Synchronization* can be adopted if there are no dependencies from the previous hop (e.g. Path for unidirectional setup). Instead, the *Sequential Synchronization* may be adopted when there are dependencies from the previous hop e.g. Resv.Label).

One of the application which may take a benefit of the here suggested RSVP implementation is the setup of the protecting LSP on restoration (comp. "make-before-break" in RFC3209). A specific of this path setup is the evaluation of the setup and holding priority, which is necessary in order decide on the preemption of traffic. Also the operations related to the resource affinity (exclude, include-any/all) may be processed in parallel on each of the intermediate nodes. The preemption is in general an operation which can benefit from the preliminary verification of sufficient bandwidth at a particular priority along the entire path. As is indicated in the RFC3209 *"If a Path message is allowed to progress when there are insufficient resources, then there is a danger that lower priority reservations downstream of this point will unnecessarily be preempted in a futile attempt to service this request".*

Another suitable application is the path setup over multiple segments (ENNI domains). The specifics of this path setup is the presence of some intermediate border nodes which are accessing potentially opaque domains/segments. They are good candidates for sequential synchronization. The final synchronization may be performed in the corresponding intermediate egress nodes.

One drawback is some increased complexity of the RSVP state machine. However it should be noticed that the modification is introducing a kind of a two phased transaction (preparation phase followed by the commit or a roll-back phase), which is in general a useful pattern. Another drawback is some decreased scalability because the additional number of pre/postPath messages. However this deficiency may be attenuated if the end nodes and intermediate nodes are well meshed. The benefit is a considerably reduced setup time for the positive case as well as for the negative case since resource conflicts can be detected faster.

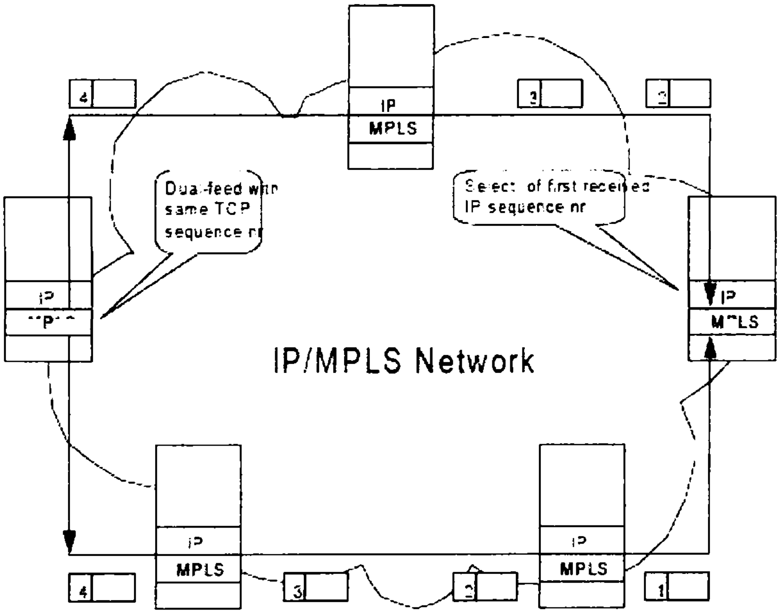
4. Improved 1+1 MPLS Packet Protection By Preventive Detection Of Quality Degradation

4.1. Background

ITU-T G.7712 mandates MPLS 1+1 Packet Protection via two associated, node disjoint Label Switched Path (LSP) (see. [60], 7.1.19.1).

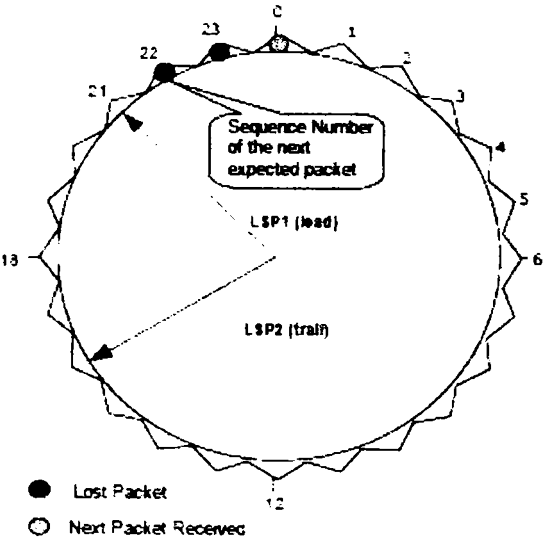
The basic idea is to send duplicates of critical messages over these LSP: a leading and a trailing (as a kind of backup) LSP. The identification of the duplicated messages is by their sequence number. The sequence number shall be carried in every packet as the first four bytes inside the so called shim header of each of the LSP providing packet 1+1 protection. If packets are lost on the leading LSP, they can be recovered by receiving the duplicates on the trailing LSP.

Fig. 20 Mechanism of Packet 1+1 Protection - ITU-T G.7712


4
IP
MPLS
3
2
Dual-feed with same TCP sequence nr
Select of first received IP sequence nr
IP
MPLS
IP
MPLS
IP/MPLS Network
IP
MPLS
IP
MPLS
4
3
2
1

In general in order to solve the wrap around problem of the sequence number when reaching the 2**N limit a so called *Sliding Window* (SW) is recommended as a well-known mechanism. ITU-T G.7712 also suggests the usage of a Sliding Window (see Appendix Iv) in particular related to the situation when packets are lost : *„The sliding window is used to solve the problem of losing packets on the leading LSP when the leading LSP sequence number is is very close to the wrap around point."*

Fig. 21 Sliding window mechanism for wrap around



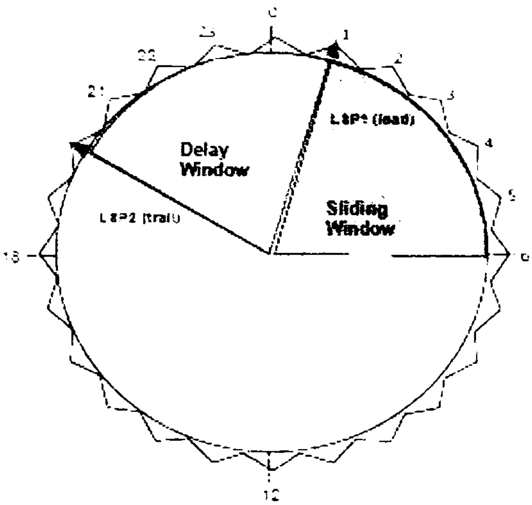
In addition, it recommends to use a *Delay Window (DW),* which size is to be configured as the maximum number of packets the trailing LSP can fall behind the leading LSP. Following relation must be fulfilled:

$$SlidingWindow + DelayWindow < 2^{N} \tag{4.1}$$

ITU-T G.7712 further states: „*One reasonable way of engineering the size of sliding and delay window is to make the size of the sliding window equal to the size of the delay window".* And further elaborating, it finally recommends:

$$SlidingWindow \geq DelayWindow \tag{4.2}$$

Fig. 22 Relation between Sliding and Delay Window



The size of the Sliding Window is usually fix, configured to ½ of the upper limit of the sequence number:

$$SlidingWindow = 2^{N-1} \tag{4.3}$$

With $DelayWindow = SlidingWindow - 1$ (4.4)

all recommendations of ITU-T G. 7712 would be fulfilled.

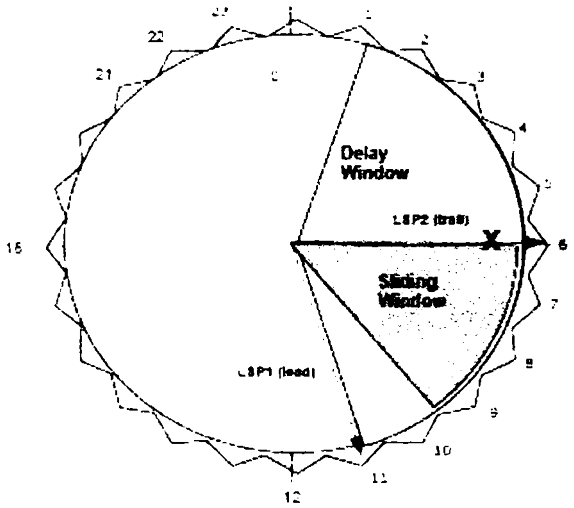
However, such a configuration has some disadvantages in practice. A large *SW* (recall that N is 32) must be supported by corresponding large buffer resources of the transport layer (TCP) because possible re-alignments of the packages.

Despite of this, the applications requiring high reliability cannot tolerate to lose a large number of packets.

This was for me the motivation to give up the maximal configuration approach. So I started to think about realistic configurations. Following related problems come up in this context:

- The *DW* should reflect the characteristics of a LSP pair (for example different length of the LSP). For a given LSP pair the characteristic differences may also vary in time (time dependent routing load). What to do if the configured DW value is exceeded?
- The SW should be „large enough" to cover the variations of the *DW (SW > DW*). What to do if the configured *SW* value is exceeded?

Fig. 23 Short Sliding Window (compared with Delay Window)



The *SW* should also be *„larger than the maximum number of consecutive packets a working LSP can lose"* [60]. This should rather read as „... a working LSP can afford to loose a certain number of packages before the using application would be affected".

Fig. 23 Short Sliding Window (compared with Delay Window) *"...illustrates a failure to the Trailing LSP. Since the Leading LSP delivers packets outside the Sliding Window and, therefore, those packets are rejected, the egress node will not start accepting packets until the Leading LSP comes all the way around and starts to deliver packets with a sequence number that falls within the SlidingWindow. This can result in a significant loss of packets. Therefore, to prevent such an occurrence, it is recommended that this type of selector algorithm set the SlidingWindow equal to the DelayWindow".*

4.2. **Basic Ideas**

In order to address the issues mentioned in the previous section:

I proposed a mechanism which extends the traditional MPLS 1+1 Packet Protection as described in ITU-TG.7712 by integrating specific application needs and taking into account the dynamics introduced by the supporting LSP.

I introduced adequate parameters to measure the quality of the LSP to detect degradation and to initiate counter measurements.

This mechanism can be used in any network which supports MPLS 1+1 Packet Protection. It is in particular recommended for time critical applications since offers significant improvements in terms of reliability and restoration performance compared to the traditional MPLS Packet 1+1 Protection by ITU-T G.7712.

4.3. **Detailed Description**

I'm proposing following extensions:

1. Define a measure ***Q*** for the overall quality of the MPLS packet 1+1 protection. This quality Q is best expressed as the inverse of a certain ***tolerance T*** of an application in loosing packets:

   T = maximum number of consecutive packets which is acceptable to be lost for a certain application.

   $$Q = f(1/T) \qquad (4.5)$$

   If an application cannot afford to loose too much packets (this includes, late packets, which did not arrived in time), then a higher quality of the MPLS packet 1+1 protection is required.
   The limit on the number of lost packets is in general far below 2 ** (N-1). Obviously exactly the time critical packets are to be protected by sending them over an MPLS packet 1+1 protection. In practice they may contain some real time data (e.g. audio or video streams), control plane messages initiating restoration, etc.

   If the number of consecutive packets lost is exceeding a certain critical value

   $$T_{critical} = f1 * T, \qquad (4.6)$$

   where f1 is a configuration factor, f1 < 1

   then the quality of the MPLS packet 1+1 protection is tending to degrade, so counter measurements may be indicated - compare extensions 4 and 5.

2. Define a ***Current Sliding Window*** (*CSW*), and initialize it with:

   $$CSW\ (t0) = T + 1 \qquad (4.7)$$

   Notice that the Sliding Window is supposed to have a variable size, large enough to recover *T* lost packets, but far below 2 ** (N-1), the value proposed by ITU-T.

3. Define a ***Current Delay Window*** (CDW) by monitoring the difference between the leading LSP and the current trailing LSP.
   Notice the dynamic property of the CDW: It is obtained by monitoring the actual LSP pair instead of having a pre-determined fix size size.

   The Delay Window is supposed to have a variable size, small enough to be able to recover T lost packets, and far below 2 ** (N-1), the value proposed by ITU-T.

   The system must enforce at any time following condition:

   $$CDW\ (t) < CSW\ (t) \qquad (4.8)$$

If this condition is violated by an increased value of the CDW, the system must react by aligning the CSW:

$$CSW\ (t+1) = CDW\ (t+1) + 1 \qquad (4.9)$$

In this case the quality of the MPLS packet 1+1 protection decreased, because the application would be forced to tolerate additional packets lost. The operator could be notified about this quality degradation.

In order to avoid such a situation,I'm proposing following preventions:

4. Define the service quality of the leading LSP ***qLSPlead*** and measure by monitoring the ratio of packets lost on the leading LSP against the CSW:

$$qLSPlead = 1 - \text{number of lost packets}/T \qquad (4.10)$$

The range of the qLSPlead is between 1 (means no lost packets) and 0 (when the maximum tolerated number of packets T was lost).

If the qLSPlead is continously decreasing and falls below some critical value:

$$qLSPlead_{critical} = 1 - f4 * CSW/\ T, \qquad (4.11)$$

f4 a configuration factor, f4 < 1

then a trigger can be raised in order to initiate counter measurements, e.g. computation and/or replacement of the leading LSP by an other one which satisfies the conditions.
Notice that the definition (4.11) is preferred to an alternative definition (4.11") since it describes better the situation when the quality degradations is caused by a larger CSW (compare. (4.9))

$$qLSPlead'_{critical} = 1 - f4, \qquad (4.11')$$

f4 a configuration factor, f4 < 1

5. Define the service quality of the trailing LSP ***qLSPtrail***, and measure by calculating each time a packet is received on the LSPtrail, the ratio of CDW(t) against the previous value CDW (t-1):

$$qLSPtrail = 1 - CDW(t)/CDW(t-1) \qquad (4.12)$$

The value of the *qLSPtrail* for a constant delay on the LSPtrail is 0, but it may vary from a maximum close to 1 for a sudden delay reduction (CDW(t) << CDW(t-1)) to a minimum -X for a sudden delay increase (CDW(t) >> CDW(t-1)).
If the qLSPtrail is constantly decreasing and falls below a critical value:

$$qLSPtrail_{critical} = f5 * (1 - T), \qquad (4.13)$$

f5 a configuration factor, f5 < 1

then a trigger is issued in order to initiate counter measurements: e.g.

computation and/or replacement by another trailing LSP which satisfies the conditions.

4.4. An example

Following examples illustrate for the lifetime of a LSP pair some specific situations which can take advantage of my proposal.

A given application may have quality requirements which translates to a tolerance T in loosing (or receiving delayed) a certain number of consecutive packets without a significant impact.

Lets assume: tolerated number of consecutive lost packets T = 5 and configuration factor f1 = 60%.

The degradation of the corresponding dedicated LSP pair begins if the following limit is exceeded.

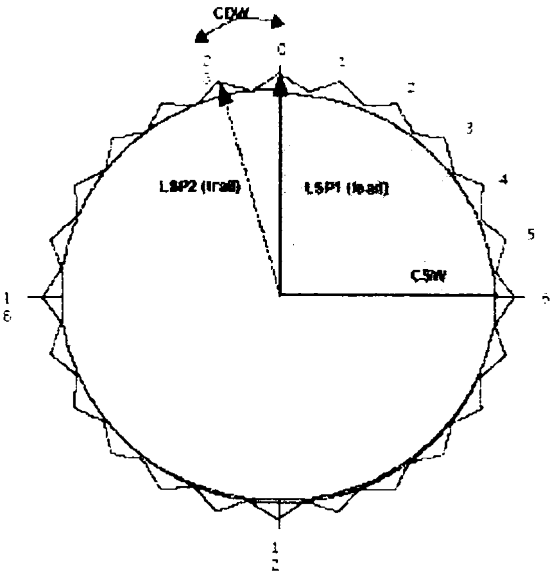
$$T_{critical} = f1 * T = 3$$

This means that if 3 consecutive packets are lost, counter measurements should be initiated for instance by an external operator. However some corrective actions can be autonomously performed by monitoring and modifying the parameter CSW and CDW as follows:

According to (4.7) the CSW is initialized to:

$$CSW(t0) = T + 1 = 6$$

Fig. 24 Initial state



By monitoring the difference between the leading LSP and the trailing LSP, the system set the CDW (t0) = 1, corresponding to the situation where the package with the same sequence number is received on the trail LSP with a minimal delay.
The initial situation is shown in the next figure.
For better understanding the wrap around of the sequence number is set to: N = 24

The next figure illustrates the situation at t12 where still no packets have been lost. In addition the packages received on trailing LSP are still one sequence number behind the leading LSP, means that the value for *CDW* is confirmed at t1, ..., t6:

$$CDW\ (t1) = \ldots = CDW\ (t6) = 1$$

The quality of the leading LSP (LSP1) is still on its initial maximal value:

$$qLSPlead\ (t6) = 1 - 0/T = 1$$

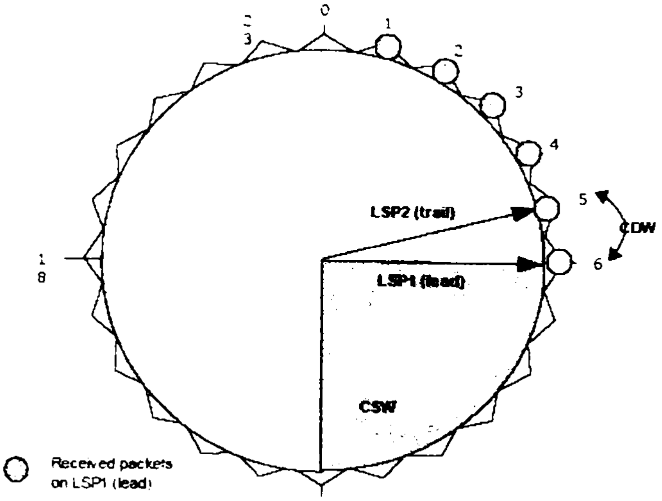
Lets set the configuration factor f4 = 33%, means if qLSPlead falls below the following critical value, than a specific trigger should be raised (4.11)

$$qLSPlead_{critical} = 1 - f4 * CSW/T = 1 - 1/3 * 6/5 = 3/5$$

The quality of the trailing LSP (LSP2) is still on its initial value:

$$qLSPtrail(t6) = 1 - 1/1 = 0$$

Fig. 25 No packets lost at t6



Lets set for the configuration factor f5 = 50%, means if qLSPtrail falls below the following critical value than a specific trigger should be raised (4.12)

$$qLSPtrail_{critical} = f5 * (1\text{-}T) = 1/2 * (1\text{-}5) = \text{-}2$$

The next figure illustrates the situation where 2 packets get lost on the leading LSP1.
This may have following consequences:

- Degradation of the LSP pair, if $T_{critical}$ is reached. This situation should be notified to the operator, which optionally may agree to a lower quality resulting in a higher value for the tolerance T. However, in this situation the $T_{critical}$ is not yet reached for 2 packages lost. It would be reached if 3 packages get lost.

- qLSPlead is decreasing form its initial value 1 to:

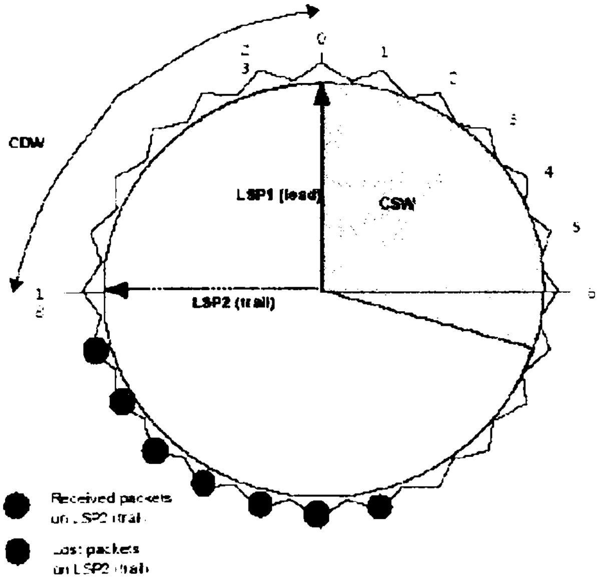
$$qLSPlead\ (24+6) = 1 - 2/5 = 3/5$$

Since the $qLSPlead_{critical} = 3/5$ is reached, a trigger can be issued to initiate counter measurements.

Notice that in general for a factor f1, which is possibly configured in agreement with the operator, it is possible to tune the factors f4 and f5 so that the values $qLSPlead_{critical}$ and/or $qLSPtrail_{critical}$ are reached before external alarms are raised to alert the operator.

Of course the packages lost on the LSPlead at t(24+3) and t(24+4) could have been possibly recovered on the LSPtrail, however only if the LSPtrail provides sufficient 'good quality'.

Fig. 26 Loosing 2 packets on LSPlead at t(24+6)


CDW
0
1
2
3
4
5
6
1
8
LSP1 (lead)
CSW
LSP2 (trail)
Received packets on LSP2 (trail)
Lost packets on LSP2 (trail)

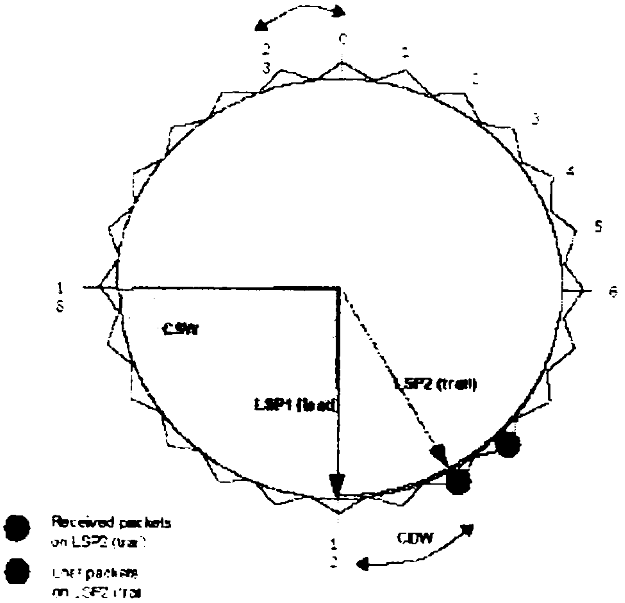
BUPT

The next figures illustrate some critical situations on the LSPtrail: One packet is lost on the LSPtrail, so the *Current Delay Window* must be increased to:

CDW (t24+10) = 2

However the next package is received in time at t(24+9) (compare the blue bubble in Fig. 27 Loosing 1 packet on LSPtrail at t(24+10)).

Fig. 27 Loosing 1 packet on LSPtrail at t(24+10)


0
3
4
5
6
1
CSW
LSP1 (lead)
LSP2 (trail)
CDW
Received packets on LSP2 (trail)
Lost packets on LSP2 (trail)

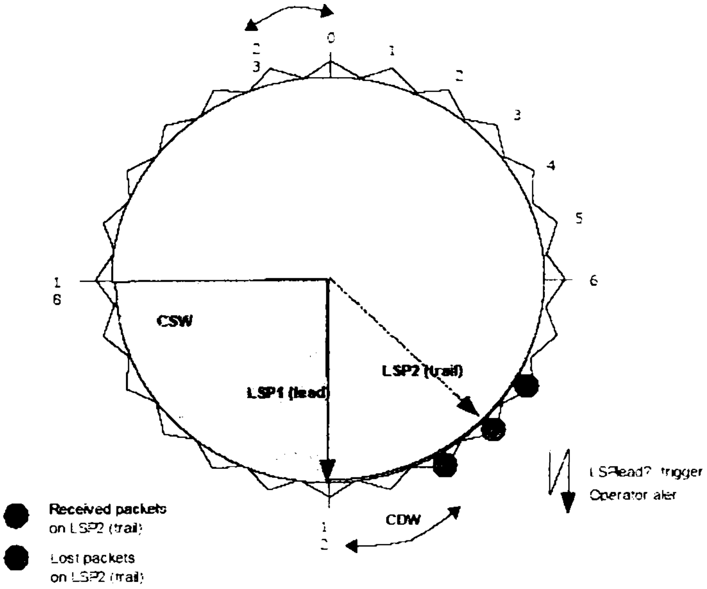
Therefore the quality of LSPtrail is:

qLSPtrail (t24+10) = 1 - 2/1 = -1

The critical value qLSPtrail critical = -1 is not yet exceeded, so no trigger must be issued yet. However if at t(24+9) already 2 packets get lost on the LSPtrail - compare next figure, then the CDW is increasing to:

CDW(t33) = 3

Fig. 28 Loosing 2 packets on LSPtrail at t(24+9)



In this case the quality of the *qLSPtrail* decreases further:

$$qLSPtrail\ (t34) = 1 - 3/1 = -2$$

The critical value $qLSPtrail_{critical} = -1$ is exceeded, so a trigger should be issued indicating a serious degradation of the LSPtrail.

If the LSPtrail continues to loose packets, then the CDW must be increased accordingly. For example in Fig. 29 at t(24+12) already 5 packages are lost on the LSPtrail. So:

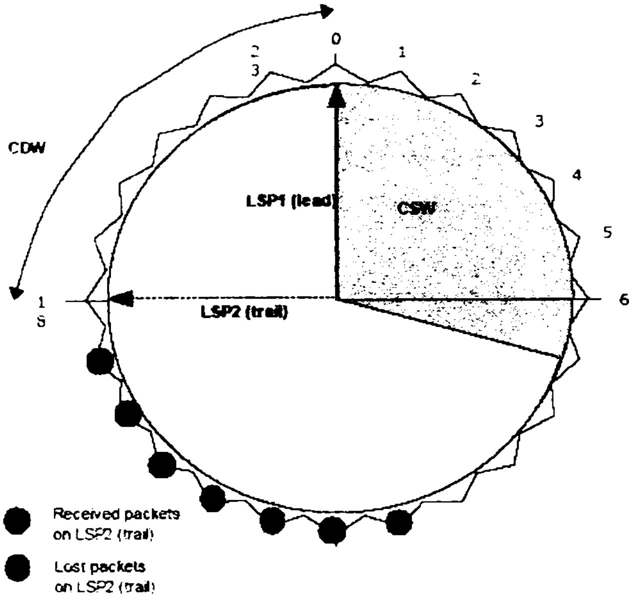
$$CDW(t36) = 6$$

But in this case the condition (4.8) is violated, so the CSW must be increased in order to adapt to the changed conditions:

$$CSW(t48) = 7$$

Recall that the CSW cannot be unlimited extended because it is to be correlated with buffering restrictions on the layer above.

Fig. 29 CSW must be increased because CDW exceeded at t(24+12)


0
1
2
3
4
5
6
18
23
CDW
LSP1 (lead)
CSW
LSP2 (trail)
Received packets on LSP2 (trail)
Lost packets on LSP2 (trail)

4.5. **Contributions**

I proposed a new service quality Q of the packet 1+1 protection, which should reflect the requirements of the application using the LSP pair, and could also possibly be a function of the operators expectations. This quality is defined independently to the range of the sequence numbers N, as the inverse of the tolerance T on loosing packets.

I introduced new LSP quality parameters: *qLSPlead* and *qLSPtrail*. They reflect the quality of the leading and trailing LSP. They are intended to be used to trigger preventive counter measurements possibly autonomously.

I proposed to give up the static configuration of the Sliding Window (originally suggested by ITU-T to be set to ½ * 2 ** N, as a solution for the wrap around problem). Instead:

I introduced the concept of a dynamic *Current Sliding Window,* as an adaptive window, changing its size so that the following essential, functional conditions are always fulfilled:

$$CSW > T \text{ and } CSW > CDW$$

where the Current Delay Window *CDW* also extends the static Delay Window as proposed by the ITU-T.

With my proposal it is possible to increase the local knowledge of the tail node by providing means for an early error detection and avoidance by:

a) monitoring the degradation of the packet 1+1 protection below the agreed value *Q*
b) monitoring the quality of the leading and trailing LSP
c) using the monitored information as trigger to initiate preventive counter measurements e.g. calculating and replacing with alternative LSPs

The proposed mechanism offers significant improvements in terms of reliability and performance compared to the traditional MPLS 1+1 Packet Protection as proposed by ITU-T G.7712.

4.6. **Future Work**

The framework suggested in my „Improved 1+1 Packet Protection" is in particular suitable for ad-hoc, mobile networks with possibly instable nodes. For instance it could be used for efficient routing in wireless networks (comp. [71]), in particular in sensor networks [72], [73], [74], [75].

Ad-hoc wireless networks, which may be built of numerous location-aware sensor nodes, spread in the surrounding environment (such as SmartDust [76]) poses new challenges in terms of reliability and recovery performance. In particular congestion was identified as having a significant impact on the performance by increasing the energy consumption per packet [77]and decreasing the throughput [81], which is also addressed by the multipath routing mechanism.

Congestion detection and avoidance in this context was explored in [77],[78]. Both are proposing some adaptive mechanism for mitigation: "open-loop" where the sending rates towards congested nodes are multiplicative decreased and "closed-loop" where the source requires constant feed back from the destination in order to maintain its rate. In [80] and [79] path diversity is proposed as a mean to avoid congestion. The interaction of congestion control with unreliable transport is analyzed in [98].

Another very promising direction of further research was recently opened in the context of the new **multipath TCP working group**. See [82] for the charter.

The next-generation transport splits the classical transport layer into the following sublayers: Endpoint, Flow Regulation, Isolation, Semantic Layer (comp. [83]). In particular the Semantic Layer is in charge for application-oriented functions serving the endpoints reliability. Its main functionality is to create separate flows over multiple paths, manage end to end states cross these flows, bundle flows for shared congestion control.
MPTCP fits for the semantic layer whereas the classical TCP fits for the end point and flow regulation.

The fundamental shift was initiated with the "Resource pooling principle" in [86]. The central idea is to integrate the existing mechanisms for load balancing and failure resilience into a general concept of resource pooling. The vision is to give the ability to the end systems to spread their load across multiple paths, so that they may react on congestion signals by quickly moving from congested (or possibly failed links) to uncongested links.
The motivation is that the current mechanisms in place like dynamic alternative routing (comp. [91] or load balancing through traffic engineering including the concept of "virtual resource constraint" (comp. [97]) are separate solutions with certain weaknesses like scalability of interdomain routing (as documented in [18], [99], [100]), slowness of failure recovery (convergence time in order of seconds [IEEE-2-3] or even minutes [96]), bad interactions between user and network caused by a possible mismatch between minimal congestion paths and minimal cost paths [88]. This last problem may result in load shifting conflict leading in turn to in substantial degradation of the performance as reported by [102], [87].

The "resource pooling principle" is proposing the multipath routing as solution for the problems mentioned before. Predecessors of this solution have been proposed in [93], [101], [90]. The multipath TCP sets up multiple subflows, which are using window-based congestion control. If a packet is dropped on one subflow it may be resent on an other subflow expecting that the reaction time on timeouts may be improved.
The rational behind is the assumption that the new subflow is less congested compared to the old one and also compared to the single path TCP connection.

The expected benefits are:

1. Fast response to failure or other way caused path change because smooth handover in make-before_break scenarios. Resource pooling is a practical way to achieve resilience at acceptable cost.
2. Fast simple traffic engineering because the network operator may mark congested or expensive links by ECN-mark. Multipath-capable hosts may decide on different ISP based on end-to-end performance.

3. Multihoming without stressing BGP which is particularly interesting for mobile hosts.

One of the major items mentioned by [82] and addressed by [86] is the congestion control, as the critical instrument to improve the network reliability for data exchange over TCP. The proposal suggests that the congestion windows of the subflows should be coupled as described by [94] and [92] with some adaptive mechanisms.

The MPTCP functionality could be supported by a generalizing my improved packet 1+1 protection in following terms:

- Extension to N LSP instead of 2 leading and trailing LSP.
  However this may not necessary since according [86] to some theory [95] suggest it may suffice to give a small amount of choice, e.g. two paths per flow, to achieve resource pooling through load balancing.
- Payload packages are not necessary sent in parallel over all the N LSPs (... but they may! - it is actually an application decision). Since [84] states in the reliability section: "*Regarding retransmissions, it MUST be possible for a packet to be retransmitted on a different subflow to that on which it was originally sent.*" the original packet 1+1 protection may be further used.
- Instead only control packages are sent periodically over the N LSPs. Their purpose is the proactive detection of LSPs running into congestion problems.

These improvements may mitigate this draw back of the doubled data rate as for instance in connection with the bi-casting technique.

The improved packet 1+1 protection proposal fits in the research agenda of the "resource pooling principle": "*We will need new tools for traffic engineering that anticipate how end systems will shift their load.*"
"*There is an issue of when to start additional subflows. TCP's ack clock really needs four packets in flight to be able to fast retransmit, so there is little point in starting subflows <u>for very short transfers</u> – unless resilience is critical in which case multiple paths might be used to send multiple redundant copies.*"
Multipath solution " *forces the applications to increase their jitter buffer so that the overall latency is that of the slowest subflow*".

A very interesting proposal to reduce the congestion effects, to improve fairness and increase throughput was made in [76]. However the solution applies only in wireless networks with location information. The approach consists of a multipath protocol routing, Biased Geographical Routing (BGR), and two alternative congestion control algorithms. The In-Network Packet Scatterer (IPS) overcomes transient congestion by splitting traffic immediately before the congested areas. The End-to-End Packet Scatter (EPS) alleviates long term congestion by splitting the flow at the source and performing rate control.
A generalization of my proposal may be used for providing the feedback of the individual path – see next section.

4.7. Latency Control and Congestion Prevention

For the end to end performance of multipath TCP it is essential to control the relative latency of the different subflows. This would require in principle comparative measurements of a kind of "MTCP ping", which would have to be generated synchronous and to travel over each of the subflows. The classical ping has the drawback of reflecting also the reverse direction. A possible option for the MPTCP is to route the subflow along a dedicated LSP, which are implemented using the well known RSVP and MPLS protocols. The existing mechanism of the MPLS 1+1 packet protection (ITU-T G.7712, 2003) can be used if it is modified as follows:

- Extension to a larger number (N >=2) of LSPs instead of the one leading and one trailing LSP.
- The head node replicates a regular package on a periodical interval and sends the copies with the MPLS shim header including specific sequence number on each subflow.
- The egress node eliminates the replicas and monitors the arrival of the copies on the different subflow. This way it may get information on: relative delay between the subflows, dynamic alterations e.g. caused by emerging congestion, or by packets lost on a subflows. This information is provided as a feed back to the ingress node that may apply the appropriate reaction like redistributing the load on the subflows, creating and/or removing subflows.

The proposed supervision may be performed per subflow and globally (for all subflows). The global supervision of the relative latency on the subflows is by control messages that are to be sent over the existing N subflows, each based on a dedicated LSP. The control message is just a certain regular TCP message including the MPLS shim header with the specific MPLS sequence number as per (ITU-T G.7712, 2003). This TCP message is replicated on every subflow containing the same value of the MPLS sequence number. The control message can be send on a periodical basis, possibly related to some requirements on the service quality and also on demand e.g. on congestion indication. They may optionally include time stamps (sending time), however from the principle they are not necessary. In general the egress node does not need any assumption of the frequency of the control message since its supervision is based on monitoring the relative delay between the one leading LSP and the following N-1 trailing LSPs.

The application at the tail end usually requires a certain maximal latency, an acceptable time interval *Lreq* for getting consecutive packets. If the messages are not received inside this Lreq some counter-measurements are indicated. The reason may be either because they are lost, which are then subject of retransmission or because they are „late", which may be caused by regular latency or congestion on a subflow.

For the supervision a ***Current Latency Window (CLW)*** can be defined as the time difference for the control message (identified by multiple copies with the same MPLS sequence number) between the leading LSP and the last trailing LSP. The system should attempt to keep at any time the *Current Latency* below the required maximal latency, fulfilling the condition:

$$CLW\ (t) < Lreq$$

Notice that role of the last trailing LSP may change in time, being assigned to different subflows, which is irrelevant for this consideration since for the overall end to end performance only the relative delay between the leading and the trailing LSP is of importance.

In order to detect already in advance communication problems leading to an increase of the latency affecting the end to end quality, it is indicated to define a factor *fh* which acts as a threshold value for the ratio CLW(t)/Lreq. If the threshold *fh* (high watermark) is exceeded,

$$CLW(t)/Lreq >= fh$$

the sender should be informed via notification, so it could take appropriate counter-measurements like reducing the load on the trailing subflow, possibly down to 0 (means withdrawing the subflow), by adding a new subflow or by redistributing to the existing subflows. This decision may be influenced by information gathered via supervision on the individual subflows (see below). The relative delays on the different subflows may provide a good basis for possible load redistribution. The value of *fh* is a matter of fine tuning taking also in consideration the size of the congestion windows. Something around 70-80% may be a good starting value.

If the counter-measurements are successful, the CLW(t)/Lreq may decrease below the low watermark *fl*:

$$CLW(t)/Lreq < fl$$

This should also be notified to the sender as a confirmation of the counter-measurements.

Resilience may be additionally improved by the supervision of each of the trailing LSPs. For this a ***Current Delay Window*** specific for the subflow *i* (CDW_i) may be defined by monitoring the difference to the leading LSP. The system should attempt to keep at any time the latency of any LSP below the required maximal latency, fulfilling the condition:

$$CDW_i\,(t) <= CLW\ (t) < Lreq$$

For each subflow/LSP the relative variations of the CDWi should be supervised inside the high/low water marks fh_i, fl_i:

$$fl_i < CDW_i\ (t+1)/CDW_i\,(t) < fh_i$$

If these values are exceeded, then it may be an indication of congestion on the subflow. In this case, the condition should be signaled along the LSP, so that each implied node receives the information and possibly use it for some local counter-measurements.

If the RSVP protocol is used for signaling the sub-flow along the LSP, then the intermediate nodes may be informed using the regular "Resv" message exchanged with the previous hop, for instance by extending the Record Route object.

Besides other local adaption or correction measurements, one possible reaction of an intermediate node could be a detour around the congested, possibly failed node. This local decision may be taken based on monitoring/evaluating the final delay at the tail end against the relative delay (latency) to the neighbors of the intermediate node. Such a counter measurement may improve the situation already in advance, before affecting latency on the tail end, thus preventing end to end quality degradation.

Notice that consecutive measurements of the ratio CDW_i $(t+1)/CDW_i(t)$ results in time series which could be used for predictions of the future behavior.

This proposal is not a substitute for the congestion control but a less 'expensive' complement acting as prevention. Based on the combined supervision information (global and per individual subflow) the escalation steps are: Local correction measurements on the intermediate node (local detour), redistribution of load on existing/new subflows as prevention, retransmissions via the congestion control.

The proposal answers some issues of the research agenda of the "resource pooling principle" by providing the base for traffic engineering tools, which are able to anticipate how end system will shift their load. It may provide some feed back for dimensioning the jitter buffer. The solution is evolutionary, it extends the implemented 1+1 packet protection with some slightly modifications, which are eliminating the bandwidth overhead and some extensions on RSVP signaling.

For future work the thresholds defined above could be analyzed on simulations from the perspective of their dependency with related parameter like buffer size, congestion windows, recovery times. The consequent reactions can be analyzed from the perspective of self-configuration theory and possibly extended towards an adaptive self-healing behavior of the network.

5. Some Recent Proposals for End-to-End Recovery

In addition to the continuous grows in size of our days networks, also their complexity is increasing. In particular this is resulting from the interconnection of different networks built of multiple domains/vendors/suppliers. Consequently new approaches dedicated to the multi-domain context becomes necessary.

This chapter summarizes some of the new IETF proposals (partly not yet standardized) in a systematical manner with particular attention to the aspect of their reliability.

Some parts of the following new approaches may be particularly suitable as applications for my enhancement proposals formulated in the previous chapter. See for the corresponding remarks where this is the case.

5.1. Inter-Domain Label Switching Traffic Engineering

The Inter-Domain Label Switching Traffic Engineering is handled by the RFC 4726, which defines the framework and by the RFC 5151 [43], which specifies the RSVP extension. A domain is a collection of network elements with a common address space and/or path computation responsibility as for example Autonomous Systems (AS), Interior Gateway Protocol (IGP) routing areas, and GMPLS overlay networks.

When interconnecting multiple domains, it is necessary to reconsider the procedure of signaling LSPs over more than one domain. Consequently specific issues on reliability - for instance recovery - are to be considered.

The RFC4726 [46] provides a framework for establishing and controlling MPLS and GMPLS Traffic Engineering (TE) LSPs in multi-domain networks. It describes procedures and protocol extensions to support establishment and maintenance of LSP across domain boundaries.

The RFC 5151 [43] provides additional details on signaling LSPs across domain. In particular some RSVP extensions are provided in order to handle in the specific inter-domain context:

- protection,
- recovery
- re-optimization of LSPs .

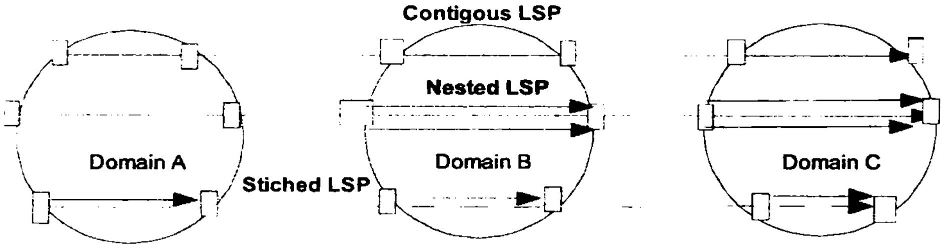
Some RSVP-TE extensions are necessary to control and select one of signaling mechanism to be used for end-to-end inter-domain TE LSP:

- **Contiguous LSP**: A single LSP set up across multiple domains. Same information is maintained along the entire path. In particular the same RSVP session and LSP_ID at is hold on every LSR along the path.

- **Nested LSP** within another LSP. Technique used to nest one/more inter-domain LSP into an hierarchical (H-) LSP. Label stacking is used for this purpose.
- **Stitched LSP**: Technique used to concatenate LSP segments to longer contiguous LSP. Each segment is separately signaled (distinct session). The stitching or H-LSP may be pre-established or dynamically signaled.

An end-to-end inter-domain LSP may be established by using a combination of the LSP types described above. The head end as well as the border node may decide whether to signal the LSP contiguously or to use hierarchical or stiching LSP. These last one may be pre-established or dynamically signaled.

Fig. 30 Signaling methods for multi domains



The border node is responsible for:

- Determine the signaling method to be used to cross the domain. The indication is via LSP_Attributes.Contiguous LSP bit.
- Carry out the ERO procedure: e.g. expanding a path in case of loose route, find existing H-LSP.
- Perform path computation to determine the path across the domain including the exit point. In case of nesting or stitching either find an existing intra-domain LSP or signal a new one.
- Handle LSP Setup Failure and possibly provide cranckback (means further attempts to establish LSP on setup failures) rerouting opportunities.
- Carry out the RRO processing: provide information about hops traversed (e.g. for loop detection). A border node MUST include itself in the RRO.
- Update the Notification Request object in the Path to show its own address. Examine, process and forward Notifications.

The "Parallel Setup" may be a suitable application for the inter-domain because the complex processing described above. An additional argument is the fact that the EROs may contain loose routes, delegating the routing decision routes inside the domain to the border nodes. This is why, border nodes are good candidates for synchronization points as part of the "Sequential Synchronization". The "Parallel Synchronisation" may be considered as well, however in this case the participation of all border nodes is necessary.

Re-optimization of inter-domain LSPs results in moving from current path to a more preferred path. The procedure involves the determination of more preferred path and make-before-break signaling. For contiguous LSP, this must be initiated already at the ingress node. However, this may be the preferred option since it may result in selecting new domain borders. H-LSP and stitching LSP may be re-optimized without impacting end-to-end LSP. A high number of hops involved in re-optimization increase the risk of disruption. On the other hand on short segments the chance of substantial improvement is also low.

The re-optimization process must consider two critical aspects:

- The trigger for the re-optimization should consider only significant changes of relevant disjoint resources (link, node, SRLGs). Otherwise unnecessary 'noise' is generated.
- The path calculation is in general very complex and time consuming. This is in particular true when multiple domains are spawned. Some of the domains may possibly hide information for the reason of topology confidentiality.

Therefore I suggest to consider the usage of the mechanism described in the "Improved 1+1 Packet Protection" as an alternative procedure since it may provide an early indication about better performing alternative LSPs.

5.2. **GMPLS End-2-End Recovery**

GMPLS End-2-End Recovery uses Control Plane (CP) mechanisms (signaling, routing, link management) to support Data Plane recovery. CP mechanism are supported by Data Plane fault detection mechanisms.

The basic requirement is that the end-2-end working and protecting LSPs must be resource-disjoint: link, node, or Shared Risk Link Group (SRLG)

Following End-2-End recovery procedures are supported:

- 1+1 Protection: uni- and bidirectional
- 1:1 Protection with Extra Traffic
- Shared Meshed Restoration
- Full LSP Rerouting
- Pre-emption
- Reversion

In order to support the GMPLS End-2-End recovery it is necessary to introduces some new and modify some of the existing RSVP objects, as follows:

For the LSP Identification:

- IPv4 tunnel endpoint address; Is part of SESSION.
- Tunnel ID: 16-bit ID used in SESSION, constant over tunnel life time.
- Extended Tunnel ID: 32-bit ID used in SESSION, constant over tunnel life time. Normally set to 0; Ingress nodes may place their IP to narrow SESSION scope.
- IPv4 tunnel sender address, used in Path.SENDER_TEMPLATE and Resv.FILTER_SPEC
- LSP ID: 16-bit ID, allows the sender to share resources with itself, used in Path.SENDER_TEMPLATE and Resv.FILTER_SPEC

New Recovery Attributes:

- **LSP Status** determines resource allocation and LSP status via PROTECTION:
    - S (Secondary) bit: distinction of primary (fully established, resource allocation committed at data plane) vs. secondary LSPs
    - P (Protecting) bit: distinction working/protecting LSPs
    - O (Operational) bit: set when protecting LSP carries traffic.

- **LSP Recovery.**
    - PROTECTION.Protection Type in the PROTECTION obj:
        - Full LSP Rerouting: primary working LSP is recoverable by non-pre-planned head-end rerouting
        - Pre-planned LSP Rerouting w/o extra-traffic: pre-reserved recovery resources for one or more ("shared-meshed") on the protecting LSP.
        - LSP Protection with Extra-traffic. Includes 1:N LSP protection
        - Dedicated LSP Protection: 1+1

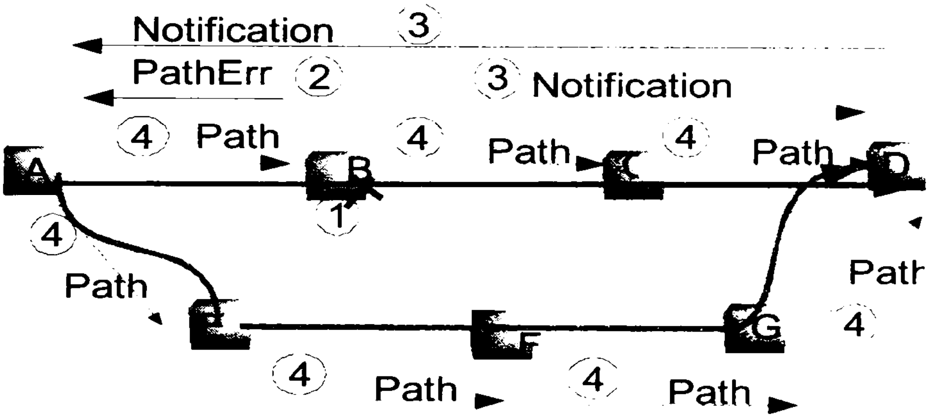
    - PROTECTION.Notification Bit: Data Plane provides automatic protection switching.

- **LSP Association**: ASSOCIATION.Association ID identifies the peer LSP. ASSOCIATION.IP identifies the IP address of the ingress node.

- **PRIMARY_PATH_ROUTE** (PPRO) informs the nodes along the path of a secondary protecting LSP about which resources (Sub-Types: IP, Label, Unnumbered Interface) are used by the associated primary LSP. Carries information extracted from the ERO and/or RRO of primary LSP.

Following figures illustrates some specific recovery procedures.

Fig. 31 1+1 Unidirectional Recovery



For the **1+1 unidirectional protection** 2 LSPs are established between the and nodes A and D. The protected LSP is signaled along A, B, C, D. The protecting LSP is signaled along A, E, F, G, D. The alternate path is resource (node, link, SRLG) disjoint to the protected path. Under normal operational conditions the traffic is sent simultaneously over both LSPs.

The figure above shows the recovery procedure following an abnormal condition (like node failure or degradation of the signal quality) on the working LSP:

1. Node failure on the on the protected path.
2. The end node D detects the failure and switches its receiver to the protecting path.

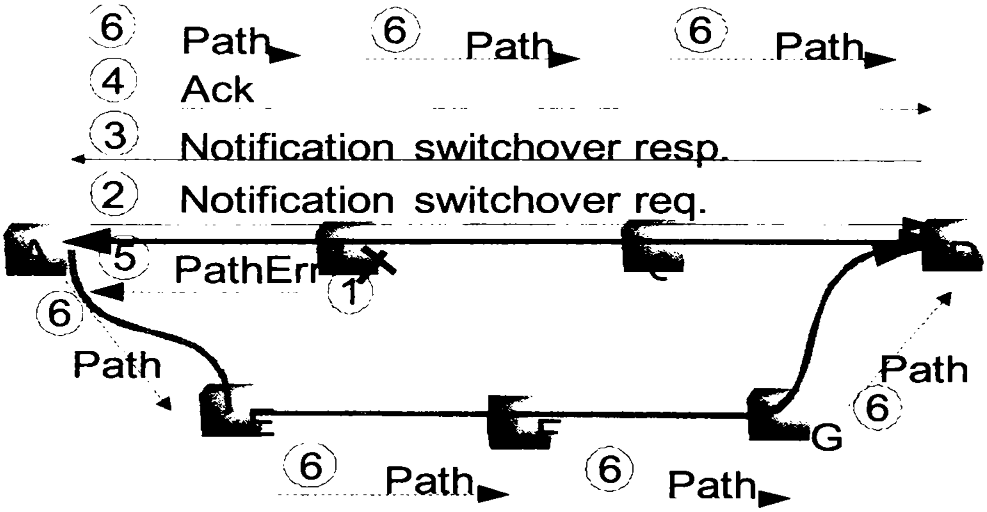
The recovery is completed since for the unidirectional 1+1 protection the protection-switching mechanism acts independently, it must not be coordinated between end nodes. However it is recommended to exchange some additional signaling messages in order to maintain the information about the changed roles of the LSPs (working vs. protecting).

3. Therefore the failure is signalized upstream via PathError message.
4. The node adjacent to the failure C may additionally indicate this by a notification to the to the end node D, which may report the completion of its receiver switching.
5. After receiving the PathErr and the notification (3) the Path message is (re-) sent on both LSP.

It includes:

- The common SESSION object with different LSP Ids.
- The ASSOCIATION object which holds in its Association ID the LSP_ID of the mate LSP (the working LSP holds the protecting LSP_ID and vice versa)
- The new PROTECTION object carrying the LSP Protection type "1+1 Unidirectional"
  - The PROTECTION.S bit is set to 0 on the working and protecting LSP
  - The PROTECTION.P bit is set to 0 on the current working LSP and to 1 on the current protecting LSP.
  - The PROTECTION.O bit is set to 1 on the LSP A, F, G, D
  - The PROTECTION.N bit is set to 1 indicating only coordination on the control plane (and not an protection switching signaling).
- The ADMIN_STATUS.A (administratively down) bit should be set and signaled on the formerly working LSP: A, B, C, D.

Fig. 32 1+1 Bidirectional Recovery


6 Path
6 Path
6 Path
4 Ack
3 Notification switchover resp.
2 Notification switchover req.
5 PathErr
1
6
Path
Path
6
G
6 Path
6 Path

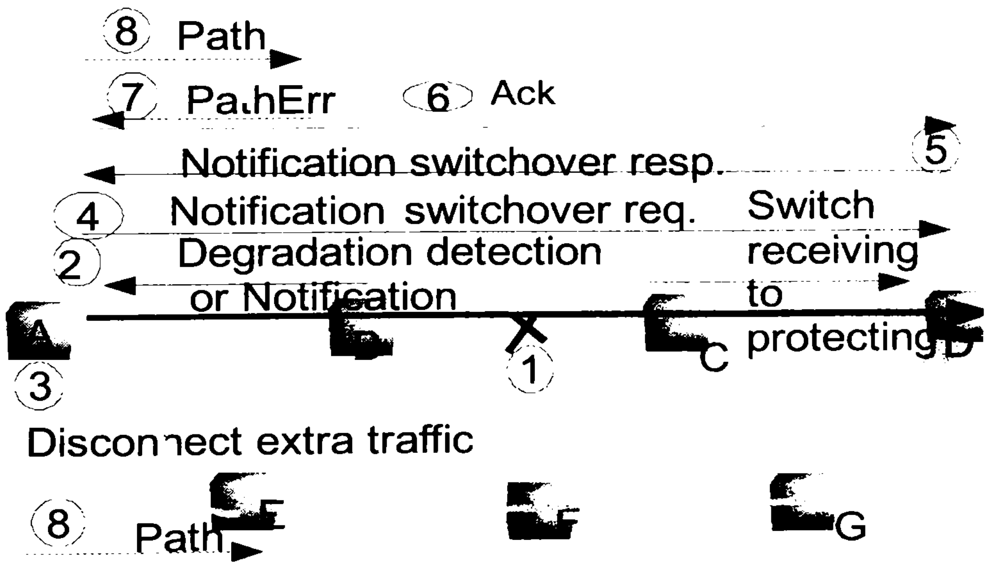
For the **1+1 bidirectional protection** 2 LSPs are established between the and nodes A and D. The protected LSP is signaled along A, B, C, D. The protecting LSP is signaled along A, E, F, G, D. The alternate path is resource (node, link, SRLG) disjoint to the protected path.

Under normal operational conditions the traffic is sent simultaneously over both LSPs. When a failure is detected by one node, both end nodes must select traffic from the protecting LSP. As opposed to the unidirectional protection this action must be coordinated between the end nodes.

The figure above shows the specifics of the recovery procedure following an abnormal condition (like node failure or degradation of the signal quality) on the working LSP.

1. Node failure on the on the protected path.
2. The end node A detects the failure and switches its receiver to the protecting path. In addition it must inform the other node via a Notify message including its SESSION object and a specific error code "Notify Error/LSP Locally Failed" in the (IF_ID)_ERROR_SPEC object. In addition the <sender descriptor> or the <flow descriptor> are present in order to resolve ambiguities or race conditions. This message must be sent reliably, means it includes a MESSAGE_ID object and the ACK desired flag set.
3. Upon receiving the Notify message, the end node D switch the receiving side and confirms completion by sending a Notify message to A. This message must be sent reliably, means it includes a MESSAGE_ID object and the ACK desired flag set.
4. Node A acknowledges the Notify message. The behavior is symmetrical, but for simplicity the perspective of node A was chosen.
5. Node A/D may also get additional informational about the failed link/node by receiving the PathErr/ResvErr
6. After PathErr was received node A must re-signalize the LSP by a Path message the in order to keep track on the changed LSP roles. For the specific settings see the 1+1 unidirectional description.

Fig. 33 1:1 Protection with Extra-Traffic

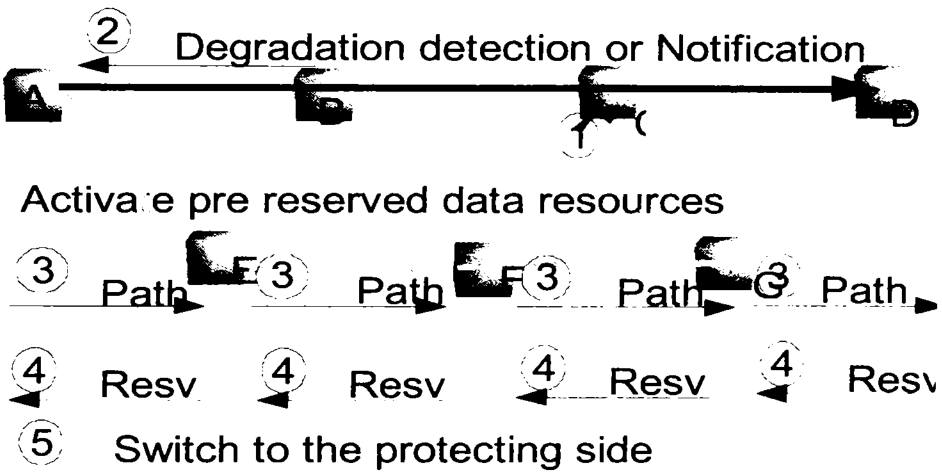

8 Path
7 PathErr
6 Ack
5
Notification switchover resp.
4 Notification switchover req.
Switch receiving to protecting
2 Degradation detection or Notification
A
1
C
D
3
Disconnect extra traffic
8 Path
E
F
G

For the **1:1 protection** 2 LSPs are established between the and nodes A and D. The protected LSP is signaled along A, B, C, D. The protecting LSP is signaled along A, E, F, G, D.

As opposed to the 1+1 protection, the protecting LSP may carry extra traffic, which must be preempted in case of failure on the working LSP in order to allow the pre-allocated resources to become effective.
The figure above shows the specifics of the recovery procedure following an abnormal condition (like node failure or degradation of the signal quality) on the working LSP. The behavior is symmetrical, but for simplicity the perspective of node A was chosen.

1. Node failure on the on the protected path.
2. The end node A detects the failure and switches its receiver to the protecting path. On the setup of the working LSP the end nodes should indicate that they want to receive Notify messages by using the NOTIFY REQUEST message. As a consequence, the Notify message is sent by the nodes adjacent to the failure including its SESSION object and a specific error code "Notify Error/LSP Locally Failed" in the (IF_ID)_ERROR_SPEC object. In addition the <sender descriptor> or the <flow descriptor> are present in order to resolve ambiguities or race conditions.
3. Upon receiving the Notify message, the end nodes must disconnect the extra traffic and switch switch to the protecting path.
4. After switching to the protecting LSP the node A send a Notify message to the node D. This message must be sent reliably, means it includes a MESSAGE_ID object and the ACK desired flag set. These Notify messages are distinguishable from that one generated by the intermediate node.
5. Node D confirms the switching of its receiving side by a reliable Notify message.
6. Node A confirms the Notify response message by an ACK.
7. After PathErr was received node A must re-signalize the LSP by ...
8. Path messages in order to keep track on the changed LSP roles. For the specific settings see the 1+1 unidirectional description.

Fig. 34 End-to-end Rerouting (Restoration) without Extra-Traffic


2
Degradation detection or Notification
A
B
C
D
1
Activa e pre reserved data resources
3
Path
E
3
Path
F
3
Path
G
3
Path
4
Resv
4
Resv
4
Resv
4
Resv
5
Switch to the protecting side

The general case is the 1:N protection, where N working LSPs are protected by 1 protecting LSPs, all mutually resource (node, link, SRLG) disjoint. The Association ID of each of the N working LSP points to the same LSP ID of the protecting LSP.

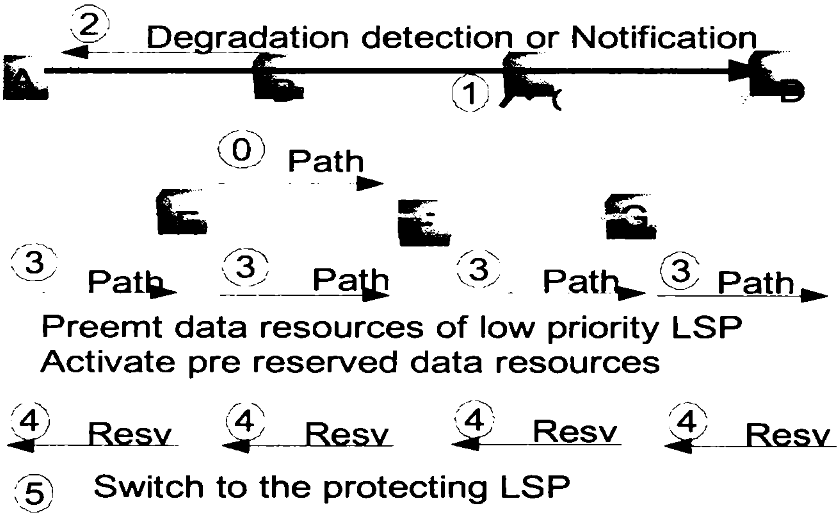
For the **end-to-end (pre-planned) rerouting** 2 LSPs are established between the and nodes A and D. The protected LSP is signaled along A, B, C, D. The protecting LSP along A, E, F, G, D is not fully instantiated means the resources are reserved at the control plane level but not yet committed at the transport plane level. The protecting LSP is signaled with the PROTECTION.S bit is set to 1. As opposed to the protection schemes the rerouting recover requires activation of the protecting LSP. The figure above shows the specifics of the recovery procedure following an abnormal condition (like node failure or degradation of the signal quality) on the working LSP.

1. Node failure on the on the protected path.
2. The head end node A detects the failure by notification from the nodes adjacent to the failure. The head-end node A must indicate its interest for receiving such a notification by using the NOTIFY REQUEST when establishing the LSP.
3. The head end node A activates the pre-reserved resources on the protecting LSP by a regular Path messages, which contains:
   - The common SESSION object with different LSP Ids.
   - The ASSOCIATION object which holds in its Association ID the LSP_ID of the mate LSP (the working LSP holds the protecting LSP_ID and vice versa)

- The new PROTECTION object carrying the LSP Protection type "Rerouting without Extra-Traffic"
  - The PROTECTION.S bit is set to 0, becoming the primary LSP.
  - The PROTECTION.P bit is set to 1
  - The PROTECTION.O bit is set to 1 on the LSP A, F, G, D

4. The head end node A receives consequently the confirmation by the Resv message.
5. ... so that it can switch to protecting side.

Fig. 35 End-to-end Rerouting (Restoration) with Extra-Traffic


2 Degradation detection or Notification
A
1
D
0 Path
3 Path
3 Path
3 Path
3 Path
Preemt data resources of low priority LSP
Activate pre reserved data resources
4 Resv
4 Resv
4 Resv
4 Resv
5 Switch to the protecting LSP

Notice that the activation of the pre-reserved resources on the protecting LSP is time critical, delaying the traffic switch. A fast activation results in a faster switch. Therefore this kind of recovery is a good application for the application of my proposal of the "Parallel Setup" (see chapter 5). This method may be applied on both directions: for the Path message and for the Resv as well.

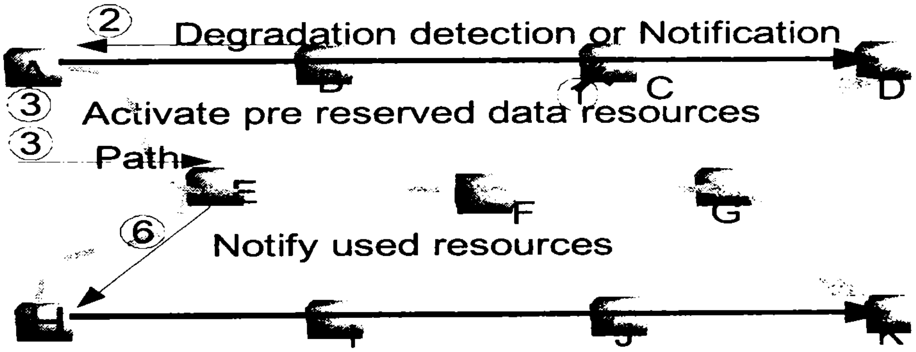
As opposed to the restoration without extra-traffic, the pre-reserved bandwidth on the protecting LSP may be made available for extra-traffic. This is illustrated by the Path message between the nodes E and F - see step (0). Following settings must be done in order to allow pre-emption:

- The SESSION_ATTRIBUTE.Setup Priority must be set to the same value (X) as the Setup Priority used when the protecting LSP was signaled.
- The SESSION_ATTRIBUTE.HoldingPriority must be set to a value greater than the Setup Priority used when the protecting LSP was signaled (at least X+1).

The figure above shows the specifics of the recovery procedure following an abnormal condition (like node failure or degradation of the signal quality) on the working LSP.

1. Node failure on the on the protected path.
2. The head end node A detects the failure by notification from the nodes adjacent to the failure.
3. The head end node A activates the pre-reserved resources on the protecting LSP by a regular Path messages. Same settings as for the rerouting without extra-traffic applies. In addition the extra-traffic between the nodes E end F must be preempted.
4. Head end node A receives consequently the confirmation by the Resv message. Some implementation may chose the upstream activation of the pre-reserved resources (matter of configuration). In this case in order to avoid confusion with the regular refresh Resv, it is indicated to include in the trigger Resv message the PROTECTION object with the S bit set to 0.
5. Head end node A can switch to protecting side.

Fig. 36 Shared-Meshed Restoration


2
Degradation detection or Notification
B
C
D
3
Activate pre reserved data resources
3
Path
E
F
G
6
Notify used resources
H
I
J
K

Notice that the activation of the pre-reserved resources on the protecting LSP is time critical, delaying the traffic switch. A fast activation results in a faster switch.

Therefore this kind of recovery is a good application for the application of my proposal of the Parallel Setup (see chapter 5). The parallel setup may be applied on both directions: for the Path message and for the Resv as well. The benefit is better than in the previous case without extra-traffic, since the preemption and re-allocation requires additional time which may be saved by the parallelization of the setup.

As opposed to the end-to-end restoration, the **shared-meshed restoration** allows more than one working LSP using disjoint resources to share the resources of the protection LSP. For this purpose the nodes E, F, G must be aware of the resources shared between the 2 protecting LSPs: A-E-F-G-D and H-E-F-G-K. This is indicated by the PROTECTION.LSP Protection Type to "Rerouting without Extra-Traffic". This way the common nodes of the protecting LSP may:

- verify if the resources of the primary, working LSP are disjoint
- inform - in case of failures on one primary LSP - the "other" working LSPs about resources which are allocated and thus not available any more.

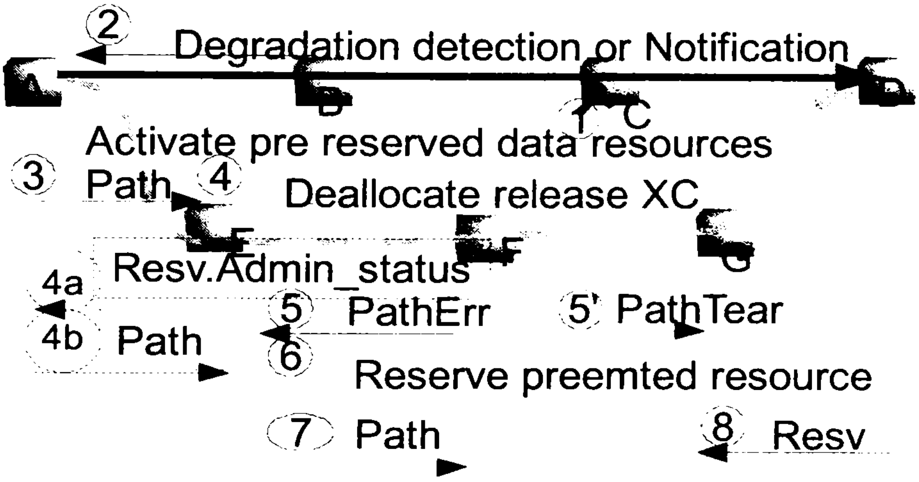
The figure above shows the specifics of the recovery procedure following an abnormal condition (like node failure or degradation of the signal quality) on the working LSP.

1. Node failure on the on the protected path.
2. The head end node A detects the failure by notification from the nodes adjacent to the failure.
3. The head end node A activates the pre-reserved resources on the protecting LSP by a regular Path messages. Same settings as for the end-to-end restoration applies. In addition a PRIMARY_PATH_ROUTE object (PPRO) is included in order to allow resource sharing on recovery. The PPRO object contains a list of sub-objects (IP address, link ID, Labels) derived from ERO/RRO of the primary LSP.
4. Head end node A receives consequently the confirmation by the Resv message.
5. Head end node A can switch to protecting side. In addition to these steps common with the end-to-end restoration.
6. ... the intermediate node E notifies the head end of the "other" working LSP that resources of the protecting LSP are no longer available.

Notice that the activation of the pre-reserved resources on the protecting LSP is time critical, delaying the traffic switch. A fast activation results in a faster switch.

Therefore this kind of recovery is a good application for the application of my proposal of the "Parallel Setup" (see chapter 5). The parallel setup may be applied on both directions: for the Path message and for the Resv as well. An additional benefit comes from the fact that the notification on step (6) may be sent earlier.

Fig. 37 LSP Preemption


2
Degradation detection or Notification
C
Activate pre reserved data resources
3
Path
4
Deallocate release XC
4a
Resv.Admin_status
4b
Path
5
PathErr
5'
PathTear
6
Reserve preemted resource
7
Path
8
Resv

The figure above shows the specifics of the **LSP preemption** procedure following an abnormal condition (like node failure or degradation of the signal quality) on the working LSP.

1. Node failure on the on the protected path.
2. Head end node A detects the failure by notification from the nodes adjacent to the failure.
3. Node A sends Path message along the secondary LSP with the purpose of preempting the lower priority transport resources,
4. The first intermediate node (node E) on the secondary LSP receives the Path message and preempts the lower priority resource, means deletes the corresponding cross connects. If alarm suppression is expected, then a Resv message including the ADMIN_STATUS object is to be sent upstream which has to be confirmed by receiving the corresponding Path message (4a, 4b). In order to avoid misconnections it is recommended for the preempting node to forward the Path message only after de-allocation has completed. This is to avoid the situation of a downstream node reassigning the resources more quickly then the preempting node.

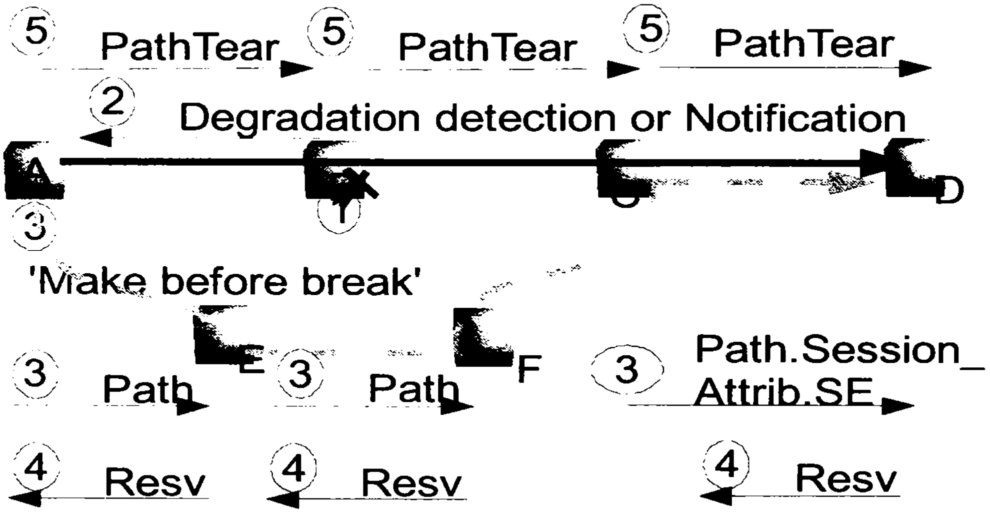
   Therefore the optimization procedure of "Parallel Setup" is not suitable.
5. After preemption the intermediate node on the secondary LSP sends a PathErr message upstream and a PathTear downstream with the error code "Hard preempted" and with the Path_State_Removed flag set. Notice that the consequent preemption on the following nodes are triggered by the PathTear message which is forwarded along the preempting LSP.

The "Parallel Setup" procedure would be suitable and is indicated because the time consuming interaction with the transport (data) resources.

6. The intermediate node reserves the preempted resource for the protecting LSP. It can be cross connected only for the an unidirectional LSP. Otherwise (for the bidirectional case) mis connections may occur.
7. Forward the Path message to the next node on the preempting LSP.
8. The intermediate node receives the trigger Resv message, cross-connects the downstream resource and the upstream resource (in case of a bidirectional LSP).

Since the resources are already reserved the optimization procedure of "Parallel Setup" for the Resv message is suitable and indicated since the preempting procedure is anyway time consuming.

Fig. 38 (Full) LSP Rerouting



As opposed to the end-to-end restoration, the **full LSP rerouting** switches the normal traffic to an alternate path which may reuse intermediate nodes included in the original route. This is illustrated in the figure above by the node C.

The rerouting procedure may be initiated by the head end node A when it detects an abnormal condition on the working LSP. The alternate route may be computed:

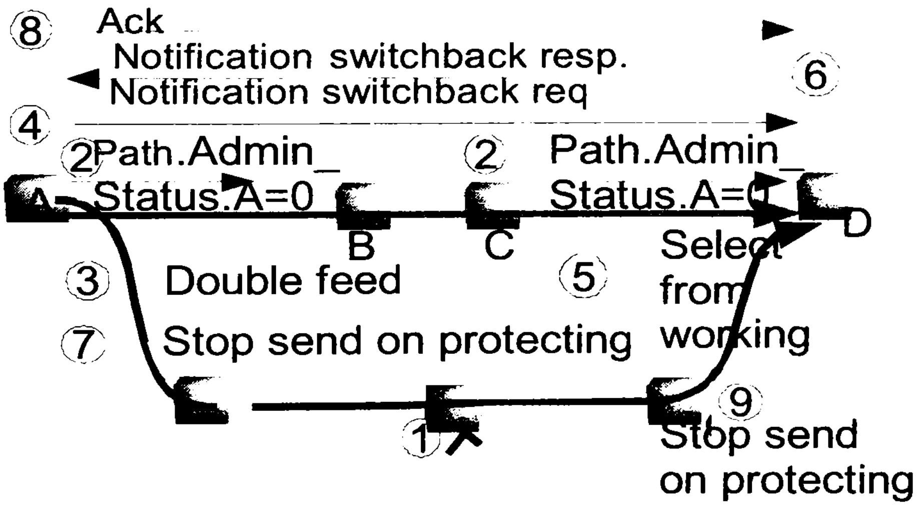
- on demand (when the abnormal condition occurs). This is the case for *Full LSP Rerouting.*
- pre-computed and stored to be used when necessary. The risk of outdated information may be mitigated by periodic re-calculations.

The procedure is as follows:

1. An intermediate node fails.
2. The head end node A is informed about the abnormal condition.
3. The head end node A re-calculates the alternate route (if not already available). For the recalculation route exclusion and cranckback signaling techniques may be used.
4. For signaling on the alternate LSP the so called "make-before-break" mechanism is used. The indication on reuse of resources on intermediate nodes is signalized via SESSION_ATTRIBUTE object and Shared-Explicit reservation style (SE)
5. Head end node A receives consequently the confirmation by the Resv message.
6. The old LSP is torn down.

The RFC 4872 allows as an option the setup of alternate LSP (different LSP ID) before the occurrence of an failure. Again the Shared-Explicit reservation style (SE) is to be used.

Fig. 39 Reversion for 1+1 Bidirectional Protection



Notice that from the principle the procedure described before could be also triggered by the tail end node which detects the quality degradation on the primary LSP via control messages, possibly generated as result of the procedure described in the chapter.

The basic idea of the **reversion** is a minimal service disruption and reconfiguration. As a pre-condition it is of course necessary that the resources of the primary working LSP remain allocated after the failure.

The procedure is as follows:

1. On reversion the normal traffic returns to the initial working LSP when it recovered from the failure. Alternatively the switch back may be triggered on a failure on the protecting LSP as described in the figure above.
2. The head end node A sends a Path message along the recovered LSP with the ADMIN_STATUS.A bit cleared.
3. The source node A feeds normal traffic onto both working and protecting LSPs.
4. The head end node A sends a reliable Notify message (MESSAGE_ID, ACK desired) indicating the Switchback Request.
5. The destination node D starts selecting from the working LSP and transmits onto both working and protecting LSP.
6. The tail end D sends a Notify messages indicating the Switchback Response
7. Upon reception of the Switchback Response, the source node A stops sending traffic on the protecting LSP...
8. ... and confirms by an ACK message addressed to the tail node D ...
9. ... which also stops sending on the protecting LSP. Recall that the case of bidirectional traffic is described here.

5.3. **GMPLS Segment Recovery**

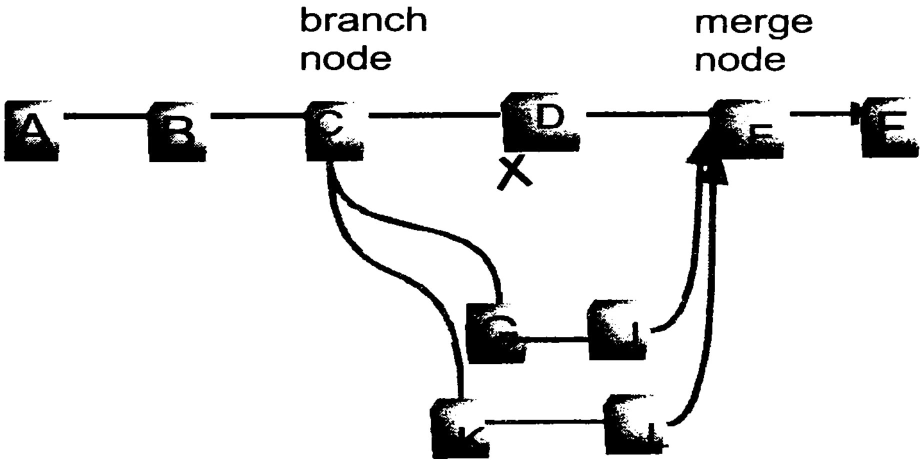
Segment recovery provides protection, restoration, re-routing, over segments as portions of an end-2-end LSP. For this purpose the corresponding RFC 4873 provides following extensions:

Signaling of desired LSP segment protection type:

- Identification of begin/end of segment protection
- Identification of used hops
- Reporting of path used for LSP protection

Segment recovery is positioned between the end-to-end recovery (see chapter before) and the fast reroute mechanism: one-to-one backup (for each protected LSP at each potential point of local repair) and facility backup (bypass tunnel which protects by label stacking a potential failure point shared by multiple LSPs).

Fig. 40 Segment Topology



The figure above describes a topology for which the end-to-end recovery is not possible because no alternate LSP with disjoint resources may be established between the end nodes A and F. However for a portion of the primary LSP namely the segment C-D-E two alternate LSPs may be established: C-G-I-E and C-K-L-E, so they could be used for either protecting or rerouting recovery.

The segment recovery LSPs are signaled as independent LSPs between the branch node (its IP address is used for the Sender_Template) and the merge node (its IP address is used for the Session object). Since they are independent, the two segment LSPs may be also subject of end-to-end recovery.

The end nodes of the protected segment play a significant role in the recovery schemes: Node C which initiates the recovery LSP is named *branch node* whereas node E which terminates the recovery LSP is named *merge node*.

The head end node of a protected LSP can:

- delegate the (dynamically) identification of branch/merge node to the downstream nodes by using the LSP segment protection bits in the PROTECTION object.
- identify itself the endpoints of the segment LSP by using a Secondary Explicit Route Object (SERO). See below for details.

SERO objects have a format similar to the Explicit Routing Objects (ERO) indicating:

- the initiator of the recovery LSP (branch node)
- protection/restoration type
- the terminator of the recovery LSP (merge node)

SERO are typically carried between the head end node and the branch node and handled as follows:

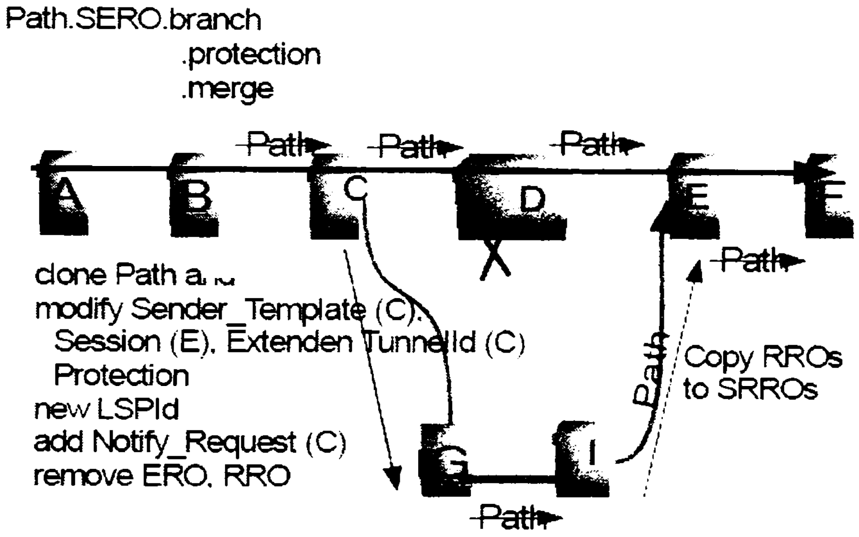
The branch node clones the received Path message and:

- modifies the Sender_Template and the Extended TunnelID inserting its own IP address.
- modifies the Session object inserting the IP address of the merge node.
- uses the SERO to build the ERO for the recovery LSP (G-H).
- removes the original ERO and RRO since they are significant only for the protecting LSP.
- creates a new ERO based on the content of the received SERO
- adds a NOTIFY_REQUEST objects with its own address.

The merge node copies its Record Route Object (RRO) filled along the recovery LSP to a new SRRO, which is sent along the protected LSP towards the tail end (see next figure). In addition the merge node has to add a NOTIFY_REQUEST objects with its own address.
The branch and the merge node should also add a NOTIFY_REQUEST object with its own address when processing the Path/Resv message for the protected LSP. The NOTIFY_REQUEST object must removed by the corresponding merge/path node, means that the NOTIFY_REQUEST object are relevant only for the protected segment (C-D-E)

Secondary Record Route Objects (SRRO) are carried by the Path message between the merge node and the tail end indicating the presence of upstream recovery LSPs. The merge node clones the Resv message received from downstream and adds a NOTIFY_REQUEST object with its own IP address.

Fig. 41 Explicit Control of LSP Segment Recovery: Path message processing



SRROs are also used in the Resv message between the branch node and the head end, indicating the presence of downs recovery LSPs. They are inserted into the Resv message by the branch node. See next figure. A SRRO contains:

- local node address of the merge node
- protection subobject containing the PROTECTION object of the recovery LSP
- a copy of the RRO as received in the Resv message on the recovery LSP

When Resv messages are merged, the resulting Resv message should contain all SRROs received. The branch node receives Resv messages from both protected and recovery LSP but they must be propagated upstream only after the Resv message for protected LSP is received.

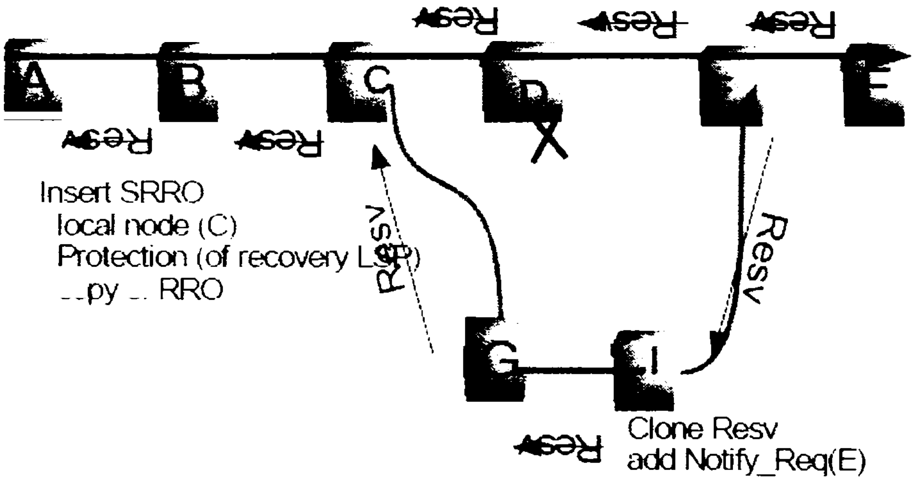
Notice that the procedure described above requires synchronization by its nature. Therefore I suggest to consider the usage of the proposed mechanism of "Parallel Setup". In this case, not the LSP end nodes but the branch/merge nodes will have to do the final synchronization. The expected benefit is limited to the nodes of the protecting and recovering LSP.

For the *Dynamic Control* of LSP Recovery in principle the same procedure applies as for the *Explicit Control* of LSP Recovery. The main difference is the dynamic creation of the ERO for the recovery LSP.

When a node dynamically identifies itself as a branch node and identifies the merge node for the type of recovery indicated in the LSP Segment Recovery Flags, it attempts to setup a recovery LSP.

Unlike with explicit control, if the creation of a dynamically identified recovery LSP fails, it is removed and node error indication is sent upstream.

Fig. 42 Explicit Control of LSP Segment Recovery: Resv message



The following summarizes the new and modified objects which are used for the recovery of LSP Segments:

- **Association.type Resource = Sharing** for make-before-break.
- **Secondary Explicit Route** (SEROs) indicate branch and merge nodes of recovery LSPs. Format as EROs.
- **Protection subobject** used in SEROs for the recovery LSP. At least 3 subobjects: segment branch node, protection are to be provided for LSP, [optional EROs, loose or strict], merge node.
- **Secondary Record Route** (SRROs) used to record the path of recovery LSP. Format as RROs.
- **Modified Protection** obj:
  - **In-Place bit**: indicates that the desired segment recovery indicated in Segment Recovery Flags is already in place.
  - **Required:** indicates that a failure to establish the required protection should result in a failure of the protected LSP.
  - **Segment Recovery Flags**: used to indicate when an upstream node desires LSP segment recovery to be dynamically initiated.
- Additional **NOTIFICATION_REQUEST** to identify recovery LSP branch node (per RFC3473 only one) and merge node.

6. Conclusions - Personal Contributions – Future directions

Modern networks are rapidly increasing in terms of size, transported data, required quality of offered services and complexity.

Therefore high performance is expected for both: on the normal operational mode as well as in the recovery case. In general it is required to minimize the impact of failures by offering a correspondent high network reliability. In particular it is required that the network as a whole should be able to guaranty a certain availability, serviceability. This may be offered by autonomous recovery actions in the network.

As a consequence, the network resilience and the related categories (recovery, reliability, survivability) continue to be a very interesting subject for the research community. Compare for instance the recent FP7 EU project ResumeNet ([58]).

My contributions focused on improving the control plane resilience, as the ability to recover from faults and to provide uninterrupted service. Improved resilience and a good performance of network recovery are key pre-requisite for deployment on a wide scale of ASON networks.
The research context was the optical networking based on the ASON and GMPLS architecture, however the results may apply also to other networks e.g. wireless.

Summary of the theoretical contributions:

1. Previous research on shared protection focused on schemes, which minimize the use of real-time message exchange between network elements [13]-[18]. Some proposals even do restoration without message protocols at all, but have to compromise on some of the carrier-grade requirements. While this direction was certainly exciting, I took the challenge to evaluate other options for fast and efficient message based signaling between real NEs, motivated by the fact that also BLSR/MS-SPRING protection uses some message based protocol over the K bytes.

   The outcome of the investigation (published as [5]) was that the control hardware of the current generation of network elements is able to provide acceptable restoration performance. It could be predicted that for larger networks the restoration time can be kept under the 50ms benchmark [48]. The conclusion was that fast meshed restoration can be efficiently implemented, with acceptable performance even if the related signal protocols have a comparatively large overhead.

   However I identified some opportunities for improving the signaling performance of the control plane - see items 4 and 5.

2. Because the heterogeneity of the available information on network recovery, reliability and resilience I had to summarize the exiting approaches in a systematical manner, as a *theoretical preparation.* Since there was not too much info available at that time, I started to analyze the suitability of RSVP for the ASON signaling procedure in "Stadiul actual si de perspectiva in sistemele moderne de comunicatii. Protocoale de rezervare a resurselor"[6].

3. In my second presentation: "Optimizari in sistemele moderne de comunicatii. Recuperarea si fiabilitatea retelelor" [7], I focused on the reliability and resilience aspects of signaling protocols in the control plane context.
   The next step was the optimization of the signaling protocol. I found two directions suitable to improve the resilience of the control plane – see following items:

4. Improving the performance of path setup. This direction is motivated by the performance penalties expected for network restoration as mentioned in the literature [12]. The performance for the setup of Label Switched Path (LSP) is in general poor because it is dependent on the path length. This affects not only regular setup but also the recovery performance being a serious handicap for the restoration schemes. Therefore I elaborated the method for Fast Source Routed Connection Setup, which applies to regular setup and also to the restoration procedure. The basic idea is to improve the signaling by a optimized path setup algorithms, which is using the inherent existing parallel processing capacity. Recall that the path setup is a time critical activity necessary for the activation of a (possibly pre-calculated) connection restoration plan. The Method for Fast Source Routed Connection Setup [11] describes the establishment of a connection between a head end node and an tail end node in which parallel transmission and processing of connection request messages is used to minimize the dependency of connection setup time on connection path length. For the parallel connection setup I'm proposing two alternative of synchronization:
   a) sequential synchronization on the intermediate nodes
   b) final synchronization in the end node

The algorithm is protected as U.S. patent application: **Method for Fast Source Routed Connection Setup, Inventor: Florin-Josef Lataretu, Pub. No.: US 2006/0034288 A1**

5. A relative good reliability of signaling between the head- and the tail-end is given by applying the MPLS 1+1 Packet Protection mechanism. However additional improvements may be obtained by the early detection of quality degradations for the scope of preventive measurements. I elaborated the method for Improved 1+1 Packet Protection motivated by the fact that the most effective recovery is the prevention and responding to the question "How to a avoid expensive and time critical recovery actions by preventive measurements?". With the "Method for Improved 1+1 Packet Protection" [10] I proposed a mechanism which extends the traditional MPLS 1+1 Packet Protection as described in ITU-TG.7712 by integrating specific application needs and taking into account the dynamics introduced by the supporting LSP.

With my proposal it is possible to increase the local autonomy of the tail node by providing means for early error detection and avoidance by:

   a) monitoring the degradation of the packet 1+1 protection below the agreed value Q;

b) monitoring the quality of the leading and trailing LSP;

c) using the monitored information as trigger to initiate preventive counter measurements e.g. calculating and replacing with alternative LSPs.

The proposed mechanism offers significant improvements in terms of reliability and performance compared to the traditional MPLS 1+1 Packet Protection as proposed by ITU-T G.7712. It can be used in any network which supports MPLS 1+1 Packet Protection. It is in particular recommended for time critical applications.

This method is protected as U.S. patent **Method for improved packet 1+1 protection, Inventor: Florin-Josef Lataretu, Patent No.: US 7,525,903 B2, April 2009.**

6. For the "Method for improved packet 1+1 protection", I found [8] in the context of the the recent Segment and End to End Recovery proposals ([19], [20]) following suitable candidates:

   a) The end-to-end LSP rerouting, since from the principle, the procedure described by the RFC 4872 and RFC 4873 could be also triggered by the tail end node, which detects the quality degradation on the primary LSP via control messages sent in this context.

   b) The LSP re-optimization process, which must consider some critical aspects: significant changes of relevant disjoint resources, path calculation is in general very complex and time consuming, high number of hops involved in re-optimization increase the risk of disruption.
   Therefore I suggested considering the usage of the mechanism described in the "Improved 1+1 Packet Protection" as an alternative procedure since it may provide an early indication about better performing alternative LSPs.

7. I found following suitable candidates [8] for applying the Fast Source Routed Connection Setup ("Parallel Setup") in the context of the recent Segment and End to End Recovery proposals ([19], [20]) :

   a) For the inter-domain traffic engineering the "Parallel Setup" may be a suitable application because the related complex, time-consuming processing in each node. An additional argument is the fact that the EROs may contain loose routes, which implies anyway the delegation of the routing decision routes inside the domain to the border nodes. This is why border nodes are good candidates for the synchronization.

   b) The end-to-end restoration is in general a good application for the "Parallel Setup", which may be applied on both directions: for the Path message and for the Resv message as well. Notice that the activation of the pre-reserved resources on the protecting LSP is time critical, delaying the traffic switch. A fast activation results in a faster switch. The benefit is significantly better in the case with extra-traffic, since the preemption and re-allocation requires additional time which may be saved by the parallelization of the setup. For the case of shared-meshed restoration an additional benefit comes from the fact that the notification informing about the current usage of the shared resources may be sent earlier compared with the regular setup.

c) The end-to-end LSP preemption: Since the resources are already reserved the optimization procedure of "Parallel Setup" for the Resv message is suitable and indicated because the preempting procedure is in general time consuming.

d) The segment recovery requires synchronization in the branch/merge node by its nature. Therefore I suggest to use the proposed mechanism of "Parallel Setup". In this case not the LSP end nodes but the branch/merge nodes will be have to do the final synchronization The expected benefit is limited to the nodes of the protecting and recovering LSP.

Summary of the applications and practical contributions:

1. My practical activities started with some simulations of a signalling protocol based on IP/UDP messages, followed by detailed network measurements. The results have been summarized in a lab report [5]. In this report I analyzed the performance of the signaling in a meshed SDH/SONET network. We found out that restoration time of 11ms and less can be achieved in a small network of 5 real network elements. From the scaling behavior it could be predicted that for larger networks the restoration time can be kept under the 50ms benchmark [48].

2. In order to validate the estimated the benefit of the "Method for Fast Source Routed Connection Setup" I performed simulations based on the ns-2 network simulator. The results confirmed the direct advantage of the parallel setup is the **increased performance** of the setup time. This is in particular of benefit for path setups which are time critical for instance in the context of Path Restoration. The implicit positive effect is the **increased reliability** because the improved performance on network recovery and better scalability. A drawback is some decreased scalability because the additional number of synchronization messages. However this deficiency may be attenuated if the end nodes and intermediate nodes are well meshed. The benefit is a considerably reduced setup time for the positive case as well as for the negative case since resource conflicts can be detected faster.

3. Two years after the registration of the "Method for Fast Source Routed Connection Setup", a similar proposal was made under the title "A Fast and Efficient Segmented Signalling Protocol for GMPLS/WDM Optical Networks" [63]). Here a summary of of the specific differences between these proposals:

- That proposal applies on WDM Networks whereas my proposal may apply to any kind of GMPLS signaling (in particular WDM, TDM) which provides source routing.
- That optimization addresses segments while my optimization addresses any intermediate nodes for the parallelization. Therefore there is a additional potential for efficiency gain for the setup time.
- That proposal suggests the modification of the RSVP protocol by defining some new control message types (RESV_INFO, RESV_SUCCESS, FAIL_INFO) while my implementation proposal is based on the existing control message.
- That proposal indicates parallelization only for the Resv message while my proposal may also include the Path message.
- In addition my proposal offers two different variants for sequential and for final synchronization which may be choose depending on the concrete context.

Therefore, I proposed an alternative implementation of my abstract method based on the RSVP protocol (to be published as [9b]). This implementation is evolutionary and backward compatible, so it can be applied also to networks containing nodes handling traditional and enhanced RSVP setup messages. Compared with the similar proposal, it is more generic and avoids the introduction of new RSVP messages. The preferred option for extending the semantic of the RSVP message is by introduction of a new ADMIN_STATUS bit. However also an implicit indication of the new semantic would be possible.

The ingress node may decide on any of these alternative procedures, since they do not require a divergent behavior in any of the implied nodes. In general the Final Synchronization can be adopted if there are no dependencies from the previous hop (e.g. Path for unidirectional setup). Instead, the Sequential Synchronization may be adopted when there are dependencies from the previous hop e.g. Resv.Label). One drawback is some increased complexity of the RSVP state machine. However it should be noticed that the modification is introducing a kind of a two phased transaction (preparation phase followed by the commit or a roll-back phase), which is in general a useful pattern.

4. Related to the "Method for improved packet 1+1 protection" I introduced and verified by tests the adequate parameters (see below) to measure the quality of the LSP, to detect degradation in order to initiate counter measurements.
   a) I proposed a new service quality *Q* of the packet 1+1 protection, which should reflect the requirements of the application using the LSP pair, and also possibly could be a function of the operator's expectations. This quality is defined independently to the range of the sequence numbers *N*, as a function of the tolerance on loosing or acting with delayed packets.
   b) I introduced new LSP quality parameters. They are intended to be used to trigger preventive counter measurements possibly autonomously.
   c) I introduced the concept of a dynamic *Current Sliding Window*, as an adaptive window, changing its size so that the related functional conditions are always fulfilled. The *Current Sliding Window* has a direct practical relevance since it must be aligned with the memory resources of the network element.

5. Related to the "Method for improved packet 1+1 protection" I proposed solution for improved resilience of a multipath TCP connection by supervising the latencies on the different subflows (published as [9]). Some specific thresholds are defined to control maximal delay or jitter. If they are exceeded, the head end may initiate appropriate reactions based on the information provided by supervision. This proposal is not a substitute for the congestion control but a less 'expensive' complement acting as prevention. Based on the combined supervision information (global and per individual subflow) the escalation steps are: Local correction measurements on the intermediate node (local detour), redistribution of load on existing/new subflows as prevention, retransmissions via the congestion control The proposal answers some issues of the research agenda of the "resource pooling principle" by providing the base for traffic engineering tools, which are able to anticipate how end system will shift their load. It may provide some feed back for dimensioning the jitter buffer. The solution is evolutionary, it extends the implemented 1+1 packet protection with some slightly modifications, which are eliminating the bandwidth overhead and some extensions on RSVP signaling.

For future work the related thresholds could be analyzed on simulations from the perspective of their dependency with related parameter like buffer size, congestion windows, recovery times. The consequent reactions can be analyzed from the perspective of self-configuration theory and possibly extended towards an adaptive self-healing behavior of the network.

Future work may also study the applicability of the "Parallel Setup" method in heterogeneous, wireless networks.

The quality parameters elaborated for the "Improved 1+1 Packet Protection" and the related procedure could be analyzed under the aspects of self-configuration theory and possibly extended towards an adaptive self-healing behavior of the network.

7. References

[1] ITU-T Rec. G.8080/Y.1304, "Architecture for the Automatically Switched Optical Network (ASON)", ITU-T Standardization Organization, November 2001

[2] OIF, "OIF-G-Sig-IW-01.0 - OIF Guideline Document: Signaling Protocol Interworking of ASON/GMPLS Network Domains ", Optical Internetworking Forum, June 2008

[3] OIF, Contribution to Architecture Signaling Working Group, SCN Design Guide, Internetworking Forum, March 2005

[4] OIF, "OIF-E-NNI-Sig-01.0 - Intra-Carrier E-NNI Signaling Specification", Optical Internetworking Forum, February 2004

[5] F. Lataretu, W. Rothkegel, D. Stoll, "Efficient Signalling for Fast Restoration in Meshed SONET/SDH Networks: a Lab-Report", Photonische Netze, Vorträge der 5. ITG-Fachtagung vom 3. bis 4. Mai 2004 in Leipzig, ISBN 3-8007-2826-5

[6] F. Lataretu, "Stadiul actual si de perspectiva in sistemele moderne de comunicatii. Protocoale de rezervare a resurselor", in Referatul de doctorat nr.1, Universitatea "Politechnica" Timisoara, Decembrie 2006

[7] F. Lataretu, "Optimizari in sistemele moderne de comunicatii. Recuperarea si fiabilitatea retelelor", in Referatul de doctorat nr. 2, Universitatea "Politechnica" Timisoara, Mai 2008

[8] F. Lataretu, "Simulari si rezultate privind optimizarile in sistemele moderne de comunicatii. Recuperarea si fiabilitatea retelelor", in Referatul de doctorat nr. 3, Universitatea "Politechnica" Timisoara, September 2008

[9] F. Lataretu, C. Toma, "Improving the Resilience of Multipath TCP by Latency Supervision", IADIS International Conference "Applied Computing 2010", Timisoara, Romania, 16-18 October 2010, pp. 281-283, ISBN 978-972-8939-30-4

[9b] F. Lataretu, C. Toma, "Fast Source Routed Connection Setup- Proposal for a RSVP Implementation", IEEE TELFOR, November 2010

[10] F. Lataretu, "Method for improved packet 1+1 protection", Patent No.: US 7,525,903 B2, April 2009

[11] F. Lataretu, "Method for Fast Source Routed Connection Setup", Inventor: , United States Patent Application 20060034288, Pub. No.: US 2006/0034288 A1
http://www.freepatentsonline.com/y2006/0034288.html

[12] Vasseur, Pickavet, Demeester, "Network Recovery; Protection and Restoration of Optical, Sonet-SDH, IP, and MPLS"; Morgan Kaufmann, 2004

[13] W.D. Grover, D. Stamatelakis, "Cycle-oriented distribute preconfiguration: ring like speed with mesh-like capacity for self-planning network restoration" IEEE International Conference on Communications, 1 (1998), 537-543

[14] W.D. Grover, D. Stamatelakis, et.al. "New options and insights for survivable transport networks" IEEE Communications Magazine, (2002), p 34-41

[15] W.D. Grover, D. Stamatelakis; "Theoretical underpinnings for the efficiency of restorable networks using preconfigured cycles ("p-cycles")" IEEE Transactions on Communications, (2000), p 1262 - 1265

[16] Stamatelakis, D.; Grover, W.D.; "IP layer restoration and network planning based on virtual protection cycles" ; IEEE Journal on selected areas in Communications, vol. 18, issue 10, (2002) p 1938 - 1949

[17] D.A. Schupke, "Fast and efficient WDM network protection using p-cycles", IEEE/LEOS Summer Topics inOptical Transmission/VCSEL an Microcavity Lasers, (2002) p 47

[18] D. Larrabeiti, et.al "Multi-domain Issues of Resilience" Proceedings of International Conference Transparent Optical Networks, vol.1 p 375-380, 2005

[19] J.P. Lang, Y. Rekhter, D. Papadimitriou, "RSVP-TE Extensions in Support of End-to-End GMPLS Recovery - Generalized Multi-Protocol Label Switching (GMPLS) Recovery", RFC 4872, May 2007

[20] L. Berger, I. Bryskin, D. Papadimitriou, A. Farrel, "GMPLS Segment Recovery", RFC 4873, May 2007

[21] Demeester, P.; Gryseels, M.; van Doorselaere, K.; Autenrieth, A.; Brianza, C.; Signorelli, G.; Clemente, R.; Ravera, M.; Jajszcyk, A.; Geyssens, A.; Harada, Y.; Network resilience strategies in SDH/WDM multilayer networks, 24th European Conference on Optical Communication, Volume: 1, 1998, Page(s): 579 – 580.

[22] Najjar, W.; Gaudiot, J.-L.; Network resilience: a measure of network fault tolerance, IEEE Transactions on Computers, Volume: 39 , Issue: 2, 1990 , Page(s): 174 - 181

[23] Hui-Ling Liu; Shooman, M.L.; Simulation of computer network reliability with congestion, Proceedings. Annual Reliability and Maintainability Symposium, 1999, Page(s): 208 - 213

[24] Penido, G.; Nogueira, J.M.; Machado, C.; An automatic fault diagnosis and correction system for telecommunications management, Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management, 1999, Page(s): 777 - 791

[25] Pullan, W.; Optimising multiple aspects of network survivability, CEC '02. Proceedings of the 2002 Congress on Evolutionary Computation, 2002, Page(s): 115 - 120

[26] Zolfaghari, A.; Kaudel, F.J. Framework for network survivability performance, IEEE Journal on Selected Areas in Communications, Volume: 12 , Issue: 1, 1994, Page(s): 46 - 51

[27] Liew, S.C.; Lu, K.W.; A framework for characterizing disaster-based network survivability; ICC '92, Conference record, SUPERCOMM/ICC '92, IEEE International Conference on Communications, 1992, Page(s): 405 - 410 vol.1

[28] Bagula, A.B.; Improving the Resilience of Emerging Generation GMPLS Networks, IEEE/OSA Journal of Optical Communications and Networking, Volume: 1 , Issue: 2, 2009 , Page(s): A56 - A68

[29] Kano, S.; Miyazaki, K.; Nagata, A.; Chugo, A.; Shared segment recovery mechanism in optical networks, APSITT 2005 Proceedings. 6th Asia-Pacific Symposium on Information and Telecommunication Technologies, 2005, Page(s): 415 - 420

[30] Nilsson, A.A.; Lai, F.-Y.; Performance evaluation of error recovery schemes in high speed networks, ICC '90, Including Supercomm Technical Sessions. SUPERCOMM/ICC '90. Conference Record., IEEE International Conference on Communications, 1990., Page(s): 722 - 726 vol.2

[31] Gobel, J.; Krzesinski, A.E.; Stapelberg, D.; A Distributed Scheme for Responsive Network Engineering , ICC '07. IEEE International Conference on Communications, 2007, Page(s): 2070 - 2075

[32] Kvalbein, A.; Hansen, A.F.; Cicic, T.; Gjessing, S.; Lysne, O.; Multiple Routing Configurations for Fast IP Network Recovery, IEEE/ACM Transactions on Networking, Vol: 17, Issue 2, 2009

[33] Bonaventure, O.; Filsfils, C.; Francois, P.; Achieving Sub-50 Milliseconds Recovery Upon BGP Peering Link Failure, IEEE/ACM Transactions on Networking, Vol: 15, Issue 5, 2007

[34] Cortes, A.; Garcia-Rubio, C.; Campo, C.; Marin, A.; Almenarez, F.; Diaz, D.; Decoupling path failure detection from congestion control to improve SCTP failovers, Communications Letters, IEEE Volume: 12 , Issue: 11, 2008

[35] Braden, R. (Ed.), Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource ReserVation Protocol Version 1 Functional Specification", RFC 2205, September 1997.

[36] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F. and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, April 2001.

[37] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V. and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[38] Berger, L., Editor, "Generalized Multi-Protocol Label Switching (MPLS) Signaling - Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[39] Z. Lin, D. Pendarakis. „Documentation of IANA assignments for Generalized MultiProtocol Label Switching (GMPLS) Resource Reservation Protocol - Traffic Engineering (RSVP-TE) Usage and Extensions for Automatically Switched Optical Network (ASON)", RFC3474, March 2003

[40] Lang, Rjagopalan, Papadimitriou, Generalized Multi-Protocol Label Switching (GMPLS) Recovery Functional Specification, March 2006

[41] Manie, Papadimitriou, Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching(GMPLS), 2006

[42] D. Papadimitriou, E. Manie, „Analysis of Generalized Multi-Protocol Label Switching (GMPLS)-based Recovery Mechanisms (including Protection and Restoration)", RFC4428, 2006

[43] A. Farrel, A. Ayyangar, JP. Vasseur, "Inter-Domain MPLS and GMPLS Traffic Engineering – Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions" , RFC 5151, February 2009

[44] Bruce Davie and Yakov Rekhter, "MPLS: Technology and Applications", Morgan Kaufmann Publishers', 2000

[45] ITU-T Rec E.800 "Terms and definitions related to quality of service and network performance including dependability", ITU-T Standardization Organization, August 1994

[46] A. Farrel, J.-P. Vasseur, A. Ayyangar, "A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering", RFC4726, Nov. 2006

[47] M. Ghanbari, C.J. Hughes, M.C.Sinclair, J.P.Eade, "Principles of Performance engineering for Telecommunication and Information Systems", Institution of Electrical Engineers, 1997

[48] ITU-T Rec G.841"Types and characteristics of SDH network", ITU-T Standardization Organization, October 1998

[49] ITU-T Rec M.20 „Maintenance philosophy for telecommunication networks", ITU-T Standardization Organization, 1992

[50] D. Katz, D. Ward, "Bidirectional Forwarding Detection", draft-ietf-bfd-base-09.txt, February 2009

[51] E. Rosen, A. Viswanathan, R. Callon, „Multiprotocol Label Switching Architecture", RFC 3031, January 2000.

[52] T. Przygienda,N. Sheth, M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs), RFC5120, February 2008

[53] V. Sharma, F. Hellstrand „Framework for Multi-Protocol Label Switching (MPLS)-based Recovery", RFC 3469, Feb. 2003

[54] E. Osborne, A. Simha, "Traffic Engineering with MPLS", Cisco Press, 2002

[55] OIF, "User Network Interface (UNI) 1.0 Signaling Specification", Optical Internetworking Forum, October 2001

[56] Kammerhuber N., Fessi A., Carle G.; Resilience: Widerstandsfähigkeit des Internets gegen Störungen – Stand der Forschung und Entwicklung, Informatik Spektrum, April 2010, S. 131 - 142

[57] Floyd, S. and V. Jacobson, "Synchronization of Periodic Routing Messages", IEEE/ACM Transactions on Networking, Vol. 2, No. 2, , 1994.

[58] FP7EU project. ResumeNet; resilience and survivability for future networking: Framework, mechanism and experimental evaluation. http://www.resumenet.eu

[59] ITU-T Rec. G.7713/Y.1704, "Distributed Call and Connection Management (DCM)", ITU-T Standardization Organization, November 2001.

[60] ITU-T G.7712/Y.1703 „Architecture and specification of data communication network", ITU-T Standardization Organization, October 2001

[61] Andrew S. Tanenbaum, "Computer networks", Prentice Hall PTR, Pearson Education, 2003

[62] L.Peterson, B.Davie, "Computer networks: a systems approach", Morgan Kaufmann, 2003

[63] Saradhi, C.V.; Kumar, G.S.; Luying Zhou; Mohan, G. "A Fast and Efficient Segmented Signalling Protocol for GMPLS/WDM Optical Networks", IEEE International Conference on Communications , Page(s):5422 - 5426, May 2008

[64] Y. Mei and C. Qiao, "Efficient Distributed Control Protocolls for WDM All-Optical Networks", Proc. IEEE IC3N, p. 150-153, 1997

[65] X. Yuan, et al., "Distributed Control Protocols for Wavelength Reservation and their Performance Evaluation, Photonic Network Communications, vol. 1, no.3, p. 207-218, 1999

[66] D. Saha, "An Efficient Wavelength Reservation Protocol for Lightpath Establishement in All-Optical Networks (AON´s), Proc. IEEE Globecom, vol. 2, p. 1262-1268, Nov 2000

[67] Katib, I.; Medhi, D.; "Performance of distributed reservation control in wavelength-routed all-optical WDM networks with Adaptive Alternate Routing", IFIP/IEEE International Symposium on Integrated Network Management, p 505-512, 2009

[68] Feifei Feng et al. "Performance Study of Distributed Wavelength Reservation Protocols with Both Single and Multi-Fiber WDM Networks", Photonic network communications, vol6, no. 2, p. 95-103, Oct. 2004

[69] K. Lu, et al., "Intermediate-Node Initiated Reservation (IIR): A New Signaling Scheme for Wavelength Routed Networks", IEEE Journal on Selected Areas in Communications, vol. 21 no. 8, p. 1285-1294, 2003

[70] H. Zang, et. al., "A Review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks", Optical Networks Magazine, vol. 1, Jan. 2000

[71] Al-Karaki JN, Kamal AE, "Routing techniques in wireless networks: A survey". IEEE Wireles Communication Magazine 11(6): p.6-28, 2004

[72] Estrin D. Govindan R, Heidemann J, Kummar S, "Next Century Challenges: Scalable Coordination in Sensor Networks", IEEE/ACM International Conference on Mobile Computing and Networking (MobiCom), Seattle, Washington, USA, S 263-270, 1999

[73] Kim S, et.al, "Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks". IEEE/ACM International Conference on Information Processing in Sensor Networks (IPSN), p 254-263, 2007

[74] Stoianov I, Nachman L, Madden S, "PIPENET: A Wireless Sensor Network for Pipeline Monitoring". IEEE/ACM International Conference on Information Processing in Sensor Networks (IPSN), Cambridge, USA, S. 264-273, 2007

[75] Zimmerling M, Dargie W, Reason JM (2007) Energy-Efficient Routing in Linear Wireless Sensor Networks. IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS), Pisa, Italy, S. 1-3, 2007

[76] Lucian Popa, Costin Raiciu, Ion Stoica, David S. Rosenblum, "Congestion Effects in Wireless Networks by Multipath Routing", Proceedings of the 14th IEEE International Conference on Network Protocols, p. 96-105, 2006,

[77] Wan C.Y. Eisenman S.B., Campbell A.T., "CODA: Congestion Detection and Avoidance in Sensor Networks," in Proc. of SenSys, p. 266-279, 2003.

[78] Hull B. Jamieson K., Balakrishnan H., "Mitigating Congestion in Wireless Sensor Networks," in Proc. of SenSys, p. 137-147, 2004.

[79] Kang J. Nath B., Zhang. Y., Yu S., "Adaptive Resource Control Scheme to Alleviate Congestion in Sensor Networks," in Proc. of Broadnets, 2004.

[80] D. A. Tran, H. Raghavendra, "Routing with Congestion Awareness and Adaptivity in Mobile Ad hoc Networks", in Proc. of WCNC, p. 1988-1994, 2005.

[81] Peter P. Pham and Sylvie Perreau, "Performance Analysis of Reactive Shortest Path and Multipath Routing Mechanism With Load Balance", in Proc. Of Infocom, vol.1 p. 251-259, 2003

[82] IETF, Multi Path TCP Working Group, http://www.ietf.org/dyn/wg/charter/mptcp-charter.html

[83] Janardhan Iyengar, Bryan Ford, "An Architectural Perspective on MPTCP", Presentation of MPTCP BoF at IETF75, 30 July 2009, http://tools.ietf.org/html/draft-iyengar-ford-tng-00

[84] A. Ford, C, Raiciu, S. Barre, J. Iyengar, "Architectural Guidelines for Multipath TCP Development", draft-ietf-mptcp-architecture-00, February 28, 2010

[85] Pister K. S. J. Kahn J. M. and Boser B. E., "Smart Dust: Wireless Networks of Millimeter-Scale Sensor Nodes," in Electronics Research Laboratory Research Summary, p. 271-278, 1999.

[86] D. Wischik, M Handle, M.B. Braun, 2008. „The Ressource pooling principle", ACM SIGCOMM Computer Communication Review. Vol. 38, Issue 5, pp. 47-52.

[87] D. Acemoglu, R. Johari, and A. Ozdaglar. "Partially optimal routing". IEEE Journal of selected areas in communications, p. 1148-1160, 2007.

[88] V. Aggarwal, A. Feldmann, and C. Scheideler. „Can ISPs and P2P users cooperate for improved performance?", ACM SIGCOMM, Computer Communication Review, ,p. 29-40, 2007.

[89] A. La Oliva , M. Bagnulo , A. García-Martínez , Ignacio Soto, "Performance Analysis of the Reachability Protocol for IPv6 Multihoming, Proceedings of the 7th international conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking", vol. 4712, p. 443-454, 2007

[90] Y. Dong, D. Wang, N. Pissinou, and J. Wang. "Multi-path load balancing in transport layer". In Proc. 3rd EuroNGI Conference on Next Generation Internet Networks, p. 135-142, 2007.

[91] Richard J. Gibbens , Frank P. Kelly , Stephen R. E. Turner, "Dynamic routing in multiparented networks", IEEE/ACM Transactions on Networking (TON)", v.1 n.2, p.261-270, 1993

[92] H. Han, S. Shakkottai, C.V. Hollot, R. Srikant, and D. Towsley. "Multi-Path TCP: A Joint Congestion Control and Routing Scheme to Exploit Path Diversity on the Internet" IEEE/ACM Trans. Networking, vol. 14/6, p. 1260-1271, 2006.

[93] Hung-Yun Hsieh , Raghupathy Sivakumar, pTCP: "An End-to-End Transport Layer Protocol for Striped Connections", Proceedings of the 10th IEEE International Conference on Network Protocols, p.24-33, November 12-15, 2002

[94] Frank Kelly , Thomas Voice, "Stability of end-to-end algorithms for joint routing and rate control", ACM SIGCOMM Computer Communication Review, v.35 n.2, April 2005

[95] P. Key, L. Massoulié, and D. Towsley, "Path selection and multi-path congestion control" In Proceedings IEEE Infocom, p. 143-151, May 2007.

[96] Craig Labovitz , Abha Ahuja , Abhijit Bose , Farnam Jahanian, "Delayed Internet routing convergence", IEEE/ACM Transactions on Networking (TON), v.9 n.3, p.293-306, June 2001

[97] C.N. Laws. "Resource pooling in queueing networks with dynamic routing" Advances in Applied Probability, p. 699-726, 1992.

[98] E. Kohler, M. Handley, S. Floyd, "Designing DCCP: congestion control without reliability", Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, vol. 36, issue 4, p. 27-38, 2006

[99] Xiaoqiao Meng , Zhiguo Xu , Beichuan Zhang , Geoff Huston , Songwu Lu , Lixia Zhang, "IPv4 address allocation and the BGP routing table evolution", ACM SIGCOMM Computer Communication Review, v.35 n.1, January 2005

[100] D. Meyer, L. Zhang, and K. Fall. "Report from the IAB Workshop on Routing and Addressing", RFC 4984, 2007.

[101] K. Rojviboonchai and H. Aida. "An evaluation of multipath transmission control protocol (M/TCP) with robust acknowledgement schemes", IEICE Trans. Communications, 2004.

[102] Tim Roughgarden , Éva Tardos, "How bad is selfish routing?", Journal of the ACM (JACM), v.49 n.2, p.236-259, March 2002

Titluri recent publicate în colecția „TEZE DE DOCTORAT" seria 7: Inginerie Electronică și Telecomunicații

1. **Adrian Lazăr Șchiop** – *Contribuții la studiul convertoarelor utilizate la acționarea motoarelor sincrone, ISBN: 978-973-625-409-3*, (2007);

2. **Ioan Gavriluț** – *Contribuții la navigația roboților mobili autonomi utilizând rețelele neuronale, ISBN: 978-973-625-417-8*, (2007);

....

....

21. **Mircea Coșer** – *Contribuții privind optimizarea sistemelor folosind tehnica TRIZ, ISBN: 978-606-554-007-1*, (2009);

22. **Marius Oltean** – *Contribuții la optimizarea transmisiei pe cale radio, folosind funcții Wavelet, ISBN: 978-606-554-050-7*, (2010);

23. **Beniamin Drăgoi** - *Cercetări privind concepția și proiectarea oscilatoarelor sinusoidale integrate în tehnologie CMOS, ISBN:978-606-554-105-4*, (2010);

24. **János Gál** - *Contribuții privind utilizarea filtrelor Kalman în telecomunicații, ISBN:978-606-554-119-1*, (2010);

25. **Mirella Amelia Mioc** - *Contributions to applications development in cryptography and testing using shift registers, ISBN: 978-606-554-123-8*, (2010);

26. **Cristian Țecu** - *Contribuții la utilizarea noilor tehnologii informaționale în diaporama digitală, ISBN: 978-606-554-156-6*, (2010);

27. **Ciprian Ovidiu Miron Dughir** – *Contribuții la monitorizarea calității energiei electrice, ISBN: 978-606-554-181-8*, (2010);

28. **Florin-Josef Lătărețu** - *Some Methods to Increase Network Resilience And Recovery Performance, ISBN:978-606-554-203-7*, (2010).

EDITURA POLITEHNICA

Lista completă a tezelor publicate sub sigla Editurii Politehnica poate fi consultată la adresa: **http://www.editurapolitehnica.upt.ro**