Management of Lightpaths in Optical WDM Networks:

A Tool for Configuration and Fault Management

Jonas Söderqvist

Master thesis performed at

Department of Teleinformatics, Royal Institute of Technology
Examiner and supervisor: Prof. Gunnar Karlsson

And

Optical Network Research Laboratory, Ericsson Telecom AB
Supervisors: Tanja Kauppinen and PhD Erland Almström

In collaboration with

Telia Research AB
Technical support: Sten Hubendick

Stockholm 2000

Abstract

In this master thesis I analyse and implement a tool for configuration and fault management of optical connections in an optical transport network. Network operators or equipment such as IP-routers can configure optical crossconnects to establish point-to-point lightpaths from one edge of the network to another. A lightpath is a series of optical WDM channels that are interconnected. Functions in the control plane can protect connections from channel, fibre or node faults. The nodes detect and locate faults and alert other nodes that automatically perform protection switching of crossconnects to pre-configured settings.

The configuration tool has been tested in simulation and in a real metropolitan area test network. Distributed fault detection and restoration routines improve the performance compared to centralised fault processing. The difference is probably quite small though. The tool has been developed in steps from a centralised processing approach towards almost fully distributed processing. Optimisations of the program code such as faster functions, smaller node messages, asynchronous message transmission etc. were added to each implementation step. Hence, the last version which is the most distributed is also the most optimised and a performance comparison of centralised versus distributed would be unfair. For example, the distributed restoration routine triggered by fault detection takes about 60ms both when the fault is in a WDM channel with one protected connection and in a fibre with four protected connections. The centralised approach needs more than 100ms to restore a fibre fault. The value could be far less with optimisation.

Contents

1 Introduction

1.1 Purpose and goal of this master thesis

The work presented in this master thesis first of all demonstrates the possibility to set up optical paths for instance between IP routers by dynamic configurations of optical crossconnects (OXC) [1]. OXCs can be used in wavelength division multiplexed (WDM) networks for interconnection of optical fibres and wavelength channels [1]. WDM is basically the same as frequency division multiplexing (FDM) used in radio systems. Especially if compared to amplitude modulated radio, WDM can be seen as a mapping of FDM into the optical domain. The demonstration tool developed in this project is intended for any optical transport network (OTN) for control of OXC nodes that are capable to dynamically connect any input wavelength to any output wavelength. The tool controls solely the OXCs; all other equipment is unknown to the tool. It is especially constructed for use in a metropolitan area test network, Winchester, which is being installed by Ericsson Telecom AB and Telia Research AB [2][3].

Further, this report is a study of distributed and automatic management functions that the demonstration tool can use or is using. Also, analyses for integration of the tool and user equipment is presented. The integration means that for example IP routers connected to OXC nodes should be able to benefit from the use of optical management functionality. In this report, the user or user equipment is someone or something that utilises services provided by the OTN.

This report can be used for a foundation of a dynamic and automatic management system for a cross-connected WDM network. It presents difficulties and necessities for configuration and fault management of optical connections, and discusses distributed configuration and protection. It is based on an earlier master thesis [4] that should be read for more information about the OXCs and their control.

1.2 Scope of this report

- Section 2 reviews optical network technology, corresponding management approaches and difficulties that arise for distributed management functions. The management system specified in open systems interconnection (OSI) is briefly described. Reconfiguration and protection issues are discussed. Two current management approaches are discussed: multi protocol label switching (MPLS), that uses wavelengths as labels, and optical domain service interconnect (ODSI).
- Section 3 gives the project’s specification, a list of the work performed and the equipment that has been used.
- Section 4 presents the solution for configuration and fault management of optical connections in WDM networks with OXCs. Subsections deal with program components and structure of distributed and central management nodes, communication between management nodes through a message passing system, programming language, both shared and specific functions, network operation and solution details.
- Section 5 discusses simulation and real network testing of the solution.
- Section 6 gives conclusions.
- Section 7 holds a list of abbreviations and denotations.
- Section 8 holds a list of references.

2 Theory and related work

Today, network providers use fibres for example in local area networks (LAN) and especially for long-haul transmission such as across the Atlantic Ocean. One benefit of fibre compared to traditional copper wire is the vast amount of data it can carry. Further, with WDM multiple optical wavelengths can be used simultaneously in a fibre, which helps when trying to utilise as much of the fibre capacity as possible. Each wavelength provides a separate channel to transport data [1].

It is possible for network providers to share for instance one WDM fibre by allocating different channels for their own usage. Equipment such as OXCs or optical add drop multiplexers (OADM) can be used to connect fibres and to send and receive data on channels. Some OXCs can even connect channels between fibres. These kinds of networks make it possible for dynamic reconfiguration of the optical network topology without any physical re-wiring of fibres. A network provider can connect nodes through configuration of one or more OXCs and then lease the connection to a third party as an optical point-to-point link. See Figure 1 for an illustration of a possible network that uses OXCs or OADMs for connection of wavelength channels.

Real-time applications need fast backup for network faults that affect them. In a WDM network that uses OXCs for dynamic configuration, protection against failing network elements can be provided at the fibre layer. Instead of re-routing in higher layers (e.g., in the operating system or the application itself) the OXCs or some management system can re-configure the network topology to avoid using the network element that has failed. Lost data has to be re-sent though, either by buffering in the WDM network or by retransmission at higher layers.

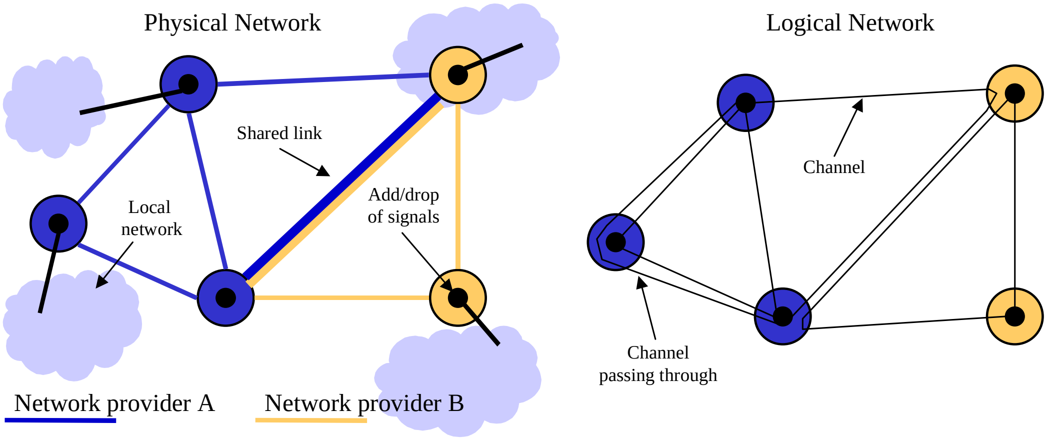


Figure 1: Example of an OTN. The network includes five WDM nodes physically connected according to the left figure. Two network providers share one link or fibre for additional connectivity. With OXCs or OADMs at three of the nodes wavelength channels can be configured to pass through those nodes. A resulting logical network is illustrated to the right.

2.1 Optical networks

Optical fibre provides transmission of data with capacity much greater than conventional copper wire. Data is packaged in frames before transmitted over fibres. A transmission format specifies how frames are sent.

Time division multiplexing (TDM) is a technique that allows multiple electrical signals to be transmitted together. TDM multiplexes signals from multiple sources as timeslots in a high capacity transmission signal. At the destination of the transmission, the opposite (de-multiplexing) is performed. When the high capacity signal is de-multiplexed, the content in each timeslot is sent to its destination [5].

The performance bottleneck for optical transmission is the equipment connected to a fibre that in the electrical domain handles the transported data. The equipment is not fast enough to utilise the whole data capacity that a fibre can carry. TDM allows multiple low capacity channels on one high capacity channel, but does not solve the bottleneck problem. Instead, WDM is one solution that in the same manner as TDM multiplexes multiple signals, but with the use of separate wavelengths instead of timeslots as signal carriers. The spectrum of an optical fibre that has good enough characteristics for transmission is divided into a number of wavelength channels. A channel count of 32 is a reality today, and 160 channels is a research area. The data capacity in a single channel can for example be 2.5Gb/s or 10Gb/s; development towards a capacity of 40Gb/s is in progress.

Various pieces of equipment exist for use in WDM networks. OADMs are used when WDM channels must be added or dropped from fibres without causing interference in the transit channels. OXCs basically connect wavelength channels from one fibre to another. An OXC can either be static or dynamic. Static means that the cross connections are locked in place and may be impossible to change. The dynamic case allows reconfiguration of the OXC. Additionally, wavelength conversion may be possible. The conversion means that an input wavelength channel does not necessarily have to be connected with the same wavelength in the output fibre. Full wavelength conversion means that an input channel can be connected to any output channel.

WDM does not specify what is transmitted in the channels. Different data formats can simultaneously be transmitted in separate channels. For example synchronous optical network (SONET) and synchronous digital hierarchy (SDH) can be used for transmission directly on some optical channels, while other channels in the same fibre are used for completely different transmission formats. SONET and SDH are TDM techniques that can be used for encapsulation of packet switching with the Internet protocol (IP) or asynchronous transfer mode (ATM) [5][6]. SDH is the international equivalent of SONET, used in U.S. Together, SDH and SONET provide standards for interconnection of international digital networks that use optical media for transmission. ATM multiplexes data streams consisting of small cells (53 Bytes) which are directed by paths configured in ATM switches. ATM is intended for fast hardware switches. See [7] for more details on SONET/SDH and ATM.

2.1.1 Protection in optical networks

One very important issue for a network provider is to meet the customer demands. Large network capacity introduces extra traffic load in one part of the network if another part fails to deliver traffic. Lost traffic has to be re-sent as fast as possible. For example, a fibre that carries 160 WDM channels, each with a data capacity of 10Gb/s, may loose as much as 80Gb data during a time of 50ms, which is a general value for restoration[1] times. A well working protection is important to satisfy the customer. Note that the 80Gb example is a rather extreme case that lies in the future. However, several fibres are most likely bundled together in cables, which introduces risks of loosing even more than 80Gb data.

[1] Restoration is in this report considered to be an act of reconfiguring the network to use alternative, but already reserved, lightpaths when a fault causes ordinary lightpaths to fail. Protection is considered to be the whole mechanism to protect and perform restoration from faults.

To give network providers a good alternative to SDH/SONET systems, any new network with automatic protection should have restoration times of no more than 50ms. The switching fabric in a SDH/SONET link allows for fast automatic protection. The time to restore traffic is specified to complete within 50ms, if the right conditions are fulfilled [8].

If an optical connection between two nodes in an OTN without support for protection suffers from a break in one fibre, then lost data must be re-sent on alternative routes. The new routes can deliver the data without difficulty if there is enough capacity. However configuration of the new routes may take some time before routers are ready to forward the extra data. The traffic delay the customers have to cope with may be too long to meet customer demands. An automatic optical restoration that finishes within 50 ms from detection of a fault could make a significant difference. After restoration, traffic could continue to flow as before the break with only small losses of data.

2.2 Management of optical networks

An optical management system provides for instance functions for configuration, starting and stopping of optical equipment. Network users may need to be restricted or allowed to use provided services or network resources. Services have to run as specified and faults have to be dealt with before users are disturbed.

In a dynamically re-configurable optical network, there are at least two approaches when it comes down to where decisions of reconfigurations are made. Either an overlay model or peer model can be used. The overlay is a model with separate control planes for the optical transport layer and the higher layers, e.g. IP layer. The optical transport layer control plane controls the optical network either by itself or with information from the IP layer. Both additional network management system and client equipment, such as IP routers that signal to the optical domain, can initiate reconfigurations. The peer model is an integrated control architecture where the IP and optical layers share protocols in a tight co-operative manner. Similarly to the possibility in the overlay model, it is IP routers that request reconfigurations and connections between each other (peer to peer). See [2] and [9] for more details.

The layering technique admits a network implementation to be partitioned into small subtasks. Instead of implementing all network functionality in one single layer, each layer can have different responsibilities. A layer can be seen as a provider of services for the layer above. Services in a layer in turn make use of services provided in the layer below. Another way of separating the implementation into subtasks is to use distributed management. The management system may be improved by distribution of control tasks to areas as near managed equipment as possible.

2.2.1 Management functional areas

Much of the work that has been done for management of networks in the electrical domain can be applied to management in the optical domain as well. *Open systems interconnection* (OSI) *systems management* is a set of standards that divide management into five functional areas [10]: accounting, configuration, fault, performance and security. These areas are common in many network management systems, whether they are based on OSI systems management or not. A good approach towards an optical management system would be to build modular management applications that each concentrates on one of the areas. Some functions can be shared among the modules. In addition, all modules could use a common data structure, where network state and different management events can be stored and accessed.

The following subsections specify management tasks for the five functional areas in the OSI systems management and concentrates on optical networks.

Accounting

The accounting provides functions that calculate and charge customer usage of network resources. Customers can be informed of costs and charges. Other functions could be for leasing of resources, limiting customer usage of resources and defining parameters such as costs and limitations. For example, network providers that share an OTN could be individually charged for optical connections they set up.

Configuration

Configuration management control and monitor network resources' condition and performance. In an OTN that has WDM cross connection possibilities, connection management is a main area. Connections are set up and taken down. Resources are initialised or closed depending on their condition or the usage of them. For example a resource that is in bad shape need to be closed down for repair, or a portion of a network might receive an upgrade and must therefore be shut down. Other tasks could be configuration of management system, naming of resources, topology discovery and port connectivity.

Either a network user or the OTN itself could initiate configuration operations. Users can be of different sorts such as network operators, customers, or user equipment attached to the OTN. An operator can design the OTN to suit specific needs, or a customer can for example request multicast connections for distribution of video. The third type of user, for example IP routers, could request connections in the same way as a customer. The IP routers must have sufficient intelligence and information about traffic patterns (i.e., load and flows) to make good decisions [3][9]. This information could also be made available for the OTN to implement its own configuration intelligence. The OTN could then use it to reconfigure the network for optimisation purposes. For example, a connection that is taken down will free resources that other connections could benefit from.

Fault

Fault management must be able to detect, isolate and alert faults, restore connections and report faults and restoration outcome. Additional logging of faults should be done. Protection of connections must be configured. Examples of protection techniques are link protection and path protection. Link protection is done by a change of the path to links that circumvent a failing link or node. Path protection uses an alternative path, which means that the connection and protection paths do not share resources other than those in the end points of the connection. Common path protections are: 1+1 that transmits on two connections with one as backup and the other as working path, 1:1 that only reserves an additional path that is set up if a fault should occur and 1:N, in which many connections share the same reserved protection paths.

WDM equipment monitors values for power levels in optical signals. The WDM system sets off an alarm if a value passes some threshold. Optical transport management should receive such alarms to be able to protect connections.

Performance

Performance management should track and log connections, faults and user activities for later examination and diagnosis. The logs should be of a standard that can be used in commercial traffic engineering programs.

Another performance issue is checking of physical network equipment. For example, optical signal quality/degradation should be monitored in an optical network. Degradation of the signal quality could indicate problems with a laser or amplifier. Performance- and fault management are related, but performance management is more focused on issues that are not yet critical. For example if an amplifier is degrading, its signal quality falls under performance monitoring until a threshold is passed. It is then treated as a fault and fault management takes over. However, before it is treated as a fault, connections that use the degrading laser could be reconfigured.

Existing methods for observation of analogue optical signal quality are not good enough for monitoring of signal quality [11]. Access to the digital contents of an analogue signal is necessary for acceptable signal quality monitoring. In addition, the WDM system may monitor the optical signal degradation and could alert the OTN in time before the degradation is critical.

Automatic optimisations can be made if information about traffic can be requested from the user equipment. An example on information is thresholds for how much traffic a router has in its buffers. Too much would mean that there is a risk of data being lost and too little would not be efficient.

Security

The security management controls access to resources, handles authentication of network users and checks authorisation for users. Other security issues concern encryption, privacy and logging of user activities. Users of an OTN may be able to set up optical connections. Such connection requests must be verified to prevent unauthorised access to other connections or to groups that use connections to form private networks.

2.3 Distributed network management

The debate over distributed or central network management may be a never-ending story. In regard to a centralised system there are some things that are gained and others that are lost when creating a distributed management system. The things concern management traffic, complex dependencies, scalability and robustness. These concerns for distributed management are explained in the following list:

+ Less management traffic overhead. Much of the management functions are done by work on local network elements instead of only by remote access. For example, huge amounts of data that are input to a function may not need to be sent across the network to a central node (or other nodes). The data can instead be located at the node where the processing is taking place.

+ Greater scalability is achieved because both management processing capability and network elements can be added to the network when and wherever needed. A central management computer may be more difficult to upgrade without disturbing the network performance.

+ Robustness is possible by replication of information. One part that goes down due to a failure may not affect the rest of the network too negatively.

– Introduction of new parameters and dependencies may complicate correct behaviour of affected processes. Such changes are probably easier to adapt to by central processing.

– The overall management traffic may in fact increase if the information in the distributed protocols is not restricted. The exchange of information is done between management nodes. Information about voting, agreement, events, updates, etc. is necessary for distributed functions where the processing is done at numerous nodes.

– A central management system can be made robust as well. Backup solutions can take over to preserve the network’s state.

Overall, the question of whether a network management system benefits best from a centralised or a distributed approach is highly dependent of the implementation and the functionality it shall provide. A control plane for configuration and protection of an OTN must be robust, which can be achieved with both centralised and distributed. Protection must be fast, which may be possible only with distributed alert and restoration routines.

2.4 Current management approaches

The Internet engineering task force (IETF) started a working group in 1997 for standardisation of a management system, multi-protocol label switching (MPLS) [12]. In the beginning, it was a solution to run IP over ATM networks. Today, extensions of MPLS, which will enable direct control of OTN equipment, are proposed [9][13]. Other proposals for standardised optical network control are in progress. An example is optical domain service interconnect (ODSI), which is an interface for utilisation of OTN control planes [14].

The optical MPLS approach is an integration of optical layer management with IP packet switching layer into one single control plane. The ODSI is an overlay model that defines an interface between any user layer and the optical layer. While the ODSI interface is straightforward to add to an already existing optical management layer, the MPLS management is a complete system that probably has to be implemented from the beginning. Both optical MPLS and ODSI have in common that they reduce the number of layers necessary for IP over WDM. Some possible realisations of IP traffic over WDM are shown in Figure 2, e.g. ODSI to the left and optical MPLS to the right.

Optical MPLS directly over WDM may be more future proof than ODSI or other layering approaches [13]. Optical MPLS will suit future WDM equipment such as optical packet switches; plug them in and use existing MPLS system. Such new optical equipment with an ability to control packets can be configured by MPLS functions. However, much will probably change before those optical packet switches are deployed in networks. And still, a separate optical layer also has advantages and is wanted by network providers [15].

Both optical MPLS and ODSI allow the IP layer to requests services that the OTN provides. Optical MPLS goes one step further by its tight integration with IP. IP routers can use the traffic engineering capabilities of MPLS to tailor the OTN. Or, the OTN may be able to do that by itself. ODSI does not specify these features, but it is possible to implement them in the OTN that ODSI interfaces.

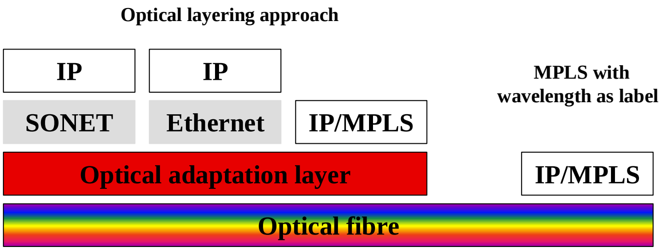


Figure 2: Example of two different approaches for management and utilisation of OTN based on WDM. In optically enabled MPLS the IP and MPLS interaction is mutual, while the optical adaptation layer is independent of the layer above.

2.4.1 MPLS today

MPLS is a technique that adds control of packet transport. It takes control of which paths packets will traverse the network by adding short labels to packets that enters an MPLS network. A label tagged to a packet at a node corresponds to a pre-configured label switched path (LSP). The path is called LSP because the label tagged to a packet is switched for a new label (may be the same or a different one) at every node before it is forwarded in the new label's direction. When a packet finally leaves the MPLS network and is sent towards its destination the label is removed and conventional IP routing proceeds.

The idea that started the work of MPLS was to solve problems with vendors' different "multi layer switches" [13]. The multi layer switches were independently developed with different approaches, all with the purpose of integrating IP with ATM. To make such switches interoperable, a standard made with regard to the different vendors' solutions was needed. MPLS uses a control-driven model for configuration of forwarding information, i.e., set-up of label switched paths (LSP) are made prior to packets' arrival at the edge nodes. The alternative, data-driven model used in some multi layer switches, would set up LSPs first when a stream of packets arrives. MPLS is also designed to run on any layer (e.g. ATM, SONET or WDM) in contrast to the preceding multi layer switches (only on ATM).

One goal with MPLS is to provide Internet service providers (ISP) means for traffic engineering, e.g., to control traffic routes. Conventional IP routers found in ISP networks forward a packet in the direction configured for the packet's destination address. An ISP may want to prevent congestion in a router that often suffers from packet losses due to full buffers. This is difficult with conventional IP routers that base the forwarding of IP packets only on the packet's destination address. The solution is MPLS that provides forwarding dependent of both destination and source addresses. See [12] for more information about this MPLS advantage. Additional advantages with MPLS as a standard compared to vendor dependent multi layer switches are faster forwarding because of removed vendor incompatibility issues and that one common management system is possible.

2.4.2 Optical MPLS

One motivation for the optical MPLS approach is based on the assumption and the fact that more network services are and will become IP based. Therefore, it may be best to provide a tightly integrated solution for management of WDM and IP equipment. Today, MPLS has powerful traffic engineering capabilities, which could be used for OTNs as well.

By identifying specific WDM requirements and applying extensions to the current MPLS standard, a management system for IP over WDM is taking form [9]. This optically extended MPLS can be applied directly on the WDM layer, see the part to the right in Figure 2. The idea is to combine IP routers and OXCs, both participating in an optical MPLS control plane. LSPs are set up in the optical domain by configuration of wavelength connections between OXCs. The LSPs are in fact lightpaths and the label swapping and forwarding performed in real time at each OXC is implicitly done in the switching fabric. The result is that labels correspond to wavelengths in lightpaths.

Not all of the functions possible in MPLS can be implemented in the OXCs. For example, OXCs cannot merge several wavelengths into one wavelength as opposed to merging of labels in MPLS. Another issue is the small number of wavelengths available in contrast to the large number of labels in MPLS. See [9] and [13] for more information on optical MPLS.

2.4.3 ODSI

The optical layering approach supports any type of higher network layer through an optical adaptation layer. This means that the optical layer is independent of the layer above and thus it can transport any traffic. The optical layer provides interfaces that let higher layers control and utilise WDM channels. The layer manages the optical equipment for provisioning of lightpaths, protection and recovery. Figure 2 illustrates some layers that may use the optical layer as an interface to the WDM system.

One instance that works for a fast standardisation process of an optical layer is ODSI [14]. It seeks co-operation among developers of equipment for the electrical domain (e.g. IP-routers and ATM switches) and the optical domain (e.g. OXCs). The goal is to produce standards for switches and routers provided with advanced traffic engineering and constraint-based routing properties. An automation, that reduces the work performed by network operators, will be achieved when the equipment can exploit these qualities and combine them with the functionality that is possible in the OTN. Details of the control plane for an OTN is unspecified. ODSI only defines how the control plane shall be used and how connected user equipment co-operates. See [14] for more information and details of their work.

2.5 Fault detection and location

To be able to automatically protect lightpaths that have been set up in an optical network two important parts need to be considered: detection and location of faults. A detected fault may need to be located to a specific wavelength, fibre, link or node before restoration of affected lightpaths can start. A good approach is to avoid the need to locate a fault by using nodes capable of detecting fibre cuts on all incoming or outgoing fibres whether it is an edge or core node. Here are some aspects of fault detection and fault location in an optical WDM system:

- Loss of signal (LOS) in a WDM channel or a fibre due to a fault may be detectable at each node the signal traverses. A fault that results in signal loss can be detected downstream of the signal path. A check of each node upstream from the detection point will locate the fault.

- Another, or complementary, approach to detect a fault is at lightpath destination nodes (egress edge nodes) where the optical signal is transformed into an electrical signal. Detection at the edge nodes can be made with framing solutions for each lightpath [2]. A frame is used for packaging of network traffic before sending on a fibre. The framing makes it possible to include signal quality bits. The quality bits can be checked for bit errors at the destination edge node. An unacceptably high bit error rate could trigger restoration actions.

- Location of faults can be found with a check for correlation of simultaneously detected faults. For example, a fibre cut could be localised if all lightpaths that use the fibre trigger a fault detection protocol. A fast fault localisation protocol built around this should be possible. The correlation is simply a map of the lightpaths that must fail if a fibre should be considered cut. However, the protocol may be too difficult to implement. See Example 1 for an illustration of why.

- Path protection makes locating a fault a non-time critical matter. The failed path can simply be restored without concern to where the fault is located. However, a fibre cut will in that case cause many separate path restoration processes that run independently at the same time. Therefore, it is probably a slow solution for fibre cuts. A better choice may be to first locate the fault and then start restoration in a more co-operative and economic way. But if a fault occurs in a wavelength channel, then at only one connection has to be restored. Knowledge about the location is in that case not important to make a fast restoration.

Example 1: Locating network faults

This is an example of a fault locating procedure that should not be time-critical by being involved in the restoration process:

> A simple network with four edge nodes and one core node is shown in Figure 3. Five lightpaths have been set up when fibre CE is cut. Failures in all lightpaths that use CE will be detected at their respective edge node. The fault is detected at edge node E in lightpath λ1. No other node detects the fault.
>
> All lightpaths have been mapped to a location table like this:

AB	AC	AE	BA	BD	CA	CD	CE	DB	DC	DE	EA	EC	ED
	λ1,λ4			λ2	λ5		λ1			λ2	λ3		

> In the attempt to locate the fault it is found to be at fibre CE which is the only match for λ1 in the location table; not at fibre AC because no failure is detected in λ4 at node C.
>
> The difficulty to implement the distributed location protocol is to make it correct. What if it is the fibre AC that is cut and the detection of fault in λ4 is somehow delayed? Then the protocol may incorrectly set the failing fibre to be CE when it should be AC. When λ4 failure finally is detected a new location protocol is started that also should include λ1 as failing, otherwise only the channel, not the fibre, that λ4 uses is decided to be the fault location. Note that more difficulties exist.

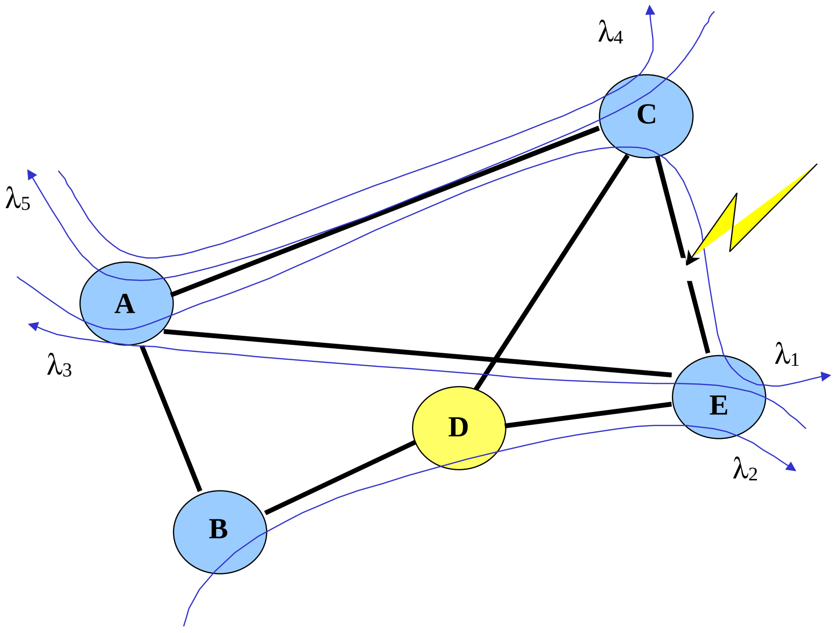


Figure 3: Node A, B, C, D and E form a small network. Lightpaths λ1, λ2, λ3, λ4 and λ5 have been set up and are in use by clients (e.g. IP routers not visible here) at the network edge nodes; D is not an edge node. The link between C and E is cut causing an alarm raised by E where λ1 failure is detected. Because that is the only alarm raised, either the fault can be located in a wavelength channel used by λ1 or in the fibre CE.

2.6 Reconfiguration of logical network topology

Network providers may want to reconfigure a network. The time a reconfiguration takes may be about one second with a logical network topology that can be changed through configuration by software. This time can be compared with the weeks it takes to reconfigure conventional SDH/SONET systems. However, reconfiguration of lightpaths while they are still in use introduces risk of disturbances in the traffic. The reconfiguration could be performed during periods of low traffic usage, or maybe temporally re-routed with electronic buffers to prevent losses.

Suppose there is a configured lightpath from node A to node B in an optical network. Unbalanced network traffic load makes it suitable to reconfigure the lightpath to use another path through the network. A switch of the old path to a new will induce a risk that packets sent from node A will never reach node B. The risk is of two kinds [16]:

- *Loss of packets* happens during actual switching of an OXC. Packets reach correct nodes but vanish inside them. Critical time where a packet can be lost is typically in order of 10ns.
- *Misdirection of packets* happens if they never reach the destination neither through the old nor the new path. Packets are misdirected to wrong nodes somewhere along the path.

The number of packets that will be misdirected during a reconfiguration depends on two things: The reconfiguration method and the relationship between the times each packet needs to traverse the old path compared to the new path. If the traversing time for the old path is less than for the new path, then packets can simultaneously be sent on both paths to prevent losses. See Figure 4B, which shows packets sent on both paths without misdirection of any packets. However, the opposite that the new path has a shorter traversing time is equally likely, and is shown in Figure 4A. It shows how both the triangular and the circular packets sent through the new path have been directed to another node than the destination. And when the switching is performed, the triangular and the circular packets along the old path have not reached the destination either. Thus, at least two packets have been misdirected. For example, misdirected packets can easily be of more than 1Mb data for a switch of paths where the difference in path traversing time corresponds to 100km of fibre at 2.5Gb/s capacity.

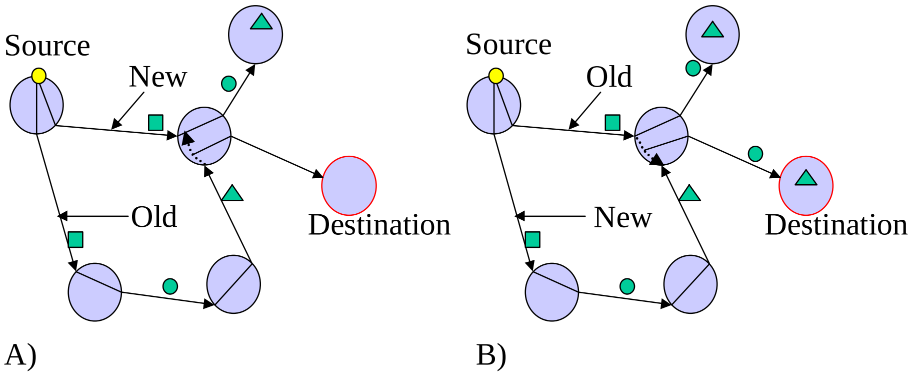


Figure 4: Misdirection of packets while reconfiguring in an optical network. The figure shows two examples: A) The circular and triangular packets do not reach their destination. B) No packets are misdirected because the relationship of old and new path traversing time is the right.

With the method in Figure 4 that simultaneously uses two paths, compared to misdirection of packets, loss of packets is of no big concern because only around 25 bits are lost for a bit rate of 2.5Gb/s during a switching time of 10ns. Note that only one switch, the last and nearest to the destination, will introduce losses.

A solution presented in [17] bases the switching on an OXC system where switching can be done without bit loss. The switching is not dependent on speed. Instead, it is done in a smooth operation where the output signal is gradually changed to the new. This means that the output signal consists of both old and new signals. This requires that the old and new signals are synchronised to be the same at the node where the switching shall be done. Either the old or the new signal is delayed to achieve synchronisation of the two signals. The synchronisation and the smooth switching prevent both misdirection of packets and loss of packets.

3 Project Specification

3.1 What has been done

- Development of a tool for network management of OTNs. The aim was to implement a tool that can perform connection and fault management. The connection and fault management should handle protection of critical connections. I.e., a network fault should trigger an automatic restoration of connections that fail to meet customers' requirements.

- Demonstration on how optical connections can be established between packet switches. OXCs forming nodes in a WDM network add a logical network topology to the physical topology. Switching in an OXC will change the logical topology and hence a user can dynamically alter the network to meet certain demands.

- Supervision of network performance. A user should be able to supervise the effects of configuration management, network faults and the network fault management. A graphical user interface has been implemented. It displays the network topology, connections, reservations and failed resources.

- Integration of the tool with an existing test network, which is metropolitan wide and consists of optically interfaced and electronically switched crossconnects (OiEXC) at four nodes in a ring around Stockholm. The tool is installed as several applications at computers connected to the test network. See Figure 5 for an illustration of an OiEXC.

- Analysis of distributed and automatic network management systems.

- Analysis for integration of packet switched network management and OTNs. The tool could interact with IP-switches to improve network usage and the switches could be configured to form lightpaths through the network.

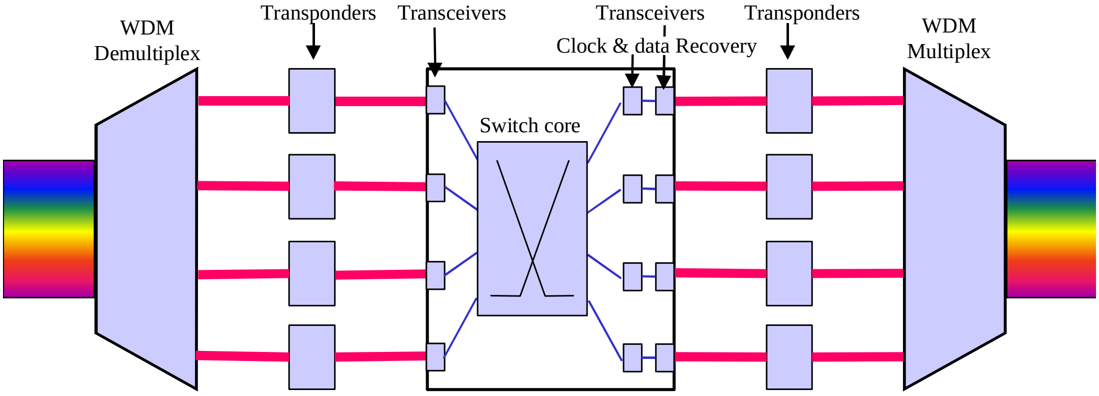


Figure 5: An OiEXC connecting one input fibre to one output fibre. A WDM system demultipxes four wavelength channels to the left. Each channel passes trough a transponder and transceiver, which regenerates a corresponding electrical signal. An electrical switch directs the signal to clock and data recovery, a transceiver, a transponder and finally a multiplexer.

3.2 How all have been done

A management system for control of one OiEXC has been developed in a previous master thesis [4]. The management tool implemented in this project, uses part of that system: A small agent that supervises and configures the OiEXC and a hardware control program that the agent uses for OiEXC access. A user interface application and a central management application, that can control a network of OiEXCs, were added to this. Each network node has one agent that controls one OiEXC, plus additional equipment such as IP-routers, which the management system is not aware of. The user interface application is currently the only way to use and supervise the functions that the tool provides.

The planning and implementation of the tool has been carried out in four steps, each one adds more distributed functionality:

1. Centralised management with only fault detection located at the network nodes.
2. Distributed set up of connections through signalling between nodes.
3. Distributed knowledge of the network's state and connections, with nodes being capable of route and resource calculation to find lightpaths for connections.
4. Distributed restoration of failed connections. After restoration, centralised management cleans and checks the effects of the performed restoration.

For each step changes was made to the central management application and the agent. The agent is divided into two parts: one for central management, to which only few changes was made (this part is basically the original agent from [4]), and one for distributed management, to which more intelligence was added as more of the system was made distributed. The central management application's functionality was reduced and adapted to the changes as more intelligence was distributed.

3.3 Equipment and environment used

The management system runs in any network that supports Java and CORBA [18]. It is developed for use in a test network with OiEXCs.

The test network has been set up in the Stockholm metropolitan area. It has four nodes; each equipped with an OiEXC, WDM equipment and a control PC that runs Windows NT. Forming an optical ring, fibre pairs interconnect the nodes (one fibre in each direction). An OiEXC connects the optical signals between the WDM equipment and additional equipment (e.g. IP routers).

The OiEXC has 16 optical input ports and as many output ports. Each output port can be connected to any input port by conversion to the electrical domain where the switching takes place and back to the optical domain for transmission. The OiEXCs are connected to PCs that run management applications. Figure 5 illustrates roughly the OiEXC that has been used; only four channels are shown though.

The control PCs use both Internet and wavelength channels in the fibres to form a management network. Figure 6 illustrates both the management network and the transport network. The management network is CORBA based, and illustrated as such. Four wavelength channels on each fibre are used for user transport. This means that eight output ports and eight input ports in a node are connected to fibres to and from neighbour nodes. The rest of the ports are free for user equipment, such as IP routers.

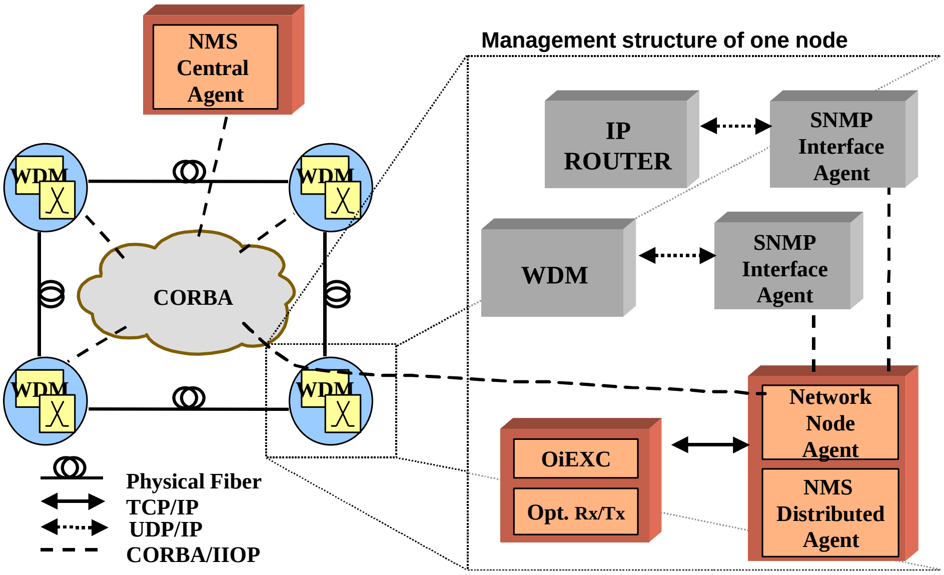


Figure 6: The physical network including WDM terminals and OiEXCs. The management applications, both central node and agent nodes, are also illustrated. The agents are separated into two parts: network part that handles OiEXC operation and distributed part that handles co-operation with central and agent nodes. A possible access to WDM or IP routers through SNMP is shown, but is not implemented in this project.

4 Implementation

The management system (the implemented tool) is capable of controlling the OiEXCs switching fabric to set up connections. It handles nodes, ports and channels and keeps track of allocations and reservations of output ports. All channels from one node to another are treated as one fibre, whether that really is the case or not. This is a simplification that does not map to reality, and should be configured by a user or with information at start up instead.

The *physical topology* includes the connectivity of nodes to links, channels to input ports and output ports to channels. The *logical topology,* in addition to the physical, also includes connectivity of input- to output ports in every node (i.e. configuration of each OiEXC).

The intention is to distribute the management system to network nodes. The system is implemented in Java and uses CORBA for communication between node applications, see Figure 6. The applications control the OiEXCs and keep track of the logical topology and the network's state. The state includes information on nodes that are up and running and those that are down for service, allocations and reservations of resources and lightpath connections that have been set up.

4.1 System structure

The system was designed with concern to the network of OiEXCs and uses centralised and distributed management nodes that communicate with each other. They co-operate and do all management of the OiEXCs. The management nodes are started as separate applications. Once started, they begin to communicate with each other. The test network in Figure 6 shows one centralised management node and four distributed management nodes. The distributed nodes are also called agents.

The nodes collect information on how they can contact each other from a name server at a known IP-address. A management node that starts logs on to the name server as a client and stores references there to interfaces that it provides. The node also searches for interfaces that other nodes have stored before it logs off and continues the start up process.

Figure 7 illustrates the structure of the management system. Part A) shows a central management node, where interaction with distributed nodes is hierarchical. Part B) shows a more distributed management node structure, which is the final result of the steps described in section 3.2.

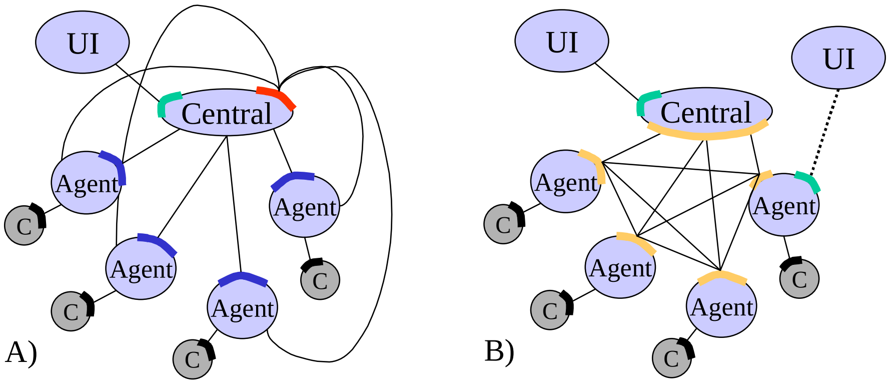


Figure 7: Management applications. Part A) shows a central structure with interfaces between OiEXC control applications to agents, agents to central management, central to user and central to agent. Part B) shows a distributed structure. The central to user interface could be implemented for agent to user as well.

4.1.1 Management nodes

The system includes two kinds of management nodes:

- Network element *agent* nodes. Each of these agents supervises an OiEXC's performance and if ordered changes the settings of it.
- One *central* manager node. The central manager is responsible for the network management and uses network element agents for this work.

In the implementation of the first step, centralised management described in section 3.2, the agent nodes do not communicate with each other. They only communicate with the central node, see Figure 7A. In the next three steps of section 3.2 distributed functionality is introduced in the agent nodes. They need to communicate network events and especially alarms of faults. Therefore, communication between all nodes is implemented with the first step of distributed functionality, see Figure 7B.

4.1.2 OiEXC control application

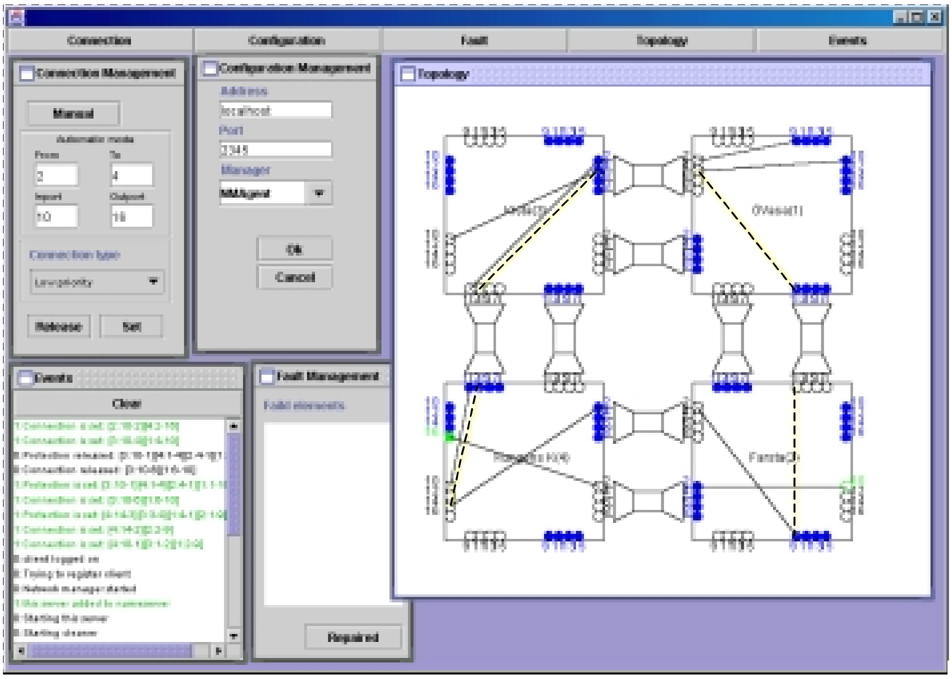
Control of OiEXC was implemented in a small application written before this project was started [4]. The application has a TCP/IP interface through which polling of OiEXC condition and setting of new cross connections are done. The control program is a slow work-around solution that should be rewritten and integrated with a commercial OiEXC and thus should provide a much faster control access and operation.

4.1.3 User interface application

In addition to the OiEXC control application, a user interface (UI) to control the network has been implemented. It provides a graphical view of the network of OiEXCs and a means for configuration and fault management. Figure 8 is a screen dump of the UI. It includes a topology window and a list of events that show all connections that have been set up and other information about the network.

The UI application's contact with the network management system is done with the central node. It is the user of the UI that requests configurations. The UI regularly requests information about the outcome of such configurations, or other information about the network's state. The central node does not send any information to the UI other than in response to the information requests.

Although the user interface solely connects to the central node the management system could, with small changes in program code, let it contact agents directly that can perform requested configurations. That would be necessary if the central node can not contact the agents, because of network partitioning, program bugs, or if the management system is fully distributed and the central node is removed. There are no obvious complications if this direct contact is allowed. However, only the aspect of connection and fault management is considered here and restrictions imposed by security management could be complicated.



From: 2
To: 4
Input: 10
Output: 16
Connection type: Low-priority

Address: localhost
Port: 2345
Manager: NMAgent

1:Connection is set: [illegible]
1:Connection is set: [illegible]
0:Protection released: [illegible]
0:Connection released: [illegible]
1:Protection is set: [illegible]
1:Connection is set: [illegible]
1:Protection is set: [illegible]
1:Connection is set: [illegible]
1:Connection is set: [illegible]
0:client logged on
0:Trying to register client
0:Network manager started
1:this server added to nameserver
0:Starting this server
0:Starting cleaner

Figure 8: The User Interface for the management system. Shows the physical topology. In the main window, nodes (squares) with connecting fibres and WDM-channels associated with ports (circles) can be seen. Over this structure, paths are shown. The dashed line is reserved resources, black are connections. The connection, that shares ports with the dashed line in the two upper nodes, is a low-priority connection.

4.2 Communication architecture

The communication architecture for this implementation is based on CORBA. For this project CORBA provides a good platform that allows investigation of distributed functionality. CORBA is a common and accepted communications architecture, which simplifies interoperability with other systems that also use CORBA.

The management nodes implement CORBA interfaces through which they communicate information, see Figure 6 and Figure 7. The interfaces were written for this management system and specify functions that can be called by nodes that obtain references to the interfaces. CORBA can be used as a communication architecture for distributed objects implemented in various programming languages [18]. Whatever the language, an object request broker (ORB) is needed for an object to call functions implemented by other objects. Some commercial ORBs are available, but Sun provides one for free with their Java 2 platform [18].

The OiEXC control applications implement TCP/IP interfaces. Functions in remote objects may be called directly through TCP sockets, which is the case for contact with OiEXC control applications. However, direct socket communication is quite dependent of the programming language. For example Java provides functionality for this, but then the communicating objects must be Java objects on both sides of a socket interface. CORBA can be used as a work-around for this dependence. It encapsulates distributed objects and hides their location, how they are implemented, what they are and how they are contacted.

Much of the functionality that CORBA provides has not been investigated. Additional communication services have been implemented in a message passing system: distributed logging of events and distributed transaction capabilities for mutual agreement of events. For this implementation, CORBA is simply used for communication of messages and for remote invocation of functions.

4.2.1 Object Interfaces

The CORBA interfaces implemented by agents, central node and the UI shown in Figure 7 can be described as:

- *Central client interface*, provided by the central node, is for operation and maintenance of the network by the UI. The UI regularly checks for new information at the central node and uses it to keep track of what is happening in the OTN and to inform itself about connection requests.
- *Central call back* interface, provided by the central node, is used by agent nodes when they want to notify the central node that it has messages queued. A level of urgency is attached to the notification. The interface is only used for centralised fault- and configuration management, see Figure 7A. This interface is unused for distributed management and replaced by the inter- management interface described below.
- *Agent client interface* is provided by an agent node and used by the central node. Only a few changes have been made to this interface that originally was implemented in [4].
- *Inter-management* is an interface that all nodes use for distributed functions and management. The interface replaces many configuration- and fault functions specified in the central call back- and agent client interfaces. This interface is required for nodes that participate in the message passing system (section 4.2.2) that was introduced with the distributed management.

The OiEXC control application implements a TCP/IP interface:

- *OiEXC interface* is provided by the control application and used by agent nodes. This interface was implemented in [4]. It allows monitoring and configuration of OiEXCs.

4.2.2 Message passing system

The distributed nodes are connected via a message passing system and use that to send management messages to each other. A message contains a timestamp and management information. The management information is the actual message. The timestamps help putting events described in messages in causal order, which means that the nodes independently can put the events in the same order. The timestamp also includes information about what events all other nodes have acknowledged.

The message passing system uses one simple function implemented in the inter-management interface described in section 4.2.1. The function is simple, just a plain method to receive messages, but on the inside of the nodes, where the interpretation of the messages is done and the logic and functionality is implemented, it is more complex.

The message passing system maintains a distributed log of events such as allocation of resources and failing elements, etc. The log grows in length for each logged event, but it is kept short and deleted wherever possible. An event can be deleted when a node, by investigating timestamps, finds out that all other nodes have received that particular message. Other nodes may still have the message in their log, because they do not know yet that it is safe to delete it. Eventually, all copies of a log message will be deleted.

Nodes implement transaction functions and use the support that is implemented in the message passing system. Transactions are used for allocation and reservation of resources. A transaction means that all participants of it will have to agree before an operation shall be conducted. If at least one participant does not agree then the transaction is aborted.

4.3 Programming language

The UI and the management nodes are written in Java. The OiEXC control application is the only exception (more details are found in [4]).

Java programs are compiled into Java bytecodes. Compared to a compiled program written in C, the Java bytecode represents machine code and the Java run time environment would be a computer processor. When running a Java program, each part of the program that is accessed by the run time environment is compiled into machine code. Sun’s runtime environment includes a just in time compiler, JIT, that compiles and stores the machine code for later and much faster access next time that particular code shall run [18].

The Java 2 version, that has been used here, includes a CORBA implementation. This made the choice of language a matter of convenience: One language for all programming. It would still be easy to change if another language would be needed. Another benefit with Java is that it is simple for programming, especially for distributed systems and for graphical interfaces. As a reference, the C language requires more tasks of the programmer: Explicit reference handling and memory allocation and de-allocation. And if Java is not enough, then it is easy to include and use parts of compiled code written in other languages into Java code. In this project Java provides sufficient performance and functionality. If other or better qualities will be needed in further work, then for example C++ would improve speed and Erlang would enhance development time, or maybe the new Mozart platform should be investigated.

The CORBA name server that has been used is an application that is included in Java 2. The server always starts fresh without any stored information. As long as the server is running it will successfully hold stored information, however, once it is shut down it will forget everything. Thus, in current implementation, the name server must live as long as the system is running. This necessity can be prevented with a name server that can use permanent memory, which probably is the case for most commercially available products.

4.4 Functionality

The functionality in the management system is implemented for configuration and fault management. Table 1 presents all main functions and services, for use internally of one or more nodes as well as for external access (functions or services that clients can use). The table shows where functions are used, which means that the same function may be found at more than one place.

Table 1: Main functions and services in the management system (the UI is not included).

	CONFIGURATION	FAULT	
Distributed management node			
Network element agent		Monitoring	Checks for changes in OiEXC configuration
		Fault detection	Detects LOS and decides if it corresponds to fibre or channel fault
	Configuration		Configures switching fabric.
Distributed agent		Alarm distribution	Reports faults to nodes that are listed as affected by detected faults
		Restoration	Distributed restoration of failed connections
		Protection	Associates short codes for faults. Maintains lists with nodes to alarm for each code.
	Connection allocation and set up		Allocates resources and sets up connections (transaction)
		Resource reservation	Reserves resources for protection purpose (transaction)
	Logging		Logs changes in topology (events)
	Reporting	Reporting	Reports changes to higher instance (central node)
	Routing	Routing	Resource constraint based route and wavelength calculation
	Transaction	Transaction	Agreement by voting among participating nodes to perform an event
Central management node			
	Connection allocation and set up		Allocates resources and sets up connections.
	Logging	Logging	Logs changes in topology.
	Monitoring		Checks if nodes go down
		Protection	Configuration and distribution of protection settings and information
	Cleaning	Cleaning	Cleans faulty or unnecessary configurations
	Reporting	Reporting	Reports changes to higher instance (UI)
	Routing	Routing	Resource constraint based route and wavelength calculation

4.5 Operation

4.5.1 Starting the system

An agent is preferably installed and run on a computer near the OiEXC that it controls. Here, a node includes the control application through which all operation of the OiEXC is managed. It is not necessary to have the agent close because it can be located anywhere as long as it has a TCP/IP connection to the OiEXC node, but a tight connection will reduce the network’s influence over OiEXC surveillance and operation. The best may be an OiEXC with processing capability and an IP-address. The OiEXC could also include a web server for downloading of the UI. Then, all applications could run without control PCs, which are a problem for fast protection.

Where the central node should be located depends on how time critical protection is handled. If the protection switching mechanism is fully distributed and the central node's only task is to clean allocations and reservations afterwards then the location is not that important. However, if the central node is involved in the time critical protection then what could be important is a central location to decrease the dependence of network performance. This is the case for the first three steps described in section 3.2. In the Winchester management network, it is placed at one of the four control PCs.

Both the central node and the agents must be configured when they are started. The configuration is done with a command line argument that specifies a file containing configuration settings. Settings for the central node include an IP-address to a name server and network topology information. The agents are provided with settings for configuration of the OiEXCs and the name server's IP-address.

First the name server is started. Provided the name server's network address is known, the node applications can then be started: One agent for each OiEXC and one central manager for all the agents. A node announces its presence to the name server, and then contacts every node that it can find in the name server. Each node will establish contact to all other nodes, because they add themselves to the name server before they search for other nodes in it.

An agent that starts tries to collect network topology and state information from other more updated agents. If that information is not available it simply has to wait. The central node knows the network topology but tries to collect network state information from the agents. If it cannot find any network state information at running agents, then it distributes the topology to them, which will result in a system without any allocated or reserved network resources.

The UI uses the GUI capabilities of Java to present the network topology and state. It is started and provided with a configuration file that specifies the name server's IP-address. A reference to the central node is retrieved from the name server. All contacts with the management system are done through the central node.

4.5.2 Running the system

The agents and the central node do not need any user input while running. All operation and maintenance is handled from the UI. Connections can be set up or taken down and network elements can be stopped or restarted. For example, if some network element is going to be replaced, it can be stopped to prevent following route calculations to include that element. The UI is shown in Figure 8.

Management nodes can be aborted by killing the node application. The rest of the nodes run and adapt to the change. They cannot use the OiEXC that was controlled by the aborted node. And protections that involve the node will not work. Restart of an aborted node must be done to fix this. A restart is done in the same way as the first time the node was started. Configurations will automatically be made to put the restarted node in the correct state (the state it had plus changes made after it was killed).

In the network management system, provisioned connections have one of three priorities: low, normal or protected. Low priority connections may use resources reserved for other purposes, but will in that case risk to be destroyed. If someone has reserved a resource that is used for protection, then a low priority connection using the same resource will risk pre-emption if a network fault should occur. When pre-empted, a low-priority connection is removed and has to be provisioned once again to work. Protected connections use reserved resources for recovery from connection failures. The connection and protection paths do not use the same node and link resources, which will protect from failures in links and nodes along the connection. Normal connections have no protection paths, but will not risk to get pre-empted as low priority connections do. Figure 9 shows how a protected connection is restored while a low priority connection is pre-empted and removed.

The central node handles uni-directional connections. To choose connection route, either an automatic mode where input consists of beginning and end points or a manual mode that allows explicit routes to be specified can be used. Resources and connections are selected by pointing and clicking with the mouse in a topology window that displays the different OiEXC's, input- and output ports and the connectivity through links and channels between them. See Figure 8 and Figure 9.

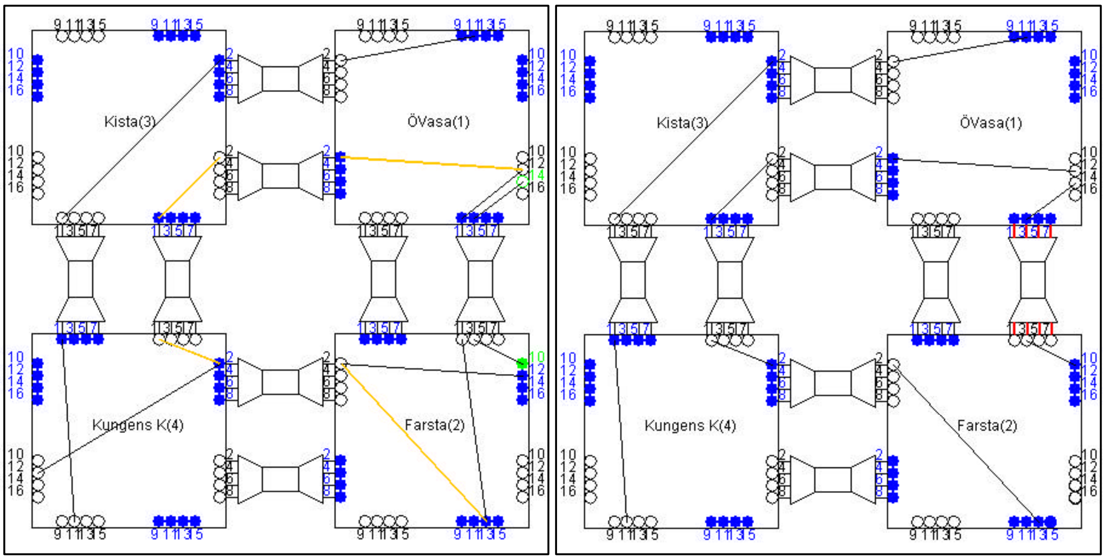


Figure 9: Two pictures of the topology window that shows what can happen with protected and low priority connections when a fibre fault is detected. The first includes a protected connection from input port 12 in node 1 to output port 13 in node 2 and a low priority connection from node 4 to node 2, which share output port 2 in node 4 with the protection path. In the second, the protected connection has been activated and the low priority connection has been removed because of pre-emption.

4.6 Management details

4.6.1 Fault management

The agents are responsible for detection and alarming of faults that occur in the OiEXCs and in the optical input signals. The detection is done by periodical checks of the signal level on all input ports. A fault has occurred if a signal level goes from high to low. If more than one such change is detected at the same time a decision is taken if they should be treated as separate channel faults or as one single fibre fault. Depending on the fault, a code is generated and sent in a short message to nodes that should take actions to limit the damage.

Messages with reports of fault are restricted and only sent to interested nodes. I.e., in the central case the central node is the only interested node. In the distributed fault management case every node with protection paths that should be used when a specific fault occurs is interested in the fault. However, if more than one fault (though unlikely) would occur and be detected at the same time and in the same node, then a concatenated fault report based on those faults will be sent to all nodes, regardless of their interest. This will make the restoration faster, because no time is spent on node interest look-up and the number of messages sent may be minimised. However, if the network is large and this will result in too many messages, then an additional fault interest group for multiple faults could be created at each node. I.e., members of such a group are interested of multiple faults detected at a specific node.

Configuration and reservation of protection settings is done in the central manager node. Necessary information such as switching actions and fault interest groups are distributed to the agents where it is stored and prepared for fast look-up when needed.

If a restoration attempt fails any possibly pre-empted connections will continue as soon as a cleaning process reverse the restoration attempt after the management system has become stable and no failure handling is going on. Failed connections and failed protections will be noted to the user who is responsible for further actions. An additional parameter could be one for connections that under all circumstances should be restored, which means failed protection should be proceeded with reconfiguration that includes new path calculation and connection set-up.

An extension of protection reservation can be to utilise reserved resources by sharing them. In the current implementation only one protected path exists for each reserved resource (1:1 protection). At the same time a low priority connection can use those reserved resources. An addition could be to let protected paths that are completely diverse reserve the same resources for protection (1:N protection). The first to reserve is given the highest priority, which means that it is allowed to pre-empt a shared resource if a lower prioritised protected path should happen to use it due to a previous network fault. Note that the diversity constraint prevents this pre-emption to happen for one single fault. At least two are needed before it can happen.

One problem must be solved in an implementation of shared protection reservation. Figure 10 shows three connections in a network. Each has an additional protection path. Two of the protections have been activated due to faults. The third connection’s protection path share resources with the other two connections. If a fault causes the third protection path to be activated pre-emptive actions must be taken for the path to be set. The restoration succeeds only if both of the already activated protection paths have lower priority. Problem occurs if one has higher and the other lower priority. A faulty protection implementation may cause one half of the protection path to be set with pre-emptive action and thus ruin one protection path when it should have been avoided.

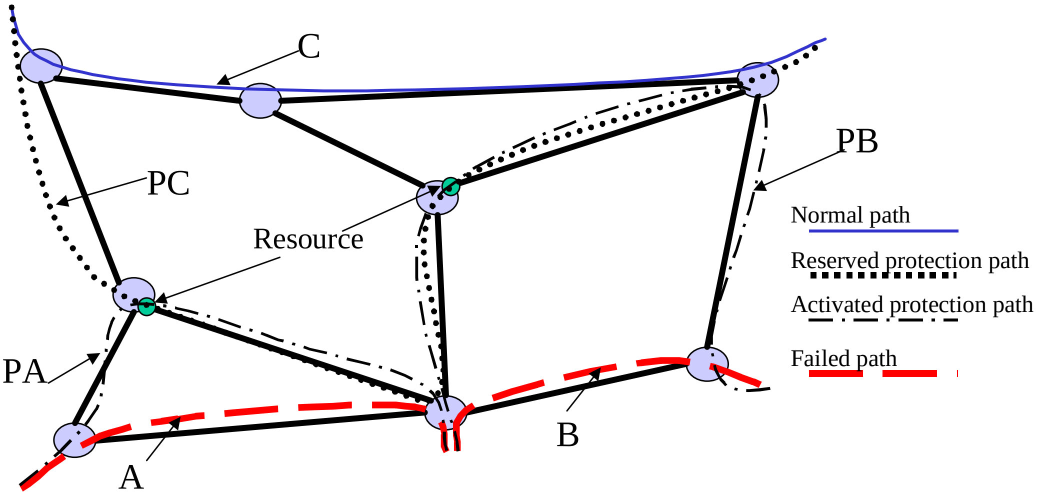


Figure 10: An illustration of a network with lightpaths sharing resources for protection purposes. PA, PB and PC share resources at common nodes. Two of the protections have been activated and are working. If PC should need to be activated it will need higher priority than both PA and PB.

4.6.2 Configuration management

Configuration is done for both the management system and the transport network. The management system is configured on start up of management nodes. The agents use configuration files that specify local environment settings. The central node uses a configuration file that specifies network nodes and network topology. The topology information is distributed from the central node to the agent nodes.

The transport network is configured from the user interface application. The user can initiate a connection request either by an explicit specification of the path or by using the automatic path calculation. The automatic path calculation mode uses Dijkstra's shortest path algorithm [19] for route calculation, with the constraint of using only free and available resources in the network. It is also possible to add other path constraints, i.e. links and/or nodes that should not be used by a calculated path. But this is currently only done by the fault management, which constraints the protection paths not to traverse the same resources as the protected paths.

A path can be calculated either at the central node or at an agent node. Currently, the distributed approach is used. The first node in a path is delegated the responsibility of path calculation and then initiates the connection set up process.

Dijkstra's algorithm uses costs added to links. The sum of all link costs in a path specifies how expensive a path is. The algorithm finds all possible paths from a node that are the least expensive ones. The calculation of cost has been implemented with respect to the load on fibres and to the number of fibres a path traverses. Currently the load factor is omitted in the calculation because of no obvious advantages from it. The cost is simply the same as the number of fibres that a path uses.

The central node takes down connections without involvement of agents. The agents receive information about the change from the distributed log in the message passing system. This is a simplification. The tear-down process should involve the agent nodes to do proper physical OiEXC configurations. For example to prevent further access to a source that sends information, it may be necessary to also change the path in a connection that is taken down. Currently this is not done, but would be simple to add in the implementation.

4.6.3 Distributed management information

A change to the logical topology that is made while switching is considered to be an event. The performed event is logged at the node responsible for it. The update of the distributed log is automatically propagated and sent to the other nodes by the message passing system. Nodes receiving the log update make necessary updates of their respective information about the logical topology. Some of the nodes that receive the update discard it, for example because they were participating in the event and know of it already.

The logical topology is distributed to all nodes through the distributed log system. Multiple versions of the topology are therefore located at the nodes. The versions may be different because, while some nodes are performing configurations of a connection others can be involved in the tear-down of another without knowing of each other's operations (an operation is one or more events).

A group of nodes that is about to perform an operation together should have consistent logical topologies. The topologies are made consistent with help from the distributed log system. And the fact that an operation actually is a transaction operation prevents inconsistencies to cause operations that are not allowed. The transaction technique essentially means that an operation is performed only if all participants agree that it should be, otherwise it is aborted. The node that controls an operation (the first node in a connection path) sends its log, containing events, to all other nodes when the operation has completed. Every node updates the logical topology and logs an event when information about it is received.

Events are propagated to all nodes, by the node that is responsible for it, immediately after it has happened. A slower and more precise propagation could be implemented: More precise in a way that events already received at a node are not sent to it again and slower to prevent a too heavy management network traffic.

A note about restarting nodes: A node N that restarts from a crash or a deliberate shut down collects network state information and all logged event messages from the node with the latest update of N's timestamp (the timestamp before it was closed). This ensures that N will be at least as updated as before it was closed, and it preserves the consistency of the network's state and the distributed log.

5 Testing and results

The implementation was tested in simulations and to some extent in a realistic trial. The tests were performed on a network that includes four nodes in a ring with four wavelength channels in both directions between the nodes. The agents do not establish contact with their respective OiEXC in a simulation, but will continue to run and do what they are ordered to do. Connection requests are completed within seconds with configuration of real OiEXCs. Restorations from simulated fibre-cuts and channel faults have been done successfully. All protected connections that are affected by the faults are restored. The restoration times are independent of the number of connections that are restored. However, measured values on restoration times can only be estimates, because the real network is simulated and all applications are run on one or more computers. Restoration times could be measured in the real trial, but would be in the order of hundreds of milliseconds because the OiEXC control application has not been optimised. Probably, the application must be integrated into the OiEXCs to operate fast enough.

Time values are obtained by calling a Java function that gives the system clock in milliseconds. Such values can be subtracted to give elapsed time in milliseconds. Even though the precision is in milliseconds, the results do not have better resolution than tens of milliseconds. This may be a result of process scheduling by Java and the operating system, and is a problem especially for simulations on one computer that runs multiple nodes. Time critical code in the implementation has been given high priority, which makes that code run faster. Still, the time it takes for one node to receive an alarm and perform protection switching varies between 0ms and 30ms for the nodes that perform the same work. This variation is mainly a result of multiple nodes that at the same time need processor time to perform protection routines, and all running at high processing priority.

Tests have been conducted in a not completely realistic trial, but a combination of one real node on one computer and three simulated on another computer. The real node shared the processor with the OiEXC control application. A connection was set up and then automatically restored due to a loss of signal when one input channel to the OiEXC was physically disconnected. The time from detection point to completed alarm routine was 30ms (measured in the real node). The time after the alarm routine to the completed configuration of the OiEXC was 30ms (in the real node), which is on the edge of the corresponding time span of 0ms to 30ms measured in the simulated nodes. This means that the total time from detection to configuration was 60ms. To estimate a total restoration time, an addition of time in the OiEXC control application before detection and between configuration and physical switching must be made. A fair estimate of the total time is 80ms if the control application is optimised.

Restoration times depend highly on the number of messages needed in the restoration routines and how sequentially they are sent and processed. The best result was achieved with delegation of restoration actions to the node agents. The result was also improved with delegation of fault alert responsibility by letting alerts to be sent from the detection point. But the best effect was achieved through optimisations of code in critical routines, which was made better for each step made towards more distribution. An optimisation of the first step, which currently needs more than 100ms for restoration, is expected to give almost the same performance figures as the fourth and most distributed step. This may not be true for a larger network where more than four nodes must be reconfigured.

Tests between two real nodes located in different Internet domains were also performed. Java and CORBA caused contact problems that did not occur with both nodes in the same domain. Some certain settings, not documented by SUN, had to be made in the control PCs before the contact worked correctly.

A small note important to Java performance: Java includes a just in time compiler, JIT, that in real-time stores compiled code making it unnecessary to compile the same code more than once. The real time critical functions, such as protection routines, must be pre-run at initialisation before they are ready to run as fast as possible. This is not implemented, but by manually letting all nodes perform restoration routines at start-up, they will be ready.

6 Conclusions

6.1 Automatic control

Performance management could include automatic reconfiguration of network topology. Only connections that allow reconfiguration would be managed automatically to improve performance. Triggering of reconfiguration could be alarms about degrading signal strength or quality, which would result in an attempt to avoid using the degrading equipment. The reconfigurations mean either new connections between the same user equipment, or redirection of connections to traverse intermediate user equipment. Reconfiguration could also be used to balance traffic loads when information about unbalanced loads during some time in certain areas is available. User equipment must be able to adapt to changes in topology. For IP routers, forwarding tables have to be updated.

6.2 Dynamic reconfiguration of lightpaths

Reconfiguration of the logical network topology introduces loss of packets and misdirection of packets. Both can be avoided with sufficient delays of signals while reconfiguring. The question is if the cost is low enough to offer this service. While the loss of packets probably is a few bits the misdirection of packets may be more than 1Mb. The best may be to accept losses and simply warn the user before a reconfiguration. Anyway, IP routers could be informed of the change for update of forwarding tables, and can temporarily change the forwarding to other destinations not using the path that is about to change. The path is then changed and the forwarding tables could be updated again to use the new path.

6.3 Distributed management

Restoration of protected connections can be made even if parts of the network elements are out of order (provided that the parts are not used to detect or restore from the fault). It is possible for a node to send a simple message that triggers restorations directly to nodes affected by the fault, thus reducing network traffic and delays and making the restoration as fast as possible. Functions such as cleaning of reservations and allocations after the restoration finishes can be the responsibility of a single node that has a central role or it can be distributed as well. Some distributed functions may need complex changes when upgrading; changes that would be small and easy to perform in a centralised processing environment.

6.4 Integration with WDM or IP routers

The optical management system could use WDM system functions for triggering of fault reporting or for performance management. For example a CORBA interface could be used to collect parameters for signal quality. Decisions based on low signal quality could trigger automatic reconfiguration of the logical topology. Or, an alarm could be sent to the user that could take actions when a monitored signal passes a threshold because of degradation in the transmitting laser.

IP routers could reconfigure the logical network topology to meet changes in the traffic loads or act in response to low signal quality reports from the optical management system. For example the ODSI interface would allow user equipment to reconfigure the OTN. Optical MPLS would be a tighter integration, in which reconfiguration of the logical network topology can be made by using data about transported traffic from IP routers. For instance, two routers could set up a direct connection to prevent overloading an intermediate router.

6.5 Further work

One further goal is to implement an optical control plane that guarantees restoration times below 50ms. A more deterministic process behaviour is then needed and can be obtained with correct thread priority: Propagation of fault alerts is highest, followed by protection switching and fault detection; all other tasks has lower priorities.

The alert propagation routine has to be optimised if the current architecture will be used. Even with only four nodes it seems too slow (30ms). The routine is sequentially run at the node that detects the fault and the time will approximately increase with 10ms for each additional node that has to be alerted. First, a real asynchronous message delivery should be implemented, either by using a commercial ORB that can do that, or by using one new thread for each message on the sender side. Second, the routine should be distributed to minimise the time it takes before the last node has received the alarm. I.e., the alert propagation should be shared by an optimum number of nodes. Figure 11 shows an optimum of 15 nodes alerted by a distributed propagation routine. Note that this would affect the probability of the control plane being able to function correctly. For example a node that fails to alert a fault to other nodes will ruin part of or the whole protection even though all other nodes succeed to alert the fault.

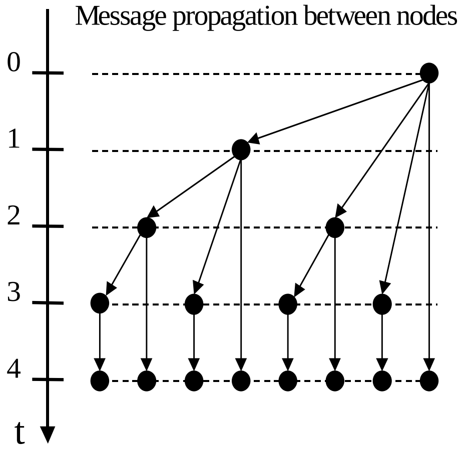


Figure 11sage that is optimised with respect to speed and amount of destination nodes. The message reaches 15 nodes within 4 time units. The operation to receive the message and then forward it is included in each time unit.

A thorough evaluation of current architecture is needed. The measured times are only estimations. Too many processes (several for each agent) share the same processor. CORBA has specification of asynchronous calls to functions, though not implemented in Java 2. In current implementation a CORBA function call is blocked until the function in the interface returns. This is one reason why the alert reporting time is proportional to the number of nodes.

7 Abbreviations and denotations

ATM	Asynchronous transfer mode
Channel	Optical WDM channel
CORBA	Common object request broker architecture
IETF	Internet engineering task force
IP	Internet protocol
LAN	Local area network
Lightpath	A series of optical WDM wavelength channels
LSP	Label switched path
LOS	Loss of signal
MPLS	Multi protocol label switching
OADM	Optical add drop multiplexers
ODSI	Optical domain service interconnect
OiEXC	Optically interfaced electrical crossconnect
ORB	Object request broker
OSI	Open systems interconnection
OTN	Optical transport network
OXC	Optical crossconnect
Protection	Fault detection and isolation, protection configuration and reconfiguration
Restoration	The process of restoring protected connections
SDH	Synchronous digital hierarchy – International version of SONET
SONET	Synchronous optical network – U.S. version of SDH
TDM	Time division multiplexing
UI	User interface application
WDM	Wavelength division multiplexing

8 References

1. R. Ramaswami and K.N. Sivarajan. “Optical Networks – A Practical Perspective”. ISBN 1-55860-445-6. Morgan Kaufmann Publisher, Inc. 1998.
2. E. Almström, P. Evaldsson, S. Hubendick, C.P. Larsen, S. Larsson, and C. Wickman. “Requirements and Solutions for Reconfigurable Metro WDM Networks”. NFOEC. 2000.
3. S. Hubendick, J. Söderqvist. “Management of Connection Services in Dynamic Optical Networks”. ECOC. 2000.
4. S. Hubendick. “Control and Management of WDM Networks with Opto-Electric Cross Connects”. TRITA-NA-E9928. Telia Research AB. 1999.
5. On-line dictionary on information technology. <http://www.whatis.com>. (2000).
6. W. Stallings. “Data And Computer Comunnications”. ISBN 0-13-571274-2. Prentice Hall, Inc. 1997.
7. A.S. Tanenbaum. “Computer Networks – Third edition”. ISBN 0-13-349945-6. Prentice Hall, Inc. 1996.
8. ITU-T Rec. G.841. “Digital networks – types and characteristics of SDH network protection architectures”. July 1995.
9. N. Chandhok, et al. “IP over Optical Networks: A Summary of Issues”. <http://www.ietf.org/internet-drafts/draft-osu-ipo-mpls-issues-00.txt>. July 2000.
10. W. Stallings. “SNMP, SNMPv2 and CMIP The Practical Guide to Network Management Standards”. ISBN 0-201-63331-0. Addison-Wesley Publishing Company. October 1993.
11. T. Kauppinen, A. Gavler, M. Adrian, C. P. Larsen, R-P. Braun, N. Hanik, T. Edhag, L. Johanneson. “Demonstration of A Histogram Based QoS Tool Integrated with a Network Management System”. ECOC. 2000.
12. C. Semeria. “Multiprotocol Label Switching Enhancing Routing in the New Public Network”. White paper. Juniper Networks, Inc. <http://www.juniper.net/techcenter/techpapers/mpls/mpls.htm>. (September 1999).
13. N. Ghani. “Lambda-Labeling: A Framework for IP-over-WDM Using MPLS”. *Optical Networks Magazine, vol. 1, No. 2*. April 2000.
14. ODSI. “ODSI Home”. <http://www.odsi-coalition.com/>. (June 2000)
15. A. Manzalini. “Milestones for the Evolution toward an Integrated Optical Transport Network”. *Optical Networks Magazine, vol.1, no 1*. January 2000.
16. S.N. Larsson and S. Hubendick. “Reduction of Hop-Count in Packet-Switched Networks using Wavelength Reconfiguration”, Optical Network Design and Modeling, pp. 1004-1024, Feb. 2000.
17. T. Shiragaki, N. Henmi, T. Kato, M. Fujiwara, M. Misono, T. Shiozawa and S. Suzuki. “Optical Cross-Connect System Incorporated with Newly Developed Operation and Management System”. *IEEE Journal on selected areas in communications, vol. 16, no. 7*, 1179-1189. September 1998.
18. Sun Microsystems, Inc. “java.sun.com – The Source for Java(TM) Techonolgy”. <http://java.sun.com>. (August 2000).
19. R. Perlman. “Interconnections – Bridges and Routers”. Addison Wesley. ISBN 0-201-56332-0. November 1996.