Telecommun Syst (2016) 61:235–246
DOI 10.1007/s11235-015-9997-5

Methods of Network Resource Provisioning for the Future Internet IIP Initiative

Janusz Gozdecki · Mirosław Kantor · Krzysztof Wajda · Jacek Rak

Published online: 19 March 2015

Abstract In this paper, we present specification, design and implementation aspects of a network resource provisioning module introduced for the Polish Initiative of Future Internet called System IIP. In particular, we propose a set of novel LP optimization models of network resource provisioning designed to minimize the network resource consumption, either bandwidth or node's computational power, as well as to maximize the residual capacity. Next, we analyze characteristics of the proposed models, outline the most important implementation aspects of our network resource provisioning module, as well as the issues of integration of this module with the respective network management module of System IIP.

Keywords Network resource provisioning · Linear Programming models · Future Internet · Virtualization

J. Gozdecki · M. Kantor · K. Wajda
Faculty of Computer Science, Electronics, and Telecommunications, Department of Telecommunications, AGH University of Science and Technology, Al. Mickiewicza 30, PL-30-059 Krakow, Poland
e-mail: gozdecki@kt.agh.edu.pl

M. Kantor
e-mail: kantor@kt.agh.edu.pl

K. Wajda
e-mail: wajda@kt.agh.edu.pl

J. Rak (✉)
Faculty of Electronics, Telecommunications, and Informatics, Department of Computer Communications, Gdansk University of Technology, G. Narutowicza 11/12, PL-80-233 Gdansk, Poland
e-mail: jrak@pg.gda.pl; jrak@ieee.org

1 Introduction

We live in the era of globalization with access to information provided at any time, and location by means of Internet - the global information network. Our everyday life more and more depends on ability to retrieve information from Internet resources. Designed more than 40 years ago, Internet is now facing efficiency problems related to scalability and Quality of Service issues. Traffic volume grows exponentially every year, while application requirements are becoming more and more stringent with respect to QoS attributes, including bandwidth, latency, jitter, and packet losses. Computer communication experts are convinced that Internet architecture based on classical IP protocols family has already reached its development limits.

As a response, research teams from all over the world are currently pursuing their concepts of Future Internet [6,19,22]. The general idea of major ongoing projects, including e.g., Akari (Japan) [13], GENI (US) [14], or 4WARD (EU) [12], is to use the best practices from the past to design the Internet architecture from scratch. The general tendency is to make the new Internet a kind of "hyper-network", i.e., composed of different types of networks providing the required level for QoS parameters [11,21]. Special focus is thus put, e.g., on new functionalities such as virtualization, parallelization, redesign of data and control planes, as well as development of new services, all monitored by Future Internet Assembly (FIA), ITU-T, and ETSI.

One of these projects – Future Internet IIP Initiative [4,5,15] – has been recently driven by leading researchers from nine Polish universities and research centres. Referring to the novel four-layer architecture of System IIP, comprising (in the bottom-up order): L1 - physical infrastructure layer, L2 - virtualization layer, L3 - Parallel Internets layer, and L4 - virtual networks layer, it provides co-existence

of differentiated types of Parallel Internets (PIs) within one physical infrastructure, including: IPv6 with Quality of Service (IPv6_QoS), Content-Aware Network (CAN), and circuit-switched Data Stream Switching (DSS).

In this paper, we address the issue of Future Internet resource provisioning with special focus on design and implementation aspects of the L1/L2 resource provisioning module considered as a Traffic Engineering (TE) procedure of the respective management system. The objective is to assign elementary resources (such as link capacity, or node processing power) to three considered Parallel Internets (PIs) and to the management system, enabling virtualization of nodes and links [7,9].

The allocation of resources, also known as the Virtual Network Embedding (VNE) problem, is one of the main problems in network virtualization. Network virtualization allows multiple heterogeneous networks (in our case Parallel Internets) to share the same physical substrate network (SN). The term "provisioning" comprises here two mechanisms:

- efficient allocation of requested elementary resources to Parallel Internets,
- Layer 3/4 topology update being result of L1/L2 allocation of elementary resources to PIs.

Allocation of requested resources to PIs is done here periodically in a static way. Each PI request consists of two parts: a set of virtual nodes (with or without CPU requirements) that must be mapped to a set of SN nodes, and a set of virtual links (with or without bandwidth requirements) to be mapped to a set of SN paths. Additionally, the PI may impose additional constraints on link propagation delay. In the paper, we assume that virtual nodes are assigned to chosen substrate nodes before the optimization process starts.

It is worth noting that the problem of virtual networks (VN) creation by means of resource splitting has been recently well investigated in the literature. More information on virtual network provisioning with respect to theoretical, as well as computational aspects, can be found, e.g., in [10,18]. Examples of efficient resource assignment algorithms can be in turn found in [8,17,23,24].

The remaining part of the paper is organized as follows. In Section 2, we introduce Linear Programming (LP) formulations of Future Internet resource provisioning problem. Section 3 provides discussions on numerical complexity of the proposed optimization models. Section 4 is to present time-efficient near-optimal metaheuristics, while Sections 5 and 6 show evaluation of the proposed provisioning module characteristics, and aspects of implementation/integration of this module with the respective network management system, accordingly. Section 7 concludes the paper.

2 Linear Programming (LP) Models Proposed for Resource Provisioning

In this section, we describe in detail our three linear programming (LP) models proposed to solve the network resource provisioning problems implemented in the resource provisioning module of System IIP. The first one includes the generic formulations and is designed to optimize the utilization of link resources. Link capacity is typically considered as limited resource subject to rigid and precise allocation by the provisioning process.

The second model includes explicit requirements in terms of nodes resources, e.g., processing power at network nodes. The third model imposes additional constraints on the upper bound of path transmission delay necessary for certain classes of service (e.g., delay-sensitive, or real-time systems).

In these models, we assume that:

- network nodes are configured to be either core (transit/forwarding) nodes, or edge nodes,
- a traffic matrix is known in advance for selected pairs of end nodes (i.e., defining the source-destination pairs) of each Parallel Internet,
- relation between node processing power and capacity of links (see Model 2) is linear, and is typically expressed as a relation: Mflops vs Mbps.

Network topology is represented by a directed graph $G = (V, E)$, where V and E denote the sets of vertices, and links (represented by graph edges, i.e., directed arcs), respectively. Capacity of each network link (expressed in Mbps) is a continuous value.

Model 1:
Formulation with the Objective of Link Bandwidth Utilization Optimization Including Basic Requirements on Routing (LBUO)

Indices:

$i = 1, 2, 3$	instances of Parallel Internets (referring to IPv6_QoS, CAN, and DSS Internets, respectively)
$t = 1, 2, \ldots, \|T\|$	transit (forwarding) nodes
$e = 1, 2, \ldots, \|E\|$	network links (represented by graph edges, i.e., directed arcs)
$d = 1, 2, \ldots, \|D\|$	demands for each ith Parallel Internet
$v = 1, 2, \ldots, \|V\|$	all nodes
$V \setminus T$	set of edge nodes

Constants:

a_{ev}	equals 1, if node v is edge e source node; 0, otherwise
b_{ev}	equals 1, if node v is edge e destination node; 0, otherwise

c_e total capacity available at edge e
γ_{ie} lower bound (percentage) of capacity required at edge e for ith Parallel Internet
h_{id} volume of demand d for a given Parallel Internet (with index i)
s_{id} source node of demand d for a given Parallel Internet (with index i)
u_{id} destination node of demand d for a given Parallel Internet (with index i)

Continuous variables:

$x_{iev} \geq 0$ capacity allocated for ith Parallel Internet at edge e incident to node v (e.g., in Mbps)
$z_{ied} \geq 0$ capacity assigned at edge e for demand d of ith Parallel Internet

Objective:

It is to minimize the total bandwidth consumption for delivering the traffic:

$$\textbf{minimize} \quad F = \sum_i \sum_e \sum_v x_{iev} \tag{1}$$

Constraints:

$$\sum_v a_{ev} x_{iev} = \sum_v b_{ev} x_{iev} \qquad i \in I; \quad e \in E \tag{2}$$

$$\sum_e b_{et} x_{iet} = \sum_e a_{et} x_{iet} \qquad i \in I; \quad t \in T \tag{3}$$

$$\sum_v a_{ev} x_{iev} \geq \gamma_{ie} c_e \qquad i \in I; \quad e \in E \tag{4}$$

$$\sum_i \sum_v a_{ev} x_{iev} \leq c_e \qquad e \in E \tag{5}$$

$$\sum_e a_{ev} z_{ied} - \sum_e b_{ev} z_{ied} = \begin{cases} h_{id} & \text{if} \quad v = s_{id} \\ -h_{id} & \text{if} \quad v = u_{id} \\ 0 & \text{in other cases} \end{cases}$$
$$\text{and} \quad i \in I; \quad v \in V; \quad d \in D \tag{6}$$

$$\sum_v \sum_d a_{ev} z_{ied} \leq \sum_v x_{iev} \qquad i \in I; \quad e \in E \tag{7}$$

According to constraint (2), the amount of allocated capacity leaving node v via edge e for ith Parallel Internet is equal to that received by another node (i.e., at the other end of edge e). Eq. (3) refers to flow conservation rules (known as "Kirchhoff's law" constraints) for transit nodes. Formula (4) is to guarantee that capacity assigned for ith Internet instance at edge e is not less than the reqiured minimum threshold. Formula (5) assures that the aggregate capacity allocated at edge e for all Parallel Internets is not greater than the total capacity available at edge e. Eq. (6) provides appropriate forwarding of each demand d between this demand's end nodes. Finally, formula (7) guarantees that the aggregate flow transported along edge e for all demands of ith Parallel Internet does not exceed the capacity allocated at edge e for this ith Parallel Internet.

As an alternative to Eq. (1), we also utilize the other objective function given by Eq. (8) to maximize the total residual (free) capacity at all edges.

$$\textbf{maximize} \quad F = \sum_e \left(c_e - \sum_v \sum_i x_{iev} \right) \tag{8}$$

Maximization of goal function (8) is an interesting option when searching for capacity assignment, which should increase network resilience and traffic overload margin.

Model 2:
Formulation with the Objective of Link Bandwidth Utilization Optimization and Including Basic Requirements on Routing Extended by Node Resource Utilization Optimization Issue (LBNR)

This model is an extension to Model 1 (LBUO) with additional node resource utilization optimization issue. Apart from constraints related to link capacity, the model additionally includes requirements on node resources (implemented as additional constraints). In order to take into account the limits for computational resources available at nodes, there is an additional requirement for processing power at each node. Besides the processing power, also other resources like Random Access Memory volumes, mass storage, or buffers can be taken into consideration.

Indices:
Compared to Model 1, the list of indices is identical.

Constants:
Compared to Model 1, the list of constants is identical, and additionally includes the following:

θ_{iev} processing power consumption (measured per unit of capacity for ith Internet instance) defined for edge e outgoing from node v
δ_{iev} consumption of processing power (measured per unit of capacity for ith Parallel Internet) for edge e destined to node v
ξ_v node v aggregate processing power

Continuous variables:

The list of continuous variables is the same as in Model 1, and additionally includes the following:

$y_{iv} \geq 0$ resources reserved for the purpose of processing the flows of ith Parallel Internet at node v (e.g., in Mflops)

Objective:

It is to minimize the total processing power used to deliver the traffic, i.e.:

$$\textbf{minimize} \quad F = \sum_i \sum_v y_{iv} \tag{9}$$

Constraints:
Constraints (2)-(7) are valid and additionally:

$$y_{iv} = \sum_e \theta_{iev} a_{ev} x_{iev} + \sum_e \delta_{iev} b_{ev} x_{iev} \qquad i \in I; \quad v \in V \tag{10}$$

$$\sum_i y_{iv} \le \xi_v \qquad v \in V \tag{11}$$

Calculation of processing power utilization at node v considering the portion of capacity reserved for each ith Internet instance is represented by Eq. (10). Formula (11) assures that the total processing power allocated at node v will not exceed the nominal processing power available at v.

Model 3:
Extension of Model 2 (LBNR) Including Additional Transmission Delay Constraints (LBDC)

Model 3 provides additional constraints on maximum transmission delay for each stream. After finding a potential path, the total delay for this path is computed. If the delay (computed as the sum of delays of all links constituting a given path) exceeds the maximum value (limit), this path is rejected, and the next one is checked. In practice, if for a given flow there is no constraint on the maximum transmission delay, it is possible to assign the required delay limit an arbitrary large value.

Indices:
Compared to Model 1, the list of indices is identical.

Constants:
Compared to Models 1 and 2, the list of constants additionally includes the following:

f_e	the upper bound on transmission delay defined for edge e
g_{id}	the upper bound on end-to-end transmission delay for demand d from ith Parallel Internet
G	a large number chosen arbitrarily

Variables:
The list of variables is the same as in Model 2 and additionally includes the following:

n_{ied}	binary variable set to 1, if edge e forwards the traffic from dth demand of ith Parallel Internet; 0 otherwise

Objective:
Minimization of the total bandwidth consumption for the purpose of traffic delivery, defined as given in Eq. (1).

Constraints:
Constraints (2)-(7), (10)-(11) are valid and additional constraint (12) is used to assure that for each demand d from ith Parallel Internet, the end-to-end transmission delay does not exceed a given upper bound.

Constraint (13) together with constant G are necessary to bind the binary variable n_{ied} (referring to utilization of edge e by a communication path serving dth demand of ith Parallel Internet) with the respective continuous variable z_{ied} used to determine the amount of capacity reserved for this demand at edge e.

$$\sum_e n_{ied} f_e \le g_{id} \qquad i \in I \quad d \in D \tag{12}$$

$$z_{ied} \le n_{ied} G \qquad i \in I \quad d \in D \quad e \in E \tag{13}$$

The models defined above are the examples of approaches used in the System IIP project. However, it is possible to apply other formulations with specific goals, when combining different objective functions with model specifications.

3 Numerical Complexity

In this section, we introduce the proof of *NP*-completeness of the considered problem to determine provisioning of network resources in the capacity-constrained Future Internet IIP architecture. Classification of the investigated problem into the class of *NP*-complete problems means that so far there has been no algorithm proposed to find the optimal solution in polynomial time. Following [16], in order to classify the considered optimization problem as *NP*-complete, it is enough to show it for its recognition version (i.e., with "yes" or "no" answer).

Definition (optimization version for Model 1)
LBUO: Given the information on network topology, resource limitations of nodes and links, demands per each Parallel Internet, find the optimal solution to this problem for which the value of objective function determined by Eq. (1) is minimized.

Definition (recognition version for Model 1)
LBUO(h): Given the information on network topology, resource limitations of nodes and links, demands per each Parallel Internet, determine if it is feasible to obtain the value of objective function defined by Eq. (1) equal to h.

Theorem 1 LBUO(h) problem is *NP*-complete.

Proof Following [1], in order to prove that the recognition version of the LBUO problem is *NP*-complete, we need to show that:

a1)	LBUO(h) belongs to the *NP* class,
b1)	a known *NP*-complete problem polynomially reduces to LBUO(h).

Re a1)	LBUO(h) belongs to the *NP* class, since it can be verified in polynomial time whether a given solution to the problem (with the objective function

value equal to h) is a valid one. The number of operations that is required to check the validity of the solution is proportional to the aggregate number of links of all established paths, which is in turn bounded from above by $O(n^3)$, since it implies checking at most $O(n^2)$ paths, each path formed by at most $O(n)$ links, where n is a number of network nodes.

Re b1) In order to provide the second part of the proof, we will show that a common problem to determine the end-to-end routing in capacity-constrained networks, here referred to as ROUTE(h), shown to be *NP*-complete in [2], can be reduced in polynomial time to LBUO(h). In other words, providing this proof will imply showing that:

a2) the number of transformations needed to obtain the instance of the LBUO(h) problem from the instance of the ROUTE(h) problem is bounded from above by a given polynomial number of operations,

b2) finding the solution to the ROUTE(h) problem can be achieved by solving the LBUO(h) problem.

Definition (optimization version of ROUTE problem) ROUTE: Given the information on network topology, capacity constraints of links, end-to-end demands defined by triples (s_{id}, u_{id}, h_{id}), where s_{id}, u_{id}, h_{id} denote the source node, destination node, and the demanded capacity of a given idth connection, accordingly, find the optimal solution to the problem of determining the routing for all demands, for which the aggregate capacity assigned to paths on network links is minimal.

Definition (recognition version of ROUTE problem) ROUTE(h): Given the information on network topology, capacity constraints of links, end-to-end demands given by triples (s_{id}, u_{id}, h_{id}), where s_{id}, u_{id}, h_{id} denote the source node, destination node, and the demanded capacity of a given idth connection, accordingly, determine, if it is possible to utilize the aggregate capacity of h units on all network links to provide routing of demands.

In general, the difference between LBUO(h) and ROUTE(h) problems is that in LBUO(h) we additionally require that:

- demands d are grouped into classes referring to Parallel Internets,
- the aggregate amount of capacity assigned to demands of each ith Parallel Internet at each edge e is not less than the assumed lower bound (percentage) γ_{ie}.

ROUTE(h) is thus less complex than LBUO(h).

Therefore:

Re a2) to transform the instance of the ROUTE(h) problem to the instance of the LBUO(h) problem, we need to:

- assign each demand an ID referring to Parallel Internet (the respective number of steps is equal to the aggregate number of demands bounded from above by $O(n^2)$),
- assign the lower bound $\gamma_{ie} = 0$ on the aggregate capacity reserved for all demands belonging to all classes of Parallel Internet i at each network link e (the respective number of operations is equal to the product of the number of Parallel Internets and the number of links (which is bounded from above by $O(n^2)$).

Re b2) A valid solution to the instance of LBUO(h) problem, is also a valid solution to the ROUTE(h) problem, since the former one satisfies all the constraints of the latter one. In order to obtain the solution to the ROUTE(h) problem from the LBUO(h) solution, it is sufficient to find the solution to the LBUO(h) problem assuming that $\gamma_{ie} = 0$ for all Internets i at all edges e, and next remove from the LBUO(h) solution the identifiers of Parallel Internets.

In a similar way, it is easy to provide the proofs of *NP*-completeness of the other two introduced models (i.e., LBNR and LBDC, accordingly), since LBNR and LBDC are the extensions of LBUO and LBNR problems, accordingly.

Following [1], if recognition versions of problems are *NP*-complete, so are their optimization versions. □

4 Searching for Metaheuristics

Due to the complexity of the considered optimization problem, in this section, we propose an offline metaheuristic approach to map a set of PI requests. The goal of the algorithm, called RAL (Resource ALlocation), is to serve all PI requests while trying to maximize the level of spare bandwidth and spare CPU in the substrate network.

RAL Algorithm for mapping virtual links to the substrate network

INPUT: substrate network topology, PI requests with bandwidth and CPU demands associated with virtual nodes and links, and CPU demands for intermediate nodes

OUTPUT: virtual links mapped to the substrate network topology

The following steps are performed in the proposed RAL heuristic:

- ***Step 1***: Determine the input data for the algorithm, such as the substrate network topology, PI requests with bandwidth and CPU demands associated with the virtual nodes and links, and CPU demands for intermediate nodes (belonging to the path via which the virtual link will be realized).
- ***Step 2***: Sort the virtual links in the descending order, based on the amount of the required resources (bandwidth or CPU capacity). Start the mapping process from the virtual link with the highest demand volume. Map one virtual link to the substrate network at a time.
- ***Step 3***: Find the best shortest path for a single virtual link accomplishing the CPU and bandwidth restrictions. Use the *k-shortest path algorithm* for increasing values of k, until a path which has enough resources to map the corresponding virtual link is found. In case more than one shortest path is found, choose a path with more remaining bandwidth or CPU.
- ***Step 4***: If the mapping of a virtual link is not possible, e.g., due to the lack of enough substrate resources for realizing the considered demand, try to reallocate the already assigned resources.
- ***Step 4.1***: Determine the bottleneck link on the substrate network to find the required resources for the considered virtual link. To determine this link, choose one of the two possible variants of algorithm:
- ***Step 4.1.1***: Check the virtual links that were already mapped to the path including the bottleneck link. Remove from the mapping the demand that requires the most bandwidth on the bottleneck substrate link.
- ***Step 4.1.2***: Cancel the smallest possible demand mapped to the path going through the congested link the cancellation of which will enable to map the just considered virtual link.
- ***Step 4.2***: Return the resources to the substrate network and make a next trial to map the virtual link. Try to perform mapping by cancelling the already mapped demands until the virtual link is assigned or until a predefined number m of trials is reached.
- ***Step 4.3***: In case the mapping has not been done after m attempts, assume that this virtual network cannot be mapped.
- ***Step 4.4***: Serve the demands for which the mapping was cancelled during reallocation in the next turn.
- ***Step 5***: After mapping the virtual link to a substrate path, update the available substrate resources by subtracting the node and link resources associated with the just mapped virtual link from the remaining bandwidth and CPU resources.
- ***Step 6***: Select the next virtual link; repeat *Steps 3-5*, until all virtual links are considered.

Apart from the CPU demand of source and destination virtual nodes, the algorithm also takes into account the CPU requirements for intermediate nodes through which the virtual links of PIs are realized.

The motivation for such an approach comes from the fact that intermediate nodes have to be configured and have to correctly forward the packets transmitted through this virtual link, which requires spending some CPU resources.

The RAL algorithm tries to obtain the efficient utilization of the substrate bandwidth resources by mapping virtual links to the shortest paths in the substrate network, with hop count as a metric, assuring that the demands for virtual nodes and links are fulfilled. Therefore, the *k-shortest path algorithm* is used for finding the substrate path for virtual link embedding. The assumed approach is motivated by the fact that the usage of the shortest path minimizes the utilization of substrate resources. This heuristic is based on the algorithm proposed in [3] with several modifications related especially to resource reallocation problem in case the mapping of a virtual link fails.

The proposed algorithm does consider information on the type of PI to which the virtual links belong. The modification of the described heuristic is another approach where the mapping process starts from the PI with the highest revenue. The revenue can correspond with economic benefit of accepting PI requests. As the bandwidth and CPU are the main substrate network resources, the revenue associated with PI requests are calculated as the weighted sum of revenues for bandwidth and CPU. Based on the obtained revenue, the PI requests are sorted in the descending order taking into account the revenue values. Then, Steps 2-6 of the main algorithm are executed for each PI.

The computational complexity of the proposed heuristic approach is bounded from above by $O(n^4)$. This is due to the execution of the common $k\text{-}shortest$ path algorithm (of complexity $O(n^2)$) at most $3 * n * (n - 1)$ times in Step 3 (i.e., for each possible virtual link created for a maximum number of three considered Parallel Internets). In case of unsuccessful allocation of resources in Step 3, this number of attempts can be increased at most m times. However, since m is a constant value set in the algorithm, the overall complexity of the algorithm is not increased.

5 Evaluation of LP Models Used in Network Resource Provisioning Module

To compare characteristics of our LP Models used for network resource provisioning, we have run calculations for an example 24-node Polish network shown in Fig. 1. This network was set up in IIP project. In general, for 24-node Polish network we distinguished two types of nodes: edge (labelled as E) and core (labelled as C) nodes.

Fig. 1 Example topology of a 24-node Polish network used in computations

The data sent by the edge nodes can be any of all three types of the considered input/output traffic. Fig. 1 additionally shows the total capacity of network links as well as the respective link propagation delay for links between core nodes. The properties of links between core and edge nodes (i.e., 1 Gbps bandwidth and 2 ms transmission delay each) are not shown due to limited space. We assumed link bandwidth as a continuous (i.e., non-modular) value. Details of demand parameters, for all traffic flows for each PIs used in calculations are shown in Table 1. We decided to present them in order to simplify the description. For flexibility of Model 3 validation, we assumed that IPv6_QoS Parallel Internet traffic does not have any specific requirements referring to the maximum transmission delay. This assumption was realised in computations by setting a high value of requested delay (250 ms in the considered case) for IPv6_QoS traffic - such big value makes this delay insignificant in the optimization process for this type of traffic (in contrast to other PIs).

Computation results for all considered models are shown in Table 2. Four parameters were analysed, namely: bandwidth utilization of links, nodes resources, hop count, and transmission delay. The results were obtained for each of three assumed types of Parallel Internet, and were supplemented by the respective total values for the entire network.

As we can expect, each model is the best one in terms of value of its objective function and constraints. For this reason, Model 1 to minimize the bandwidth utilization, gave the lowest bandwidth utilization values, as expected.

Model 2 imposes additional constraints for node resources, such as mass storage, RAM, or processing power. These constraints can thus influence the solution, e.g., by rejecting some shortest paths including nodes without enough resources. Since the above node constraints are independent from link bandwidth, this makes up some freedom in searching for feasible solutions.

In Model 3, an additional upper bound on end-to-end transmission delay for each flow (calculated as the sum of delay values on all links constituting the transmission path) was introduced. Nevertheless, the upper bound imposed on transmission delay value for certain demands may cause selection of paths for delay-sensitive demands, which are not the shortest ones. The above constraint impacts the average total result of the average transmission delay for Model 3, which occurred to be higher than the respective one for Model 1.

In order to further evaluate all three models in different settings, we have carried additional optimization process for a 14-node NSF network. The topology was taken from refer-

Table 1 Details of demands used in computations for a 24-node Polish IIP network

Demand (from → to)	Internet type	Bandwidth [Mbps]	Max. delay [ms]
7 → 16	IPv6_QoS	200	250
16 → 7	IPv6_QoS	200	250
9 → 14	IPv6_QoS	200	250
14 → 9	IPv6_QoS	200	250
4 → 11	IPv6_QoS	200	250
11 → 4	IPv6_QoS	200	250
1 → 5	IPv6_QoS	200	250
5 → 1	IPv6_QoS	200	250
10 → 3	IPv6_QoS	200	250
3 → 10	IPv6_QoS	150	250
6 → 14	IPv6_QoS	150	250
14 → 6	IPv6_QoS	150	250
8 → 14	CAN	250	12
11 → 7	CAN	250	7
10 → 13	CAN	250	9
12 → 8	CAN	250	12
7 → 11	CAN	175	7
14 → 8	CAN	175	12
8 → 12	CAN	175	12
13 → 10	CAN	175	9
9 → 2	CAN	115	7
2 → 9	CAN	115	7
4 → 14	CAN	115	12
14 → 4	CAN	115	12
6 → 13	DSS	250	8
2 → 14	DSS	250	12
9 → 5	DSS	250	12
1 → 12	DSS	250	12
14 → 2	DSS	150	12
13 → 6	DSS	150	12
12 → 1	DSS	150	12
5 → 9	DSS	150	12
7 → 15	DSS	100	10
15 → 7	DSS	100	10
6 → 12	DSS	100	8
12 → 6	DSS	100	8

ence [20] and is presented in Fig. 2. In comparison with previously used IIP topology, for NSF topology we did not differentiate nodes into edge and core nodes (each node could thus be either a source, destination, or a transit one for demands). The properties of links between nodes (i.e., 2 Gbps bandwidth per each link) are not shown due to limited space. Results for the NSF network for three introduced models are presented in Table 4, preceded by details of demands shown in Table 3.

Table 2 Results of resource provisioning module for the example network from Fig. 1

		Model 1	Model 2	Model 3
Bandwidth [Mbps]	IPv6_QoS	18600	18600	18670
	CAN	16890	16890	16890
	DSS	17600	17600	18670
	Total	53090	53090	53360
Nodes resources [Mflops]	IPv6_QoS	64700	64700	64910
	CAN	59345	59575	59575
	DSS	60900	60900	62000
	Total	184945	185175	186485
Hop count	IPv6_QoS	6.25	6.5	6.625
	CAN	6.0	6.0	6.0
	DSS	6.5	6.625	6.625
	Total	6.25	6.375	6.417
Transmission delay [ms]	IPv6_QoS	10.5	10.875	11.25
	CAN	10.75	10.75	10.75
	DSS	11.0	11.375	11.5
	Total	10.75	11.0	11.17

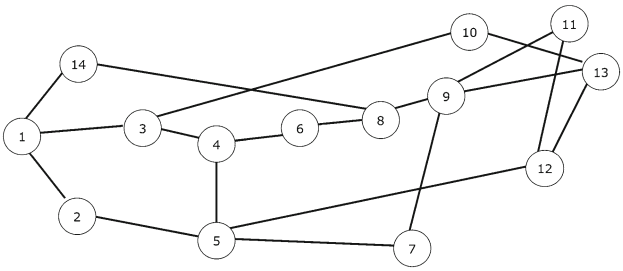

Fig. 2 NSF backbone network topology used in the second experiment

6 Implementation and Integration Aspects of the Resource Provisioning Module

Resource provisioning module is an integral part of the management system proposed in the IIP project. In this section, the description of processes implementing the network resource provisioning procedure, as well as aspects of integration with the management system are presented. The approaches discussed in Section 2 were implemented using GNU Linear Programming Kit (GLPK) [11] libraries, which were integrated by means of a program implemented using *C* programming language. For testing and evaluation of our algorithms, standard GLPK packages from three Linux distributions were used: Debian 6.0, Ubuntu 11.04 and Ubuntu 12.04. Our tests proved the algorithms validity as well as usability of GLPK library for solving Linear Programming

Table 3 Details of demands used in computations in the second experiment (for NSF network)

Demand (from → to)	Internet type	Bandwidth [Mbps]	Max. delay [ms]
2 → 13	IPv6_QoS	200	250
13 → 2	IPv6_QoS	200	250
12 → 14	IPv6_QoS	200	250
14 → 12	IPv6_QoS	200	250
8 → 5	IPv6_QoS	200	250
5 → 8	IPv6_QoS	200	250
3 → 11	IPv6_QoS	200	250
11 → 3	IPv6_QoS	200	250
14 → 12	IPv6_QoS	200	250
12 → 14	IPv6_QoS	150	250
6 → 7	IPv6_QoS	150	250
7 → 6	IPv6_QoS	150	250
2 → 13	CAN	250	12
13 → 2	CAN	250	7
12 → 14	CAN	250	9
14 → 12	CAN	250	12
8 → 5	CAN	175	7
5 → 8	CAN	175	12
3 → 11	CAN	175	12
11 → 3	CAN	175	9
14 → 12	CAN	115	7
12 → 14	CAN	115	7
6 → 7	CAN	115	12
7 → 6	CAN	115	12
2 → 13	DSS	250	8
13 → 2	DSS	250	12
12 → 14	DSS	250	12
14 → 12	DSS	250	12
8 → 5	DSS	150	12
5 → 8	DSS	150	12
3 → 11	DSS	150	12
11 → 9	DSS	150	12
14 → 12	DSS	100	10
12 → 14	DSS	100	10
6 → 7	DSS	100	8
7 → 6	DSS	100	8

problems (equations) on all Linux distributions. The program can be easily extended for different algorithms using not only Linear Programming tools. Also, integration of provisioning algorithms with the management system can be easily implemented, which makes it possible to be run as a standalone process that can be launched on demand at any time.

The high level procedure implementing the network resource provisioning and its interfaces to the management system is shown in Fig. 3. The procedure is activated on-demand by the management system. The reasons to run the provisioning process can be twofold: (1) admittance (realization) of new customers demands using free resources, or (2) significant changes referring to the network resources or topology, such as link or node failure or network infrastructure upgrade. The procedure consists of the following sequence of tasks:

Table 4 Results of resource provisioning for NSF network from Fig. 2

		Model 1	Model 2	Model 3
Bandwidth [Mbps]	IPv6_QoS	15800	15800	15800
	CAN	15120	15120	15120
	DSS	14000	14000	14000
	Total	44920	44920	44920
Nodes resources [Mflops]	IPv6_QoS	53450	54250	53450
	CAN	51460	51110	51110
	DSS	47600	47900	47600
	Total	152510	185175	186485
Hop count	IPv6_QoS	5.37	5.37	5.25
	CAN	5.25	5.25	5.25
	DSS	5.25	5.37	5.25
	Total	5.29	5.33	5.25
Transmission delay [ms]	IPv6_QoS	8.87	8.75	8.5
	CAN	8.75	8.87	8.87
	DSS	8.5	8.87	8.5
	Total	8.71	8.83	8.62

1. At the beginning of provisioning procedure, the Management System triggers two processes (see Fig. 3) using the *Request_For_Building_PI()* signal, i.e., Generate Demands Description process and Generate Network Topology Description process. The triggered processes are responsible for preparing the data in the unified format for the Resource Allocation process.
2. The Generate Demands Description process is responsible for translating PIs demands provided by the Management System in an XML format to the format used by the Resources Allocation process. The PIs demands are included in the PIs Resources Request data file. The PIs Resources Request data file also points at the resource allocation algorithm to be used by Resources Allocation process.
3. In parallel to the Generate Demands Description process, the Generate Network Topology Description process is run. The main role of this process is to retrieve from Management Database the information about: links prop-

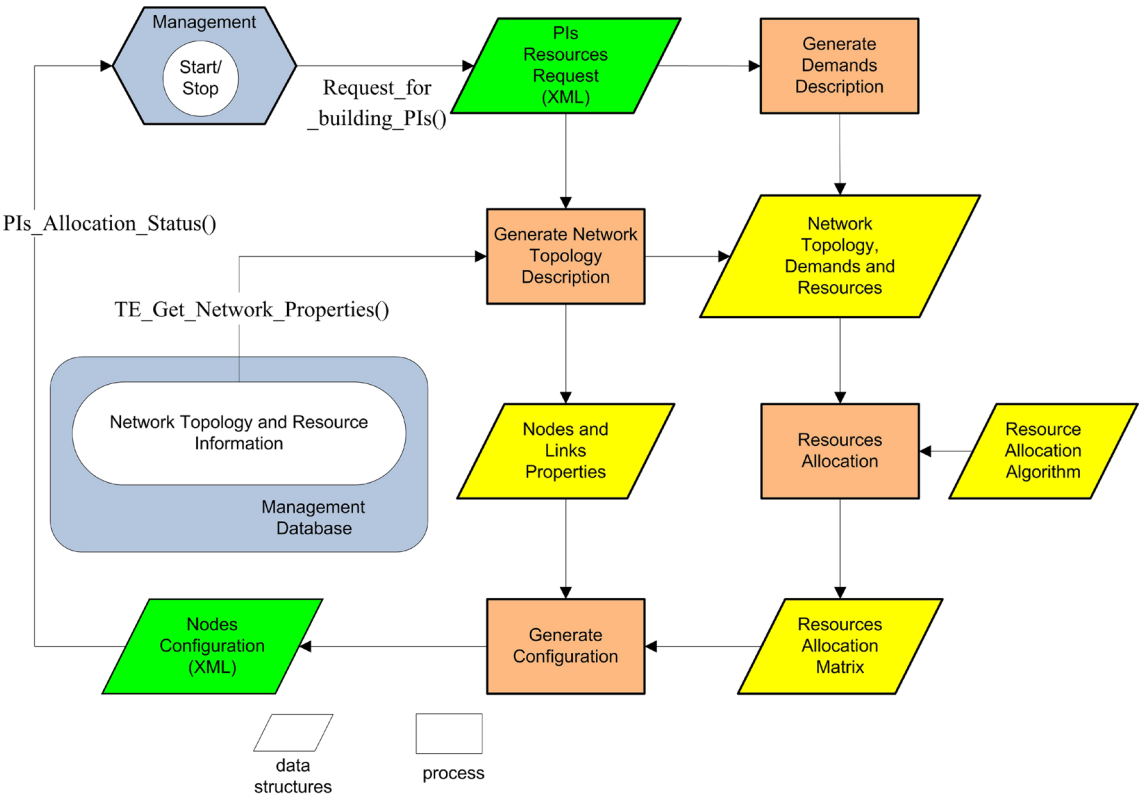

Fig. 3 Functional diagram of network resource provisioning module used in System IIP project

erties, node properties, and interconnections between nodes. Based on the above information, the network topology with resources to be allocated is generated in format required by the Resources Allocation process. This process also generates Nodes and Link Properties data required by the Generate Configuration process.

4. Next, the Resource Allocation process based on the data prepared by Generate Demands Description and Generate Network Topology Description processes maps PI resources requirements to the available network resources. As an output from this process, Resources Allocation Matrix data records are created with resources allocated to each PIs on the selected links. The result of the Resources Allocation process is strongly dependent on the availability of network resources. If appropriate resources (compliant with PIs Resources Request data) cannot be allocated, then the management system is sent a negative response.
5. If Resources Allocation Matrix data records are computed, then the Generate Configuration process generates the Nodes Configuration file. In the file, a configuration for all network nodes (taken into account by the Resource Allocation process) is described and each node configuration is hardware-dependent. XML is used as a format of the configuration file, which is the output of the resource provisioning subsystem. If Resources Allocation Matrix data records are not calculated, then the management system receives a negative response via the *PIs_Allocation_Status()* message.

The database of Resources Allocation Algorithms contains the set of optimisation algorithms definitions. The algorithms can be selected alternatively according to the assumed optimization goal and additional requirements. The management system initiates the provisioning process running it with an .xml input file describing PIs demands and the type of an algorithm to be used by the resource allocation process.

7 Conclusions

In this paper, we characterized our concept of a network resource provisioning module designed for System IIP – the architecture of Polish Future Internet, supporting parallelization of Internets. Three optimization models of network resource provisioning with different goal functions and constraints were proposed. In the latter part, we outlined the implementation aspects, as well as discussed the issues of our module integration with the respective management system.

The main idea was to design a relatively simple procedure of network resource provisioning that would allow for easy management of available elementary resources. Initial

requirements defined for the analyzed Parallel Internets were transformed into input data matrix, while the described provisioning process resulted in producing the Resources Allocation matrix needed to configure the virtual devices.

Acknowledgments This work has been supported in part by Ministry of Science and Higher Education, Poland, under the European Regional Development Fund, Grant POIG.01.01.02-00-045/09-00 Future Internet Engineering. The authors would like to address special thanks to Halina Tarasiuk, Wojciech Burakowski, Piotr Cholda, and Janusz Granat for their valuable comments.

References

1. Ahuja, R. K., Magnanti, T. L., & Orlin, T. L. (1993). *Network Flows: Theory, Algorithms, and Applications*. : Prentice Hall.
2. Andersen, R., Chung, F., Sen, A. & Xue, G., (2004). On disjoint path pairs with wavelength continuity constraint in WDM networks, in *Proc. IEEE INFOCOM*, pp. 524–535.
3. Botero, J., Hesselbach, X., Fischer, A., & Meer, H. (2012). Optimal mapping of virtual networks with hidden hops. *Telecommunication Systems*, *51*(4), 273–282.
4. Burakowski, W., et al., (2011). IIP System Specification level 1 and 2, POIG IIP project deliverable.
5. Burakowski, W., et al., (2011). The Future Internet engineering project in Poland: goals and achievements, in *Proc. Future Internet Poland Conference*, Poznan, Poland.
6. Cetinkaya, E., Broyles, D., Dandekar, A., Srinivasan, S., & Sterbenz, J. P. G. (2013). Modelling communication network challenges for Future Internet resilience, survivability, and disruption tolerance: a simulation-based approach. *Telecommunication Systems*, *52*(2), 751–766.
7. Cholda, P., Gozdecki, J., Kantor, M., Wielgosz, M., Pach, A.R., Wajda, K. & Rak, J., (2011). Provisioning concepts or the IIP Initiative, in *Proc. ICTON'11* 1–4.
8. Chowdhury, N., Rahman, M. & Boutaba, R. (2009). Virtual network embedding with coordinated node and link mapping, in *Proc. IEEE INFOCOM'09* 783–791.
9. Dedecker, P., Hoebeke, J., Moerman, I., Moreau, J., & Demeester, P. (2013). Network virtualization as an integrated solution for emergency communication. *Telecommunication Systems*, *52*(4), 1859–1876.
10. Fischer, A., Botero, J. F., Duelli, M., Schlosser, D., Hesselbach, X., & de Meer, H. (2011). ALEVIN - A framework to develop, compare and analyze virtual network embedding algorithms. *Electronic Communications of the EEAST*, *37*, 1–12.
11. Garcia, A. E., Rodriguez, L., & Hackbarth, K. D. (2012). Cost models for QoS-differentiated interconnecting and wholesale access services in future generation networks. *Telecommunication Systems*, *51*(4), 221–231.
12. http://www.4ward-project.eu/
13. http://akari-project.nict.go.jp/eng/index2.htm
14. http://www.geni.net/
15. http://www.iip.net.pl/
16. Karp, R. M. (1972). Reducibility among combinatorial problems, In *Complexity of Computer Computations* 85–103.
17. Kumar, A., Rastogi, R., Silberschatz, A., & Yener, B. (2002). Algorithm for provisioning Virtual Private Networks in the Hose model. *IEEE/ACM Transactions on Networking*, *10*(4), 565–578.
18. Liu, X., Chan, Y., & Xu, W. (2006). Stochastic programming methods used for network optimization. *Performance Eval.*, *63*, 1005–1015.
19. Meer, H., Hummel, K., & Basmadjian, R. (2012). Future Internet services and architectures: trends and visions. *Telecommunication Systems*, *51*(4), 219–220.
20. Qin, Y., Kia, J., & Mason, L. G. (2003). Study on a joint multiple layer restoration scheme for IP over WDM networks. *IEEE Network*, *17*(2), 43–48.
21. Ricciardi, S., Careglio, D., Santos-Boada, G., Sol-Pareta, J., Fiore, U., & Palmieri, F. (2013). Towards an energy-aware Internet: modeling a cross-layer optimization approach. *Telecommunication Systems*, *52*(2), 1247–1268.
22. Wittevrongel, S., Fiems, D., & Walraevens, J. (2008). Modelling and performance evaluation of future generation Internet networks. *Telecommunication Systems*, *39*(2), 61–62.
23. Yu, M., Yi, Y., Rexford, J., & Chiang, M. (2008). Rethinking virtual network embedding: substrate support for path splitting and migration. *ACM SIGCOMM CCR*, *38*(2), 17–29.
24. Zhu, Y., & Ammar, M. (2006). Algorithms for assigning substrate network resources to virtual network components, In Proc. *IEEE INFOCOM'*, *06*, 1–12.

Janusz Gozdecki received his M.Sc. and Ph.D. degrees in Telecommunications from AGH University of Science and Technology (AGH-UST), Krakow, Poland in 1995 and 2010, respectively. Since 1995 he has been working at AGH-UST, Department of Telecommunications (first as a researcher, and later as an assistant professor). His research is focused on wireless networks, as well as quality of service in IP networks. He has frequently served as a reviewer for international journals/conferences. He has actively participated in several European projects, including BTI, MOBYDICK, DAIDALOS I, DAIDALOS II, CARMEN, MEDUSA, HECTOR, FLAVIA, PROACTIVE, as well as grants supported by the Ministry of Science and Higher Education, Poland. He is also the co-author of nearly 70 papers and six books. He is a member of IEEE.

Mirosław Kantor holds M.Sc. and Ph.D. degrees in Telecommunications from AGH University of Science and Technology (AGH-UST), Krakow, Poland in 2001 and 2010, respectively. In 2001 he joined the Department of Telecommunications of AGH-UST as a researcher, where he is employed until now (currently as an assistant professor). His research interests are in the areas of Internet routing, inter-domain traffic optimization, as well as techno-economic analysis of telecommunication networks. He has been a reviewer/TPC member for international journals/conferences. He has actively participated in several European projects, including LION, NOBEL, BONE, SmoothIT, and Euro-NF, as well as grants supported by the Ministry of Science and Higher Education, Poland. He is also the co-author of nearly 40 papers and two books.

Krzysztof Wajda received his M.Sc. in Telecommunications in 1982 and Ph.D. in 1990, from AGH University of Science and Technology (AGH-UST), Krakow, Poland. In 1982 he joined AGH-UST, where he is employed until now. At AGH-UST, he was first responsible for laboratory of switching technology. Next he spent a year at Kyoto University and half a year in CNET (France). Krzysztof Wajda is currently an assistant professor at AGH University of Science and Technology. He actively participated in a few international projects, including: Leonardo da Vinci (JOINT and ET-NET), COST 242, ACTS 038 BBL, Copernicus ISMAN, IST LION, IP NOBEL, NoE e-photon/ONe(+), BONE, SmoothIT, and SmartenIT. He was recently involved in Future Internet Engineering project – Polish initiative towards NG Internet. He also participated in several grants supported by National Science Foundation, Poland (he was a project leader for two of them). He has also served as a reviewer of journals, including e.g., IEEE Communications Magazine, Telecommunications Systems, and international conferences. He has been a consultant to major Polish telecommunication companies. The main research interests include: traffic management, performance evaluation, network reliability, control plane, management systems, network services. Dr. Wajda is the author/co-author of 6 books, and nearly 110 technical papers. He is a member of IEEE.

Jacek Rak received his M.Sc. and Ph.D. degrees in computer science (options: computer networks and computer communications, respectively) with distinction from Gdansk University of Technology (GUT), Poland in 2003 and 2009, accordingly. He is currently an Assistant Professor at the Department of Computer Communications at GUT. His main research areas include: routing, design, and analysis of communication networks with special focus on network resilience. He is the author/co-author of over 60 publications, including around 20 publications in journals Dr. Rak has been involved in numerous projects related to optimization of reliable computer networks. He has also served as a TPC member of numerous international conferences on communications, e.g., IEEE ICC, IEEE GLOBECOM, DRCN, and journals, including IEEE Trans. on Networking, IEEE Communications Letters, or IEEE Trans. on Multimedia. Recently, he has been the TPC Co-chair of ICUMT 2011&2012, NETWORKS 2010, Publication Chair of BCFIC 2011&2012, NETWORKS 2010&2012, Workshops Chair of ITST 2013, and Publicity Chair of DRCN 2013. Between 2012 and 2014, he served as a member of the Editorial Board of Telecommunication Systems (Springer). Dr. Rak is currently the Vice Chair of IFIP TC6 WG 6.10, a senior member of IEEE, Steering Committee Member of NETWORKS and ICUMT conferences, as well as the founder and the General Chair of International Workshop on Reliable Networks Design and Modeling (RNDM).