Applying p-cycle protection for a reliable IPTV service in IP-over-DWDM networks

Ahmed Frikha, Bernard Cousin, Samer Lahoud

Journal of Internet Services and Applications
a SpringerOpen Journal

RESEARCH Open Access

Applying *p*-cycle protection for a reliable IPTV service in IP-over-DWDM networks

Ahmed Frikha*, Bernard Cousin and Samer Lahoud

Abstract

Today, television over Internet protocol (IPTV) has become very popular and service providers must deal with the rapid growth in the number of IPTV customers. Service providers must ensure the reliability of IPTV to satisfy customers' needs, as a network failure could disrupt an IPTV transmission.

Survivable multicast routing is important for providing a reliable IPTV service. Generally, most carriers route multicast traffic using the protocol-independent multicast source-specific mode (PIM-SSM) based on the routing information provided by the interior gateway protocol (IGP). Restoration using the IGP reconfiguration is slow and typically takes from 10 to 60 seconds. To ensure a fast restoration, we consider node and link failure recovery in the Dense Wavelength Division Multiplexing (DWDM) optical layer. The backup path is provided in this layer. Thus, the multicast tree does not change at the IP layer (the logical links do not change) and the restoration time is faster (typically of the order of 50 to 80 ms).

In this paper, we apply *p*-cycles in IP-over-DWDM networks to provide a robust IPTV service. In addition, we propose a novel concept for node protection using *p*-cycles to achieve more efficient resource utilization. We also propose a new algorithm, the node and link protecting candidate *p*-cycle based algorithm (NPCC). This algorithm integrates our new concept for node protection. Extensive simulations show that it outperforms the existing approaches in terms of blocking probability, resource utilization efficiency and computation time rapidity.

Keywords: IPTV service; multicast routing; IP-over-DWDM networks; Reliability; *p*-cycles; node and link protection

1 Introduction

Nowadays, many telecoms companies offer the television over IP (IPTV) service and distribute television channels using backbone networks. An IPTV service requires stringent quality of service constraints (for example, for packet loss, jitter and end-to-end delay) to satisfy customers' needs. Service providers must also ensure the reliability of the IPTV service. A simple link or router failure could disrupt the television content distribution for several seconds, if no protection mechanism is implemented.

IPTV contents could be carried using the IP multicast to save bandwidth capacity. In fact, multicasting enables a single packet to be sent to multiple destinations at once. Although many multicast routing algorithms have been proposed for IPTV services, most of the carriers today implement the protocol-independent multicast source-specific mode (PIM-SSM) [1]. For many reasons, this protocol is efficient for internet broadcast-style applications such as IPTV. Obviously, PIM-SSM is simple to implement for network operators thanks to the source-specific mode (SSM) [1], which makes this protocol ideal for IPTV. SSM does not require the network to maintain knowledge about which sources are actively sending multicast traffic, unlike the internet standard multicast (ISM) protocol. This advantage makes PIM-SSM more scalable than ISM.

To ensure IPTV reliability, survivable multicast routing must be guaranteed. Moreover, service providers must ensure a fast restoration time for link and router failure recovery. The PIM-SSM protocol uses the routing information provided by the interior gateway protocol (IGP) to compute a multicast tree. Thus, restoration at the IP layer using IGP reconfiguration requires IGP to be aware of the failure, then PIM-SSM can use the new IGP shortest paths to rebuild a new multicast tree using the prune-and-join process. This operation is slow, and typically takes from 10 to 60 seconds [2]. To avoid rebuilding the multicast tree

*Correspondence: ahmed.frikha@ahmedfrikha.com
IRISA, University of Rennes 1, Campus universitaire Beaulieu, 35042 Rennes, France

and to ensure a fast restoration, we consider node and link failure recovery at the DWDM layer. The backup path is provided at this layer. This makes restoration time faster as the multicast tree does not change at the IP layer (the logical links do not change).

The *p*-cycle protection approach was introduced by WD Grover [3] for link failure recovery at the DWDM layer. A *p*-cycle is cycle-oriented spare capacity preconfigured at the DWDM layer. In Figure 1 we show an example of a *p*-cycle in an optical network. Note that a *p*-cycle does not traverse a node or a link more than once. Moreover, a *p*-cycle is oriented.

The advantage of using *p*-cycles for protection can be summarized in two main points. First, *p*-cycles ensure a fast restoration time (typically of the order of 50 to 80 ms) as the protection is done at the DWDM layer and *p*-cycles are preconfigured [4]. Second, the *p*-cycle protection approach can achieve an efficient use of the network capacity compared to other protection approaches, such as the one-plus-one (1 + 1) and the one-by-one (1:1) restoration methods. In fact, a *p*-cycle can protect both on-cycle links and straddling links. An on-cycle link belongs to the *p*-cycle, and is directed opposite to the *p*-cycle. In Figure 2, we show an example of an on-cycle link protected using a *p*-cycle. The on-cycle link is represented using a red line and the protection segment provided by the *p*-cycle is represented using a dashed green line. In this figure, we see that the *p*-cycle and the failed link are in opposite directions.

A straddling link does not belong to a *p*-cycle. However, its extremity nodes are traversed by the *p*-cycle. The *p*-cycle provides two protection segments: one protection segment for each directed-link. In Figure 3, we show an example of a straddling link protected using a *p*-cycle. In this figure, we show the protection segment that protects the directed-link used by the light tree. The protection segment in the opposite direction to the link is not shown in this figure. This characteristic of *p*-cycles allows the required backup bandwidth capacity to be reduced.

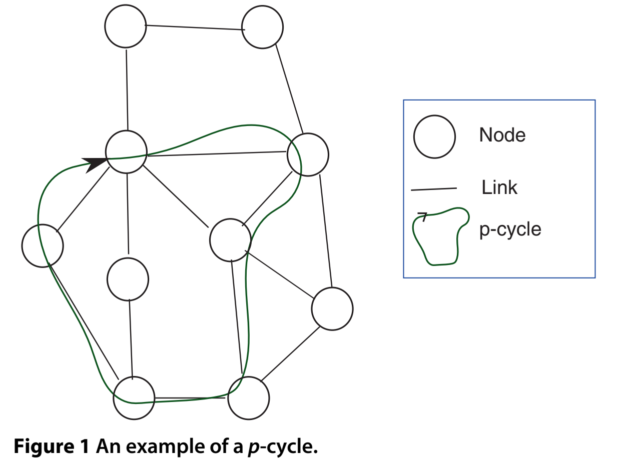

Figure 1 An example of a *p*-cycle.

The *p*-cycle technique was extended to support node protection in the DWDM layer using the node encircling *p*-cycle concept (NEPC) [5]. According to this concept, a protecting *p*-cycle of a given node must link all neighboring nodes of the failed node. This constraint is too strict. The method could discard some nodes that can be protected by the *p*-cycle but which do not satisfy the constraint. This will affect the efficiency of the *p*-cycles in terms of capacity saving.

In this paper, we consider link and node failure recovery at the DWDM layer using *p*-cycles. We extend the node protection concept of the *p*-cycle approach to achieve more efficient resource utilization. Then, we propose a novel algorithm, the node and link protecting candidate *p*-cycle based algorithm (NPCC). The NPCC algorithm integrates our proposed concept for node protection. This algorithm ensures node and link failure recovery based on a set of candidate *p*-cycles to overcome the high computation time problem.

The rest of this paper is organized as follows. In Section 2, we present the IPTV architecture and discuss the restoration mechanisms to ensure a reliable IPTV service. In Section 3, we extend the *p*-cycle protection concept for protecting nodes in light trees. In Section 4, we present our novel algorithm for combined node and link failure recovery using the novel node protection concept. Extensive simulations and numerical results are presented in Section 5. The conclusions are given in Section 6.

2 IPTV architecture and restoration mechanisms

In this section, we present the main components of the IPTV architecture and we give an example. Then, we discuss the restoration mechanisms, and we highlight the advantages of applying the *p*-cycle protection approach for IPTV.

2.1 IPTV architecture

The main components of an IPTV architecture are [2-6]:

- A super headend (SHE): The SHE is located in the core network. Also called the IPTV backbone, it collects television content from television networks, such as satellites and off-air distributions. After video processing, encoding and management, the SHE distributes the television content using IP routers to multiple video hub offices (VHOs).
- A video hub office (VHO): A VHO receives IPTV content transmitted by the SHE through the IPTV backbone routers. Then, it combines this content with the local television and the video on demand (VoD) content. The SHE routers, the VHO routers and the links that connect them form the IPTV

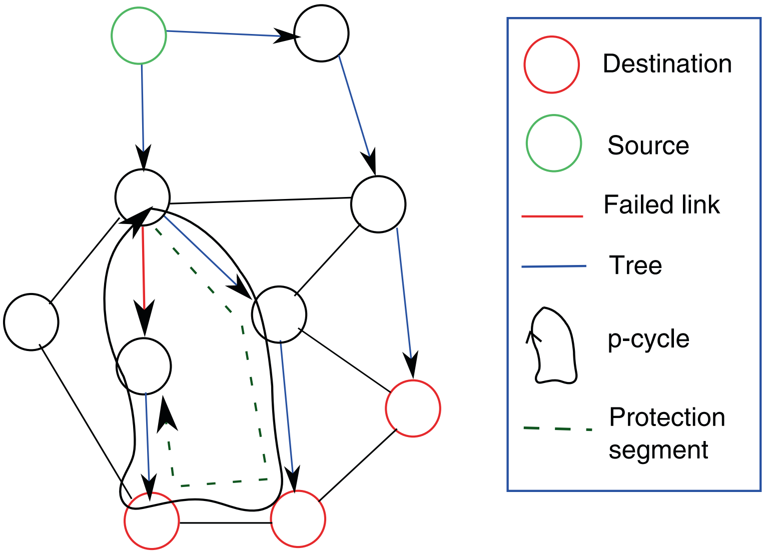

Figure 2 On-cycle link protection.

backbone. Each VHO in turn serves a metro area by transmitting the IPTV content to multiple video serving offices (VSOs).
- A video serving office (VSO): A VSO contains the aggregation routers that aggregate local loop traffic from subscriber homes, i.e., local digital subscriber line access multiplexers (DSLAMs).

A simplified example of an IPTV architecture is illustrated in Figure 4. In this example, the SHE gathers the national channel content from off-air and the international channel content from satellites. Then, it sends this content to multiple video hub offices (VHOs) using IP multicast and through the underlying DWDM layer. IP multicast is very important for saving network bandwidth as it allows a packet to traverse a link once to reach multiple destinations. The example does not show the traversed IP routers that connect the SHE to each VHO. The figure shows a house with a television and a set-top box, which is connected via a residential gateway to a DSLAM, connected in turn to a VSO.

PIM-SSM is largely used for IPTV video distribution with an IPTV backbone. A multicast tree is computed using this protocol based on the routing information

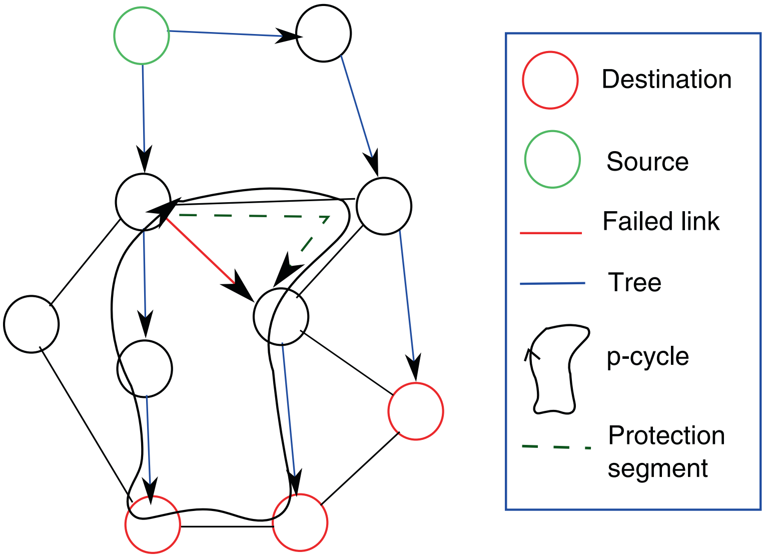

Figure 3 Straddling link protection.

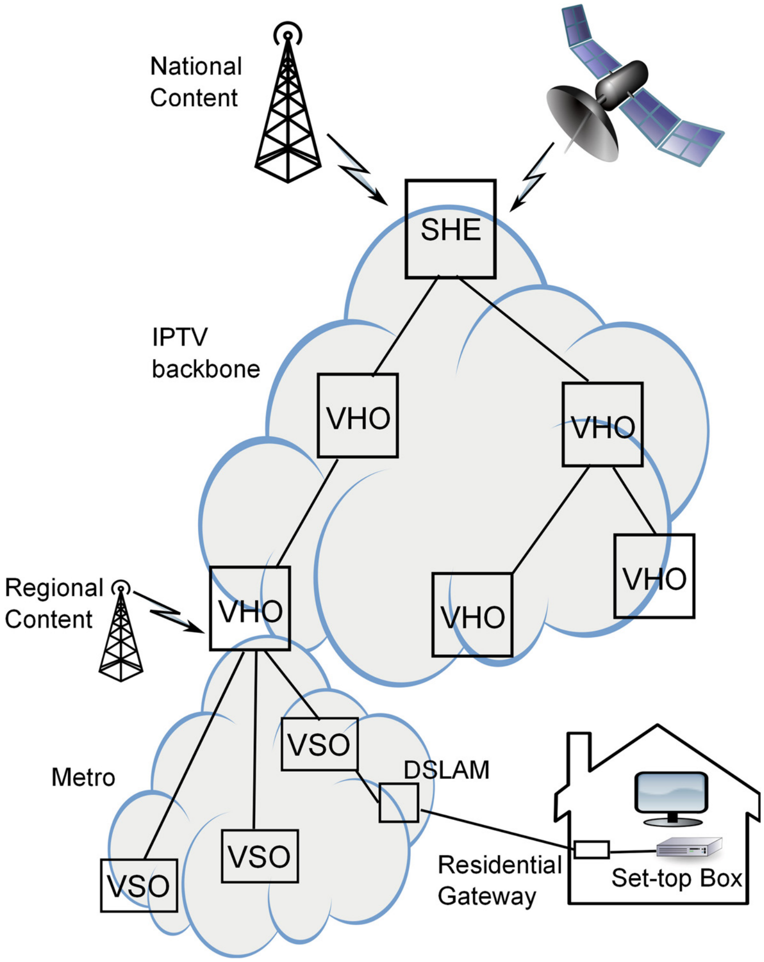

Figure 4 Example of an IPTV architecture. DSLAM, digital subscriber line access multiplexer; IPTV, television over IP; SHE, super headend; VHO, video hub office; VSO, video serving office.

provided by the IGP. The multicast tree is used to deliver IPTV content from the SHE to each VHO. Each television channel is assigned to a unique multicast group.

An IPTV service requires a high bandwidth and stringent quality of service constraints. In particular, IPTV is very sensitive to packet loss, as one lost packet could disrupt video quality. Delay and the jitter are also critical for the quality of an IPTV service. These three quality of service constraints rely on network dependability. A link or node failure could disrupt the IPTV service, if there is no restoration mechanism for the multicast tree.

2.2 Restoration mechanisms

With IGP reconfiguration, after a link or node failure, PIM-SSM must rebuild the multicast tree. But before that, IGP must be aware of the failure and compute the new shortest paths at the level of each router. PIM-SSM will use these new shortest paths to rebuild the multicast tree using the prune-and-join process. This approach is not suitable for a real-time IPTV service as the restoration process takes too much time, typically between 10 and 60 seconds [2]. With multi-protocol label switching (MPLS) fast reroute protection, the restoration time can be reduced to between 50 ms and 100 ms [2]. Backup label-switched paths (LSPs) are pre-established, and stored in the router forwarding tables and this makes fast rerouting possible at the IP MPLS layer.

Other restoration mechanisms have been used in the DWDM layer and can achieve lower restoration times. One-plus-one (1 + 1) and one-by-one (1:1) restoration can be implemented in the DWDM layer. The restoration time for these approaches is less than 20 ms [2]. However, they are not efficient in terms of bandwidth saving, as backup paths cannot be shared. In these approaches, a backup path is dedicated for one and only one working path. The *p*-cycle protection approach, described in the previous section, ensures node and link failure recovery while maintaining a fast restoration time (typically of the order of 50 to 80 ms) [4]. Moreover, this approach achieves an efficient use of the network capacity compared to the other protection approaches.

These restoration mechanisms are proposed for unicast traffic. Some other restoration mechanisms focus on protecting multicast trees against network failures. In 2009, F Zhang and WD Zhong proposed the efficiency-score based heuristic algorithm of node and link protecting *p*-cycle (ESHN) [7]. Although the ESHN algorithm has a lower blocking probability than the OPP-SDP algorithm [8] and the ESHT algorithm [9] in dynamic multicast traffic, ESHN does not efficiently use the protection capacity provided by a *p*-cycle, especially when protecting nodes. The ESHN algorithm does not take into consideration all nodes that a *p*-cycle can protect when selecting a protecting *p*-cycle. This is due to the two hard constraints imposed by the method used by ESHN for protecting nodes. The first constraint is that a node protecting a *p*-cycle has to link all one-level downstream nodes of the failed node. The second constraint is that the *p*-cycle must contain one of the upstream nodes of the failed node in the light tree. Of course these reduce the computation time for the algorithm as they limit the search space for the *p*-cycles. However, they prevent the ESHN algorithm from achieving the best resource utilization. Furthermore, when the traffic load is high, the computation time for the ESHN algorithm remains high and does not satisfy the IPTV service requirements.

In this work, we focus on the design of a reliable IPTV service. We use link and node failure recovery at the DWDM layer to give a short restoration time. We use *p*-cycles to ensure efficient use of network capacity. We also extend the node protection of the *p*-cycle approach to achieve more efficient resource utilization. In Section 3, we provide a detailed study of node protection using *p*-cycles and we present our proposed concept for protecting nodes in the light tree.

3 Node protection using *p*-cycles

3.1 Existing approaches for node protection using *p*-cycles

In this section, we present some existing well-known concepts for node protection using *p*-cycles. NEPC [5] uses *p*-cycles for node protection. Figure 5 illustrates an example of node protection using this concept. The *p*-cycle must traverse all neighboring nodes of the failed node to protect it. The drawback of this concept is that in some cases finding such a *p*-cycle is not possible. Moreover, some *p*-cycles that do not meet this constraint could protect the failed node while reserving less spare capacity. The constraint imposed by this concept is too strict and prevents the protection algorithm from achieving good resource utilization.

Some systems that ensure link and node failure recovery in a multicast session simplify the node protection to reduce the computation time. For example, in the ESHN algorithm, the *p*-cycle has to link: (1) all one-level downstream nodes of the failed node and (2) one of the upstream nodes in the light tree. Figure 6 illustrates a simple example for protecting a node using the ESHN algorithm. In this example, the failed node (or protected node) is represented by a grey circle, the source node by a green circle, the destination nodes by red circles and the multicast tree by a blue line. The *p*-cycle links the two one-level downstream nodes of the failed node (nodes belonging to the tree) and the source node (the upstream node of the failed node). Thus, the *p*-cycle satisfies the constraints and can protect the failed node. On node failure, the source node detects the failure and reroutes the multicast traffic through the protection segment (dashed green line). Although this approach relaxes the constraint imposed by NEPC, the protection capacity provided by the *p*-cycles is still not used efficiently as some *p*-cycles could protect a node without meeting the first or second constraint of this approach.

3.2 The proposed concept for node protection using *p*-cycles

In this section, we present our novel concept for protecting nodes in multicast traffic. Before presenting our concept, we will introduce some notation. Let T be a multicast light tree to be protected, s be the source node in T, N_f be an intermediate node in T, and $D = \{d_1, d_2, .., d_i\}$ be the set of destinations in T that are affected when a failure occurs on the node N_f.

Theorem 1. *A p-cycle C_j in the network can protect the node N_f if and only if there exists a protection segment $[N_a, N_e] \in C_j$ such that:*

1. *The node $N_a \in T$, the node $N_e \in T$ and $N_f \notin [s, N_a]$ where $[s, N_a]$ is the segment in T linking the source s to the node N_a.*
2. *$\forall d_k \in D$, $\exists$ a node $N_k \in [N_a, N_e]$ and $N_k \in]N_f, d_k]$, where $]N_f, d_k]$ is the segment in T linking N_f to d_k.*
3. *$N_f \notin [N_a, N_e]$.*

Proof Once a failure occurs on the node N_f, the multicast traffic is rerouted through the *p*-cycle C_j to ensure the survivability of the multicast session. The *p*-cycle must provide a protection segment to deliver the multicast content to all destinations that are affected by the failure of N_f. This segment is denoted by $[N_a, N_e]$.

First, we justify why constraint (1) is required. The extremities N_a and N_e of this segment must be in T. In fact, the node N_a is responsible for injecting the multicast traffic into the protection segment $[N_a, N_e]$ when N_f fails.

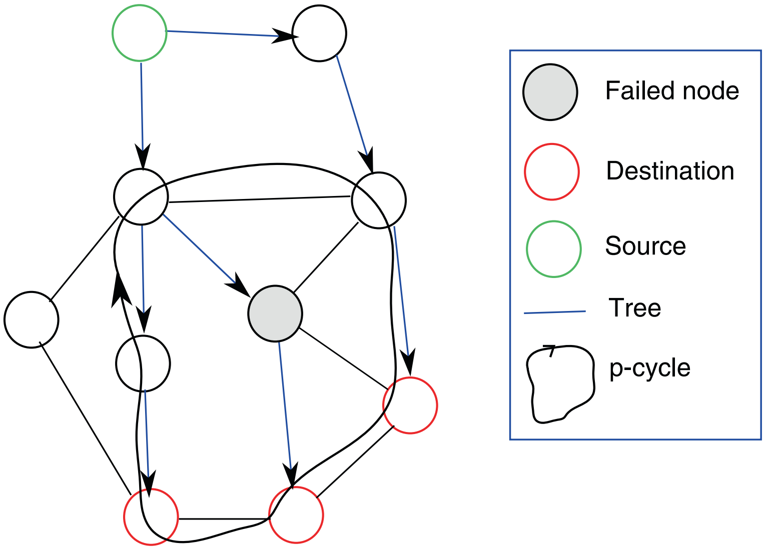

Figure 5 Node encircling *p*-cycle concept (NEPC).

In addition, N_a must not be affected by the failure of N_f, i.e., N_a must continue to receive the multicast traffic even if a failure occurs on node N_f ($N_f \notin [s, N_a]$). The node N_a must split the incoming light signal into two outgoing signals. The first is injected into the protection segment and the second is forwarded to the downstream node of N_a in the light tree T. The node N_e is the last intersection node between T and C_j.

Second, we prove the necessity of constraint (2). To ensure failure recovery, we must make sure that all destinations affected by the failure of N_f continue to receive the multicast traffic through the protection segment $[N_a, N_e]$. Two scenarios are possible for delivering the multicast traffic to an affected destination d_k. In the first, the segment $[N_a, N_e]$ carries the multicast traffic directly to d_k, i.e., the protection segment traverses the node d_k. In the second scenario, the segment $[N_a, N_e]$ carries the traffic to the destination through an intermediate node N_k. The node N_k must be an upstream node of d_k and a downstream node of N_f in the light tree. This constraint ensures that the failed node N_f does not belong to the segment $[N_k, d_k]$ of the light tree. The node N_k splits the incoming signal into two signals. The first is sent to the next node in the protection segment to ensure that the remaining

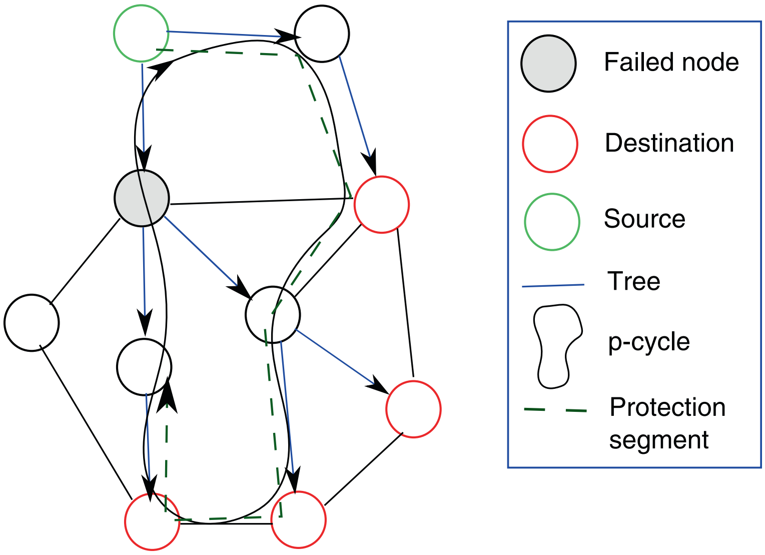

Figure 6 Protecting a node using the ESHN algorithm.

affected destinations will receive the multicast content. The second is forwarded to the downstream node of N_k in the light tree to reach d_k.

Finally, we prove that constraint (3) is necessary. We must make sure that the protection segment is not affected by the failure of N_f. Therefore, the protection segment $[N_a, N_e]$ should not traverse the node N_f.

3.3 Example

In Figure 7, we provide an example of a p-cycle that can protect the node N_f based on our concept. The set of destinations affected by the failure of N_f is $D = \{d_1, d_2, N_e\}$. This p-cycle has two original characteristics that other node protection concepts [5-7] do not have. First, it can traverse the protected node. Second, it does not have to traverse all affected destinations or neighboring nodes of the protected node. The p-cycle provides a protection segment represented with a dashed green line in the figure. The node N_a activates the p-cycle by injecting the multicast traffic into the protection segment $[N_a, N_e]$. This segment carries the traffic to d_2 through the intermediate node N_2, and to d_1 and N_e directly as it traverses them.

4 The proposed protection algorithm

In this section, we present our proposed algorithm NPCC for protecting nodes and links in the DWDM layer to give a reliable IPTV service. Our algorithm deploys the aforementioned concept for node protection using p-cycles.

4.1 Selection of candidate *p*-cycles

First, the NPCC algorithm enumerates a set of candidate p-cycles in an offline phase, i.e., before any requests have been received. Using these candidate p-cycles will considerably reduce the computation time for the algorithm. In fact, considering the total p-cycle set when selecting a new p-cycle to be established, is a very slow task, especially when the number of p-cycles in the network is high. Therefore, we select a set of candidate p-cycles to reduce the computation time for our algorithm.

We have defined a new score, the protection capacity (PC), for each p-cycle in the network. We use this score to select a candidate p-cycle set. This score is computed in advance for each unity p-cycle before routing requests. A unity p-cycle is a p-cycle in the network that reserves only one bandwidth unity (e.g., one wavelength) on each traversed link. The PC score of a unity p-cycle C_j, specified by equation (1), is defined as the ratio of the largest amount of link capacity on the network LC_j that C_j can protect over the spare capacity required for setting up C_j. $|C_j|$ is given by the number of links traversed by C_j.

$$PC(C_j) = \frac{LC_j}{|C_j|} \tag{1}$$

A p-cycle with a high PC is useful as it maximizes the amount of protected capacity while reserving less spare capacity. The l p-cycles with highest PC are selected as the candidate p-cycle set, where l is a parameter for the algorithm. The goal in selecting this set is to maximize the capacity that can be protected on the network.

4.2 Flow chart for the NPCC algorithm

Figure 8 is a flow chart for the NPCC algorithm. We will introduce some notation before describing how this

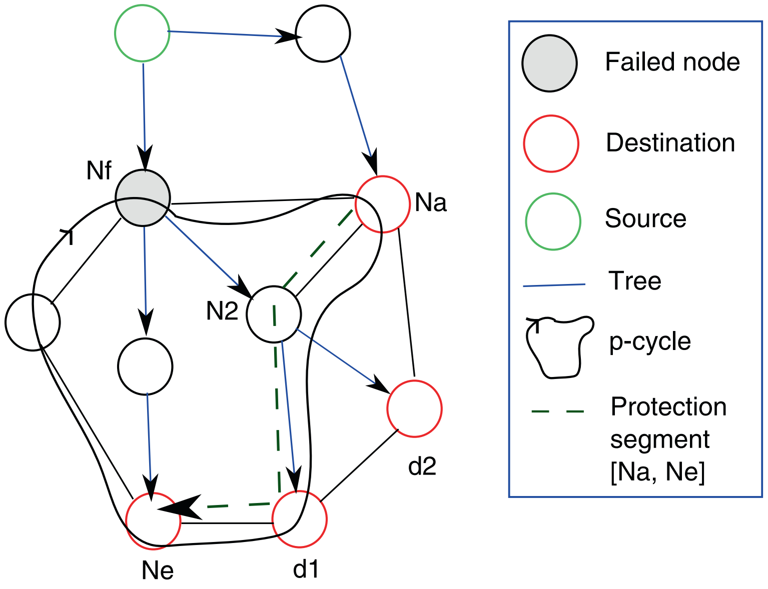

Figure 7 Protecting a node using the proposed concept.

algorithm works. Let us consider a multicast request and its corresponding light tree T. The light tree is constructed using the PIM-SSM [1] multicast routing protocol. Let L denote the set of links in T and N denote an unprotected intermediate node in T. The links in T that can be protected by the existing p-cycles in the network are removed from L and the nodes in T that are protected by the existing p-cycles are removed from N. Note that the existing p-cycles were previously established to protect other light trees in the network. If $L \neq \phi$ or $N \neq \phi$, the algorithm computes new p-cycles to protect the remaining unprotected links in L as well as the remaining unprotected nodes in N.

To select a new protecting p-cycle, the algorithm uses the unity p-cycle procedure, which is based on the efficiency score (*ES*). In this procedure, we deploy the same *ES* used in the ESHN algorithm to measure the efficiency of each p-cycle in the candidate p-cycle set. This score adapts the efficiency-ratio-based unity p-cycle heuristic algorithm (ERH) [10] to deal with node and link failures in multicast traffic. This score considers the highest number of unprotected nodes as well as the highest number of unprotected links in the multicast tree that a unity p-cycle can protect. Let C_j be a unity p-cycle in the network. The *ES* of C_j is given by equation (2), where $W_{j,L}$ is the highest number of unprotected links in L that C_j can protect, $W_{j,N}$ is the highest number of unprotected nodes in N that C_j can protect, and $\left|C_j\right|$ is the spare capacity required for setting up a unity p-cycle C_j. $\left|C_j\right|$ equals the number of links traversed by C_j.

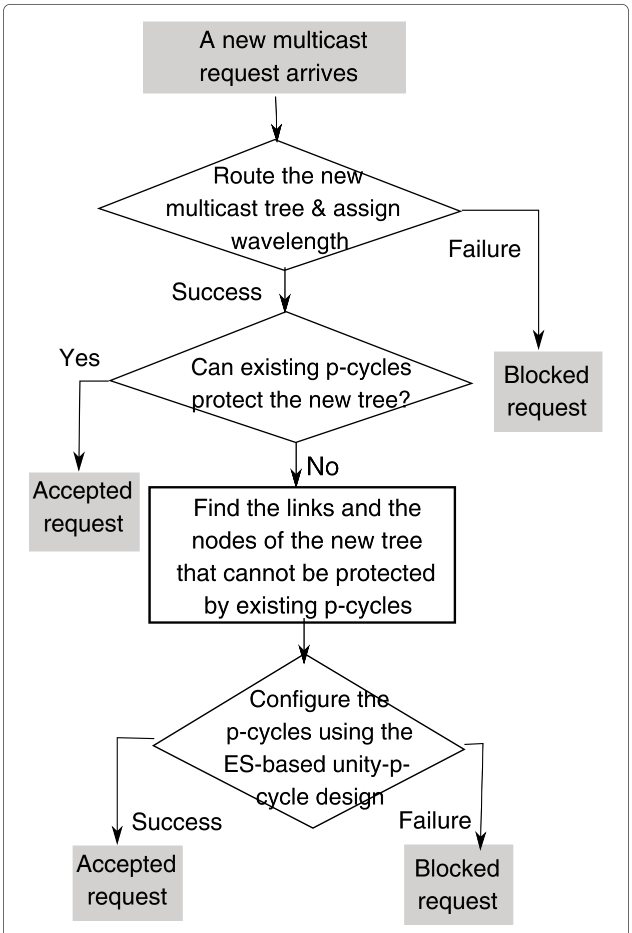
A new multicast request arrives
Route the new multicast tree & assign wavelength
Failure
Success
Yes
Can existing p-cycles protect the new tree?
Blocked request
No
Accepted request
Find the links and the nodes of the new tree that cannot be protected by existing p-cycles
Configure the p-cycles using the ES-based unity-p-cycle design
Success
Failure
Accepted request
Blocked request

Figure 8 **Flow chart for the NPCC algorithm.**

$$ES(C_j) = \frac{W_{j,L} + W_{j,N}}{\left|C_j\right|} \tag{2}$$

The *ES*-based unity p-cycle procedure calculates the *ES* of each unity p-cycle in the candidate p-cycle set and selects the p-cycle with maximum *ES*. The set of links protected by the selected unity p-cycle is removed from L and the set of protected nodes is removed from N. This process is iterated until all the links and all the nodes of T are protected, i.e. $L = \phi$ and $N = \phi$. The selected unity p-cycles are configured and the corresponding wavelengths are reserved. Note that the reserved p-cycles may serve to protect subsequent multicast requests. This is why after routing a multicast tree, we compute the set of links in L and the set of nodes in N that can be protected by the existing p-cycles in the network. Finally, the reserved capacity of an existing p-cycle in the network is released when the p-cycle does not protect any working link and nodes in the network.

5 Performance evaluation

In this section, we present the evaluation of our algorithm NPCC, which is proposed for providing a reliable IPTV service. Our method guarantees the recovery from link and node failure at the DWDM layer with a fast restoration time. We compare our algorithm with the ESHN algorithm, which was reported to be the most efficient algorithm for dynamic multicast traffic protection in terms of resource utilization efficiency and blocking probability.

In our simulation, we assumed that request arrival follows a Poisson process with an average arrival rate λ, and the request holding time follows an exponential distribution with an average holding time μ. Hence, the offered traffic load for the network is given by $\lambda\mu$.

We ran simulations on the following well-known and frequently used European optical topologies developed within the COST-266 [11] and COST-239 [12] projects:

- The COST-266 core topology [11] contains 16 nodes and 23 links, with an average nodal degree of 2.88. The total number of p-cycles in this topology is 236 (118 p-cycles in each direction).

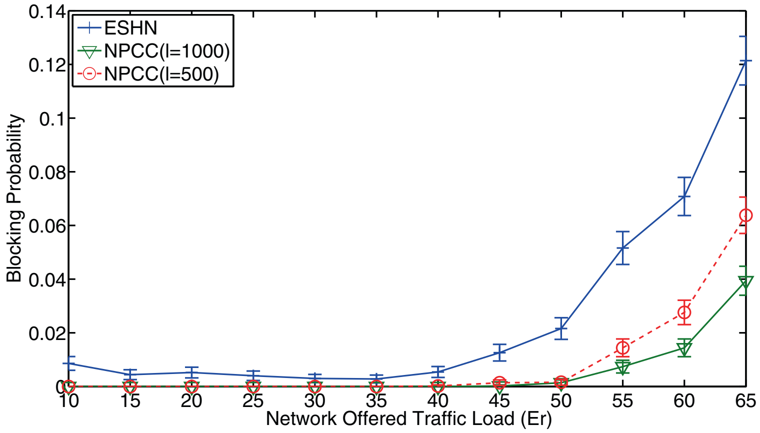

Figure 9 Comparison of the blocking probabilities *BP* for the COST-239 network.

- The COST-239 topology [12] contains 11 nodes and 26 links, with an average nodal degree of 4.727. The total number of p-cycles in this topology is 5,058 (2,029 p-cycles in each direction).

In our study, without loss of generality, we assumed that each link has two fibers. The two fibers transmit in opposite directions; 16 wavelengths are available on each fiber. The source and the destinations of each multicast session are randomly selected (uniform distribution law). We chose the number of destinations in each multicast request $D = 5$, which seems to be reasonable as the total number of nodes in the used topologies is less than 16. We compared the performance of the algorithms using the following performance criteria:

- The blocking probability (BP), which is the percentage of requests that cannot be routed or protected among the total number of requests.
- The resource utilization (RU), which is the percentage of reserved wavelengths in the network among the total number of wavelength links.

$$RU = \frac{W_R}{E \times W}$$

  where W_R is the total number of wavelength links reserved in the network, E is the number of fibers in the network and W the number of wavelengths per fiber.
- The average computation time (CT), which is required for routing and protecting a traffic request.

Performance criteria BP, RU and CT were computed according to the traffic load. For each traffic load value, 5×10^5 requests were generated. This number of requests is enough to measure BP, RU and CT with a 95% confidence interval.

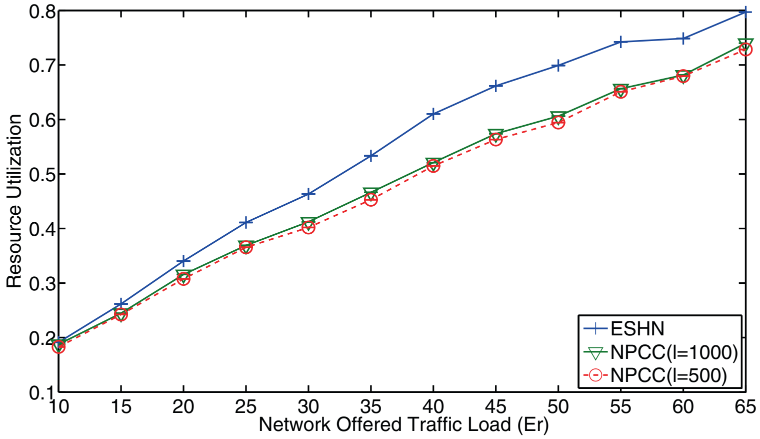

Figure 10 Comparison of resource utilization *RU* for the COST-239 network.

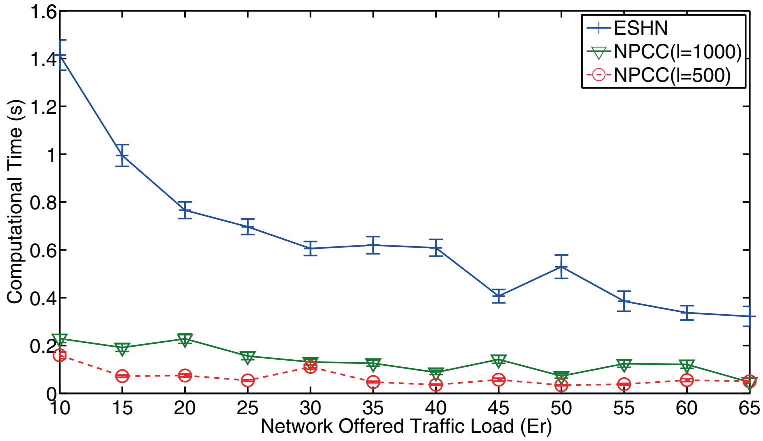

Figure 11 Comparison of the average computation time *CT* for setting up a multicast request using the COST-239 network.

First, we considered the COST-239 topology. The total number of p-cycles in this topology is 5,085. We ran the NPCC algorithm with two different values for the number of candidate p-cycles, $l = 1000$ and $l = 500$. The blocking probability measured for the COST-239 network is shown in Figure 9. For all the algorithms, the blocking probability increased when the traffic load was high. The NPCC algorithm, with both $l = 1000$ and $l = 500$, outperformed the ESHN algorithm having a lower blocking probability, especially when the traffic load was high. The NPCC algorithm with $l = 1000$ had the lowest blocking probability. When $l = 500$, the blocking probability of NPCC increased but remained lower than that of ESHN. This is because $l = 500$ is very low compared to the total number of p-cycles in the COST-239 network (5,058).

Figure 10 shows the resource utilization of the algorithms. When the traffic load increases, the wavelength percentage reserved per link is higher for each algorithm. The wavelength percentages reserved by NPCC with $l = 1000$ and NPCC with $l = 500$ are very close. This percentage is very low compared with that of the ESHN algorithm, especially when the traffic load is high. For a traffic load equal to 65 erlang, almost 70% of the wavelengths on each link are reserved for the NPCC algorithm and 80% for the ESHN algorithm.

To assess the speed of our proposed algorithm, we looked at the average computation time CT for setting up a multicast request. Figure 11 shows the value of CT for each algorithm, measured for the COST-239 network according to the network traffic load. As shown in this figure, the NPCC algorithm with $l = 500$ has a shorter computation time than the NPCC algorithm with $l = 1000$ or the ESHN algorithm. This is due to the low number of p-cycles considered for the protection ($l = 500$). The average computation time CT of the NPCC algorithm with both $l = 500$ and $l = 1000$ is very low compared with that of the ESHN algorithm. The NPCC algorithm

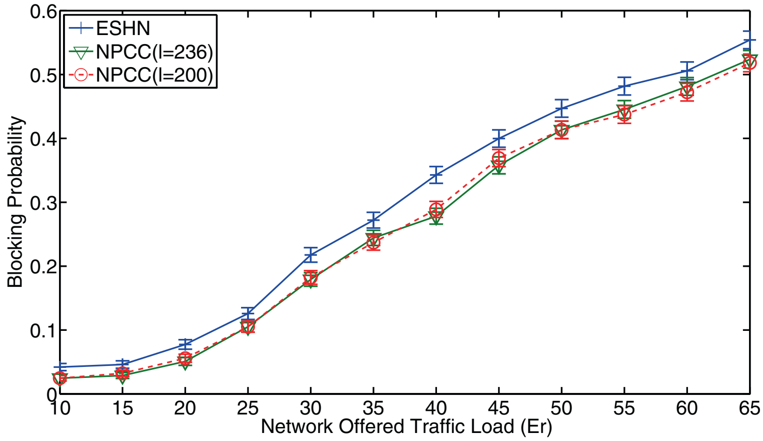

Figure 12 Comparison of the blocking probabilities *BP* for the COST-266 network.

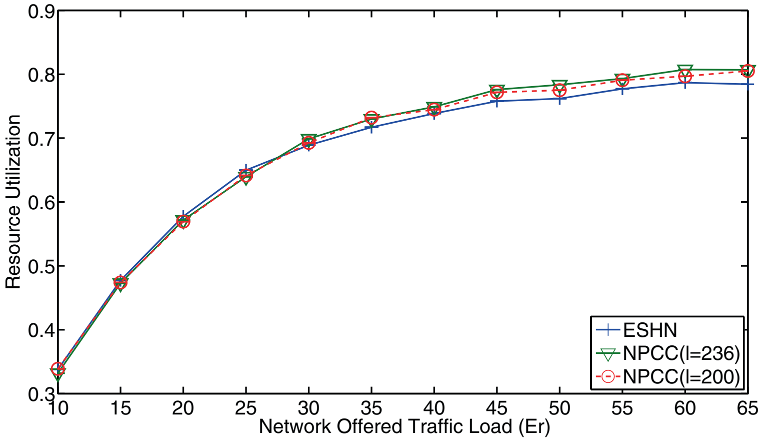

Figure 13 Comparison of resource utilization *RU* for the COST-266 network.

outperforms the ESHN algorithm in terms of blocking probability, resource utilization and computation time.

Now, we consider the COST-266 topology. The total number of p-cycles in this topology is 236. We ran the NPCC algorithm with two different values for the number of candidate p-cycles, $l = 236$ and $l = 200$. Figure 12 shows the blocking probabilities for the COST-266 network. The connectivity of this topology is very low (2.88). Therefore the blocking probabilities of the algorithms are very high compared with those for the COST-239 topology for the same network traffic load. For all the algorithms, the blocking probability increased rapidly as the traffic load increased. The ESHN algorithm has a higher blocking probability than the NPCC algorithm with $l = 236$ or the NPCC algorithm with $l = 200$. The blocking probability of NPCC with $l = 236$ and the blocking probability of NPCC with $l = 200$ were very close since the values of l were close.

Figure 13 shows the resource utilization of the algorithms for the COST-266 topology. The wavelength percentage reserved by the algorithms was almost the same. The percentage of reserved wavelengths per link increased as the traffic load increased. Note that the resource utilization of the ESHN algorithm was slightly lower than that of our algorithm NPCC when the traffic load was higher than 35 erlang. This is because the blocking probability was high. In other words, the probability of rejecting requests for ESHN increased and no resource had been reserved for the rejected requests. This reduced the resource utilization of ESHN.

6 Conclusion

In this work, we focused on the reliability of an IPTV service. First, we presented the main components of the IPTV architecture, then we discussed existing restoration mechanisms for the IP and DWDM layers. The restoration methods proposed for DWDM are more efficient and more suitable for IPTV in terms of restoration time. We also highlighted the advantage of applying p-cycle protection for producing a reliable IPTV service.

Second, we extended the concept of node protection using p-cycles to deal with multicast traffic. Our novel concept allows the protection capacity provided by a p-cycle to be used efficiently. We proposed a novel algorithm, NPCC, which uses our concept for node protection. The NPCC algorithm ensures both link and node failure recovery in the DWDM layer for dynamic multicast traffic. This algorithm reduces the computation time for setting up a multicast traffic request by enumerating a set of candidate p-cycles based on the *PC* score.

Finally, we compared our proposed algorithm with the ESHN algorithm, which was reported to be the most efficient algorithm for node and link failure recovery for dynamic optical multicast traffic. Extensive simulations showed that the NPCC algorithm had a lower blocking probability and outperformed the ESHN algorithm in terms of resource utilization and computation time.

Abbreviations

BP: blocking probability; CT: computation time; DSLAM: digital subscriber line access multiplexer; DWMD: dense wavelength division multiplexing; ERH: efficiency-ratio-based unity p-cycle heuristic algorithm; *ES*: efficiency score; IGP: interior gateway protocol; IP: Internet protocol; IPTV: television over IP; ISM: internet standard multicast; LSP: label-switched path; MPLS: multi-protocol label switching; NEPC: node encircling p-cycle concept; NPCC: node and link protecting candidate p-cycle based algorithm; PIM-SSM: protocol-independent multicast source-specific mode; QoS: quality of service; RU: resource utilization; SHE: super headend; SSM: source-specific mode; VHO: video hub office; VoD: video on demand; VSO: video serving office.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

AF carried out the study of the IPTV service reliability, proposed the NPCC algorithm, performed the simulations and wrote the manuscript. BC and SL participated in discussing the idea of algorithm and the selection of parameters values and reviewed the manuscript. All authors read and approved the final manuscript.

Received: 8 April 2013 Accepted: 28 January 2014
Published: 4 March 2014

References

1. Bhattacharyya S (2003) An overview of source-specific multicast (SSM). IETF RFC 3569
2. Doverspike R, Ramakrishnan KK, Chase C (2010) Structural overview of ISP networks In: Kalmanek C, Misra S, Yang R (eds) Guide to reliable internet services and applications, Computer Communications and Networks. Springer, London, pp 19–93
3. Grover WD, Stamatelakis D (1998) Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration. In: Proceedings of IEEE International Conference on Communications (IEEE ICC), vol 1, pp 537–543
4. Clouqueur M, Grover WD (2005) Availability analysis and enhanced availability design in p-cycle-based networks. Photonic Netw Commun 10(1): 55–71
5. Doucette J, Giese PA, Grover WD (2005) Combined node and span protection strategies with node-encircling p-cycles. In: Proceedings of the workshop on design of reliable communication networks (DRCN). Ischia (Naples), Italy, pp 213–221
6. Cisco Systems (2006) White paper on optimizing video transport in your IP triple play network. http://www.cisco.com.
7. Zhang F, Zhong WD (2009) Performance evaluation of optical multicast protection approaches for combined node and link failure recovery. J Lightw Technol 27(18): 4017–4025
8. Singhal NK, Sahasrabuddhe LH, Mukherjee B (2003) Provisioning of survivable multicast sessions against single link failures in optical WDM mesh networks. J Lightw Technol 21(11): 2587–2594
9. Zhang F, Zhong WD (2009) *p*-cycle based tree protection of optical multicast traffic for combined link and node failure recovery in WDM mesh networks. IEEE Commun Lett 13(1): 40–42
10. Zhang ZR, Zhong WD, Mukherjee B (2004) A heuristic method for design of survivable WDM networks with p-cycles. IEEE Commun Lett 8: 467–469
11. Maesschalck SD, Colle D, Lievens I, Pickavet M, Demeester P, Mauz C, Jaeger M, Inkret R, Derkacz BM (2003) Pan-European optical transport networks: an availability based comparison. Photonic Netw Commun 5(3): 203–226
12. Batchelor P, Daino B, Heinzmann P, Weinert C, Spath J, Van Caenegem B, Hjelme DR, Inkret R, Jager HA, Joindot M, Kuchar A, Le Coquil E, Leuthold P, De Marchis G, Matera F, Mikac B, Nolting HP, Tillerot F, Wauters N (1999) Ultra high capacity optical transmission networks. Final report of Action COST 239

doi:10.1186/1869-0238-5-3
Cite this article as: Frikha *et al.*: **Applying *p*-cycle protection for a reliable IPTV service in IP-over-DWDM networks.** *Journal of Internet Services and Applications* 2014 **5**:3.