Cost-optimal design of VoIP networks using the VPN concept

Levente Tamási, Dániel Orincsay *, Balázs Gábor Józsa

Ericsson Research, Traffic Analysis and Network Performance Laboratory, P.O. Box 107, H-1300 Budapest, Hungary

Received 1 September 2004; received in revised form 18 May 2005; accepted 23 May 2005
Available online 21 July 2005

Abstract

This paper addresses the issue of cost-optimal voice over IP (VoIP) network design. In the applied model, the whole VoIP network is divided into two logical components: the access network and the transport network. The access network consists of VoIP end-points that connect to the transport network through edge routers serving as gateways. Since multiple edge routers may be available for any given VoIP node, one task of the design process is to assign a particular edge router to every VoIP node. The edge routers have to be connected in a way that security and availability can be assured for the VoIP traffic. One obvious approach to fulfilling these requirements, which is assumed throughout the paper, is to define a virtual private network (VPN). Supposing a large volume of VoIP traffic, the cost of the VPN can be significant; thus, the other task of VoIP network design is to specify the transport VPN in the most economical way. These two tasks of VoIP network design can be solved separately using existing methods; nevertheless, the specification of VoIP regions influences the cost of the final solution to a great extent. Therefore, in this paper a novel approach is proposed in which the edge router assignment process takes the objective function of VPN specification into consideration as well. In order to realize the new approach, multiple methods are introduced which are based on the paradigms of genetic algorithms and simulated annealing. These methods perform a sophisticated optimization of the gateway assignments using various cost calculation methods. To evaluate the new algorithms, a method based on a well-known greedy solution to the problem is used as reference. Moreover, a VPN specification algorithm is presented which utilizes the stepwise nature of the cost functions. The performance of the presented methods is evaluated with the help of simulations. It is shown that the proposed methods outperform the reference algorithm significantly in the simulation scenarios investigated.

© 2005 Elsevier B.V. All rights reserved.

Keywords: VoIP; VPN; Cost-optimal; Network design; Stepwise function
1. Introduction

Nowadays, the all-IP concept is favored by the infocommunication industry, which intends to conduct all types of traffic over the internet protocol dominant in the networking area. As a part of all-IP, an increasing number of companies in the telephony area commit themselves to using the voice over IP (VoIP) technology. Assuming a large VoIP network with a huge number of customers, it is necessary to take quality of service (QoS) as well as economic criteria into account during the design phase.

QoS requirements may be considered in multiple ways in the VoIP network. One possible approach is using a best effort IP service, the drawbacks of which are its incapability to provide QoS guarantees and the consequent fact that the service level can only be maintained by overprovisioning [1]. On the other hand, guaranteeing per flow QoS between all customer end-points, using e.g., the IntServ architecture, is both expensive and inefficient. In the current study, a practically viable intermediate approach is followed: the VoIP network is divided into two logical components, with a pure IP based service used in the access network and QoS guaranteed in the transport network. The access network includes the VoIP nodes, i.e., the customer end-points that intend to use the VoIP service, while the transport network serves the purpose of carrying the aggregated VoIP traffic between the various access areas. The main parts of the transport network are the edge routers, the transit routers, and the connections between them. Since a VoIP node may reach more than one edge router so that the transmission path fulfills the QoS requirements of the telephone service (e.g., limited maximal delay), it is necessary to select the one that will serve as a gateway towards the transport network. The VoIP nodes assigned to the same edge router form a so-called VoIP region.

Generally, it is more economical to apply a virtual private network (VPN) instead of deploying a brand-new physical network to realize the transport network for a VoIP service (see e.g., [2,3]). The most important advantages of VoIP VPNs are cost efficiency, security, and scalability [4,5]. Therefore, this study follows the VPN approach, which can be applied in different layers of the open system interconnection (OSI) model. One possibility is using a Layer 1 (L1) optical VPN (oVPN), in which case a network user rents separate optical connections from a service provider [6] and installs its own routers at their end-points. This results in total separation of the users’ traffic at the physical level, similarly to leased lines. Another novel approach is using a Layer 2 (L2) VPN [7], which provides a data link layer service to customers over a wide area network (WAN) using multiprotocol label switching (MPLS); therefore, hosts connected by a wide area network appear to be on the same local area network (LAN). MPLS can also be used to realize a traditional Layer 3 (L3) IP VPN. The latter two approaches provide logical separation of users’ traffic, i.e., at the physical level, the optical connections are shared, and the traffic of different customers is only separated at the MPLS level with the help of label switched paths (LSPs).

To sum up, there are several approaches to realizing a VoIP VPN. In all cases, however, the cost of the VPN depends on the capacity of the corresponding devices, i.e., the routers and the connections between them. The area of cost-optimal VPN design has been widely studied in the literature, with the topology, the set of traffic demands, and the cost functions of devices assumed as input parameters. However, during VoIP network design, the traffic distribution between the VPN nodes cannot be considered a fix input since it largely depends on the specification of VoIP regions. Therefore, two interdependent tasks can be differentiated between in the case of VoIP network design: (1) VoIP region specification, i.e., the assignment of each VoIP node to exactly one VPN edge router and (2) the design of the transport network covering the selected VPN nodes. These two tasks can be solved independently by applying a number of existing methods, in which case the objective of the VPN transport network design is disregarded during the VoIP region specification. However, it can be more efficient to have the cost and quality factors concerning the transport network taken into consideration already in the first task. This approach is followed in this paper by making several propositions with a view...
to solving the VoIP region specification task of the VoIP network design problem based on the principles of evolutionary algorithms [8] and simulated annealing [9]. Furthermore, in order to solve the VPN transport network design subproblem, a core network design algorithm (CND) is presented, which is applied after specifying the VoIP regions. This heuristic algorithm exploits the stepwise nature of the cost functions of transport network devices and has been shown to be efficient in the area of cost-efficient VPN specification [10]. The numerical investigation of the proposed methods is performed by means of simulations, using the final cost of the VoIP network given by CND as the main performance measure. During the investigations, a region specification method based on a well-known greedy algorithm [11] is used as reference.

The rest of the paper is organized as follows. The next section describes the network, traffic, and cost models used. It also includes the formulation of the VoIP network design problem. In Section 3 novel approaches are introduced aimed at solving the VoIP region specification subproblem. Section 4 describes the core network design algorithm proposed for transport VPN design. Section 5 presents numerical results obtained from the performed simulations. Finally, the conclusions are drawn.

2. Problem statement

This section introduces the interpretation of the VoIP network design problem considered in the paper. First, the applied network, traffic, and cost models used. It also includes the formulation of the VoIP network design problem. In Section 3 novel approaches are introduced aimed at solving the VoIP region specification subproblem. Section 4 describes the core network design algorithm proposed for transport VPN design. Section 5 presents numerical results obtained from the performed simulations. Finally, the conclusions are drawn.

2.1. Network model

The VoIP network is modeled by a graph in the following way. A customer end-point that uses the VoIP service is called a VoIP node. The set of VoIP nodes is denoted by $W$. The possible routers of the VPN transport network to be composed are called VPN nodes. The set of VPN nodes is denoted by $V$.

Further, so-called VoIP edges are given which connect the VoIP-VoIP and VoIP-VPN node-pairs. The set of VoIP edges is denoted by $F$. Each VoIP edge $f \in F$ has a delay attribute $\text{delay}_f$ assigned representing its maximal one-way latency. Based on these edge delay values the delay value $d_{nv}$ can be determined for each VoIP node $w \in W$ and VPN node $v \in V$ pair, representing a guarantee on the maximal latency between them. If a VPN node $v$ is not available for a VoIP node $w$, the value of $d_{nv}$ is considered to be $\infty$. In the current interpretation, the QoS requirement that the route between the VoIP node and the corresponding VPN gateway has to fulfill is the maximal access network latency $d_{\text{max}}$. Thus, a VPN node $v \in V$ is only considered as a candidate gateway for a given VoIP node $w \in W$ if the value of $d_{nv}$ does not exceed $d_{\text{max}}$. Since these delay requirements have to be fulfilled at both end-points of a VoIP call, and the VPN concept guarantees that latency limits are satisfied in the transport network, the maximum delay requirements of the voice service (i.e., a one-way delay of 150 ms as recommended by the ITU-T) can be assured.

VPN nodes are connected by so-called VPN edges, the set of which is denoted by $E$. The use of VPN nodes as well as VPN edges is optional; typically only a subset of them is included in the final solution of the design. The bandwidth of VPN nodes and edges is limited; their particular capacity values are determined during the design phase using the corresponding cost functions (see Section 2.3). While most VPN specification methods take the maximal capacity constraint of connections into account, the capacity aspect of routers is often neglected. However, during the design phase this latter factor also has to be considered, i.e., VPN nodes must be handled similarly to VPN edges. For this reason, the VPN nodes are substituted by virtual VPN edges for the investigations as shown in Fig. 1. For instance, node $u$ is decomposed into two nodes $u_{\text{in}}$ and $u_{\text{out}}$, which belong to the incoming and outgoing traffic of the original node, respectively. Henceforth, node $u_{\text{out}}$ is responsible for the traffic originated, while node $u_{\text{in}}$ manages the traffic terminated by the original VPN node $u$. Further, since the traffic transferred by the original node $u$ traverses the virtual edge...
between nodes $u_{in}$ and $u_{out}$, the capacity constraint of the original router can be easily taken into consideration.

### 2.2. Traffic model

Although a VoIP node may refer to one particular customer owning a VoIP phone, it typically represents a private branch exchange (PBX) including a VoIP media gateway, which serves a number of users conducting a significant amount of VoIP calls. Due to the large number of VoIP nodes the generally applied approach of the pipe model [12] (also known as the trunk model), i.e., the source–destination pair based handling of traffic, is cumbersome. Therefore, the hose model [13] is followed, which defines only the sum incoming and outgoing traffic of a node. Assuming that telephony calls are handled, the incoming and outgoing traffic are equal corresponding to the symmetric hose model. Thus, the traffic of any given VoIP node $w$ is modeled by a bandwidth demand value $t_{rw}$ that shows the amount of capacity needed to satisfy the calls generated (and received) by the VoIP users in the given node. This value can be derived from the number and calling habits of different VoIP users; however, this issue is related to the area of traffic modeling, and it is thus beyond the scope of the paper.

Although the hose model based design results in networks that can accommodate extreme traffic distributions as well, the high amount of spare capacity and the consequent extra price make it unacceptable in cost-sensitive situations. Therefore, in the case of the transport VPN the pipe model is applied, i.e., the traffic between a pair of VPN nodes $u$, $v \in V$ is modeled by a bandwidth value $T_{uv}$, which is estimated in the following way. First, the total hose traffic value of each edge VPN node is calculated by summing the hose traffic values of VoIP nodes in the corresponding VoIP region. Considering that these traffic values represent large numbers of users and the number of VPN nodes is relatively low, the pipe model is approximated by distributing the sum traffic of a particular VPN node among the other VPN nodes in direct proportion to their total hose traffic values.

### 2.3. Cost model

The relationship between the cost and capacity values of VPN nodes and edges can be represented with the help of monotonic nondecreasing cost functions. In the simplest case, linear functions can be used. In this way, every required capacity unit has the same cost value, which gives the slope of the curve. In the overwhelming majority of cases, these approximations are inadequate to model real cost relations; however, it is a relatively frequent approach because of the simplicity of the computations involved.

In real-life situations, the various devices, i.e., the routers and the connections between them have discrete capacity values and consequently discrete cost amounts. These types of cost dependencies can be described with so-called stepwise functions [14–17] (see Fig. 2). Although this approach makes the VPN design subproblem mathematically complex [18], it can fulfill the high accuracy requirements arising in real-life situations. Moreover, in the applied model each VPN node $v$ and VPN edge $e$ has an individual cost function $c_v$ and $c_e$, respectively, which enables special cost modifying factors as well as policy reasons to be considered.
2.4. Problem formulation

This section contains the description of the inputs and outputs of the two tasks of the VoIP network design problem. The VoIP region specification subproblem has the following input parameters: (1) the set of VoIP nodes \( W \), (2) the set of possible VPN nodes \( V \), (3) the set of delay values \( d_{wv} \) for all VoIP node \( w \in W \) and VPN node \( v \in V \) pairs (derived from the structure of VoIP nodes and edges) and the maximal access network latency \( d_{\text{max}} \), and (4) the hose traffic values \( t_{rw} \) for all VoIP nodes \( w \in W \).

As the output of region specification, each VoIP node is assigned to exactly one VPN node, which determines a set of VPN nodes that are mandatory elements of the VPN transport network to be formed. This assignment can be used to calculate the traffic demands between pairs of VPN nodes as described in Section 2.2. Therefore, the transport VPN design subproblem takes the following input parameters: (1) the set of VPN nodes \( V \) (both the gateways and the possible transit VPN nodes), (2) the set of possible VPN edges \( E \), (3) the bandwidth values \( T_{ruv} \) for all pairs of VPN nodes \( u, v \in V \), and (4) the cost functions of VPN nodes and edges denoted by \( \text{cost}_v \) and \( \text{cost}_e \), respectively. As a result of the design process, the VPN transport network is specified including the exact capacity values of all devices. The paths to be established for the aggregated VoIP traffic demands are also given by the applied design method.

Aiming at minimizing the overall establishment cost of the VoIP network, which basically depends on the cost of the VPN transport network, the objective of the whole design process is to

\[
\min \left\{ \sum_{e \in E} \text{cost}_e(\text{load}_e) + \sum_{v \in V} \text{cost}_v(\text{load}_v) \right\},
\]

where load refers to the actual capacity need on a certain device, and cost(load) indicates the corresponding price of the device.

3. VoIP region specification

This section proposes several methods that are capable of solving the VoIP region specification subproblem. Since it is related to the so-called set-covering problem widely studied in the literature [19,20], the term covering is used throughout the descriptions of various algorithms referring to the situation when a VPN node is selected as a gateway for a VoIP node.

3.1. Greedy covering algorithm (GC)

This section presents greedy algorithms that are based on a well-known greedy solution to the set-covering problem (see e.g., [11]). The algorithms consist of two main steps. In the first step, VoIP nodes with only one candidate gateway each are identified. Since each of these VoIP nodes has access to only one VPN node fulfilling the delay
requirements, the corresponding VPN nodes define a set of gateways that is a mandatory part of the solution. Consequently, all VoIP nodes that have at least one candidate gateway among these VPN nodes are considered covered. The second step of the algorithm is an iteration in which further VPN nodes are selected one by one based on a certain utility value. When a particular VPN node is selected, the uncovered VoIP nodes that can access the VPN node fulfilling the delay limits get covered. The iteration stops if the selected VPN nodes cover all VoIP nodes. In the following, two utility value variants are presented.

3.1.1. VPN node number based GC (GC-N)
In this variant the utility value corresponds to the number of yet uncovered VoIP nodes that can be covered by the given VPN node. Therefore, this approach aims at minimizing the number of selected VPN nodes; however, it does not take any cost factors into consideration.

3.1.2. VPN node cost based GC (GC-C)
The utility value used by this variant is the number of yet uncovered VoIP nodes that the given VPN node can cover divided by the cost value corresponding to the first step of its cost function. The idea behind this utility value is to take the establishment costs of the VPN nodes also into account during the iteration.

3.2. Evolutionary covering algorithm (EC)
A common property of the greedy covering algorithms presented above is that they assign a VPN node to each VoIP node only once, which means that they do not vary the existing assignments. This approach is referred to as construction method, i.e., the algorithm stops when the first feasible solution is found. This provides fast region specification; however, its drawback is that there is no possibility for sophisticated optimization.

This section proposes an algorithm that is based on the well-known paradigm of evolutionary algorithms (also called genetic algorithms) [8], which enables selection between more feasible solutions using complex cost calculation methods (see Section 3.4). The representation of the VoIP region specification subproblem applied in the evolutionary algorithm is the following. An entity defines a valid assignment, where each gene corresponds to a VoIP node, and its value refers to one of its candidate VPN nodes the particular VoIP node is assigned to. In each iteration, either a crossover or a killing operation is performed based on the actual population size. In the case of crossover, each of the two parents is selected by choosing the entity with the lowest cost (regarding the actual cost calculation method) from a set of randomly selected entities, and the child inherits each of its genes from either of its parents with equal probability. Similarly, the selection of entities that do not survive is performed by killing the oldest entity from a number of randomly selected entities, and killing the one with the highest cost (based on the cost calculation method used) from another set of randomly selected entities. In the case of both operations, the size of the set of randomly selected entities is given by a parameter \( k \). Moreover, during all iterations, mutation is performed, i.e., one gene of an entity is changed randomly, which means that the given VoIP node is re-assigned to another of its candidate VPN nodes. In order to create an appropriate initial population, the GC-C algorithm is applied in the following way. Several feasible solutions are sought by GC-C, excluding a different VPN node from the initial set of VPN nodes in each iteration. Naturally, VPN nodes that are mandatory elements of the solution cannot be excluded. Using this method, several different feasible coverings consisting of VPN nodes with low establishment cost values can be generated to form the initial population. The evolutionary algorithm stops if the cheapest solution considering the actual cost calculation has not changed during the last \( n \) steps.

3.3. Simulated annealing based covering algorithm (SC)
This section proposes an algorithm for the VoIP region specification subproblem based on the principle of simulated annealing [9]. The representation used is similar to the one presented in the previous section: a state defines a valid assignment between VoIP nodes and VPN nodes and consists
of values each of which refers to the VPN node the particular VoIP node is assigned to. The initial state is based on a solution by GC-C, while random changes in the state are generated by re-assigning a randomly selected VoIP node to another of its candidate VPN nodes. States are evaluated using the same cost calculation methods as in the case of the evolutionary algorithm (see Section 3.4). A state is accepted with a probability of $e^{-\frac{C}{C_0}}$, where $C$ and $C_0$ denote the cost of the previous and current state, respectively, while $T$ is the temperature of the previous state. This means that a state with a higher cost value can also be accepted by the algorithm; however, the probability of this event decreases heavily against both the cost difference and the temperature. The annealing schedule takes the form of $T' = T \cdot \phi$, where $T$ is the temperature in the previous iteration, $T'$ is the current temperature, and $\phi$ denotes the annealing factor. The initial temperature is denoted by $T_0$. As seen in the case of the evolutionary algorithm, the simulated annealing algorithm terminates if the cost of the cheapest solution has not improved during the last $n$ iterations.

3.4. Entity and state cost calculation methods

The quality of the solutions by both EC and SC is heavily influenced by the cost calculation method used for evaluating the entities and states, respectively. Thus, more approaches are investigated in this study, as it can be seen in the following sections.

3.4.1. Cost approximation based methods (EC-C, SC-C)

The main idea behind the cost approximation based methods (EC-C and SC-C) is trying to foresee the final cost of the VPN transport network to be designed. The set of the aggregated VoIP traffic demands are routed several times, based on different random orders. Dijkstra’s shortest path algorithm is applied for this purpose with an edge weight function that favors devices with low cost per unit traffic values. Finally, the algorithm considers the price of the cheapest configuration the cost of the entity or state.

3.4.2. Distance weighted traffic based methods (EC-D, SC-D)

When applying the distance weighted traffic based methods (EC-D or SC-D), the product of the bandwidth requirement and the length of the possible shortest path (in terms of hop-count) is calculated for each aggregated VoIP traffic demand. Then the cost of the entity or state is specified as the sum of these products regarding the whole network. This metric aims at reducing the number of transit VPN nodes required, especially between edge VPN nodes with higher hose traffic values.

3.4.3. Two-level cost metric based variants (EC-C2, EC-D2, SC-C2, SC-D2)

The two-level variants of the above cost metrics were also investigated in the following way. The number of the selected VPN edge routers serves as primary metric, while the value computed by the cost approximation or distance weighted traffic methods is normalized by the initial value of the metric, scaled by an importance factor $s$, and used as a secondary metric. These metrics generally favor assignments where the number of gateways is low, however, they may also select a larger number of edge VPN nodes if the gain in the secondary metric is significant enough.

3.4.4. Interconnection based methods (EC-I, SC-I, EC-TI, SC-TI)

The interconnection based methods (EC-I and SC-I) aim at reducing the total network cost by selecting a group of heavily interconnected edge VPN nodes, thus diminishing the number of the required transit VPN nodes. The degree of interconnection is defined as the number of VPN edges connecting the selected VPN nodes directly, divided by the maximal possible number of edges between them, which corresponds to the case when the selected VPN nodes are fully meshed. The cost of an entity or state is then defined as the number of selected VPN nodes, divided by the degree of interconnection.

In the case of the traffic weighted interconnection based methods (EC-TI and SC-TI), each VPN edge between the currently selected VPN nodes is assigned a weight defined as the sum hose traffic of...
the two VPN nodes it connects. The weights of all VPN edges between the selected VPN nodes are summed, and this sum is divided by its maximal possible value corresponding to a fully interconnected group of VPN nodes. The cost of the entity or state is defined as the number of selected VPN nodes divided by the above ratio. The idea behind this metric is to favor assignments where VPN nodes with higher hose traffic values can transmit their traffic on direct links, while taking the number of selected VPN nodes into consideration at the same time. Note that the above interconnection based methods (EC-I and SC-I) can be viewed as specializations of these methods where all VPN edges between the selected VPN nodes have unit weights.

4. Transport VPN design

This section describes the core network design (CND) algorithm proposed as a solution to the transport VPN specification subproblem defined in Section 2. The outline of the algorithm is given in Section 4.1, followed by a detailed description of its phases.

4.1. Algorithm outline

The algorithm presented for transport VPN design is a heuristic method which outperformed a reference algorithm based on the principle of greedy randomized adaptive search procedures (GRASP), a method which was shown to be able to solve the location and dimensioning problems arising in the field of cost-optimal VPN specification efficiently. Note that although CND is used to solve the transport VPN specification subproblem in this study, it may be applied in the case of any underlying networking technology that supports explicit routing and bandwidth reservation (e.g., ATM or MPLS).

The process can be divided into three subsequent phases (see Fig. 3). The algorithm starts with the initial capacity estimation (ICE) phase, whose task is to estimate the approximate capacity conditions. The main phase is the iterative routing optimization (IRO), which searches for an appropriate network configuration based on the initial estimations. Although this second phase already provides an economical solution to the VPN specification subproblem, a cheaper configuration can be reached with the help of the last phase called posterior capacity refinement (PCR).

Generally, two types of search methods can be differentiated between. The first two phases (ICE and IRO) perform global search, utilizing comprehensive information about the current network structure and the traffic demands. The purpose of this is to diminish the whole state space in a way that its remaining part approaches the optimal solution. Then the third phase (PCR) performs a local search aimed at finding the most favorable solution in the reduced state space.

4.2. Initial capacity estimation

The main purpose of the initial capacity estimation (ICE) phase is to assess the necessary device capacities by analyzing the set of traffic demands. Thus, prior information can be provided for IRO by supplying a partially dimensioned network as an initial state. Therefore, the remaining task of IRO is to finalize the dimensioning of devices and the routing of demands. Although ICE typically does not give a full solution to the task, it specifies a favorable starting configuration based on a global picture of the transport VPN design subproblem.

The operation of ICE is based on an iteration (see Fig. 4). In Step 1, the traffic demands to be accommodated are shuffled randomly. Then in Step 2, they are routed one after another (in the

**Phase 1:** initial capacity estimation (ICE)

**Phase 2:** iterative routing optimization (IRO)

**Phase 3:** posterior capacity refinement (PCR)

Fig. 3. The three phases of the core network design (CND) algorithm.
above order) using Dijkstra’s shortest path search algorithm with the following edge weight function:

\[ \text{newDeviceCost} \]

\[ \text{newDeviceTraffic} \], where \( \text{newDeviceCost} \) is the cost of the given device if it is involved in the path of the actual demand, and \( \text{newDeviceTraffic} \) is the sum traffic that the device has to manage in the above case. This weight function favors devices with low cost per unit traffic values with a view to keeping the overall network cost at a low level. In Step 3, the total network cost and device capacity values corresponding to the accommodation calculated in Step 2 are stored. These steps are repeated a predefined number of times (see Step 4), which is set to 100 during the experiments. Finally in Step 5, those accommodations are determined whose total network cost is worse than the best cost value encountered during the above iterations at most by a predefined percentage (set to 10% during the simulations), and the capacity values of the devices are specified in a way that each device is assigned the lowest value of the ones saved during the above accommodations. This idea is based on the assumption that if several random accommodations with low total network cost values need a certain amount of capacity on a particular device, then it is probable that the optimal accommodation will also need as much bandwidth.

4.3. Iterative routing optimization

The main phase of the algorithm, namely the iterative routing optimization (IRO) phase is based on an algorithm that can accommodate a given set of traffic demands in a graph with capacity limits. In this study, the algorithm proposed in [23] is used for this purpose. As it was shown in [23], this algorithm performs well in terms of feasibility, i.e., it finds a solution for a given problem instance, supposing that there is a valid solution. An important advantage of IRO is that the routing optimization procedure applied can be substituted by any other method fulfilling the same task.

The steps of the IRO process are shown in Fig. 5. In Step 1, IRO attempts to accommodate the traffic demand set considering the actual capacity constraints. If the algorithm terminates with success (Step 2), i.e., all demands are accommodated, IRO finishes. Otherwise, the capacity of a particular device is increased by one capacity step in the following way. In Step 3, the remaining demands that could not be fit into the graph under the actual capacity conditions are accommodated, disregarding the capacity constraints. The device to be extended in Step 4 is the one on which the capacity violation, i.e., the extra capacity required

<table>
<thead>
<tr>
<th>Step 1:</th>
<th>Attempt to accommodate traffic demands.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2:</td>
<td>If successful, stop.</td>
</tr>
<tr>
<td>Step 3:</td>
<td>Route remaining traffic demands disregardng the capacity constraints, and select device based on capacity violation.</td>
</tr>
<tr>
<td>Step 4:</td>
<td>Increase device capacity, and go to Step 1.</td>
</tr>
</tbody>
</table>

Fig. 5. Steps of the iterative routing optimization (IRO) phase.
by the actual accommodation is the largest. After
the capacity increase, Step 1 follows again with
the updated capacity limits.

4.4. Posterior capacity refinement

As it was discussed in Section 4.1, after the IRO
phase a suitable solution to the VPN specification
subproblem is already available. However, this re-
result might be improved with the help of posterior
capacity refinement (PCR). This greedy method
is based on a local search procedure, which means
that the process concentrates only on one part of
the network at a time.

The idea behind PCR is to reduce the size of de-
vices that are underutilized in the sense that if a
relatively small amount of traffic was removed
from them, a device with lower capacity and con-
sequently lower cost level would suffice. For exam-
ple, if the traffic to be managed on a connection is
170 Mbps, then two 155 Mbps devices have to be
installed in parallel, while if traffic on the particu-
lar connection could be reduced by about 10%,
one 155 Mbps device would be sufficient, whose
cost would be the half of the original. The remain-
ing 10% of the traffic should be redirected onto
other connections utilizing their spare capacity, if
possible.

Fig. 6 shows the operation of the PCR method.
In Step 1, the devices are sorted by their relative
step utilization, i.e., the utilization of the capacity
range belonging to the current cost value. The rea-
son is that in the case of a device with a lower rel-
ative step utilization, it is more probable that if its
capacity is decreased by one step, the traffic de-
mands can still be satisfied under these tighter
capacity conditions. Next in Step 2, the devices
are shrunk one by one in the above order, and
the accommodation of the traffic demands is at-
ttempted by IRO in Step 3. If, after the shrinking
of a particular device, IRO results in a configura-
tion that is cheaper than the best overall cost so
far (Step 4), the process restarts from the sorting
step. Otherwise, the solution before the capacity
decrease step is restored in Step 5. The process is
terminated if the overall cost could not be
improved since the last sorting (Step 6).

5. Results

In order to investigate the performance of the
proposed algorithms, simulations were carried
out using artificial problem instances. First, the
automated method of problem instance generation
is described; then, the performed simulation sce-
narios are presented including the analysis of the
numerical results.

5.1. Problem instance generation

During the simulations the aim was to create
problem instances that provide a good representa-
tion of real-life situations. The first task was to
generate the topology of the network, including
the VoIP nodes and edges as well as the possible
VPN nodes and edges. For this purpose a random
VoIP graph generator method was applied that is
based on the Barabási–Albert model [24,25]. This
approach is based on the power laws of Internet
topology [26,27], and nowadays it is frequently
used to model wide area communication networks.
Topologies of various sizes were examined; how-
ever, results are presented only for networks with

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Sort devices by relative step utilization, and select first device.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Shrink selected device.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Perform IRO.</td>
</tr>
<tr>
<td>Step 4</td>
<td>If the result is the best so far, save it, and go to Step 1.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Restore the solution before the capacity decrease step.</td>
</tr>
<tr>
<td>Step 6</td>
<td>If there are more devices, select next device, and go to Step 2, otherwise stop.</td>
</tr>
</tbody>
</table>

Fig. 6. Steps of the posterior capacity refinement (PCR) phase.
500 VoIP nodes and 50 possible VPN nodes. The delay values of VoIP edges were specified randomly using a distribution that generates values proportionally to the lengths of edges, i.e., the distances between their end-points. The maximum access network latency $d_{\text{max}}$ was set to 50 ms for the investigations, as was the guarantee on the maximal delay in the transport VPN. These values assure that the ITU-T recommendation for a maximal one-way delay of 150 ms is complied with.

The traffic demands of VoIP nodes were generated randomly in the following way. First, the maximal number of parallel calls was generated for each VoIP node in the interval $[1, 2^N]$. The value of $N$ refers to the average number of maximal parallel calls, and it was shifted from 32 to 256. Then the number of parallel calls had to be transformed into a bandwidth value based on the codec type and packetization overhead (RTP/UDP/IP header) actually used. Assuming the use of the ITU-T recommendation G.711 PCM codec, which can provide the highest voice quality, with silence suppression having an activity factor of 55% and a packetization time of 5 ms, this resulted in a call bandwidth of 64 kbps. Thus, the investigated average VoIP node traffic ($tr_w$) interval was 2–16 Mbps.

Note that the actual codec type only affects the bandwidth calculation, with silence suppression having an activity factor of 55% and a packetization time of 5 ms, this resulted in a call bandwidth of 64 kbps. Thus, the investigated average VoIP node traffic ($tr_w$) interval was 2–16 Mbps. Note that the actual codec type only affects the bandwidth calculation; moreover, since the average VoIP node traffic interval is relatively wide, the results presented can apply to various codec types.

The cost functions of VPN edges were based on the Synchronous Transfer Mode (STM) standards referring to a physical layer approach as described in Section 1, while in the case of VPN nodes two different device sizes were assumed (see Table 1). As the cost values of these functions only represent ratios, they can be considered cost units. In the case of VPN edges, the capacity values 311 Mbps and 1244 Mbps refer to the situation when two connections of the same size are deployed in parallel. The cost function of VPN edges is based on the assumption that the deployment of two devices of similar capacity is reasonable, while it is worth installing a device of a larger capacity instead of installing three parallel devices of a given size.

Since in real-life situations the cost functions of devices may deviate from the average, each cost function was distorted randomly during the simulations. The original costs were multiplied or divided (with equal probability) by $1 + r$, where $r$ is a random variable with a uniform distribution in the interval $[0, 1)$.

In order to get accurate results, numerous network topologies and traffic distributions were examined for each network size resulting in a confidence interval size of 2% at a significance level of 95%.

5.2. Optimization of parameters

The first simulation scenario targeted the optimization of the parameters of EC and SC. It turned out that in the case of EC the parameter value combination of $k = 3$ and $s = 500$, where $k$ and $s$ refer to the size of the sets of randomly selected entities and the importance factor of the secondary metric, respectively, can provide the most favorable results. In the case of SC, the combination of the initial temperature $T_0 = 10$, annealing factor $\phi = 0.9995$, and the importance factor $s = 500$ proved to be the best choice. Therefore, these parameter value combinations were used throughout the next simulation scenarios. The value of the parameter $n$ used in the stop condition of the methods was fine-tuned independently for all variants of both algorithms.

5.3. Total network cost

The most important performance indicator is the total cost of the resulting network, which is shown in Fig. 7 for the variants of the evolutionary algorithm EC. As it can be seen in the figure, algorithms EC-C and EC-C2 performed similarly. By

<table>
<thead>
<tr>
<th>Capacity</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Gbps</td>
<td>15</td>
</tr>
<tr>
<td>10 Gbps</td>
<td>45</td>
</tr>
<tr>
<td>155 Mbps</td>
<td>10</td>
</tr>
<tr>
<td>311 Mbps</td>
<td>20</td>
</tr>
<tr>
<td>622 Mbps</td>
<td>30</td>
</tr>
<tr>
<td>1244 Mbps</td>
<td>60</td>
</tr>
</tbody>
</table>
applying EC-I, about 2–3% improvement can be achieved. The use of EC-D results in more economical configurations in the higher traffic intervals. EC-TI provided the lowest total network cost in the case of lower average VoIP node traffic values, while EC-D2 proved to be the best algorithm as it outperformed all other variants in the traffic interval between 6 and 16 Mbps. The improvement may reach 15% compared to EC-C and EC-C2 at higher traffic volumes.

Fig. 8 shows the results of applying the different variants of the simulated annealing based algorithm SC. It can be seen that the results provided by SC-I, SC-C and SC-C2 converge in the higher traffic intervals. About 7–11% improvement can be gained by applying SC-D in the average VoIP node traffic interval between 12 and 16 Mbps. As in the case of EC, the methods based on the two-level distance weighted traffic and the traffic weighted interconnection metrics (SC-D2 and SC-TI) provided the best results. However, the gap between the results of the two methods was relevantly smaller than in the case of EC at every average VoIP node traffic value investigated.

Fig. 9 shows the total network cost for the two variants of the greedy covering algorithm GC as well as the two best variants of EC and SC. As it can be seen in the figure, GC-C provided 3–5% lower network costs than the simplest algorithm GC-N. The sophisticated optimization algorithms EC and SC proved that more efficient results can be achieved by selecting from multiple feasible solutions, as they outperformed the greedy covering algorithms in almost all cases. While the two variants of the evolutionary algorithm EC achieved better results than the methods based on simulated annealing at lower average VoIP node traffic values, above 12 Mbps the results of EC and SC converge. However, EC-D2 proved to be the best algorithm overall providing an improvement of 7–20% compared to the greedy algorithm GC-N over the traffic interval investigated.

5.4. Number of VPN nodes

Besides the cost of the VPN transport network, its size is also an important attribute, which can be
Fig. 8. Total network cost using SC.

Fig. 9. Total network cost using the different algorithms.
described well by the number of VPN nodes. Thus, in this scenario the algorithms were compared focusing on this basic measure. Fig. 10 presents the number of VPN nodes used in the final transport network differentiating between the edge and transit nodes for the two variants of the greedy covering algorithm GC as well as the best two variants of EC and SC. Since these measures did not show relevant change against the change of average VoIP node traffic, results are presented only for the 10 Mbps value.

An important observation is that in the case of both EC and SC the number of edge VPN nodes is higher for the two-level distance weighted traffic based variant than the traffic weighted interconnection based method, while the relation between the total network cost values of the two variants is exactly the opposite. This means that the sophisticated selection of VoIP traffic aggregation points is more important than keeping their number as low as possible. Another point to note is that in the case of the methods based on EC and SC, the ratio of transit VPN nodes is significantly smaller than in the case of the greedy algorithms. This can be explained by the fact that both the two-level distance weighted traffic and the traffic weighted interconnection metric aim at diminishing the number of transit VPN nodes, by reducing the length of the shortest path and increasing the degree of interconnection between the edge VPN nodes with higher hose traffic values, respectively.

5.5. Running time

Although in the case of off-line network design the running time has only secondary importance, it is worth examining this factor also in order to make the investigations complete. Fig. 11 depicts the values for the two versions of the greedy covering algorithm GC and the best two variants of EC and SC measured on a Sun Ultra Enterprise 420R machine with an Ultra II 450 MHz processor and 1 GByte RAM. As it can be seen, the GC algorithms were very fast as they provided results within 1 min on average. Another important point to note is that for both cost calculation methods shown the running time values of the simulated annealing based algorithm SC were higher than those of the evolutionary algorithm EC. This can be partly attributed to the fact that the running time of the VPN specification method presented in Section 4 depends heavily on the number of edge VPN nodes selected.

6. Conclusions

This paper addressed the topic of cost-optimal VoIP network design. The whole design problem consists of two main tasks: the specification of the VoIP regions and the design of the VPN transport network. A novel approach was proposed aimed at improving the cost efficiency by taking the objective of transport VPN design into consideration during region specification. Various algorithms were proposed that realize the approach...
based on the paradigms of evolutionary algorithms and simulated annealing, which perform a sophisticated optimization of the VoIP regions. Moreover, a VPN specification method utilizing the stepwise nature of cost functions was presented.

In order to evaluate the performance of the algorithms, numerous simulations were carried out. It turned out that significant reduction in total network cost can be achieved by applying sophisticated cost evaluation in the VoIP region specification phase. Based on the performed simulations, the evolutionary algorithm using the two-level distance weighted traffic metric seems to be the best choice.

Possible future work in the area includes the investigation of situations where other types of traffic demands with high bandwidth requirements, e.g., video telephony, are likewise handled. Another direction is to take reliability issues also into consideration during VPN transport network design.

Acknowledgments

This work was partially supported by project IKTA-0092/2002 of the Ministry of Education, Hungary. The authors would also like to thank Gábor Magyar and Gergely Tamási for their insightful suggestions and comments, and the people at Ericsson Research Hungary for their valuable remarks.

References

Communications Conference (GLOBECOM’2001), San Antonio, TX, USA, vol. 4, November 2001, pp. 2586–2590.


Levente Tamási received his M.Sc. degree from the Budapest University of Technology and Economics (Hungary), where he is currently a Ph.D. student at the High Speed Networks Laboratory of the Department of Telecommunications and Media Informatics. He also works as a research fellow at the Traffic Analysis and Network Performance Laboratory of Ericsson Research Hungary. His main interests are design and optimization problems in wide area networks.

Daniel Orincsay received his M.Sc. degree in 2000 from the Budapest University of Technology and Economics, where he is currently finalizing his Ph.D. Parallely, he also works as a research fellow at the Traffic Analysis and Network Performance Laboratory of Ericsson Research Hungary. His main interests are traffic engineering and performance optimization in high speed networks.

Balázs Gábor Józsa received his M.Sc. degree in computer science from the Budapest University of Technology and Economics (Hungary) in 2000. He is currently finishing his Ph.D. dissertation at the High Speed Networks Laboratory (HSNLab) of the same university. Meanwhile, he works as a research fellow at Ericsson Traffic Analysis and Network Performance Laboratory in Budapest, Hungary. His main interests are traffic engineering, routing and performance optimization problems in telecommunication networks.