# Distributed Failure Restoration for Asynchronous Transfer Mode (ATM) Tactical Communication Networks

Alexander Zavialov

Submitted in partial fulfilment of

the requirements for the degree of

Doctor of Philosophy

De Montfort University

in collaboration with

Moscow Bauman State Technical University

September 2002

# Abstract

Asynchronous Transfer Mode (ATM) is an attractive choice for future military communication systems because it can provide high throughput and support multi-service applications. Furthermore its use is consistent with the 'off the shelf' technology policy that is currently operated by the Defence Engineering Research Agency of Great Britain. However, ATM has been developed as a civil standard and is designed to operate in network infrastructures with very low failure rates. In contrast, tactical networks are much less reliable. Indeed tactical networks operate on the premise that failures, particularly node failures, are expected. Hence, efficient, automatic failure restoration schemes are essential if an ATM based tactical network is to remain operational. The main objective of this research is the proposal and verification of one or more new restoration algorithms that meet the specific requirements of tactical networks.

The aspects of ATM networks that influence restoration algorithms' implementation are discussed. In particular, the features of ATM networks such as the concept of Virtual Paths Virtual Channels and OAM (Operation And Maintenance) mechanisms that facilitate implementation of efficient restoration techniques. The unique characteristics of tactical networks and their impact on restoration are also presented.

A significant part of the research was the study and evaluation of existing approaches to failure restoration in civil networks. A critical analysis of the suitability of these approaches to the tactical environment shows no one restoration algorithm fully meets the requirements of tactical networks. Consequently, two restoration algorithms for tactical ATM networks, DRA-TN (Dynamic Restoration Algorithm for Tactical Networks) and PPR-TN (Pre-planned Restoration Algorithm for Tactical Networks), are proposed and described in detail. Since the primary concern of restoration in tactical networks is the recovery of high priority connections the proposed algorithms attempt to restore high-priority connections by disrupting low-priority calls. Also, a number of additional mechanisms are proposed to reduce the use of bandwidth, which is a scarce resource in tactical networks.

It is next argued that software simulation is the most appropriate method to prove the consistency of the proposed algorithms, assess their performance and test them on different network topologies as well as traffic and failure conditions.

For this reason a simulation software package was designed and built specifically to model the proposed restoration algorithms. The design of the package is presented in detail and the most important implementation issues are discussed. The proposed restoration algorithms are modelled on three network topologies under various traffic loads, and their performance compared against the performance of known algorithms proposed for civil networks. It is shown that DRA-TN and PPR-TN provide better restoration of higher priority traffic. Furthermore, as the traffic load increases the relative performance of the DRA-TN and PPR-TN algorithms increases. The DRA-TN and PPR-TN algorithms are also compared and their advantages and disadvantages noted.

Also, recommendations are given about the applicability of the proposed algorithms, and some practical implementation issues are discussed. The number of problems that need further study are briefly described.

# Table of Contents

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| AAL | ATM Adaptation Layer |
| AAU | ATM-user-to-ATM-user |
| ABR | Available Bit Rate |
| AIS | Alarm Indication Signal |
| APS | Automatic Protection Switching |
| ATM | Asynchronous Transfer Mode |
| B-ISDN | Broadband Integrated Services Digital Network |
| CAC | Call Admission Control |
| CBR | Constant Bit Rate |
| CCITT | Consultative Committee for International Telegraph and Telephone |
| CDV | Cell Delay Variation |
| CDVT | CDV Tolerance |
| CLP | Cell Loss Priority |
| CLR | Cell Loss Ratio |
| COTS | Commercial Off the Shelf |
| CRC | Cyclic Redundancy Code |
| CS | Convergence Sublayer |
| CTD | Cell Transfer Delay |
| DERA | Defence Engineering Research Agency of Great Britain |
| DRA | Dynamic Restoration Algorithm |
| DRA-TN | Dynamic Restoration Algorithm for Tactical Networks |
| FERF | Far End Receive Failure |
| FRVP | Failure-Resistant Virtual Path |
| GFC | Generic Flow Control |
| HEC | Header Error Control |
| HLC | Hop Limit Counter |
| HT | hold-off timeout |
| IR | Immediate Re-routing |
| ITU | International Telecommunication Union |
| LAN | Local Area Network |
| LOS | Loss of Signal |
| LR | Late Re-routing |

| | |
|---|---|
| MBS | Maximum Burst Size |
| MCR | Minimum Cell Rate |
| NBU | Network Bandwidth Utilization |
| NNI | Network-Node Interface |
| NRT-VBR | non-Real-Time VBR |
| OAM | Operation And Maintenance |
| PCR | Peak Cell Rate |
| PDU | Protocol Data Unit |
| PDH | Plesiochronous Digital Hierarchy |
| PPR-TN | Pre-planned Restoration Algorithm for Tactical Networks |
| PT | Payload Type |
| QoS | Quality of Service |
| RP | Restoration Probability |
| RT-VBR | Real-Time VBR |
| SAR | Segmentation and Reassembly |
| SCR | Sustainable Cell Rate |
| SDH | Synchronous Digital Hierarchy |
| SHR | Self-Healing Rings |
| SONET | Synchronous Optical Network |
| UBR | Unspecified Bit Rate |
| UNI | User-Network Interface |
| VBR | Variable Bit Rate |
| VC | Virtual Channel |
| VCC | Virtual Channel Connection |
| VCI | Virtual Channel Identifier |
| VP | Virtual Path |
| VPC | Virtual Path Connection |
| VPI | Virtual Path Identifier |
| VPPS | Virtual Path Protection Switching |
| WRP | Weighted Restoration Probability |

# Acknowledgements

First of all I would like to thank my principal supervisor Amelia Platt. She has supported my research in all the possible ways always providing invaluable help and assistance.

Also I would like to thank Mike Morse with whom I started this project at DMU and my second supervisor Professor Valery Galkin from BMSTU for their help and constructive criticism during the research.

I gratefully acknowledge the Defence Engineering Research Agency of Great Britain, which funded this research in the frames of the join project with De Montfort University and Bauman Moscow State Technical University. I want personally thank project coordinators Brian Foxon, Alexander Chernikov and Graham Hudson.

I would also like to express my gratitude to Tracey Gosling, Charlotte Kitson and all the staff of the De Montfort University and Bauman Moscow State Technical University who helped me to overcome various problems.

Finally, much appreciation is given to family and friends who supported me during my research and writing of this thesis.

# Chapter 1. Introduction

## 1.1 Introduction

In the context of communication networks, the term restoration refers to the capability of the network to reconfigure itself when failure of one or more network elements (links and nodes) occurs. Disrupted connections are re-routed over the elements that continue to function correctly. Ideally, restoration should be automatic upon detection of a failure. Furthermore it should be completed rapidly, to ensure that no calls are dropped.

There has been much work carried out into network restoration. This was largely in response to failures in civil networks that caused major communications disruption for both residential and business customers. Furthermore the nature of the failures meant that it took a long time for the networks to be restored. In contrast little research has been carried out into restoration in land tactical networks; these are military networks used at the battlefront. Tactical and civil networks have major differences both in the physical network configuration and in the traffic that they carry. Similarly, the types of failures that they experience and the probability of failure also differ and the constraints for network restoration are much more stringent for tactical networks.

In the past, bespoke systems were typically developed for military applications, however the recent trend is to use Commercial Off the Shelf (COTS) technologies wherever possible. Asynchronous Transfer Mode (ATM) is an attractive choice for future military communication systems because the next generation of land tactical networks are required to offer far greater network throughput and support multi-service (voice, data and video) applications [4, 5, 58, 60] and ATM was designed specifically to support these requirements, albeit in civil networks. Furthermore, ATM defines a number of mechanisms that are useful for network restoration. Hence, it is expected that future tactical networks will be based on ATM, with necessary adaptations included to allow it to operate in the tactical environment [4, 5, 58].

## 1.2 Aims and Objectives of the Research

To use ATM in the tactical arena a number of problems must be solved. These include the performance issues associated with error prone radio links, issues associated with network infrastructure dynamics, service characteristics such as priority and pre-emption, and others. Also in order that services can be carried reliably in a tactical ATM network, an automatic failure restoration function must be implemented efficiently.

This project was funded by DERA (the Defence Engineering Research Agency of Great Britain) as a part of the research programme on adaptation of ATM for tactical environment. The subject of the study covers the concepts, algorithms and techniques needed to support the distributed failure restoration (self-healing) function in tactical ATM networks. The aims and objectives of this research are as follows:

- identify specific characteristics of tactical networks (especially their distinctions from civil networks) and the corresponding requirements to restoration algorithms;
- analyse existing approaches to civil networks restoration and assess applicability of different methods, algorithms and techniques for tactical environment;
- propose one or more restoration algorithms suitable for tactical networks;
- define simulation environment parameters and model proposed restoration algorithms;
- compare the performance of the proposed algorithms against comparable civil restoration algorithms.

These objectives as well as tactical network characteristics and restoration algorithms' requirements were discussed and refined during the meetings with DERA researchers. The detailed project definition, results of the initial research and the work programme were presented in the transfer report, which was well received by DERA.

Correspondingly, the results of the work presented in this thesis will be forwarded to DERA, and they will be discussed along with the further work if necessary.

## 1.3 Outline of the Research

### 1.3.1 Use of ATM Technology

Chapter 2 discusses the unique characteristics of tactical networks and the traffic that they carry and how this places different and more stringent constraints on the restoration algorithms for these networks compared with civil networks. The significance of the four-level priority system for connections in tactical networks and how this affects the restoration process is also explained. Typical tactical network topologies are also discussed, including network size and connectivity. One major constraint for tactical networks is the low link bandwidth available; this means that it is not possible to build in bandwidth redundancy, specifically for restoration, which is the typical approach of civil networks. Finally, the differences in the tactical network environment, compared to the civil environment are discussed and the effect these have on the restoration is discussed. For instance, it is noted that the use of radio links and the nature of the tactical environment, where nodes can be destroyed by enemy action, make node failure the most typical scenario, compared to civil networks which consider link failures (cable cuts) only.

The basics of ATM technology are outlined then. In particular, the ATM cell format, the ATM protocol reference model, Quality of Service (QoS) types supported are presented. Several features of ATM networks that influence the restoration process and allow to implement specific restoration techniques are noted, including the concept of Virtual Paths and Virtual Circuits (VP/VC). The advantage gained from separating the route establishment process from the bandwidth allocation process is explained. Finally, the way in which restoration can make use of the Operation and Maintenance (OAM) mechanisms, particularly error detection (Alarm Indication Signal (AIS) and Far End Receive Failure (FERF) signal) is also explained.

### 1.3.2 Restoration Algorithms for Civil Networks

A significant part of the research was the study and evaluation of existing approaches to failure restoration. A critical analysis of the suitability of these approaches for tactical networks is given in Chapter 3. The various approaches to restoration in civil networks

are presented first. Some of these evolved from those of conventional networks, such as Synchronous Digital Hierarchy (SDH), Plesiochronous Digital Hierarchy (PDH), or Synchronous Optical Network (SONET), while others have been proposed specifically for ATM networks. Generally, all the restoration schemes can be characterised by the control scheme used (centralised or distributed), the layer at which restoration is implemented (physical or ATM layer), the type of restoration approach (real-time (dynamic), or pre-planned), and the type of rerouting which is carried out (link, local-destination or path rerouting).

It is argued that only two groups of distributed restoration algorithms proposed for civil networks can potentially be applicable for tactical environment. These are dynamic restoration algorithms and pre-planned restoration algorithms. Both of these have been proposed for mesh network topologies. Dynamic restoration algorithms use flooding techniques to discover alternative paths for disrupted connection recovery after a failure, while pre-planned restoration algorithms use pre-defined alternative routes which are activated after a failure occurs. Consequently, these two types of algorithms were studied in more detail and a description and critical analysis of these algorithms is presented.

The overall conclusion of the study was that there are no existing restoration algorithms fully meeting the needs of tactical networks, although some of the techniques used have some potential.

### 1.3.3 Restoration Algorithms for Tactical Networks

Since no one restoration algorithm proposed for civil networks fully meets the requirements of tactical environment, two restoration algorithms for tactical ATM networks are proposed in Chapter 4. One of them is a dynamic restoration algorithm, while the other is a pre-planned restoration algorithm.

The proposed algorithms take into consideration the peculiarities of the tactical environment. In particular, they do not attempt to restore *all* disrupted traffic, and, second, the primary concern is the restoration of high priority connections that are of much greater importance. Therefore, in the situation where there is not enough

bandwidth in the network to restore all the failed connections, the algorithms attempt to restore high-priority connections by disrupting low-priority calls. Also, a number of additional mechanisms are proposed to reduce the use of bandwidth which is a scarce resource in tactical networks.

The new Dynamic Restoration Algorithm for Tactical Networks (DRA-TN) uses some features of existing algorithms, and implements new mechanisms that have been designed to take account of the differences between civil and tactical networks. It is described in detail using the Rudin network topology as an example. Message formats and flow charts, which describe the algorithm, are given in Appendix A.

A detailed description of the Pre-Planned Restoration Algorithm for Tactical Networks (PPR-TN) is also presented in this chapter, while additional information is given in Appendix B.

### 1.3.4 Software Simulations

It is argued in Chapter 5 that software simulations is the most appropriate method to prove the consistency of the proposed algorithms, assess their performance characteristics, and test them on different network topologies as well as traffic and failure conditions.

Chapter 5 describes the approach to modelling. In the first part of the chapter the package that was designed and built to model restoration algorithms is presented. Its architecture and the most important aspects of restoration algorithms' implementation are presented in details. The package allows different algorithms to be modelled, using various network topologies, different traffic scenarios and failure modes. Because of the different constraints of tactical network restoration, some of the metrics typically used to assess the performance of restoration algorithms are inappropriate. Thus different metrics are proposed and these are explained in this chapter.

The second major part of the chapter presents the simulation environment parameters that were chosen for modelling. Three network topologies were selected from well-known experimental networks to test the algorithms on networks of various sizes. The characteristics of these networks are described and it is argued that these are typical

examples of small, medium and large scale tactical networks. The choice of values for other significant parameters has also been taken from the literature and these are discussed in the chapter.

To provide a bench mark with which to measure the effectiveness and efficiency of the new restoration algorithms proposed by this study, it was necessary to model existing restoration algorithms and calculate similar metrics. Details of the existing, benchmark algorithms which were chosen for comparative analyses are presented in the chapter together with reasons why these were the most appropriate algorithms to model.

### 1.3.5 Modelling Results

Chapter 6 present the simulation results, their analysis and discussion. Having to model four algorithms on three different network topologies, a total of 12 sets of experiments were necessary. All the metrics produced by the simulations were analysed. A summary of the results of these experiments is given in this chapter together with graphs of the most important metrics; the full experimental results are presented in Appendix C.

The analysis of the results is presented next. The results of the DRA-TN and PPR-TN algorithms are compared with their equivalent benchmark algorithms and a detailed analysis given. It is shown that the proposed algorithms provide better restoration of higher priority traffic. Furthermore, as the traffic load increases the performance of the proposed algorithms increases compared with the existing algorithms.

DRA-TN and PPR-TN algorithms are also compared. The results indicate that DRA-TN is faster and generates less messages, while PPR-TN provides a higher restoration ratio. It is argued that DRA-TN is the most suitable algorithm for tactical networks restoration.

### 1.3.6 Further Work

In the last chapter a summary of the work completed is given, and the directions of further research are indicated.

Research achievements and limitations are summarised. Also recommendations are given about the applicability of the proposed algorithms, and some practical implementation issues are discussed. The number of problems that need further study is indicated and briefly described.

## 1.4 Summary

Chapter 1 introduced the concept of network restoration and the differences between civil and tactical networks. An overview of the research carried out and the structure of thesis are also presented.

# Chapter 2. ATM Technology

## 2.1 Introduction

This chapter discusses the characteristics of tactical networks and the implications these characteristics have on restoration. The rest of Chapter 2 is concerned with explaining the various aspects of ATM technology that is an attractive choice for future military communication systems.

## 2.2 Tactical Networks

Four levels of military communication networks are usually distinguished: strategic, operational, tactical and battlefield [5]. The last two levels are mobile networks and hence radio links are typically used. In contrast, the first two levels are stationary and use cable connections. The majority of connections are established using radio links while optical channels are engaged rarely [5].

This study concentrates on the tactical and battlefield networks that are called tactical networks in this thesis and are assumed to be wireless networks created between mobile military vehicles and possibly a limited number of fixed nodes. The mobile vehicles take up positions in an exercise or war zone and establish the network whilst stationary. From this point of view a tactical network can be considered as stationary. A possible topology change scenario can occur when the nodes in the tactical networks migrate as the battlefield moves. Then the nodes may switch off, move to the new position and establish a contact with neighbouring nodes again.

However, in the tactical environment the network is highly dynamic with non-optimally sited equipments. Equipment destined for a node can fail to arrive, network topologies are ad hoc because network nodes are a subject of enemy attacks, radio equipments are not sited for optimum performance, equipments fail, etc. [4]

According to information received from DERA during our meetings and discussions, the topology of a tactical network is a sparsely-connected mesh with average

connectivity of about 3 - 4. Radio waves are used as the transmission medium, and this results in higher bit error rates and lower transmission speeds, compared to fixed networks. The maximum trunk speed is assumed to be 2 Mbps [5], although frequently it is 0.5 or 1 Mbps.

Voice, data and video services are supported, and according to information received from DERA each service has four priority levels. However, there is no specific information (yet) about how priorities are assigned to different services. Moreover, the issues about the number of priorities and a possibility to change a priority level during an operation are still unclear [58]. Though some authors suggest that a three-level priority system could be used in military networks, it is assumed in this research that there are four priority levels because this was one of the initial DERA requirements identified at the project definition stage. Note however that restoration algorithms proposed in this study can function correctly for any number of priority levels.

The queuing strategy described in [58] allows us to suppose that the high-priority traffic is always more important than the low-priority one. According to this policy a priority sensitive switch puts traffic in the queue in the priority order. So in case of congestion only the low-priority traffic is lost or retransmitted.

Asynchronous Transfer Mode (ATM) technology is an attractive choice for future military communication systems because the next generation of land tactical networks are required to offer far greater network throughput and support multi-service (voice, data and video) applications [4, 5].

Another important reason for adopting ATM for military communications is that due to the development and extensive use of ATM in the highly competitive commercial marketplace, the military see significant cost benefits in its implementation [4]. To realise these benefits it is essential that the Commercial Off the Shelf (COTS) technology is modified as little as possible.

However, to use ATM in the tactical arena a number of problems must be addressed [4, 58]. These are:

- performance issues associated with error prone radio links;

- service characteristics such as priority and pre-emption;
- diverse routing;
- admission and congestion control;
- security;
- survivability (including automatic failure restoration), etc.

Some of these problems have been already addressed by various researchers. For example, a number of link hardening mechanisms are proposed to sustain the performance of ATM over narrowband and error prone radio links [4]. These include implementation of Automatic Repeat Request (ARQ) mechanism or a hybrid-ARQ scheme based on using Forward Error Correction (FEC) on a link-by-link basis to recover ATM cells from multiple bit errors in the header. Other problems need further investigations.

ATM was developed for operation in network infrastructures which provide very low failure rates that are characteristic of current and future civil communication networks; tactical networks are much less reliable. Therefore, if multiple services are to be supported in tactical ATM networks, efficient automatic failure restoration techniques are critical.

According to DERA recommendations the following constraints must be imposed on the failure restoration function in tactical ATM networks:

- it needs to be implemented in a distributed way to avoid the vulnerability of a single point failure;
- the amount of spare capacity available is limited;
- it must be implemented efficiently to avoid dropping the highest priority calls (i.e. it is preferable to provide restoration without call re-establishment);
- service precedence should be considered, so that critical services are restored in favour of less critical ones.

Another important DERA concern was the recognition of specific characteristics of tactical communication networks. Indeed, there are many important differences between civil and military networks which influence the implementation of automatic failure restoration function considerably.

The topology of civil ATM networks is also likely to be a mesh but the transmission bandwidth is much greater compared to tactical networks. For this reason, civil network providers may, by design, include enough redundancy of the network resources to allow restoration of all disrupted calls. The large amount of bandwidth in optical fibre cables and its low cost allow reservation of up to 50% of the total bandwidth for restoration. This is not possible in tactical networks.

The other significant difference between civil and tactical networks is the likely failure modes. Civil networks will most probably suffer single link failure (damaged fibres), rarely node failures. Typical failure scenarios in tactical networks however are likely to be single and multi-node failures and several consecutive failures, possibly brought about by enemy action. Restoration techniques for tactical networks must recognise that difference as well.

This project was funded by DERA as a part of the research programme on adaptation of ATM for tactical environment. The study concentrates on the concepts, algorithms and techniques needed to support the distributed failure restoration function in tactical ATM networks. The basics of the ATM technology are outlined in the rest of this chapter.

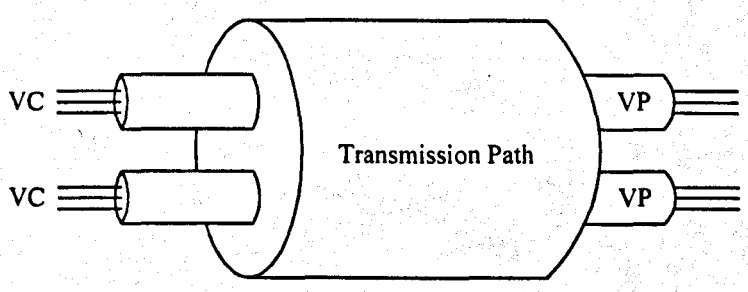
## 2.3 Virtual Channels and Virtual Paths Concept

ATM is a connection-oriented, fast packet switching technology that has been standardised for B-ISDN networks. A logical connection between the two end-systems needs to be set up across an ATM network prior to any data transfer. Then, a variable-rate, full-duplex flow of fixed-size packets, called ATM cells, is exchanged over the connection.

Logical connections in ATM are referred to as Virtual Channels (VC). Virtual channels are set up between two end users through the network, and are used to transfer data, control signalling, network management and routing information.

Another sublayer of processing introduced in ATM deals with the concept of virtual paths (Figure 2.1). A Virtual Path (VP) is a bundle of VCs that have the same endpoints. In the network, all these VCs are switched together by referring to the VP Identifier (VPI) in the ATM cell header (see below for ATM cell format). Grouping connections sharing common paths through the network into a single unit, the virtual path, minimises the switching and maintenance costs. Network management actions can then be applied to a small number of *groups of connections* (VPs) instead of a large number of individual connections (VCs).

Accordingly, there are two types of switching in ATM networks: VP/VC switching and VP switching, as illustrated respectively in Figures 2.2a and 2.2b. With VP switching VCs multiplexed onto VPs are switched from input to output links using the VPI (Virtual Path Identifier) field of cell headers only; note that the VPIs change but the VCIs remain the same. For instance, in Fig 2.2b VC23 on VP2 is switched to VC23 on VP4. In contrast with VP/VC switching each VC is switched individually based on both its VPI and VCI (Virtual Channel Identifier) values. For instance, in Fig 2.2a VC22 on VP1 is switched to VC32 on VP3.

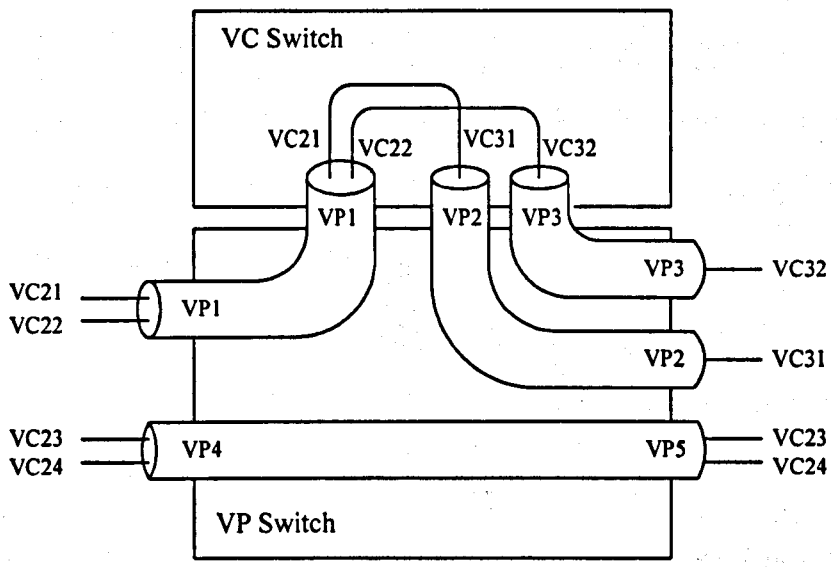



VP - Virtual Path
VC - Virtual Channel

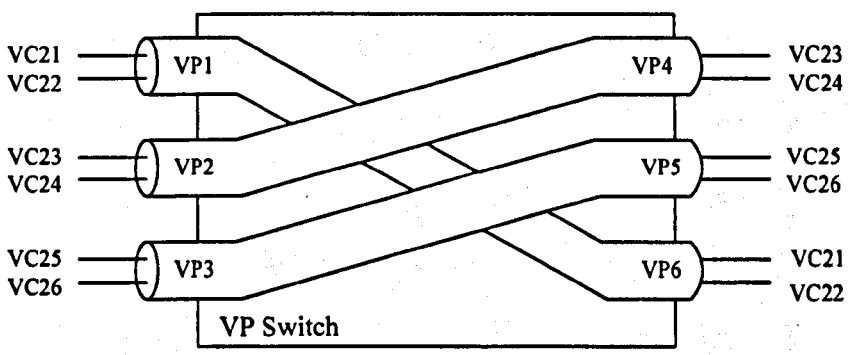
Figure 2.1 ATM Connection Relationship.

The basic functions of an ATM switch are as follows:

- to receive a cell across a link (port) on a known VPI or VCI/VPI value;
- to look up this VPI (or VCI/VPI) in a local switch table (Figure 2.3) to determine the outgoing port of the connection and the new VPI (or VCI/VPI) value of the connection;
- to retransmit the cell on that outgoing link with the appropriate connection identifiers.




(a) VC and VP Switching.




(b) VP Switching.

Figure 2.2 VP and VC Switching Hierarchy.

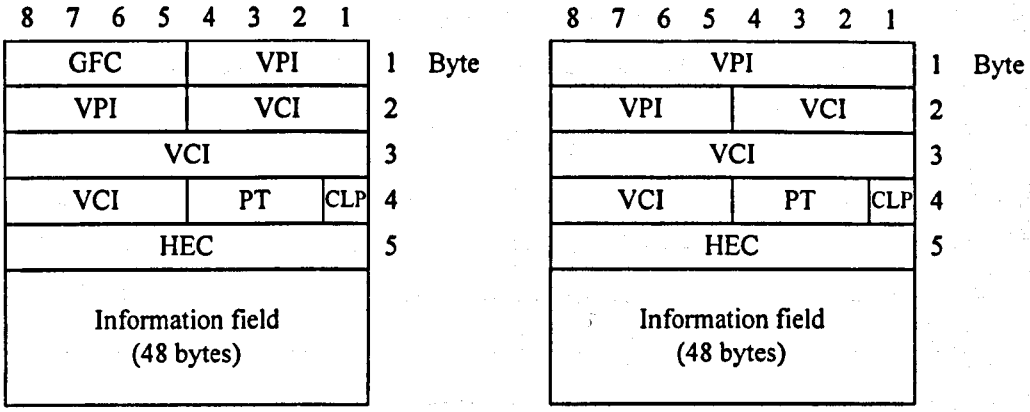
| Input | | | Output | | |
|---|---|---|---|---|---|
| Port | VPI | VCI | Port | VPI | VCI |
| 11 | 1 | 1 | 12 | 2 | 2 |
| 12 | 2 | 2 | 11 | 1 | 1 |
| 13 | 5 | 6 | 14 | 5 | 7 |
| 14 | 5 | 7 | 13 | 5 | 7 |

Figure 2.3 Example of an ATM Node Switch Table.

Note that one of the most important features of ATM is that VP and VC establishment is independent from bandwidth assignment (i.e. a connection can be assigned zero bandwidth). This is a useful feature for network restoration as backup VPs and VCs can be pre-established and is one of the main reasons why the concept of virtual paths and virtual channels is exploited in several restoration techniques proposed for ATM networks.

## 2.4 ATM Cell Format

An ATM cell consists of a 5-byte header and a 48-byte information field (Figure 2.4). Information in the header is used to perform switching (as defined above), flow control, and other functions. There are two different header formats defined: one for the User-Network Interface (UNI) and the second for the Network-Node Interface (NNI).




(a) User-Network Interface

(b) Network-Network Interface

| | | | |
|---|---|---|---|
| GFC | Generic Flow Control | PT | Payload Type |
| VPI | Virtual Path Identifier | CLP | Cell Loss Priority |
| VCI | Virtual Channel Identifier | HEC | Header Error Control |

Figure 2.4 ATM Cell Format.

In the UNI cell format, four bits are assigned to the Generic Flow Control (GFC) field, which is primarily responsible for shared-media, local-access, flow control for the traffic originated at user equipment and directed to the network. The remaining fields in the header are as follows:

- Virtual Path Identifier (8 bits) and Virtual Channel Identifier (16 bits) fields are used for routing.
- 3-bit long Payload Type (PT) field is used to identify the type of information carried by ATM cell. Table 2.1 shows the interpretation of the PT bits. A value of 0 in the first bit indicates user information, while 1 means that this cell carries network management or maintenance information. ATM-user-to-ATM-user (AAU) indication bit identifies cells conveying information between end users.
- One bit is assigned to the Cell Loss Priority (CLP) field to determine if the cell could be discarded based on network conditions.

- 8 bits are assigned to the Header Error Control (HEC) field to monitor header correctness and perform single bit error correction.

Table 2.1 Payload Type field coding.

| PT Coding | Interpretation |
|---|---|
| 0 0 0 | User data cell, congestion not experienced, AAU=0 |
| 0 0 1 | User data cell, congestion not experienced, AAU=1 |
| 0 1 0 | User data cell, congestion experienced, AAU=0 |
| 0 1 1 | User data cell, congestion experienced, AAU=1 |
| 1 0 0 | OAM F5 segment associated cell |
| 1 0 1 | OAM F5 end-to-end associated cell |
| 1 1 0 | Resource management cell |
| 1 1 1 | Reserved |

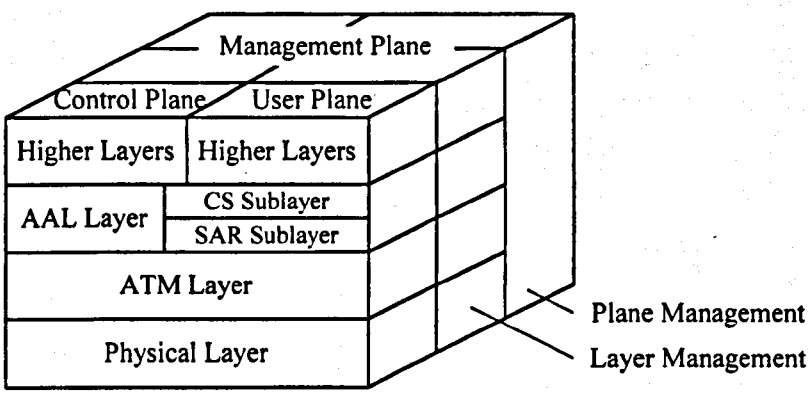
The NNI header structure is almost the same as the UNI's except that it has no GFC field, because this field has no use within the network. Accordingly, these bits are used as a part of the VPI field.

## 2.5 ATM Protocol Reference Model

The B-ISDN Protocol Reference Model for ATM defined for the host in CCITT I.321 "B-ISDN Protocol Reference Model and Its Application" [54] is shown in Figure 2.5. It consists of user, control and management planes. The user plane is responsible for user information transfer. The control plane processing signalling information is responsible for such functions as call set-up, maintenance and release. The management plane performs two types of operations: management of different planes and layer management. Layer management functions have a layered structure and handle specific OAM (Operation and Maintenance) information flows for each layer.

The user and control planes are organised into three layers: Physical Layer, ATM Layer and ATM Adaptation Layer (AAL).

The physical layer transports ATM cells between two ATM entities. Its functionality includes cell rate decoupling, header error control, cell delineation, transmission frame adaptation, generation and recovery.




AAL - ATM Adaptation Layer
CS - Convergence Sublayer
SAR - Segmentation and Reassembly Sublayer

Figure 2.5 B-ISDN Protocol Reference Model.

The ATM layer is common for all services and provides the cell transfer capability and defines the functions of the cell header. The ATM layer provides cell multiplexing and de-multiplexing, header addition and deletion, VPI/VCI translation (switching), and generic flow control.

The AAL supports the service requirements of various applications (CS sublayer) and maps higher layer PDUs into the information field of ATM cells and vice versa (SAR sublayer).

ATM network restoration can be performed at the physical or ATM layer. Physical level protection may be used when a dedicated redundant facility is available to restore disrupted connections, and for ATM layer restoration redundant VPIs/VCIs are required.

For ATM layer protection, restoration can be performed at the VC, VP (or group of VPs) level. Service restoration at the VC level is more flexible but slower and more expensive than at the VP level due to greater VC-based ATM network complexity.

## 2.6 Quality of Service

In order to support different service requirements for network applications, the ATM Forum specified several Quality of Service (QoS) categories:

- CBR (Constant Bit Rate);
- real-time VBR (Variable Bit Rate);
- non-real-time VBR;
- ABR (Available Bit Rate);
- UBR (Unspecified Bit Rate).

During a connection setup CBR services reserve a constant amount of bandwidth. This service is conceived to support applications such as voice, video and circuit emulation, which require small delay variations. The source is allowed to send at the negotiated rate any time and for any duration.

VBR services negotiate the Peak Cell Rate (PCR), the Sustainable Cell Rate (SCR) and Maximum Burst Size (MBS) traffic parameters. VBR sources are bursty. Typical VBR sources are voice and video. These applications require small delay variations. The VBR service is further divided in real-time VBR (RT-VBR) and non-real-time VBR (NRT-VBR). They are distinguished by the need for an upper bound delay (Max CTD, cell transfer delay). Max CTD is provided by RT-VBR, whereas for NRT-VBR no delay bounds are applicable.

ABR and UBR services should efficiently use the remaining bandwidth, which is necessarily changing in time dynamically because of the VBR traffic. Both transfer data without constraints on end-to-end delay and delay variation. Typical applications are computer communications, such as file transfer and e-mail.

UBR service provides no feedback mechanism. If the network is congested, UBR cells may be lost; this is analogous to the 'best-effort' service provided by IP. In contrast, an ABR source gets feedback from the network. The network provides information periodically about the bandwidth available to the connection and the state of congestion. The source's transmission rate is continually adjusted in accordance with this feedback information. This dynamic adjustment of transmission rate by the source, during the lifetime of the connection, is designed to reduce the probability of congestion. For ABR service, a guaranteed minimum bandwidth (MCR) is negotiated during the connection setup negotiations.

The traffic information, discussed above, together with the QoS requirements are used by the network to perform Call Admission Control (CAC) in which bandwidth is allocated to the connections in order to meet the QoS requirements of the new and existing connections. If sufficient bandwidth is not available, the connection request is rejected.

One of the most popular methods proposed for performing CAC is to use equivalent (or effective) bandwidth allocations, although this has not been standardised yet. The equivalent bandwidth of a connection is characterised by a value lying between the peak and mean bit-rates of the call. When a new connection is set up, an amount of bandwidth equal to its equivalent bandwidth is reserved on the whole route for the duration of connection. CAC control consists of determining whether there is sufficient available bandwidth to accept the new call.

## 2.7 OAM Functions

Recommendation I.610 "B-ISDN Operation and Maintenance Principles and Functions" [55] describes the minimum functions that must be implemented for maintaining the physical layer and the ATM layer at the user-network interface. Operation and Maintenance (OAM) functions specified for five types of actions are given in the Table 2.2.

The OAM functions are organised into five hierarchical levels (F1 to F5), which correspond to the hierarchical levels of transport functions. F5, F4, F3, F2 and F1

correspond to VC level, VP level, transmission path level, digital section level and regenerator section level respectively. F5 and F4 are functions at the ATM layer, and F3, F2 and F1 are functions at the physical layer. The ATM Layer's OAM functions are summarised in Table 2.3.

Table 2.2 OAM Actions.

| Function Name | Description | Result |
|---|---|---|
| Performance monitoring | The managed entity is monitored by continuous or periodic checking of functions. | Maintenance event information is produced. |
| Defect and failure detection | Malfunctions or predicted malfunctions are detected by continuous or periodic checking. | Maintenance event information or various alarms are produced. |
| System protection | Effect of failure of a managed entity is minimised by blocking or changeover to other entities. | The failed entity is excluded from operation. |
| Provision of failure or performance information | Failure information is given to other management entities. | Alarm indications are given to other management planes. Response to a status report request is also given. |
| Fault localisation | Determination by internal or external test system of a failed entity if failure information is insufficient. | |

Table 2.3 OAM Functions of the ATM Layer.

| Level | Function | Flow | Defect/Failure Detection | System Protection and Failure Information |
|---|---|---|---|---|
| Virtual Path | Monitoring of path availability. | F4 | Path not available. | For further study |
| | Performance monitoring. | | Degraded performance. | |
| Virtual Channel | Monitoring of channel availability. | F5 | Channel not available. | For further study |
| | Performance monitoring. | | Degraded performance. | |

ATM layer OAM information is transferred by OAM cells. VP level OAM cells are identified by a unique set of VCI values (3 and 4), whereas VC level OAM cells are

identified by a unique set of payload type values (4 and 5). ATM layer OAM functions include performance monitoring, failure reporting, continuity checking, and loopback testing.

Failure detection may occur in the physical or ATM layer. If a failure is detected in the physical layer, and it is not protected by any restoration technique, then the ATM layer will be notified. For failure reporting, two kinds of alarm indications are generated in the ATM layer: AIS (Alarm Indication Signal) and FERF (Far End Receive Failure). The AIS is used to alert the downstream nodes that a failure has been detected upstream. The FERF cell is sent upstream along the failed path indicating for the source that there is a problem downstream to the receiving node. This mechanism provides efficient means for error detection and notification.

Figure 2.6 shows the OAM cell format at the ATM layer. The OAM cell type field is used to indicate one of the three OAM cell types: performance management OAM cell, fault management OAM cell and activation/deactivation OAM cell. The OAM function type field indicates the OAM cell function (e.g., alarm reporting, continuity check, etc.). The OAM cell payload field differs depending on the type of OAM cell. 6 bits are unused and all set to zero. The CRC field contains an error detection code generated over the 48-byte payload.

OAM cells are widely used in various restoration techniques as restoration message transfer mechanism.

| Header | OAM cell type | OAM function type | OAM cell payload | Unused | CRC |
|---|---|---|---|---|---|
| 5 bytes | 4 bits | 4 bits | 45 bytes | 6 bits | 10 bits |

Figure 2.6 OAM Cell Format.

## 2.8 PNNI

In attempt to provide a standard-based solution to routing and signalling the ATM Forum defined a PNNI (Private Network-Node Interface, or Private Network-to-

Network Interface) protocol for use between ATM nodes and between groups of ATM nodes [8].

PNNI includes two categories of protocols. The first protocol is defined to distribute topology information between ATM nodes. This information is used to compute paths through the network. A hierarchy mechanism ensures that this protocol scales well for large worldwide ATM networks. A network is segmented into the peer groups. All the nodes within a peer group exchange link information among them obtaining topology database representing the group. Peer groups are organised into a hierarchy where peer groups are associated with parent groups, which in turn are grouped into higher layer groups and so forth. Routing outside a peer group is based on the same principles, but is achieved at the higher layers hierarchy. PNNI topology and routing is based on the well-known link-state routing technique.

The second protocol is defined for signalling, that is message flows used to establish point-to-point and point-to-multipoint connections across the ATM network. This protocol is based on the ATM Forum UNI signalling, which mechanisms added to support source routing, crankback[1] and alternate routing of call setup requests in case of connection setup failure.

PNNI could be potentially exploited for restoration purposes to implement, for example, a selective flooding mechanism. However, it is a very complex protocol requiring considerable traffic overhead. Consequently, it is doubtful that it will be used in tactical networks because of their limited resources.

## 2.9 Summary

ATM technology will be the technology used in future military communication systems. This chapter explained the various aspects of ATM. It also described the characteristics of tactical networks and the implications these characteristics have for restoration algorithms in tactical networks.

[1] Crankback procedures are used to reroute calls that are rejected within a PNNI domain.

# Chapter 3. Restoration Algorithms for Civil Networks

Various approaches to restoration, sometimes called self-healing, have been studied for civil networks. This chapter gives a critical analysis of this research. There are a number of ways in which restoration algorithms can be classified; these are explained in section 3.1. The remaining sections give a description and critical analysis of proposed algorithms, grouped according to the approach they use. The final section presents a tree diagram that shows how the algorithms discussed in the chapter are classified. It also summarises the disadvantages of the algorithms and draws the conclusion that none are suitable for restoration in a tactical network.

## 3.1 Classification of Network Restoration Schemes

There are several alternative approaches to restoration in ATM networks. Some of them evolved from those of conventional networks (e.g., Synchronous Digital Hierarchy), while others have been initially proposed for ATM networks. Restoration schemes can be classified using several parameters [6, 10] and the purpose of this subsection is to explain this classification.

### 3.1.1 Centralised Restoration versus Distributed Restoration

Centralised Restoration schemes use a single management system to perform all restoration functions, which include failure detection, selection of alternate routes, and path generation. Examples of this approach are FASTAR proposed for PDH networks [11, 12] and SUCCESS for SDH networks [13]. The basic mechanism of these typical centralized control schemes can be applied in ATM networks with minor modifications.

The algorithms with centralized control take an all-network view of the failure, and make it easier to optimise the restoration plan. As a result, network resources can be used more effectively compared to distributed restoration algorithms.

In contrast, restoration speed is relatively slow with the centralized control [14, 45]. This is because of the communication delay between the centralised controller and network elements, and the heavy processing load on the central management system. Accordingly, it may be difficult with this scheme to complete restoration within the two seconds limit required by most existing services for normal functioning [15]. Another disadvantage of centralised algorithms is their vulnerability to single (central) node failure. This makes them unsuitable for a tactical environment. Hence, only distributed restoration schemes are considered in this study.

### 3.1.2 Layer of Restoration (Physical Layer versus ATM Layer Restoration)

The restoration can be performed either at the physical layer or the ATM layer. Restoration at the physical layer is provided by protecting communication links by identical spare elements (for more information, see description of Automatic Protection Switching mechanism presented in section 3.2). Traffic can be switched to the spare element very quickly and with simple electronics. However, with this approach resources are used inefficiently, priority system support is not possible and restoration from node failures or multiple failures cannot be guaranteed.

ATM layer restoration techniques reroute individual calls around a failure site and are more suitable for tactical networks, because they are flexible, use resources more efficiently and can support a priority system. Several methods and algorithms that are of particular interest are described in sections 3.4 and 3.5 in details.

### 3.1.3 Type of Restoration (Real Time versus Pre-Planned Restoration)

There are two ways to implement restoration at the ATM layer: *dynamic restoration algorithms* look for alternative paths for disrupted connections recovery after a failure, while *pre-planned restoration algorithms* are realized by preparing backup routes beforehand and activating them when a failure occurs.

Pre-planned restoration techniques are costly in terms of storage space and pre-computations required. Their advantages are the ability to make more optimal use of

resources, and the relative simplicity of execution. Pre-planned restoration is usually faster, but it does not always guarantee successful restoration because of the changing conditions in a network.

Both methods are implemented at the ATM layer and are of particular interest for tactical environment.

### 3.1.4 Type of Rerouting (Span versus Path Restoration)

There are three main alternatives, when a restoration algorithm tries to reroute failed connections over the working facilities. These are:

- *path restoration* that provides complete path reconfiguration (alternate route is usually selected to be node disjoint from the original);
- *link* (or span, or line) *restoration* that diverts connections only around the failed link or node;
- *local-destination* approach that reroutes failed connections from node adjacent to the failure site to their destination (or source) nodes.

The link restoration techniques are often less efficient than path restoration in terms of bandwidth required because inevitably diverting the connection around a failed link produces an elongated version of the original end-to-end connection. Path restoration uses free bandwidth more efficiently, but it is slow when used in dynamic restoration algorithms.

Local-destination approach to restoration is not so widely used as the other two, because it inherits to a certain extent the disadvantages of both path and link restoration approaches and requires more complicated route search procedures.

## 3.2 Automatic Protection Switching

The Automatic Protection Switching (APS) is the simplest restoration mechanism that uses distributed control [6]. Initially, it was proposed for SDH and PDH networks [16, 17]. The APS is constructed on a set of working and backup communication links,

where traffic is switched from the failed working link(s) to pre-assigned backup link(s) upon failure detection. APS schemes are classified into three types:

- The 1 + 1 APS is based on pairing one working link with one backup link. The signal is transmitted on both links in parallel ("+" means parallel transmission). When the working link fails, only the receiver side node switches the connection from working to backup.
- The 1:1 APS also pairs each working link with one backup link, but signals are not transmitted on the backup link unless a failure occurs (":" means non parallel transmission). Therefore, when the working link fails, both the receiver side node and the transmitter side node switch the connection from working to the backup link.
- The m:n APS is an enhancement of 1:1 APS, and associates *m* working links with *n* backup links. Generally, *m* is bigger than *n*, so a backup link may be shared by several working links.

APS utilises the spare resources less effectively than self-healing techniques proposed for mesh network topologies because it does not share (or shares primitively) spare resources. Moreover, it cannot restore node failures, which are a primary concern for tactical networks. Hence, it can be concluded that APS is not a suitable restoration scheme for tactical networks.

### 3.3 Self-Healing Rings

Self-healing ring (SHR) is a high-speed restoration scheme for ring topology networks [6, 17]. Imposing a network ring topology allows the restoration algorithm to be simplified. When a failure occurs, traffic is simply switched in the backwards direction. As a result, high-speed restoration can be achieved (e.g., 50 msec in SONET networks). In terms of algorithm and construction details, SHR techniques implemented at the physical layer are similar to 1 + 1, or 1:1 APS. In the SDH/SONET layer, SHR can decrease the network cost and simplify the network management [6]. ATM layer SHR schemes have also been proposed in the literature [18, 19].

The SHR restoration techniques are also not suitable for tactical networks, because the topology of tactical networks is not a ring. Also, multiple node or link failures can fragment a ring-based network making some nodes unreachable.

## 3.4 Dynamic Restoration Algorithms

The largest group of algorithms refers to the class of distributed restoration algorithms proposed for networks with mesh topologies. These algorithms are of particular interest for us. They can be divided in two main groups: dynamic restoration algorithms and the algorithms of pre-planned restoration, and they can potentially be applicable in tactical environment. Consequently, these methods and algorithms are described in more details below.

Dynamic restoration algorithms have been developed for mesh network topologies and use flooding techniques to discover alternative paths for disrupted connection recovery after link or node failure. Three main approaches to dynamic restoration can be distinguished. These are the sender-chooser approach, the double-search self-healing method and the Komine algorithm.

### 3.4.1 Sender-Chooser Approach

Several dynamic restoration algorithms [20-22, 31, 46, 49] use the sender-chooser approach for failed connection re-routing after single link failures (Fig. 3.1).

In the event of a link failure, both nodes adjacent to the failure recognise the fault, and by some predetermined, but arbitrary, ranking of nodes, one node becomes a sender, while the other one becomes the chooser (Fig. 3.1a). The sender broadcasts request (search) messages, containing such fields as the failure identification information, the sender and the chooser ID, the requested bandwidth and a hop count field (Fig. 3.1b). The requested bandwidth field is used to collect information about the spare capacity on the route taken by the message.

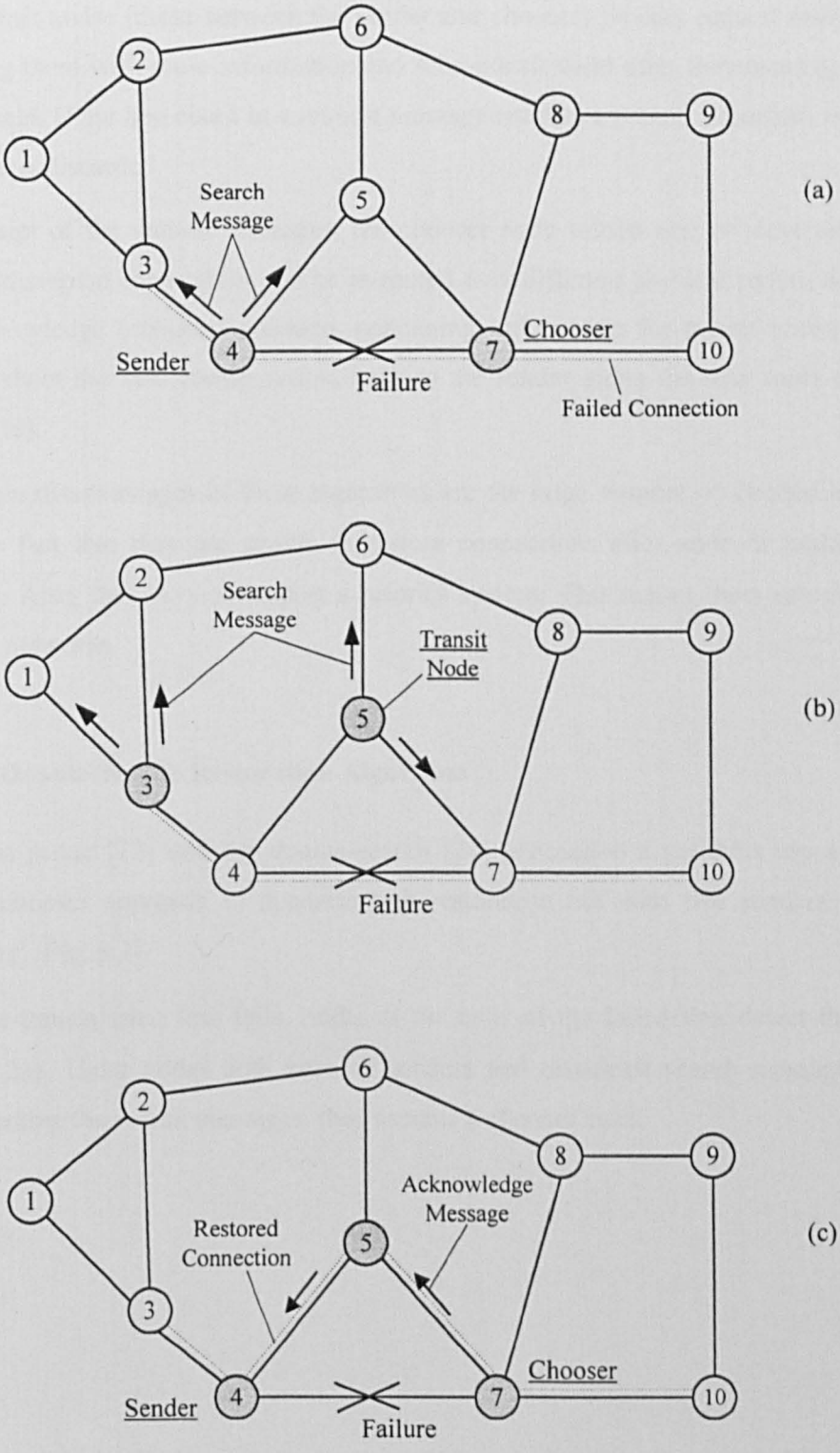



Figure 3.1 Sender-Chooser Approach to Dynamic Restoration.

The transit nodes (those between the sender and chooser) process request messages by updating them with route information and rebroadcast them after incrementing the hop count field. If the hop count in a request message reaches a pre-set maximum value, the message is discarded.

On receipt of the request messages, the chooser node selects one or more alternative routes (disrupted connections can be re-routed over different physical paths), and sends an acknowledge (connect) message, containing information for transit nodes and the sender about the new configuration, back to the sender along the new route or routes (Fig. 3.1c).

The main disadvantages of these algorithms are the large number of flooded messages and the fact that they are unable to restore connections after node or multiple link failures. Also, they do not support a priority system. This makes them unsuitable for tactical networks.

### 3.4.2 Double-Search Restoration Algorithm

The two prong [23] and the double-search [24] restoration algorithms use a similar, sender-chooser approach to dynamic link restoration but with two senders and two choosers[2] (Fig. 3.2).

When a transmission link fails, nodes at the ends of the failed link detect the failure (Fig. 3.2a). These nodes both become senders and broadcast search messages. After broadcasting the search messages, they assume a chooser state.

[2] Note that nodes have a dual role in this algorithm. First, they act as Senders, then they assume a Chooser state.

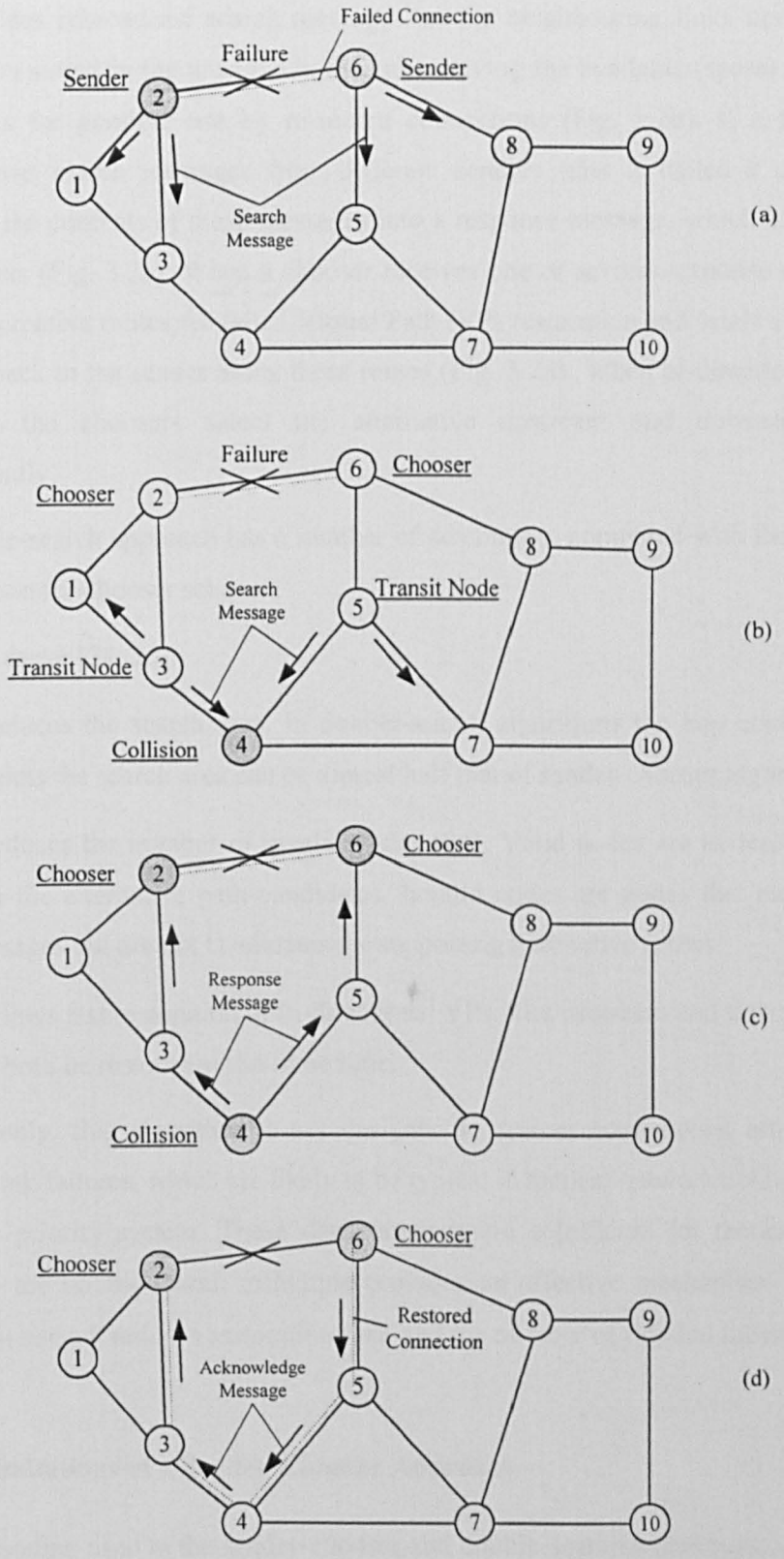



Figure 3.2 Double-Search Restoration Algorithm.

Transit nodes rebroadcast search messages on the neighbouring links updating route information stored in the message body and reserving the available (spare) capacity on these links for possible use by re-routed connections (Fig. 3.2b). If a transit node receives two search messages from different senders (this is called a collision), it combines the contents of these messages into a response message, which is returned to the choosers (Fig. 3.2c). When a chooser receives one or several response messages, it selects alternative routes for failed Virtual Path (VP) restoration and sends acknowledge message back to the sender along these routes (Fig. 3.2d). When bi-directional VPs are disrupted, the choosers select the alternative upstream and downstream paths independently.

The double-search approach has a number of advantages compared with the algorithms using the sender-chooser scheme:

- It is faster [24, 25].
- It reduces the search area. In double-search algorithms the hop count limit that restricts the search area can be almost half that of sender-chooser algorithms.
- It reduces the number of invalid nodes [24]. Valid nodes are nodes, which exist over the alternative path candidates. Invalid nodes are nodes that receive search messages but are not candidates for supporting alternative routes.
- It allows fast restoration of bi-directional VPs. The upstream and downstream VPs can both be restored at the same time.

Unfortunately, this algorithm is not designed to restore connections after node and multiple link failures, which are likely to be typical in tactical networks. Also it does not support a priority system. These disadvantages are significant for tactical networks. However, the double-search technique provides an effective mechanism for dynamic restoration since it reduces restoration time and the number of flooded messages.

### 3.4.3 Limitations of a Sender-Chooser Approach

Simple flooding used in the sender-chooser and double-search techniques cannot handle multiple-link or node failures because of the following problems:

1. *Fault location.* A node adjacent to the failure cannot distinguish between link or node failure. The Komine algorithm described below solves this problem using multi-destination flooding.
2. *Contention for spare capacity.* In the case of multiple failures, restoration messages coming from different nodes might contend for spare capacity on the same link. The simplest approach is to assign spare capacity on a first-come, first-served basis [26]. Another way to solve this problem is presented in section 3.4.5.

Consequently, new approaches or modifications to these techniques are necessary to provide restoration from node failures and multiple failures.

### 3.4.4 Komine Algorithm

The Komine algorithm [26] is the only dynamic restoration algorithm catering for restoration of both link and node failures. It uses a modified sender-chooser approach in which a sender node uses multi-destination flooding to the two preceding (upstream) nodes *for each failed path.* The IDs of two preceding nodes for every path traversing the node are determined beforehand by a path route monitoring procedure. Both these nodes are considered to be choosers and either, or both, can respond to the flooded request messages. Thus, link or node failures do not have to be distinguished because there is always at least one chooser.

The disadvantages of this approach are:

- less efficient handling of link failures (this is not important for tactical networks);
- for multi-destination flooding support, each node must store information about all the connections passing through the node;
- a large number of flooded restoration request messages are generated;
- an inability to support restoration from multiple node failures;
- the technique does not support a priority system.

The last three disadvantages are very important for tactical networks. However, the ability of the Komine algorithm to restore connections after single node failures could be exploited in tactical networks.

### 3.4.5 Dynamic Restoration Algorithm for Double-Link Failures

In [27] the dynamic restoration algorithm using the sender-chooser approach, which can also recover from double-link failures has been proposed. Its basic operations are as follows.

After two concurrent link failures, a node adjacent to each failure becomes sender and broadcasts request messages. Transit nodes rebroadcast these messages performing the same operations as in sender-chooser algorithms. If a transit node receives request messages from different senders, it assumes a contention state. It stores the request from one of the senders, for instance the sender with lower ID, and rebroadcasts the request message received from the sender with higher ID. When the restoration process initiated by the sender with higher ID is complete, this sender broadcasts a clear reservation message. When a contention node receives this message, it broadcasts the stored request message from the sender with lower ID to resume restoration of this failure. From then on, all standard steps of the sender-chooser algorithm for single link failures are performed.

The disadvantage of this algorithm is that it uses the sender-chooser scheme and therefore cannot recover from node failures. Also it does not support a priority system and generates a very large number of messages. However, this mechanism for handling multiple failures can be modified for use in tactical networks.

## 3.5 Pre-planned Restoration Algorithms

Pre-planned restoration algorithms have also been designed for mesh networks. Alternative routes are found beforehand, and activated after a failure occurs. This section introduces a number of these algorithms.

### 3.5.1 Virtual Path Protection Switching

In the self-healing schemes described in [28-30] every working Virtual Path (VP) is pre-assigned a backup VP that is selected to be node disjoint from the working one. The bandwidth of a backup VP is set at *zero*. Path restoration is realised by acquiring the required bandwidth along the backup VP from shared spare resources[3] when failure occurs, and redirecting cells from the failed VP to the backup VP.

When a link or node failure occurs in an ATM network AIS (Alarm Indication Signal) is sent to the downstream VP termination node (VP destination node) to notify it about the event. This node starts a restoration procedure, then. Three alternative backup VP activation protocols have been proposed.

#### 3.5.1.1 NTT approach

With the method proposed by NTT (Nippon Telegraph and Telephone Corporation) [28] (Fig. 3.3), the VP downstream node, which detects a failure, sends a confirmation message including the bandwidth required, along the backup VP and switches the failed VP to the backup VP (Fig. 3.3a). Transit nodes allocate the bandwidth if the spare capacity is greater than or equal to the bandwidth required by this VP, and retransmit a confirmation message to the next node along the backup VP (Fig. 3.3b). If the required bandwidth is not available along the entire route, an uncapturable message is sent back to the VP destination to start the cancelling process. When the upstream VP termination node receives the confirmation message, it switches traffic from the failed VP to the backup VP (Fig. 3.3c). This completes failure restoration for this VP. No other actions are proposed if there is insufficient spare capacity along a backup VP.

[3] Potentially, all available bandwidth on all links in the network can be used for restoration.

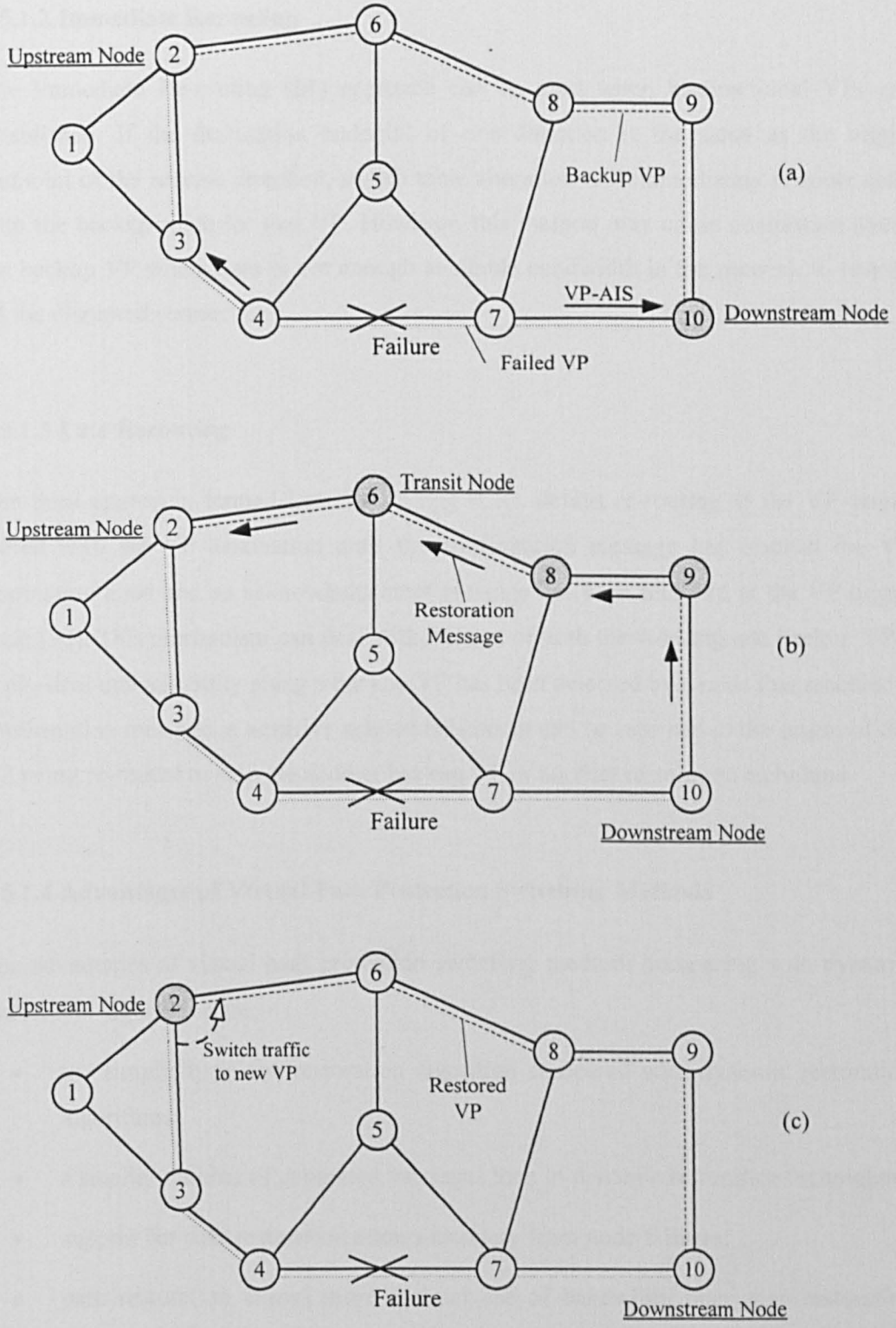



Figure 3.3 Pre-planned Restoration Algorithm.

### 3.5.1.2 Immediate Rerouting

The Immediate Re-routing (IR) approach can be used when bi-directional VPs are established. If the destination endpoint of one direction is the same as the origin endpoint of the reverse direction, switch table alteration will immediately re-route cells onto the backup route for that VP. However, this method may cause congestion along the backup VP when there is not enough available bandwidth in the network to restore all the disrupted connections.

### 3.5.1.3 Late Rerouting

The third approach, termed Late Re-routing (LR), delays re-routing at the VP origin paired with the VP destination until the confirmation message has reached the VP destination node and an acknowledgement message has been received at the VP origin node [30]. This mechanism can deal with failures of both the working and backup VPs. If physical unavailability along a backup VP has been detected by a node that received a confirmation message, a negative acknowledgement can be returned to the origin of the VP being re-routed to activate another backup VP or another restoration technique.

### 3.5.1.4 Advantages of Virtual Path Protection Switching Methods

The advantages of virtual path protection switching methods comparing with dynamic restoration algorithms are:

- the simplicity of the restoration algorithm compared with dynamic restoration algorithms;
- a smaller number of generated messages than in dynamic restoration techniques;
- support for path restoration allows recovery from node failures;
- path restoration allows more optimal use of bandwidth than span restoration used in dynamic restoration algorithms.

These advantages of VP protection switching make it a possibility for use in tactical networks. However, it must be modified to support a priority system and restoration

from multiple failures in the case when both working and backup VPs have been disrupted. Also the necessity of extra storage space and the burden of pre-computations may turn out to be significant disadvantages taking into account the limited resources and complex control problems of tactical networks.

## 3.6 Restoration Algorithms Supporting Connection Priorities

To distinguish restoration of different network services several techniques that realise multiple reliability level restoration have been proposed. They are of particular interest because they treat different classes of traffic differently, which is important for tactical networks. These approaches are QoS Restoration, Failure-Resistant VP Scheme, Hybrid Self-Healing Mechanism with VP Priorities, and Multiple Reliability VP Restoration.

### 3.6.1 QoS Restoration that Maintains Minimum QoS Requirements

With this approach, called QoS Restoration [32], customers at the call set-up phase define for each connection the QoS requirements that are used for normal operations as well as minimum (worst case) QoS requirements that can be applied in case of a failure. Accordingly, when a failure occurs resources of working connections are reallocated, that is reduced from normal down to minimum QoS level. This is necessary to get additional bandwidth that can be used for restoration along with free resources available initially.

This method can be used in conjunction with any conventional restoration algorithm described above with minor modification.

The main advantage of this approach is that it achieves a higher restoration ratio, if there are insufficient spare resources in the network after a failure. However, it assumes that QoS degradation is acceptable for customers. Consequently, this approach has the following shortcomings:

- This method does not take into account a priority system, though it can be easily modified to do so.

- It is not clear whether it is applicable to tactical networks, or not. Degradation of QoS requirements can be inapplicable for some (or many) types of traffic in tactical environment.
- This approach requires a lot of computational resources and a large number of messages to reallocate resources in a distributed manner.
- Even if the resources are successfully reallocated, full restoration cannot be guaranteed taking into account the severity of failures in tactical network and its low link capacities.

### 3.6.2 A Failure-Resistant Self-Healing Scheme

In [33] the concept of Failure-Resistant Virtual Path (FRVP) has been proposed. The similar method was also described in [37]. This approach is based on parallel transmission of cells between the VP source and VP destination nodes. Several VPs using usually node disjoint routes are established between source and destination. The source node duplicates user cells and transmits them on all these VPs simultaneously. As a result failure along one of the alternative VPs will not affect the other(s), and service interruptions will not occur.

This approach implements a very reliable data transmission scheme and provides high reliability. It can be potentially applied in tactical networks to traffic of highest priorities, but has several disadvantages:

- Extra traffic is generated. Taking into account a limited throughput of tactical networks this can be a very serious problem.
- Generation of disjoint paths and other management overhead (such as synchronisation of alternative VPs) are necessary.
- Multiple failures, which are possible in tactical environment, can disrupt even VPs that use independent routes. If this were the case, another restoration technique is still necessary.

### 3.6.3 Hybrid Self-Healing Mechanism with VP Priority

This approach proposed in [34 and 44] uses a modified VP Protection Switching method. The primary distinction of this scheme, called Hybrid Self-Healing, is that each VP is assigned some priority level. And upon failure detection the VP downstream node, which receives an AIS signal, schedules the restoration process according to the priorities of failed VPs.

To schedule restoration the hold-off timeout (HT) is used. The restoration of traffic of the highest priority is started immediately. After sending request messages for all disrupted VPs of the highest priority a node waits for the HT timeout before starting restoration process for the next priority level. The process is repeated until there exist any unrestored connections. The authors select the HT value according to the following equation:

$$HT = \overline{h} * (\overline{T}_l + \overline{T}_p) \qquad (3.1)$$

Here $\overline{h}$ is the average length of backup VPs in the networks, $\overline{T}_l$ and $\overline{T}_p$ are average link propagation and node processing delays respectively.

Other distinctions of this algorithm from the conventional VP Protection Switching algorithm are insignificant and do not influence parameters and performance of the restoration process considerably.

The main advantage of this approach is that the highest priority connections are restored first eliminating contention for spare resources with VPs of lower priorities. However, it still has some shortcomings:

- Overall restoration time is increased by introducing extra delays (HT timeout).
- The algorithm cannot provide efficient restoration in case of high network load.

It is also supposed that new algorithms for tactical networks proposed below provide more flexibility and controllability than the proposed algorithm and better correspond to specific requirements of tactical networks.

### 3.6.4 Multiple Reliability VP Restoration

The algorithm proposed in [35] implements a modified VP Protection Switching approach to restoration. Its first characteristic feature is that a priority, called restoration probability, is assigned to each VP in the network. Also several additional functions are added to realise prioritised restoration of disrupted connections. The flow of the algorithm is as follows (Fig. 3.4).

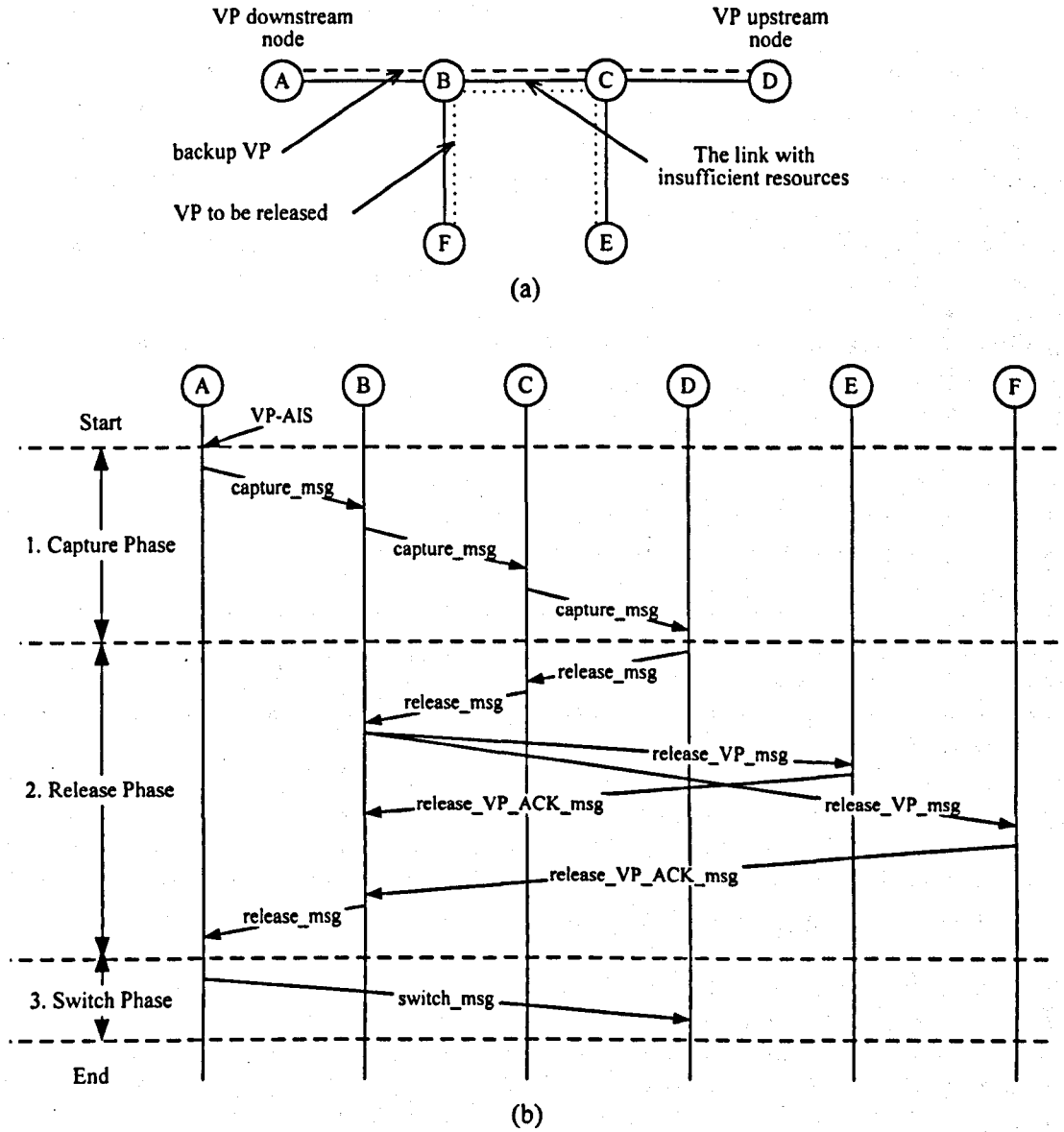



Figure 3.4 Multiple Reliability VP Restoration.

The first phase of the algorithm is referred to as a Bandwidth Capture Phase. When a downstream VP terminator node detects a failure by receiving an AIS signal, it starts a restoration process by sending a capture message along the backup VP. Transit nodes receiving a capture message attempt to reserve the appropriate bandwidth on the link. If it is available, it updates the message in a usual manner and sends it further along the backup VP. Otherwise, one or more lower priority VPs are selected and reserved to release their bandwidth in the next phase.

To control the restoration process and limit the bandwidth release two additional parameters are added to the capture message. $W_{sum}$ is the sum of priorities of VPs reserved for release, while $VPI_{bump}$ stores VPI values of these VPs. Also, the following rule is checked every time a node attempts to release low-priority VPs: $W_{sum}$ must not exceed the priority of the VP being restored.

In the second phase the bandwidth release process is executed. The phase is started when the upstream VP node receives a release message. If it is not necessary to release any lower-priority VPs, the transit nodes and VP upstream node work as in standard VP protection switching algorithm (see section 3.5.1).

If any VPs need to be released, the VP upstream node sends a release message along the backup VP back to the downstream node. Any transit node that receives a release message and needs to disrupt a low-priority VP, sends a release_VP message along this VP and waits for an acknowledge message. When VP is released and an acknowledge message is received by a transit node, it forwards release message further along the backup VP.

When the VP downstream node receives a release message, the last phase (Switch Phase) is activated by sending switch message along the backup VP and updating switch tables.

This approach provides the prioritised restoration, however it has a considerable disadvantage; it is very complex for a pre-planned restoration algorithm losing one of

the main advantages of this approach. In particular, the Bandwidth Release Phase can increase the restoration time considerably.

This restoration algorithm provides a specific understanding of priorities than is not applicable for tactical environment. Authors imply that in civil networks several low-priority connections can be more important that one high-priority, while it may not be true in military systems. Some assumptions of military networks researchers (see section 2.2 for details) allowed us to suppose that any connection of a higher priority is more important than any number of connections of lower priorities. Otherwise, Multiple Reliability VP Restoration approach needs further study and modelling.

Also, it is not clear that a single pre-planned restoration algorithm can meet the requirements of tactical networks, because pre-planned restoration cannot guarantee recovery from multiple failures.

## 3.7 Conclusion on Existing Restoration Techniques

The part of restoration algorithms' classification that is of particular interest in tactical networks is presented in Figure 3.5. The algorithms discussed in this chapter and described here in details are shown in this figure as well as two new restoration algorithms proposed for tactical networks (DRA-TN and PPR-TN) that are presented in details in Chapter 4.

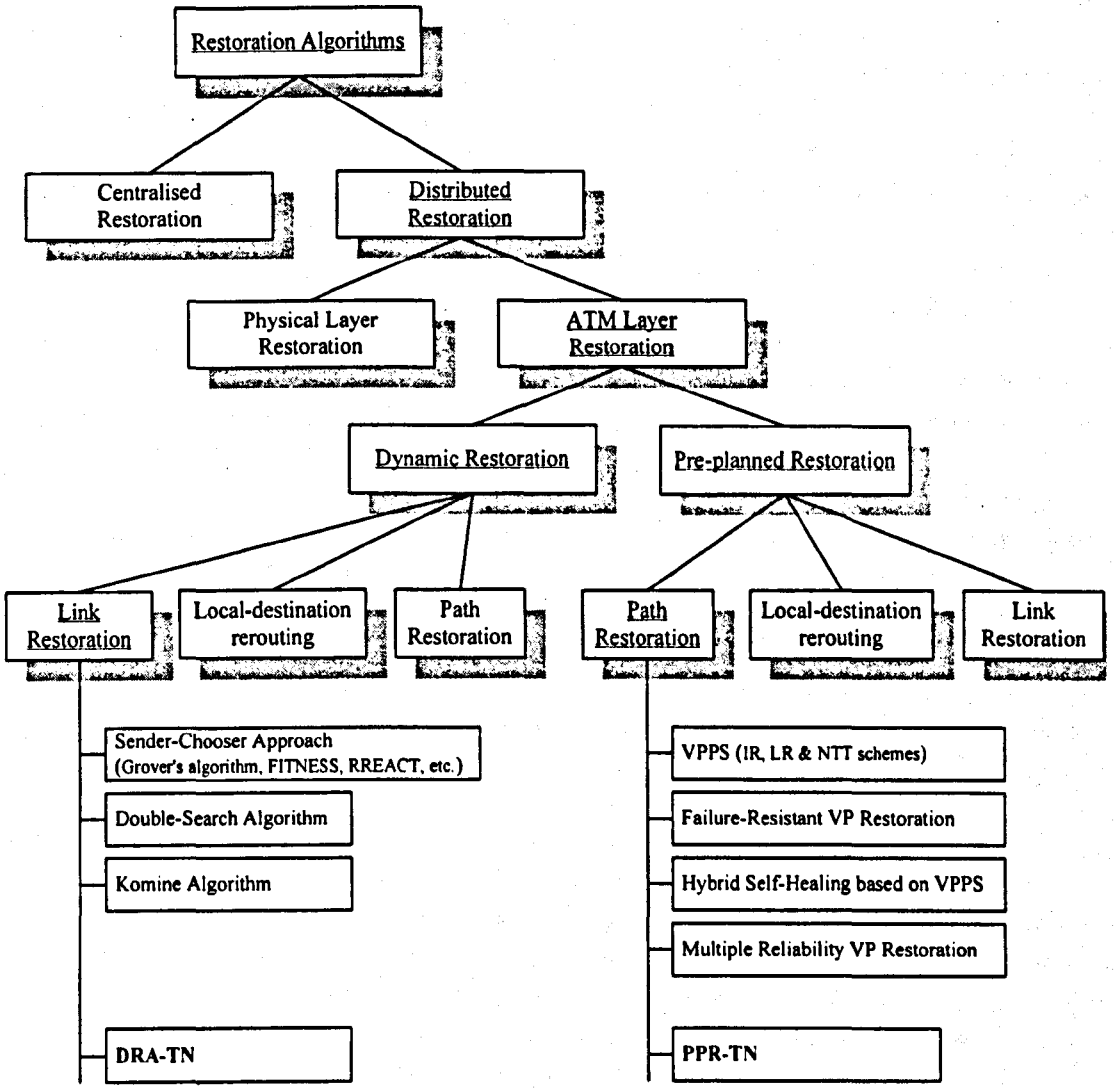



Figure 3.5 Classification of Restoration Algorithms.

The analysis of existing restoration algorithms has shown that they have the following disadvantages:

(i) Most *dynamic restoration algorithms* support recovery from single-link failures. Only the Komine algorithm is designed to support restoration from single-node failures. Modified sender-chooser algorithms can restore double-link failures, but no algorithm supports restoration from multi-node failures. Also algorithms using the sender-chooser approach generate a large number of restoration messages.

(ii) *Virtual path protection switching methods* support restoration from node failures. However, they cannot provide restoration in multiple failure scenarios if both working and reserve VPs have been disrupted.

(iii) The algorithms use bandwidth inefficiently. Single and multiple node failures, which are likely to be typical of tactical networks would require the provision of too much spare capacity to restore all affected connections [36].

(iv) Most algorithms do not support a full-scale priority system. There is only one algorithm, Multiple Reliability VP Restoration, fully supporting a priority system. However, it has several disadvantages as well (see 3.6.4 for more details).

Consequently, the conclusion from the initial research is that there are no restoration algorithms fully meeting the needs of tactical networks.

New algorithms for tactical networks are presented in the next chapter. It is supposed that these algorithms better correspond to the requirements of tactical environment, and can provide high performance.

# Chapter 4. Restoration Algorithms for Tactical Networks

Since no one restoration algorithm proposed for civil networks fully meets the requirements of tactical environment, two new algorithms are proposed in this chapter. One of them takes a dynamic restoration approach, while another one is a pre-planned restoration algorithm. Both of them are described in this chapter in detail.

It is argued that these algorithms better correspond to the requirements of tactical networks than known techniques, and can provide high performance.

## 4.1 Dynamic Restoration Algorithm for Tactical Networks

### 4.1.1 Background

The new Dynamic Restoration Algorithm for Tactical Networks (DRA-TN) has been designed specifically for the tactical environment. It is assumed that tactical networks are wireless networks created between mobile military vehicles and possibly a limited number of fixed nodes. The mobile vehicles take up positions in an exercise or war zone and establish the network whilst stationary. The protocol used on the wireless links between adjacent nodes will be ATM, with extensions to satisfy military requirements, and the network topology will be sparsely connected mesh with typical node connectivity being 3 or 4 (see section 2.1 for more details). The algorithm uses some features of the Double-Search Restoration algorithm, described in section 3.4.2, and the Komine algorithm (section 3.4.4).

The most likely failures in tactical networks are node failures and not link failures as in civil networks. Although a single link failure may occur because of the failure of a (directional) transmitter or receiver, the assumption is made that it is more likely that nodes will be spontaneously removed from the network as a result of enemy action. In any event, the proposed algorithms can deal with link or node failures but for illustration purposes the emphasis in this chapter is on node failures.

Tactical network traffic can be classified as voice, video and data, and each traffic class will have four priority levels. The restoration algorithm will attempt to restore all

priorities in all classes but if there is insufficient spare capacity to do this, Virtual Channel (VC) restoration will be in priority order, irrespective of class.

### 4.1.2 Introduction to Basic Assumptions, Terminology and Operating Procedures of Proposed Algorithms

Before describing the algorithm in detail it is helpful to explain some of the assumptions made in defining the algorithms, the terminology used to describe the algorithms and the basic operating procedures of the algorithms.

1. The aim of all restoration algorithms is to find alternative routes, which avoid the failure, for the disrupted traffic. In ATM, calls are established on separate outgoing and incoming VCs, creating effectively a bi-directional communications channel. Since each direction of a bi-directional communications channel (Fig 4.1a) is treated separately[4], restoring channels out-going from A to B via F (Fig. 4.1b) is a separate exercise to restoring channels out-going from B to A via F (Fig. 4.1c). When the failure occurs, the downstream node detects Loss of Signal (LOS) and initiates the restoration process on behalf of disrupted incoming uni-directional VCs.

[4] It is supposed that bi-directional channels between two nodes follow the same path, but in opposite directions.

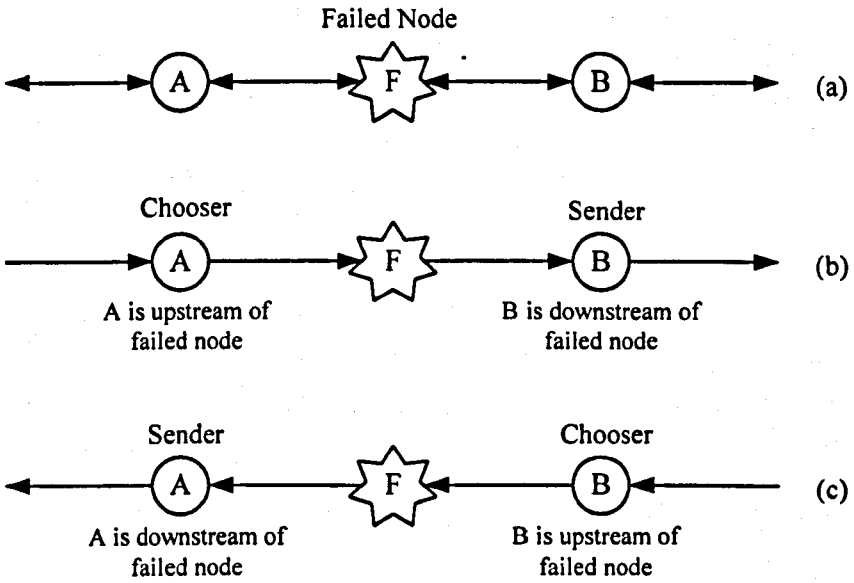



Figure 4.1 Identification of Upstream and Downstream Nodes.

2. The downstream node is called the Sender[5] node since this is the node that initiates the restoration process by sending search messages. The node upstream of a failure is called the Chooser node since this is the node that ultimately chooses to restore the VCs once an alternative route is found. Each single node failure will create several Sender nodes, depending on the number of nodes connected to the failed node, and for each Sender node there may be several Chooser nodes (upstream nodes through which the VCs established at the Sender node are routed).

3. The restoration process begins with all Sender nodes building search messages and flooding these through the network. The search messages are the means by which Sender nodes broadcast the identity of failed connections and the network bandwidth requirement needed to restore them. The flooding technique should ensure that at least some of the search messages eventually reach Chooser nodes.

4. For every VC, each node must have the identities of two previous upstream nodes along the route recorded in the switch table. This is so the Sender node can

[5] Please, note that here and below "Sender" and "Chooser" mean node status values, and sometimes "Chooser" node sends a message while "Sender" receives it.

identify the failed node (the previous node) and the Chooser node (the one before that again) for every disrupted VC. These nodes identifiers are known at call set-up time and the mechanism for recording these is assumed to be a modification of the normal call set-up procedure. Thus in Figure 4.1b, B is the Sender node and, in its search message, it will identify F as the failure and A as one of the chooser nodes. In Figure 4.1c, A is the Sender node identifying F as the failure and B as one of its Choosers nodes.

5. Flooded search messages pass through intermediate nodes and one or more of these messages will eventually arrive at the Chooser node. As described below, Sender and Chooser nodes then cooperate to select alternative routes through intermediate nodes for the disrupted traffic.

6. The algorithm will attempt to restore all connections after a single node failure or multiple failures with the following exception:

   VCs, which are routed through two neighbouring failed nodes (Fig. 4.2), for example A to B and B to A, cannot be restored because the Chooser nodes for these VCs (F2 for A and F1 for B) cannot respond. However, VCs between A and W, A and X, B and Y and B and Z, can (potentially) be restored. The search messages launched by nodes A and B, in Sender mode, will still identify F2 and F1, respectively, as Choosers (as well as other Chooser nodes for other VCs) but nothing will come of this search because the Chooser is unavailable. Note however it is possible to extend the algorithms to deal with multiple failures simply by storing details of more upstream nodes.

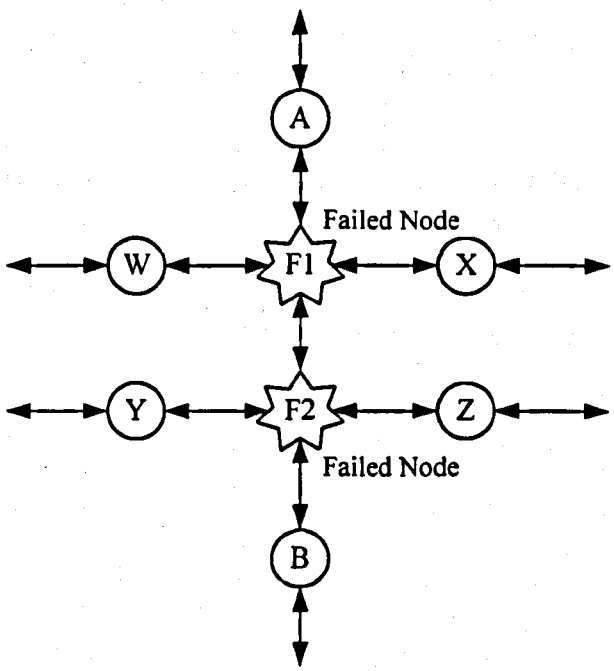



Figure 4.2 An Example of Multiple Node Failure.

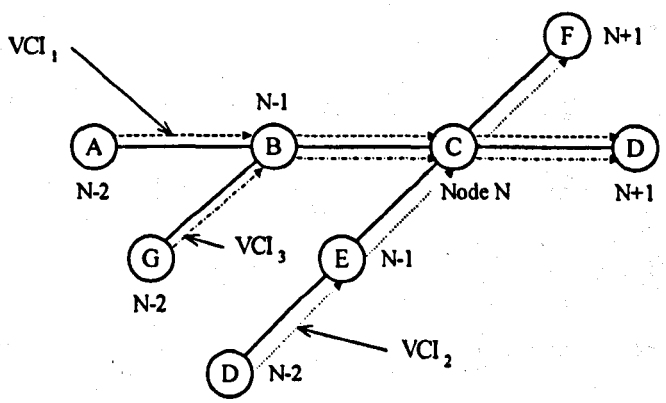
7. Every new call, which is accepted onto a link, has an equivalent bandwidth $B_{eq}$. We assume that $B_{eq}$ is the same for every link over which the call is routed. It is used in the Call Admission Control algorithm when the call is first placed and is entered into the switch table at call set-up. The other, essential, switch table entries are shown below.

8. It is assumed that every link has a known maximum bandwidth. This is the link bandwidth, which is available to carry user traffic before any calls are accepted. The available bandwidth from node A to B need not be the same as the bandwidth from B to A. $B_{eq}$ is deducted from the available (unassigned) link bandwidth when the call is accepted.

9. It is assumed that switch table format is as shown in Figure 4.3b. Node (N-1) (Fig 4.3a) is the immediate upstream node and similarly (N-2) is upstream of node (N-1) on this VC. When a failure at node (N-1) occurs, node N will scan the switch table and identify all the VCs, which are incoming from (N-1). There may be several different entries in the N-2 column (different choosers). The Sender will build and flood a search message, which contains all the Chooser IDs and the

bandwidth available on the link on which the message is transmitted. Clearly the messages sent on each link will be identical except for the link available bandwidth, which will be different for each link. After that, each Sender node changes its status to Chooser ready to accept search messages from Senders on the other side of the failure. For instance, if node C in Figure 4.3a notes a LOS failure at node B, it would search the switch table looking for entries 'B'.in the Node N-1 column. This identifies all the VCs coming through B. The N-2 column for these VCs will then identify the Choosers which must be contacted (nodes A and G in Figure 4.3a).

The format of the search message, and other messages, which are associated with restoration, and the actions, which follow the arrival of messages, are described below in the detailed description of the algorithm.




(a)

| VC Info | | Switching Info | | | | Restoration Info | | | |
|---|---|---|---|---|---|---|---|---|---|
| Priority | $B_{eq}$ | In VCI | Input Port | Output Port | Out VCI | Node N-2 (potential chooser) | VCI at N-2 | In Node (potential failure) | Out Node (node N+1) |

(b)

Figure 4.3 Switch Table format for node N.

10. (VCI at N-2) field in switch table for node N stores the VCI values at the node (N-2). This is necessary because new VCI values are allocated at every

node. However, in this study, for simplicity, we assume VCI has same values across all links of the connection.

11. Finally, it assumed that every node knows the identity of all its neighbours including the bandwidth available on link which connects them.

### 4.1.3 Detailed Description of the Dynamic Restoration Algorithm for Tactical Networks

#### 4.1.3.1 Step 1 - Failure Detection

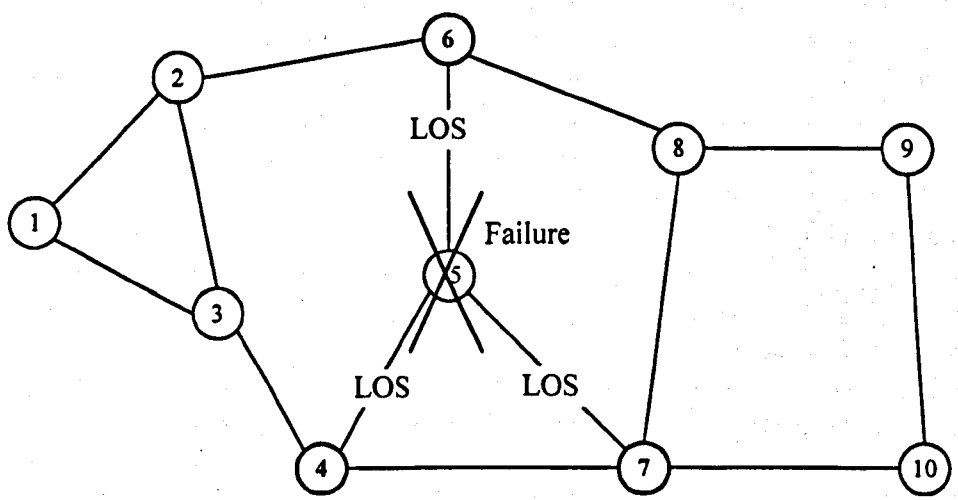



Figure 4.4 Failure Detection.

Suppose we have the following network (Fig. 4.4). Let us consider how the dynamic restoration algorithm for tactical networks works in case of node 5 failure. Suppose also, the following bi-directional connections are established at node 5 (Table 4.1):

Table 4.1 Connections established at node 5.

| Connection ID | Direction | Route |
|---|---|---|
| 1 | Forward (F) | 2 – 6 – 5 – 7 – 10 |
| | Backward (B) | 10 – 7 – 5 – 6 – 2 |
| 2 | F | 3 – 4 – 5 – 7 – 10 |
| | B | 10 – 7 – 5 – 4 – 3 |
| 3 | F | 3 – 4 – 5 – 6 – 8 – 9 |
| | B | 9 – 8 – 6 – 5 – 4 – 3 |
| 4 | F | 2 – 6 – 5 – 4 |
| | B | 4 – 5 – 6 – 2 |
| 5 | F | 3 – 4 – 5 – 7 – 8 |
| | B | 8 – 7 – 5 – 4 – 3 |

Then the neighbours of node 5 (failed node) have the following switch tables:

Table 4.2 Switch tables at nodes neighbouring to node 5.

| **Node 4** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C | D | in VCI | in Node | out Node | out VCI | Node N-2 | Priority | $B_{eq}$ |
| 2 | F | 230 | 3 | 5 | 230 | - | 2 | 10 |
| | B | 232 | 5 | 3 | 232 | 7 | 1 | 16 |
| 3 | F | 130 | 3 | 5 | 130 | - | 1 | 8 |
| | B | 131 | 5 | 3 | 131 | 6 | 3 | 12 |
| 4 | F | 454 | 5 | - | 454 | 6 | 3 | 14 |
| | B | 450 | - | 5 | 450 | - | 4 | 32 |
| 5 | F | 375 | 3 | 5 | 375 | - | 4 | 32 |
| | B | 377 | 5 | 3 | 377 | 7 | 2 | 12 |
| **Node 6** | | | | | | | | |
| C | D | in VCI | in Node | out Node | out VCI | Node N-2 | Priority | $B_{eq}$ |
| 1 | F | 120 | 2 | 5 | 120 | - | 3 | 10 |
| | B | 121 | 5 | 2 | 121 | 7 | 4 | 24 |
| 3 | F | 130 | 5 | 8 | 130 | 4 | 1 | 8 |
| | B | 131 | 8 | 5 | 131 | 9 | 3 | 12 |
| 4 | F | 454 | 2 | 5 | 454 | - | 3 | 14 |
| | B | 450 | 5 | 2 | 450 | 4 | 4 | 32 |
| **Node 7** | | | | | | | | |
| C | D | in VCI | in Node | out Node | out VCI | Node N-2 | Priority | $B_{eq}$ |
| 1 | F | 120 | 5 | 10 | 120 | 6 | 3 | 10 |
| | B | 121 | 10 | 5 | 121 | - | 4 | 24 |
| 2 | F | 230 | 5 | 10 | 230 | 4 | 2 | 10 |
| | B | 232 | 10 | 5 | 232 | - | 1 | 16 |
| 5 | F | 375 | 5 | 8 | 375 | 4 | 4 | 32 |
| | B | 377 | 8 | 5 | 377 | - | 2 | 12 |

Note that in theory these switch tables should have the format shown in Fig 4.3, however only information essential to the understanding of the restoration algorithm is shown here for simplicity. Note also the following remarks regarding Table 4.2 fields:

1. (Node N-2) - a potential chooser for a given VC.

2. Value "-" in (Node N-2) column means that the (in Node) node is terminator for this VC.

3. Value "-" in column (in Node) or (out Node) means that this connection is terminated at the current node.

4. (C) – auxiliary connection ID used only in the algorithm description to identify the connection. It is effectively a simpler VCI.

5. (D) – connection direction (one more auxiliary field). Again, this is used only in the algorithm description.

6. ($B_{eq}$) – equivalent bandwidth.

7. Note also, that the (N-2 VCI) value is also stored and processed by the algorithm. But for simplification, it is assumed that it is the same value as (inVCI), and thus is not shown in the table above.

When a failure occurs at node 5, nodes 4, 6, and 7 recognise it by detecting the LOS. They each assume the Sender state, and start creating search messages as follows. The search message format is given in Appendix A. The Sender node searches its switch table for rows where the in Node value is equal to 5. It then retrieves all the Chooser identifiers (Node N-2 values) from these rows; the Chooser identifiers are stored in the search messages. Other information such as Sender Node ID are also stored in the message. The Sender node then floods search messages on all ports updating them in the following way:

- ID of the node to which the message is sent is added to the list of Transit Nodes (this effectively stores the route that the message takes).
- Hop Limit Counter is decremented.

- Information about the Traffic load for the corresponding link is inserted into the message[6]. Note that this information relates to the opposite direction to which the message will travel. For instance, a message sent by node 6 to node 8 will store the traffic load information relating to the direction 8 to 6. The reason for this will be explained later.

For example, node 6 creates the following message to be sent to node 8:

Table 4.3 Search message sent by node 6 to node 8.

| Field Name | Value | Description |
|---|---|---|
| Message Type | 1 | Identifies Search message |
| Sender Node ID | 6 | Identify Sender and failure location. |
| Failed Node ID | 5 | |
| Number of Choosers (other Senders) | 2 | Chooser Identifiers<br>All information is taken from local switch table (see above). |
| Chooser Node ID | 4 | |
| Chooser Node ID | 7 | |
| Hop Limit Counter | 3 | Limits search area. |
| Transit Node | 8 | Transit nodes' identification, and information about the route 8-6 passed by the message. |
| Total bandwidth on the route | 256 | |
| $B_4$, $B_3$, $B_2$, $B_1$ | 16, 64, 128, 216 | |
| **End of message** | | |

Parameters $B_4$, $B_3$, $B_2$, and $B_1$ characterise the following traffic on the route passed by a search message:

- $B_4$ is the free bandwidth (i.e., bandwidth that can be used to restore connections of priorities 4 or higher),
- $B_3$ is free bandwidth ($B_4$) plus bandwidth occupied by $4^{th}$ priority traffic (this is the bandwidth that can be used to restore connections of priorities 3 or higher),
- $B_2$ is equal to $B_3$ plus bandwidth occupied by traffic of the $3^{rd}$ priority (this is the bandwidth that can be used to restore connections of priorities 2 or higher),
- $B_1$ equals to $B_3$ plus $2^{nd}$ priority traffic (this is the bandwidth that can be used to restore connections of the $1^{st}$ priority).

[6] It is supposed that node has information about traffic load of all outgoing links (it is necessary for routing). Therefore, link traffic information is inserted/updated before the message is sent on this link.

Here and below values of parameters $B_4$, $B_3$, $B_2$, and $B_1$ are selected arbitrarily, as in this description of the algorithm they are for illustration purposes only.

Table 4.4 shows another example of a search message sent by Node 7 to node 8:

Table 4.4 Search message sent by node 7 to node 8.

| Field Name | Value | Description |
|---|---|---|
| Message Type | 1 | Identifies Search message |
| Sender Node ID | 7 | Identify Sender and failure location. |
| Failed Node ID | 5 | |
| Number of Choosers (other Senders) | 2 | Chooser Identifiers<br>All information is taken from local switch table (see above). |
| Chooser Node ID | 4 | |
| Chooser Node ID | 6 | |
| Hop Limit Counter | 3 | Limits search area. |
| Transit Node | 8 | Transit nodes' identification, and information about the route 8-7 passed by the message. |
| Total bandwidth on the route | 512 | |
| $B_4$, $B_3$, $B_2$, $B_1$ | 64, 96, 164, 256 | |
| **End of message** | | |

Accordingly,

- messages flooded by node 4 will arrive to nodes 3 and 7;
- messages flooded by node 6 will arrive to nodes 2 and 8;
- messages flooded by node 7 will come to nodes 4, 8 and 10.

This completes the first step of the dynamic restoration algorithm for tactical networks.

#### 4.1.3.2 Step 2 - Rebroadcast of Search Messages.

At this stage of the algorithm a search for alternative routes is performed.

When a node, which is not a nominated Chooser receives a search message (Fig. 4.5), it is deemed to be a Transit node. Transit nodes store message information, add their ID and data about traffic load on the corresponding link to the appropriate fields in the search message, and then rebroadcast it further. This is described in more details below.

For example, the event when node 8 receives a search message from node 6 (format of this message is given in Table 4.3) can be studied.

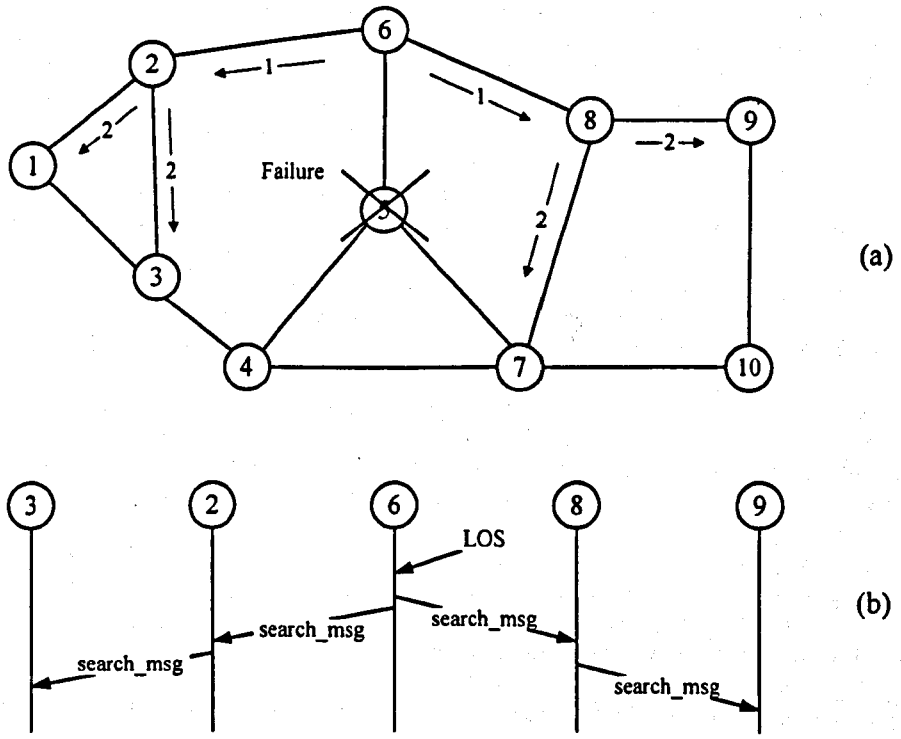



Figure 4.5 Rebroadcast of search messages.

In particular, Node 8 performs the following operations:

i. It saves the message in a database[7] - thus information about the routes from Sender to this node is available.

ii. A check for a possible collision is carried out.

Node 8 searches its database: if the Sender ID of the new message equals any potential Chooser ID stored in any previously received search messages (available in the database), such event is called a Collision, and it is described in section 4.1.3.3.

[7] The design of this database can be defined at the implementation phase, because it does not effect the overall algorithm operation.

iii. Node 8 checks a Hop Limit Counter field of the search message. If it is equal to zero, message is not forwarded further. Otherwise, the message is flooded further checking first that the routeing information does not create loops. This is done by checking the Transit Node information stored in the message; this gives the route that the message has already taken.

iv. Before sending the search message on any outgoing link it is updated in the following way:

- ID of the node to which the message is sent is added to the list of Transit Nodes.
- Information about the traffic load on the route passed by the message is updated.
- Hop Limit Counter is decremented.

Information about the traffic load stored in the message is adjusted as follows. For each priority, the minimum value is chosen between that stored in the message being processed and that associated with the outgoing. In this instance, assuming the values for the message being processed are those shown in row 1 of Table 4.5, and the values associated with 8-9 are shown in row 2, then row 3 shows how the message will be updated.

Table 4.5 Bandwidth information update.

| | Total bandwidth | $B_4$ | $B_3$ | $B_2$ | $B_1$ |
|---|---|---|---|---|---|
| Received Message | 256 | 16 | 64 | 128 | 216 |
| Outgoing Link Info | 512 | 32 | 48 | 102 | 228 |
| Updated Message | 256 | 16 | 48 | 102 | 216 |

This rule can be defined as

$$B_{iM} = \text{minimum of } [B_{iM}, B_{i9}] \qquad (4.1)$$

where i is the priority

M – is the traffic load information contained in the message being processed,

9 – is the traffic load information for the link connecting Node 9 to this node.

Consequently, we have the following updated message to be sent from node 8 to node 9:

Table 4.6 Updated search message sent by node 8 to node 9.

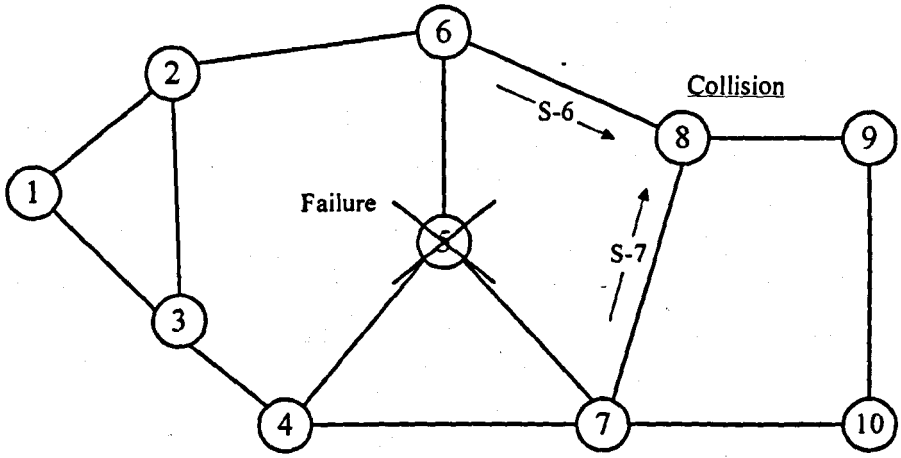
| Field Name | Value | Description |
|---|---|---|
| Message Type | 1 | Identifies Search message |
| Sender Node ID | 6 | Identify Sender and failure location. |
| Failed Node ID | 5 | |
| Number of Choosers (other Senders) | 2 | Chooser Identifiers. |
| Chooser Node ID | 4 | All information is taken from local switch table (see above). |
| Chooser Node ID | 7 | |
| **updated part of the message** | | |
| Hop Limit Counter | 2 | Limits search area (also identifies length of the route passed by message). |
| Transit Node | 8, 9 | Transit nodes' identification, and information about the route 9-8-6 passed by the message. |
| Total bandwidth on the route | 256 | |
| $B_4$, $B_3$, $B_2$, $B_1$ | 16, 48, 102, 216 | |
| **End of the message** | | |

Search messages at other nodes are processed in the same way.

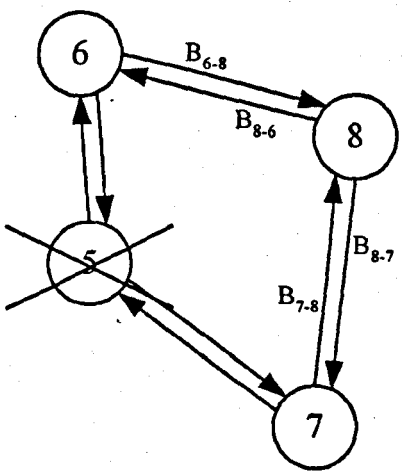
### 4.1.3.3 Step 3 - Collisions

#### 4.1.3.3.1 Collision at Transit node

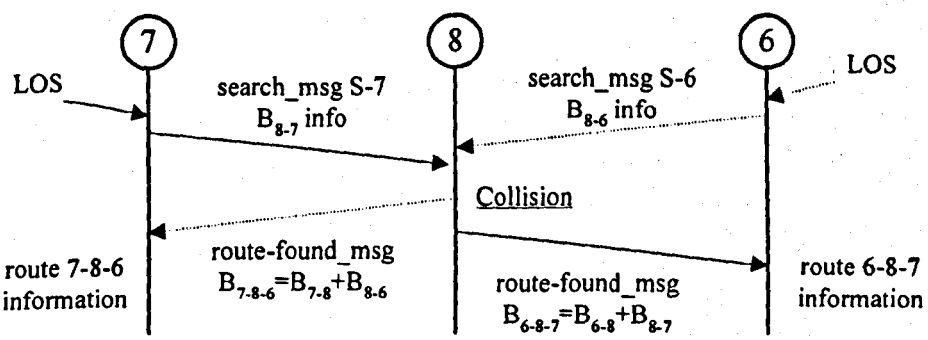
In the process of broadcasting search messages a Transit node may receive two search messages from different senders (Fig. 4.6a). If the Sender's ID of one message coincides with one of the Choosers' ID in another message, such event is called a collision. This means that an alternate route around the failed node is found, and the Chooser needs to be notified about it. Hence, the Transit node creates a route-found message and sends it to the Chooser.




(a)




(b)




(c)

Figure 4.6 Collision.

As an example, consider the scenario shown in Fig 4.6a, where node 6 sends a search message to node 8 and node 7 is noted as a possible Chooser. Similarly Node 7 sends a search message to node 8 and node 6 is noted as a possible Chooser; these search messages are shown in Table 4.7 and Table 4.8 respectively. After completing the standard search message processing operations described in 4.1.3.2, Node 8 checks its database and notes that node 7 is among the potential choosers for search messages from node 6. Effectively the 2 messages combined contain the routeing information for the restoration from 6 to 7, in both directions, around the failed node 5. The message from 6 also contains the traffic load information on the link from 8-6, similarly the message from 7 contains the traffic load information on the link from 8-7; this is shown in Fig 4.6c as $B_{8\text{-}6}$ info and $B_{8\text{-}7}$ info respectively.

Table 4.7 Search message sent by node 6 to node 8.

| Field Name | Value | Description |
|---|---|---|
| Message Type | 1 | Identifies Search message |
| Sender Node ID | 6 | Identify Sender and failure location. |
| Failed Node ID | 5 | |
| Number of Choosers (other Senders) | 2 | Choosers Identifiers |
| Chooser Node ID | 4 | All information is taken from local switch table (see above). |
| Chooser Node ID | 7 | |
| Hop Limit Counter | 3 | Limits search area. |
| Transit Node | 8 | Transit nodes' identification, and information about the route 8-6 passed by the message. |
| Total bandwidth on the route | 256 | |
| $B_4$, $B_3$, $B_2$, $B_1$ | 16, 64, 128, 216 | |
| **End of message** | | |

Table 4.8 Search message sent by node 7 to node 8.

| Field Name | Value | Description |
|---|---|---|
| Message Type | 1 | Identifies Search message |
| Sender Node ID | 7 | Identify Sender and failure location. |
| Failed Node ID | 5 | |
| Number of Choosers (other Senders) | 2 | Chooser Identifiers |
| Chooser Node ID | 4 | All information is taken from local switch table (see above). |
| Chooser Node ID | 6 | |
| Hop Limit Counter | 3 | Limits search area. |
| Transit Node | 8 | Transit nodes' identification, and information about the route 8-7 passed by the message. |
| Total bandwidth on the route | 512 | |
| $B_4, B_3, B_2, B_1$ | 64, 96, 164, 256 | |
| **End of message** | | |

Node 8 proceeds by forwarding route-found messages to both nodes 7 and 6. The route-found message to 7 is an updated version of the search message received by node 8 from node 6; this is shown in Table 4.9. Note that the message type has changed, all Choosers except 7 have been deleted, the complete route between Sender and Chooser is stored (by combining the routeing information from the 2 search messages which collided). All other information remains the same, including the traffic load information which has been gathered along the route from node 6 to node 8. Note that for simplicity the values $B_4, B_3, B_2, B_1$ will not be changed in this description of the algorithm, from this point on, although the reader is reminded that in realty the Minimum Function (4.1) is applied at each node along the route. Similarly, node 8 sends a route found message to node 6. This is an updated version of the search message received by node 8 from node 7 and is shown in Table 4.10.

Table 4.9 Route-found message sent by node 8 to node 7.

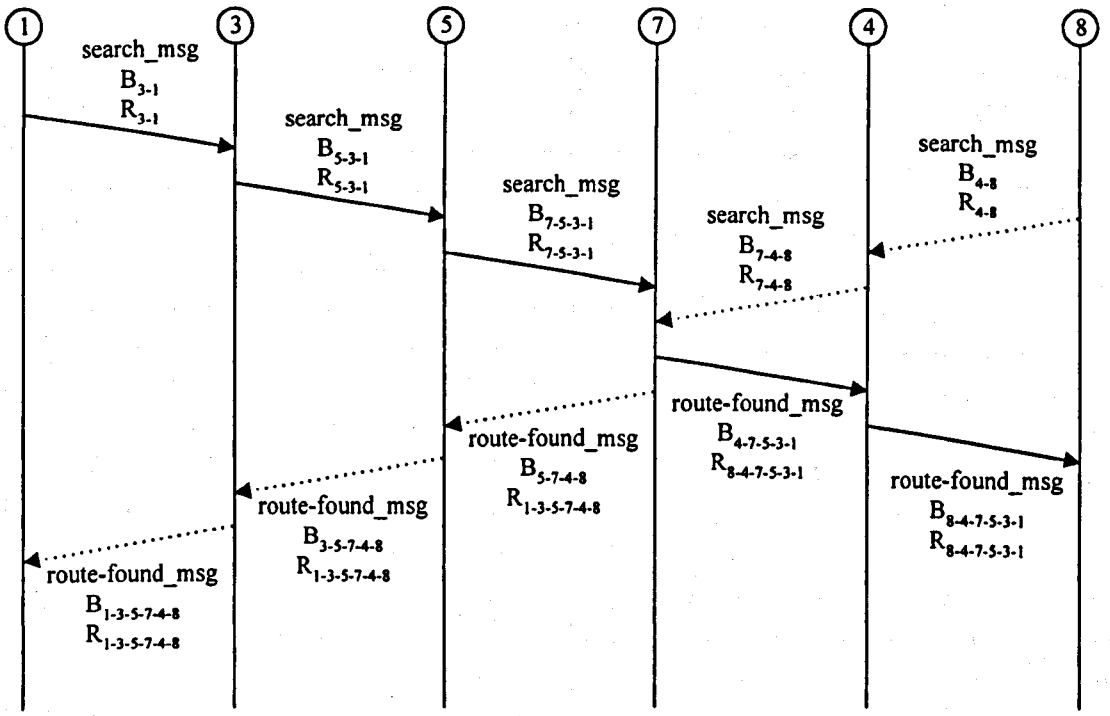
| Field Name | Value | Description |
|---|---|---|
| Message Type | 2 | Identifies route-found message |
| Sender Node ID | 6 | Identify Sender and Failure location. |
| Failed Node ID | 5 | |
| Chooser Node ID | 7 | Chooser for this restoration |
| Hop Limit Counter | 2 | Route length from transit node to chooser. |
| Transit Node ID | 8, 7 | Route 7-8-6 identification and description of the route 8-6 (amount of bandwidth that can be restored using this route). |
| Total bandwidth | 256 | |
| $B_4, B_3, B_2, B_1$ | 16, 64, 128, 216 | |
| **End of message** | | |

Table 4.10 Route-found message sent by node 8 to node 6.

| Field Name | Value | Description |
|---|---|---|
| Message Type | 2 | Identifies route-found message |
| Sender Node ID | 7 | Identify Sender and Failure location. |
| Failed Node ID | 5 | |
| Chooser Node ID | 6 | Chooser for this restoration |
| Hop Limit Counter | 2 | Route length from transit node to chooser. |
| Transit Node ID | 8, 6 | Route 6-8-7 identification and description of the route 8-7 (amount of bandwidth that can be restored using this route). |
| Total bandwidth | 256 | |
| $B_4$, $B_3$, $B_2$, $B_1$ | 64, 96, 164, 256 | |
| **End of message** | | |

A more complex collision scenario is shown in Figure 4.7. Here the search messages have travelled over a number of links before arriving at Node 7. The search message that arrives at node 7 originating from Node 8 stores the routeing and bandwidth information gathered along the route at each node; these are shown as ($R_{7\text{-}4\text{-}8}$) and ($B_{7\text{-}4\text{-}8}$) respectively. Note that the route-found message that is then sent by Node 7 to the Chooser (node 1) in response to the collision stores the full route between Node 1 and Node 8 ($R_{1\text{-}3\text{-}5\text{-}7\text{-}4\text{-}8}$) and that this is an amalgamation of the routeing information stored in the 2 search messages which collided. This information is used to route the route-found message to the Chooser Node 1 in much the same way as source routeing works. In contrast, the bandwidth information relates only to the path fragment ($B_{7\text{-}4\text{-}8}$) and that as the route-found message is forwarded along the second part of the route (from Node7 to the Chooser Node 1) the bandwidth information must be collected in the same way as explained earlier for search messages. Note also that when the route-found message arrives at Node 1 the bandwidth information relates to the route direction Node 1 to Node 8, hence the reason for collecting the bandwidth information in the opposite direction to which the message is travelling. This is necessary because Node 1 will restore the VCs in the direction Node 1 to Node 8.

Clearly the same procedure is executed for both directions.




$B_{A-B-C}$ - Information about the bandwidth on the A-B-C route.

$R_{A-B-C}$ - Route Information (the route A-B-C exists).

Figure 4.7 Example of a collision scenario.

#### 4.1.3.3.2 Collision at Sender node

A collision may also happen at a sender node. For example, node 7 can receive a search message originated at node 6 coming via node 8. Then, it contains the following information:

Table 4.11 Search message received at node 7.

| Field Name | Value | Description |
|---|---|---|
| Message Type | 1 | Identifies Search message |
| Sender Node ID | 6 | Identify Sender and failure location. |
| Failed Node ID | 5 | |
| Number of Choosers (other Senders) | 2 | Choosers identifiers<br>All information is taken from local switch table (see above). |
| Chooser Node ID | 4 | |
| Chooser Node ID | 7 | |
| Hop Limit Counter | 0 | Limits search area (also identifies length of the route passed by message). |
| Transit Node | 8, 7 | Transit nodes' identification, and information about the route 6-8-7 passed by the message. |
| Total bandwidth on the route | 256 | |
| $B_4$, $B_3$, $B_2$, $B_1$ | Bandwidth along the route 7-8-6 | |
| **End of the message** | | |

Node 7 checks its database containing information about failed connections and previously received messages, and identifies that node 6 is a potential chooser for some VCs. Therefore, it performs the following actions:

i. It initiates the restoration of the VCs to 6 by sending an acknowledge message to it (acknowledgement messages are discussed later in section 4.1.3.4).

ii. It sends a route-found message to node 6 (via node 8) to inform it that an alternative route is found. This message has the following format:

Table 4.12 Route-find message sent by node 7 to node 6.

| Field Name | Value | Description |
|---|---|---|
| Message Type | 2 | Identifies route-found message |
| Sender Node ID | 6 | Identify Sender and Failure location. |
| Failed Node ID | 5 | |
| Chooser Node ID | 7 | Chooser |
| Hop Limit Counter | 2 | Route length from transit node to chooser. |
| Transit Node ID | 7, 8, 6 | Route identification and description (amount of bandwidth that can be restored using this route). |
| Total bandwidth | 256 | |
| $B_4$, $B_3$, $B_2$, $B_1$ | Null | |
| **End of message** | | |

Note that the routeing information from the search message is stored in the route-found message and used to route it along the same path (but in the reverse direction) as the search message. Note also that the Bandwidth values are null. The nodes along the route will complete these in the same manner as described earlier for the search messages.

#### 4.1.3.3.3 Transit node receives a response message

When a Transit node receives a route-found message, it first checks if the same route has been processed already. Fig 4.8 shows how this could happen. Node 4 receives a search message from Node 8, and forwards it onto Node 7. Node 4 then receives a search message from Node 7 and a collision is noted. Node 4 responds to the collision by sending route-found messages to Node 7 and Node 8. At the same time Node 7 also detects collision upon receiving a search message from Node 4 and sends route-found messages to Node 4 and Node 5. When Node 4 receives the route-found message from Node 7, it will be discarded because Node 8 has been already sent a route-found message identifying route 8-4-7-5-3.

If this is not a duplicate route-found message for the route the Hop Counter is decremented. Also, the route information is updated in the same way as described in 4.1.2.1. Then, the node sends the updated route-found message to the next node on the route identified by the "Transit Node ID" field.

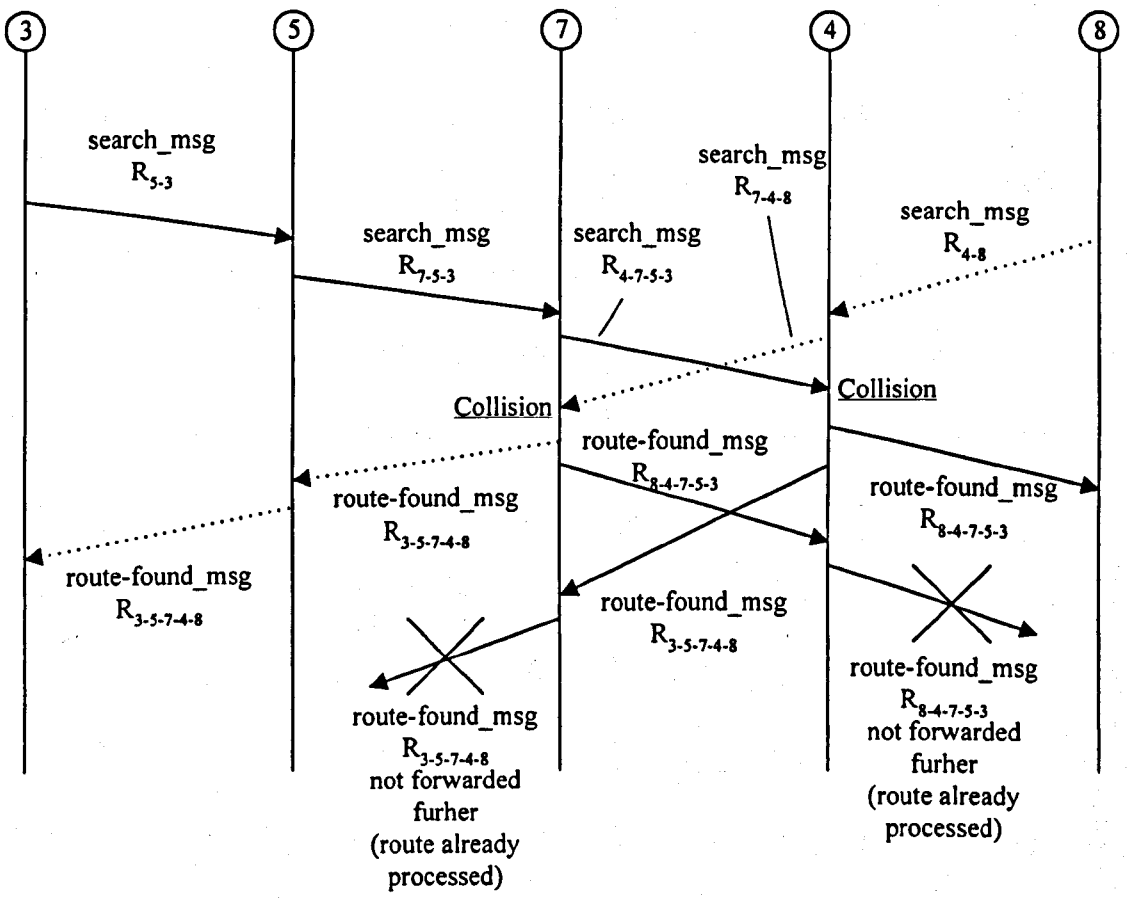



Figure 4.8 Duplicated route-found message processing.

### 4.1.3.4 Step 4 - Alternate Route Acknowledgement

#### 4.1.3.4.1 Chooser creates an acknowledge message

When a chooser node receives a route-found message it checks if the same route has already been processed, by checking the database of previously processed messages. If this is a new route, the chooser selects connections that can be restored via this route, starting with the high priority connections. Note that it uses the information about the priority and bandwidth requirements for the connections that are stored in the switch table at the node and the bandwidth available along the route, for each priority that is stored in the route-found message to decide which connections to restore. The chooser node then creates and sends acknowledge messages along the route to inform the nodes to modify their routeing tables (Fig. 4.9).

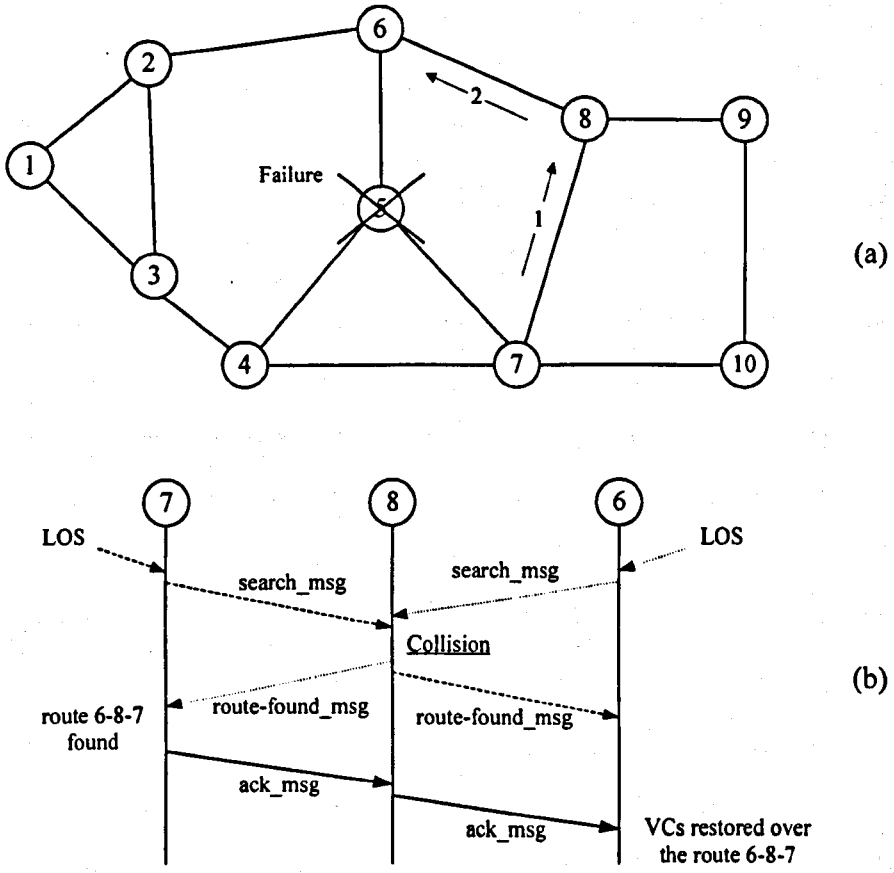



Figure 4.9 Alternate Route Acknowledgement.

For example, Node 7 creates the following acknowledge message (Table 4.13) in response to the route-found message after collision at node 8 (note that information about several connections can be transferred in a single message).

Not all the connection disrupted by the failure can necessarily be restored via a single route. Also, more than one alternate route can be found during the broadcast process. Therefore, the restoration for this Chooser node may not be completed at this step, and it may need to wait for other route-found messages (other routes found) for a predefined period of time. Only, after this timeout expires does the node clear its database, free the resources occupied by unrestored connections and return to the Normal state.

Table 4.13 Acknowledge message sent by node 7 to node 6.

| Field Name | Value | Description |
|---|---|---|
| Message Type | 3 | Identifies Acknowledge message |
| Sender Node ID | 6 | Identify Sender and Failure location. |
| Failed Node ID | 5 | |
| Chooser Node ID | 7 | Identifies Chooser Node. |
| Number of Connections | 1 | Identify connections being restored by this message. |
| Next VPI/VCI | 120 | Next VPI/VCI value to set in RT in the next node on alternate route |
| Sender VPI/VCI | 120 | Sender VPI/VCI value to set in RT at the Sender Node (identifies VC that is being restored at its destination). |
| Connection Priority | 3 | |
| Connection Bandwidth | 10 | |
| Hop Limit Counter | 2 | Route length. |
| Transit Node ID | 8 | New route identification. |
| **End of message** | | |

#### 4.1.3.4.2 Transit node receives acknowledge message

When a Transit node receives an acknowledge message, it checks if there is sufficient free bandwidth to restore given connection(s). The following situations can occur here:

(i) If the bandwidth required is available, the following information is added into the switch table (node 8 in this example):

Table 4.14 Switch table update.

| Incoming VPI/VCI | Outgoing VPI/VCI | Bandwidth | Priority |
|---|---|---|---|
| 120 | 120 | 10 | 3 |

(ii) If no free bandwidth is available, the node checks if there are any connections of lower priorities. If there are any such connections, they are disrupted to free bandwidth required to restore the connection of higher priority.

(iii) There is neither free bandwidth, nor low-priority bandwidth to accommodate the connection being restored. The cancel message is returned back to the Chooser node in this case to release resources occupied by this call.

#### 4.1.3.4.3 Assumptions

Here, it is assumed that:

1. Once the failure is known, no new connections are accepted until restoration is completed.
2. The resources on the route identified by search/response message exchange are not occupied by another process (e.g., providing restoration from another failure).

Otherwise, another additional mechanism needs to be implemented. For example, in case of multiple failures, information about each failure should be stored and processed separately. This can be easily achieved by storing and analysing 'Failed Node ID' field of restoration messages.

#### 4.1.3.4.4 Sender receives an acknowledge message

When a sender node receives an acknowledge message (e.g., node 6 receives an acknowledge message from node 8 originated at node 7), it modifies its switch table in the following way:

Table 4.15 Switch table update.

| | in VCI | in Node | **out Node** | out VCI | Node N-2 | VCI N-2 | Priority | $B_{eq}$ |
|---|---|---|---|---|---|---|---|---|
| Old Value | 120 | 2 | **5** | 120 | - | 120 | 3 | 10 |
| *New Value* | *120* | *2* | *8* | *120* | - | *120* | *3* | *10* |

Corresponding connections are restored.

#### 4.1.3.5 Time-out event at any node

When timeout set at the beginning of restoration process expires at any node in the network, this node removes all the information associated with restoration after this failure (except failure identification data). Other messages associated with this failure are discarded.

## 4.2 Pre-Planned Restoration Algorithm for Tactical Networks

### 4.2.1 Background

The late re-routing (LR) approach to VP Protection Switching (refer to section 3.5.1.3) currently cannot support priorities or multiple failures' restoration. Multiple Reliability VP Restoration algorithm also has a number of disadvantages, and it is argued that its performance can be improved. Therefore, a new algorithm based on the LR approach to restoration is proposed in this chapter. It supports connection priorities and can be modified to work at either the VP, or the VC level depending on the VP/VC structure (complexity) of tactical networks.

### 4.2.2 Algorithm Overview

After a failure is detected at the VC destination node a request message is sent along all backup VCs. The message includes both standard fields for VP Protection Switching, the attributes of the VC being restored (priority and the required bandwidth) and fields to collect information about traffic of different priorities along the backup VC.

If there is bandwidth at transit nodes occupied by traffic of lower priority, the node stores information about the priority and bandwidth of the VC being restored (i.e. identifies low-priority traffic for disruption) and passes the request message further along the backup VC. Otherwise (if no resources are available), a cancel message is returned to the VC destination node. A cancel message is also sent to the VC destination node if a failure along the backup VC is detected.

The VC origin node analyses request messages received and if the resources required to restore a given connection are available, it sends a confirmation (connect) message back along this VC.

Upon receiving the confirmation message transit nodes occupy necessary resources and update their switch tables with new VC information (low-priority connections are disconnected if necessary).

When the confirmation message reaches the VC destination node the restoration of the given VC is completed.

### 4.2.3 Algorithm Details

#### 4.2.3.1 Step 1 - Failure detection

Let us study how the pre-planned restoration algorithm works on an example of the standard Rudin Network in the case when there is a failure at node 5 (Fig. 4.10).

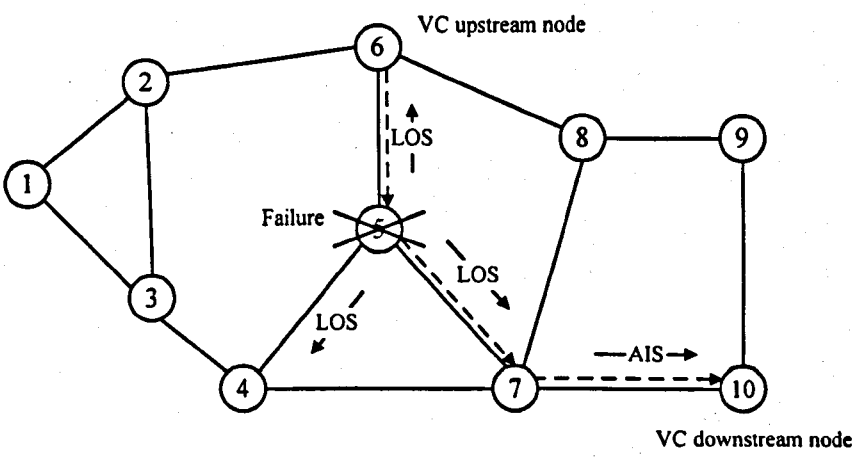



Figure 4.10 Failure Detection and Notification.

Suppose that the following VC connections traversing via node 5 are disrupted:

Table 4.16 Connections traversing via node 5.

| Value | Route | $B_{eq}$ | Priority | Backup VCI | Backup Route |
|---|---|---|---|---|---|
| 23 | 6 – 5 – 7 – 10 | 16 | 2 | 123 | 6 – 8 – 9 – 10 |
| 75 | 10 – 7 – 5 – 6 | 16 | 2 | 346 | 10 – 9 – 8 – 6 |
| 34 | 2 – 6 – 5 – 7 | 32 | 1 | 234 | 2 – 3 – 4 – 7 |
| 196 | 7 – 5 – 6 – 2 | 32 | 1 | 85 | 7 – 4 – 3 – 2 |
| 45 | 4 – 5 – 6 | 48 | 3 | 345 | 4 – 3 – 2 – 6 |
| 215 | 6 – 5 – 4 | 48 | 3 | 72 | 6 – 8 – 7 – 4 |

When a failure occurs at node 5, nodes 4, 6, and 7 recognise it by detecting the loss of signal.

It is assumed that nodes that detected a failure send AIS signal to VC downstream nodes to notify them about a failure. Then, nodes 2, 4, 6, 7, and 10 (VC destination nodes) recognise disruption of VCs. For instance, node 10 detects that VC 23 has been broken.

These nodes assume the Sender state and start checking backup routes for availability.

#### 4.2.3.2 Step 2 - Request message processing

To check if there are available resources along the backup route, Sender nodes (VC destination nodes) create request message for each disrupted VC, and send them along backup VCs (Fig. 4.11).

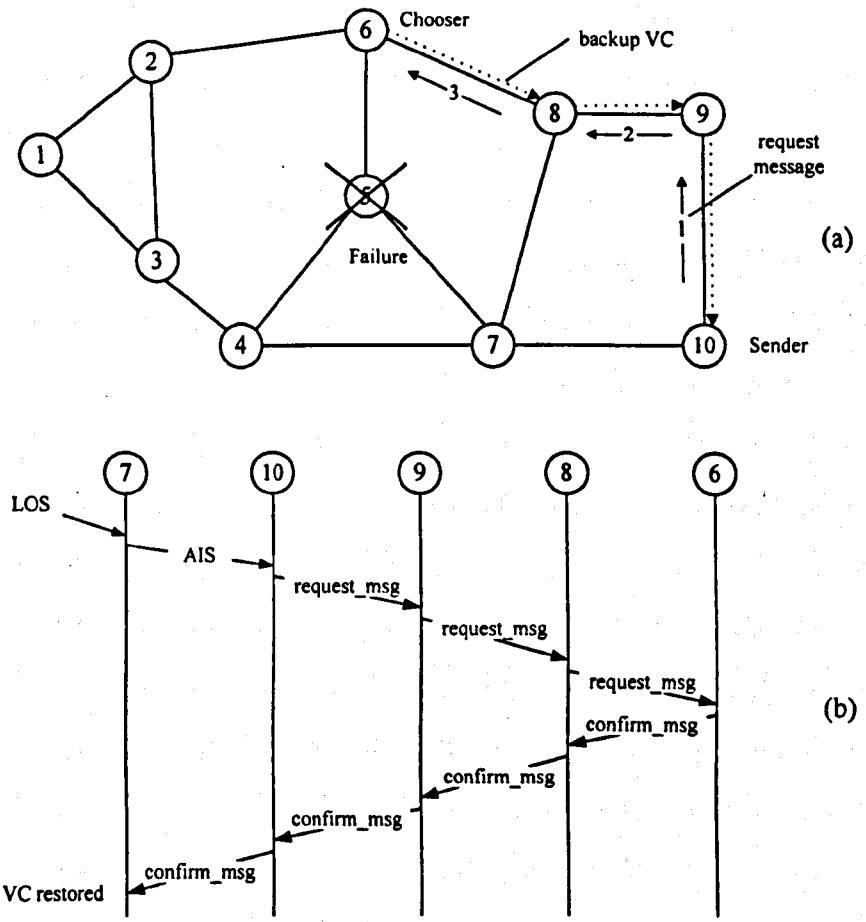



Figure 4.11 Normal Message Flow.

For example, node 10 creates the following request message for VC 23 (detailed message formats are presented in Appendix B

):

Table 4.17 Request message sent by node 10 to node 9.

| Field Name | Value | Description |
|---|---|---|
| Message Type | 1 | Request Message |
| Sender ID | 10 | Identify Sender and Failure location. |
| Failed Node ID | 5 | |
| Connection Priority | 2 | VC priority |
| Bandwidth | 16 | VC bandwidth |
| Backup VCI | 123 | Backup VCI value |
| Route length | 0 | Length of the route passed by this message. |
| Transit Node ID | 0 | Transit nodes' identification, and route information (it is taken from node's switch table). |
| $B_4$, $B_3$, $B_2$, $B_1$ | - | |
| **End of message** | | |

As in the case of Dynamic Restoration Algorithm for Tactical Networks parameters $B_4$, $B_3$, $B_2$, and $B_1$ characterise the traffic on the route passed by a search/request message:

- $B_4$ is free bandwidth (i.e., bandwidth that can be used to restore connections of priorities 4 or higher),
- $B_3$ is free bandwidth ($B_4$) plus bandwidth occupied by $4^{th}$ priority traffic (this is the bandwidth that can be used to restore connections of priorities 3 or higher),
- $B_2$ is equal to $B_3$ plus bandwidth occupied by traffic of the $3^{rd}$ priority (this is the bandwidth that can be used to restore connections of priorities 2 or higher),
- $B_1$ equals to $B_3$ plus $2^{nd}$ priority traffic (this is the bandwidth that can be used to restore connections of the $1^{st}$ priority).

In this case, $B_4$, $B_3$, $B_2$, and $B_1$ are zero.

Next, node 10 sends this message to node 9.

When a transit node receives a request message it performs the following operations:

(i) Checks if there is free bandwidth or bandwidth occupied by traffic of lower priorities on the link, which the message was received from[8].

(ii) If the requested VC cannot be restored because there is not enough available bandwidth or there is a failure along the backup VC, a cancel message is returned back to the VC destination node.

(iii) Otherwise, the node updates the request message with the link information[9], saves data about priority and bandwidth of the VC being restored in its local database (i.e. identifies low-priority traffic for disruption), and passes the request message further along the backup VC.

(iv) It also checks, if the VCI of the backup VC is changing at this node's switch table, and updates the corresponding field of the request message if necessary (i.e., the VCI value of VC origin node is backtracked)[10].

For example, when node 9 receives the request message from node 10 (Table 4.17), it updates the message in the following way before sending it to node 8:

Table 4.18 Request message sent by node 9 to node 8.

| Field Name | Value | Description |
|---|---|---|
| Message Type | 1 | Request Message |
| Sender ID | 10 | Identify Sender and Failure location. |
| Failed Node ID | 5 | |
| Connection Priority | 2 | VC priority |
| Bandwidth | 16 | VC bandwidth |
| **Updated part of the message** | | |
| Backup VCI | 123 | Backup VCI value |
| Route length | 1 | Length of the route passed by this message. |
| Transit Node ID | 9 | Transit nodes' identification, and route |

[8] Not only the switch table, but the database containing information about previously processed messages is checked as well. If available bandwidth have been already reserved for another high-priority connection restoration, it cannot be reserved again by a connection of the same or lower priority.

[9] Note that here we collect information about the VC going on the route 6-8-9-10, while the message is going in the backwards direction.

[10] Generally, VCI value can be changed at the VC switching node, but here for simplification we suppose that its value remains the same along the whole backup route.

| | | |
|---|---|---|
| $B_4, B_3, B_2, B_1$ | 16, 128, 196, 356 | information (describes traffic load on the route 9-10). |
| **End of message** | | |

The rules for updating the fields $B_4$, $B_3$, $B_2$, and $B_1$ at the next nodes of the route are the same as for Dynamic Restoration Algorithm for Tactical Networks (section 4.1.3.2):

$$B_{iM} = \min[B_{iM}, B_{iN}] \qquad (4.2)$$

where i is a connection priority (1..4),

M – means message being processed,

N – current link identifier.

If there was not enough bandwidth at node 9 (e.g., all the bandwidth on the link 9 – 8 is occupied by priority 1 traffic), cancel message returned to the node 10 would have the following format:

Table 4.19 Cancel message sent by node 9 to node 10.

| Field Name | Value | Description |
|---|---|---|
| Message Type | 3 | Cancel message |
| Transit Node ID | 10 | Identify Message Originator and Failure location. |
| Failed Node ID | 5 | |
| Connection Priority | 2 | VC priority |
| Bandwidth | 16 | VC bandwidth |
| Backup VCI | 123 | Backup VCI value |
| Route length | 1 | Route length. |
| Node IDs | 10 | Route identification. |
| **End of message** | | |

### 4.2.3.3 Step 3 - Backup VC Confirmation

The VC origin node (a Chooser node) analyses request messages received doing the same functionality as a Transit node. Then, if the requested resources are available, it updates its switch table and sends confirmation messages along backup VCs. For example, node 6 receives the following request message from node 8:

Table 4.20 Request message received by node 6 from node 8.

| Field Name | Value | Description |
|---|---|---|
| Message Type | 1 | Request Message |
| Sender ID | 10 | Identify Sender and Failure location. |
| Failed Node ID | 5 | |
| Connection Priority | 2 | VC priority |
| Bandwidth | 16 | VC bandwidth |
| Backup VCI | 123 | Backup VCI value |
| Route length | 3 | Length of the route passed by this message. |
| Transit Node ID | 9, 8 | Transit nodes' identification, and route information (describes traffic load on the route 8-9-10). |
| $B_4$, $B_3$, $B_2$, $B_1$ | 16, 128, 196, 356 | |
| End of message | | |

Node 6 updates the message with link 6-8 information, and if the requested VC can be restored via this route, node 6 sends the confirmation message along the route 6 – 8 – 9 – 10:

Table 4.21 Confirmation message sent by node 6 to node 8.

| Field Name | Value | Description |
|---|---|---|
| Message Type | 2 | Confirmation |
| Chooser ID | 6 | Identify message sender and Failure location. |
| Failed Node ID | 5 | |
| VCI | 123 | VCI of the VC being restored |
| Connection Priority | 2 | VC priority |
| Bandwidth | 16 | VC bandwidth |
| Route length | 3 | Route length. |
| Transit Node ID | 8, 9, 10 | Route identification. |
| End of message | | |

When a transit node receives a confirmation message it checks if the resources required are still available, updates its switch table and forwards the message further on the route. For example, node 8 updates its switch table:

Table 4.22 Switch table update.

| | in Node | in VCI | out Node | out VCI | Priority | $B_{eq}$ |
|---|---|---|---|---|---|---|
| Old Value | 6 | 123 | 9 | 123 | 0 | 0 |
| *New Value* | *6* | *123* | *9* | *123* | *2* | *16* |

If it is necessary, low-priority connections are disconnected.

### 4.2.4 Possible Algorithm Modifications

Several backup VCs can be assigned to high-priority working VCs.

This would require only to change the Step 3 of the algorithm. With this approach, the VC origin node waits for a predefined timeout for messages from all alternate routes. Then it analyses all the request messages received and selects one of the backup VCs for working VC restoration. The node then sends confirmation message along this VC and cancel messages along the others to free the reserved bandwidth.

## 4.3 Conclusions

In this chapter two restoration algorithms for tactical ATM networks were proposed and described in detail. These are Dynamic Restoration Algorithm for Tactical Networks (DRA-TN) and Pre-planned Restoration Algorithm for Tactical Networks (PPR-TN). Additional information about the algorithms such as message formats and block-scheme descriptions of the algorithms are presented in Appendices A and B.

# Chapter 5. Software Simulation

To test the performance and efficiency of the restoration algorithms for tactical networks presented in the previous chapter the following tasks need to be accomplished:

- consistency of the proposed algorithms (i.e. the ability to find the correct solutions and the absence of deadlocks, bas cycles, improper terminations [64]) needs to be proved;
- their performance characteristics in conditions typical for tactical environment need to be assessed and compared with those of known algorithms;
- the algorithms need to be tested on different network topologies as well as under traffic and failure conditions.

Building a model of the algorithms operating in the network is the most appropriate means by which these tasks can be accomplished.

The three techniques used to build models of communication networks are analytical, experimental and simulation. A critique of these is given in Section 5.1 and the conclusion is drawn that in this instance simulation is the most appropriate method. Hence the majority of this chapter is concerned with describing the design and implementation of the simulation package that was built to model the restoration algorithms and also to discuss aspects of the environment which have a critical effect on operation of the algorithms and thus on the output. The simulation package is covered in Section 5.2 and the Simulation Environment is the subject of Section 5.3.

## 5.1 Modelling Approaches

For communication networks modelling, analytical, experimental and simulation models are the three most common approaches [67]. Analytical models are mathematical and incorporate all the parameters which determine the behaviour of the real system. Results are then obtained by substituting values for the parameters in the model. Although in theory, an analytical approach can produce more accurate results, in practise it has a number of disadvantages. Firstly, the mathematical theory needed to

build a model may not exist. Secondly, if the problem is complex it is not always possible to derive an accurate model without making exaggerated assumptions. If too many assumptions are made then the mathematical model will not accurately represent the real system. An analytical model is not suitable in this instance because of the complexity of the system.

Taking an experimental approach, a working model of the system under investigation must be built and actual measurements can then be taken. This approach is seldom a viable option, since the reason for considering modelling in the first instance is to quantify some performance issues before taking a decision to build the system. Unless building the model is inexpensive, or there is no alternative, then this approach is seldom practical.

The disadvantages of an experimental approach are the advantages of a simulation approach. It is especially suited to comparing alternative designs, particularly if the alternatives do not differ much. Essentially only one model needs to be developed and variations of this can be produced. Furthermore, the use of Object Oriented design allows much of the program code to be reused; the benefits of this are explained more fully in the next section.

## 5.2 Modelling Software

A number of commercial simulation packages are available for modelling communication networks, for example OPNET, BONeS, COMNET, and others.

Initially, it was supposed by DERA that OPNET could be used for restoration algorithms modeling. Therefore the temporary license was obtained and the evaluation of OPNET was carried out to ascertain its suitability for modelling the restoration algorithms. It was found that the complexity of implementing the restoration algorithms was comparable with the complexity of using a high level language such as C, C++, Java, etc. While OPNET has many support libraries, these are useful mainly in the modelling of user traffic and this is of no benefit in this study, because user-defined algorithms need to be written in a special C dialect anyway. Furthermore the licence was very expensive and the problems with obtaining a long-term license were

encountered. Consequently, it was thus decided to build the simulation software using C++. This choice gave the benefits of OO, particularly code reusability, together with efficient execution that is necessary for simulation software.

### 5.2.1 Basic Assumptions

Restoration algorithms are concerned largely with the setting up and tearing down of connections. Storing and updating connection records in node switching tables can simulate these functions; this effectively extends the existing procedures related to connection set-up and termination. Hence, only the process of transferring restoration messages (OAM cells) between network nodes needs to be modelled and the user traffic is of no consequence in this study.

All the necessary statistics about the traffic can be calculated using information stored in switching tables, or on the data associated with restoration process.

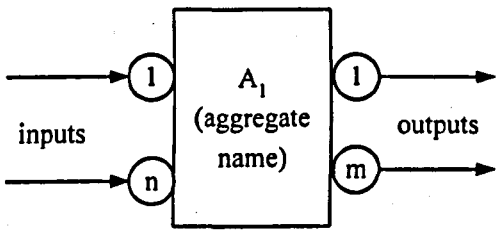
This approach simplifies the model and drastically improves the speed of simulations.

### 5.2.2 Model Description

A three-level model can be used for writing simulation software (Fig. 5.1a). This is a general approach for communication networks modelling used in other packages and systems (for example, in OPNET). At the highest level (level 1) the view of the system is at its most abstract, while the lower levels provide increasing levels of detail. Using this approach of refining the various levels of abstraction any system can be described by a general diagram consisting of interconnected blocks (Fig. 5.1b). Each block is characterised by its inputs, outputs, internal state and functions describing its behaviour. Complex blocks (subsystems) in turn can be represented as a set of next-level blocks. This process can be repeated several times until we get simple aggregate descriptions.

| Level | Detailisation | Objects | Description |
|---|---|---|---|
| 1 | Low | Environment, Network | Describes modelling conditions and external parameters. |
| 2 | Medium | Network Nodes | Network description (topology, etc). |
| 3 | High | Node Elements | Describes node structure and behaviour. |

(a)




(b)

Figure 5.1 Model Description: Abstraction Levels.

At the highest level of an ATM network model description (Fig. 5.2), the system can be considered as an interaction of the environment, which generates traffic and failures, and the network that processes the traffic and collects statistics.

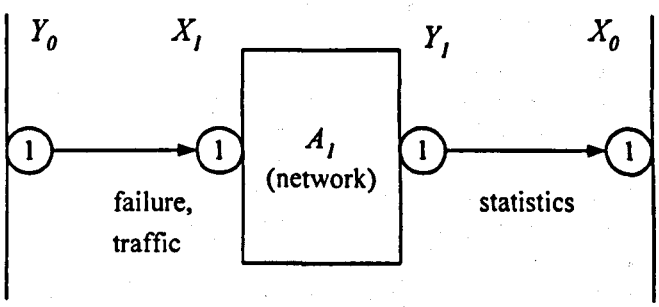



Figure 5.2 The highest level of system representation.

At the next level down a network is represented as a set of nodes interconnected by links (Fig. 5.3). This represents the next level of abstraction for the Block $A_1$ shown in Fig. 5.2. The network structure can be very different at this level. For example, here it consists of three nodes ($A_1$, $A_2$ and $A_3$), all interconnected by links and exchanging information between them and with the network object ($A_0$). Note that a node is a complex object, which needs to be described in more detail at the next abstraction level.

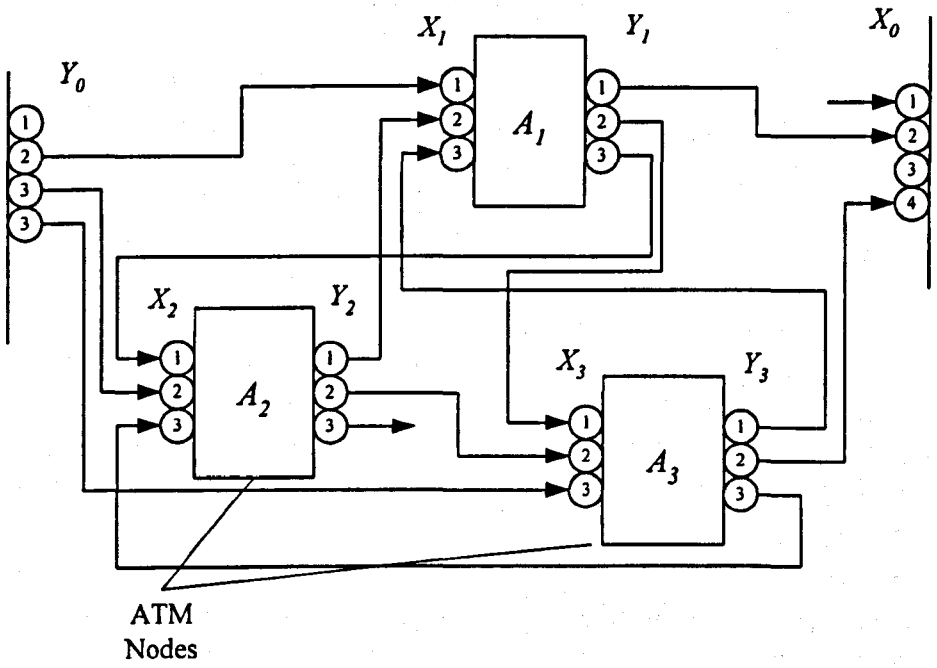



Figure 5.3 The second level of system description.

Consequently, at the third level a single ATM node can be represented as a subsystem consisting of a buffer, a switch table and a cell processing block (Fig. 5.4). This is a detailed description of the blocks A1, A2 and A3 shown in Fig. 5.3.

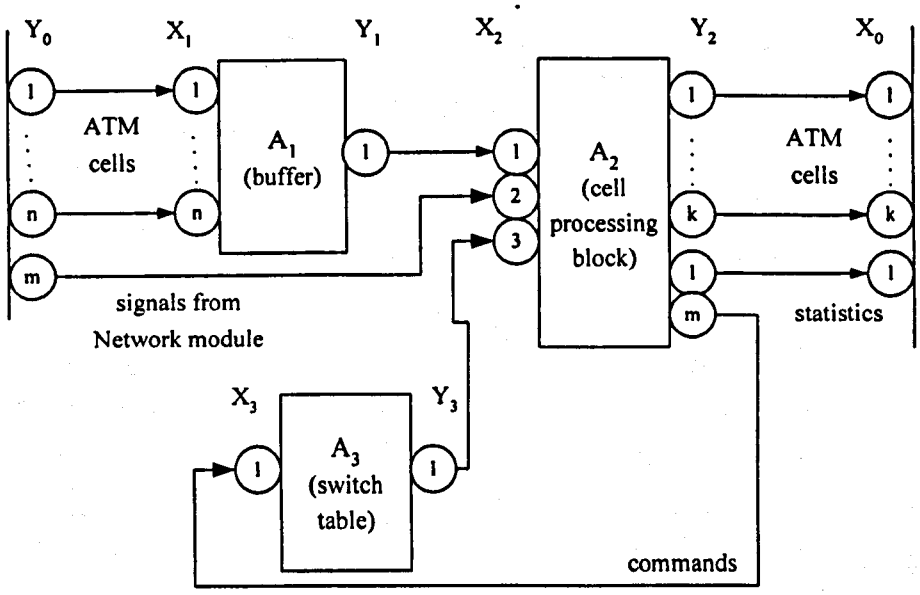



Figure 5.4 The third level of system description.

The functionality of these blocks can be described as follows:

- *Buffer functionality*: the buffer stores incoming cells from all the neighbouring (upstream) nodes in a queue. Every time a node is interrogated by the Network object it reads the first cell from the buffer, and calls the corresponding function to process it.
- *Switch table*: it is responsible for storing and processing the list of switch table records containing incoming and outgoing VPI/VCI values, bandwidth, and priority. It is normally used by the node to switch ATM cells from input to output links. In this study it is used in restoration process and for collecting statistics about traffic.
- *Cell processing block*: restoration algorithm functionality is implemented in this block. A flow diagram or state chart can describe any algorithm implemented in the system.

Thus, with this approach the second level is responsible for generation of network topology, traffic and failure scenarios. At the third level restoration algorithms are implemented. The advantage gained by using this abstraction approach is that only the cell processing block needs to be changed to implement another restoration algorithm. This has the advantage of minimising the implementation required. Though more importantly it helps maintain consistency between the models for the different restoration algorithms.

### 5.2.3 Implementation

The simulations are based on the discrete-event technique. A network is treated as a discrete-event system where the events of interest (message, time-out, failure, end-of-run) occur at discrete points in time. At any given time the state of the network is determined by the state of its components. For example, the state of a link is represented by its status, total bandwidth and the amount of occupied bandwidth.

The class diagram designed for this simulation model is given in Figure 5.5[11].

The seventeen classes shown can be divided into two main logical groups: classes common to all restoration algorithms that implement the model and classes specific for any particular restoration algorithm. The following groups of common classes can be identified:

(i) Classes SimEvent, SimCell, SimFailure, SimMessage and RouteInfo are used to implement basic blocks for event and message transfer mechanisms and define basic restoration message format.

[11] The primary aim of the class diagram was to show all the classes and their interaction. Therefore, some attributes and functions are omitted on the diagram due to the limited space available and to avoid overburdening it with excessive details.

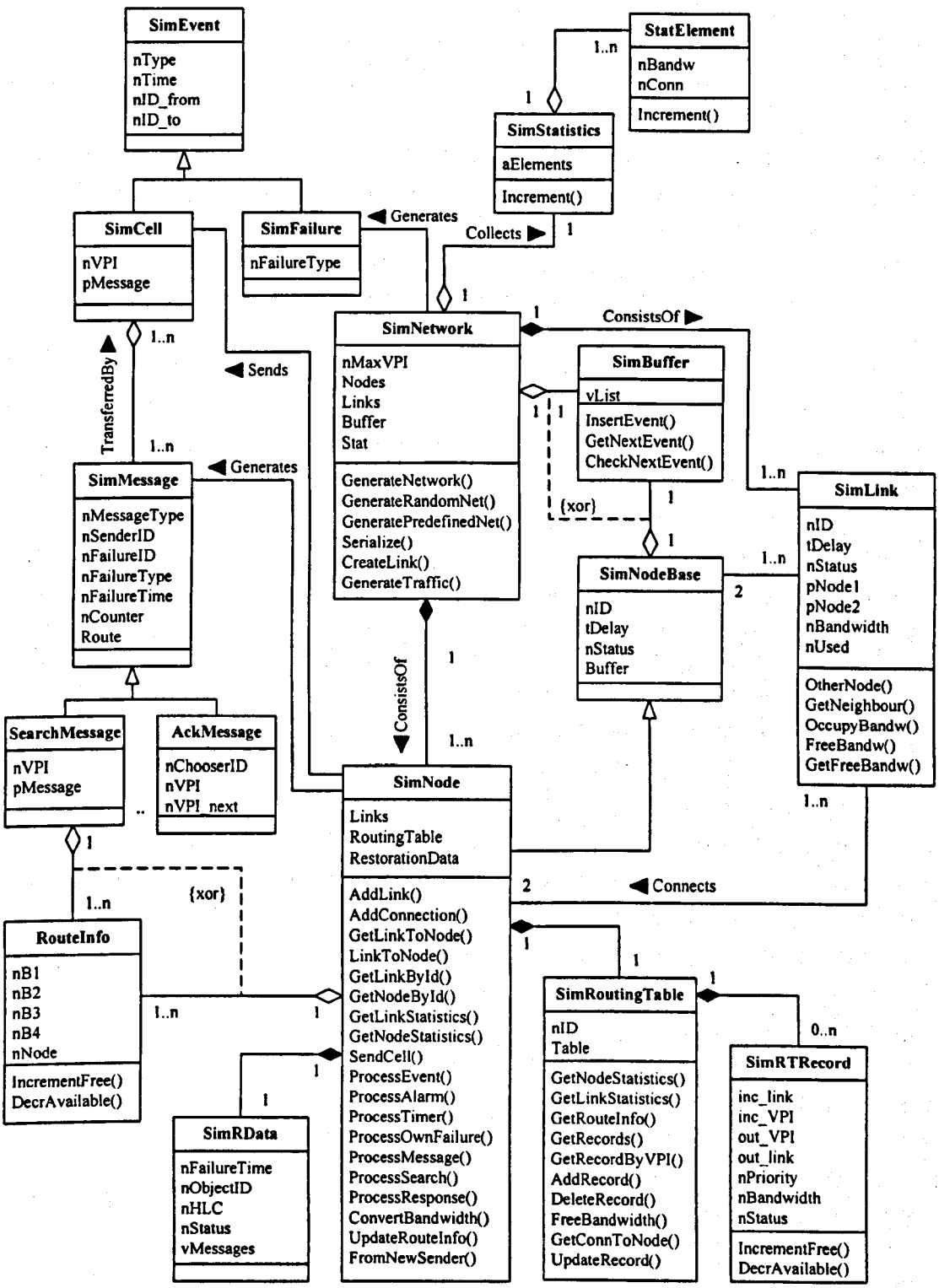



Figure 5.5 Class Diagram.

(ii) Classes SimBuffer, SimLink, SimNodeBase, SimRData, SimNode, SimRTRecord and SimRoutingTable implement event-driven simulations and behaviour of the basic network elements, and are used to store and process the corresponding service information. Also some functionality of SimNode class implements functions specific for different restoration algorithms.

(iii) Classes SimStatElement and SimStatistics implement statistics collection mechanism and are also the same for all algorithms.

(iv) Class SimNetwork is responsible for storing information about the network and coordinating the overall simulation process.

Event and message related classes (i) describe all the possible events that are processed in the system and all the message formats required for restoration algorithms that were implemented (Fig. 5.6). SimEvent is the parent class for all these classes. SearchMessage implements format of Search and Route-Found messages while AckMessage describes Acknowledge and Cancel messages for DRA-TN algorithm. RequestMessage[12] specifies format of Request, Confirm and Cancel messages for PPR-TN algorithm.

[12] This class is not show in the Fig. 5.5 for the lack of space. It is similar to classes SearchMessage and AckMessage and differs from them by different attributes only.

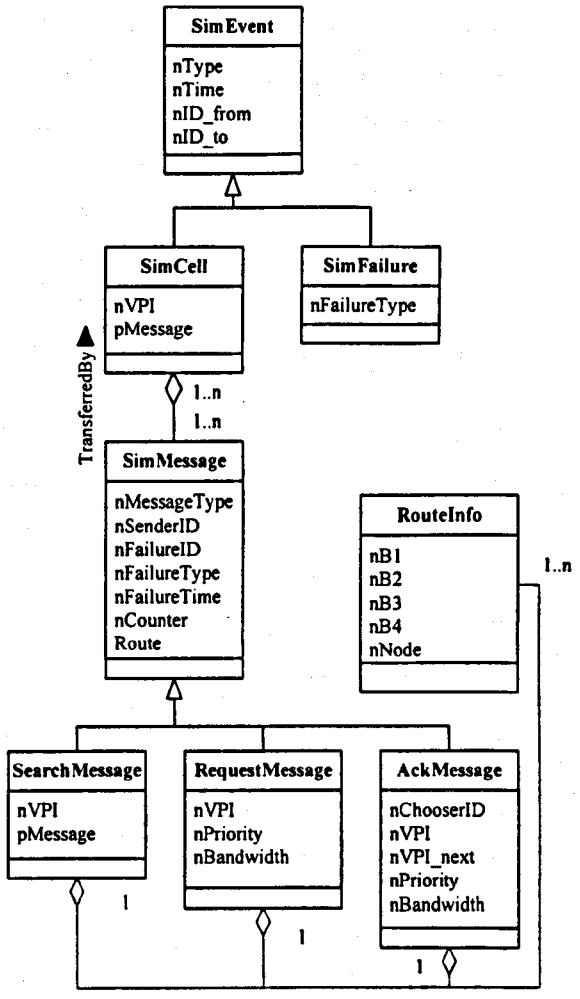



Figure 5.6 Event and message related classes.

Classes (ii) implement basic network elements functionality (Fig. 5.7). SimBuffer represents a queue of events in a chronological order that must be processed by the corresponding object. SimNode or SimNetwork object identifies the type of the object and calls corresponding methods to process them. If some method creates a new event, it is inserted into the corresponding object's buffer.

SimRData class implements a storage mechanism for data being stored at the ATM node during the restoration process. Effectively, this is status information and the list of messages processed by the node.

SimNode class processes events stored in its buffer. Its event- and message-processing methods correspond to the block $A_2$ in Figure 5.4 and this is the only block that needs to

be changed to implement a different restoration algorithm. SimNode also uses SimRoutingTable object to store, retrieve and process information related to connections set up at the node.

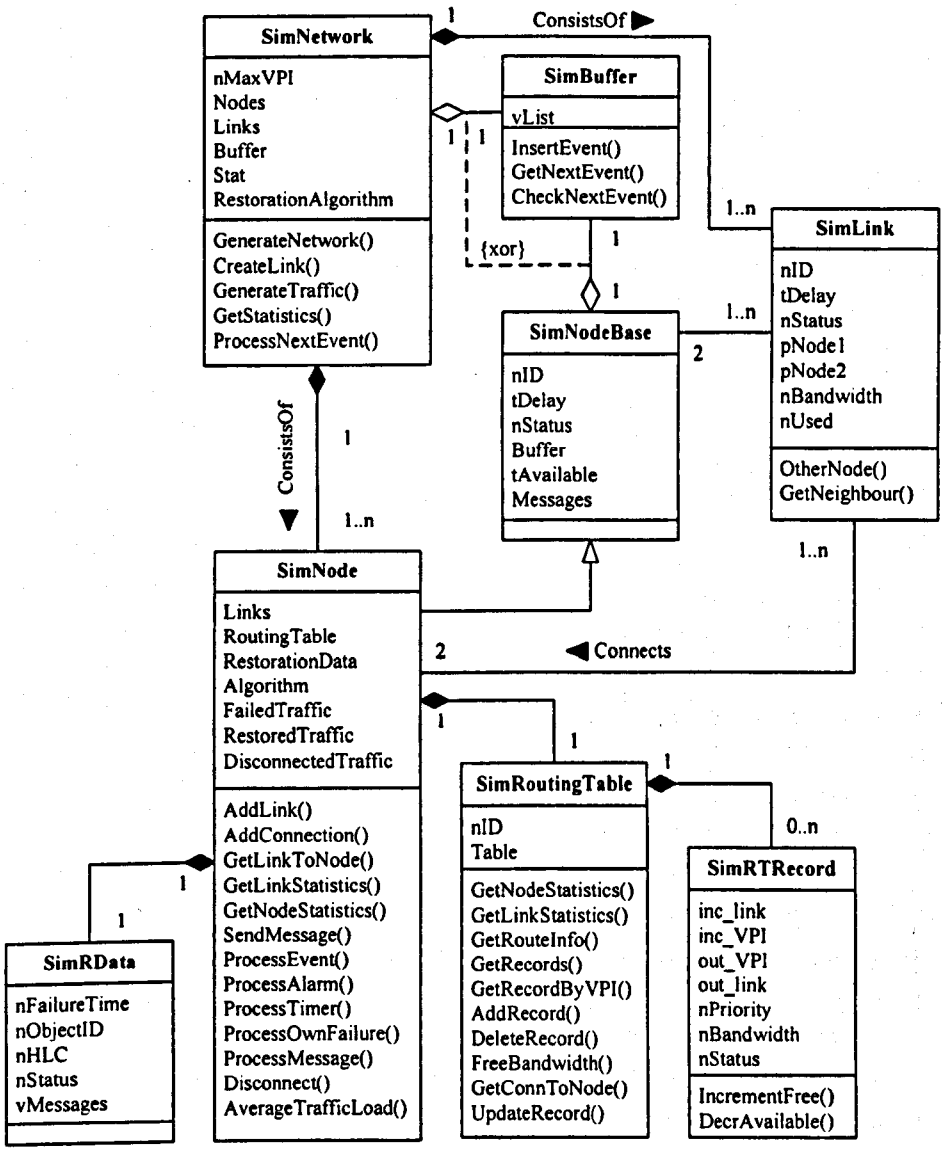



Figure 5.7 Classes implementing network elements.

SimNetwork class stores all the network information. It provides topology and traffic generation and is responsible for overall simulation process coordination. In each cycle of the simulations the SimNetwork object checks all other objects that can process events and have their own buffers. If the object is not occupied by processing another event, and there are events to be processed, the corresponding method is called. After all such objects are checked the system clock is incremented and the cycle is repeated until the restoration is completed or end-of-run event occurs.

The C++ programming language was used for the simulation system implementation. This powerful and flexible object-oriented language allows implementation of very complex systems and provides efficient means for programming systems, which require a lot of computations. The package consists of 31 source code files and about 5.500 lines of code.

### 5.2.4 Modelling Algorithm

The major steps necessary to run one experiment on any restoration algorithm are shown in Fig. 5.6. First, the network topology and traffic conditions are set or generated randomly, and the current network configuration is stored to be able to reproduce it at subsequent runs. Then the simulations are repeated several times for this configuration for various failure scenarios (different nodes' failures), and the total statistics on the experiment is calculated. This completes the single experiment of modelling a restoration algorithm for a given set of network topology, traffic conditions and failure type.

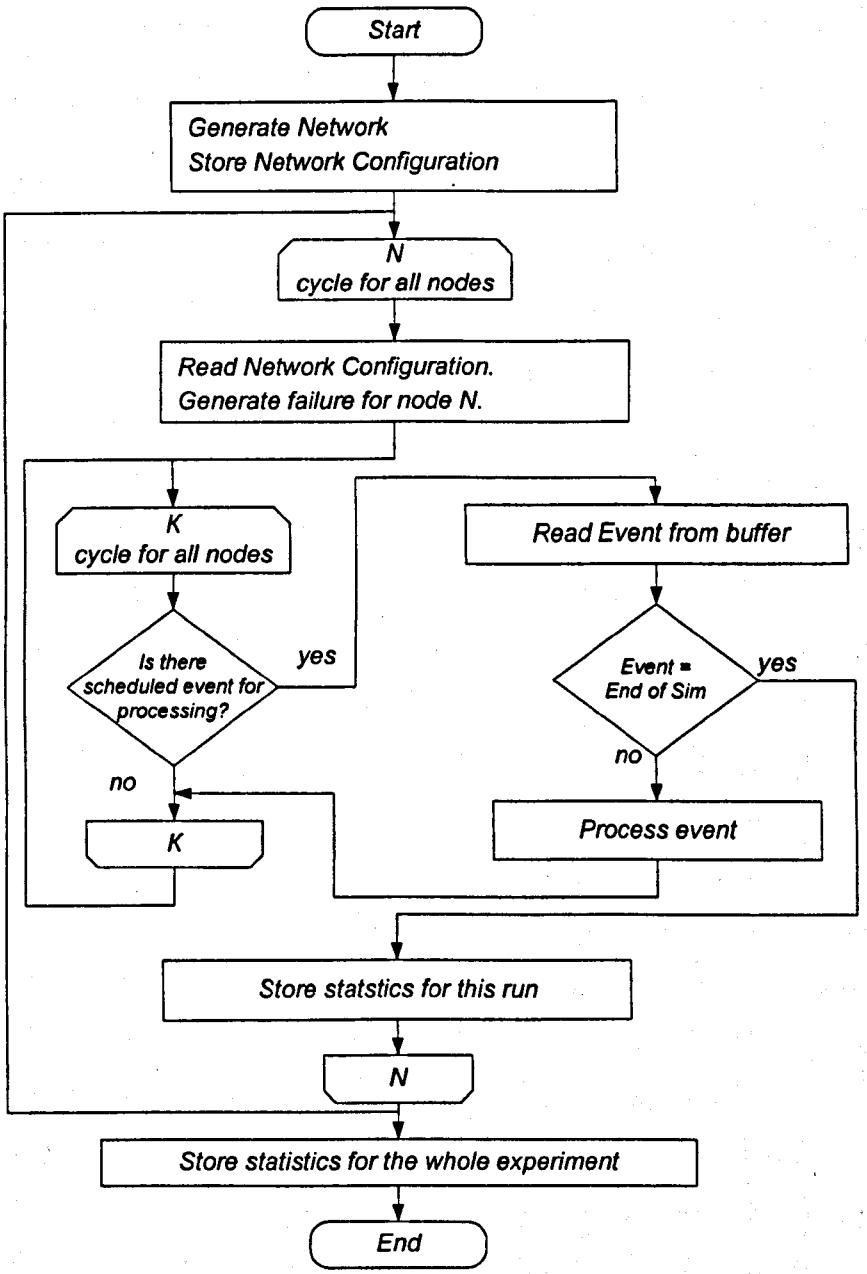



Figure 5.8 Modelling Algorithm.

### 5.2.5 Modelling Issues

#### 5.1.5.1 Network topology

The software package developed for modelling the restoration algorithms provides functionality both for entering predefined network topologies (such as US Network or Rudin Network), and for generating random network topologies using parameters such as connectivity and number of nodes.

#### 5.1.5.2 Traffic generation

Traffic information is stored in switch tables. The switch table typically has the following standard fields:

- incoming link id;
- incoming VPI/VCI value;
- outgoing VPI/VCI;
- outgoing link id;
- equivalent bandwidth.

A number of additional fields are necessary to model restoration algorithms. These are:

- priority;
- (Node N-2) - a potential chooser id for a given connection (zero value of this field means that the incoming node is a terminator for this connection - see chapter 3 for more details).
- (VPI/VCI N-2) - VPI/VCI value of the corresponding VC at the (Node N-2) node.

The assumption is that ATM tactical network traffic consists of bi-directional connections (occupying the same route in both directions) with asymmetric traffic characteristics. An equivalent bandwidth parameter is used to estimate the VC traffic amount.

Simulation software allows predefined traffic patterns to be entered (this is useful for modelling particular traffic scenarios) or randomly generate network traffic with different route lengths. The main parameter used to evaluate the traffic load is the network bandwidth utilisation (NBU):

$$NBU = \frac{\sum_i L_i}{\sum_i B_i} \qquad (5.1)$$

$B_i$ - bandwidth of link i,

$L_i$ - traffic load on link i

Connectivity and distance (number of hops) matrices are used to generate connections. The traffic is generated on the following basis:

- connections have the minimum and maximum length (some predefined values, which can be arbitrary chosen depending on network size);
- routes are randomly generated between different node pairs using the distance matrix and the reachability matrix;
- connection parameters (priority and $B_{eq}$) are randomly selected from the set of available values (see Table 5.3 below for more details);
- the process of generating connections is repeated until the network bandwidth utilisation or the number of connections in the network reach their limits selected for the given experiment.

#### 5.1.5.3 Statistics

The simulation model is designed to collect measurements of metrics typically used to assess the performance of restoration algorithms, as follows:

- Restoration time – the time required by an algorithm to achieve the required level of restoration [25]. In our case, this is the time from the moment of failure until the time the last connection is restored.
- Number of restoration messages generated by a restoration algorithm.

Note that it may not be possible to restore all the connections after a node failure. Therefore, the efficiency of the algorithm can be measured by comparing the network state just after a failure has occurred and after the restoration is completed. If high-priority connections have been restored (even if some of low-priority connection have been disrupted) that would mean that an algorithm worked successfully.

In civil networks the Restoration Level and Spare Usage [25] metrics are typically used to evaluate the performance of restoration algorithms. Restoration Level refers to how many failed connections are restored:

$$\text{Restoration Level} = \frac{\text{Number of restored connections}}{\text{Number of failed connections}} \qquad (5.2)$$

Spare Usage refers to how much free bandwidth was occupied during the failed connection rerouting from the point of view of the whole network:

$$SpareUsage = \sum_i B_i * l_i \qquad (5.3)$$

where $B_i$ – equivalent bandwidth of connection $i$, $l_i$ – length of the route that was used to restore connection $i$.

However these are inappropriate for tactical networks as they fail to take priorities into consideration. Instead different metrics are needed that consider the number of high-priority connections that have been restored and the number of low priority connections that were disrupted during the restoration process. A metric proposed in [35] is the restoration probability (RP). Another one proposed here is the weighted restoration probability (WRP). They are defined as follows:

$$RP_i = \frac{RN_i - DN_i}{FN_i} \qquad (5.3)$$

$$WRP = \frac{\sum_i w_i \cdot RP_i}{\sum_i w_i} \qquad (5.4)$$

$RP_i$ – restoration probability for priority i

$RN_i$ – number of restored connections for priority i

$DN_i$ – number of disrupted connections for priority i

$FN_i$ – number of failed connections for priority i

WRP – weighted restoration probability

i – priority

$w_i$ – weight coefficient for priority i

$RP_i$ is a measure of the net connections restored (taking into consideration that some lower priority connections may have been disrupted) as a ratio for the number of connections that failed. WRP is a weighted average of $RP_i$ that gives a higher weighting to the higher priorities. The following coefficients are assigned to different priorities (Table 5.1):

Table 5.1 Weight coefficients assigned to different priorities.

| **Priority** | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **Weight Coefficient** | 4 | 3 | 2 | 1 |

Thus the following statistics are collected during each simulation:

- Time taken to carry out the restoration.
- Number of connections and amount of bandwidth, for every priority level, that were disrupted by the failure.
- Number of connections restored, their priorities and bandwidth.
- Number of low-priority connections disrupted to facilitate restoration of higher priority connections, their priorities and bandwidth.

### 5.2.6 Implementation of Restoration Algorithms

To be able to properly assess the efficiency and performance of the proposed algorithms, it was necessary to evaluate them against other comparable algorithms. The Komine Algorithm and the VPPS Algorithm were chosen as the most suitable examples of dynamic and pre-planned algorithms, respectively, because they share some of the characteristics of the algorithms proposed in this research, and performance information was available for them [26, 28-30]. The availability of performance results served mainly to validate the simulation model.

Thus in total, four restoration algorithms were implemented, as follows:

- Dynamic Restoration Algorithm for Tactical Networks.
- Pre-planned Restoration Algorithm for Tactical Networks.
- Komine Algorithm for Civil Networks.
- VPPS Restoration Algorithm for Civil Networks (the Late Rerouting approach).

## 5.3 Simulation Environment

### 5.3.1 Network Topology

As stated in Chapter 2, the topology of tactical network is a sparsely-connected mesh with average connectivity of about 3 - 4. The link bandwidth is assumed to be 2 Mbps, although more often it is 0.5 or 1 Mbps.

Three standard network topologies have been chosen to model proposed algorithms. These are well known experimental networks, used by many researchers in their studies [21, 24, 25, 31, 35, 42, 43, etc.]:

- the Rudin Network,
- the Metropolitan LATA Network,
- the US Network.

The *Rudin Network* (Figure 5.7) described and used in [39, 40] corresponds to the requirements of tactical networks, and can be considered as a small-sized tactical network.

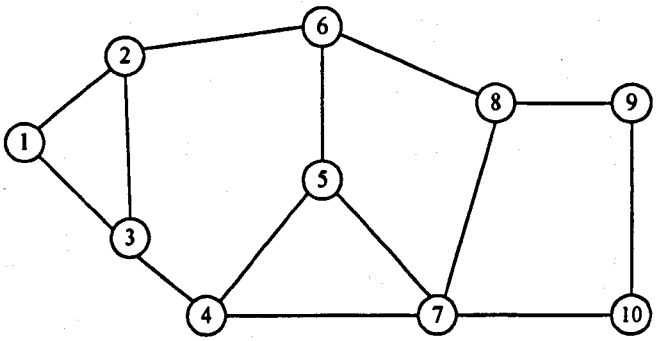



Figure 5.9 Rudin Network.

The *Metropolitan LATA Network* (Figure 5.8) [21, 25, 31, 35, 42, 43, 48, 56] represents the medium-sized tactical network.

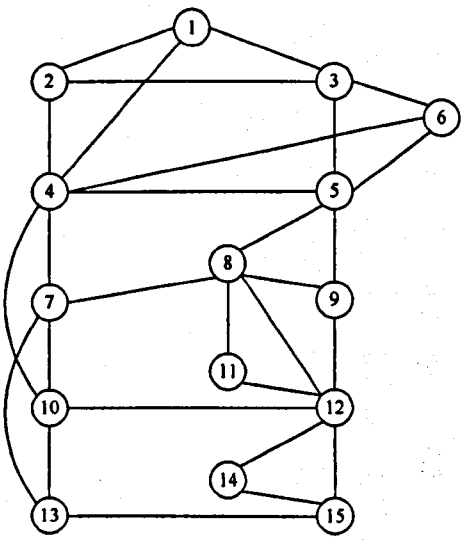



Figure 5.10 Metropolitan LATA Network.

The *US Network* (Fig. 5.9) used in [24, 25, 48, 50] is selected to study the performance of the algorithms in a large network. This is important for two reasons. It allows the

scalability of the algorithms to be evaluated. Also, failures at the edge of a network have different characteristics compared to failures in the middle. With larger networks failures are less likely to involve edge nodes.

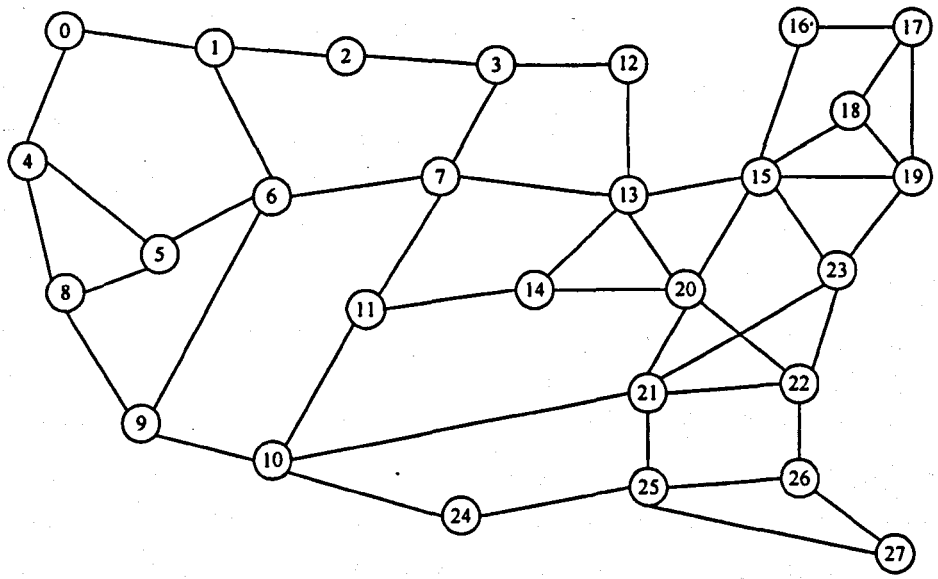



Figure 5.11 US Network.

The characteristics of these networks are summarised in the table below:

Table 5.2 Summary of network parameters.

| Network | Number of Nodes | Number of Links | Connectivity |
|---|---|---|---|
| Rudin Network | 10 | 14 | 2.8 |
| Metropolitan LATA Network | 15 | 28 | 3.7 |
| US Network | 28 | 47 | 3.4 |

### 5.3.2 Link and Node Parameters

According to descriptions of tactical networks provided by DERA, link bandwidth values selected are in the range from 512 Kbs to 1 Mbs.

To be able to compare the numeric results of simulations, link and node delays have been selected to be similar to those used by other authors in their studies [24, 25, 31, 35, 41 – 43, 56].

Link delay values used by different authors were 5 microseconds/km [24], 2 msec [43, 56], 2-12 msec [42], 25msec [31], and some authors do not identify this parameter in publications. A 2 msec delay is selected for this study.

Message processing (node delay) time is varied in different experiments, and was assigned values of 0.5 msec [24], 1-3 msec [41], 4 msec [42], 5 msec [43, 56], 10 msec [25, 31], 10-12 [35], etc. A 5 msec delay is selected for this study.

### 5.3.3 Traffic

According to [4], voice, data and video services must be supported. Furthermore, every traffic type has four priority levels. However, there is no specific information about how priorities are assigned to different services. Also, there is no difference for proposed restoration algorithms between different types of services. It is proposed to use the following traffic classes for simulations according to [4]:

Table 5.3 Traffic types characteristics.

| Traffic type | Description | Mean data rate (kbps) | Peak data rate (kbps) | Percentage of total traffic | $B_{eq}$ |
|---|---|---|---|---|---|
| A | CBR voice traffic | 16 | 16 | 60 % | 16 |
| B | Terminal traffic | 0.8 | 40 | 30 % | 8 |
| C | LAN applications | 10 | 100 | 10 % | 32 |

Total network bandwidth utilisation can be varied from 20 to 80 % (e.g., run algorithm for 20%, 40%, 60% and 80% load), while the number of connections can be limited to values used by other authors (e.g., 315 for LATA Network [35]).

## 5.4 Summary

It was argued in this chapter that software simulation is the most appropriate method to prove the consistency of the proposed algorithms and to assess their performance characteristics.

The proposed approach to modelling was presented and details of simulation software designed to model restoration algorithms given. Four restoration algorithms were selected for modelling and subsequent results comparison. The simulation conditions that were chosen for modelling are discussed and values selected explained. They are summarised below in Table 5.4:

Table 5.4 Simulation parameters.

| **Topology** | **Rudin Network** | **LATA Network** | **US Network** |
|---|---|---|---|
| Number of Nodes | 10 | 15 | 27 |
| Number of Links | 14 | 28 | 47 |
| Connectivity | 2.8 | 3.7 | 3.5 |
| Message Processing Time | 5 msec | | |
| Link Delays | 2 msec | | |
| Link Bandwidth | 512 Kbs – 1 Mbs | | |
| VP Bandwidth | 8 Kbs, 16 Kbs and 32 Kbs – assigned randomly. | | |
| Available Bandwidth | 20%, 40%, 60%, 80% | | |
| Number of VPs | Necessary to provide 20%, 40%, 60% and 80%NBU. | | |
| Failure Type | Single node failures. | | |

# Chapter 6. Simulation Results

Two new Restoration algorithms have been proposed as part of this research ; the specifications of these were given in Chapter 4. It was argued in Chapter 5 that simulation was the most appropriate way in which to model these algorithms and an architecture for a simulation package together with the relevant simulation parameters and parameter values was presented. Chapter 5 also mapped out the experiments, which would be carried out. This chapter is concerned with analysing the results of these experiments.

It was apparent from the results of the first set of experiments for the DRA-TN algorithm that the automatic restoration of low priority connections after a failure was not a sensible approach because these could subsequently be disconnected to free bandwidth to allow high priority connections to be restored. Thus a number of modifications were included in the proposed algorithm and these are explained in Section 6.1.

Section 6.2 compares the performance of the DRA-TN algorithm proposed as part of this research with a comparable dynamic algorithm originally proposed for civil networks. The algorithms are compared on the basis of the parameters outlined in Chapter 5. Similarly Section 6.3 compare the PPR-TN algorithm proposed as part of this research with a comparable pre-planned algorithm originally proposed for civil networks. A comparison of the DRA-TN and PPR-TN algorithms is presented in Section 6.4. Finally a summary of the results is presented in section 6.5.

## 6.1 Restoration Algorithms Modifications and Parameters

### 6.1.1 DRA-TN. Restoration Threshold Algorithm Modification

The DRA-TN algorithm specified in Chapter 4 worked on the principle that it would try to restore all connections affected by the failure, and that restoration of high priority connections would take precedence over lower priority connections. Thus it was expected that lower priority connections would be disrupted to allow higher priority

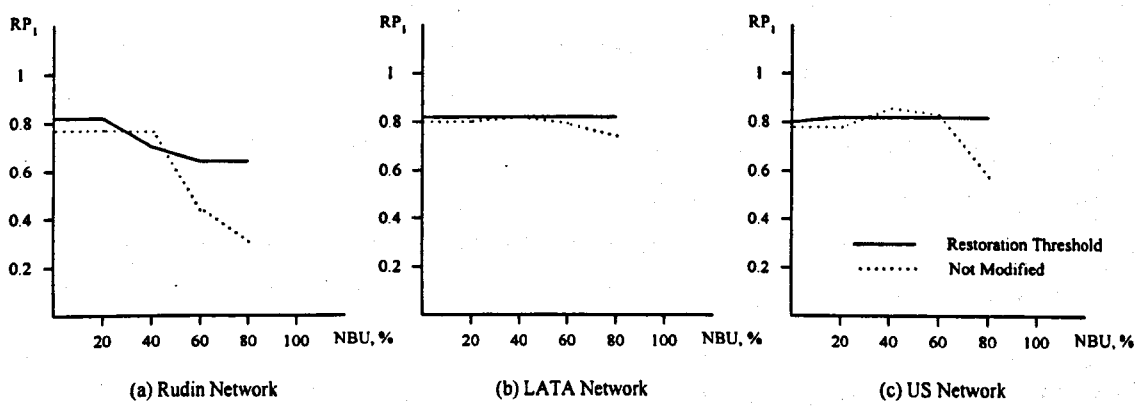
connections to be restored. However, when the first experimental results were produced, it was clear that a significant number of low-priority calls, restored at the beginning of the restoration process, were subsequently disrupted, specifically, to restore the higher priority connections; to restore a connection and then disrupt it soon after is not a sensible policy and wastes time and resources. Hence it was decided to modify the algorithm slightly such that the restoration of traffic of a given priority is initiated by a node only if the average traffic load on all of its outgoing links does not exceed some predefined threshold. This modification was intended to reduce the number of restoration messages generated and increase the restoration speed without significant impact on restoration ratio.

The expected improvements in performance were confirmed by the experimental results shown in Figs 6.1 - 6.3. In the graphs the dashed lines show the results of the unmodified DRA-TN, while the solid line corresponds to the modified algorithm for the following average traffic load thresholds set: $P_4 = 45$ %, $P_3 = 60$ %, $P_2 = 75$ %. $P_3 = 60$ % means for example, that if the average load on all outgoing links is more than 60 % a node not will not attempt to restore the third and the fourth priority traffic. Note that the node will automatically always try to restore the highest priority traffic.

Fig. 6.1 compares the performance of the Restoration Probability for priority 1 traffic, for the unmodified and modified DRA-TN algorithm, for all 3 networks, Rudin, LATA and US, in Figs. 6.1a,b,c respectively. In all 3 graphs, at low load there is little difference between the performance of the algorithms, because there is enough free bandwidth to restore all the high priority traffic. As the load increases, the difference between the modified and unmodified algorithms becomes apparent. The modified algorithm shows little variation in performance between low and high load, because as the load gets higher the average load threshold modification will prevent it from restoring lower priority connections. Thus all the free bandwidth will be used exclusively for higher priority connections. In contrast, the unmodified algorithm will try to restore all connections, so some of the free bandwidth will be used for lower priority connections, thereby reducing the Restoration Probability of the Priority 1 connections. The performance difference between the two algorithms is more marked in the Rudin network (Fig. 6.1a) because it is the smallest of the 3 networks and thus has

little resources to begin with. The complete results of the comparison of the 2 versions of the DRA-TN algorithm are contained in Appendices C.2 and C.3.

Note that the Weighted Restoration Probability (WRP) values for both versions were approximately the same (full results are contained in Appendices C.2 and C.3). This means that both algorithms restore approximately the same amount of traffic but there will be a difference between the distribution of the restored traffic across the 4 priorities, with the modified algorithm restoring a higher level of high priority traffic. These results confirm the arguments put forward in the previous paragraph.




(a) Rudin Network (b) LATA Network (c) US Network

Figure 6.1 Analysis of DRA-TN Modifications. $RP_1$ parameter.

The modified DRA-TN algorithm has other advantages compared to the unmodified version. The number of messages is smaller (Fig. 6.2) and restoration time is also smaller (Fig 6.3) for the modified algorithm. Furthermore, as the load increases so the difference between the results for the modified and unmodified algorithms increases. This indicates that the modified algorithm is more efficient, and the comparative efficiency increases with increasing load[13]. The results also suggest that the modified algorithm is more scalable, although further experiments on larger networks are necessary to confirm this.

[13] Both modifications provided similar performance on Rudin network due to its limited topology.

The difference in the results is explained by the fact that for high traffic loads the unmodified algorithm produces extra messages while attempting to restore low-priority connections. These attempts are either unsuccessful, or successful temporarily but then these restored connections are disrupted again to allow the bandwidth to be used to restore higher priority connections. Many restoration messages are generated in these attempts with no extra connections restored. Hence the unmodified algorithm for the same level of restoration produces more messages.

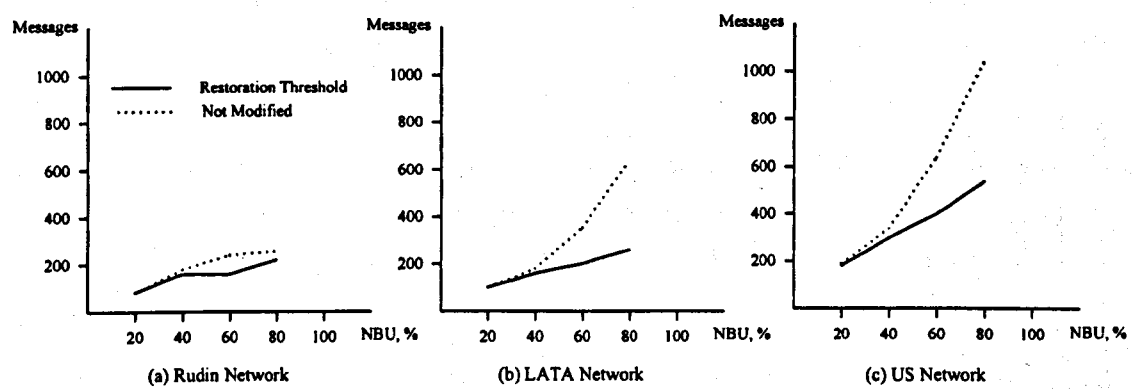



Figure 6.2 Analysis of DRA-TN Modifications. Number of Messages.

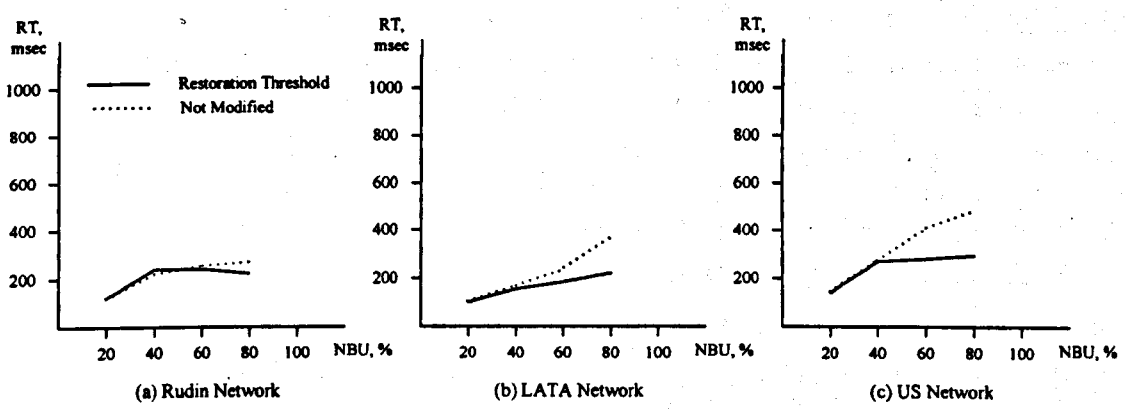



Figure 6.3 Analysis of DRA-TN Modifications. Restoration Time.

The conclusion is that the DRA-TN algorithm with the Restoration Threshold modification has much better performance and is much more efficient, particularly at high loads. Consequently, only this version of the algorithm was chosen for further study and the results are presented below.

### 6.1.2 PPR-TN. Restoration Threshold Algorithm Modification

The Restoration Threshold modification was also included in the PPR-TN algorithm and the same set of experiments were carried out to compare the modified and unmodified versions of the algorithm. The results comparing both versions are qualitatively the same as for the DRA-TN algorithm and the rationale for the results is similar, and thus are not presented here (see Appendices C.4 and C.5 for full details). Hence, the same conclusion is drawn for PPR-TN. The Restoration Threshold modification restores a higher level of high priority calls in a shorter time, while generating fewer messages. Thus only the results of the modified algorithm are presented below.

### 6.1.3 Dynamic Restoration Algorithms. Hop Limit Values

It is standard practice in routing problems to specify the maximum number of hops that a route can have otherwise the route can become very inefficient and use an excessive amount of resources. Experiments were carried out in this study to find the optimum Hop Limit Counter (HLC) values for the dynamic restoration algorithms (DRA-TN and Komine algorithms). The optimal values for DRA-TN are 2 for Rudin and LATA networks and 3 for US network (refer to Appendices C.3.1 – C.3.5 for results of modelling several HLC values). The Komine algorithm provides the best performance with a HLC of 3 for Rudin network and 4 for LATA and US networks (refer to Appendices C.6.1 – C.6.6 the results of modelling several HLC values). The HLC values given in this paragraph for the respective algorithms and networks are those used in the results presented below.

## 6.2 DRA-TN Algorithm against Komine Algorithm

It was necessary to compare the performance of the DRA-TN against a comparable algorithm to be able to quantify the strengths and weaknesses of the DRA-TN. The Komine algorithm, which was developed for civil networks was chosen. It shares a number of characteristics with the DRA-TN, for instance, it uses dynamic restoration and can cope with node failures. It was modified to cater for priority traffic, specifically for this study. The complete simulation results for the DRA-TN and the Komine, are presented in Appendices C.3 and C.6 respectively, and an analysis of these is given in the following section.

### 6.2.1 High Priority Traffic Restoration

It is argued in this thesis that for a tactical network restoration of connections should take place in order of precedence, with high priority traffic clearly having the highest precedence. Thus in evaluating the performance DRA-TN it is sensible to consider the results for the two highest priorities, Priority 1 and Priority 2.

The Restoration probability results for Priority 1 are shown in Fig. 6.4, for each of the 3 experimental networks. For the LATA and US networks, at low load, there is little difference between the algorithms, however at loads of 40% and above the DRA-TN algorithm is clearly superior. Specifically, DRA-TN gives a higher level of restoration, and this is constant across all loads, which implies that the algorithm is stable even under extreme conditions. In contrast the level of restoration begins to decrease rapidly for the Komine algorithm at loads of about 40%.

The results for the Rudin network are different. Although the DRA-TN restores a higher level of traffic compared to the Komine algorithm, for all loads, both decrease as the load increases. This is because the Rudin network is small and thus has limited bandwidth. So as the load increases there is less free bandwidth that can be used for restoration.

These results can be explained by the differences in the algorithms' specifications. DRA-TN achieves its main goal – the restoration of high-priority traffic. The distinction in $RP_1$ values is more evident for high NBU values. At this point Komine algorithm

cannot restore a large proportion of the high-priority connections because there is no available bandwidth on the alternate routes, while DRA-TN disrupts lower priority connections to make the necessary bandwidth available. These results quantify the benefits of disrupting lower priority connections.

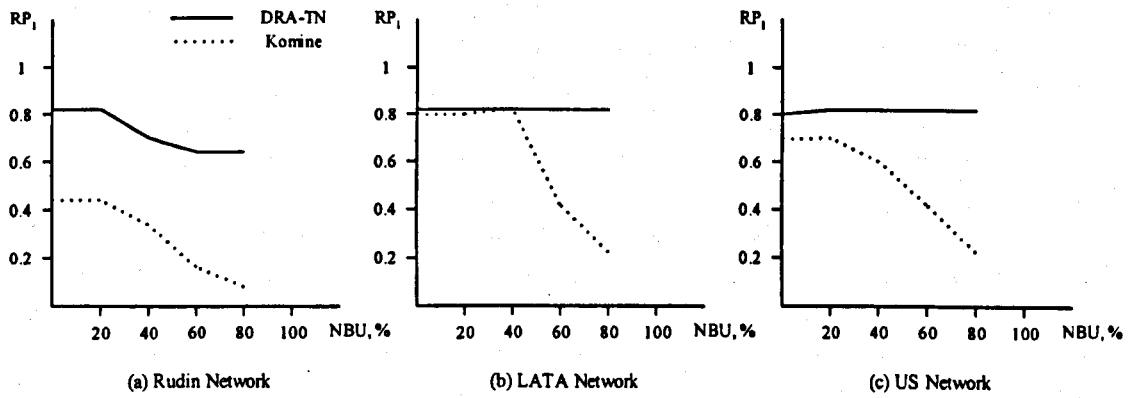



Figure 6.4 DRA-TN against Komine. $RP_1$ parameter.

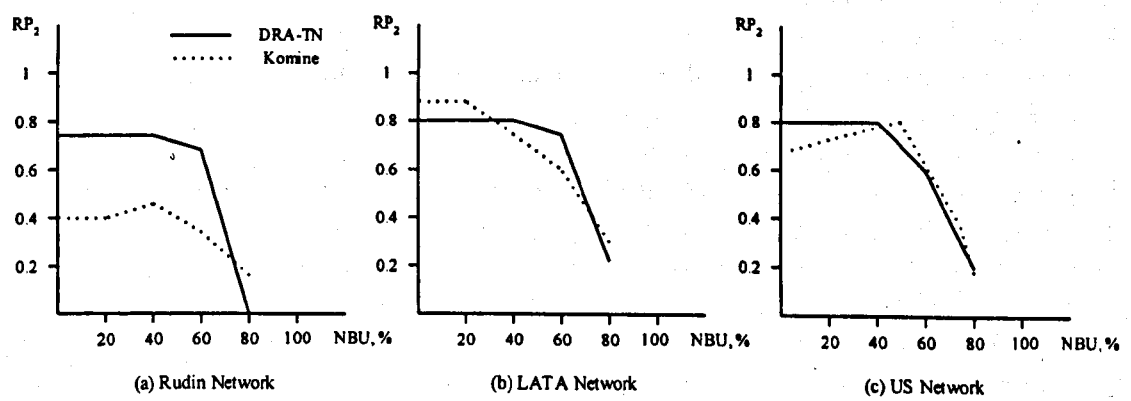



Figure 6.5 DRA-TN against Komine. $RP_2$ parameter.

The restoration levels for Priority 2 traffic are shown in Figs. 6.5 and while the performance of the DRA-TN algorithm is marginally better, there is little significant difference between them. The decrease in the restoration probability at high loads occurs because having restored all the high priority traffic already, there is no available

bandwidth left to use. Clearly then there is no bandwidth left for the lower priority connections.

The Weighted Restoration Probability (WRP) is similar for both algorithms (Fig. 6.6). This is to be expected because DRA-TN disconnects low priority calls, while the Komine algorithm restored some.

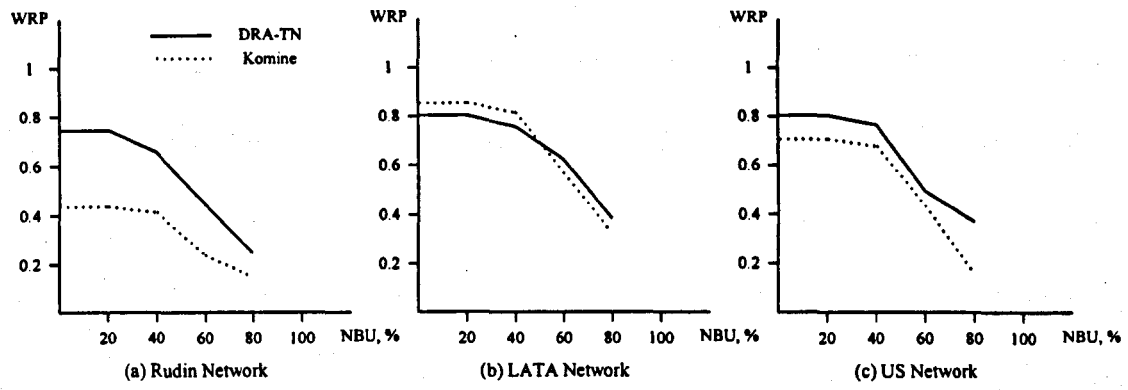



Figure 6.6 DRA-TN against Komine. WRP parameter.

Again the results are different for the Rudin algorithm because it is a small network. Note however that the DRA-TN algorithm maintained a high restoration probability in this instance because the double search technique used in DRA-TN provided better route search for this particular topology.

### 6.2.2 Number of Messages

The number of messages required to accomplish the restoration is much less for DRA-TN algorithm in the LATA and US Networks (Fig. 6.7), for the same restoration threshold. The number of messages for the DRA-TN algorithm is larger in the Rudin network but it must be remembered that the Komine algorithm restored approximately half the connections in this instance compared to the DRA-TN algorithm.

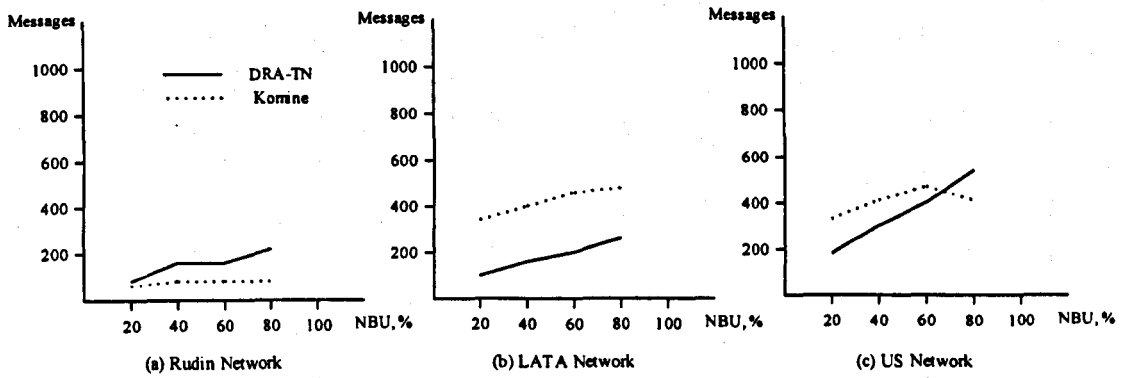



Figure 6.7 DRA-TN against Komine. Number of Messages.

The exact values of number of search messages generated by each algorithm are shown in Table 6.1. There are a number of reasons for the significant difference between the results. The Hop Limit Counter values are less for DRA-TN because the double-search technique reduces the flooding area considerably. Also, the use of collisions by the DRA-TN algorithm means that the alternative route converges more quickly. Note that the number of messages generated by the DRA-TN algorithm, for high NBU values are disconnect messages, in contrast to the Komine algorithm which does not disconnect calls.

Table 6.1 HLC values and Number of Search Messages for DRA-TN and Komine algorithms.

| Restoration Algorithm | Rudin | | LATA | | US network | |
|---|---|---|---|---|---|---|
| | HLC | Search Messages | HLC | Search Messages | HLC | Search Messages |
| **Komine algorithm** | 3 | 32 | 4 | 286 | 4 | 253 |
| **DRA-TN** | 2 | 29 | 2 | 57 | 3 | 102 |

### 6.2.3 Restoration Time

The Restoration Time is shown in Figure 6.8. It is clear from the graphs that the Komine algorithm is faster (Fig. 6.8). However, taking into account that DRA-TN completes the restoration within 300 msec and that this is well within the accepted maximum

restoration time of 2 sec for most applications [15], then this performance difference does not appear to be critical.

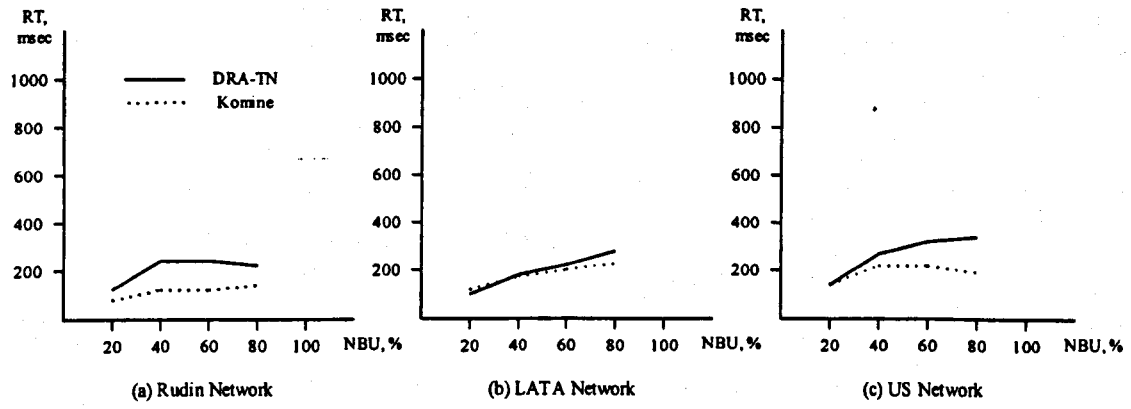



Figure 6.8 DRA-TN against Komine. Restoration Time.

### 6.2.4 Summary and Conclusions

The additional mechanisms included in the DRA-TN algorithm provide several advantages compared to Komine algorithm. In particular, the double-search technique and the Restoration Threshold modification reduce the total number of messages and the technique of restoring the connections in priority order maximise the restoration probability for the high priority traffic. These advantages are considered essential for tactical networks and thus the DRA-TN algorithm is considered the more suitable algorithm.

## 6.3 PPR-TN against VPPS

The performance of the PPR-TN algorithm also needed to be assessed against some existing pre-planned restoration algorithm proposed for civil networks and the VPPS algorithm (Late Rerouting modification, see section 3.5.1.3 for details) was considered the most suitable. This section provides an analysis of the simulation results and a complete set of results for the PPR-TN and VPPS algorithms are contained in Appendices C.5 and C.7, respectively.

### 6.3.1 High Priority Traffic Restoration

The restoration probability for Priority 1 traffic is shown in Fig. 6.9. It is clear from the graphs that the PPR-TN algorithm restores all the Priority 1 traffic, on all networks, at all traffic loads. In contrast, for the VPPS algorithm the Restoration probability decreases as the load increases. The results for Priority 2 traffic (Fig. 6.10) show that the level of restoration is approximately the same level for both algorithms, although for the Rudin network, the performance of the PPR-TN is better.

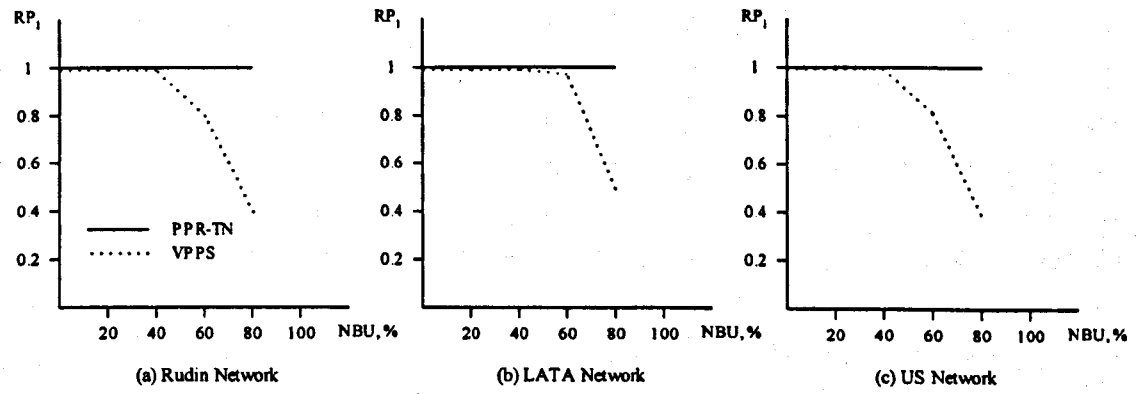



Figure 6.9 PPR-TN against VPPS. $RP_1$ parameter.

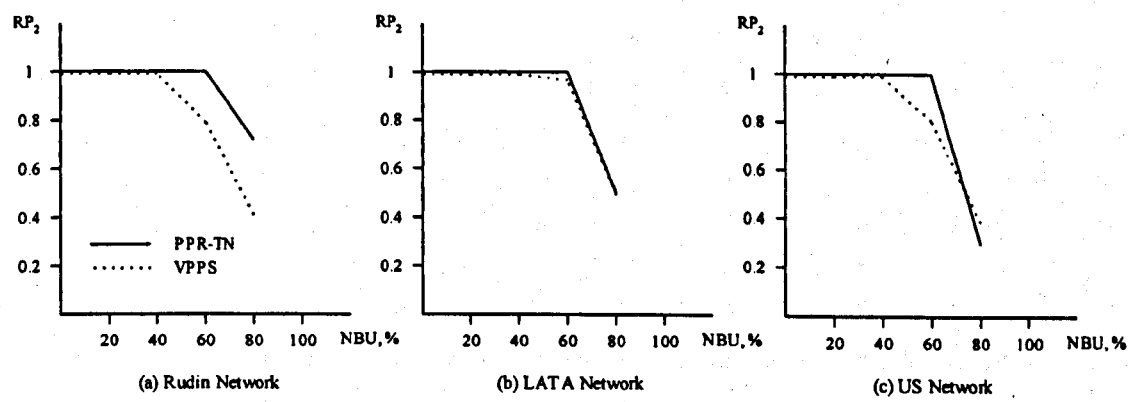



Figure 6.10 PPR-TN against VPPS. $RP_2$ parameter.

The WRP is similar for both algorithms (Fig. 6.11) because with a heavy traffic load the PPR-TN disconnects many low priority calls while the VPPS manages to restore some.

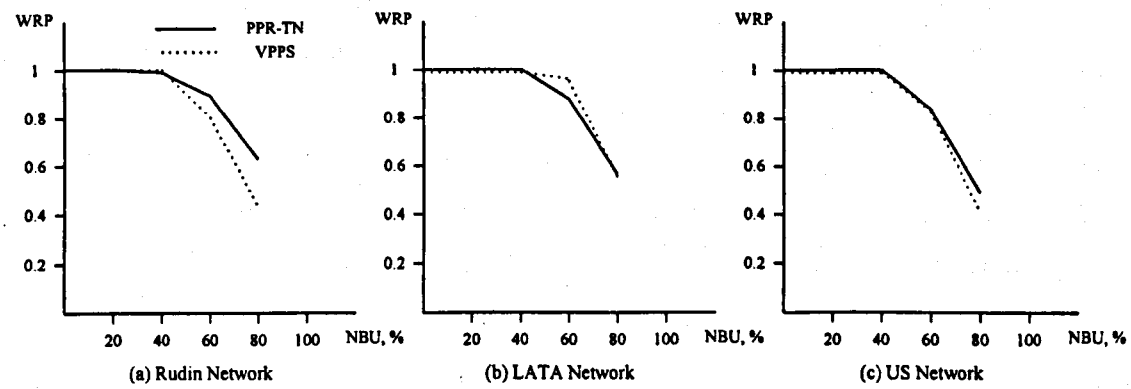



Figure 6.11 PPR-TN against VPPS. WRP parameter.

These results can be explained by the specification of the PPR-TN algorithm. It is designed to disconnect lower priority calls, if necessary, in order to obtain sufficient bandwidth to restore high priority calls. Note that connections cannot be restored if the failure occurs at a VC/VP terminator and this is the reason why the $RP_1$ does not have a value of 1.

### 6.3.2 Number of Messages

The graphs shown in Fig. 6.12 show the number of messages generated by the PPR-TN algorithm is much less compared to the VPPS algorithm, for the LATA and US networks, particularly for high traffic loads. This is because the Restoration Threshold ensures that the restoration of low priority connections is attempted only if there is sufficient bandwidth available. Hence, the number of messages will be lower and the restoration of high priority traffic higher for the PPR-TN algorithm

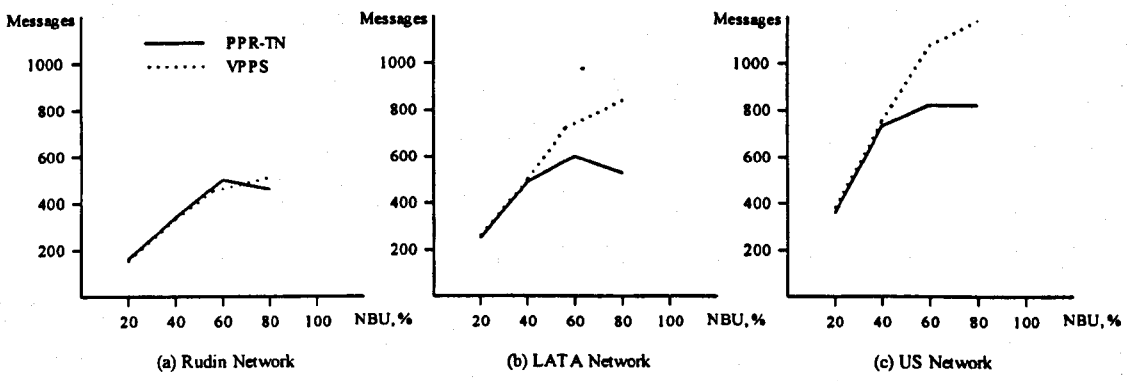



Figure 6.12 PPR-TN against VPPS. Number of Messages.

### 6.3.3 Restoration Time

The Restoration Time for the two algorithms is compared in Figs. 6.13. The performance for the PPR-TN is better, particularly at high loads.

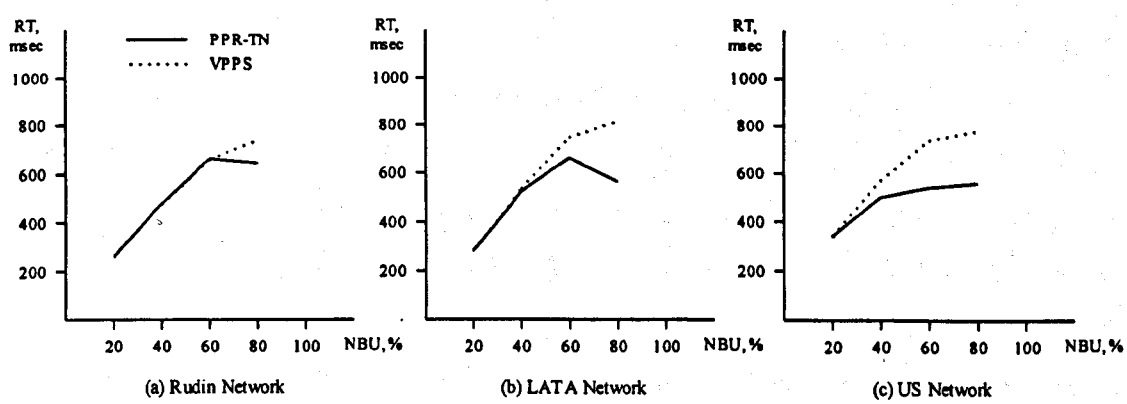



Figure 6.13 PPR-TN against VPPS. Restoration time.

By comparing Fig. 6.2 with Fig. 6.3 and Fig. 6.12 with Fig. 6.13 then it is clear that there is a strong correlation between the number of messages and restoration time; the more messages that are generated, the longer the Restoration Time.

### 6.3.4 Summary

The overall conclusion is that although the VPPS algorithm manages to restore some low-priority traffic under a heavy traffic load and provides almost the same WRP, the high level of high priority traffic restoration combined with the generation of fewer messages and faster restoration time makes the PPR-TN algorithm more suitable for tactical networks.

## 6.4 Comparison of the DRA-TN algorithm with the PPR-TN

The two new algorithms proposed for tactical networks, DRA-TN and PPR-TN, are compared in this section. They implement different approaches to restoration, and accordingly both of them have their own strengths and weaknesses.

The advantages of DRA-TN are faster restoration and smaller number of messages generated. PPR-TN, in turn, provides better restoration both for high-priority and low-priority traffic.

### 6.4.1 Restoration Time and Number of Messages

The Restoration Time for both algorithms is shown in Figs. 6.14 and it is clear that the DRA-TN algorithm is at least twice as fast as the PPR-TN algorithm.

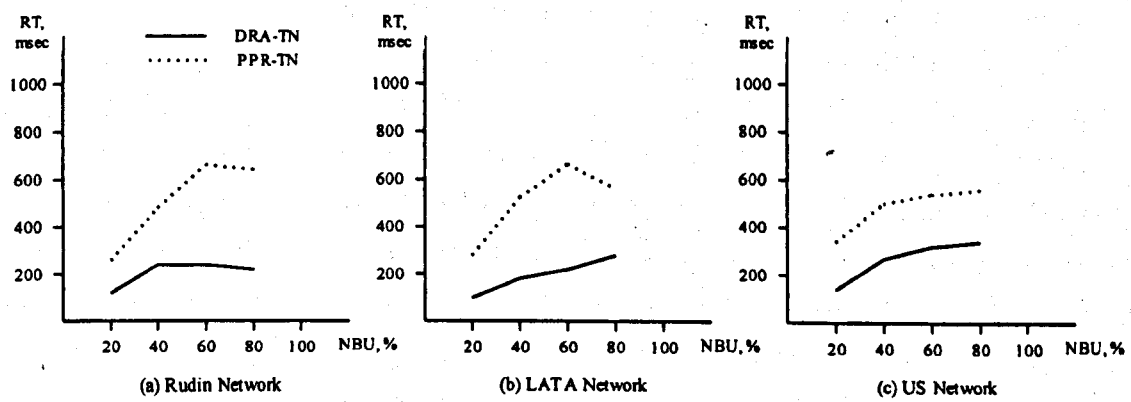



Figure 6.14 DRA-TN against PPR-TN. Restoration time.

Also, the number of messages generated by the DRA-TN algorithm is significantly less; this is shown in Figure 6.15.

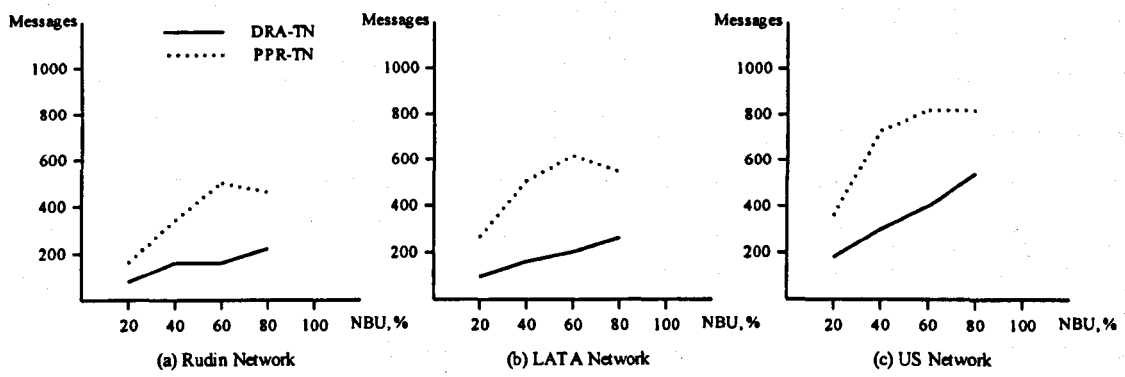



Figure 6.15 DRA-TN against PPR-TN. Number of Messages.

Both of these results can be explained by the fact that DRA-TN uses local rerouting, while PPR-TN implements end-to-end path restoration. Local rerouting is faster and requires fewer messages to be transferred because the restoration only involves diverting the failed connection around the failed node. In contrast, the path restoration requires the full path reconfiguration between the connection end points.

### 6.4.2 Advantages of PPR-TN

The PPR-TN always achieves a greater level of restoration; this is shown in Figures 6.13 – 6.15). In particular, the Restoration probability is about 20% higher for both for first and second priorities, compared to the performance of the DRA-TN algorithm. Similarly, the WRP performance of the PPR-TN is improved when compared to the DRA-TN algorithm.

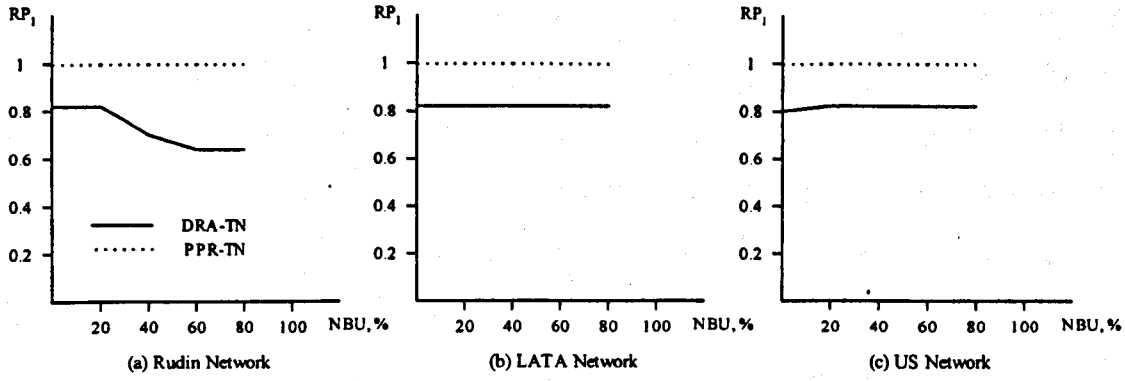



Figure 6.16 DRA-TN against PPR-TN. $RP_1$ parameter.

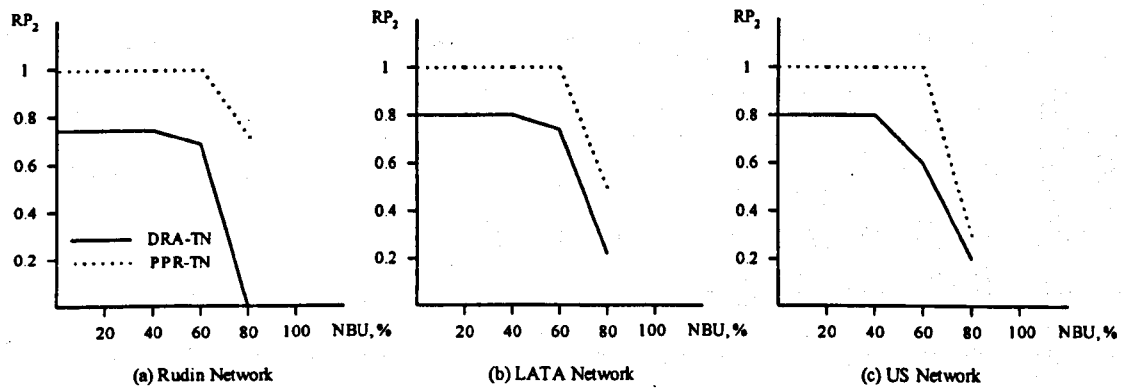



Figure 6.17 DRA-TN against PPR-TN. $RP_2$ parameter.

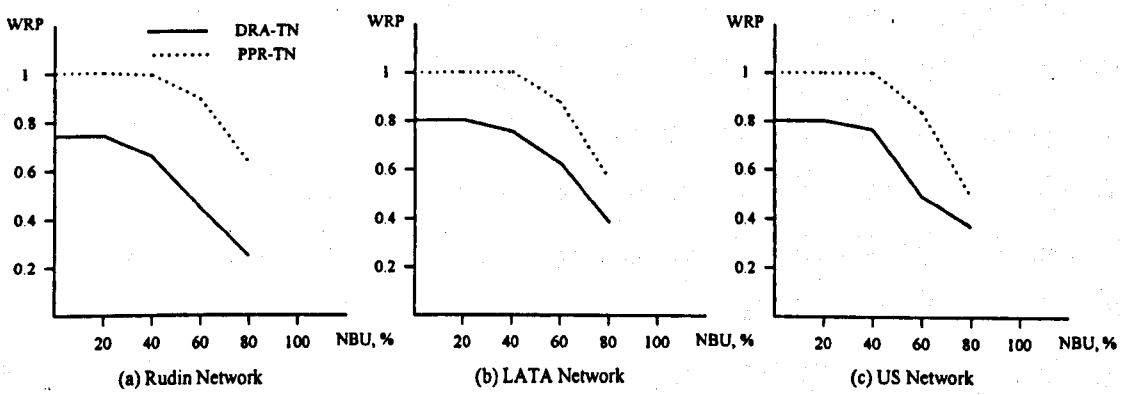



Figure 6.18 DRA-TN against PPR-TN. WRP parameter.

The superior performance of the PPR-TN algorithm can be explained by the specification of the algorithm. Path restoration exploited in PPR-TN uses network resources more effectively by distributing failed traffic more evenly over the network. In contrast, the dynamic restoration algorithm utilises local rerouting and so there is contention for the available bandwidth in the proximity of the failed network element. Hence, the DRA-TN algorithm sometime fails to find enough routes, or enough bandwidth on the routes.

### 6.4.3 Summary

Two new algorithms proposed for tactical networks have their own advantages, which logically follow from their specifications. PPR-TN algorithm restores more traffic. However, DRA-TN not only provides a high restoration ratio, it is also faster and generates a smaller number of messages. Another advantage of DRA-TN is the use of dynamic restoration approach that corresponds better to the dynamic nature of tactical environment (see section 2.2 for details). It may not be always possible to provide node disjoint backup routes which is a mandatory requirement for PPR-TN. Rearranging backup routes after a failure (to prepare the network for the subsequent failure's restoration) can be a very complex task as well.

Therefore, given that the DRA-TN algorithm provides dynamic restoration and requires fewer resources to execute (fewer messages generated and thus fewer to process and transport) this is considered the most suitable for a tactical network.

## 6.5 Summary and Conclusions

The performance of the DRA-TN and PPR-TN algorithms proposed as part of this research were compared to similar types of algorithms designed for civil networks. A simulation package was built which modelled the algorithms operating on three network topologies: the Rudin, LATA and US networks. Different values of NBU values were used to test algorithms' behaviour under various traffic load conditions. Two algorithms

proposed for civil networks, Komine algorithms and VPPS, were tested under the same conditions in order to provide a performance bench mark.

The results indicate that the DRA-TN and PPR-TN algorithms achieve a higher level of restoration for high-priority traffic compared to the Komine and VPPS algorithms respectively. The difference becomes very significant at high traffic loads. Furthermore, the number of messages required to achieve restoration is less for the proposed algorithms.

DRA-TN and PPR-TN algorithms were also compared. The results indicate that DRA-TN is faster and generates less messages, while PPR-TN provides a higher level of restoration.

# Chapter 7. Conclusions and Further Work

## 7.1 Summary of the Research

ATM is an attractive technology for future military communication systems because the next generation of tactical networks are required to offer high network throughput and support multi-service applications. However, ATM was developed for operation in network infrastructures that feature the very low failure rates, which are characteristics of current and future civil communication networks. In contrast, tactical networks are much less reliable. Therefore, if services are to be supported in tactical ATM networks, efficient automatic failure restoration techniques are critical.

Tactical networks have specific characteristics that put very different constraints on restoration algorithms compared to civil networks. In particular, they must support a four-level priority system and they are unable to reserve bandwidth specifically for restoration because of the limited resources and node failures are the most typical failure scenario.

An extensive literature analysis revealed that a large number of restoration algorithms have been proposed for civil networks. They were carefully studied and critically assessed and the conclusion was drawn that none of these fully met the needs of tactical networks, however, some of the techniques used by various algorithms had some potential. Hence, two new restoration algorithms were proposed specifically for tactical ATM networks, the Dynamic Restoration Algorithm for Tactical Networks (DRA-TN) and the Pre-Planned Restoration Algorithm for Tactical Networks (PPR-TN). As the names suggest, these algorithms used a dynamic approach and a pre-planned approach, respectively.

The main thrust of the proposed algorithms is that when there is insufficient bandwidth in the network to restore all the failed connections, bandwidth is made available to restore high-priority connections by disconnecting the low-priority ones. A number of additional mechanisms were also included to reduce the use of resources that are very limited in tactical networks.

To verify the proposed algorithms and assess their performance characteristics a software simulation package was developed. It allowed the algorithms to be tested on different network topologies with various traffic scenarios and failure modes. The package measured the following metrics: restoration time, number of messages, restoration levels for each traffic priority and all the traffic in total.

The proposed algorithms were tested on three experimental network topologies: the Rudin network (an analogue of a small-scale tactical network), the LATA network (medium-sized), and the US network (large-scale network). Different NBU values (20 % - 80%) were used to test algorithms' behaviour under various traffic load conditions.

It was demonstrated that the DRA-TN and PPR-TN provide better restoration of high-priority traffic than algorithms proposed for civil networks (Komine and VPPS algorithms). The difference becomes significant at high traffic loads. The number of messages required to achieve restoration is less for the proposed algorithms as well.

DRA-TN and PPR-TN algorithms were also compared. The results indicate that DRA-TN is faster and generates fewer messages, while PPR-TN provides higher restoration ratio. Although these results were conclusive, nevertheless there are a number of aspects of the work that could be investigated further. These are outlined in the next section.

The primary research objectives defined in Section 1.2 were successfully achieved and can be summarised as follows:

- Tactical networks parameters and their distinctions from civil networks were identified as well as requirements to restoration algorithms in tactical environment.
- Known approaches to civil networks restoration were studied. Methods, algorithms and techniques proposed for civil networks were analysed and their applicability for tactical environments was assessed.
- Two new restoration algorithms for tactical networks were proposed. They take into account the specific characteristics of the subject area and provide better performance than known algorithms.

- Simulation software was designed and implemented. Since standard performance metrics cannot fully represent the specifics of prioritised restoration new restoration metrics were proposed and used.
- Restoration algorithms for tactical networks were modelled for different network topologies and under various traffic conditions. Their performance was assessed against comparable algorithms proposed for civil networks.

There are also a number of limitations of the research that could have been avoided:

- Efficiency of the proposed algorithms could have been improved.
- The possibility of an integrated restoration technique implementation was not studied and modelled.
- Restoration from multiple node failures was not modelled though the proposed algorithms can be easily extended to provide this.
- Contact with DERA at the latest stages of the project should have been more intimate.

Some of these problems are addressed in section 7.3 in more detail.

## 7.2 Recommendations

Additional tests are recommended to model the algorithms for specific traffic patterns (various proportions of the 4 traffic priorities) and selected network topologies. The objective is to obtain optimal values for the Restoration Thresholds for each priority and the Hop Limit Counter value for the DRA-TN algorithm.

All Restoration algorithms require resources during execution. The recommendation is to ensure the nodes have enough resources to execute the algorithms.

## 7.3 Further Work

The results of the research presented in this thesis will be forwarded to DERA, and they will be discussed along with the further work. Some of the issues for further research are identified in this section.

### 7.3.1 Algorithms' Efficiency

There are several modifications that can be made to improve the efficiency of the DRA-TN algorithm. One proposal is to allow a single message to carry information about several connections. This will reduce the number of messages generated and hence increase restoration speed. In a tactical network where the bandwidth is limited reducing the number of messages generated would be an advantage. However, unpacking the messages will mean slightly more processing but this should be negligible. The technique of combining information relating to multiple connections is established practice in various network protocols.

The PPR-TN algorithm could benefit from having several backup routes assigned for high-priority traffic, thus a second back-up route is available in the event that the first is not.

### 7.3.2 Integrated Restoration Technique

An integrated restoration technique, effectively an amalgamation of two or more algorithms, has been proposed for civilian networks [44, 47, 50, etc.] and it has been suggested that it can be more effective than any single restoration algorithm. This is questionable. First, the integrated algorithm would require more network resources and possibly increase the restoration time. Nevertheless, this approach needs to be evaluated for tactical networks.

### 7.3.3 Multiple Failures

Multiple failure scenarios (where two or more non adjacent nodes fail) need to be studied also. The proposed algorithms can easily be adopted to provide multiple failure restoration. This would require minor modifications to the algorithm to store information relevant to different failures separately and analyse "failed node id" when processing restoration messages. Similar modifications can be made to the proposed algorithms to allow them to operate with multiple adjacent node failures.

Regardless of how straightforward these modifications are additional extensive simulation studies and analysis are necessary to quantify the relative efficiency of the algorithms.

### 7.3.4 Traffic

Another important issue relates to what the nodes should do with traffic arriving for failed connections during the restoration process. Taking into account the low transmission speed, it is evident that all the arriving traffic can be easily stored in the buffer. For example, if restoration takes 250 msec (DRA-TN), then for a 1 Mb link it would require only 32 K of the buffer space to store the traffic during the restoration. This seems more than realistic.

However, there still a number of topics to be addressed:

- How different QoS classes should be processed. For instance, should delay sensitive, real-time traffic simply be discarded because the late arrival of this traffic at the destination is unacceptable?
- How can the increase of transmission speeds influence buffer requirements?

### 7.3.5 Simulation Package Refinement

The simulation package that has been built as part of this research can easily be extended to allow other restoration algorithms, and other networks topologies to be modelled. It would be useful to further develop the package by providing a graphic user

interface that allows the user to easily configure the simulation input parameters, monitor the simulation process and analyse results automatically.

## 7.4 Conclusion

This chapter concludes the thesis. The work that has been undertaken on the project was summarised here and general recommendations about implementation of the proposed restoration algorithms in tactical networks were presented. The further research areas were identified as well.

# Appendix A Dynamic Restoration Algorithm for Tactical Networks. Algorithms and Message Formats.

## A.1 Overview

Appendix A provides additional information about DRA-TN algorithm that was not covered in its description given in Chapter 4. These are detailed message formats (section A.2) and block-scheme of the algorithm, which is executed at every network node during the algorithm functioning (section A.3).

## A.2 DRA-TN. Message Formats

Table A.1 DRA-TN. Search message format.

| Field Name | Size | Description |
|---|---|---|
| Message Type | 2 bits | Message type ID |
| Sender Node ID | 2 bytes | Identify Sender and failure location. |
| Failed Node ID | 2 bytes | |
| Number of Choosers (other Senders) | 4 bits | Chooser Identifiers (M- number of choosers) All information is taken from local switch table (see above). |
| Chooser Node ID | 2 bytes * M | |
| Hop Limit Counter | 4 bits | Limits search area. |
| Transit Node | 2 bytes * N | Transit nodes' identification, and information about the route passed by the message (N – route length) |
| Total bandwidth on the route | 1 byte | |
| $B_4, B_3, B_2, B_1$ | 4 bytes | |
| **End of message** | | |

Table A.2 DRA-TN. Route-found message format.

| Field Name | Size | Description |
|---|---|---|
| Message Type | 2 bits | Message type ID |
| Sender Node ID | 2 bytes | Identify Sender and Failure location. |
| Failed Node ID | 2 bytes | |
| Chooser Node ID | 2 bytes | Chooser for this restoration |

| | | |
|---|---|---|
| Hop Limit Counter | 4 bits | Route length from transit node to chooser. |
| Transit Node ID | 2 bytes * N | Route identification and description (amount of bandwidth that can be restored using this route). N – route length. |
| Total bandwidth | 1 byte | |
| $B_4$, $B_3$, $B_2$, $B_1$ | 4 bytes | |
| **End of message** | | |

Table A.3 DRA-TN. Acknowledge message format.

| Field Name | Size | Description |
|---|---|---|
| Message Type | 2 bits | Message type ID |
| Sender Node ID | 2 bytes | Identify Sender and Failure location. |
| Failed Node ID | 2 bytes | |
| Chooser Node ID | 2 bytes | Identifies Chooser Node. |
| Number of Connections | 4 bits | Identify connections being restored by this message. K – number of connections. |
| Next VPI/VCI | 3 bytes * K | Next VPI/VCI value to set in RT in the next node on alternate route |
| Sender VPI/VCI | 3 bytes * K | Sender VPI/VCI value to set in RT at the Sender Node (identifies VC that is being restored at its destination). |
| Connection Priority | 2 bits * K | |
| Connection Bandwidth | 1 byte * K | |
| Hop Limit Counter | 4 bits | Route length. |
| Transit Node ID | 2 bytes * N | New route identification. N – route length. |
| **End of message** | | |

Table A.4 DRA-TN. Cancel message format.

| Field Name | Size | Description |
|---|---|---|
| Message Type | 2 bits | Message type ID |
| Sender Node ID | 2 bytes | Identify Sender and Failure location. |
| Failed Node ID | 2 bytes | |
| Chooser Node ID | 2 bytes | Identifies Chooser Node. |
| Number of Connections | 4 bits | Identify connections that cannot be restored. K – number of connections. |
| Previous VPI/VCI | 3 bytes * K | Next VPI/VCI value to set in RT in the previous node on the route |
| Hop Limit Counter | 4 bits | Route length. |
| Transit Node ID | 2 bytes * N | New route identification. N – route length. |
| **End of message** | | |

## A.3 DRA-TN. Algorithm Block-Scheme

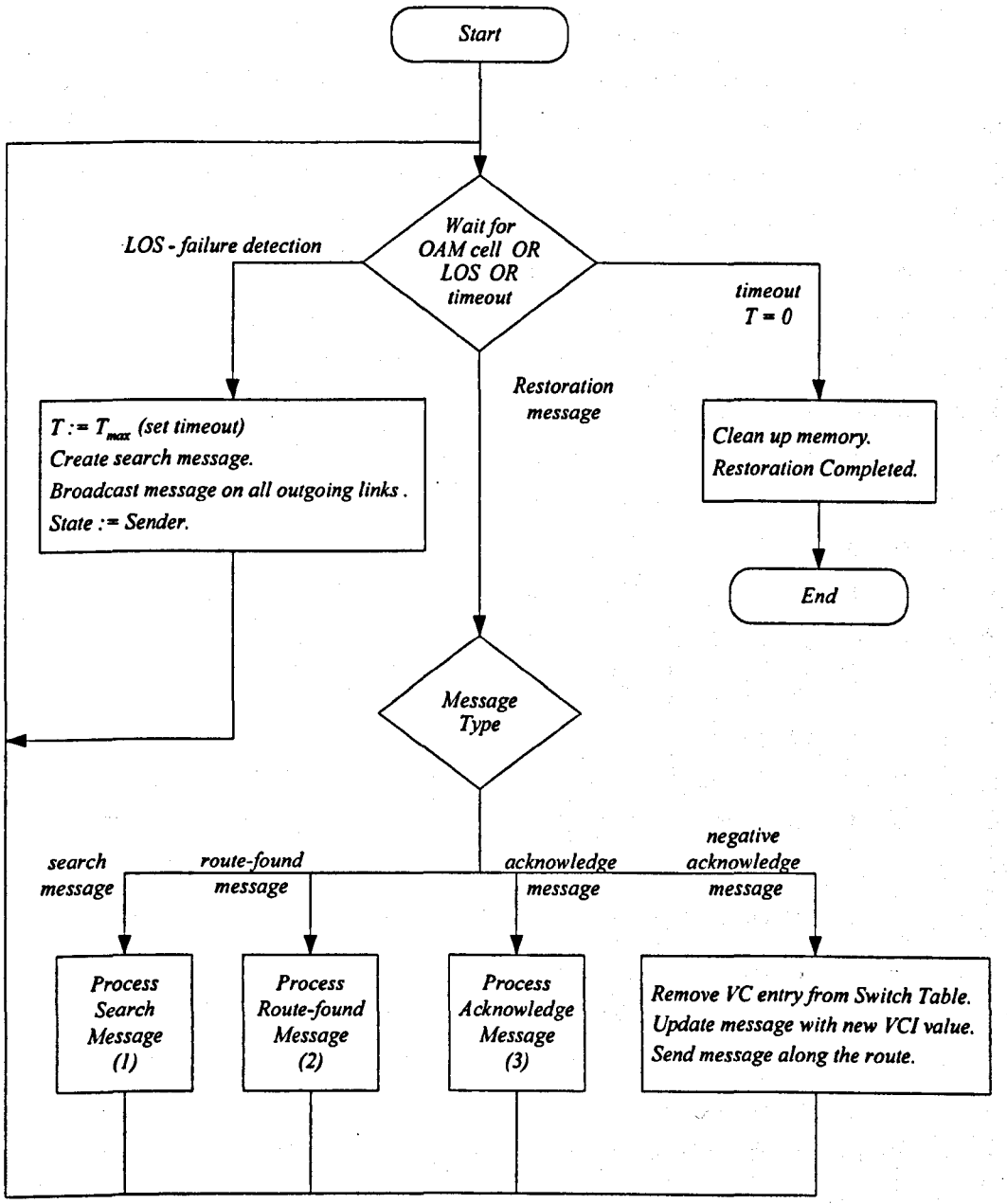



Figure A.1 DRA-TN: Block-Scheme of the Algorithm.

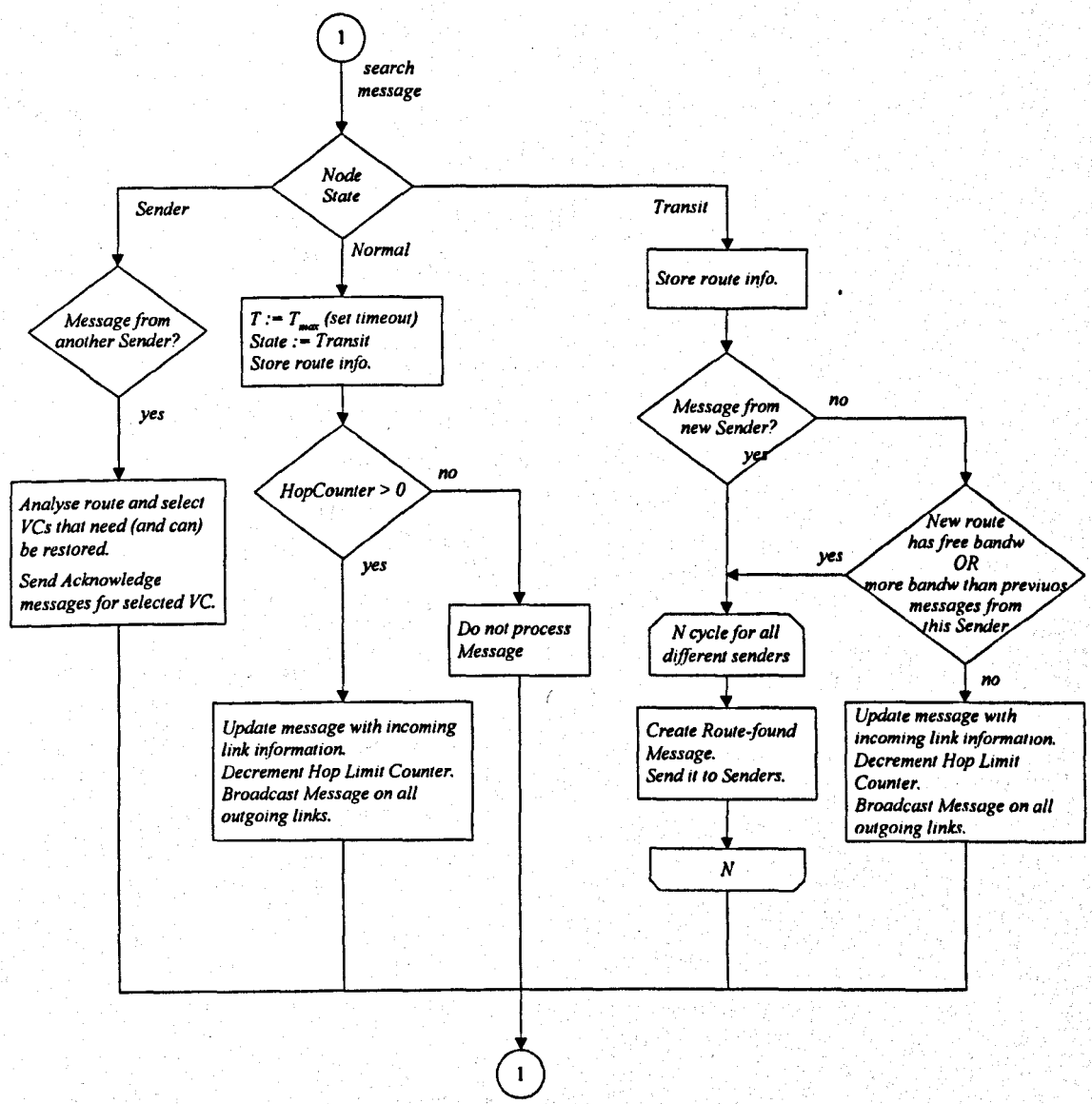
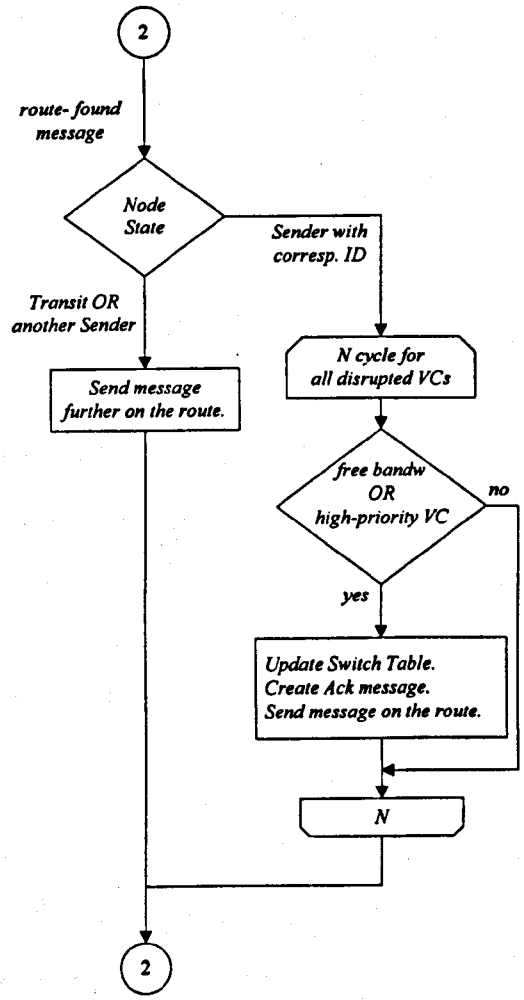
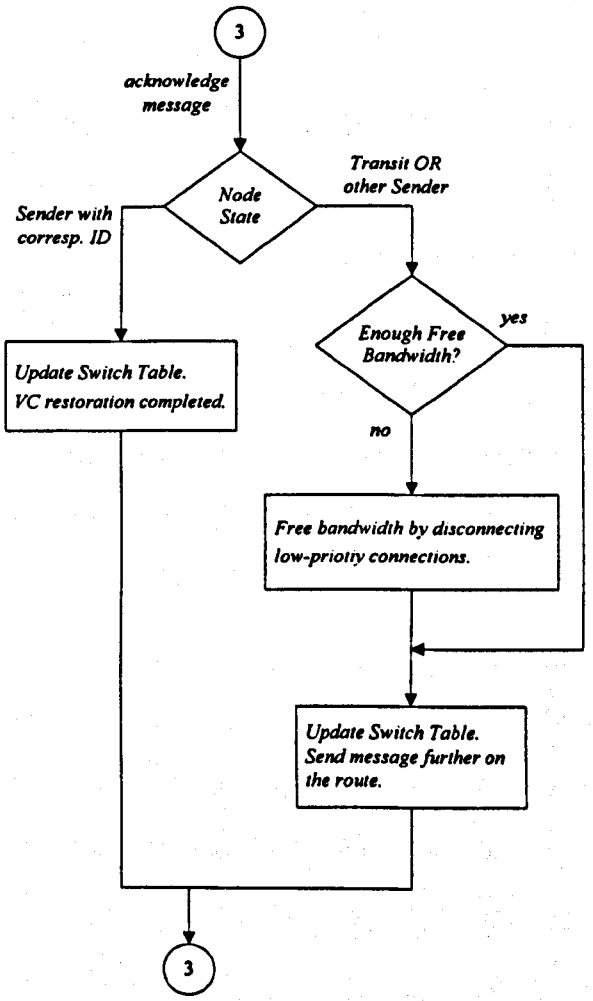



Figure A.2 DRA-TN: Search Message Processing Algorithm.




(a) Route-found Message Processing Algorithm




(b) Acknowledge Message Processing Algorithm

Figure A.3 DRA-TN: Route-found and Acknowledge Message Processing Algorithms.

# Appendix B Pre-Planned Restoration Algorithm for Tactical Networks. Algorithms and Message Formats.

## B.1 Overview

Appendix B presents additional information about PPR-TN algorithm that was not covered in its description given in Chapter 4. These are detailed message formats (section B.2) and block-scheme of the algorithm executed at every network node during the algorithm functioning (section B.3).

## B.2 PPR-TN. Message Formats

Table B.1 PPR-TN. Request message format.

| Field Name | Size | Description |
|---|---|---|
| Message Type | 2 bits | Message type ID |
| Sender ID | 2 bytes | Identify Sender and Failure location. |
| Failed Node ID | 2 bytes | |
| Connection Priority | 2 bits | VP/VC priority |
| Bandwidth | 1 byte | VP/VC bandwidth |
| Backup VPI/VCI | 3 bytes | Backup VPI/VCI value |
| Route length | 4 bits | Length of the route passed by this message. |
| Transit Node ID | 2 bytes * N | Transit nodes' identification, and route information (it is taken from node's switch table). |
| $B_4$, $B_3$, $B_2$, $B_1$ | 4 bytes | |
| **End of message** | | |

Table B.2 PPR-TN. Cancel message format.

| Field Name | Size | Description |
|---|---|---|
| Message Type | 2 bits | Message type ID |
| Transit Node ID | 2 bytes | Identify Message Originator and Failure location. |
| Failed Node ID | 2 bytes | |
| Connection Priority | 2 bits | VP/VC priority |
| Bandwidth | 1 byte | VP/VC bandwidth |
| Backup VPI/VCI | 3 bytes | Backup VPI/VCI value |
| Route length | 4 bits | Route length. |
| Node IDs | 2 bytes * N | Route identification. N – route length. |
| **End of message** | | |

Table B.3 PPR-TN. Confirmation message format.

| Field Name | Size | Description |
|---|---|---|
| Message Type | 2 bits | Message type ID |
| Chooser ID | 2 bytes | Identify message sender and Failure location. |
| Failed Node ID | 2 bytes | |
| VPI/VCI | 3 bytes | VPI/VCI of the connection being restored |
| Connection Priority | 2 bits | VP/VC priority |
| Bandwidth | 1 byte | VP/VC bandwidth |
| Route length | 4 bits | Route length. |
| Transit Node ID | 2 bytes * N | Route identification. N – route length. |
| **End of message** | | |

## B.3 PPR-TN. Algorithm Block-Scheme

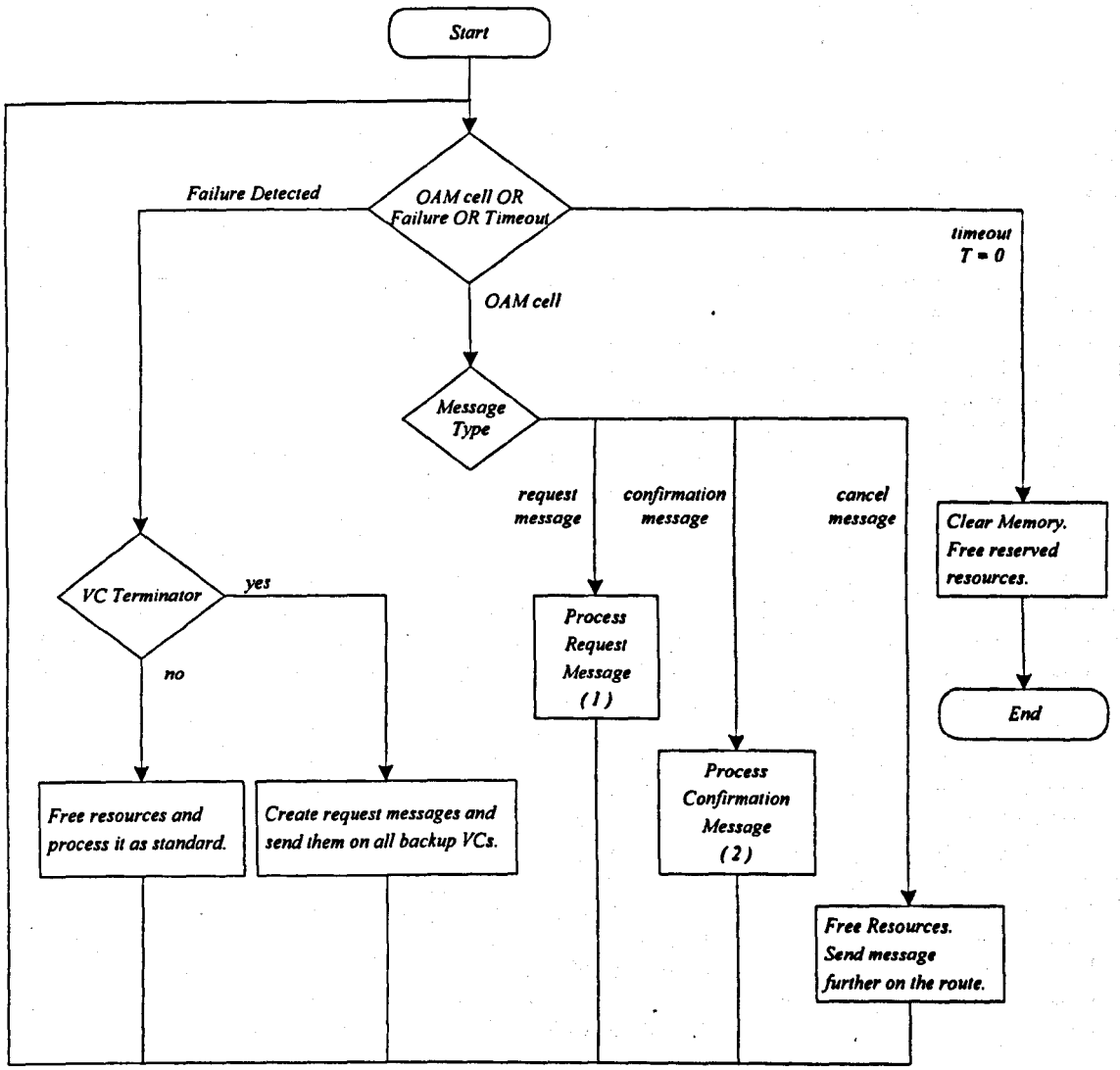



Figure B.1 PPR-TN: Block-Scheme of the Algorithm.

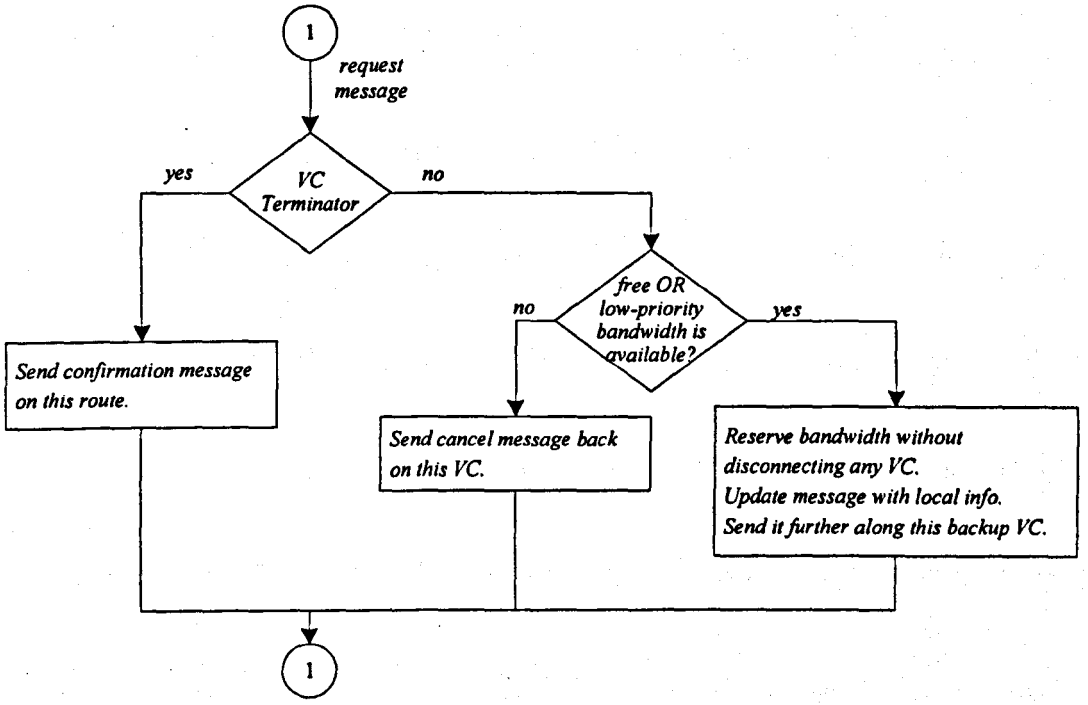



Figure B.2 PPR-TN: Request Message Processing Algorithm.

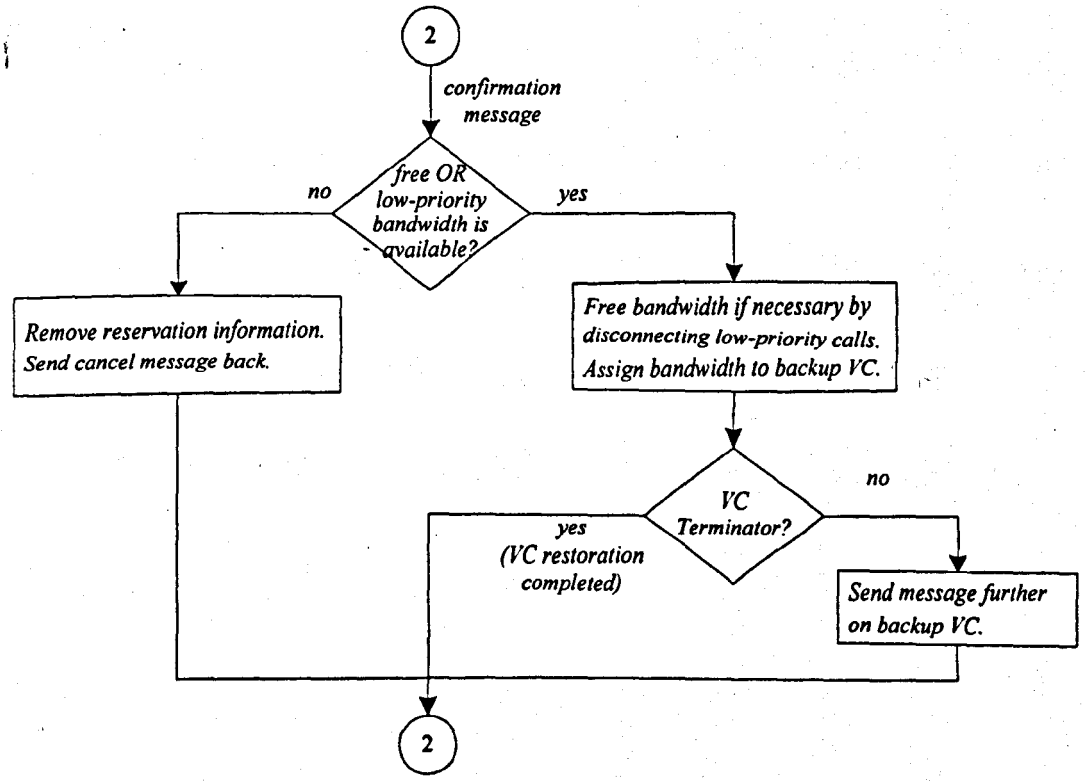



Figure B.3 PPR-TN: Confirmation Message Processing Algorithm.

# Appendix C Simulation Results

## C.1 Overview

To verify the proposed restoration algorithms and compare their performance against known algorithms a number of experiments were conducted. Four restoration algorithms (DRA-TN, PPR-TN, Komine and VPPS algorithms) were tested on three network topologies (Rudin, LATA and US networks). Consequently, 12 sets of experiments were completed. Two more sets of experiments were done to test several modifications of the proposed algorithms. Also, a number of additional experiments were conducted on DRA-TN and Komine algorithms to identify optimal HLC values for particular network topologies.

All the results were processed and average values of restoration parameters calculated. These results are compared and explained in Chapter 6. Full experimental results and graphs drawn for the most important parameters (restoration time, number of messages, restoration thresholds for each traffic priority and all the traffic in total) are presented in this Appendix.

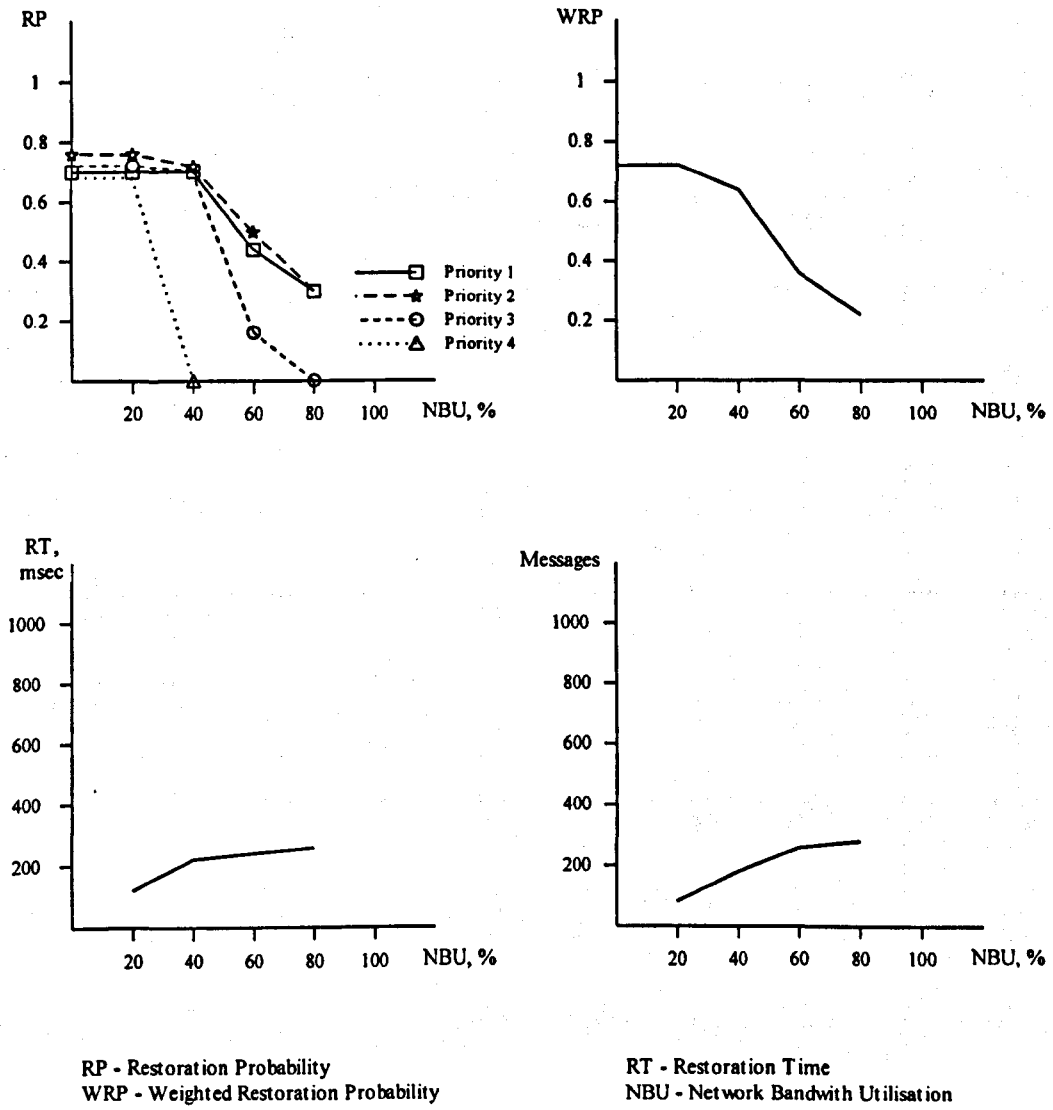
## C.2 DRA-TN

### C.2.1 Simulation Results for Rudin Network

Hop Limit Counter in this experiment was set to 2.

Table C.1 Simulation results: DRA-TN Algorithm; Rudin Network.

| **Simulation Parameter** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 72 | 152 | 232 | 305 |
| Number of Failed VPs | 23 | 47 | 72 | 96 |
| Failure Type | Single Node (2, 3, 4, 5, 6, 7) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 81 | 184 | 251 | 277 |
| • Search Messages | 29 | 29 | 29 | 29 |
| • Acknowledge Messages | 52 | 104 | 111 | 88 |
| • Cancel Messages | 0 | 1 | 2 | 1 |
| • Disconnect Messages | 0 | 52 | 110 | 159 |
| Restoration Time, msec | 121 | 224 | 246 | 250 |
| Weighted Restoration Probability | 0.72 | 0.64 | 0.36 | 0.21 |
| $RP_1$ | 0.68 | 0.70 | 0.45 | 0.31 |
| $RP_2$ | 0.77 | 0.72 | 0.50 | 0.30 |
| $RP_3$ | 0.74 | 0.70 | 0.15 | 0 |
| $RP_4$ | 0.70 | 0 | 0 | 0 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

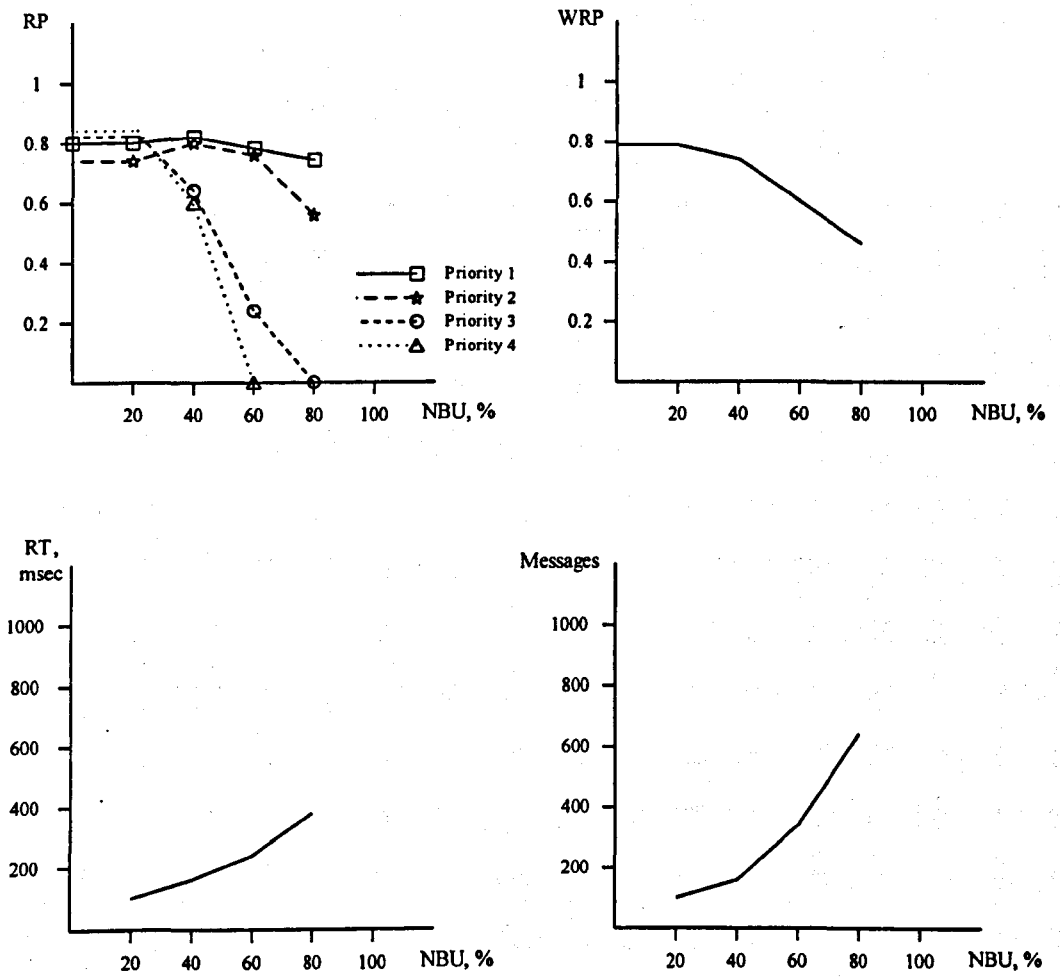
Figure C.1 Simulation results: DRA-TN Algorithm; Rudin Network.

### C.2.2 Simulation Results for LATA Network

Hop Limit Counter in this experiment was set to 2.

Table C.2 Simulation results: DRA-TN Algorithm; LATA Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 132 | 268 | 410 | 546 |
| Number of Failed VPs | 27 | 53 | 80 | 106 |
| Failure Type | Single Node (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 102 | 170 | 351 | 639 |
| • Search Messages | 58 | 57 | 57 | 57 |
| • Acknowledge Messages | 44 | 92 | 133 | 169 |
| • Cancel Messages | 0 | 0 | 1 | 3 |
| • Disconnect Messages | 0 | 21 | 160 | 410 |
| Restoration Time, msec | 94 | 167 | 250 | 383 |
| Weighted Restoration Probability | 0.79 | 0.75 | 0.59 | 0.47 |
| $RP_1$ | 0.80 | 0.82 | 0.79 | 0.74 |
| $RP_2$ | 0.74 | 0.80 | 0.76 | 0.58 |
| $RP_3$ | 0.82 | 0.62 | 0.25 | 0 |
| $RP_4$ | 0.85 | 0.60 | 0 | 0 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

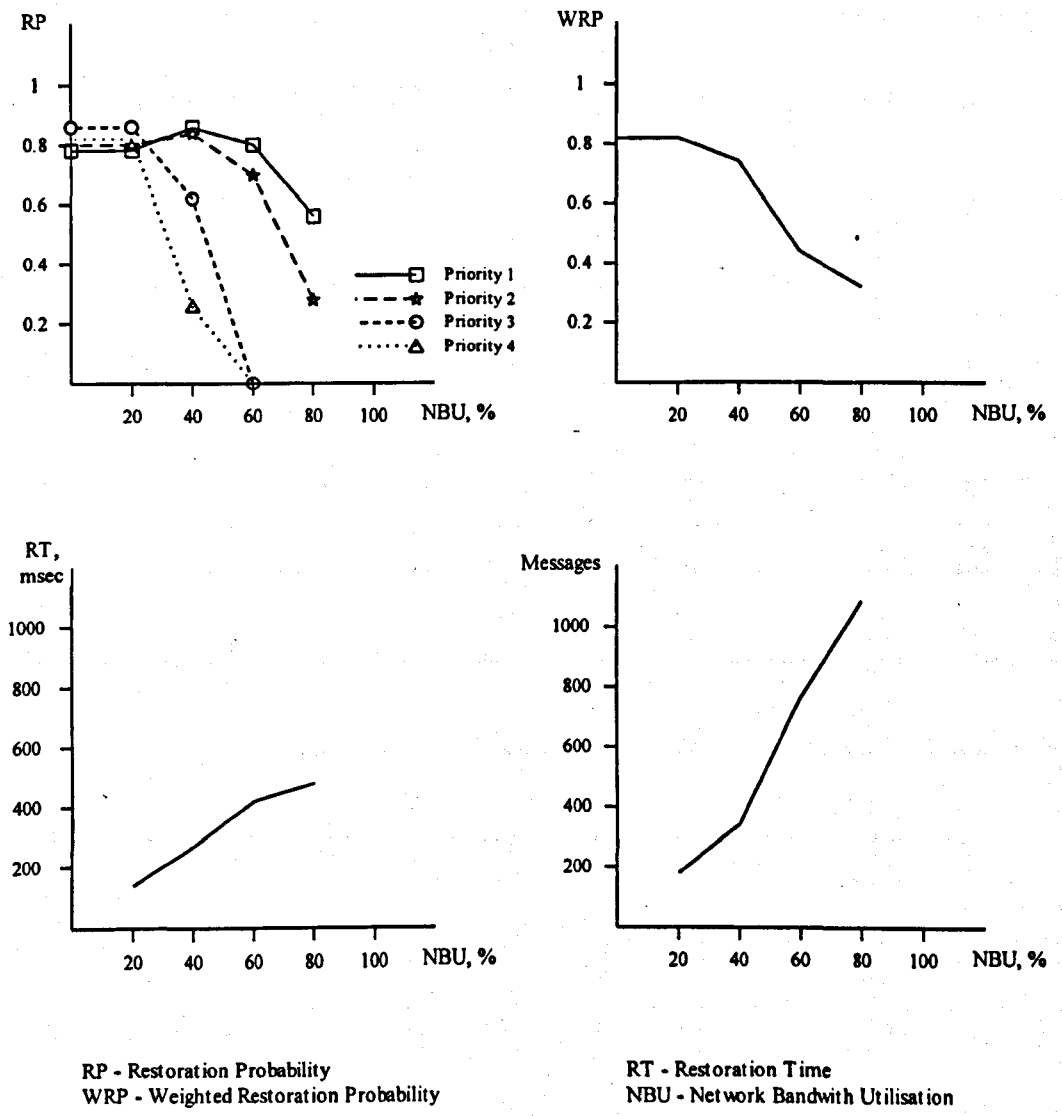
Figure C.2 Simulation results: DRA-TN Algorithm; LATA Network.

### C.2.3 Simulation Results for US Network

Hop Limit Counter in this experiment was set to 3.

Table C.3 Simulation results: DRA-TN Algorithm; US Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 172 | 356 | 536 | 715 |
| Number of Failed VPs | 32 | 67 | 99 | 132 |
| Failure Type | Single Node (7, 8, 9, 10, 11, 12, 14, 15, 16, 20, 21, 22, 23, 24, 25, 26) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 186 | 334 | 730 | 1040 |
| • Search Messages | 105 | 101 | 102 | 101 |
| • Acknowledge Messages | 81 | 168 | 241 | 238 |
| • Cancel Messages | 0 | 0 | 3 | 4 |
| • Disconnect Messages | 0 | 65 | 384 | 696 |
| Restoration Time, msec | 140 | 272 | 411 | 485 |
| Weighted Restoration Probability | 0.81 | 0.75 | 0.54 | 0.32 |
| $RP_1$ | 0.78 | 0.86 | 0.81 | 0.58 |
| $RP_2$ | 0.80 | 0.84 | 0.71 | 0.29 |
| $RP_3$ | 0.87 | 0.61 | 0 | 0 |
| $RP_4$ | 0.84 | 0.28 | 0 | 0 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

Figure C.3 Simulation results: DRA-TN Algorithm; US Network.

## C.3 DRA-TN with Restoration Threshold Modification

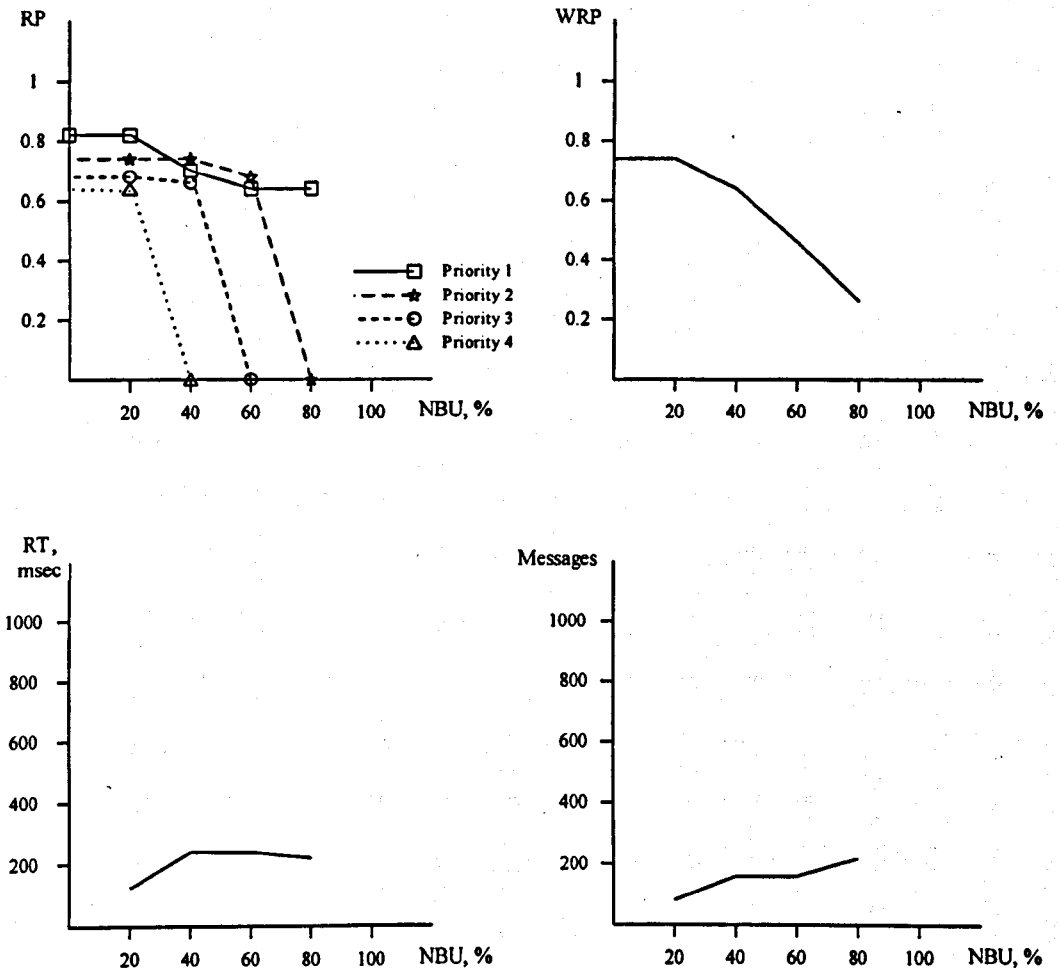
In this section results of modelling DRA-TN algorithm with Restoration Threshold modification are presented. The following threshold values were set for priorities four, three and two respectively: $P_4$=45, $P_3$=60, $P_2$=75.

### C.3.1 Simulation Results for Rudin Network

Hop Limit Counter in this experiment was set to 2.

Table C.4 Simulation results: DRA-TN Algorithm (RT); Rudin Network.

| **Simulation Parameter** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 72 | 152 | 230 | 305 |
| Number of Failed VPs | 23 | 47 | 71 | 94 |
| Failure Type | Single Node (2, 3, 4, 5, 6, 7) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 81 | 180 | 183 | 205 |
| • Search Messages | 29 | 29 | 29 | 29 |
| • Acknowledge Messages | 52 | 101 | 84 | 58 |
| • Cancel Messages | 0 | 1 | 0 | 0 |
| • Disconnect Messages | 0 | 49 | 70 | 118 |
| Restoration Time, msec | 126 | 222 | 214 | 207 |
| Weighted Restoration Probability | 0.75 | 0.64 | 0.46 | 0.26 |
| $RP_1$ | 0.81 | 0.67 | 0.63 | 0.64 |
| $RP_2$ | 0.75 | 0.75 | 0.70 | 0 |
| $RP_3$ | 0.67 | 0.71 | 0 | 0 |
| $RP_4$ | 0.69 | 0 | 0 | 0 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

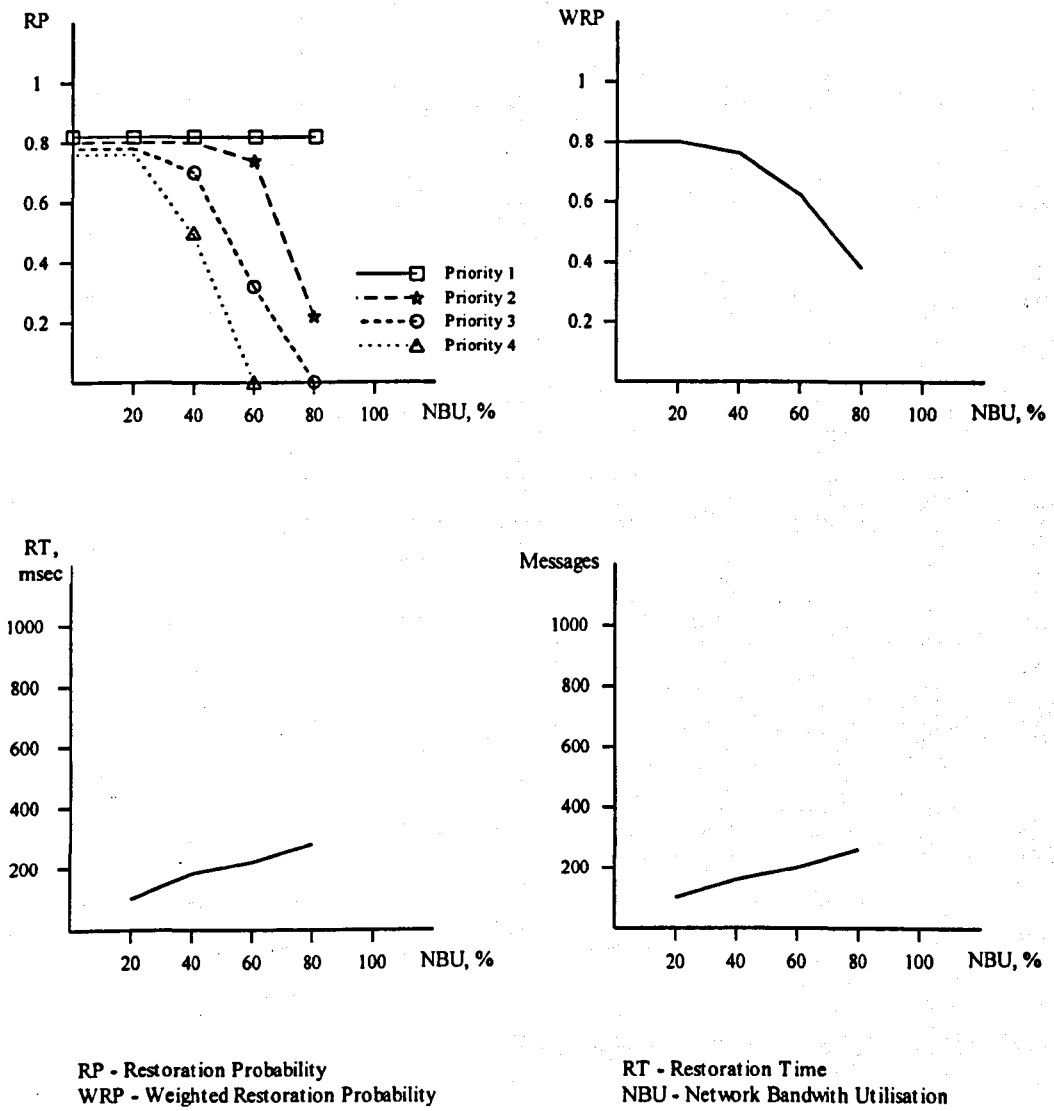
Figure C.4 Simulation results: DRA-TN Algorithm (RT); Rudin Network.

### C.3.2 Simulation Results for LATA Network

Hop Limit Counter in this experiment was set to 2.

Table C.5 Simulation results: DRA-TN Algorithm (RT); LATA Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 132 | 268 | 410 | 547 |
| Number of Failed VPs | 27 | 53 | 80 | 106 |
| Failure Type | Single Node (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 102 | 160 | 196 | 279 |
| • Search Messages | 58 | 57 | 57 | 57 |
| • Acknowledge Messages | 44 | 87 | 90.00 | 73 |
| • Cancel Messages | 0 | 0 | 0 | 0 |
| • Disconnect Messages | 0 | 16 | 49 | 148 |
| Restoration Time, msec | 95 | 158 | 185 | 221 |
| Weighted Restoration Probability | 0.80 | 0.76 | 0.62 | 0.39 |
| $RP_1$ | 0.81 | 0.80 | 0.82 | 0.82 |
| $RP_2$ | 0.81 | 0.78 | 0.74 | 0.22 |
| $RP_3$ | 0.78 | 0.76 | 0.33 | 0 |
| $RP_4$ | 0.81 | 0.49 | 0 | 0 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

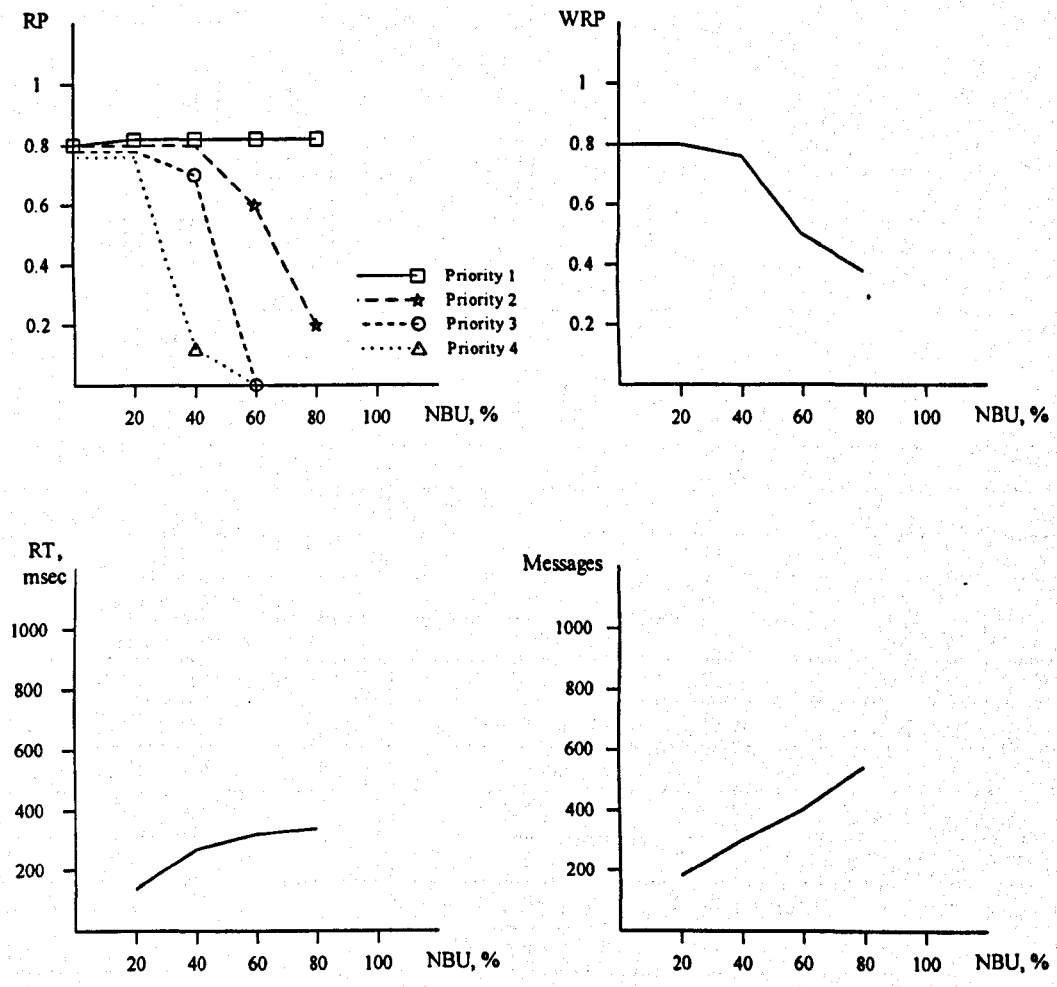
Figure C.5 Simulation results: DRA-TN Algorithm (RT); LATA Network.

### C.3.3 Simulation Results for US Network

Hop Limit Counter in this experiment was set to 3.

Table C.6 Simulation results: DRA-TN Algorithm (RT); US Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 172 | 356 | 536 | 713 |
| Number of Failed VPs | 32 | 67 | 99 | 131 |
| Failure Type | Single Node (7, 8, 9, 10, 11, 12, 14, 15, 16, 20, 21, 22, 23, 24, 25, 26) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 186 | 303 | 405 | 535 |
| • Search Messages | 105 | 101 | 102 | 101 |
| • Acknowledge Messages | 81 | 155 | 144 | 102 |
| • Cancel Messages | 0 | 0 | 0 | 0 |
| • Disconnect Messages | 0 | 47 | 160 | 331 |
| Restoration Time, msec | 140 | 247 | 256 | 266 |
| Weighted Restoration Probability | 0.80 | 0.76 | 0.51 | 0.38 |
| $RP_1$ | 0.77 | 0.84 | 0.82 | 0.81 |
| $RP_2$ | 0.78 | 0.88 | 0.60 | 0.20 |
| $RP_3$ | 0.85 | 0.72 | 0 | 0 |
| $RP_4$ | 0.88 | 0.13 | 0 | 0 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

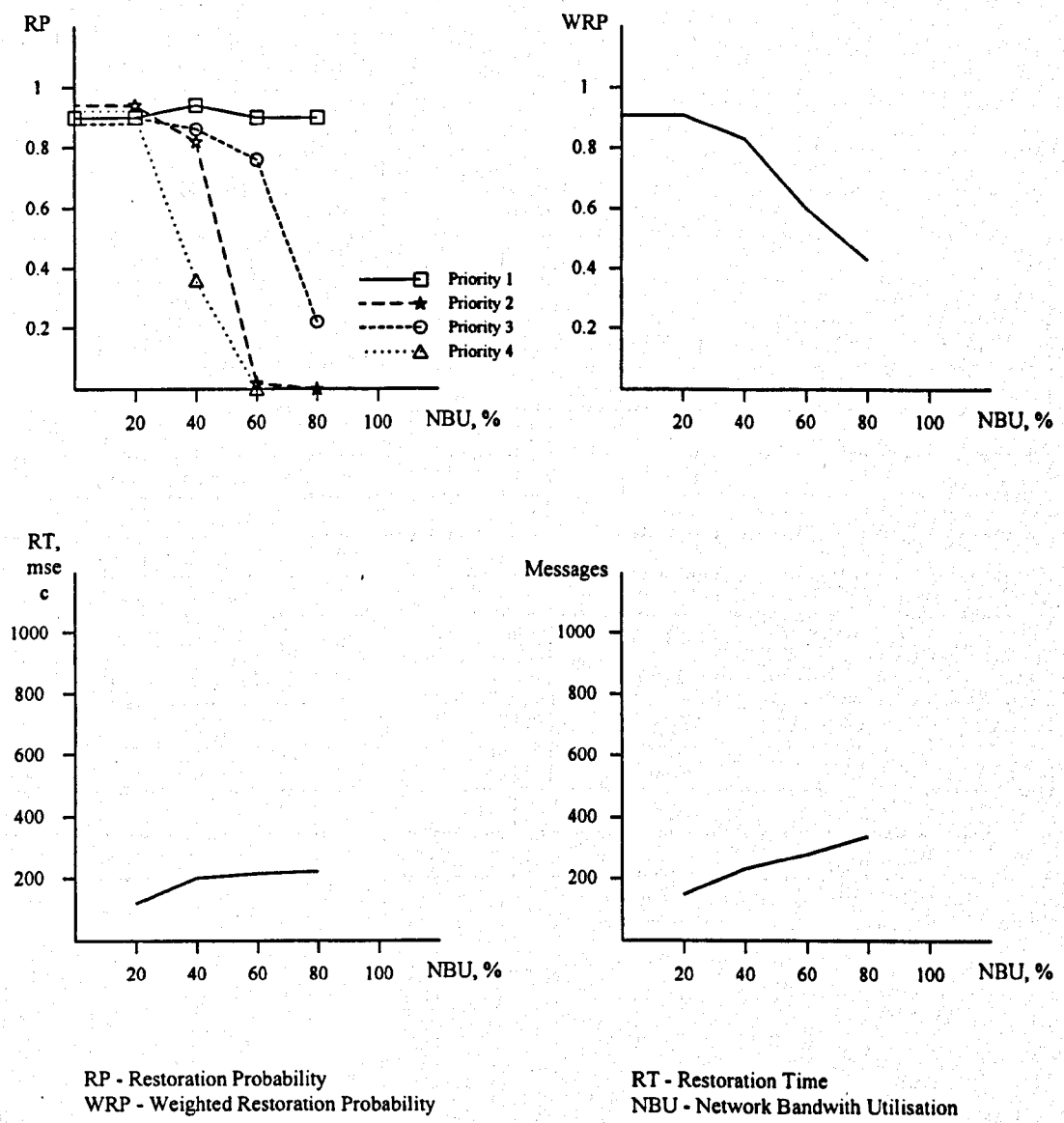
Figure C.6 Simulation results: DRA-TN Algorithm (RT); US Network

### C.3.4 Simulation Results for LATA Network, HLC = 3

Hop Limit Counter in this experiment was set to 3.

Table C.7 Simulation results: DRA-TN Algorithm (RT); LATA Network; HLC=3.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 132 | 268 | 411 | 540 |
| Number of Failed VPs | 27 | 53 | 80 | 105 |
| Failure Type | Single Node (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 149 | 233 | 285 | 341 |
| • Search Messages | 93 | 92 | 92 | 91 |
| • Acknowledge Messages | 56 | 109 | 107 | 82 |
| • Cancel Messages | 0 | 0 | 0 | 0 |
| • Disconnect Messages | 0 | 32 | 86 | 168 |
| Restoration Time, msec | 117 | 199 | 205 | 215 |
| Weighted Restoration Probability | 0.91 | 0.83 | 0.60 | 0.43 |
| $RP_1$ | 0.90 | 0.94 | 0.90 | 0.90 |
| $RP_2$ | 0.90 | 0.85 | 0.77 | 0.22 |
| $RP_3$ | 0.95 | 0.84 | 0.05 | 0 |
| $RP_4$ | 0.92 | 0.35 | 0 | 0 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

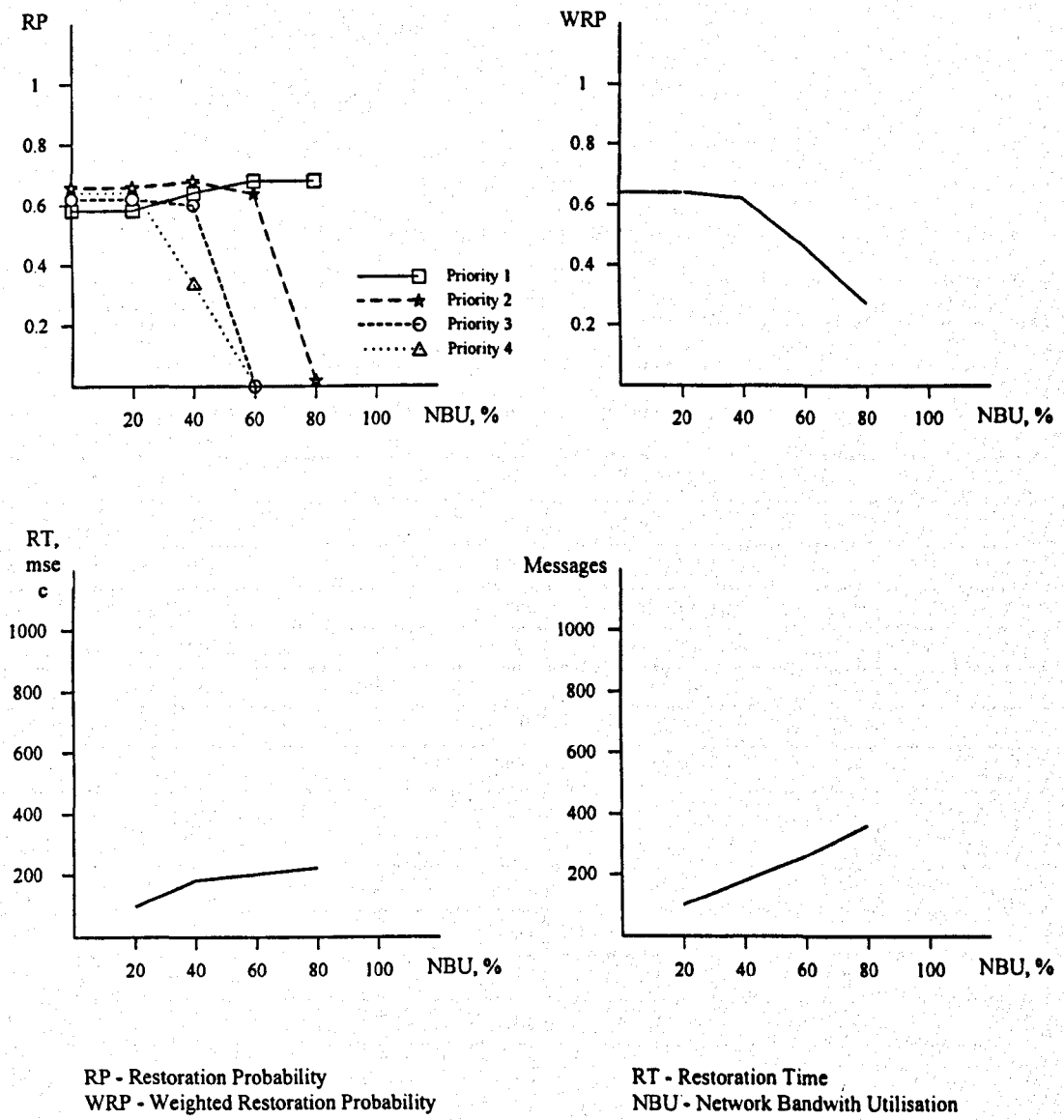
Figure C.7 Simulation results: DRA-TN Algorithm (RT); LATA Network; HLC=3.

### C.3.5 Simulation Results for US Network, HLC = 2

Hop Limit Counter in this experiment was set to 2.

Table C.8 Simulation results: DRA-TN Algorithm (RT); US Network; HLC=2.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 172 | 356 | 536 | 713 |
| Number of Failed VPs | 32 | 67 | 99 | 130 |
| Failure Type | Single Node (7, 8, 9, 10, 11, 12, 14, 15, 16, 20, 21, 22, 23, 24, 25, 26) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 108 | 183 | 264 | 368 |
| • Search Messages | 52 | 52 | 52 | 52 |
| • Acknowledge Messages | 56 | 110 | 105 | 79 |
| • Cancel Messages | 0 | 0 | 0 | 0 |
| • Disconnect Messages | 0 | 21 | 107 | 237 |
| Restoration Time, msec | 103 | 191 | 201 | 212 |
| Weighted Restoration Probability | 0.63 | 0.62 | 0.46 | 0.28 |
| $RP_1$ | 0.59 | 0.64 | 0.68 | 0.69 |
| $RP_2$ | 0.66 | 0.69 | 0.63 | 0.02 |
| $RP_3$ | 0.63 | 0.61 | 0 | 0 |
| $RP_4$ | 0.68 | 0.36 | 0 | 0 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

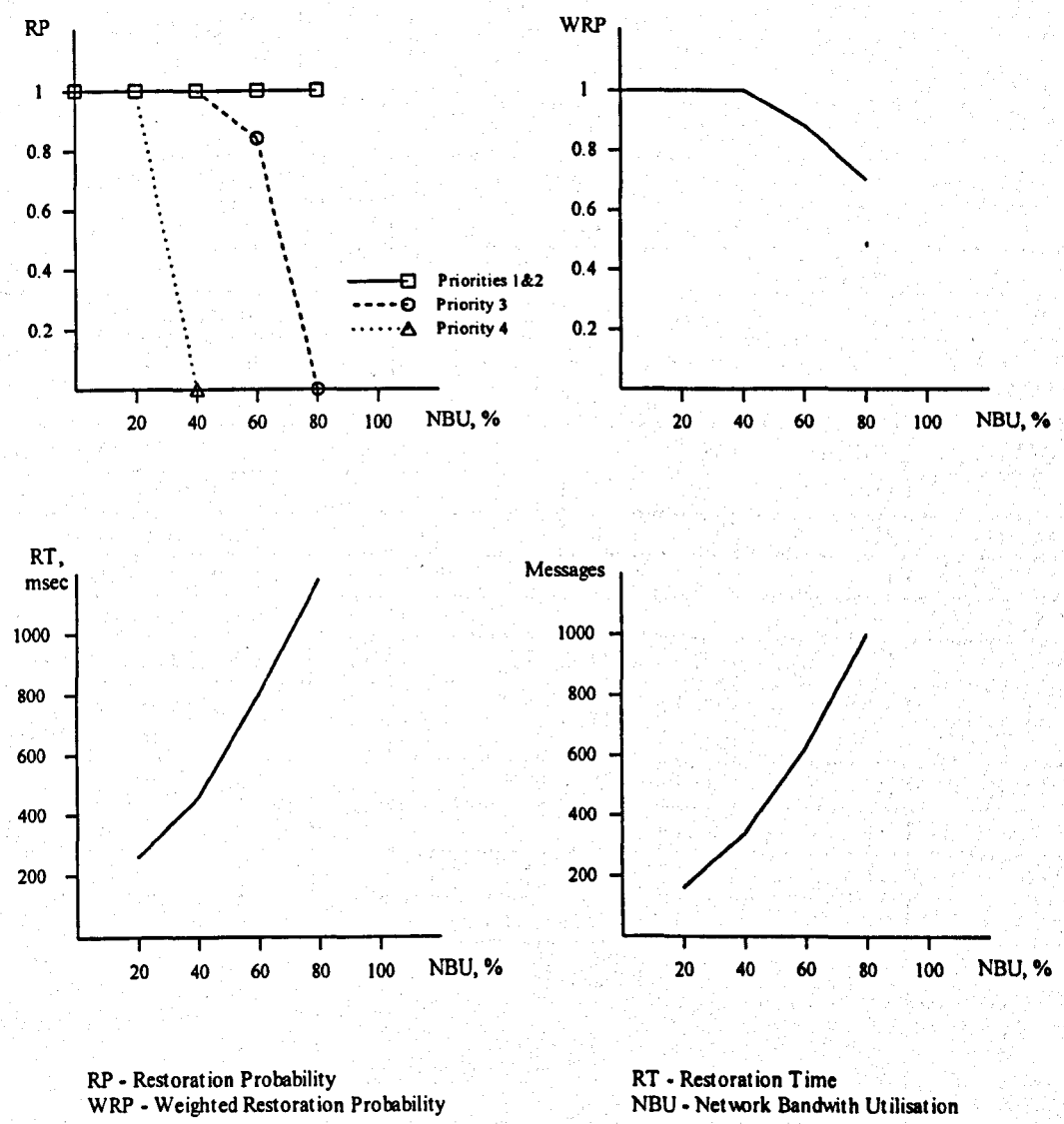
Figure C.8 Simulation results: DRA-TN Algorithm (RT); US Network; HLC=2.

## C.4 PPR-TN

### C.4.1 Simulation Results for Rudin Network

Table C.9 Simulation results: PPR-TN Algorithm; Rudin Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 72 | 152 | 229 | 304 |
| Number of Failed VPs | 23 | 47 | 71 | 94 |
| Failure Type | Single Node (2, 3, 4, 5, 6, 7) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 165 | 343 | 625 | 1001 |
| • Search Messages | 89 | 185 | 279 | 370 |
| • Acknowledge Messages | 76 | 158 | 235 | 299 |
| • Cancel Messages | 0 | 0 | 0 | 3 |
| • Disconnect Messages | 0 | 0 | 111 | 329 |
| Restoration Time, msec | 249 | 479 | 806 | 1192 |
| Weighted Restoration Probability | 1.00 | 1.00 | 0.87 | 0.70 |
| $RP_1$ | 1.00 | 1.00 | 1.00 | 1.00 |
| $RP_2$ | 1.00 | 1.00 | 1.00 | 1.00 |
| $RP_3$ | 1.00 | 1.00 | 0.83 | 0 |
| $RP_4$ | 1.00 | 1.00 | 0 | 0 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

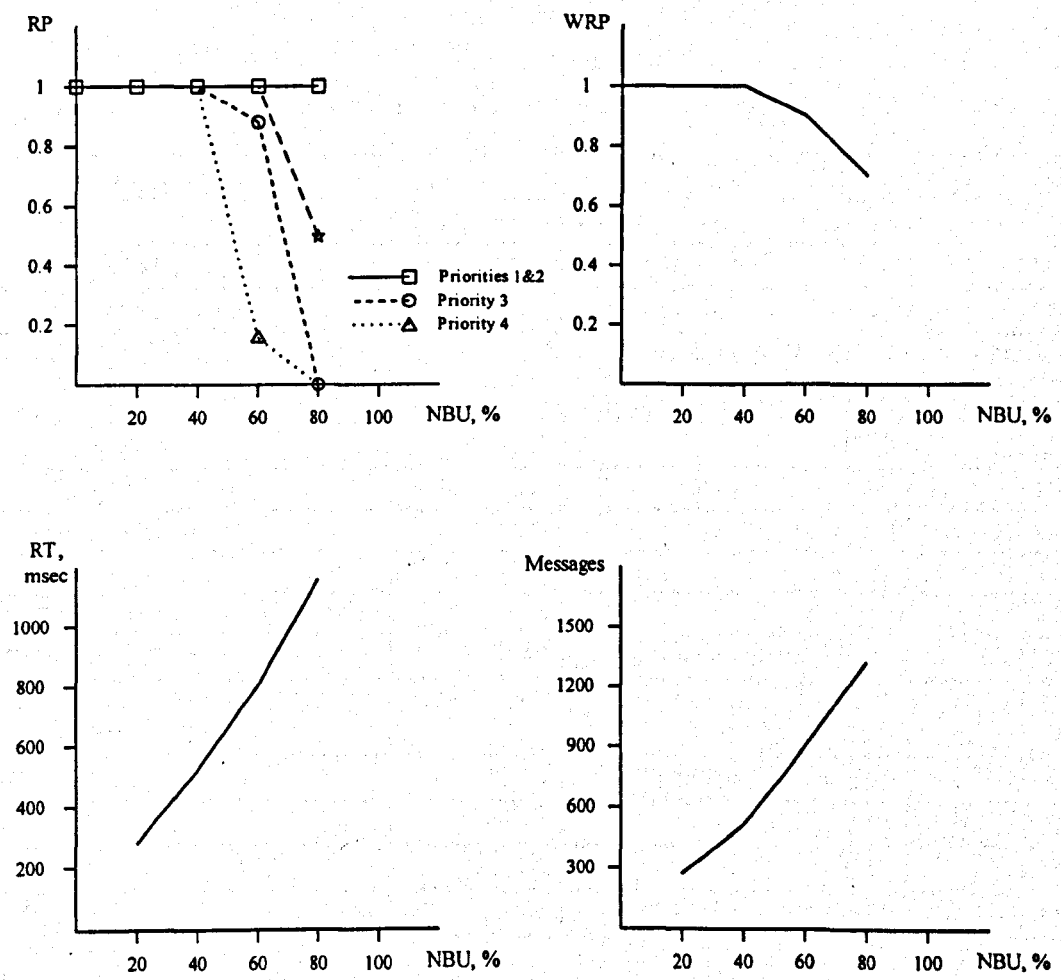
RT - Restoration Time
NBU - Network Bandwith Utilisation

Figure C.9 Simulation results: PPR-TN Algorithm; Rudin Network.

## C.4.2 Simulation Results for LATA Network

Table C.10 Simulation results: PPR-TN Algorithm; LATA Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 132 | 268 | 410 | 550 |
| Number of Failed VPs | 27 | 53 | 80 | 107 |
| Failure Type | Single Node (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 253 | 503 | 802 | 1330 |
| • Search Messages | 140 | 278 | 410 | 543 |
| • Acknowledge Messages | 113 | 225 | 328 | 427 |
| • Cancel Messages | 0 | 0 | 1 | 3 |
| • Disconnect Messages | 0 | 0 | 63 | 357 |
| Restoration Time, msec | 272 | 521 | 797 | 1178 |
| Weighted Restoration Probability | 1.00 | 1.00 | 0.89 | 0.70 |
| $RP_1$ | 1.00 | 1.00 | 1.00 | 1.00 |
| $RP_2$ | 1.00 | 1.00 | 1.00 | 1.00 |
| $RP_3$ | 1.00 | 1.00 | 0.88 | 0.00 |
| $RP_4$ | 1.00 | 1.00 | 0.17 | 0 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

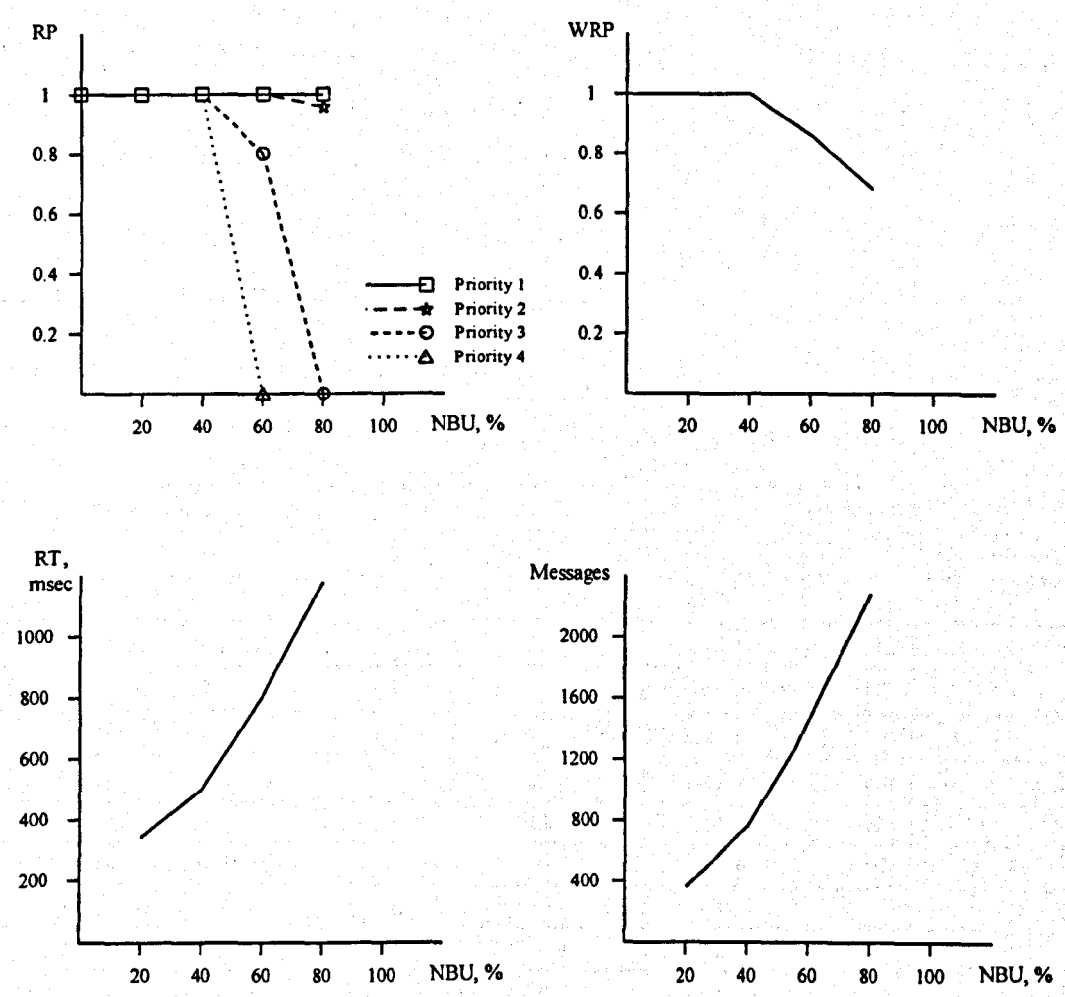
RT - Restoration Time
NBU - Network Bandwith Utilisation

Figure C.10 Simulation results: PPR-TN Algorithm; LATA Network.

### C.4.3 Simulation Results for US Network

Table C.11 Simulation results: PPR-TN Algorithm; US Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 172 | 356 | 536 | 715 |
| Number of Failed VPs | 32 | 67 | 99 | 131 |
| Failure Type | Single Node (7, 8, 9, 10, 11, 12, 14, 15, 16, 20, 21, 22, 23, 24, 25, 26) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 365 | 734 | 1288 | 2209 |
| • Search Messages | 191 | 386 | 571 | 756 |
| • Acknowledge Messages | 174 | 348 | 512 | 652 |
| • Cancel Messages | 0 | 0 | 2 | 8 |
| • Disconnect Messages | 0 | 0 | 203 | 793 |
| Restoration Time, msec | 258 | 489 | 792 | 1182 |
| Weighted Restoration Probability | 1.00 | 1.00 | 0.86 | 0.69 |
| $RP_1$ | 1.00 | 1.00 | 1.00 | 1.00 |
| $RP_2$ | 1.00 | 1.00 | 1.00 | 0.96 |
| $RP_3$ | 1.00 | 1.00 | 0.80 | 0.00 |
| $RP_4$ | 1.00 | 1.00 | 0.00 | 0.00 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

Figure C.11 Simulation results: PPR-TN Algorithm; US Network.

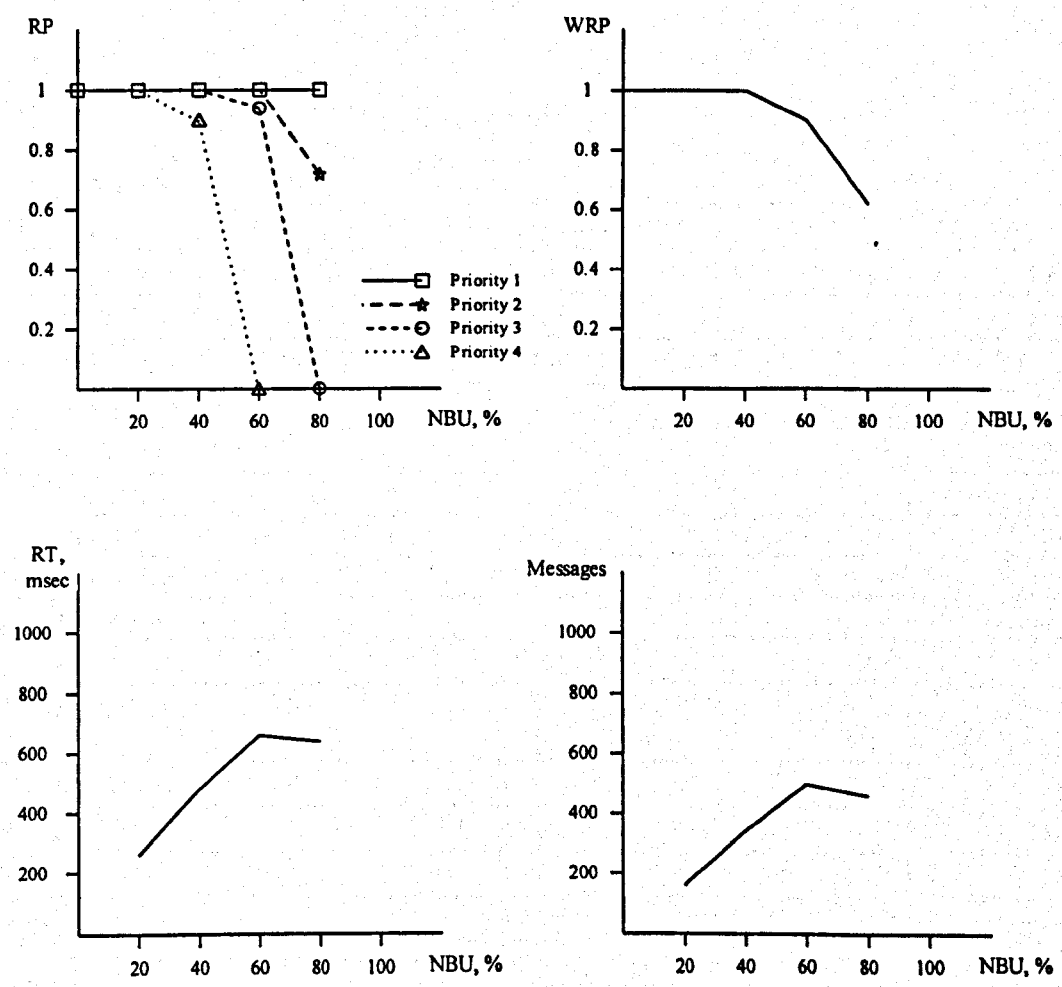
## C.5 PPR-TN with Restoration Threshold Modification

In this section results of modelling PPR-TN algorithm with Restoration Threshold modification are presented. The following threshold values were set for priorities four, three and two respectively: $P_4$=50, $P_3$=65, $P_2$=80.

### C.5.1 Simulation Results for Rudin Network

Table C.12 Simulation results: PPR-TN Algorithm (RT); Rudin Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 72 | 152 | 232 | 306 |
| Number of Failed VPs | 23 | 47 | 72 | 95 |
| Failure Type | Single Node (2, 3, 4, 5, 6, 7) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 165 | 344 | 504 | 468 |
| • Search Messages | 89 | 184 | 225 | 175 |
| • Acknowledge Messages | 76 | 157 | 192 | 153 |
| • Cancel Messages | 0 | 0 | 0 | 0 |
| • Disconnect Messages | 0 | 3 | 87 | 140 |
| Restoration Time, msec | 249 | 484 | 661 | 651 |
| Weighted Restoration Probability | 1.00 | 0.99 | 0.89 | 0.62 |
| $RP_1$ | 1.00 | 1.00 | 1.00 | 1.00 |
| $RP_2$ | 1.00 | 1.00 | 1.00 | 0.72 |
| $RP_3$ | 1.00 | 1.00 | 0.95 | 0 |
| $RP_4$ | 1.00 | 0.91 | 0 | 0 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

Figure C.12 Simulation results: PPR-TN Algorithm (RT); Rudin Network.

## C.5.2 Simulation Results for LATA Network

Table C.13 Simulation results: PPR-TN Algorithm (RT); LATA Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 132 | 267 | 409 | 545 |
| Number of Failed VPs | 27 | 53 | 79 | 106 |
| Failure Type | Single Node (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 253 | 504 | 608 | 550 |
| • Search Messages | 140 | 279 | 312 | 222 |
| • Acknowledge Messages | 113 | 225 | 253 | 176 |
| • Cancel Messages | 0 | 0 | 0 | 0 |
| • Disconnect Messages | 0 | 0 | 43 | 152 |
| Restoration Time, msec | 272 | 521 | 604 | 594 |
| Weighted Restoration Probability | 1.00 | 1.00 | 0.88 | 0.56 |
| $RP_1$ | 1.00 | 1.00 | 1.00 | 1.00 |
| $RP_2$ | 1.00 | 1.00 | 1.00 | 0.51 |
| $RP_3$ | 1.00 | 1.00 | 0.89 | 0.04 |
| $RP_4$ | 1.00 | 1.00 | 0 | 0 |

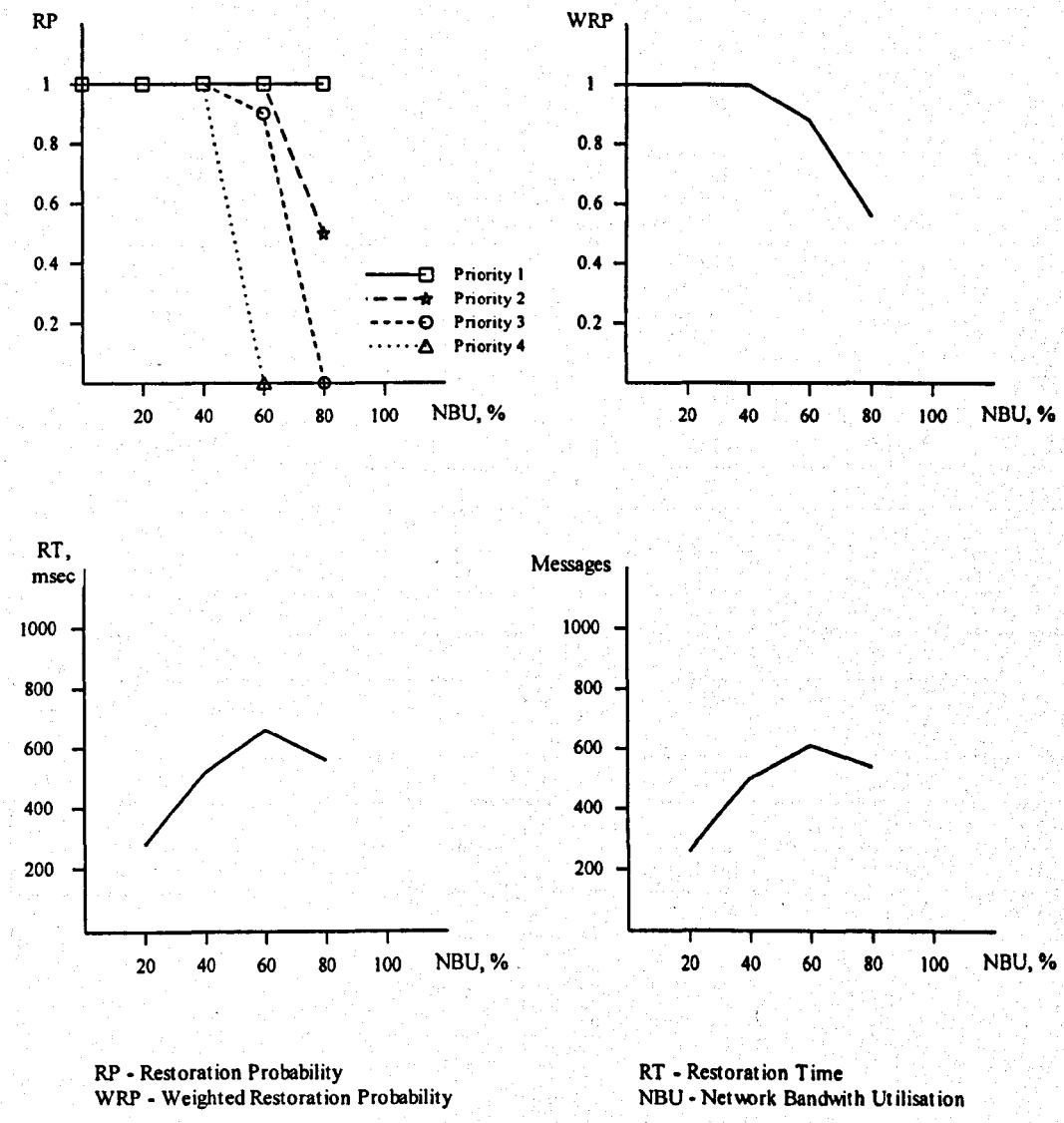
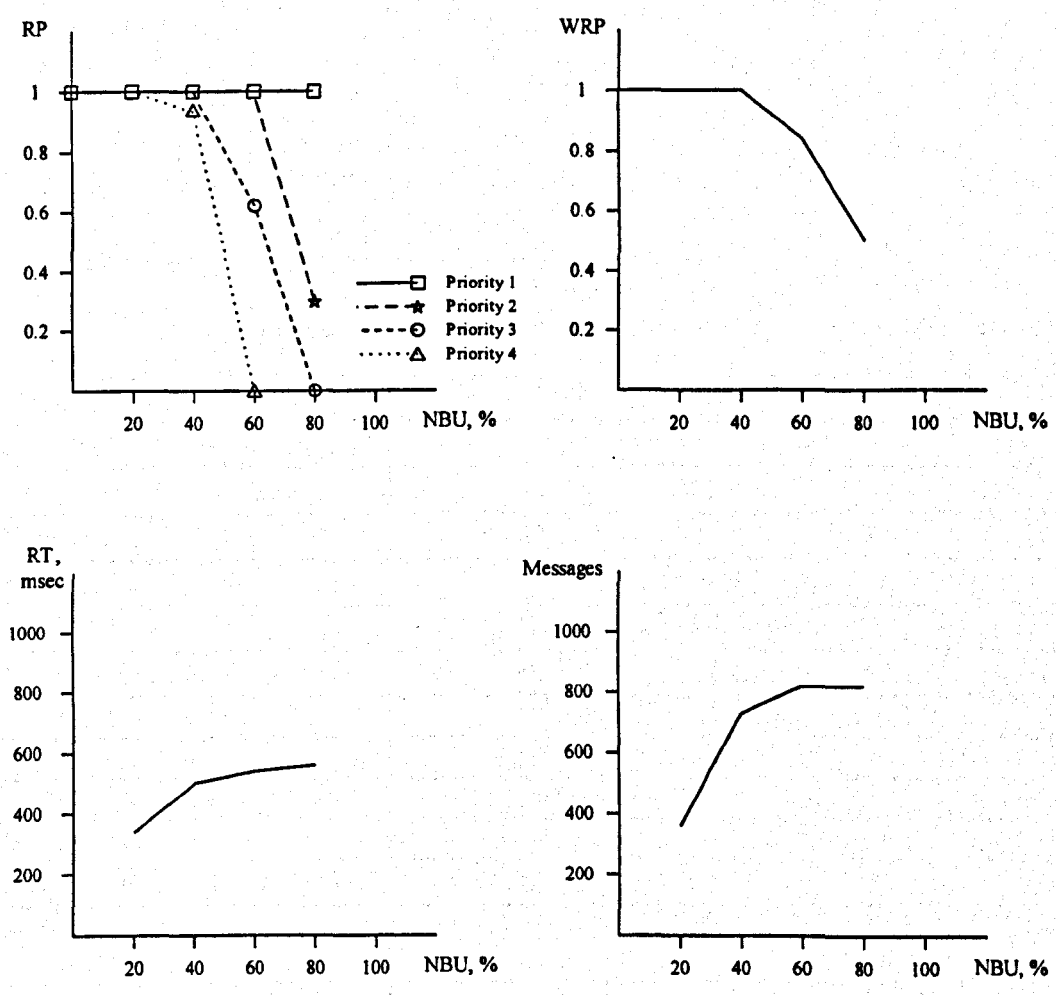



Figure C.13 Simulation results: PPR-TN Algorithm (RT); LATA Network.

### C.5.3 Simulation Results for US Network

Table C.14 Simulation results: PPR-TN Algorithm (RT); US Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 172 | 356 | 536 | 717 |
| Number of Failed VPs | 32 | 67 | 99 | 133 |
| Failure Type | Single Node (7, 8, 9, 10, 11, 12, 14, 15, 16, 20, 21, 22, 23, 24, 25, 26) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 365 | 724 | 810 | 760 |
| • Search Messages | 191 | 381 | 380 | 259 |
| • Acknowledge Messages | 174 | 343 | 345 | 235 |
| • Cancel Messages | 0 | 0 | 0 | 0 |
| • Disconnect Messages | 0 | 0 | 86 | 265 |
| Restoration Time, msec | 257 | 487 | 550 | 496 |
| Weighted Restoration Probability | 1.00 | 1.00 | 0.83 | 0.49 |
| $RP_1$ | 1.00 | 1.00 | 1.00 | 1.00 |
| $RP_2$ | 1.00 | 1.00 | 1.00 | 0.31 |
| $RP_3$ | 1.00 | 1.00 | 0.63 | 0.00 |
| $RP_4$ | 1.00 | 0.95 | 0.00 | 0.00 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

Figure C.14 Simulation results: PPR-TN Algorithm (RT); US Network.

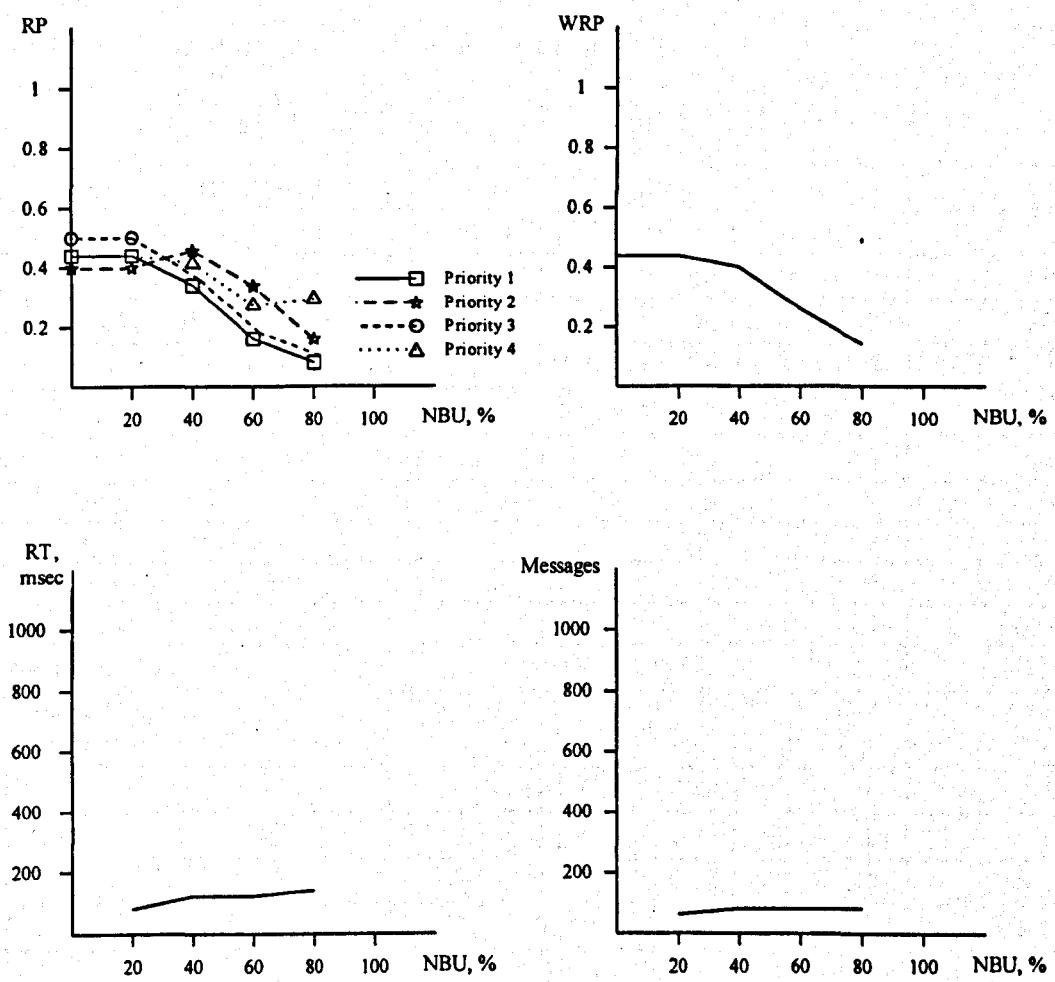
## C.6 Komine Algorithm

### C.6.1 Simulation Results for Rudin Network

Hop Limit Counter in this experiment was set to 3.

Table C.15 Simulation results: Komine Algorithm; Rudin Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 72 | 152 | 227 | 309 |
| Number of Failed VPs | 23 | 47 | 71 | 95 |
| Failure Type | Single Node (2, 3, 4, 5, 6, 7) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 59 | 84 | 83 | 85 |
| • Search Messages | 32 | 32 | 32 | 32 |
| • Acknowledge Messages | 27 | 52 | 50 | 47 |
| • Cancel Messages | 0 | 0 | 1 | 6 |
| • Disconnect Messages | 0 | 0 | 0 | 0 |
| Restoration Time, msec | 87 | 125 | 121 | 139 |
| Weighted Restoration Probability | 0.44 | 0.40 | 0.25 | 0.13 |
| $RP_1$ | 0.44 | 0.36 | 0.17 | 0.09 |
| $RP_2$ | 0.41 | 0.47 | 0.36 | 0.15 |
| $RP_3$ | 0.50 | 0.34 | 0.22 | 0.11 |
| $RP_4$ | 0.41 | 0.42 | 0.27 | 0.29 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

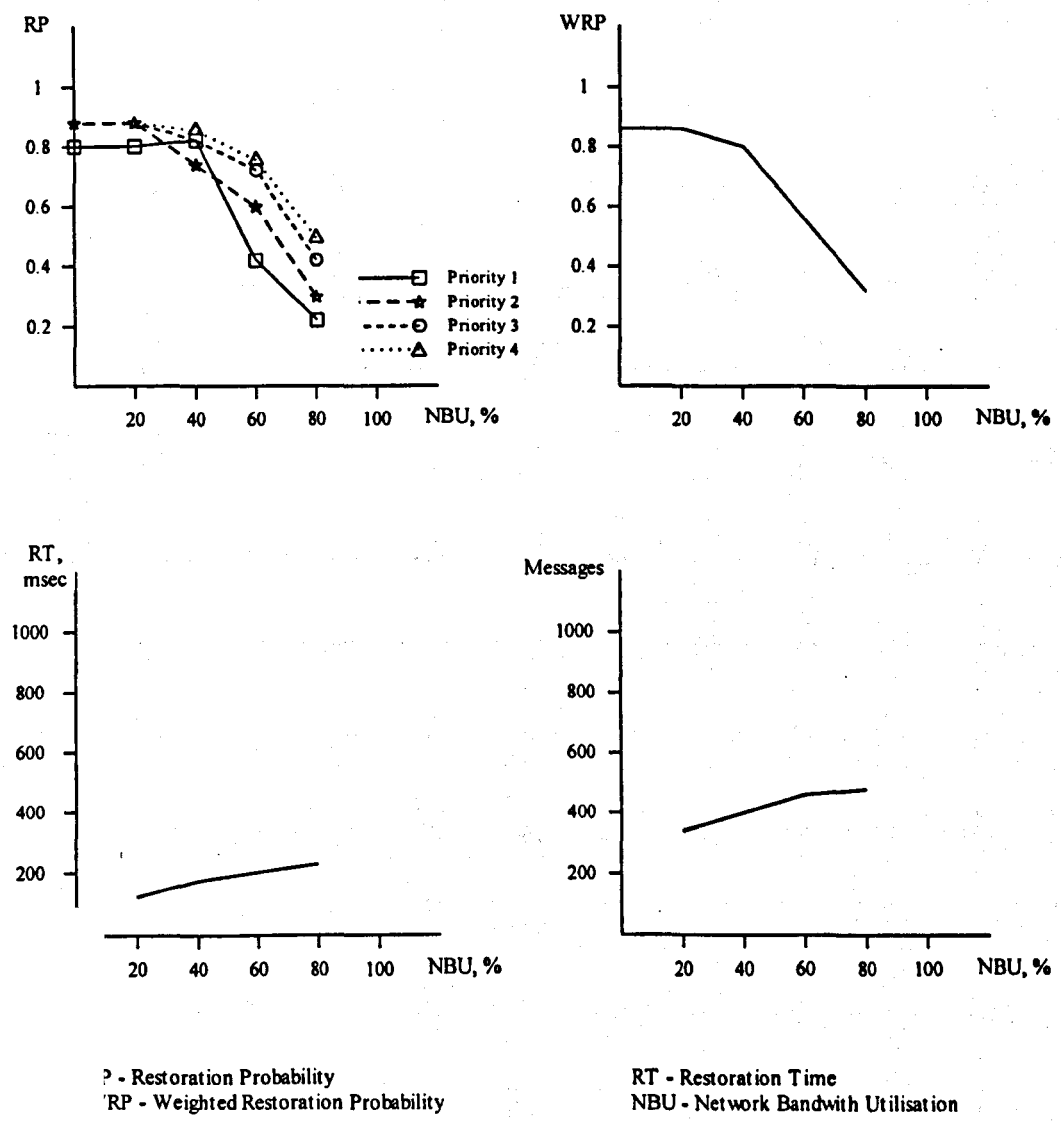
Figure C.15 Simulation results: Komine Algorithm; Rudin Network.

### C.6.2 Simulation Results for LATA Network

Hop Limit Counter in this experiment was set to 4.

Table C.16 Simulation results: Komine Algorithm; LATA Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 132 | 268 | 409 | 546 |
| Number of Failed VPs | 27 | 53 | 80 | 106 |
| Failure Type | Single Node (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 338 | 395 | 458 | 468 |
| • Search Messages | 286 | 286 | 286 | 286 |
| • Acknowledge Messages | 51 | 104 | 141 | 131 |
| • Cancel Messages | 0 | 5 | 31 | 50 |
| • Disconnect Messages | 0 | 0 | 0 | 0 |
| Restoration Time, msec | 112 | 172 | 202 | 224 |
| Weighted Restoration Probability | 0.85 | 0.81 | 0.57 | 0.32 |
| $RP_1$ | 0.81 | 0.83 | 0.44 | 0.22 |
| $RP_2$ | 0.87 | 0.77 | 0.60 | 0.31 |
| $RP_3$ | 0.89 | 0.82 | 0.72 | 0.42 |
| $RP_4$ | 0.88 | 0.84 | 0.73 | 0.50 |




P - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

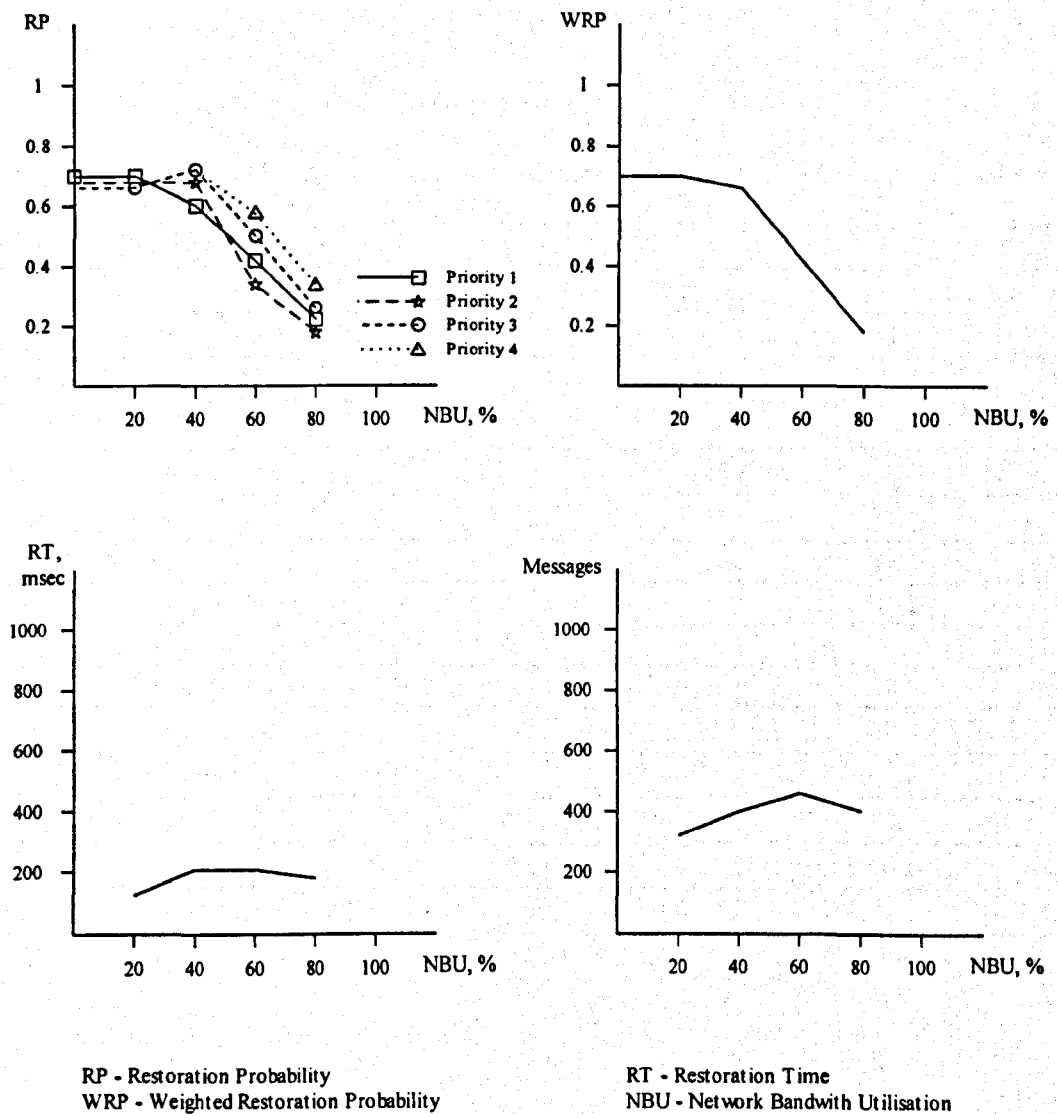
Figure C.16 Simulation results: Komine Algorithm; LATA Network.

### C.6.3 Simulation Results for US Network

Hop Limit Counter in this experiment was set to 4.

Table C.17 Simulation results: Komine Algorithm; US Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 172 | 356 | 536 | 716 |
| Number of Failed VPs | 32 | 67 | 99 | 132 |
| Failure Type | Single Node (7, 8, 9, 10, 11, 12, 14, 15, 16, 20, 21, 22, 23, 24, 25, 26) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 318 | 392 | 456 | 409 |
| • Search Messages | 253 | 253 | 253 | 253 |
| • Acknowledge Messages | 65 | 133 | 162 | 114 |
| • Cancel Messages | 0 | 6 | 41 | 42 |
| • Disconnect Messages | 0 | 0 | 0 | 0 |
| Restoration Time, msec | 124 | 206 | 209 | 174 |
| Weighted Restoration Probability | 0.70 | 0.67 | 0.42 | 0.18 |
| $RP_1$ | 0.71 | 0.61 | 0.39 | 0.10 |
| $RP_2$ | 0.68 | 0.71 | 0.35 | 0.18 |
| $RP_3$ | 0.68 | 0.71 | 0.51 | 0.25 |
| $RP_4$ | 0.73 | 0.71 | 0.59 | 0.34 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

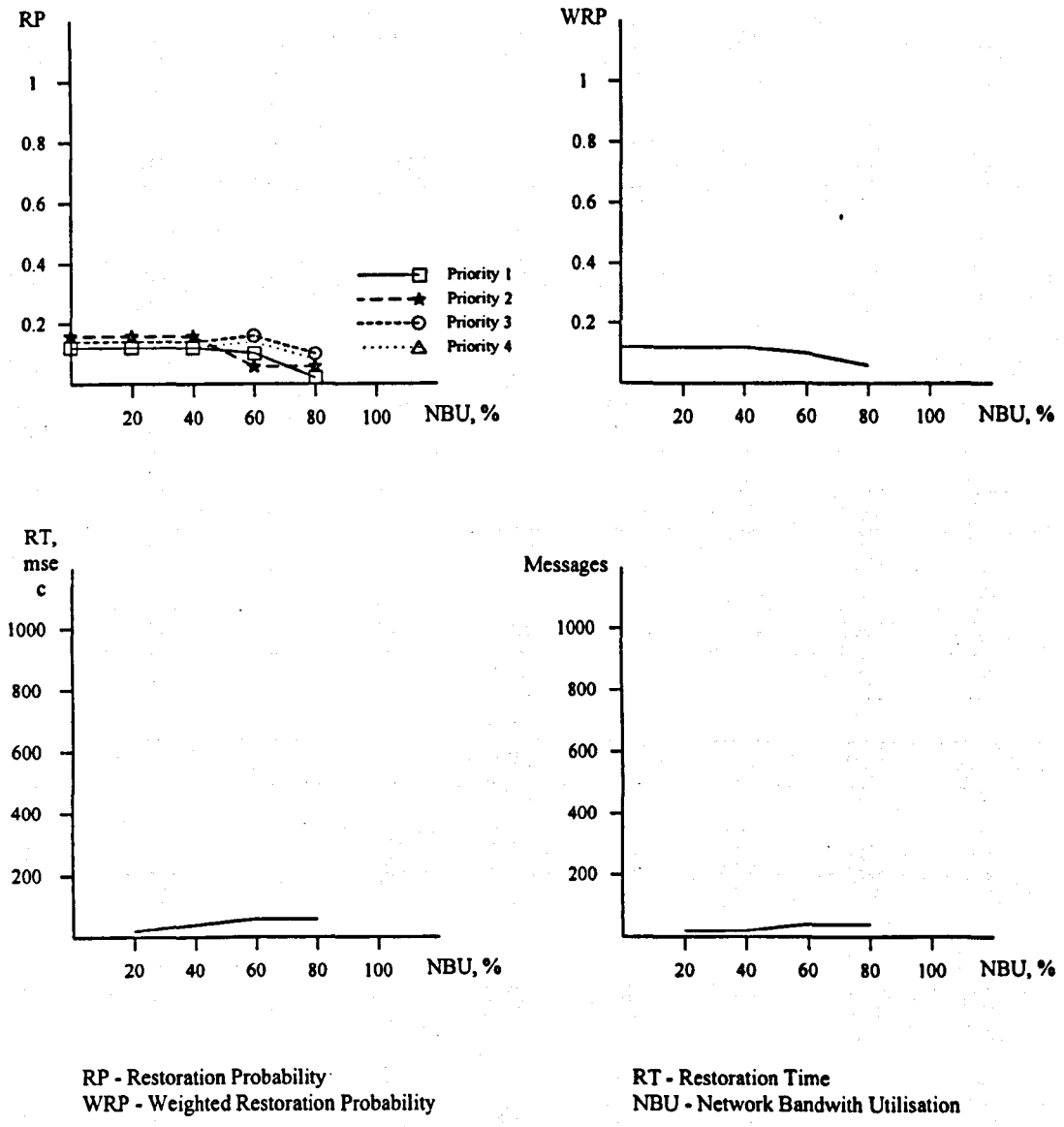
Figure C.17 Simulation results: Komine Algorithm; US Network.

### C.6.4 Simulation Results for Rudin Network, HLC = 2

Hop Limit Counter in this experiment was set to 2.

Table C.18 Simulation results: Komine Algorithm; Rudin Network; HLC=2.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 72 | 152 | 235 | 310 |
| Number of Failed VPs | 23 | 47 | 73 | 96 |
| Failure Type | Single Node (2, 3, 4, 5, 6, 7) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 21 | 27 | 31 | 29 |
| • Search Messages | 16 | 16 | 16 | 16 |
| • Acknowledge Messages | 5 | 11 | 15 | 13 |
| • Cancel Messages | 0 | 0 | 0 | 0 |
| • Disconnect Messages | 0 | 0 | 0 | 0 |
| Restoration Time, msec | 23 | 41 | 52 | 49 |
| Weighted Restoration Probability | 0.12 | 0.11 | 0.10 | 0.06 |
| $RP_1$ | 0.12 | 0.12 | 0.10 | 0.03 |
| $RP_2$ | 0.14 | 0.14 | 0.06 | 0.05 |
| $RP_3$ | 0.09 | 0.07 | 0.13 | 0.10 |
| $RP_4$ | 0.10 | 0.10 | 0.14 | 0.11 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

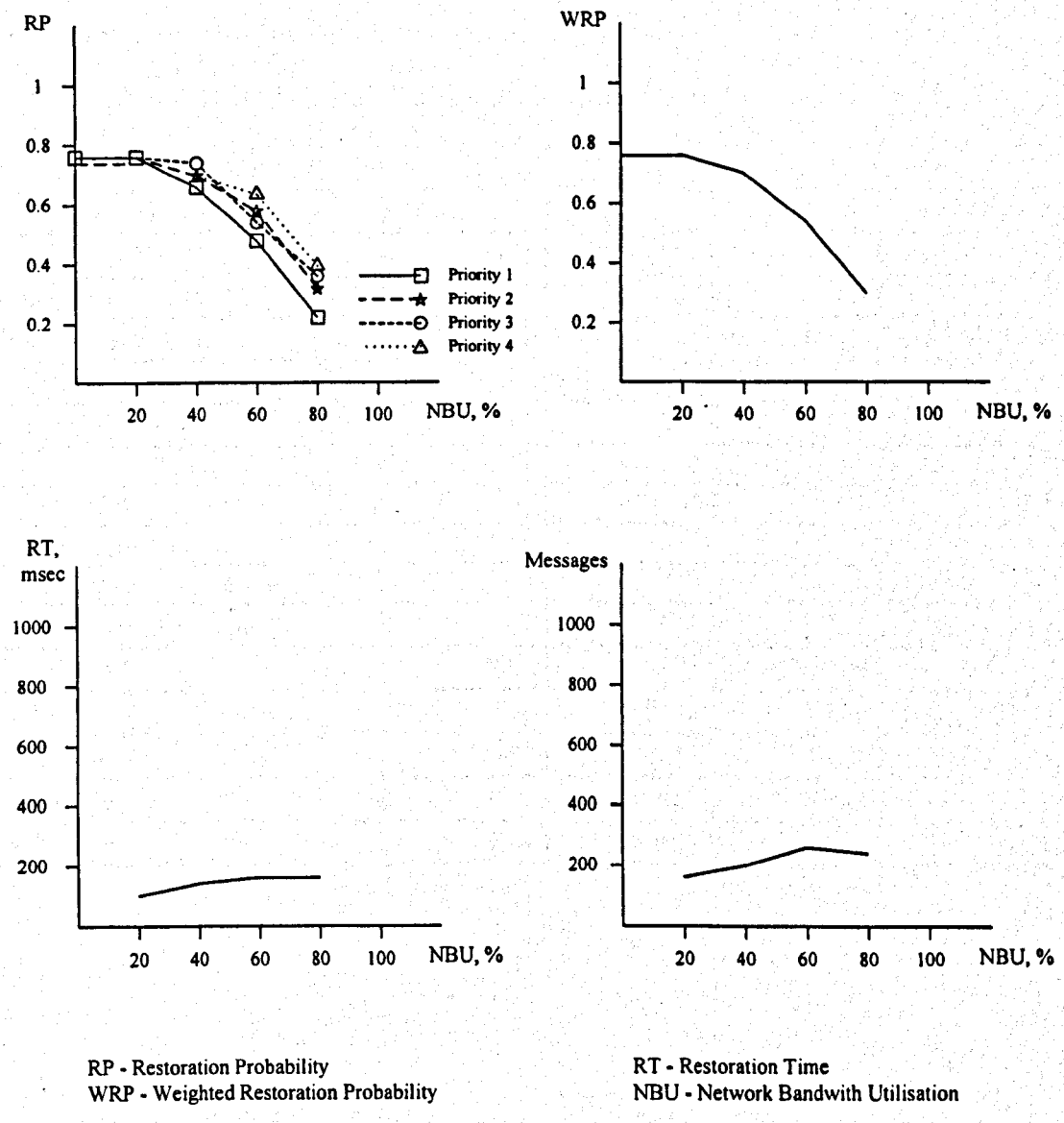
Figure C.18 Simulation results: Komine Algorithm; Rudin Network; HLC=2.

### C.6.5 Simulation Results for LATA Network, HLC = 3

Hop Limit Counter in this experiment was set to 3.

Table C.19 Simulation results: Komine Algorithm; LATA Network; HLC=3.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 132 | 268 | 409 | 549 |
| Number of Failed VPs | 27 | 53 | 80 | 106 |
| Failure Type | Single Node (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 159 | 202 | 254 | 242 |
| • Search Messages | 119 | 119 | 119 | 119 |
| • Acknowledge Messages | 40 | 79 | 110 | 94 |
| • Cancel Messages | 0 | 4 | 25 | 29 |
| • Disconnect Messages | 0 | 0 | 0 | 0 |
| Restoration Time, msec | 93 | 148 | 178 | 172 |
| Weighted Restoration Probability | 0.76 | 0.69 | 0.54 | 0.30 |
| $RP_1$ | 0.76 | 0.65 | 0.48 | 0.22 |
| $RP_2$ | 0.76 | 0.67 | 0.58 | 0.33 |
| $RP_3$ | 0.77 | 0.74 | 0.55 | 0.38 |
| $RP_4$ | 0.74 | 0.76 | 0.64 | 0.40 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

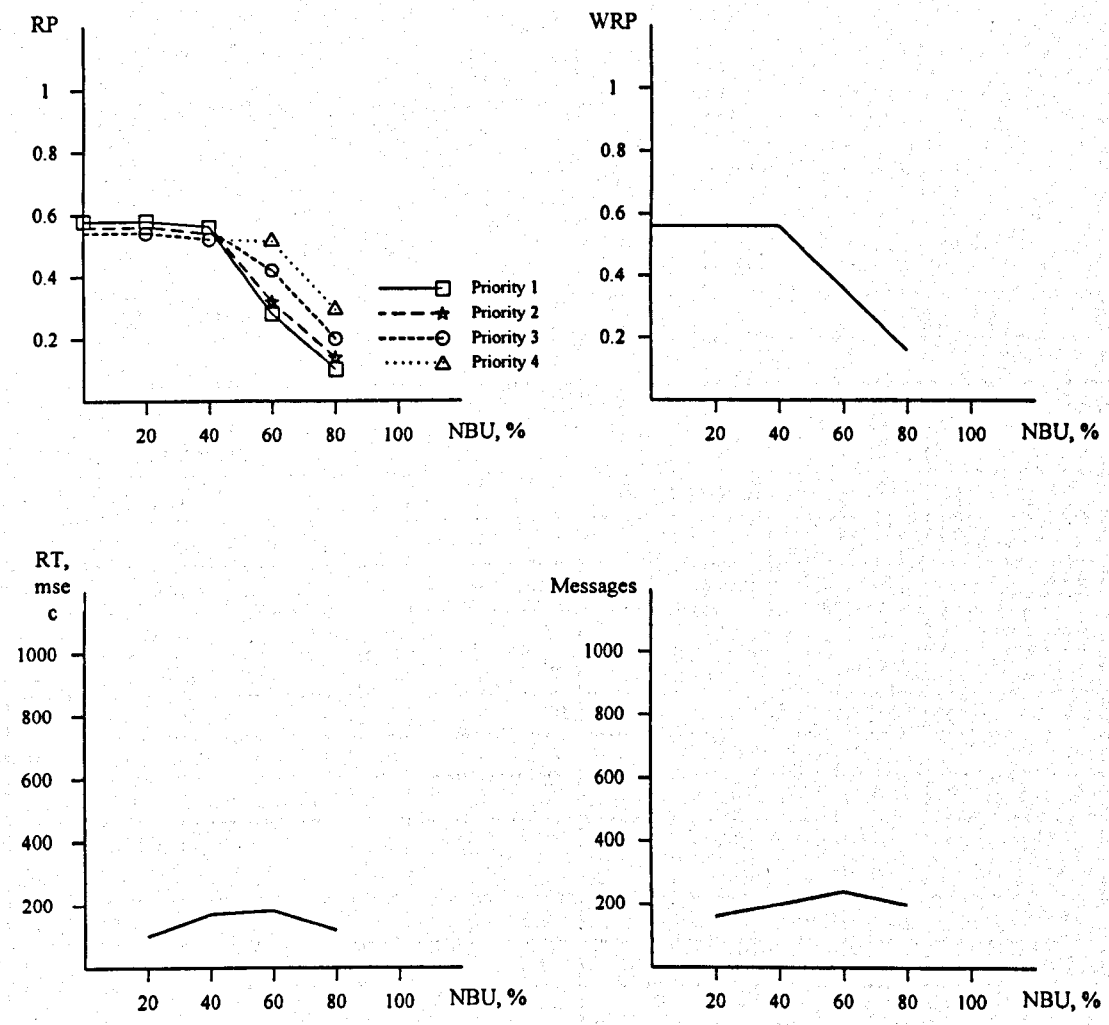
Figure C.19 Simulation results: Komine Algorithm; LATA Network; HLC=3.

### C.6.6 Simulation Results for US Network, HLC = 3

Hop Limit Counter in this experiment was set to 3.

Table C.20 Simulation results: Komine Algorithm; US Network; HLC=3.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 172 | 356 | 536 | 714 |
| Number of Failed VPs | 32 | 67 | 99 | 131 |
| Failure Type | Single Node (7, 8, 9, 10, 11, 12, 14, 15, 16, 20, 21, 22, 23, 24, 25, 26) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 153 | 206 | 235 | 196 |
| • Search Messages | 106 | 106 | 106 | 106 |
| • Acknowledge Messages | 47 | 97 | 111 | 73 |
| • Cancel Messages | 0 | 3 | 18 | 17 |
| • Disconnect Messages | 0 | 0 | 0 | 0 |
| Restoration Time, msec | 97 | 179 | 185 | 128 |
| Weighted Restoration Probability | 0.57 | 0.56 | 0.35 | 0.15 |
| $RP_1$ | 0.58 | 0.55 | 0.29 | 0.11 |
| $RP_2$ | 0.57 | 0.55 | 0.33 | 0.13 |
| $RP_3$ | 0.57 | 0.59 | 0.42 | 0.20 |
| $RP_4$ | 0.54 | 0.57 | 0.53 | 0.30 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

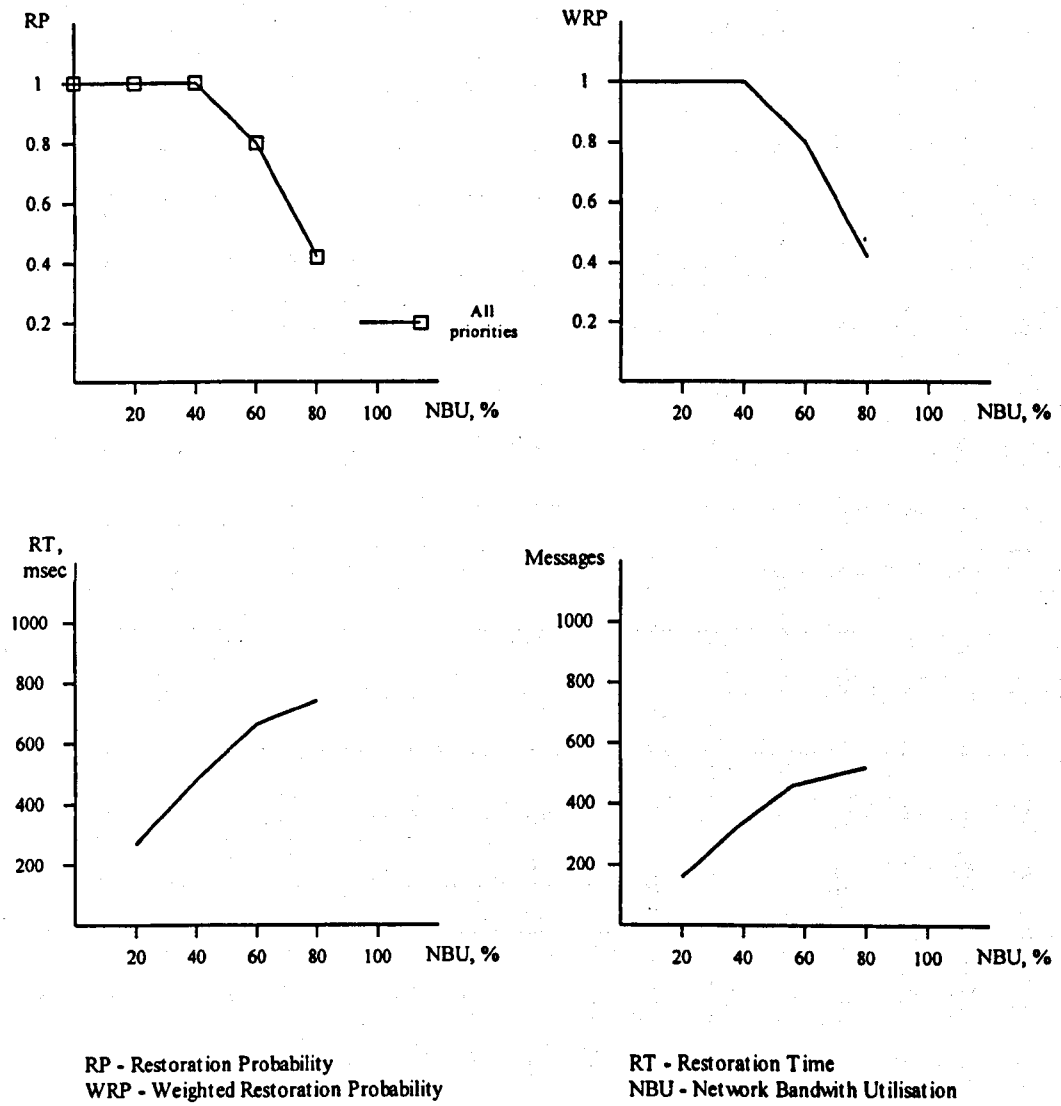
Figure C.20 Simulation results: Komine Algorithm; US Network; HLC=3.

## C.7 VPPS Algorithm

### C.7.1 Simulation Results for Rudin Network

Table C.21 Simulation results: VPPS Algorithm; Rudin Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 72 | 152 | 232 | 306 |
| Number of Failed VPs | 23 | 47 | 72 | 95 |
| Failure Type | Single Node (2, 3, 4, 5, 6, 7) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 165 | 343 | 491 | 535 |
| • Search Messages | 89 | 185 | 279 | 360 |
| • Acknowledge Messages | 76 | 158 | 199 | 151 |
| • Cancel Messages | 0 | 0 | 13 | 24 |
| • Disconnect Messages | 0 | 0 | 0 | 0 |
| Restoration Time, msec | 250 | 481 | 677 | 750 |
| Weighted Restoration Probability | 1.00 | 1.00 | 0.80 | 0.43 |
| $RP_1$ | 1.00 | 1.00 | 0.80 | 0.44 |
| $RP_2$ | 1.00 | 1.00 | 0.83 | 0.42 |
| $RP_3$ | 1.00 | 1.00 | 0.77 | 0.44 |
| $RP_4$ | 1.00 | 1.00 | 0.78 | 0.39 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

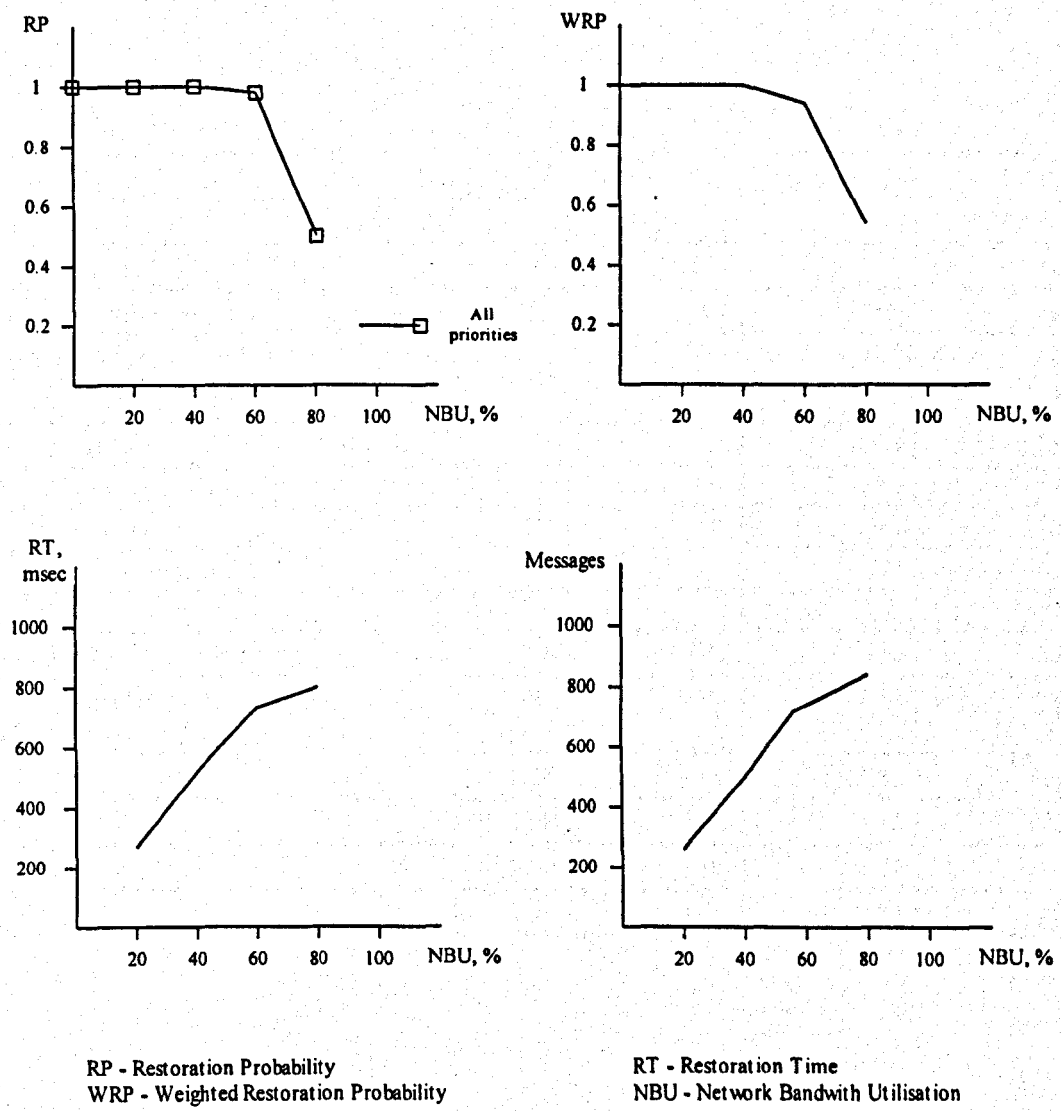
RT - Restoration Time
NBU - Network Bandwith Utilisation

Figure C.21 Simulation results: VPPS Algorithm; Rudin Network.

## C.7.2 Simulation Results for LATA Network

Table C.22 Simulation results: VPPS Algorithm; LATA Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 132 | 268 | 410 | 545 |
| Number of Failed VPs | 27 | 53 | 80 | 105 |
| Failure Type | Single Node (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 253 | 503 | 737 | 846 |
| • Search Messages | 140 | 278 | 410 | 517 |
| • Acknowledge Messages | 113 | 225 | 318 | 280 |
| • Cancel Messages | 0 | 0 | 9 | 49 |
| • Disconnect Messages | 0 | 0 | 0 | 0 |
| Restoration Time, msec | 270 | 517 | 727 | 801 |
| Weighted Restoration Probability | 1.00 | 1.00 | 0.95 | 0.55 |
| $RP_1$ | 1.00 | 1.00 | 0.96 | 0.55 |
| $RP_2$ | 1.00 | 1.00 | 0.95 | 0.55 |
| $RP_3$ | 1.00 | 1.00 | 0.92 | 0.57 |
| $RP_4$ | 1.00 | 1.00 | 0.92 | 0.55 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

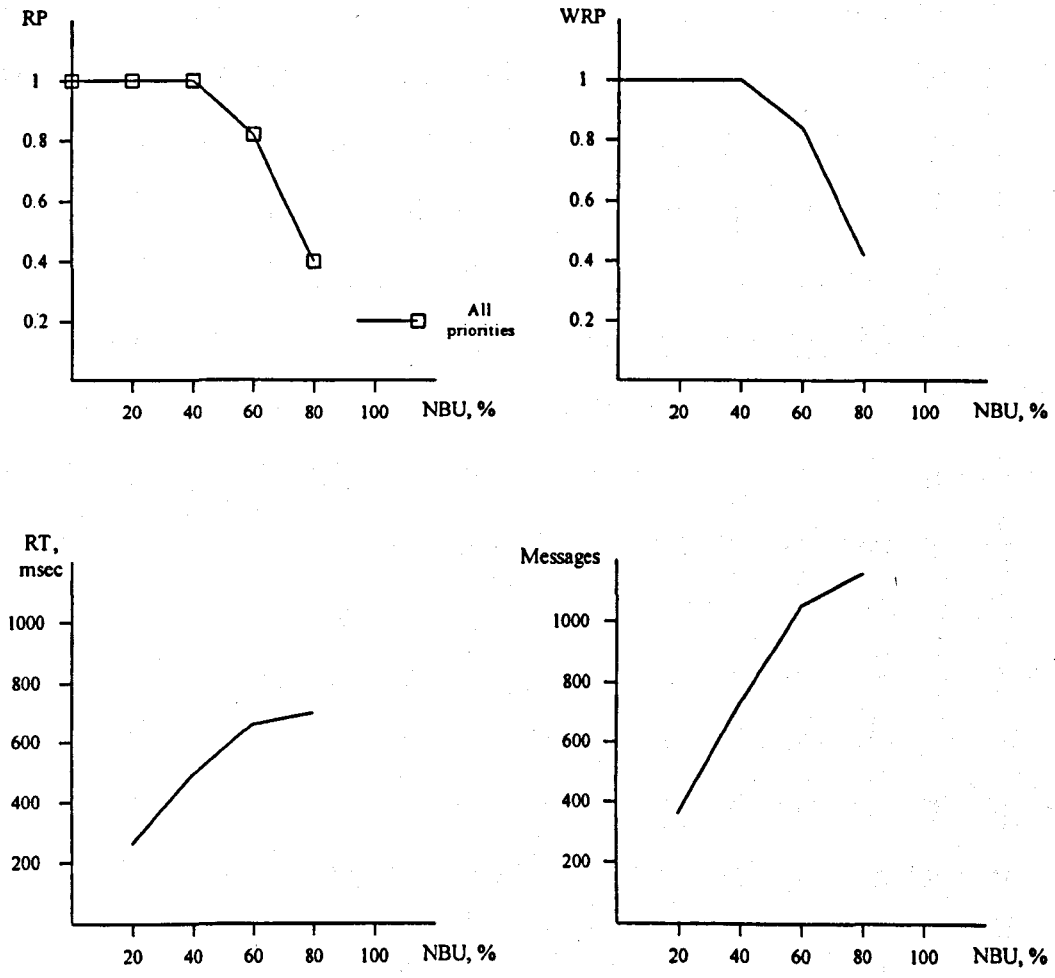
RT - Restoration Time
NBU - Network Bandwith Utilisation

Figure C.22 Simulation results: VPPS Algorithm; LATA Network.

### C.7.3 Simulation Results for US Network

Table C.23 Simulation results: VPPS Algorithm; US Network.

| Simulation Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| NBU, % | 20 | 40 | 60 | 80 |
| Number of VPs | 172 | 356 | 536 | 712 |
| Number of Failed VPs | 32 | 67 | 99 | 131 |
| Failure Type | Single Node (7, 8, 9, 10, 11, 12, 14, 15, 16, 20, 21, 22, 23, 24, 25, 26) | | | |
| **Results** | **1** | **2** | **3** | **4** |
| Messages Sent | 365 | 734 | 1051 | 1162 |
| • Search Messages | 191 | 386 | 566 | 712 |
| • Acknowledge Messages | 174 | 348 | 451 | 350 |
| • Cancel Messages | 0 | 0 | 34 | 100 |
| • Disconnect Messages | 0 | 0 | 0 | 0 |
| Restoration Time, msec | 258 | 492 | 663 | 685 |
| Weighted Restoration Probability | 1.00 | 1.00 | 0.83 | 0.42 |
| $RP_1$ | 1.00 | 1.00 | 0.83 | 0.42 |
| $RP_2$ | 1.00 | 1.00 | 0.82 | 0.44 |
| $RP_3$ | 1.00 | 1.00 | 0.84 | 0.40 |
| $RP_4$ | 1.00 | 1.00 | 0.84 | 0.40 |




RP - Restoration Probability
WRP - Weighted Restoration Probability

RT - Restoration Time
NBU - Network Bandwith Utilisation

Figure C.23 Simulation results: VPPS Algorithm; US Network.

# Appendix D DERA Correspondence

From Andrew Wood, CIS 1 Tactical Communications Division

St Andrews Road
MALVERN
Worcs, WR14 3PS, UK
Tel: (01684) 894494 Fax:
(01684) 895646

Our Ref: DRA/CIS(CISI)/P132/F/04/03
Your Ref:

27 October, 1997

To: Dr. Mike Morse/Dr. Amelia Platt
School of computing sciences De
Montfort University The Gateway
LEICESTER LEI 9BH

**"Distributed failure restoration..." by Alexander Zavjalov**

Dear Mike/Amelia,

Thank you for sending me Alexander's transfer report. I have read it and, quite frankly, am impressed. It is a competent and complete item of work which certainly gives me all the right impressions that things are going okay. In particular, it seems that Alexander and yourselves have a firm grip of what is needed in a tactical network.

However, this doesn't mean that I have no comments to make! But there are not very many.

The most important topic I would like to bring up in relation to the work is how does it relate to PNNI? I appreciate that PNNI came into existence after the work was started - or at least the first published standard came into existence - but I am worried that PNNI may subsume many of the ideas that Alexander has written about. For instance, PNNI is explicitly set up to broadcast topology changes whenever something 'significant' happens to links. In this scenario, I guess a link going down is significant! Also, PNNI will 'scale' (if that is the right word) to handle node failures.

Another feature of PNNI is its hierarchical addressing and the way it is used by end nodes to decide how to route any particular call. That is, as long as PNNI works, at all times every end node should have access to a complete and up-to-date routeing table. Often this table will be 'abstract' in the sense that it will not detail every single node between point A and point B. Instead the table may describe intermediate nodes in only a high-level manner. For instance, one may envisage that to send email from Malvern (point A) to Singapore (point B), the route is via Germany but there is no description of what precise nodes (towns) in Germany to go through. Therefore, getting back to Alexander's work, PNNI will handle link or node failures in Germany transparently to the end systems.

# References


1. Stallings W. "ISDN and Broadband ISDN with Frame Relay and ATM", Prentice-Hall, USA, 1995.
2. Onvural R. "Asynchronous Transfer Mode Networks: Performance Issues", Artech House, USA, 1994.
3. Lee B.G. "Broadband Telecommunications Technology", Artech House, USA, 1993.
4. Camm D., Sharp B., and Pardoe P. "Wireless ATM in the Tactical Arena - A solution for next generation military communications?" EuroMilcomp'96, Nice, France, 4-5 November, 1996.
5. Van Waveren C.J., Luiif H.A.M., Burakowsky W., Kopertowsky Z. "ATM Tactical Network – a Challenge for the Military Networks", WKTiI-97 (Military Conference on Telecommunications and Informatics), part 1, pp. 349-357, October, 1997.
6. Wu T-H. "Fiber Network Service Survivability", Artech House, USA, 1992.
7. Anderson J., Doshi B., Dravida S., Harshavardhana P. "Fast Restoration of ATM Networks", IEEE JASC, vol. 12, pp. 128-138, January 1994.
8. ATM Forum 'Private Network-Network Interface Specification', Version 1.1, 2002.
9. Goralsky W. "Introduction to ATM Networking", McGraw-Hill Inc., 1995.
10. Veitch, Smith D.G., Hawker I. "Restoration Strategies for Future Networks", Electronics&Communication Engineering Journal, pp. 97-103, June 1995.
11. Deacon A. G. "Network Reliability Better Than Ever," AT&T Technology, vol. 7, no. 4, pp. 20-23, 1994.
12. Chao C.-W., Fuoco G., and Kropfl D. "Faster Platform Gives the Network A Competitive Edge," AT&T Technology J., pp. 69–81, July/August 1994.

13. Nishihata. K. "SDH-based 52Mb/s Digital Cross-connect Systems with hitless function, TMN-based operation system," Proc. IEEE DCS Work-shop VI, Banff, Canada, June 1995.

14. Kawamura R. "Architectures For ATM Network Survivability", IEEE Communications Surveys, http://www.comsoc.org/pubs/surveys, vol. 1, no. 1, 1998.

15. Sosnosky J. "Service Applications for SONET DCS Distributed Restoration," IEEE JSAC, vol. 12, no. 1, pp. 59-68, 1994.

16. ITU-T Rec. G.783, "Characteristics of Synchronous Digital Hierarchy (SDH) Equipment Functional Blocks," 1996.

17. ITU-T Rec. G.841, "Types and Characteristics of SDH Network Protection Architectures," 1995.

18. Kajiyama Y., Tokura N., Kikuchi K., "An ATM VP-based Self-Healing Ring", IEEE JSAC, vol. 12, pp. 171-177, January 1994.

19. May K., Semal P., Du Y., Herrmann C., "A Fast Restoration System for ATM-Ring-Based LANs", IEEE Communications Magazine, pp. 90-98, September 1995.

20. Grover W., 'The self-healing network: a fast distributed restoration technique for networks using digital cross-connect machines', Proc. GLOBECOM'87, vol. 2, pp. 1090-1095, 1987.

21. Yang C., Hasegava S., 'FITNESS: Failure Immunization Technology for Network Service Survivability', Proc. GLOBECOM'88, pp. 47.3.1-47.3.6, December, 1988.

22. Chow C.-H., McCaughey S., and Syed S., 'RREACT: a distributed protocol for rapid restoration of active communication trunks', Proc. 2nd Network Management and Control Workshop, pp. 391-406, 1993.

23. Chow C.-H., Bicknell J., McCaughey S., and Syed S., 'A fast distributed network restoration algorithm', Proc. Int'l Phoenix Conf. on Computer and Communications, March 22-26, 1993, Tempe, Arizona, pp. 261-267.

24. Fujii H., Yoshikai N., 'Restoration Message Transfer Mechanism and Restoration Characteristics of Double-Search Self-Healing ATM Network', IEEE JSAC, vol. 12, pp. 149-157, January 1994.

25. Chow, C.E.; Bicknell, J.D.; McCaughey, S.; Syed, S., 'Performance analysis of fast distributed link restoration algorithms', International Journal of Communication Systems, vol: 8, iss. 5, pp. 325-345, September/October 1995.

26. Komine H., Chujo T., Ogura, Miyazaki, and Soejima, 'A distributed restoration algorithm for multi-link and node failures of transport networks', Proc. IEEE GLOBECOM'90, San Diego, CA, pp. 403.4.1-403.4.5, December 1990.

27. Ayanoglu E., Gitlin R., 'Broadband Network Restoration', IEEE Communications Magazine, pp. 110-119, July 1996.

28. Kawamura, I. Tokizawa, 'Self-healing Virtual Path Architecture in ATM Networks', IEEE Communications Magazine, pp. 72-79, September 1995.

29. Veitch, D.G. Smith, I. Hawker, 'Administration of Restorable Virtual Path Mesh Networks', IEEE Communications Magazine, pp. 96-101, December 1996.

30. Jones, K., and Henry, R., 'A fast ATM Rerouting Algorithm for Networks with Unreliable Links', Proc. IEEE ICC'94, New Orleans, LA, vol. 1, pp. 91-95, May 1994.

31. Wipusitwarakun K., Tode H., Ikeda H., "Fast Failure Restoration Algorithm with Reduced Messages Based on Flooding Mechanism", IEICE Transactions on Communications, vol. E80-B, pp. 564-572, April 1997.

32. Kubota F., Egawa T., Saito H, etc., "QoS Restoration that Maintains Minimum QoS Requirements – A New approach for Failure Restoration", IEICE Transactions on Communications, vol. E83-B, pp. 2626-2634, December 2000.

33. Kawamura R., "A Failure-Resistant Self-Healing Scheme in ATM Networks", IEICE Transactions on Communications, vol. E81-B, pp. 699-705, April 1998.

34. Nam Seok Ko; Dong Yong Kwak; Yool Kwon; Hong Shik Park, "Hybrid self-healing mechanism with VP priority and dynamic bandwidth assignment policy in ATM networks", Proc. of IEEE TENCON 99, USA; vol. 2, pp. 813-816, 1999.

35. Yahara T., Kawamura R., "New self-healing scheme that realizes multiple reliability on ATM networks", IEICE Transactions on Communications, vol. E83-B, pp. 2615-2625, December 2000.

36. Venables, B., Grover, W., and MacGregor, M., 'Two strategies for spare capacity placement in mesh restorable networks', Proc. ICC'93, pp. 267-271.

37. Landegem T.V., Vankwikelberge P., Vanderstraeten H. 'A Self-Healing ATM Network Based on Multilink Principles', IEEE JSAC, vol. 12, pp. 139-148, January 1994.

38. Dunn D., Grover W., MacGregor M., 'Comparison of k-shortest Paths and Maximum Flow Routing for Network Facility Restoration', IEEE JSAC, vol. 12, pp. 88-99, January 1994.

39. Rudin H.; Muller H. 'On routing and flow control', Flow Control in Computer Networks, Amsterdam, Netherlands, pp. 241-255, 1979.

40. Rudin H.; Mueller H. 'Dynamic Routing and Flow Control', IEEE-Transactions on Communications, vol. COM-28, no.7; pp. 1030-1039, July 1980.

41. Kawamura R., Sato K., Tokizawa I. 'Self-healing ATM Networks based on Virtual Path Concept', IEEE JSAC, vol. 12, pp. 120-127, January 1994.

42. Greca A., Nakagawa K. 'A Coordination Based Restoring Algorithm for High Speed Broadband Networks', IEICE Transactions on Communications, pp. 1517-1526, July 2000.

43. Wang Y.-F., Huang J.-F. 'Preplanned Restoration and Optimal Capacity Placement on ATM Multicast Tree', IEICE Transactions on Communications, pp. 281-292, February 2000.

44. Saito H., Slominski M., Yoshida M., 'An improved guided restoration algorithm for ATM crossconnect networks', IEEE Network Operations and Management Symposium, New York, USA, vol. 1, pp. 225-234, 1996.

45. Hadama, H.; Kawamura, R.; Sato, K., 'Virtual path restoration techniques based on centralized control functions', Electronics and Communications in Japan, Vol. 78, Iss: 3, pp. 115-123, March 1995.

46. Sakauchi, H., Nishimura, and Hasegava, 'A self-healing network with an economical spare-channel assignment', Proc. IEEE GLOBECOM'90, San Diego, CA, pp. 403.1.1-403.1.6, December 1990.

47. Lee D.-H., Park J.T., Lee K.H., Woo W.-D. 'A Hybrid Escalation Mechanism for the Efficient Restoration of ATM Networks', Computers ind. Engineering, vol. 35, pp. 279-282, 1998.

48. Xiong Y., Mason L., 'Restoration Strategies and Spare Capacity Requirements in Self-Healing ATM Networks', IEEE/ACM Transactions on Networking, vol. 7, pp. 98-110, February 1999.

49. S.Hasegawa, Y.Okanoue, T.Egawa, and H.Sakauchi, "Control Algorithms of SONET Integrated Self-Healing Networks" IEEE JSAC, vol. 12, pp. 110-119, January 1994.

50. Panicker V., Siva Ram Murthy, Mittal R. 'An Improved Scheme for Self-Healing in ATM Networks', Computer Communications, vol. 22, pp. 1400-1414, 1999.

51. Gersht A., Shulman A. 'Architecture for Restorable Call Allocation and Fast VP Restoration in Mesh ATM Networks', IEEE Transactions on Communications, vol. 47, vol. 47, pp. 397-403, March 1999.

52. Nace D., Carlier J., 'Distributed rerouting in DCS mesh networks", Proc. Combinatorics and Computer Science, Berlin, Germany, pp. 406-415, 1996.

53. Ashraf S., Lac C, 'A novel rerouting algorithm for VP restoration', Second International Symposium on Communication Systems Networks and Digital Signal Processing 2000, Bournemouth Univ., Poole, UK, pp. 267-271.

54. CCITT I.321 'B-ISDN protocol reference model and its application', 1990.

55. ITU-T Recommendation I.610, 'B-ISDN operation and maintenance principles and functions', 1992.

56. Azuma R., Fujii Y., Sato Y., Chujo T., Murakami K., 'Network Restoration Algorithm for Multimedia Communication Services and Its Performance Characteristics', IEICE Transactions on Communications, vol. E78-B, pp. 987-994, July 1995.

57. Peyravian M., Kshemkalyani A., "Connection Preemption: Issues, Algorithms, and a Simulation Study", Proc. IEEE INFOCOM '97, vol. 1, pp. 143-151. Los Alamitos, CA, USA, 1997.

58. Barfoot R., Camm D., Daniell J., Thorlby P., 'Mapping Commercial ATM to the Tactical Radio Environment', MILCOM-98, vol. 3, pp. 1055-1059, Boston, USA, October 1998.

59. Johnson T.D., Nourry G.R., Rahman M.H., 'Connection Level Priority/Pre-Emption Service for ATM Based Defence Networks', CCECE-'97, vol. 2, pp. 590-594, Canada, May 1997.

60. Sass P., Gorr L., 'Communications for the Digitized Battlefield of the 21st Century', IEEE Communications Magazine, pp. 86-95, October 1995.

61. Sivabalan M., Mouftah H.T., 'QUARTS-II: a Routing Simulator for ATM Networks', IEEE Communication Magazibe, pp. 80-87, May 1998.

62. Kawamura R., Ohta H.. 'Architectures for ATM network survivability and their field deployment', IEEE Communications Magazine, pp. 88-94, August 1999.

63. Bajaj V.B., Sarje A.K., 'A comparative study of two self healing protocols for ATM networks', International Journal of High Speed Computing, vol.10, no.3, p.235-255, September 1999.

64. Holzmann G.J., 'Design and Validation of Computer Protocols', Prentice-Hall, USA, 1991.

65. Yahara T., Kawamura B., Ohta S., 'New Self-Healing Scheme that Realizes Differentiated Bandwidth Requirement on ATM Networks', IEICE-Transactions on Communications, vol. E83-B, no.3, pp. 672-679, March 2000.

66. T'Joens Y., Georgatsos P., Griffin D., Huth P.T., Georgiades L., Pavlou G., Manikis D., Mykoniati E., 'An integrated approach to switched VC ATM restoration in the REFORM system', Design of Reliable Communication Networks (DRCN 2000), pp. 160-165, Munchen, Germany, April 2000.

67. MacDougall, M.H. 'Simulating Computer Systems Techniques and Tools'. The MIT Press 1987.