



Technical Report – D41

Design of study case solutions

Abstract:

This document describes problems, approaches and solutions in the area of network management, from the TIGER2 point of view. Higher manageability issues are tackled in six areas. 1) The hitless maintenance approach eases the work of the human operator by mastering maintenance tasks through automated processes. 2) The approach for energy aware routing exploits ideas of flow aggregation, and aims at finding the optimal type and amount of resources for balanced network performance and power efficiency. 3) OPEX and CAPEX awareness through a new networking architecture is described through the LOCARN concept of TIGER2. 4) Further self-optimized network resource allocation can be achieved by fine-tuning the characteristics of the Spanning Tree Protocol. 5) Inter-domain traffic engineering problems are proposed to be tackled through sharing the intelligence between control planes. 6) Finally, more knowledge could also be gathered through extensive traffic monitoring, which allows traffic mix and traffic matrix analysis (among others) to support decision making in order to actuate proper modifications at the network node level for optimal operation.

Editor:

Pal Varga – AITIA International

Authors:

Samir Ghamri-Doudane, Laurent Ciavaglia – Alcatel-Lucent Bell Labs France

Dario Rossi – Telecom ParisTech

Lluís Fàbrega i Soler - Universitat de Girona

Rémi Clavier – France Telecom Orange Labs

Csaba Simon – Budapest University of Technology and Economics

Pal Varga – AITIA International

Reviewers:

Samir Ghamri-Doudane, Laurent Ciavaglia – Alcatel-Lucent Bell Labs France

Date of publication: October 2010



Table of Contents

1	<u>EXECUTIVE SUMMARY.....</u>	<u>5</u>
2	<u>HITLESS MAINTENANCE</u>	<u>7</u>
2.1	<u>Problem Statement</u>	<u>7</u>
2.2	<u>OSPF Graceful Restart.....</u>	<u>8</u>
2.3	<u>Maintenance Process</u>	<u>9</u>
2.4	<u>Planning Algorithm and Metrics.....</u>	<u>10</u>
2.5	<u>Metric Choice</u>	<u>11</u>
2.6	<u>Preliminary Evaluation Results</u>	<u>12</u>
2.7	<u>Synthesis and Future Steps</u>	<u>13</u>
3	<u>ENERGY-AWARE ROUTING: A FIXED CHARGE NETWORK FLOW FORMULATION</u>	<u>14</u>
3.1	<u>Problem Formulation</u>	<u>14</u>
3.2	<u>Preliminary Results</u>	<u>16</u>
4	<u>LOCARN.....</u>	<u>18</u>
4.1	<u>Problem Statement</u>	<u>18</u>
4.2	<u>Solution</u>	<u>19</u>
4.2.1	<u>FRAME</u>	<u>19</u>
4.2.2	<u>NODE</u>	<u>20</u>
4.2.3	<u>LOCARN FRAME PROCESSING</u>	<u>23</u>
4.2.4	<u>OTHER ISSUES</u>	<u>23</u>
4.3	<u>Preliminary evaluation results</u>	<u>24</u>
4.3.1	<u>LOCARN PoC IMPLEMENTATION</u>	<u>24</u>
4.3.2	<u>LOCARN COMPLIANCY WITH AUTONOMIC PROPERTIES</u>	<u>25</u>
5	<u>SELF-OPTIMIZATION OF NETWORK RESOURCE ALLOCATION IN MSTP</u>	<u>26</u>
5.1	<u>Optimization of network resource allocation.....</u>	<u>26</u>
5.2	<u>The proposed scheme for self-optimization.....</u>	<u>26</u>
5.3	<u>Optimization model for MSTP based technologies.....</u>	<u>27</u>
5.4	<u>Evaluation results</u>	<u>30</u>

6	<u>INTER-DOMAIN TRAFFIC ENGINEERING FOR BALANCED NETWORK LOAD</u>	<u>32</u>
6.1	<u>Network model</u>	<u>32</u>
6.1.1	REFERENCE NETWORK MODEL	32
6.1.2	CORE NETWORK MODELS	33
6.1.3	INVESTIGATED NETWORK TOPOLOGIES	33
6.2	<u>The proposed solution: Inter-domain TE cooperation</u>	<u>34</u>
6.3	<u>Proposed simulation model</u>	<u>35</u>
6.3.1	TRAFFIC MODEL	35
6.3.2	SPANNING TREES IN THE AGGREGATION DOMAIN	36
6.4	<u>Planned work</u>	<u>36</u>
7	<u>TRAFFIC ANALYSIS FOR THE KNOWLEDGE PLANE</u>	<u>37</u>
7.1	<u>Problem Statement</u>	<u>37</u>
7.2	<u>Solution</u>	<u>38</u>
7.2.1	THE MONITOR PLANE	38
7.2.2	BASIC FUNCTIONS OF THE PROBES	39
7.2.3	TRAFFIC PROCESSING	40
7.2.4	DECISION MAKING	41
7.3	<u>Preliminary Results</u>	<u>43</u>
7.3.1	TRAFFIC MATRIX CALCULATIONS	43
7.3.2	TRAFFIC MIX STATISTICS	44
8	<u>CONCLUSION</u>	<u>46</u>
9	<u>ANNEX A: GENERIC LOCARN FRAME FORMATS</u>	<u>48</u>
10	<u>REFERENCES</u>	<u>49</u>

1 Executive Summary

One of the main objectives of the TIGER2 WP4 work is to identify and solve concrete operational scenarios related to network control and management. The first WP4 deliverable (D40) has discussed the rationale behind the TIGER2 approach, addressed its implementation, and detailed the list of study cases that have been identified for further investigation [8]. Then, the goal of this deliverable is to present, in details, the solutions that have been proposed to solve these relevant study cases, and also to outline their use of self-* paradigms. Finally, an updated version of this deliverable (D41-2) is planned and shall provide detailed evaluation results regarding these proposals.

Therefore, there are altogether six areas of higher network and service manageability addressed in this document.

The Hitless Maintenance approach proposes a solution for the automation and optimized orchestration of maintenance operations for IP/MPLS networking elements, based on an adaptive and fully-distributed planning process. It allows the operator to focus on productive activities, since he/she is greatly relieved from tedious maintenance tasks.

Energy aware routing is the second topic of this document. One of the most common practices for acting in a *green* fashion in network dimensioning consists in *resource consolidation*. This technique aims at reducing the energy consumption due to devices underutilized at the considered interval of time. The solution aims for an optimal balance, where the required level of performance will still be guaranteed, but using an amount of resources that is dimensioned for the current network traffic demand rather than for the peak demand (or more). Flow aggregation may be achieved, for example, through a proper configuration of the routing weights.

Higher manageability should be considered from the OPEX and CAPEX point of view, since theoretically feasible solutions often mean practically unrealistic possible investments. LOCARN (Low Opex and Capex Architecture for Resilient Networks) is an imaginative, new network paradigm, which aims to explore two concepts (auto-forwarding and enhanced broadcast) in order to increase as much as possible network simplicity and hence increase savings in both OPEX and CAPEX domains. Chapter 4 of this document aims to offer a sufficient enough in-depth view of the LOCARN network concepts with the objective to prepare its Proof-Of-Concept implementation.

Self-optimized network resource allocation in MSTP (Multiple Spanning Tree Protocol) takes a deep dive into management algorithms. Self-optimization of network resource allocation aims to provide automatically without human intervention an optimal allocation. Periodically an optimization model is run, then the calculated optimal resource allocation is compared with the actual one, and if their difference is large enough, reconfiguration of network resources is initiated. The goal of the optimization model can be either minimizing the number of allocated spanning trees or the amount of allocated capacity.

Another aspect of network manageability appears at inter-domain areas. Chapter 6 of this document proposes new traffic engineering methods for balanced network load. The main idea of the solution is to use shared intelligence between control planes, where the core intra-network functions are unchanged and only the inter-network control planes co-operate which enhances the performance.

The knowledge plane concept allows self-management networks and networking services by applying sensors, processing their data, making (probably fine-tuning) decisions based on the processing results, and applying the corrective steps at the network or servicing node level. The elementary steps toward the solution consist of gaining knowledge about network status and traffic characteristics is to gather and process such data, which then provide a basis to trigger corrective actions. The Monitor plane concept, and proposed physical equipment (including SGA10GED, a network interface card developed inside TIGER2), is introduced in in the final technical chapter.

Note: The D40 deliverable contains the description of an additional study case entitled: "Adaptive control of Path Computation Elements" [8]. This study case shall be considered only as an example of a concrete operational scenario related to network control and management. It has not been investigated further due to budget restrictions in Spain. On the other hand, the D40 deliverable did not initially provide the description of the "Traffic analysis for the knowledge plane" study. It has been identified a posteriori since it is a common requirement shown by several of the initial WP4 study cases.

2 Hitless Maintenance

The maintenance of communication systems is a critical operation which monopolizes human and network resources. The management of multiple, parallel maintenance jobs is a complex task that can generate faults and undesirable service interruption. In the context of IP/MPLS networks, Graceful Restart mechanisms allow, under strict conditions, the maintenance of a single router without impacting its forwarding plane. Still, the network-wide coordination of the routers restarts is an unresolved problem. To solve this issue, the "Hitless maintenance" study case has been defined [8]. It proposes a solution for the automation and optimized orchestration of maintenance operations, based on an adaptive and fully-distributed planning process. The operator is then relieved from tedious maintenance tasks and is only responsible for setting performance objectives and assessing progress reports.

2.1 Problem Statement

Maintenance operations are frequent tasks in telecommunication networks, and cover a large panel of hardware and software interventions. Although necessary, these operations result, most of the time, in partial or complete service interruption, which is detrimental to the end user and to the network operator.

Additionally, network-wide maintenance activities can rapidly become a complex task preempting precious human skills and time. They are error-prone as well, and one of the main causes of network downtime [1]. As a result, the maintenance process is a major generator of costs, and constitutes a relevant example of the increasing complexity in operating today's networks.

Consequently, a key challenge is to relieve human operators from tedious maintenance tasks, to minimize service disruptions, and hence to drive down the costs. This is part of the Opex challenge that is pushing towards the introduction of trusted autonomic processes within network operations.

Technically speaking, the challenge is to provide network operators with a solution for the automation and optimized orchestration of maintenance operations, with minimal impact on the network. The operator is thus only responsible for the publication of maintenance targets, the setting of high level objectives and the track of progress reports.

Within this context, we focus here on the maintenance of the control plane functionalities in IP/MPLS transport networks by capitalizing on the standardized graceful restart mechanisms [2][3]. These mechanisms are enabled by the physical separation of the control and data plane features in today's routers. Indeed, it is possible to restart the control software while the data plane functionalities are still up and running. Therefore, the router can be kept within the forwarding path during this maintenance period. The OSPF graceful restart mechanism [2] has thus no impact on the traffic and aims at minimizing the need for routing re-convergence and, by the same, at optimizing the usage of the network resources. It can be used to implement several maintenance activities such as: software upgrades or patching, error corrections, re-initializations and even hardware interventions.

However, in case of network-wide maintenance activities, human operators are still required in order to manually start and drive the maintenance process of each targeted component, and this is generally done in a sequential way. It is hence very consuming in terms

of time, manpower and costs. This study case proposes a solution that completely automates the planning and implementation of such maintenance activities through distributed processes. Nevertheless, the operator remains in the network control loop as he defines the performance objectives and can continuously check their achievement.

2.2 OSPF Graceful Restart

Graceful Restart (GR) is a set of mechanisms developed at the IETF for signaling and routing protocols such as LDP [3], and OPSF [2]. The OSPF-GR technique enables to restart the routing process on an individual node or interface without interrupting the overall network service. This mechanism is also called “non-stop forwarding” since the data plane of the restarting node can run, for a short period of time, out of synchronization from the control plane and therefore can continue to forward the traffic normally through the node. This behavior is feasible thanks to the physical separation of the control and forwarding functions as it is the case in most of today’s carrier-class routers.

Usually, when a given router interrupts its routing process, the neighboring nodes trigger automatically a re-convergence of the topology to reflect the router unavailability. OSPF-GR adapts this default behavior of the OSPF protocol and allows the restarting router to remain visible thanks to the coordinated action of its neighbors. However, in order to work properly, the network topology must remain stable and the restarting router must maintain the integrity of its forwarding table across the graceful restart procedure. In case network topology changes are detected, the normal OSPF restart will take over for safety reasons (routing loop avoidance).

The OSPF router performing the maintenance operation is designated as the Restarting router whereas its directly connected neighbors are called Helper routers. These nodes must cooperate in order to achieve a graceful restart. RFC3623 [2] details the conditions and responsibilities for a router running in helper mode, in addition to the operation of the restarting router. The overall process is summarized in Figure 1.

The main constraint and particularities that characterize the functioning of the OSPF Graceful Restart mechanism are:

- A single router can simultaneously serve as a helper for multiple restarting neighbors.
- A router cannot perform a graceful restart while it acts as helper for its neighbor(s).
- If the restarting router is normally reloaded and flushes its Grace-LSAs within the grace period, the graceful restart is considered as successfully terminated.
- However, if the grace period expires, the helpers revert back to a standard OSPF behavior.

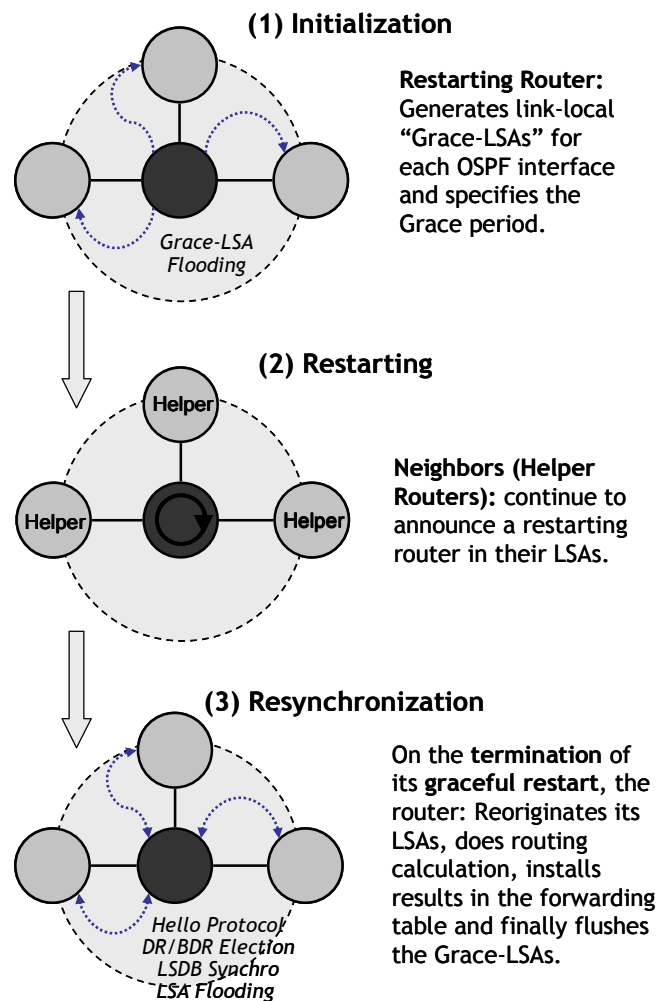


Figure 1 - OSPF Graceful Restart Mechanism

2.3 Maintenance Process

The graceful restart mechanism allows performing maintenance operations on the control plane of network equipments with a minimal impact on the traffic. These maintenance operations ranges from simple restarts and software upgrades to hardware interventions. Even if it is not advised, graceful restarts can be used for unplanned outages (such as the crash of a router's control software, an unexpected switchover to a redundant control processor, etc).

Based on these mechanisms, we propose here a solution that completely automates the maintenance process in network-wide environments. Figure 2 illustrates the proposed concept. The role of the operator is to simply publish maintenance jobs (targeted equipments and required procedures), define high-level performance objectives and specify safety parameters. The computation and implementation of the maintenance plan is done within the network elements through distributed and cooperative processes. Finally, the operator has the ability to track live progress and completion reports, which keeps him within the control loop of its network.

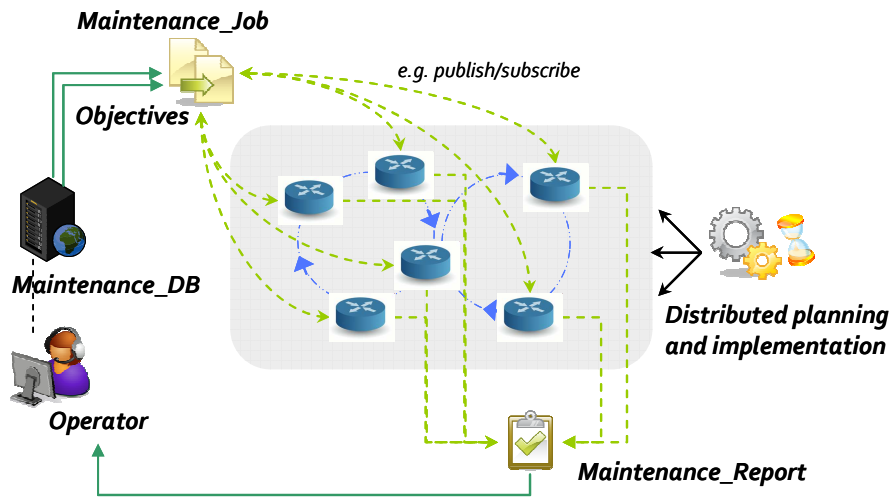


Figure 2 - Automating the maintenance process

The implementation of this solution requires addressing the following key features:

- **Providing relevant management tools:** This includes the implementation of management interfaces and communication platforms that allow operators to publish maintenance targets, to enforce high level objectives and then to track progress reports. It should be also used by operators to specify safety parameters that allow stopping the automated processes and reverting back to manual procedures, in order to avoid cascading failures and guarantee the network consistency. A relevant candidate option to implement the above communication requirements and interfaces is the use of publish/subscribe systems. These can be instantiated through either centralized or distributed repositories.
- **Empowering the network:** This is a key enabler for autonomic behaviors. Besides the extraction of dependency models and technical constraints that shall drive the planning of maintenance actions, it is mainly about the design and implementation of distributed planning algorithms that are driven by the functional constraints as well as the performance objectives enforced by the operator.

The reminder of this chapter (solution description) focuses on the proposal and assessment of such a distributed process in the case of domain-wide control plane maintenance using OSPF graceful restarts. The objective is to compute and implement, in a distributed way, the optimal maintenance plan that complies with the operator's objectives and underlying technical constraints. It is actually an election process that designates the candidate routers at each maintenance step.

2.4 Planning Algorithm and Metrics

The characteristics of the distributed planning algorithm that should be implemented by all the network nodes during the election process are as follows:

- Compliance with the protocol constraints in terms of graceful restart (previously outlined in section 2),
- Support of multiple election metrics in order to fulfill the operator directives and performance objectives,

- Minimizing the communication overhead necessary to complete the election process,
- Ensuring the convergence of the election process in a finite and short period of time.

```

N: List of candidate neighbours and their
      respective metrics
l: Metric of the local Node.

function Initialisation()
  l <- ComputeLocalMetric()
  Broadcast(l, N.all_addresses)
  while N is not completed
    M := ReceiveMessage()
    N[M.sender].metric := M.received_metric
  StartPlanification()
  return

function StartPlanification()
  while(true)
    if (Best(l,N)) // is l the best one among all
the metrics within N?
      Broadcast("Maintenance",
N.all_addresses)
      PerformMaintenance()
      break
    else
      M := ReceiveMessage()
      if M.status = "Maintenance"
        Broadcast("Helper",
N.all_addresses)

```

Figure 3 - Pseudocode of the distributed planning algorithm (one round)

A version of such a distributed algorithm is detailed in Figure 3. At each maintenance step, this algorithm allows to elect one candidate node per neighborhood. The choice of the appropriate nodes is based on a decision metric. In case of equality between the metric of two or several nodes, an arbitrary deterministic parameter is used to break indecisions (for example, a hash value of the router identifiers). Of course, multiple metrics are possible for such an election; the relevant metric should be chosen according to the performance objective set by the operator.

Furthermore, when implementing the proposed algorithm, each node communicates only with its direct neighbours and, based on the exchanged metrics, the appropriate nodes are designated for maintenance. Therefore, the communication overhead is very limited and the convergence time is bounded by the depth of the network.

2.5 Metric Choice

It is important to remind that the main constraint when implementing graceful restarts is the impossibility for two adjacent nodes to simultaneously perform a graceful restart.

Consequently, the planning of maintenance operations can be modeled as graph coloring problem [4]. Based on this, we propose a first set of possible metrics that shall be considered in the assessment study. In the following, these are described along with their expected behavior and performance results:

- **Maximum Degree:** In this case, the election process designates first the nodes with the highest degree (number of one-hop neighbors). This is a well known simple heuristic [5] to minimize the number of rounds necessary to perform the maintenance and restart of all the routers, which will thus minimize the overall maintenance time.
- **Minimum Degree:** Oppositely, the election process designates first the nodes with the lowest degree. Using this strategy, the maintenance process focuses first on the routers within the periphery of the network and then proceeds to the more central ones.
- **Random Metric:** The election process is here based on an arbitrary criterion which allows to fairly distribute the position of restarting routers within the network topology. The results of this metric can be used as a comparison reference as well.
- **Maximum 2-hop Degree:** The metric is computed as the degree of the node plus the mean degree of its one-hop neighbors. It is a more complex metric, in the sense that its computation is less straightforward than the previous ones. It also tends to minimize the overall maintenance time.

2.6 Preliminary Evaluation Results

In order to evaluate the previous metrics and confirm their expected behavior, we have run a first set of experiments using realistic ISP topologies [6]. The proposed election process and candidate metrics are implemented among the nodes of these topologies with a target to perform a network-wide maintenance. The values of the arbitrary deterministic parameter, that are used to compute the random metric and to break indecisions, are changed in each experiment. This leads to a variation of the obtained performance results.

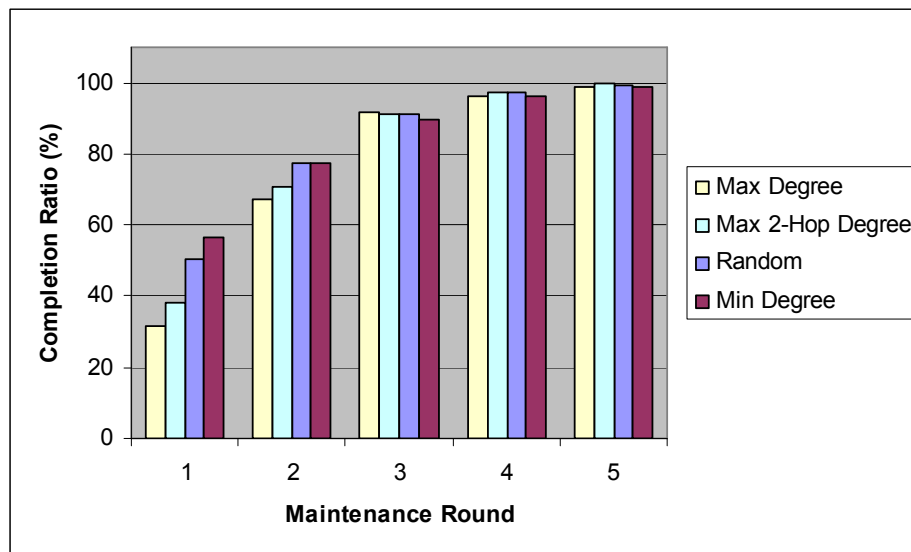
Table I shows the mean number of maintenance rounds that are necessary to address all the network routers for six different topologies. The metrics based on maximum degrees confirm their performance in term of total maintenance time, even if the optimality is not guaranteed. Indeed, minimizing the number of rounds (graph coloring) is NP-complete [4].

Besides this, the completion ratios after each maintenance round are plotted in Figure 4. These are mean values for all the tested topologies and relative experiments. The "minimum-degree" metric shows very high completion ratios during the first maintenance rounds. However, the central nodes, which are strongly connected, are addressed almost sequentially at the end of the maintenance process. This trend significantly increases the total maintenance time. The "maximum 2-hop degree" and "random" metrics show trade-off performance results and hence represent relevant alternatives when multiple objectives are expressed.

In addition to these experiments, we have developed a proof-of-concept demonstrator that confirms the previous conclusions, as well as the convergence of the planning procedure. This prototype is based on an emulation of the control plane features and an implementation of the distributed election processes. This prototype is in the scope of WP5; more details can be found in the D51 deliverable [7].

TABLE I. Number of Rounds to Restart all the Routers

TOPOLOGY SIZE	RANDOM	MIN DEGREE	MAX DEGREE	MAX 2-HOP DEGREE
79 NODES	4.56	4.58	4	4
87 NODES	4.67	5.66	4	4
108 NODES	5.3	5.86	5	5
141 NODES	5.71	6.98	4.82	5
161 NODES	5.49	5.64	4	5
315 NODES	7.11	8.08	6	6

**Figure 4 - Mean completion ratios after each maintenance round**

2.7 Synthesis and Future Steps

One of the main challenges of the proposed maintenance solution is to translate the operator's objectives into appropriate election metrics and processes within the network. The studied metrics are relevant candidates to address the operator requirements, and their priorities, in terms of total maintenance time and execution order, as shown by the obtained results.

However, it is necessary to augment this portfolio in order to implement further operator constraints, especially to secure the maintenance process in case of failures. For example, the operator may limit the number of simultaneous restarting routers or impose the constant availability of backup paths.

Indeed, as future work, we are planning to evolve the maintenance process so that it supports a larger range of performance targets, and integrates learning functionalities as well. Besides this, we will capitalize on the developed emulator and prototype (WP5) in order to deepen the evaluation of the proposal and to study the impact of unpredicted failures.

3 Energy-Aware Routing: a Fixed Charge Network Flow Formulation

Reduction of unnecessary energy consumption is becoming a major concern in wired networking, in reason of both the potential economical benefits and its forecast environmental impact. These issues, usually referred to as “green networking”, relate to embody energy-awareness in the network elements and processes.

Once a network has been designed (i.e., the resources that will compose it have been chosen), a periodical, on-line, process decides how the network resources will be used. This process is referred to as “network dimensioning”. One of the most common practices for acting in a *green* fashion in network dimensioning consists in *resource consolidation*. This technique aims at reducing the energy consumption due to devices underutilized at the considered interval of time. Given that the traffic level in standard networks approximately follows a well-known daily and weekly behavior [12], there is an opportunity to aggregate traffic flows over a subset of the network devices and links, allowing others to be switched off temporarily or be placed in sleep mode (if available). This solution shall preserve connectivity and Quality of Service (QoS), for instance by limiting the maximum utilization over any link. In other words, the required level of performance will still be guaranteed, but using an amount of resources that is dimensioned for the current network traffic demand rather than for the peak demand (or more). Flow aggregation may be achieved, for example, through a proper configuration of the routing weights.

3.1 Problem Formulation

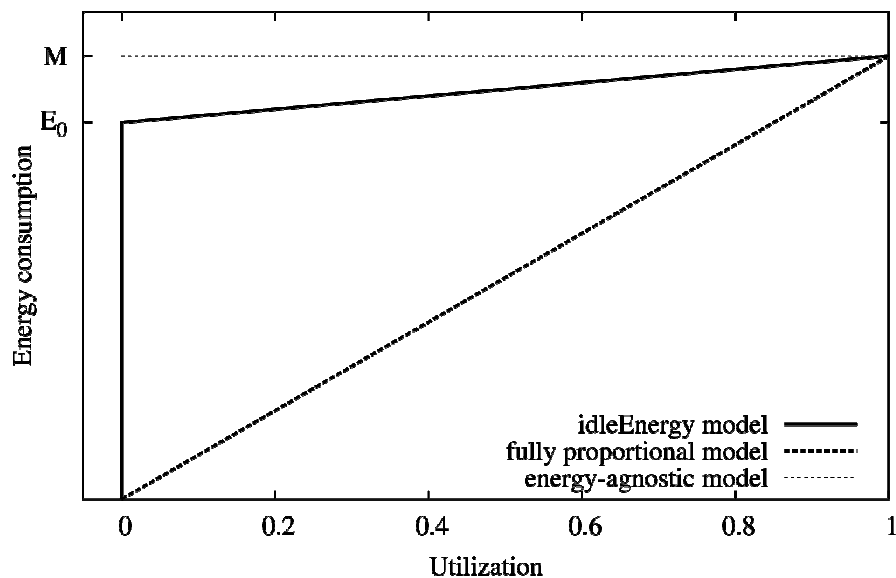


Figure 5 - The used models for the network device energy consumption as parametrized function of the device utilization.

This approach has been evoked in [11] as a hypothetical working direction, and in [10], with the proposal, and the evaluation, of some greedy heuristics, based on the ranking of nodes and links with respect to the amount of routed traffic in the energy-agnostic configuration. In our work, we instead model the problem of the optimization of the total energy consumption of a network as a function of the utilization level of the network devices. We analyze the Integer Linear Programming (ILP) formulation for this problem, falling into the set of Fixed Charge Network Flow (FCNF) problems. We found that the complexity of the solution largely depends on the model used for the Energy consumption of the network devices. Also, it is known that modeling the energy consumption of network components is an hard task, mainly because of inconsistency, scarcity and oldness of data. For these reasons, we decided to use two complementary approaches:

1. a more realistic approach, where device energy consumption presents a strong idle component as soon as the device utilization is greater than 0, and a smaller component proportional to the utilization itself (this model will be referred to as “idleEnergy”, it is illustrated in Figure 5);
2. a more ideal approach, where device energy consumption behaves proportionally to their utilization (this model will be referred to as “fully proportional”, it is illustrated in Figure 5 and was originally described in [9] as the ideal case of “proportional computing”).

The “idleEnergy” model brings to a rather complex solution, even if there exist mathematical tools allowing to obtain a solution in a reasonable time. The “fully proportional” model brings, instead, to a formulation of the solution as a linear problem, and to a consequent strong reduction of the solution complexity with respect to the one of the “idleEnergy” formulation. The problem is mathematically defined by the following *LP* formulation:

$$\min \frac{1}{2} \sum_{(i,j) \in L} \left(\frac{(l_{ij} + l_{ji})E_{fij}}{c_{ij}} + x_{ij}E_{0ij} \right) + \sum_{n \in N} \frac{l_n E_{fn}}{c_n} + x_n E_{0n}$$

subject to:

$$\sum_{i,s,d \in N} f_{ij}^{sd} - \sum_{i,s,d \in N} f_{ji}^{sd} = \begin{cases} r_{sd} & \forall s, d, i = s \\ -r_{sd} & \forall s, d, i = d \\ 0 & \forall s, d, i \neq s, d \end{cases}$$

$$\sum_{s,d} f_{ij}^{sd} = l_{ij} \leq \alpha c_{ij} \quad \forall i, j \in L$$

$$l_n = \sum_{(i,n) \in L} l_{in} + \sum_{(n,i) \in L} l_{ni} \quad \forall n \in N$$

$$Zx_{(ij)} \geq l_{(ij)} + l_{(ji)}$$

$$Zx_n \geq l_n$$

where N is the set of nodes and L the set of links in the considered network; l_a is the load of the network element a and c_a its capacity; $f_{s,d}$ is the amount of the flow from node s to node d that has been routed on the link (i,j) ; Z is a “big” number (used as part of the “big-M method”); and ϵ and δ are the two parameters profiling the energy consumption of the network element a , as previously defined in Figure 5, and ΔE_a is the difference between the maximum energy consumption (M) and the idle energy consumption (E_0).

For the evaluation of our solution we chose to use the GEANT topology [13]. Since in the GEANT network all nodes are sources and destination of traffic, this constitutes a *worst case* scenario, as nodes can not be generally turned off. This is a good candidate for representing a lower bound benchmark for the evaluation and comparison of different algorithms. Moreover GEANT is a real network offering public data on both its *topology* as well as its *traffic matrices*, which ensures a certain degree of realism in the evaluation. We took as a reference the routing performed using IGP-WO optimized weights, and enabling Equal Cost Multi Path. IGP-WO is the standard practice in the operators networks, which we will refer to as “standard routing case”. We evaluated our algorithm on the basis of the percentage of energy that may be saved, with respect to the standard routing case.

Switching off network elements and optimizing their utilization brings to energy saving from one side, but also to a reduction of the system robustness from the other side. Nowadays, the common practice in the operators’ network to guarantee robustness and a Quality of Service (QoS) level, is the limitation of the charge of the network elements. In order to obtain a realistic solution, we introduced in our formulation a parameter α representing this limitation of the network element charge. We evaluated the effects of this parameter on the achieved energy saving.

3.2 Preliminary Results

Our preliminary results show that it is possible to obtain an optimal solution to this problem in a reasonable time, employing standard computational power, and considering realistic and fairly complex topologies.

Observing the results for the two considered energy models, we can see how in the first case (idleEnergy model) the energy saving is consequence of switching off network devices, while in the second case (fully proportional model) energy saving comes as consequence of aggregating the traffic into the path involving the most energy efficient devices. Given the specific topology and traffic level we chose, it is not possible switching off nodes, since every node is source and destination of traffic requests, but is possible switching off links. As a consequence, we can achieve a small energy saving due to nodes and a considerable one due to links. However, it should be noticed that the link component represents a small contribution to the total one in our model, so that the overall saving is modest: saving account to about 0.2% in the case of the idleEnergy model, and about 4% in the case of the fully proportional model, as evaluated on a typical day (i.e., over 24 hourly traffic matrices). Notice that we considered consumption figures typical of Ethernet links, which are two to four order of magnitude smaller than the one used for the nodes. We point out that the situation may considerably change when taking into account optical interconnections over real distances of thousands of kilometers, requiring periodical signal regenerators, involving much higher energy consumptions.

From the comparison of the two proposed energy models, we can also notice how a network involving devices whose energy consumption is fully proportional to their utilization allows achieving much higher energy saving with respect to one involving more energy-agnostic

devices, as expected. This is due to the fact that in the second case, the main opportunity to reduce the energy consumption is switching off network elements, which strongly depends on the considered scenario.

In future work we plan to consider topologies that including multi-homed nodes and transport nodes (i.e., nodes that do not represent sources or destination of traffic requests), and to compare the optimal solution with the existing heuristics, in terms of achievable energy saving, of the robustness of the solution, and of the solution complexity. We are also interested considering different energy profiles for the network elements, as they may result from different network technologies.

4 LOCARN

LOCARN is an imaginative network paradigm compliant with many autonomic networking properties. LOCARN stands for "Low Opex and Capex Architecture for Resilient Networks". It aims to explore two concepts (auto-forwarding and enhanced broadcast) in order to increase as much as possible network simplicity and hence increase savings in both OPEX and CAPEX domains.

Main concepts of LOCARN may be successfully compared to the principles defined and specified for Ad-Hoc Mobile networks (e.g. Dynamic Source Routing, IETF RFC-4728). LOCARN addresses more specifically Transmission Networks by an intrinsic simplicity and the willingness to take into account some "Carrier Grade" properties.

As detailed later, LOCARN will be compliant with main properties encountered in Autonomic Networking (self-* properties).

This document aims to offer a sufficient enough in-depth view of the LOCARN network concepts with the objective to prepare its Proof-Of-Concept implementation. That latter will be done through TIGER2-WP5 activity.

4.1 Problem Statement

A LOCARN network is intrinsically able to auto-configure itself dynamically, taking into account the load of nodes and links within a rather low time scale compatible with the creation of a new service and/or any topological changes on the network.

To that end, a constant self-analyzing process allows that very quick (re) configuration of the network behaviour. The bottom line of these operations that are "transparent" at the management level, is, from a service point of view, a network somehow "Plug&Play".

Moreover, in classical packet networks (e.g. IP or Ethernet), nodes use information included in the well-known "routing/forwarding tables" to forward data frames. The difficulty is to maintain up-to-date the content of these tables which implies the use of a dedicated control and/or management plane to fill in these tables. To alleviate this drawback, auto-forwarding paradigm (implemented in LOCARN) leverages of only information directly present in the header of the frame to switch it without the necessity to perform look-up in any table in each transfer node.

Obviously, the downside of this method is the need, at the Origin edge port, of a specific control plane to build the routing information added in the frame header. The global principle proposed in LOCARN relies on the use of a broadcast to discover the possible paths once a given service is under creation (pre-registered at the management plane).

Several variants of that principle may be conceived and implemented in LOCARN with different performance objectives. However, all should use specific control frames as PATH_REQUEST/PATH_DISCOVER to elaborate GO_PATH and BACK_PATH information to be inserted in each frame header at the Origin Edge Ports.

In this document, a rather basic broadcast named "enhanced incremental broadcast (EIB)" is used and specified.

4.2 Solution

This paragraph proposes a generic description of LOCARN components. Frames used in a LOCARN network are listed and described. Then the structure of a possible LOCARN node is specified both at the architecture and functional levels. Finally, some other issues will be introduced with the willingness to open possible further studies.

4.2.1 Frame

The instantiation of LOCARN paradigm as a packet transport network (as targeted in TIGER2 project) leads to cover three main functionalities:

- the **transport** itself of data frames
- the **service creation** between Service End points (discovery process)
- the **service maintenance** in case of any change in server layers as an example (e.g. topology change, ...)

In order to perform these functionalities, several kinds of LOCARN frames are specified. Only five types of frames are necessary to cover these three items and to make the LOCARN solution an operational network.

Each LOCARN frame is referenced by a LFT (LOCARN Frame Type). Details regarding LOCARN frame format structures are provided in Annex A: Generic LOCARN Frame Formats.

Let's notice that this section specifies generic LOCARN frames that are agnostic to any server layer. Any real LOCARN implementation will need to adapt LOCARN frames to a specific server layer.

4.2.1.1 Frame dedicated to the Transport function

The **LOCARN Data Frame (LDF)** is the internal frame format inside the LOCARN network. It is composed of the LOCARN payload (DF) and the LOCARN overhead.

- a) The LOCARN payload is somehow the "customer" frame entering inside the LOCARN network and named here **Data Frame (DF)**. DF is transparently routed from a physical Edge ingress port to a physical Edge egress port.
- b) The LOCARN overhead contents the information necessary for the DF propagation along the LOCARN path (from one Edge to another one). Among other things, the LOCARN overhead contains the LOCARN Service Identifier.

4.2.1.2 Frames dedicated to the Path Discovery Process

The path discovery process uses two kinds of frames:

- a) The **Path Request LOCARN Frame**, generated by a Service Origin Edge Point and broadcasted inside the LOCARN network. To provide the Enhanced Incremental Broadcasting (EIB) process, each node adds some information to an incoming Path Request before relaying it. This information mainly contents some identifiers (input port, output port and optionally the node identifier) and some information about the estimated "quality" to relay the frame through this port on this node. Other more classical information items are generated by the Service Edge Origin Point like LOCARN Frame Type and the Service Identifier.
- b) **Path Discover LOCARN Frame** is generated by a Service Edge Destination Point upon reception of a LOCARN Path Request if the Service Identifier contained in the

LOCARN Path Request received has been previously "registered" at this destination point. This destination point uses the Information inside the received Path Request frame to compute the path to go back to the Service Edge Origin Point and transmit a copy of the information found in the received Path Request LOCARN Frame to inform it that a path exists from the Service Edge Origin Point to the Service Edge Destination Point.

4.2.1.3 Frames dedicated to the Service Maintenance

Two types of frames are related to the Service maintenance process:

- a) The **Hello Forward LOCARN Frame** initiated by the Service Edge Origin Point at periodic interval. This frame is carried in the Data Plane using the standard LOCARN auto-forwarding scheme.
- b) The **Hello Back LOCARN Frame** initiated at the Service Edge Destination Point upon reception of an Hello Forward LOCARN Frame.

Note that these two types of frames carry a Service Identifier. Thus, the maintenance process is related to each End-to-End service.

4.2.2 Node

A LOCARN Node may be described in a "mecano" way as an organization of different functional blocks. This way, a LOCARN node is composed of the three following functional blocks as depicted in Figure 6:

- LOCARN Matrix Functional Block (LMFB)
- LOCARN Tee Path Request Functional Block (LTPRFB)
- LOCARN Port Functional Blocks (LPFB). That latter can also be seen as a compound block and divided into two sub-blocks:
 - The LOCARN Port Edge Origin Functional Blocks (LPEOFB)
 - The LOCARN Port Edge Destination Functional Blocks (LPEDFB)

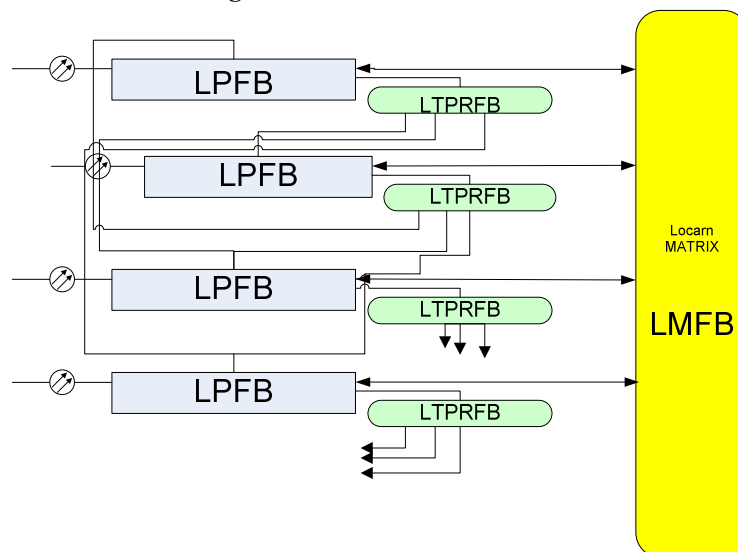


Figure 6 - LOCARN Node (4 ports)

LMFB is common for the whole node whereas LPFB and LTPRFB are local to the physical port.

The role of each bloc is described below.

4.2.2.1 LOCARN Matrix Functional Bloc (LMFB)

The LOCARN Matrix Functional Block (LMFB) is only able to manage LOCARN Data Frames. If the frame is not recognized as a LOCARN Data Frame (role of LFT), the frame is discarded. Else, LMFB handles the LOCARN Data Frame header in order to relay it to the relevant egress port according to the following actions:

- Look-up inside the LOCARN Data Frame header the value of the pointer LOCARN Path Hop (LPH) indicating the current position (i.e. the current node) in the whole path.
- Select the egress port given by the pointer LPH
- Increment LPH (will be used by the following nodes)
- For security reason, if the new value of LPH is greater than the LOCARN Path Length (LPL) contained in the LOCARN Data Frame header, discard the frame
- Push the frame (modified with LPH incremented) to the relevant port.

4.2.2.2 LOCARN Port Edge Origin Functional Bloc (LPEOFB)

As mentioned above, the LOCARN Port Functional Blocs (LPFB) is divided into two LPEOFB and LPEDFB sub-blocs. Their internal structure is depicted in Figure 7.

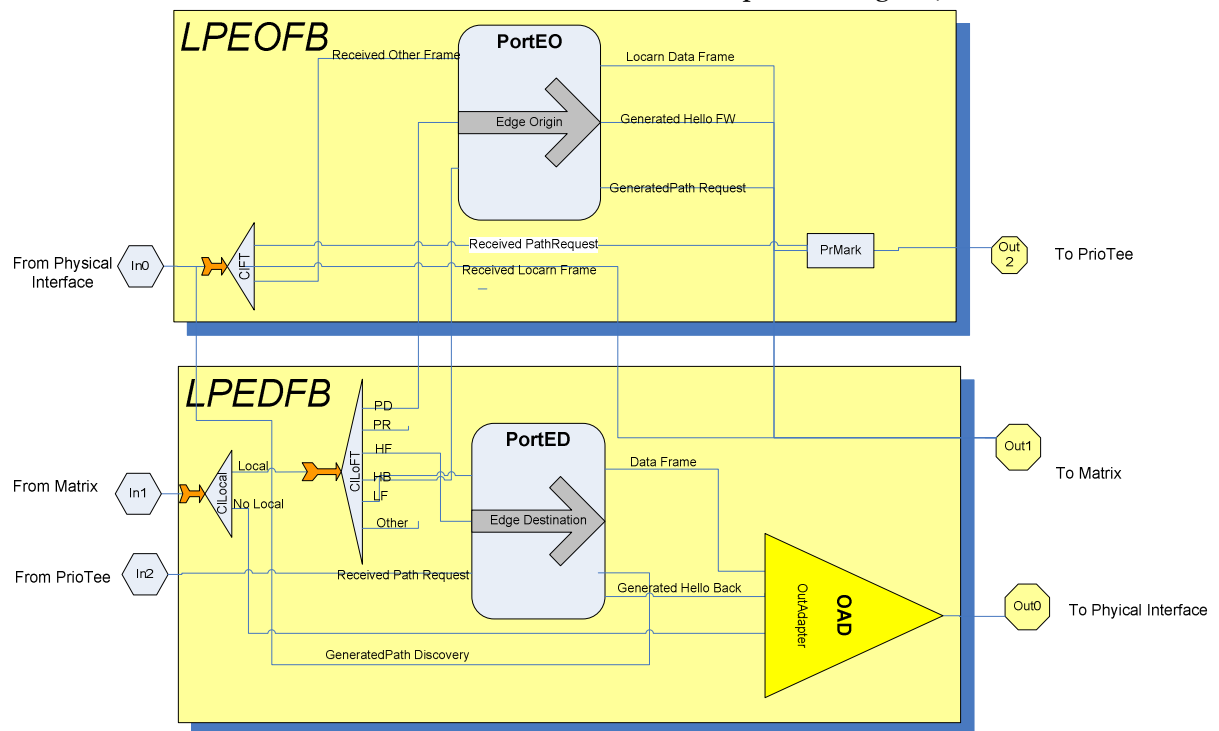


Figure 7 : Interconnection of sub blocs in a LOCARN port

The LPEOFB sub-bloc performs the following actions:

CIFT (Classifier Frame Type):

- Classify frames to relay them to the relevant block
 - All LOCARN Frames, except Path Request LOCARN Frame, towards the LMFB
 - Path Request LOCARN Frame towards the PrMark
 - Other frames towards the PortEO to be eventually transformed into LOCARN Data Frame.

PortEO: Manage the Service Origin Edge Point for this physical port:

- Generate Path Request LOCARN Frame for services in case of service creation or service maintenance
- Generate Hello Forward LOCARN Frame for all active services
- Encapsulate the Data Frames adding the LOCARN overhead

PrMark:

- Temporary mark the frame with the Port Id (creation of an Internal Path Request Frame - IPR)
- Push the IPR to the LTPRFB associated to this port

Note that the use of IPR is only to manage (store) the input port Id during the transfer of the Path Request LOCARN Frame inside the node. IPR will be transformed into a Path Request LOCARN Frame later at the egress port once the output port Id will be known and added.

4.2.2.3 LOCARN Port Edge Destination Functional Block (LPEDFB)

The LPEDFB sub-block performs the following actions:

ClLocal (Classifier Local):

- Classify the frame according the LPH and LPL values to determine if the frame must be parsed by PortED or directly pushed downside (towards OAD).

ClassOut: Used in the case the frame must be handled locally. In this case, classify the frames between

- Path Discover LOCARN Frame and Hello Back LOCARN frame transmitted to the PortEO
- Hello Forward LOCARN Frame transmitted to PortED

PortED: Manage the Service Destination Edge Point for this physical port:

- Generate Path Discover LOCARN Frame upon reception of an Internal Path Request (IPR) if the Service Identifier has been registered on this port. The Path Discover LOCARN Frame includes information from ingress and egress ports Id and Qlo (Local Quality measurement of the bit-rate of the link connected to this port as well as an estimation of the used rate for this link (queue filling rate, rate measurement, ...))
- Generate Hello Back LOCARN Frame upon reception of Hello Forward LOCARN Frame
- De encapsulate the LOCARN Data Frames to retrieve the customer Data Frame

OAD:

- Output Adapter (Queuing, shaping...)

4.2.2.4 LOCARN Tee Path Request Functional Bloc (LTPRFB)

Each LFTFB duplicates IPR and pushes the cloned frames to all LPEDFB of the node (all other ports). No change is made inside the frames.

4.2.3 LOCARN Frame processing

This section analyses the relation between LOCARN Frames and functional blocs described above.

PR (Path Request LOCARN Frame)

- Pre generated by PortEO
- Finalized and transmitted by PortED

HF (Hello Forward LOCARN Frame)

- Generated by PortEO
- Upon reception, PortED generates an Hello Back LOCARN Frame

LDF (LOCARN Data Frame)

- Generated by PortEO upon reception of a DF
- De encapsulated by PortED
- Routed by LMF

DF (Data Frame)

- Customer frame entering at the boundary of the LOCARN Network
- Encapsulated or discarded by PortEO
- De encapsulated by PortED

PD (Path Discover LOCARN Frame)

- Generated by PortED
- Analyzed and discarded by PortEO

HB (Hello Back LOCARN Frame)

- Generated by PortED

IPR (Internal Path Request)

- Generated by PortEO
- Duplicated by LTPRFB
- Analyzed and completed by all PortEDs

4.2.4 Other issues

This section aims at introducing some open issues that have not been studied yet but that should require more in-depth investigations. It can not be considered as part of the LOCARN specification itself.

4.2.4.1 Specific Path Request and Path Discover issue (functional issue)

At the path discovery step, information about nodes (more exactly about ports) are collected by the Path Request LOCARN Frame and returned at the Service Edge Origin Point using the Path Discover LOCARN Frame. At the generation step of the Path Discover LOCARN frame, specific information about the service itself (cost for example) may be added by any potential Service Edge Destination Point.

This information may be used by the Service Edge Origin Point to choose "the best" Service Edge Destination Point and "the best" way to be connected to this Service Edge Destination point.

The exact semantic of "THE BEST" depends on the kind of service managed by the LOCARN network. Any LOCARN implementation MUST instantiate:

- Specific fields to carry the information in the Path Request LOCARN Frame
- A specific algorithm to compare Path Discover LOCARN frame content.

4.2.4.2 LOCARN Edge Port specificities (architecture issue)

This item raises the question to declare or not ports at the boundaries of the LOCARN network as LOCARN Edge Ports.

Beyond the question of the security, this LOCARN Edge Port specificity brings network architecture interests and scalability improvements. Indeed, if a port is stamped as LOCARN Edge Port, all incoming frames will be processed as DF, even if they are already LDF. By this way, we can manage an overlay "LOCARNinLOCARN" network architecture and increase the scalability with a LOCARN hierarchy.

4.2.4.3 "Big" node architecture (scalability issue)

Inside LOCARN Frames, the LOCARN Path (pointed by LPH) is a part of the traffic overhead inside the network. This overhead is linked to the network size. For rather big network, it could be interesting to reduce this extra-traffic part generated by this overhead by reducing both a) the size needed to code each port, b) the network diameter accessible by a node.

So, it is important to avoid big Port Identifier fields. The drawback is clearly to limit the size of the node (the number of ports). A workaround may be to have a specific node architecture where "virtual LOCARN nodes" are interconnected together by a LOCARN port (Virtual LOCARN nodes have exactly the same specification as LOCARN nodes). In this case, the "Big" LOCARN node will have to handle the LPH as many time as the number of Virtual LOCARN nodes crossed over.

This mechanism offers the ability to bypass the limitation of number of ports on a LOCARN node. The drawback is a reduction of the networks diameter that includes all LOCARN nodes, virtual or not.

4.2.4.4 "LOCARN Path Port List" size (interoperability issue)

In order to increase the Plug&Play property of a LOCARN network, it may be interesting to use equipment with different LPL (LOCARN Path Length) and to expect that the network take automatically the different values of LPL acceptable by the equipment of the network. With such this very useful enhancement of the protocol, some changes have to be made in the LOCARN Frames management.

At best, an enhancement of the Path Request / Path Discovery protocol will be able to automatically discover the LPL per service.

4.3 Preliminary evaluation results

4.3.1 LOCARN PoC implementation

A first LOCARN implementation has been realised using this specification and demonstrated in the scope of TIGER2 – WP5 work package activity.

4.3.2 LOCARN compliancy with Autonomic properties

LOCARN intrinsically offers autonomic properties, simplicity and efficiency. Most of these autonomic properties are directly linked to the frame management and fundamentals of LOCARN. As an example,

- self-discovery (Implicit)
 - using Path Request LOCARN frame and Path Discover LOCARN Frame
- self-configuration (Implicit)
 - discovered information about the network are automatically usable by the Data Plane. Only Services need to be configured
- self-healing
 - End-to-end maintenance OAM frames per service
- self-optimization
 - Periodic Path Request LOCARN Frame can help to reconfigure the path of services taking into account the availability of network resources
- self-management (not applicable, not needed)
- self-protection
 - Possibility to define Edge Ports with filtering behaviors.
 - Only "registered" services are allowed on the network

5 Self-optimization of network resource allocation in MSTP

In this study case there is a centralized entity in the network that is in charge of routing and network resource allocation, based on the knowledge of the actual network state. For each new traffic demand, the online routing allocates the necessary network resources, but it usually results in suboptimal allocation.

Self-optimization of network resource allocation aims to provide automatically without human intervention an optimal allocation (the minimum). Periodically an optimization model is run, then the calculated optimal resource allocation is compared with the actual one, and if their difference is large enough, reconfiguration of network resources is initiated. The time between runs is based on the time required running the optimization model, which usually is slow.

This study case of self-optimization considers Carrier Ethernet networks using MSTP (Multiple Spanning Tree Protocol) based technologies. In this context the goal of the optimization model can be either minimizing the number of allocated spanning trees or the amount of allocated capacity.

5.1 Optimization of network resource allocation

A centralized entity in the network knows the actual network state, i.e., the topology, the carried traffic and the network resource allocation. For each new traffic demand arriving to the network, a new resource allocation is necessary. Optimal allocation could be achieved through complex algorithms that use all the “historical” information about traffic and topology, but they are too slow to be applied upon the arrival of each new traffic demand. Instead other simple algorithms (“online”) can be used, which are faster at the cost of being not optimal because they are based on “snapshot” information.

Online routing results in a resource allocation that is not optimal (in the case of MSTP based technologies, it means either the number of allocated spanning trees or the amount of allocated capacity are more than necessary). An optimal allocation (the minimum) could be achieved by running periodically the optimization algorithm and then deciding whether to do reconfiguration.

The use of optimization models for network resource allocation is optional. If it is not used, the resource allocation provided by online routing becomes not optimal and the network is run inefficiently. If used, a human administrator decides when to run the optimization model (which usually is slow), and whether to perform reconfiguration if the actual resource allocation is too far from the optimal. Human intervention implies higher operational costs.

5.2 The proposed scheme for self-optimization

Self-optimization of network resource allocation in MSTP provides two basic benefits, the minimization of network resource allocation (either the number of allocated spanning trees or the amount of allocated capacity) and the reduction of operational costs. The optimal resource allocation is obtained automatically, without human intervention, through the following way:

- Topological and traffic information is constantly communicated from network nodes to the centralized entity.
- Periodically (the time period is chosen by the operator) the optimization model is run for either minimizing the number of allocated spanning trees or the amount of allocated capacity (this is chosen by the operator). Then the calculated optimal resource allocation is compared with the actual one, and if their difference is large enough (the threshold is chosen by the operator), reconfiguration is started.
- If reconfiguration is started, the proposed reconfiguration is communicated from the centralized entity to the network nodes.

Therefore the proposed solution requires the following:

- It requires an infrastructure for the communication between the centralized entity and the network nodes.
- It requires the operator to define how often the optimization model is run, which threshold would trigger the reconfiguration, and whether to optimize either the number of allocated spanning trees or the amount of allocated capacity.
- It requires an optimization model for network resource allocation in networks based on MSTP technologies. Specifically, an algorithm for optimizing either the number of allocated spanning trees or the amount of allocated capacity. The information that this algorithm would need is the network topology and the actual traffic matrix, provided by the ingress nodes to the centralized entity, each time there is a traffic or topology change. The output is the optimal allocation.

Next section is devoted to describe the optimization model for MSTP that we have designed.

5.3 Optimization model for MSTP based technologies

In this study case we use optimization models based on Integer Linear Programming (ILP) for routing and resource allocation. We consider Carrier Ethernet networks based on MSTP technologies, although we also include in the study, for comparison, those technologies based on label forwarding (i.e., ELS - Ethernet VLAN Label Switching-, PBB-TE - Provider Backbone Bridges Traffic Engineering -).

We start by reviewing the related work on optimization models in Carrier Ethernet. In the case of label forwarding based technologies, the work done in the design of ILP models to optimize resources (for technologies like MPLS) is substantial (e.g., [14]), and there is no need to propose new models since the existing ones can be applied without any modification. In the case of MSTP based technologies, the work done in the design of ILPs presents, to the best of our knowledge, the following two shortcomings:

- In all the studies, ILP models are always compared either among themselves or against the use of basic native Ethernet protocols.
- There is not any ILP model that guarantees a complete global optimum. The existing models rely on a heuristic to calculate part of the solution. For example, the model proposed in [15], uses a heuristic to pre calculate a set of spanning trees.

Following the previous related work, our goal is to propose an ILP model for routing and resource allocation in MSTP based technologies, which obtains the set of spanning trees that accommodate traffic to network resources achieving a guaranteed optimal allocation.

The tackled problem can be stated in the following way. Given a maximum number of trees $maxt$, a network graph $G = (N;E)$ and a traffic matrix $TM = NxN$, where N is the set of nodes, and E the set of links, the goal is to find a set of undirected trees T ($|T| \leq maxt$) and accommodate the traffic described by TM , so that the traffic is routed through the paths given by T , and the accommodated traffic is maximized.

The ILP is based on the multi-commodity flow problem. For each pair of nodes (s,d) , where $TM(s,d) > 0$, we refer to a commodity $c \in C$ such that the requested bandwidth of the commodity $BW(c) = TM(s,d)$ and the destination and source of c are s,d , respectively.

Based on this, the proposed ILP consists of the following indices:

- i,j for representing nodes in the network
- c a commodity given by TM

And the following parameters:

- BW_c set of requested bandwidths given by TM
- $S_{(c,i)}$ is set to 1 if node i is the source of commodity c , -1 if it is the destination and 0 otherwise.
- $C_{(i,j)}$ capacity of a link
- $maxt$ maximum number of spanning trees

The variables used in the model are the following:

- $f_{(i,j)}^{c,t}$ represents the amount of bandwidth accommodated for commodity c on link (i,j) as part of the tree t
- $x_{(i,j)}^t$ is 1 if link (i,j) belongs to tree t , 0 otherwise
- r_i^t is 1 if node i is the root of tree t , 0 otherwise
- h_i^t represents the height of node i in tree t

In order to ensure that each tree t has no cycles and is connected, the trees are modeled as unidirectional hierarchical trees, each tree has a root, and the root has height 0. If link (i,j) belongs to tree t , then $h_i^t - h_j^t = 1$, this property ensures that there are no cycles in the tree. Regardless of the fact that the trees are modeled unidirectional, the flow constraints are designed to consider them bidirectional.

The objective function is to accommodate as much bandwidth as possible through the entire network.

MAXIMIZE

$$\sum_{j,c,t} f_{(i,j)}^{c,t} \quad \forall i \mid S_{(c,i)} = 1 \quad (1)$$

SUBJECT TO

Routing constraints:

$$\sum_{c,t} f_{(i,j)}^{c,t} \leq C_{(i,j)} \quad \forall i, j \quad (2)$$

$$\sum_{j,t} f_{(j,i)}^{c,t} \leq BW(c) \quad \forall i, c \mid S_{(c,i)} = -1 \quad (3)$$

$$\sum_{j,t} f_{(i,j)}^{c,t} \leq BW(c) \quad \forall i, c \mid S_{(c,i)} = 1 \quad (4)$$

$$\sum_{j,t} f_{(j,i)}^{c,t} = 0 \quad \forall i, c \mid S_{(c,i)} = 1 \quad (5)$$

$$\sum_{j,t} f_{(i,j)}^{c,t} = 0 \quad \forall i, c \mid S_{(c,i)} = -1 \quad (6)$$

$$\sum_{j,t} f_{(i,j)}^{c,t} - \sum_{j,t} f_{(j,i)}^{c,t} = 0 \quad \forall i, c \mid S_{(c,i)} = 0 \quad (7)$$

Constraint 2 ensures that the accommodated traffic on a link does not exceed the link capacity. Constraints 3 and 4 ensure that the accommodated traffic does not exceed the demanded traffic. Constraints 5, 6 and 7 are the flow conservation constraints.

Tree shape constraints:

$$h_i^t \leq |N| \cdot \sum_i (x_{(i,j)}^t) \quad \forall j, t \quad (8)$$

$$h_j^t - h_i^t \geq 1 - (!x_{(i,j)}^t \cdot (|N| + 1)) \quad \forall j, i, t \quad (9)$$

$$h_j^t - h_i^t \geq 1 + (!x_{(i,j)}^t \cdot |N|) \quad \forall j, i, t \quad (10)$$

$$x_{(i,j)}^t + x_{(j,i)}^t \leq 1 \quad \forall j, i, t \quad (11)$$

$$\sum_j (r_j^t) \leq 1 \quad \forall t \quad (12)$$

$$\sum_i (x_{(j,i)}^t) \geq r_j^t \quad \forall j, t \quad (13)$$

$$\sum_i (x_{(i,j)}^t) \geq !r_j^t \quad \forall j, t \quad (14)$$

$$\sum_i (x_{(j,i)}^t) \leq |N| \cdot (r_j^t + \sum_i (x_{(i,j)}^t)) \quad \forall j, t \quad (15)$$

Constraint 8 ensures that nodes with height zero are only nodes that have no father in the tree, which are either the root or a node not belonging to the tree. Constraints 9 and 10 ensure that the difference between the height of two connected nodes in the tree is 1. Constraint 11 ensures unidirectionality. Constraints 12 and 13 ensure that there is only one root per tree and that the root is connected to at least one node. Constraint 14 ensures that the root does not have a father, and the other nodes do not have more than one. Constraint 15 ensures that a node that is not the root and does not have a father is not connected with any node.

Tree-flow constraint:

$$f_{(j,i)}^{c,t} \leq \text{MAX}_c(BW(c)) \cdot (x_{(i,j)}^t + x_{(j,i)}^t) \quad \forall i, j, c, t \quad (16)$$

This constraint ensures that the accommodated traffic follows the paths given by the trees. Note that the expression $(x_{(i,j)}^t + x_{(j,i)}^t)$ ensures that even though the trees are modeled unidirectional, traffic can flow in any direction given by the links belonging to the tree.

More details of the proposed ILP model can be found in [16].

5.4 Evaluation results

In this section we evaluate the performance of the proposed optimization model for MSTP based technologies in terms of the total accommodated traffic. We use different network topologies and traffic patterns. We also compare these results with the ones obtained with those network technologies based on label-based forwarding.

Two types of topologies have been used, grid topologies (for example in [17]) and defined topologies (for example in [18]). Specifically, two topologies are considered in this section: a grid topology of 36 (6 x 6) nodes and the defined cost266 topology [19]. For both topologies link capacity is set to 10 Gbps. For all the experiments it is assumed that all nodes are sources and destinations, meaning traffic is generated among all nodes.

This is an offline scenario in the sense that all traffic is known in advance and used in the optimization model presented in the previous section. The traffic between source and destination is uniformly distributed between [100, 1024] Mbps. For modeling the label based forwarding technologies the proposed models is modified in the following way: the variable $f_{(i,j)}^{c,t}$ is replaced by $f_{(i,j)}^c$, and the rest of the variables are removed. Additionally all the tree shape and tree-flow constraints are removed. In order to perform a fair comparison, the objective functions are the same but using the replaced variables. The models are solved using the Xpress-Optimizer [19].

Performance is evaluated in terms of the accommodated traffic. The accommodated traffic is the sum of the amount of traffic that is routed through all the sources and destinations, and it is the objective function of the proposed models. This value is measured against the number of allowed trees (*maxt*) parameter specified in the model. The results for label-based forwarding approaches, given that they are not subject to the maximum number of trees *maxt*, are plotted as a constant horizontal line among the number of trees. This means that only one value is calculated for the label-based forwarding per plot. On the other hand, for the STP-based, one value (represented as a point in the line) for each of the different number of trees (*maxt*), is calculated per plot. The results are presented in Figure 8.

Results show that when using just one tree the optimal performance of the STP-based approaches is between 36% (grid) and 41% (cost266) less than the label-based forwarding ones. The minimum number of trees that give the same performance than label-based approaches is 70 and 110 for the cost266 and grid topologies, respectively. If the total accommodated traffic is divided by the number of trees, then the average accommodated traffic per tree is 4381 Mbps (for grid) and 5406 Mbps (for cost266). This means that even thought in the grid topology more traffic can be routed, in the cost266 topology more traffic can be routed per tree.

Summing up, our proposed model successfully solved the stated problem for two topologies, and the results showed that the model can be used to determine the minimum number of VLANs the network must support in order to route a specific traffic matrix. Results also show that an optimal use of multiple spanning trees can make the MSTP based approaches accommodating the same amount of traffic than the label-based forwarding ones.

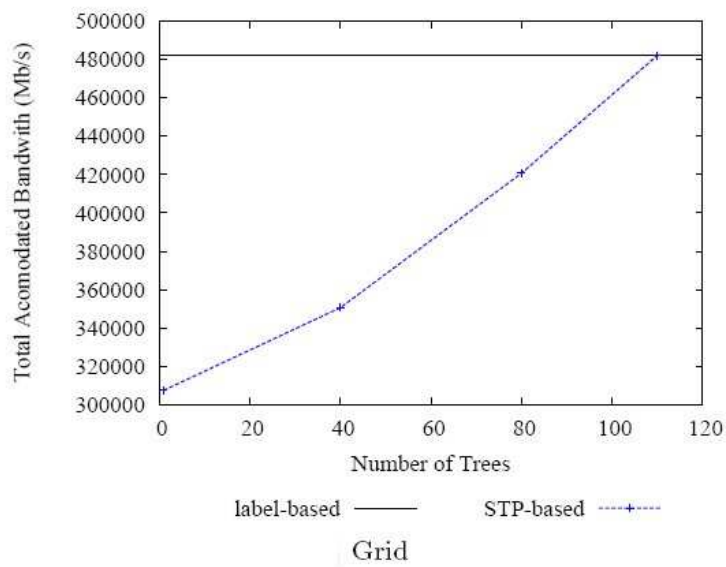
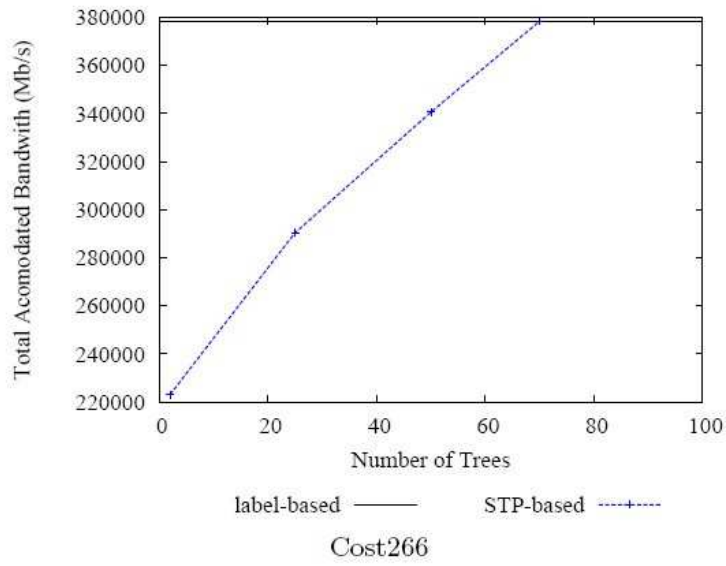


Figure 8 - Traffic accommodated for offline scenario.

6 Inter-domain Traffic Engineering for balanced network load

6.1 Network model

6.1.1 Reference network model

Our proposal is based on the presumption that both the aggregation and access domains are, or in the near future are expected to be, based on switched layer 2 (L2) technologies [12], which offer lower bit costs [13]. L2 switched networks deploy Spanning Tree Protocol variants (STP, MSTP, etc.) [14] to convey the traffic towards the core.

Figure 9 gives a picture of the reference network model developed within the CELTIC TIGER2 project [15]. The reference network [16] reflects the view of major service providers and vendors on the evolution of networking infrastructure and the way it will assimilate the new technologies. As seen in Figure 9, the networks are divided in three segments: the Access, the Metro and the Core. Depending of the country and/or local geographic specificities as well as the Internet Service Provider (ISP) choices, part of the sub-segments depicted in Figure 9 may be missing, but based on the current practices and medium-term forecasts, this generic model describes all networks.

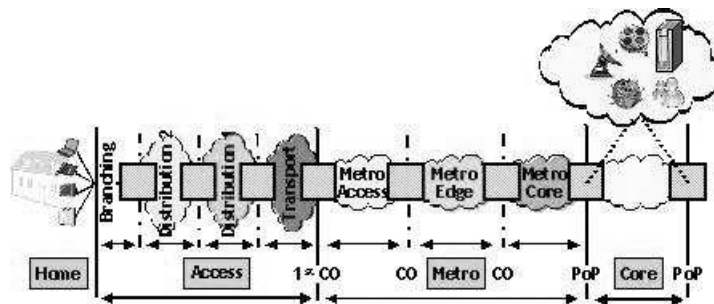


Figure 9 - TIGER2 generic network reference model.

The Access network is a local area network, and is widely studied in the literature. It connects the end users to the first Central Office (CO). Typically they have a tree-based topology, which aggregates the traffic to the COs. Core networks are also well studied and in this model we define it as the national or wider area domain. Typically they have a meshed topology. As seen in Figure 9, the metro network, which links the access to the core, is split into three sub-segments. In legacy infrastructures, these sub-segments together form a hierarchy. Access areas may be connected to any metro sub-segment by COs, and each metro is connected to the core through a Point-of-Presence (PoP).

The above model is way too generic for analysis though. The roles of the sub-segments should be specified in the context of the deployed technologies. In this paper we assume that carrier-grade Ethernet-based layer 2 technologies become dominant not only in the aggregation, but also in the access [12]. Based on these assumptions we obtain the particular reference network presented in Figure 10, derived from the generic network model [16].

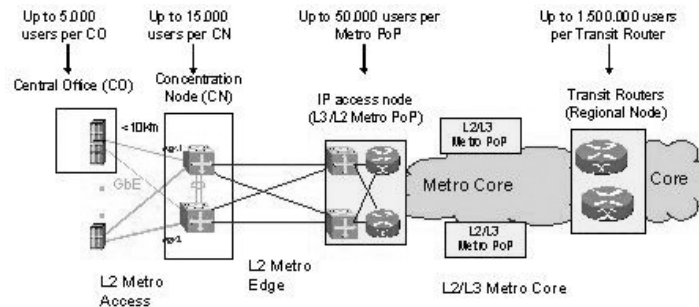


Figure 10 - Carrier-grade Ethernet in the metro as envisioned by CELTIC TIGER2

In this specific model the metro sub-segments use layer 2 (L2) switching in the access and edge, while the core deploys L2/L3 TE mechanisms. Thus, the first two sub-segments of the metro represent two successive aggregation levels of the user traffic. In the metro-access, the first aggregation level, the traffic from multiple COs (Central Office) is aggregated in Concentration Nodes (CN). This segment is based on a dual star topology with GbE or 10GbE links between COs. As seen in Figure 10, each CN aggregates up to 15 000 nodes. In the metro-edge, the second level of aggregation, traffic from different CNs is processed by a L3/L2 metro node, and the PoPs at L2/L3 boundary are handling up to 50 000 users. As a summary, we can say that metro-access and metro-edge are pure aggregation networks, while the metro core segment presents a meshed traffic distribution.

6.1.2 Core network models

A specific core network model has been proposed [16], starting from current ring topologies, widely deployed in optical networks. The model is a Double Rings with Dual Attachments (DRDA) and it can be used in core networks. In such topologies two rings, (the inner and the outer metropolitan rings) are interconnected in such a way, that every node in the outer ring is directly connected with its associated node in the inner ring, via double links (dual attachment). These provide high connectivity and multiple back-up paths for restoration purposes while reusing current network fiber deployments.

6.1.3 Investigated network topologies

Based on the reference networks presented in the previous section we designed a network that was used for our simulation based investigations, and its topology is presented in Figure 11 (left). This network is divided in two main parts, an aggregation network using Multiple Spanning Tree Protocol (MSTP) [14] and a core part with Constraint based Shortest Path First (CSPF) TE [10].

The traffic sources are depicted on the left-most part of the figure, the aggregation network conveys the packets to the core network. At the boundary we have only three edges. In real-life networks the number of edges is kept as low as possible for reasons of costs. The network has six destination nodes (sinks) represented by the exit points of the core network on the right side. The main function of the aggregation domain is to channel end-user traffic towards the core, thus its nodes are connected to two neighboring devices, at most.

The core domain has a meshed topology, with a 3 hop shortest distance between the ingress and the egress. The nodes of the core have a degree of connectivity of 3 or 4. This is a trade-off between cost effectiveness and the assurance of alternative paths. The aggregation domain

uses Ethernet-switched technology, and the core uses WDM extended with an electronic control layer.

Apart from investigating the efficient network capacity usage and balanced load of the core domain, we also investigated the possibility to minimize the operations in the electronic layer and the usage of longer optical paths. These last two parameters are characteristics of the dual opto-electronic models.

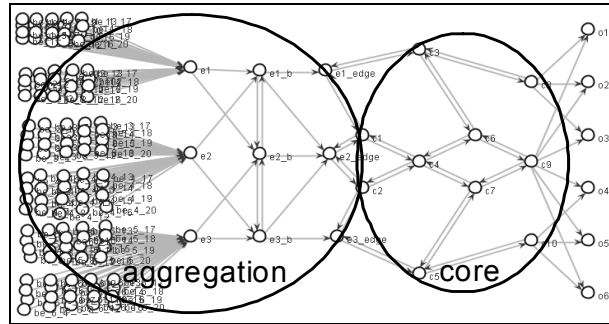


Figure 11 - Topologies of aggregation and meshed core

Our proposal supposes that the domains have a control plane that apart of running TE and other control functions are capable of communicating/cooperating with the control planes of the neighbouring domains. Such a control plane model is the Knowledge Plane [23] that can use MSTP in the aggregation and CSPF in the core domains.

We will also investigate the behavior of the core if it deploys a dual ring topology . We will build that topology in such a way as the edge nodes and the output nodes from the previous topology will be kept in order to use the same aggregation network and to be able to compare the two results.

6.2 The proposed solution: Inter-domain TE cooperation

Our proposal is to use shared intelligence between control planes, where the core intra-network functions are unchanged and only the inter-network control planes co-operate which enhances the performance.

In Figure 11 the traffic reaches the core network through the aggregation domain. In case of any event (congestion on a link, link failure, etc.) the classical TE works with the assumption that the traffic matrix remains unchanged and it has to re-distribute the traffic volume relying on load redistribution inside the core. Our proposal is to use the Knowledge Plane and rearrange the input traffic distribution outside the core edge routers. This means that –from the point of view of the core– we change the traffic matrix, since the load on the edges will be different.

Let us take the topology presented in Figure 11. Now in the situation when the aggregation domain directs all the traffic to the e1_edge (the “northern” one), while e2_edge (the “middle” edge) and e3_edge (the “southern” edge) do not feed any traffic to the core. This is the worst case situation to overload the core and corresponds to the situation when only the tree rooted in e1_edge is used to collect the traffic in the aggregation domain. Now, if we take the opposite situation, when we use each of the trees in the aggregation domain to forward the same amount of traffic, then the aggregation domain distributes the traffic evenly among the three ingresses. In this case all regions of the core will be evenly loaded.

It is the task of the Knowledge Plane to map the traffic sources among the trees. In our simulations we used small individual flow throughputs. Each tree is collecting such individual demands and the sum of these represents the traffic load at the edges. Practically the granularity of the traffic is small enough to allow us to finely balance the load. In what follows we will use the term load balancing as the operation of load redistribution in the aggregation domain as described above. The goal of load balancing will be to decongest a certain area of the core network with a minimal redistribution of the original load.

6.3 Proposed simulation model

6.3.1 Traffic model

During the simulations the traffic flows originated from the sources have the same bandwidth. We consider that we know the traffic matrix and the paths in the core are computed by a PCE using CSPF protocol. Additionally we will generate background traffic, as well, which enter the core at the edge nodes and sink on the most right-hand side destination nodes. The links of the core networks will have 200 Mbps capacity, which defines the load region where the core network is congested, but not overloaded of 400 Mbps to 800 Mbps.

In our investigations we will use the `e2_edge` node where we directed all the traffic and tried to serve it using CSPF. The resulting paths are called the main branch. If the demand is high enough, the traffic demand cannot be served. If we will apply our solution to this situation that means that some part of the traffic will be shifted to the other two edges, `e1_edge` and `e3_edge`. The paths that follow the flows entering on these two edges are called secondary branches.

We will use the background traffic to “fill” the network up to the point where congestion might start to develop. We will send 200 Mbps background traffic on the main branch. Then we will start to add new traffic demands until we will reach the total one, which will be set differently from case to case: all our simulations are planned to be run with the 500Mbps, 600Mbps, 700Mbps and 800Mbps total traffic demand. These are the situations when we can test the usefulness of our proposal and evaluate its impact on the efficiency of the opto-electronic core transport.

We use a flow level simulator, already used for the research of opto-electronic networks [24]. We generate individual flows, and the sum of these demands result in the overall traffic demand. Each link is divided into lightpaths of 10 Mbps capacity. This results in 20 lightpaths within each link that offers enough flexibility for multiplexing the flows within the core. Based on earlier work with the simulator we opted for 12 individual flows per lightpath, resulting in a flow capacity of 0.83 Mbps.

Within each scenario –that is for different overall traffic demands– we have simulated several sub-cases, where the load of the main branch was gradually re-distributed among the secondary branches. At first we started with the situation where 30% of the traffic was entering at edges `e1_edge` and `e3_edge` (15% on each of them). From there on, we stepwise directed more and more traffic towards to the secondary branches while the network was able to carry the traffic without loss. In order to be sure on that, we also simulated the next step following this point. The individual flow demands were scheduled randomly. For each situation we run ten simulations and averaged the results.

6.3.2 Spanning Trees in the aggregation domain

In order to get the input traffic at the ingresses of the core domain, we had to build the trees that bring the traffic to the edges of the core. For this, we had to build the set of trees that form the basis of the MSTP operation. We used a combination of the TOTEM [25] and BridgeSim [26] tools to simulate these trees.

With the combination of these two tools we could use the topologies created in TOTEM and apply the STP formation protocol implemented in BridgeSim. We generated all the spanning trees that can potentially be used in our scenarios, that is, all the trees that are rooted in one of the three edges and the traffic sources are their leaves. Figure 12 presents the tree generated for the northern edge, the other two trees have a similar shape. The actual traffic distribution among the three trees is decided according to our solution and should be enforced by the Knowledge Plane, as mentioned earlier.

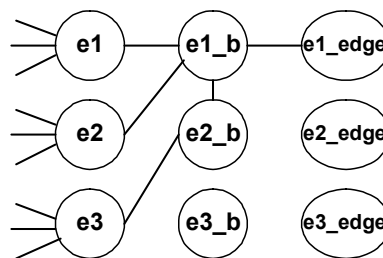


Figure 12 - The STP rooted in node e1_edge.

We will not investigate the behavior (delay, blocking, packet loss, etc.) of the aggregation domain, only used it to generate the MSTPs and determine the input traffic for the core domain. In our investigations we will be interested only about the traffic distribution among the three ingress nodes.

6.4 Planned work

In this document we have presented in detail the motivation and the envisaged environment in which our proposal on inter-domain cooperation for improved TE of the core networks would help the operators to increase the efficiency of their networks. We presented our solution and we prepared a simulation model, defined the network topology based on the TIGER2 reference network. We defined the traffic model, and after a set of preliminary simulations we identified those traffic conditions, which simulated in detail will help us evaluate our proposed solution.

In the future steps we plan to conduct the above mentioned simulations. Additionally we plan to prepare a core network topology that conforms to the DRDA model, as mentioned at the end of Section 6.1, and compare the performance of our solution in the two different networks. The results will be reported in D41-2.

7 Traffic Analysis for the Knowledge Plane

Traffic analysis of network segments is an effective method to reveal suboptimal configuration, hidden faults and security threats. If the analysis results are promptly acted upon, improvements in service quality are experienced by both the network operator and the end-user. The concept of the Knowledge Plane (KPlane), and later the Monitor Plane (MPlane) has been introduced to support Autonomous Networking goals. The tasks of processing the network element-, service-, and traffic-information belong to the MPlane. It feeds the KPlane with valuable information, based on which configuration changes are actuated. Although the concept of KPlane is widely used in various levels of network and service management, general traffic analysis is not yet utilized to support decision making procedures. Traffic mix and traffic matrix analysis results are of major interest in the decision making process at the KPlane.

7.1 Problem Statement

The optimization of network and service resources and the maximization of end-user experience are not necessarily conflicting terms. The reason for such belief lies in the fact that current network operators and service providers lack of up-to-date, usable information on their traffic. The questions of “how much” of “what” actually are traversed on the various network segments, where is that traffic “originated from” and where is it “distributed toward” are rarely answered.

According to the main argument of [21], the users and the operators suffer from the lack of a serious, purposeful optimization effort in the traditional Internet. The transparent core has no knowledge about the data transported, and even if the intelligent edge nodes realize that there is a problem, the core might not be aware of what should be done. The low-level decisions (at the edge) are rarely relate to the higher-level goal (of the core). On the user side this results in meeting the service level agreement only in coarse granularity: it is measured in long periods and more at a network level, rather than on a per-service basis.

The solution for gaining knowledge about network status and traffic characteristics is to gather and process such data, which then provide a basis to trigger corrective actions. The authors of [21] suggest to handle this knowledge in the Knowledge Plane (KP), an abstract entity that completes a triad together with Data Plane and Control Plane (see Figure 13).

In the original KPlane concept, the input is taken by *sensors* and the output is given by *actuators*. A practical variation of this architecture, detailed in [22], splits the KPlane into *monitoring plane* and *knowledge plane*. The separation of those is an obvious step: the actual “network monitoring units” (sensors) that capture and pre-process traffic data represent the “monitoring plane”, similarly as depicted by Figure 13. There are further variations and additions to this architecture; we will review these in the section of Related Works, together with a short review of decision making methodologies and practical examples from the field.

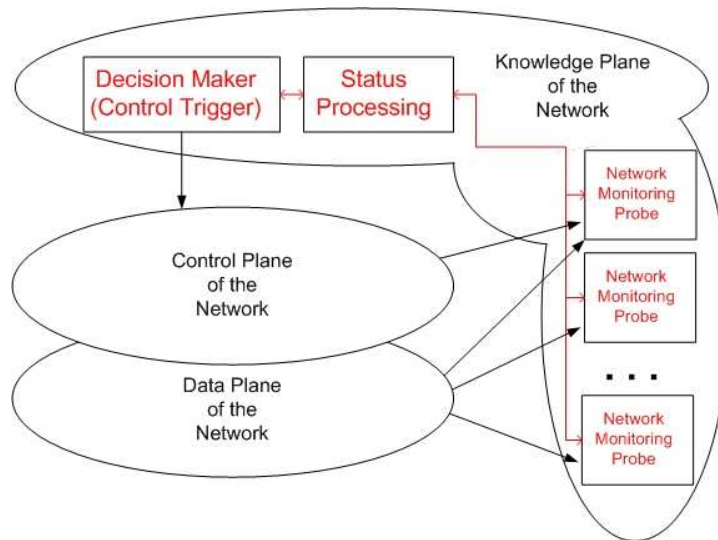


Figure 13 – Functions of the Knowledge Plane and its connections to Control and Data Planes

Figure 13 depicts the relation between the Knowledge, Control, and Data Planes. The probes/sensors take data from both the control and data planes, and report pre-processed information for the status processing module, where further analysis takes place. The *actuator* in the model is the decision maker module, which provides triggers for the control plane, completing the self-management cycle.

The main source of “knowledge” is the actual traffic of the Control and Data Planes. Although some traffic characteristics can be gathered by analyzing the Control Plane messages, many important applications – such as Peer-to-Peer (P2P) downloads, Video Streaming, or interactive voice – hide their control messages, hence their identification is only possible through Deep Packet Inspection (DPI) of the traversed traffic. The aim of Traffic Mix analysis is to determine the distribution of volumes for services and applications utilizing the network. Similarly, Traffic Matrix analysis provides results about traffic volumes – and if possible, further characteristics – broken down by route directions.

It is clear that the concept of Knowledge Plane is widely used in various levels of network and service management. Nevertheless, general traffic analysis is not yet utilized in order to support decision making in the KPlane. In the following sections we describe the suggested management architecture, traffic analysis concept and two methods to extract valuable information about the traffic mix and the traffic matrix.

7.2 Solution

7.2.1 The Monitor Plane

We follow the architecture suggested in [23] (see Figure 14), and closely examine the functions and requirements of the Monitor Plane. This function is crystallized at the original definition of autonomous networks, in [25], defining the foursome of “Monitor-Analyze-Plan-Execute” (MAPE) functions. The core function of the MPlane is to provide complete and detailed view of the network and its services. Probes at every element (access nodes, routers, switches, content servers, links, etc.) monitor the element status as well as traffic parameters.

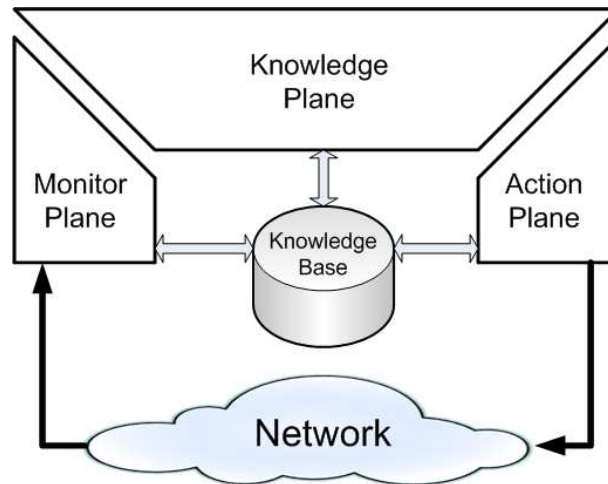


Figure 14 – The original Knowledge Plane concept extended with Monitor and Action Planes

Although built-in probe modules seem convenient, passive probing is more desirable. Active network elements (such as routers or switches) keep their processing priorities to their main job, occasionally leaving the Knowledge Base without information. These occasions of degradation in the status reporting function happen at the worst time from the KPlane's point-of-view – for practical reasons. It gets degraded at the time when the element is getting overloaded. Coincidentally, such detailed reports of overloading would be the most beneficial for the KPlane. This is why passive probing is more desirable to gather information on these elements.

After capturing the raw data, processed, grouped, and filtered traffic information gets inserted into the Knowledge Base by the probes. Both packet- and flow-level analysis reveal important characteristics on losses, delays, and jitters in the traffic, routing specialties, network structure changes and violations of the SLS (Service-Level Specification).

We are focusing on gathering these characteristics by passive monitoring. In the following subsections we briefly describe the basic requirements and mechanisms enabling this method.

7.2.2 Basic Functions of the Probes

The inevitable function of the network monitoring probes is catching, filtering, and preprocessing the traffic. These tasks should be completed for the whole network. Since installing and maintaining such a monitoring network could be an enormous effort for the operator, introducing the MPlane at the highest aggregation parts (i.e. monitoring the fastest links) can be a good decision. Monitoring these relatively few points allows gathering all packets that traverse the network, although some locally looping traffic could be left out of the analysis.

The probes should have the following crucial functionalities:

- Creating timestamps for the packets. Time-stamping done by hardware (firmware) facilitates much more precision than by software, since it avoids possible latencies due to the operating system.

- Filtering on hardware level. High-speed traffic (i.e. currently 10 Gbps or above) presently allows no option for on-the-fly filtering in software. Clearly defined, low level filters are very useful: they can dramatically decrease the data to be analyzed.
- Truncating incoming packets. For the majority of the network analysis functions, statistics-counting, or fingerprint analysis, it is not necessary to use the whole IP packet, but the first portion of it. A practical example is truncating at 128 bytes, which keeps TCP and IP headers as well as the beginning part of application headers that are helpful for identification, since it contains fingerprints for P2P or video.
- Traffic processing. The main traffic processing functionalities are briefed in the next section.
- Encapsulation and presentation of preprocessed data. The traffic analysis results must be structured and packed when passed over to the Knowledge Base.

7.2.3 Traffic Processing

The time-stamped, filtered, truncated packets, must be processed in order to reveal network and service statuses. Depending on the traffic volume, and the depth of the analysis, this processing can be fed into one or many processors. In order to keep up with the ever increasing traffic and the demand for complex analysis, the processing system must be highly scalable. As discussed earlier, monitoring core links has the advantage of utilizing all through-traffic (that traverses the network), although it requires equipment being able to monitor high-speed (currently 10 Gbps Ethernet) links without frame loss.

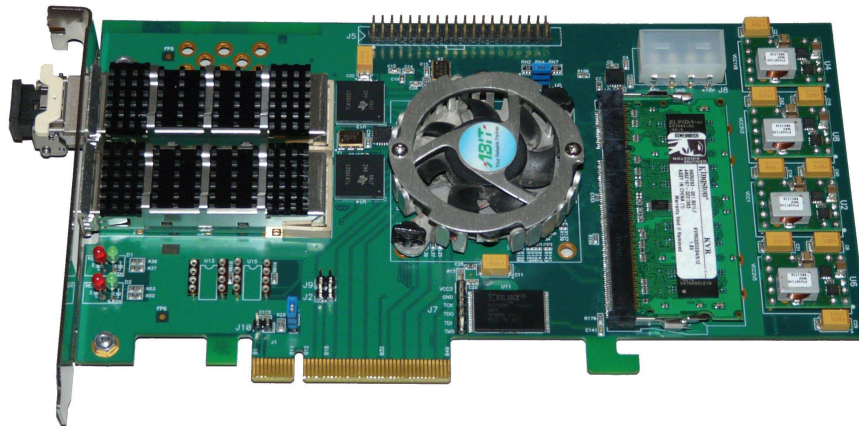


Figure 15 – A 10 G Ethernet-capable, lossless network monitoring card, SGA10GED

For low analysis demand (when one CPU can deal with the challenges), a highly reliable monitoring card, such as SGA10GED can be used to capture the traffic. It fits into a PCI slot of an industrial grade PC, where it captures, timestamps, filters, and truncates packets before passing it to the main CPU where Traffic Analysis is performed. Figure 15 shows an SGA10GED card.

In cases where on-the-fly, complex analysis is required on highly utilized links, the SCALOPES C-board is a highly scalable solution¹. It is a standalone, FPGA-based hardware,

¹ The C-board has been developed as part of the ARTEMIS SCALOPES project, partially funding our research.

equipped with 2x 10 Gbps Ethernet interfaces and 16x 1 Gbps Ethernet interfaces. When used as part of the Monitor plane, it is also preprocessing the packets, but rather than passing their data to one CPU, it distributes them among many monitor units through its 1 Gbps Ethernet Interface. The standalone Monitor Units then carry out traffic analysis, and present the results to the knowledge base. Figure 16 depicts such a scenario. Detailed description of this system can be found in [26].

The distinct analysis tasks – such as flow separation, application identification, QoS-related parameter calculation per flow/application/route – are managed by separated modules, so the parallel tasks can be run on distinct processors in the same time. Moreover, the inactive modules can be turned off to save power.

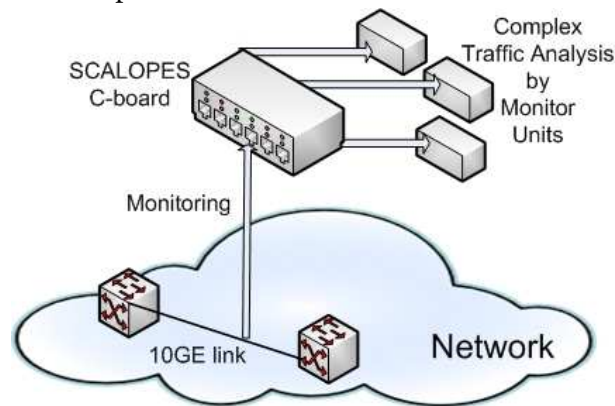


Figure 16 – A scalable solution for Traffic Analysis of high-speed network links

The tasks of the monitor units in this architecture are the following:

- collect and decode all the incoming information continuously (in 7/24 manner)
- check filtering rules predefined by the network operator, execute conditional controlled orders/commands (conditional packet saving, alarming)
- structured data storage (raw data, statistics, assays, alerts)
- generation of packet- and flow-level counters on volume, loss, delay, jitter
- generation of specialized traffic reports, such as traffic mix and traffic matrix
- database handling, remote access/query (Remote Capture, Session/Flow Trace)

7.2.4 Decision Making

Since processing of network status is continuous at the KPlane, and faults/attacks may happen at any time, so decisions on corrective actions have to be made on-the-fly as well. The Action Plane should be notified (instructed) about these actions for execution. Although the accuracy of decision making process is the key, it is limited by the variety of the input information – which is in this case merely traffic-related. Beside the accuracy, speed is also a key factor.

In order to understand the complexity of the decision making problem, a short review the main challenges are necessary. Clark et al. [21] points out three significant issues that needs to be addressed by the Knowledge Plane.

- The KPlane needs to operate in the presence of incomplete and inconsistent information, with the possibility of even misleading or malicious pieces of data.
- The KPlane needs to be able to handle conflicting or inconsistent high level goals.

- The KPlane needs to be general and future proof, i.e., the introduction of new technologies and novel applications should be possible. Moreover, the environment in which optimization needs to take place is highly dynamic, where both short and long term changes are possible in the structure and complexity of the network system.

Such challenges are not uncommon in the research and applications of the last decades of Artificial Intelligence (AI) literature. In particular, multi-agent systems (see[33]) are often proposed to handle such challenges. A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents, where intelligent agents, shortly put, mean autonomous decision making entities with individual information processing capabilities and individual goals. Such agents can naturally incorporate different viewpoints or goals in a system and also provide a natural way to embody components with different levels of data access. As a consequence, however, the goals and actions of agents in a multi-agent system may partially be aligned or conflicting. Also, even if conflicts are missing or resolvable, information may be unevenly distributed among the agents. Therefore, agents interact and try to resolve conflicts and collaborate according to various protocols and methods. A vast body of the recent AI and MAS literature deals with conflict management, collaboration and cooperation, and distributed optimization in such systems (see [34],[35], and [36]).

It is worth pointing out that the agent metaphor is a natural abstraction layer to describe conflicting or inconsistent goals – independent of the particular problem at hand. This is also true for matters of trust (c.f., malicious information). This way, these issues can be handled by general solution methods and need not be developed for each particular application domain. In other words, these challenges of the Knowledge Plane may be handled by “canned solutions” developed in other research domains.

Multi-agent systems are often said to provide a solution for the introduction of novel applications as well. The idea behind this proposal is that if a new application or requirement appears, a new agent (or bunch of new agents) may be introduced to the system at any point in time. With the general conflict resolution and collaboration protocols in place, the new goals and requirements represented by the new agents will be seamlessly integrated in the system. Similarly, should some of the goals rendered outdated by time, the sets of agents can be gracefully eliminated from the multi-agent system.

Still, in order to proceed towards a decision making solution in the autonomous networking field, further research is required. Although recent AI-related research should be exploited in the area of network management, currently there are no real-time, scalable solutions available. The canned multi-agent solutions have not yet break into the network management field, and the few prototype systems (e.g. the one described in [37]) remained prototypes up to now. Due to the above mentioned limiting factors, a scalable, high-performing, yet less accurate solution is suggested for decision making: rule-based reasoning. It is used with success in many areas; see [24] as an example. In connection with the KPlane, we continue future research in the AI-field, and further developments and integration toward a scalable, rule-based reasoning engine that is applied in a distributed manner throughout the KPlane.

7.3 Preliminary Results

7.3.1 Traffic Matrix Calculations

Traffic Matrix is a network planning and development tool. During Traffic Matrix analysis, basic QoS statistics are periodically created on flow-level, and matched to originating and destination routes, network segments, or endpoint pairs (such as IP address(-range) pairs, MPLS tunnel endpoints, etc.). The first step of the analysis is determining the flows by an n-tuple (i.e., “5-tuple”: from-IP, to-IP, from-port, to-port, protocol), and building/refreshing the flow-database. Once the targeted data structure is clarified, the algorithms of Traffic Matrix calculation are of low complexity. Such algorithms are described in [27]. The result of the measurement can be used to display periodical statistics that support network planning or service marketing activities.

The actual Traffic Matrix can easily contain endpoint-pairs in the magnitude of 10^5 . It is challenging to display such huge amount of data in a way that humans understand. While the raw results should be made available for reference in the Knowledge Base, some kind of data grouping should also be applied for visual presentation. One example of a good solution is to group the matrix elements into network segments, based on their destination addresses. The aim of the grouping algorithm is never to display high, invisible amount of segments (e.g. more than 15). When the operator wishes to peek inside a segment’s statistics, he/she get it displayed as a deeper layer of the matrix. This way the calculated QoS parameters show up in an aggregated manner in the segment-to-segment relation. If the system allows manual definition of segment-creation rules, operators can gather valuable information by grouping their endpoints into various segments. An example screenshot from a solution integrated in our system is shown by Figure 17.

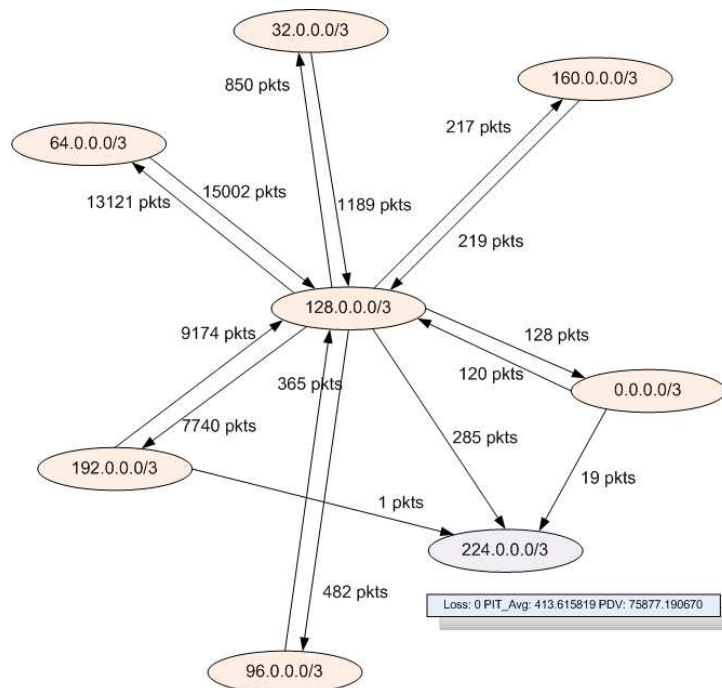


Figure 17 - Screenshot of a Traffic Matrix visualization application ([28])

7.3.2 Traffic Mix Statistics

Traffic mix analysis is the classification of traffic flows into application types, and then evaluating these for the service parameters important for the given application type. Flows are classified by means of statistical indicators and, if necessary, behavior heuristics. The most important flow types include video stream, video conference, or simple download of videos, audio stream, VoIP, and peer-2-peer.

An application belonging to a traffic-class can be identified by using static identifiers (e.g. port-based), dynamic identifiers (e.g. changing ports, fingerprints) or by applying packet-level statistics-based evaluation methods (i.e., Naïve Bayes). Powerful identification methods for VoIP, video and p2p applications are described in [29], [30], and [31] respectively. We used these methods successfully during the WP2 – see [32] for details.

Once a traffic flow is identified (i.e., based on 5-tuple), various metrics are calculated in order to help identifying the traffic-class. These metrics are the following:

- throughput: transferred data bytes per second,
- packet loss: the rate of received packets and total transmitted packets in a given time interval, or during the connection,
- packet delay: depending on the network topology and link load it takes a certain amount of time to receive a packet after it was sent; there is also a gap (delay) between packets on the wire,
- jitter: network load is not always static: as conditions and usage changes over time, packet delay changes as well - this is called jitter,
- round-trip-time: interactive applications require fast replies, which can be characterized with this parameter,
- out of order/duplicated packets.

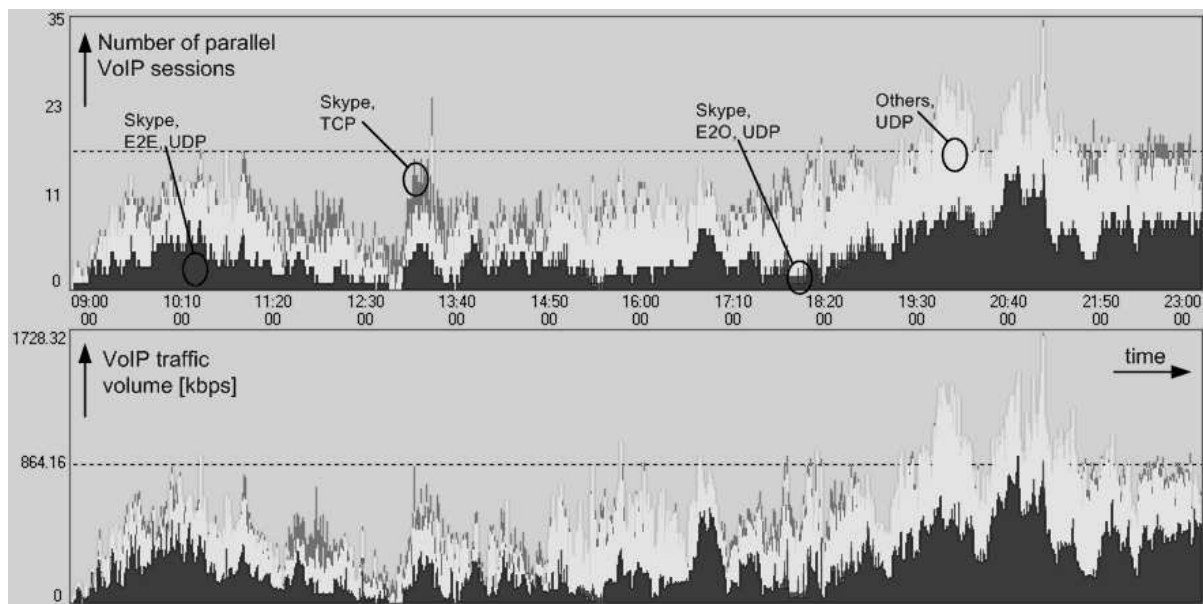


Figure 18 – VoIP Portion visualization of a Traffic Mix analysis

Figure 18 depicts a partial result of one of our measurement at a major ISP. It visualizes the number of parallel VoIP sessions (upper diagram) and the traffic volume (in kbps). The different kind of VoIP traffic are represented with different colors, which are - from bottom to top - a) Skype over UDP, end-to-end; b) Skype over UDP, end-to-office; c) other type of VoIP, d) Skype over TCP.

8 Conclusion

This deliverable presented the solutions to the study cases that have been identified in the scope of TIGER2 WP4 activities. Figure 19 recalls the list of these study cases and highlights the characteristics of the proposed solutions in terms of self-* features. As mentioned in the introduction, an updated version of this deliverable (D41-2) is planned and shall provide further evaluation results regarding these proposals.



Figure 19 – Map of TIGER2 WP4 study cases ²

The following briefly describes the current status of each study case, as presented in this deliverable, and the planned further activities that will be reported in the next version of the deliverable:

- *Hitless maintenance*: The detailed solution and preliminary evaluation results are provided. Performance evaluation results, based on the implemented prototype [7], are planned for the next deliverable release.
- *Greening TIGER2 architecture*: the problem formulation regarding the energy-aware routing study is provided. The design analysis of green routing algorithms and the evaluation of the achievable energy savings that such mechanisms could allow are planned for the next deliverable release.
- *LOCARN*: The detailed specification of the LOCARN architecture is provided. D41-2 shall include a study on adaptive probabilistic flooding for multipath routing aimed at improving the performance of the LOCARN architecture.

² The study case on "Adaptive control of Path Computation Elements" has been kept in this map. On the other hand, the study case on "Traffic analysis for the knowledge plane" has not been reported in this map as it represent a common functionality possibly required by several self-* solutions.

- *Self-optimization of network resource allocation in MSTP*: the proposed solution is described and evaluated. This study case is completed.
- *Inter-domain traffic engineering for balanced network load*: the proposed solution and simulation model are presented. The obtained evaluation results are planned for the next deliverable release.
- *Traffic analysis for the knowledge plane*: the solution design as well as traffic mix calculations and statistics are provided. This study case is completed.

9 Annex A: Generic LOCARN Frame Formats

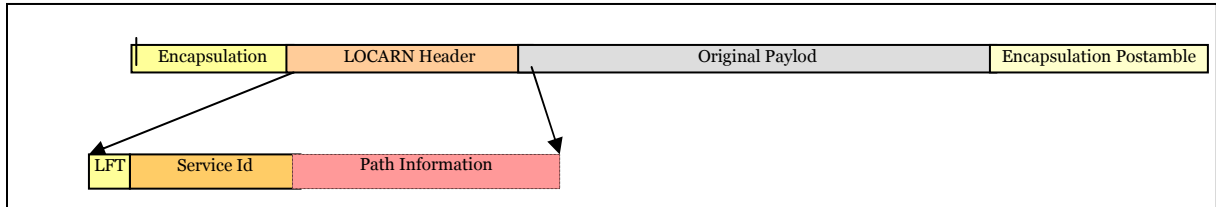


Figure 20 - Generic LOCARN Data Frame structure

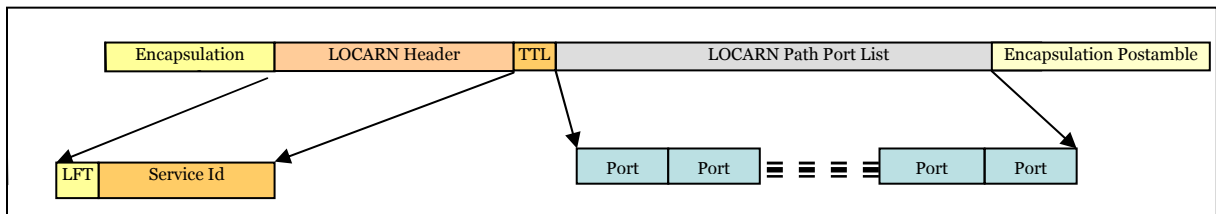


Figure 21 - Generic Path Request LOCARN Frame structure

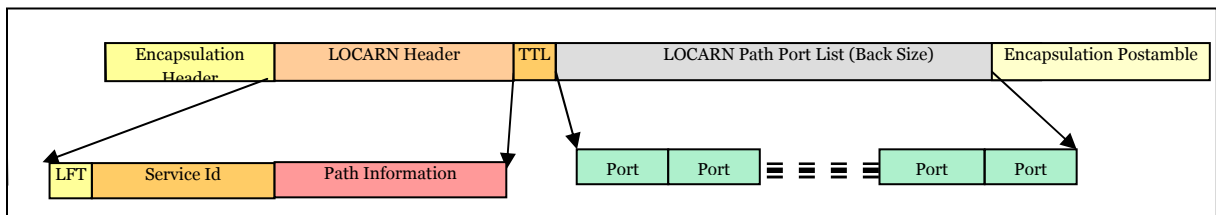


Figure 22 - Generic Path Discover LOCARN Frame structure

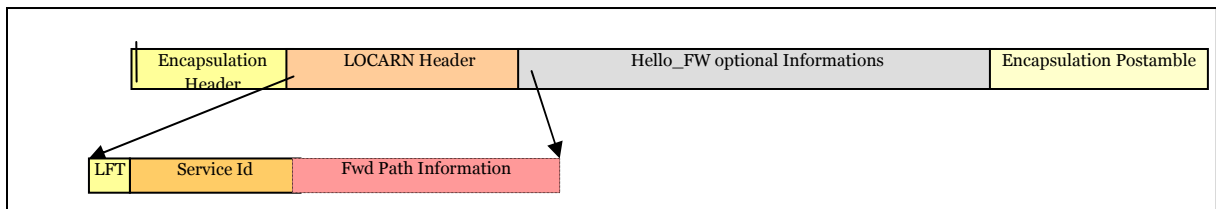


Figure 23 - Generic Hello Forward LOCARN Frame structure

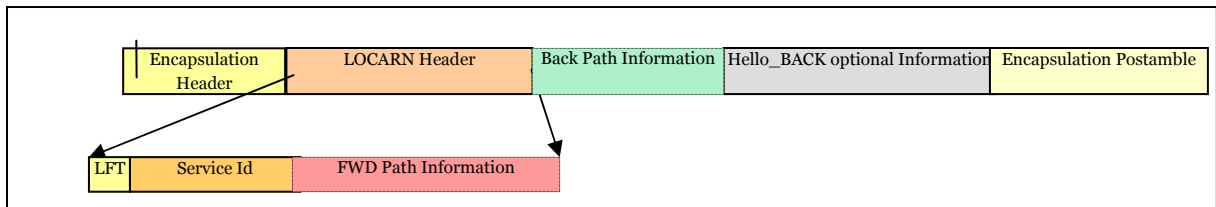


Figure 24 - Generic Hello Back LOCARN Frame structure

10 References

- [1] Z. Kerravala, "Enterprise Networking and Computing : the Need for Configuration Management," Yankee Group report, January 2004.
- [2] J. Moy, P. Pillay-Esnault, A. Lindem, "Graceful OSPF Restart," IETF RFC 3623, November 2003.
- [3] M. Leelanivas, Y. Rekhter, R. Aggarwal, "Graceful Restart Mechanism for Label Distribution Protocol," IETF RFC 3478, February 2003.
- [4] M. Kubale, "Graph Colorings," American Mathematical Society, 2004.
- [5] D. J. A. Welsh, M. B. Powell, "An upper bound for the chromatic number of a graph and its application to timetabling problems," *The Computer Journal*, vol. 10, no. 1, pp. 85–86, 1967.
- [6] D. J. A. Welsh, M. B. Powell, "Measuring ISP Topologies With Rocketfuel," *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, pp. 85–86, February 2004.
- [7] The TIGER2 D51 Deliverable, "Final report on the emulation platform," July 2010.
- [8] The TIGER2 D40 Deliverable, "Network control and management evolution towards autonomic networking including study cases definition," April 2010.
- [9] Barroso, Luis André and Hölzle, Urs. The case for energy-proportional computing. *IEEE Computer*, 40(12):33–37, December 2007.
- [10] L. Chiaraviglio, M. Mellia, and F. Neri. Reducing power consumption in backbone networks. In *Proceedings of the IEEE International Conference on Communications*, June 2009.
- [11] M. Gupta and S. Singh. Greening of the internet. In *ACM SIGCOMM*, pages 19–26, August 2003.
- [12] Qureshi, Asfandiyar, Et al. Cutting the electric bill for internet-scale systems. In *ACM SIGCOMM*, August 2009.
- [13] The GEANT Project. <http://www.geant.net/>.
- [14] Michal Pioro, Deepankar Medhi, "Routing, Flow and Capacity Design in Communication and Computer Networks". San Francisco, CA: Morgan Kaufmann, 2004.
- [15] Jian Qiu, Gurusamy Mohan, Kee Chaing Chua, Yong Liu, "Local restoration with multiple spanning trees in Metro Ethernet", *Proceedings of the 2008 International Conference on Optical Network Design and Modeling (ONDM 2008)*, March 2008.
- [16] Luis F. Caro, Dimitri Papadimitriou, Jose L. Marzo, "A performance analysis of carrier Ethernet schemes based on Multiple Spanning Trees", *Proceedings of the VIII Workshop in G/MPLS networks*, June 2009.
- [17] Jian Qiu, Gurusamy Mohan, Kee Chaing Chua, and Yong Liu. Local restoration with multiple spanning trees in metro ethernet. *Optical Network Design and Modeling, 2008. ONDM 2008. International Conference on*, pages 1-6, March 2008.
- [18] S.M. Ilyas, A. Nazir, F.S. Bokhari, Z.A. Uzmi, A. Farrel, and F.R. Dogar. A Simulation Study of GELS for Ethernet Over WAN. *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pages 2617-2622, 2007.
- [19] R. Inkret, A. Kuchar, and B. Mikac. Advanced infrastructure for photonic networks european research project. In *Extended Final Report of COST 266 Action*, ISBN 953-184-064-4, 2003. p. 21.

- [20] D. Associates. Xpress-Mosel Reference Manuals and Xpress-Optimizer Reference, Manual. Release 2004G, 2004.
- [21] Clark, D.D., Partridge C., and Ramming, J.C., "A knowledge plane for the Internet", In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, August 25-29, 2003, Karlsruhe, Germany
- [22] De Vleeschauwer, B., Van de Meerssche, W., Simoens, P., De Turck F, Dhoedt, B., Demeester, P., Gilon E., Struyve, K., Van Caenegem T., "On the Enhancement of QoE for IPTV Services through Knowledge Plane Deployment", In Proceedings of Broadband Europe, December 11-14, 2006, Geneva, Switzerland
- [23] Latre, S., Simoens, P., Vleeschauwer, B.D., Van de Meerssche, W., De Truck, F., Dhoedt, B., Demeester, P., Van Den Berghe, S., Gilon, E., "Design for a Generic Knowledge Base for Autonomic QoE Optimization in Multimedia Access Networks", In Proceedings of 2nd IEEE Workshop on Autonomic Communications and Network Management, April 2008, Salvador, Brazil
- [24] Varga, P., Moldovan, I., "Integration of Service-Level Monitoring with Fault Management for End-to-End Multi-Provider Ethernet Services", IEEE Transactions on Network and Service Management, Vol.4 No.1, 2007
- [25] IBM, "Architectural Blueprint for Autonomic Computing", 2003
- [26] Plosz, S., Moldovan, S., Varga, P., Kantor, L., "Dependability of a Network Monitoring Hardware", In Proceedings of DEPEND 2010, Venice, Italy
- [27] AITIA, BME, "Report on New Architectural Platform and Specification of Example SW Code for Analysis", ARTEMIS SCALOPES Deliverable DA1.3., 2010
- [28] Szendrei, G., "Calculation and visualization of Traffic Matrices", Technical Report, BME-TMIT, 2010
- [29] Bonfiglio D., Mellia, M., Meo, M., Rossi, D., Tofanelli, P., "Revealing Skype Traffic: When Randomness Plays with You", In Proceedings of SIGCOMM, Japan, 2007
- [30] Varga, P., Kovacs, L., Moldovan, I., Illes, A.Cs., Kun, G., Sey, G., Turzo, G., "Analysis of Media Communication over the Internet", Technical Report for Hungarian Telecom, 2007
- [31] Karagiannis, T., Broido, A., Faloutsos, M., Kc claffy, "Transport Layer Identification of P2P Traffic", In Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, Sicily, Italy, 2004
- [32] Dorgeuille, F., Varga, P., Betoule, C., Thouenon G., Petitdemange, G., Palacios J.F., "Rationales and scenarios for investigations on next generation of access, backhauling and aggregation networks", CELTIC TIGER2 Technical Report, 2009
- [33] Wooldridge, M., "An Introduction to MultiAgent Systems", John Wiley & Sons Ltd, 2002, ISBN 0-471-49691-X
- [34] Lander, S.E., "Issues in multiagent design systems", IEEE Expert, April 1997
- [35] Tessier C., Chaudron L., Mueller, H.J., "Conflicting agents: conflict management in multi-agent systems", Kluwer Academic Publishers, 2001
- [36] Hirayama, K., Yokoo, M., "The distributed breakout algorithms", In Artificial Intelligence, 2005, Vol. 161, pp. 89-115
- [37] Gaiti, D., Pujolle, G., Salaun, M., Zimmermann, H., "Autonomous Network Equipments", In LNCS 3854, 2006, Springer