Report of IPLU-II project case study: reliability of MPLS multicast

Pirkko Kuusela and Ilkka Norros
VTT, Technical Research Centre of Finland

P.O.Box 1000
FI-020044 VTT, Finland

May 7, 2008

Abstract

This case study is motivated by the task of reliable data delivery in a MPLS core network. We compare two principles of delivering data in an example network in terms of tolerance to link and/or node failures. This analysis can be used in designing the topology of the a future network and in selecting the actual data delivery method.

1 Framework of the case study

This second case study in IPLU - II project focuses on reliability of MPLS multicast in a (small) core network.

The aim is to demonstrate how classical reliability methods can be applied in IP networks. In particular, we illustrate how they can be used, in comparing the failure tolerance of different approaches of sending MPLS traffic, in the process of designing of a network and selecting the actual implementation.

2 The concrete task

We formulate the concrete task in delivering the data:

Task 1. Deliver data from source S to destinations $D_1, \ldots, D_6$ in the network illustrated in Figure 1. If all non-failed destinations receive the data, the task is considered succesfull.

In this case study we compare two principles of delivering data in terms of their tolerance to link and node failures.

Solution principles to compare are:

- Multicast data from S twice in the network using two independent (i.e., no links in common) spanning trees.
- Multicast data from S only once, and use fast rerouting, if link or node failures occur.

2.1 Topology

The comparison of solution principles is carried out in the topology shown in Figure 1. It is a generic example of a possible core network in Finland. It is also simplified in a sense that there could be "strings of nodes", whenever the degree of a node, i.e., the number of links attached to a node, seen as an undirected graph, is 2. For example, a path from D_2 to D_6 and to D_4 could contain more nodes like node D_6, but essential properties of the topology would not change. Only the event of failure in the whole path would have more options where the actual link or node failure would take place.

Nodes are connected by using two directed links, one in each direction. Links connecting the nodes are considered independent and a failure of a directed link does not lead to restrictions on using the link going to the opposite direction - unless there is a particular failure event preventing also that link from functioning. This setting corresponds to a situation in which all actual links would be doubled, i.e., reliability of the network is particularly high. This is, in fact, in the scope of this case study, as it is motivated by the problem of delivering television programs over the IP network.

In a normal network, without extensive doubling, a failure in a directed link may in fact prevent the oppositely directed link from being used. Not because of a failure in the link itself, but because of the physical implementation of a link card.

3 Results

As the approach relies heavily on utilizing spanning trees, we first review what can be said about them – without actually computing any spanning trees.

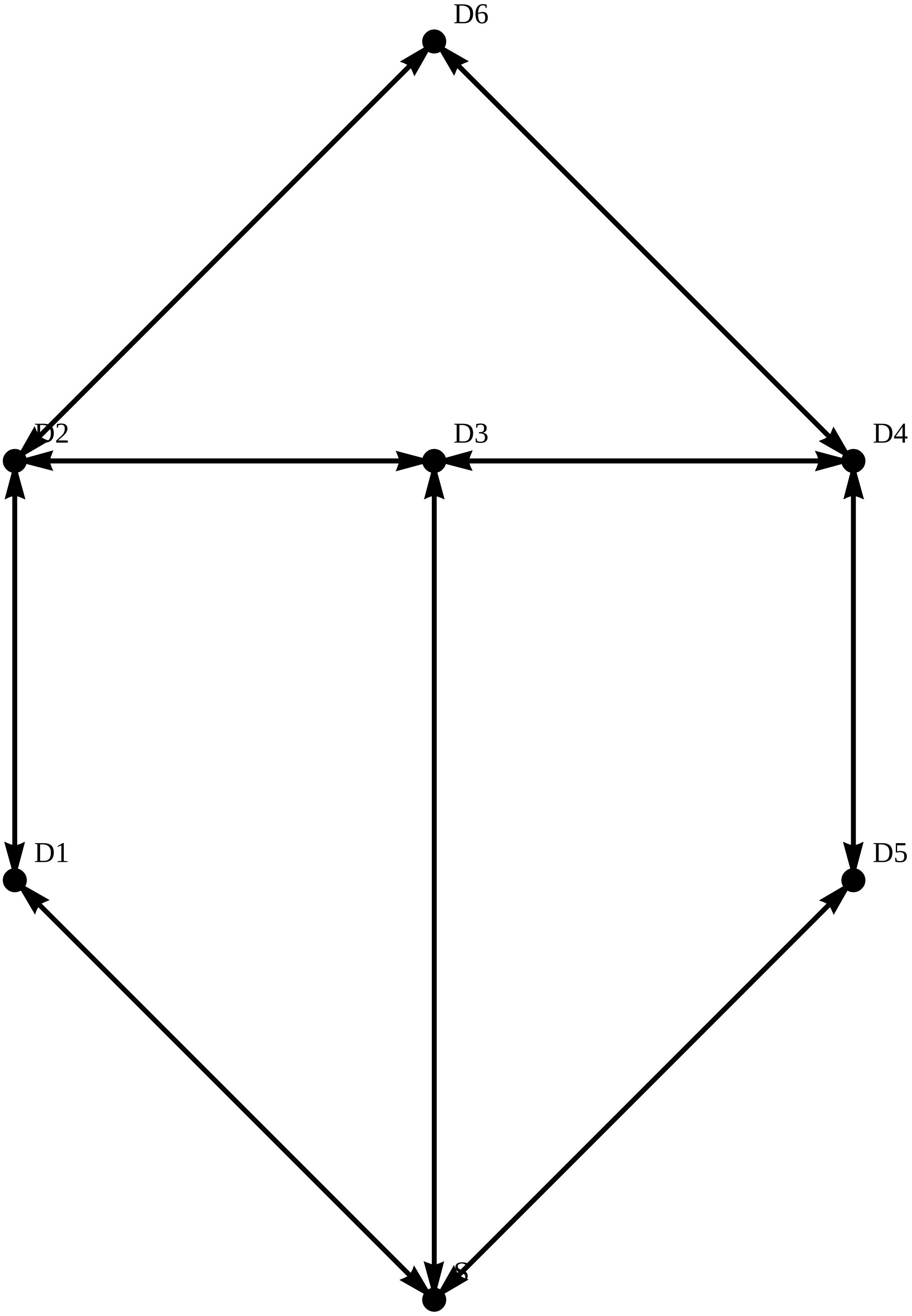

Figure 1: The topology used in the case study.

3.1 About the number of spanning trees

We first quantify the number of nonidentical spanning trees available for the chosen topology.

The matrix tree theorem states that the number of nonidentical spanning trees of a graph G is equal the determinant of any principal minor (obtained by removing one row and one column) of the Laplacian matrix $L(G)$ of the graph G.

The topology in Figure 1 gives rise to a undirected graph G. The Laplacian matrix $L(G)$ contains the degree of each node (number of nodes adjacent to it) in the diagonal. Entry in position i, j, $i \neq j$ is -1, if there is a link from node i to node j, and 0 otherwise. All row and column sums equal to zero. In our case the Laplacian matrix (using order S, $D_1, \ldots, D_6$) becomes

$$\begin{pmatrix} 3 & -1 & 0 & -1 & 0 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 & 1 \\ -1 & 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 2 \end{pmatrix} \tag{1}$$

and the determinant of the principal minor, obtained by removing the first row and column, equals to 50. That is, there are exactly 50 nonidentical spanning trees when the topology in Figure 1 is seen as an undirected graph.

Any undirected spanning tree gives rise to a directed spanning tree with root S. However, when all links are bidirectional, there is a bijection between the set of spanning trees of undirected graph G and the set of spanning trees of directed graph G_d, both given by the topology in Figure 1. This implies that also the directed graph G_d has exactly 50 nonidentical spanning trees rooted at S.

3.1.1 J.Edmonds result

There are some theoretical results on the number of link-disjoint spanning trees (typically called edge-disjoint or independent spanning trees) in the context of general graph theory as well as for applications in multicasting and resilient networks. To the best of our knowledge there is no general theorem giving exact criteria when a directed graph has k pairwise link-disjoint spanning trees. Result by J. Edmonds gives a partial answer to that problem:

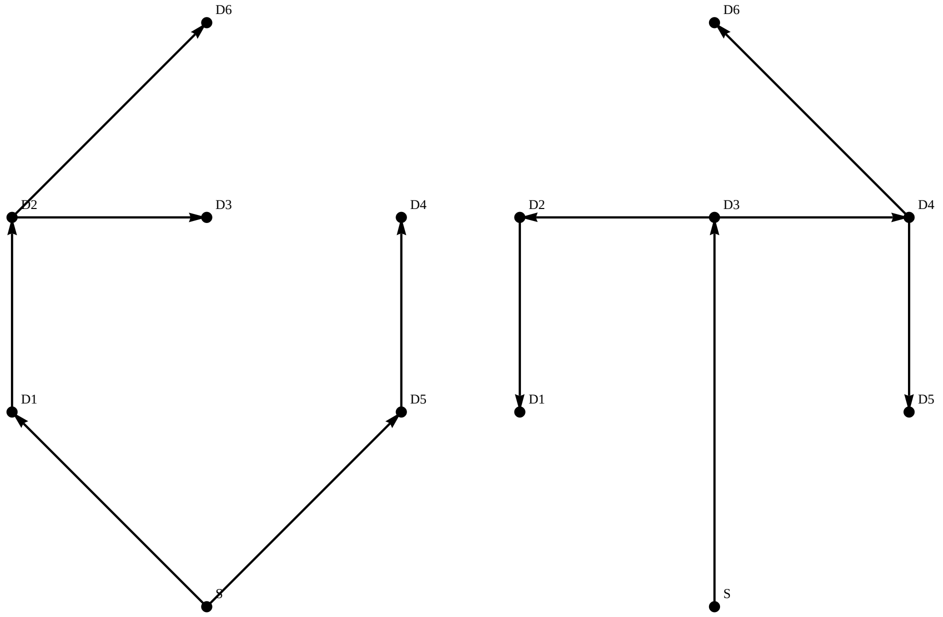

Figure 2: A natural pair for link-disjoint spanning trees.

Result 1 (J.Edmonds result [1])**.** There exists k mutually link-disjoint spanning trees rooted at $r \iff$ there exists at least k link-disjoint paths from r to any other node v.

When we apply the above result to the directed graph G_d given by the chosen topology, we find that there exits two link-disjoint spanning trees and it is not possible to have three such spanning trees.

3.2 Multicast twice

The first approach was to multicast the data twice in the network by using two independent spanning trees. The results given below clearly depend on the spanning trees chosen and we illustrate it by taking two pairs of spanning trees. The first pair is a natural choice, the paths have length at most 3 links. The second pair is less natural, and the longest path contains 5 links. These pairs of trees are illustrated in Figures 2 and 3.

3.2.1 Tolerance to 1-link failure

As the data is delivered twice in the network by using spanning trees that do not share any links, this approach can tolerate any one link failure.

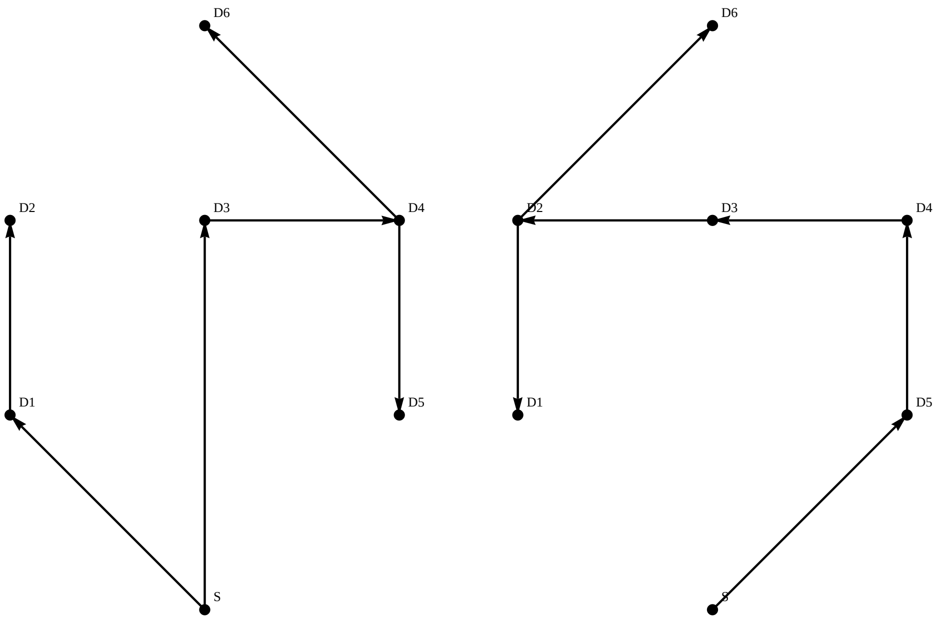

Figure 3: A second pair of link-disjoint spanning trees.

3.2.2 Tolerance to 2-component failures

There are total of 276 possible 2-component failures. (Node failure is counted as one failure although it prevents all links attached to that node from being used.) All these failures have been analyzed: first failed components are removed and then it is checked weather the data can be delivered to all non-failed destinations using the network that remains and the predefined spanning tree pair.

Natural spanning tree pair

When we used the natural choice for the spanning tree pair illustrated in Figure 2, we obtained following results:

- In 42 cases (15%) the 2-component failure spreads, i.e., some non-failed nodes could not receive the data.
- The maximal set of non-reachable nodes was $\{D_1, D_2, D_3, D_6\}$, that could result in 4 different 2-failures. In all failure patterns destinations D_1 and D_3 had to be involved, either one or both were failed or a link to D_1 or D_3 was failed. For failures where 3 or fewer destinations did not receive the data, there were not such clear patterns as above.

- In 12 cases (4%) the 2-component failure spreads so that 3 or 4 nodes were not reached.

- There are two nodes, namely D_4 and D_5 that are protected from all 2-component failures (other than those failures in which nodes are down by themselves). That is to say that no 2-component failure can prevent data delivery to these two nodes. As the topology is symmetric across the vertical axis, one can reflect the given pair of spanning trees to ones that protect nodes D_1 and D_2 from any spread of a 2-component failure. This is a particularly usefull observation, if some major nodes can be made very resilient.

Second pair of spanning trees

As the results in this section depend on the chosen spanning trees, we illustrate how the pair depicted in Figure 3 performs. Corresponding results are:

- In 82 cases (30%) the 2-component failure spread. No clear pattern was detected contrary to the above case.

- In 22 cases (8%) the 2-component spread so that 3 or 4 nodes were not reached.

- There are *no* nodes that are protected from spreading of 2-component failures.

Comparison of the tree pairs

When the failure cases were analyzed, it is clear the the natural pair considered first outperforms the second pair — when looking at plain numbers the natural pair is twice as good as the second pair, and it has additional preferable node protection properties not found in the second pair. Hence any pair of trees should not be taken, but in this view, the pair that appears natural, is also the best choice.

3.3 Multicast once

Second solution principle for Task 1 is to deliver data only once in the network and then to use fast rerouting, if a link or node failure occurs. Both Cisco and Juniper offer their own versions for fast rerouting in MPLS networks, see [2, 3] and the RFC 4090 [4].

We quote the basic idea from [4]:

> This document extends RSVP [RSVP] to establish backup label-switched path (LSP) tunnels for the local repair of LSP tunnels. This extension will meet the needs of real-time applications such as voice over IP, for which user traffic should be redirected onto backup LSP tunnels in 10s of milliseconds. This timing requirement can be satisfied by computing and signaling backup LSP tunnels in advance of failure and by re-directing traffic as close to the failure point as possible. In this way, the time for redirection includes no path computation and no signaling delays, including delays to propagate failure notification between label-switched routers (LSRs). ...

The first approach was to find local repairs for failed links or nodes. However, core networks in Finland, as well as the network considered here, tend to be so small, that it not possible to talk about local repairs - backup routes circle about 1/3 of the network, if not even more. Nice examples given in documents to demonstrate the technique were far from practise.

In general, this idea of local repair is tempting when one consideres only one failure. However, two failures are more likely to be close to each other than to be far from each other - as one failure will increase traffic in functioning links and give rise to conditions that favour further failures. Or weather conditions may damage more than one component is some region. Then there would be a fix-of-a-fix scenario, that is more complicated than the case of a single failure.

The above observations motivated the approach of using global repair to be discussed next.

3.3.1 Spanning tree approach

As we pointed out that small networks do not have local behaviour, we looked for a global repair approach at the principle level. Moreover, our main motivation is to find out how different approaches can tolerate 2-component failures. When there are two (or more) failures in the network, the local repair techniques can be problematic, if they influence to each other - that is, if there is not enough locality in the network.

The global repair approach is the following: Data is delivered into the network by using some spanning tree. If there are link or node failures that influence to the used spanning tree, a new spanning tree is selected. All spanning trees are computed in advance. For any 1-element or 2-element failure event, there is a predefined spanning tree to be used.

In this report, we do not consider probing the network, signalling and other issues that need to be solved before the above approach can be used.

3.3.2 Tolerance to 1- element failure

Contrary to the approach of delivering the data twice in the network, this spanning tree approach requires switching to the use of a new spanning tree, if a failure occurs in links used in the original spanning tree. Whenever one link fails, the connectivity of the network does not decrease and it is possible to find a new spanning tree. If a node fails (other than the source S), also a new spanning tree can be found.

3.3.3 Tolerance to 2-component failures

If a graph is connected, it has a spanning tree. Hence, our task is to find all those 2-element failures that result to a loss of connectivity. The cases in which two links fail can be detected by inspecting the topology. Only D_1, D_5 and D_6 can be isolated by removing the incoming links to these nodes. After that the cases in which two nodes or a link together with a node fail needed to be checked by using Mathamatica. Failed components were removed and connectivity of the remaining network was tested. There were only 8 cases out of 276 failure combinations ($\leq 3\%$) in which a spanning tree can not be found. These cases are listed in a table below

what fails:		can't
links	nodes	reach:
$S \to D_1$, $D_2 \to D_1$		D_1
$S \to D_5$, $D_4 \to D_5$		D_5
$D_2 \to D_6$, $D_4 \to D_6$		D_6
	D_2, D_4	D_6
$D_4 \to D_6$	D_2	D_6
$S \to D_1$	D_2	D_1
$D_2 \to D_6$	D_4	D_6
$S \to D_5$	D_4	D_5

Note that the only non-failed destination that can not be reached is always D_1, D_5 or D_6. This means that destinations D_2, D_3 and D_4 are protected.

4 Comparison of methods

We now list some remarks for comparing the two approaches analyzed in this report in terms of resistance to link and/or node failures.

- Advantages of multicast twice approach:

    1. Once this delivery has been set up, it requires no additional actions in case of component failures.
    2. It can tolerate any one component failure.
    3. With proper selection of spanning tree pairs, some nodes can be protected from the spredding of the failure event.
    4. It is likely that no unforeseen technical problems arise.

- Disadvantages of multicast twice approach:
    1. All data is delivered twice – usage of the network capacity is larger than necessary.
    2. About 15% of 2-element failures leads to failure in delivering the data to some non-failed destination(s).

- Advantages of multicast once approach:
    1. Superior in resistance to 2-element failures. Only less than 3% of 2-element failures lead to failure in reaching one (either D_1, D_5 or D_6) non-failed destination.
    2. Three destinations (D_2, D_3 and D_4) are protected from the spreading of 2-element failures.
    3. Data is delivered only one – optimal use of network capacity.

- Disadvantages of multicast once appraoch:
    1. Failure events and corresponding spanning trees need to be precalculated and stored. In this case, there would be 268 (=276 - 8) failure scenarios and 50 nonidentical spanning trees.
    2. Also one element failures require actions, if that network element is used in the current spanning tree.
    3. There is a need for detection of failures and signalling when spanning tree is changed.
    4. If the change of spanning trees can be done for real-time traffic remains an open problem.
    5. There are likely to be various technical problems in implementation.

References

[1] J. Edmonds, “Edge-disjoint branchings,” in *Combinatorial Algorithms*, R. Rustin, Ed. Algorithmics Press, 1972, pp. 91–96.

[2] Cisco Systems Inc., “Advanced topics in MPLS-TE deployment,” white paper, 2002. [Online]. Available: http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/mwglp_wp.pdf

[3] JUNOS Internet Software, “MPLS fast reroute solutions network operations guide,” January 2007. [Online]. Available: http://www.juniper.net/techpubs/software/nog/nog-mpls-frr/download/nog-mpls-frr.pdf

[4] “RFC 4090 - fast reroute extension to the RSVP- TE for LSP tunnels,” Network Working Group, Internet draft, May 2005, editors P. Pan, G. Swallow, A. Atlas. [Online]. Available: http://www.faqs.org/rfcs/rfc4090.html