Routing VoIP traffic in Large Networks

Martin Hruby, Michal Olsovsky, Margareta Kotocova

***Abstract*-Providing quality of service should be one of the main objectives when deploying sensitive applications into the network. Since network performance parameters are subject to frequent change, in this paper we propose a novel approach to routing sensitive VoIP traffic in large networks.**

Our approach takes measured delay and jitter into consideration and we establish an overlay of the original network to route primarily VoIP traffic. We achieve this by first modeling the probability distributions of network performance parameters and then by calculating the best paths by means of modified Dijkstra's and Kruskal's algorithms. Our approach also identifies weak network areas not suitable for VoIP deployment which can be subject to future network improvements. We provide preliminary functional and performance results.

***Index Terms*—network, proactive measurement, routing, QoS.**

I. Introduction

THIS paper is focused on sub-optimal traffic distribution in heterogeneous computer networks as this is the main issue we are facing when delivering bandwidth intensive services with QoS guarantees. Usually the decision on which paths to use to forward traffic is left on the interior routing protocol [1], [3] and the process of adding IP telephony functionality to an existing network environment is often cumbersome [2], [4], [17], [19].

Our objective is to optimize the flow of delay sensitive traffic by creating a VoIP backbone overlay in the network in a computationally efficient and effective manner. Two parameters which mostly impact the quality of VoIP traffic are delay and jitter (delay variance) [5], [6]. In our approach, we aim to identify network links which provide minimum delay and jitter by utilizing network probes as a method of active NPP measurements in production environments. The task of deploying active network measurement probes is non-trivial in large-scale networks [15], [16]. The number of logical links and load imposed on measurement endpoints (performing measurements and providing results) is a limiting factor when it comes to deciding which links to monitor. The value of measurement information however is considerable [14], [18]. While gathering results continuously, a multi-dimensional data structure is built wherein maxima can be found which represent optimal links for VoIP traffic and these are marked as VoIP backbone components. Further, these VoIP backbone components can be joined by well-known graph algorithms [11], [12], [13]. Having a VoIP backbone clearly identified in the topology, design considerations apply which can strongly improve the quality of service for VoIP traffic by placing crucial components of the IP telephony architecture (e.g. call managers, gatekeepers, Unity servers, PSTN gateways, border elements, etc.) on the VoIP backbone [7], [8].

Manuscript received April 12, 2012; revised April 16, 2012
Martin Hruby (phone: +421 2 654 29 502), Michal Olsovsky and Margareta Kotocova are with the Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava, Bratislava, Slovakia (e-mails: hruby@fiit.stuba.sk, olsovsky@fiit.stuba.sk and kotocova@fiit.stuba.sk).

II. Concept

We assume a transit IP network (e.g. service provider backbone) which consists of large number of routers and is using an interior gateway routing protocol (e.g. OSPF). Generally, VoIP flows will traverse the best path as chosen by the routing protocol together with ordinary data flows; however this path doesn't have to provide the best time-variables like delay and jitter which are necessary for VoIP traffic (Fig. 1), as in most cases these parameters are not taken into consideration by the routing protocol [20].

Using standard routing voice-related devices can be placed arbitrarily as there is no specific VoIP backbone [9]. In our approach standard routing will be divided into two independent routing instances, one will be used for data flows (existing one) and the second one will be used inside the determined optimal VoIP backbone based on our proposed network modeling.

As the VoIP backbone has to be a continuous network [10], it will consist not only of the best links chosen by our algorithm but of an additional minimal amount of links which will make the backbone continuous. List of these additional links can be considered for a potential future upgrade.

Knowing the new VoIP backbone we are able to decide where to place and connect specific voice-related devices in the real network environment.

To be able to decide which links will form the VoIP backbone, we need to use some key elements like the proposed cube model, active variable discovery and multivariate normal distribution. These techniques are described in more detail in the following chapters.

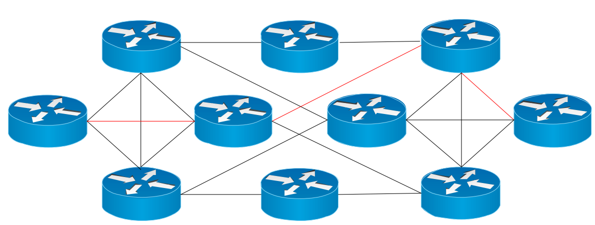

Fig. 1. VoIP and data flows traverse the same best path

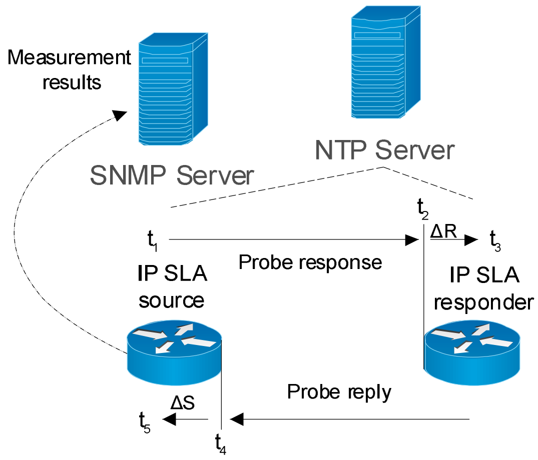

Fig. 2. Measurement overview

A. Variable discovery

Key component to time-variables' measurement is a periodic active measurement probe (SAA). This probe is small datagram sent over specific link towards a known destination. Sender of this probe knows the exact time when the probe was sent out as well as waiting period between consecutive probes. Once the probe reaches its destination, the probe is sent back towards to the original sender. As each probe contains unique identifier, original sender is able to calculate time-variables once the probe is received. Using the probes a network node is able to measure delay and jitter for a specific link in the network.

Every node in the network is capable to reply to delay-based probes; however every node is not suitable for jitter-measurements. For better measurements results we are using IP SLA responder which can be used for all suitable measurements and even subtract the processing delay which occurs in the node when the probe is being processed.

Final phase of the variable discovery is dedicated to the data collection. To facilitate measurement data exports, SNMP is used. In our approach, threshold of each probe will be set to 0. This means that once the probe will be received back, the router will always send an SNMP trap to a central SNMP server with the measurement results (Fig. 2). Once the SNMP server collects a representative amount of the measurement results, these results can be released for further processing (cube model, minimal tree in the graph).

B. Multivariate normal distribution

Measurement results are only group of results. To be able to make decision about their properties, it is necessary to plot these results in a specific model. For our purposes we use model of a cube. As it is 3D model, we are able to use each of the 3 axes for a separate attribute:

- Axis X – variable jitter
- Axis Y – probability density
- Axis Z – variable delay

Each link will have its own cube model with a multivariate normal distribution. Axes X and Z will represent the base of this distribution while axis Y will represent the probability density of delay-jitter pairs. Later on all these multivariate normal distributions will be merged into one cube and processed to determine the best links.

Our objective is to identify specific areas of the plotted surface inside the cube model, which are characterized as peaks of a multivariate normal distribution that represent edges with discovered values of delay and jitter, as depicted in the Fig. 3. We model the multivariate normal distribution peaks by determining parameters of the probability density function of the d-dimensional multivariate normal distribution (see Formula 1) where the parameter μ is a 1-by-d vector of mean values of vectors of gathered variables (delays and jitters) and σ is determined as a covariance matrix of the vectors of gathered variables, as can be seen in Formula 2.

$$y = f(x,\mu,\sigma) = \frac{1}{\sqrt{|\sigma|(2\pi)^d}} e^{-\frac{1}{2}(x-\mu)\sigma^{-1}(x-\mu)'} \quad \textbf{(1)}$$

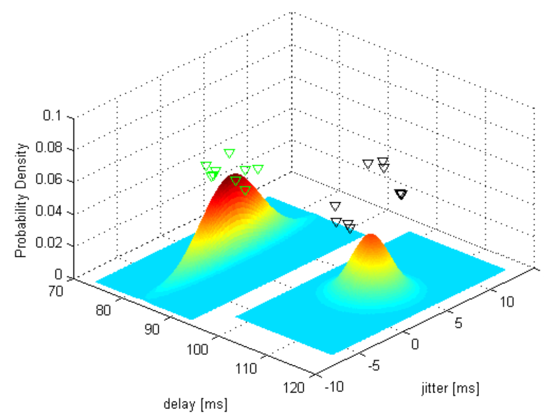

Fig. 3. Model with 2 links

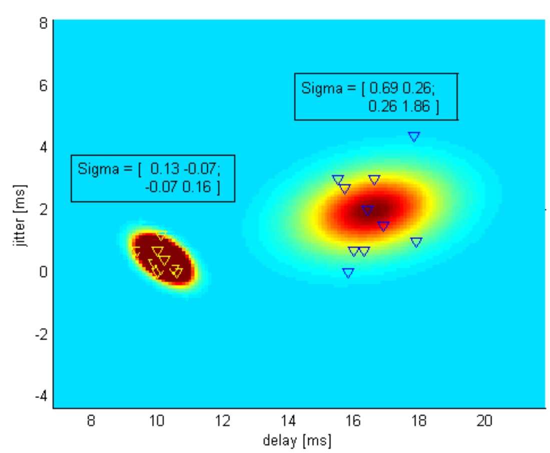

Fig. 4. Trustworthiness of a link in the cube model

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$\mathrm{cov}(x_1, x_2) = E[(x_1 - \mu_1) * (x_2 - \mu_2)] \quad (2)$$

In Formula 2, E[x] is the expected value of x. Representation of the network links as edges with directly measured parameters in the cube model has several advantages.

First, the trustworthiness of a link can be directly determined by the σ parameter as depicted in Fig. 4. The σ parameter corresponds to the steepness of the distribution. Those edges with steeper modeled distributions experience low variation in delay and jitter as opposed to those with gradually distributed probability function.

III. ALGORITHM

Routing of VoIP traffic is operation based on many suboperations. First group of operations is responsible for gathering data via measurements and further processing. Later on, using the multivariate normal distribution new representative results are calculated for every single link and plot into one common model. Finally, we can apply specific rules to determine voice-capable links. However the deployment of this backbone into production network is conditioned with turning these links into continuous graph.

As this algorithm is still under development, it's necessary to provide functional and performance validation. These validations are described in the chapter dedicated to preliminary results.

A. Voice-capable links

The main idea of seeking voice-capable links is to take the cube with multivariate normal distributions of all links and do a three phase filtering. First phase will filter out all unstable links as stable links are important for VoIP backbone. We are able to apply some mechanisms to reduce the effect of worse time-variable in case we can predict these variables. From the point of the cube model and multivariate normal distribution stable links are represented with small base, narrow distribution and high peaks. Small base means that measured values don't differ in significant way and high peak represents the probability density. To be able to find these stable links we have introduced new variable η_1 – threshold of probability density for voice-capable links (Formula 3 and 4). Based on this threshold, new layer (parallel to area of axes X-Z) is added to the final cube model. Every link whose peak exceeds this layer (threshold) has matched the first of three conditions to become voice-capable link.

$$\eta_{\max} = f(x', \mu', \sigma') \geq f(x, \mu, \sigma) \quad (3)$$

$$\eta_1 = 0.8 * \eta_{\max} \quad (4)$$

$$\eta_2 = 0.56 * \eta_{\max} \quad (5)$$

Last two conditions are related to time-variables. Matching just the first conditions isn't enough because this condition will guarantee only the stability of the link – it doesn't matter how high the delay or jitter is, once the values aren't varying excessively, the link is stable. To be able to filter out links with worse time-variables we have introduced another 2 variables (thresholds).

First variable φ_1 is jitter threshold (Formula 7). We don't keep exact jitter, this value represents the fraction of the worst measured result (Formula 6). Based on this threshold another layer (parallel to area of axes X-Z) is added to the final cube model. Every link whose multivariate normal distribution belongs to the area bordered with the new layer and the area of axes X-Z has matched the second condition to become voice-capable link.

$$\varphi_{\max} \in \Phi, \forall \varphi' \in \Phi; \varphi' \leq \varphi_{\max} \quad (6)$$

$$\varphi_1 = 0.8 * \varphi_{\max} \quad (7)$$

$$\varphi_2 = 0.5 * \varphi_{\max} \quad (8)$$

Second variable called χ_1 represents the delay. As for φ_1, we don't keep exact delay value (Formula 10), this value represents the fraction of the worst measured result (Formula 9). Based on this threshold another layer (parallel to area of axes X-Y) is added to the final cube model. Every link who's Gaussian distribution belongs to the area bordered with the new layer and the area of axes X-Y has matched the last, third condition to become voice-capable link.

$$\chi_{\max} \in \mathrm{X}, \forall \chi' \in \mathrm{X}; \chi' \leq \chi_{\max} \quad (9)$$

$$\chi_1 = 0.8 * \chi_{\max} \quad (10)$$

$$\chi_2 = 0.5 * \chi_{\max} \quad (11)$$

To sum it up all three thresholds η_1, φ_1 and χ_1 and appropriate layers created sub-cube model in the main cube model and all links with their peaks in this sub-cube model are voice-capable links.

With thresholds calculated it is now possible to separate measured data in the cube model into regions. Each region Ω is defined by upper and lower planes and links are considered to be part of a region when the link's probability function maximum is located between the regions planes (Formulas 12, 13, 14).

$$\Omega_1 = \begin{bmatrix} \varphi_1 & \chi_1 & \eta_{\max} \\ 0 & 0 & \eta_1 \end{bmatrix} \quad (12)$$

$$\Omega_2 = \begin{bmatrix} \varphi_2 & \chi_2 & \eta_1 \\ \varphi_1 & \chi_1 & \eta_2 \end{bmatrix} \quad (13)$$

$$\Omega_3 = \begin{bmatrix} \varphi_{\max} & \chi_{\max} & \eta_2 \\ \varphi_2 & \chi_2 & 0 \end{bmatrix} \quad (14)$$

A continuous sequence of links will be called a VoIP backbone component. To calculate a single VoIP backbone, all unique VoIP backbone components must be connected by a minimal series of other links (edges). To be able to make the connection of all VoIP backbone components into one single component with the best of remaining links, these links needs to be classified. As this connection process will be based on the theory of graphs, we need to assign each link one composite metric – weight of the edge.

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

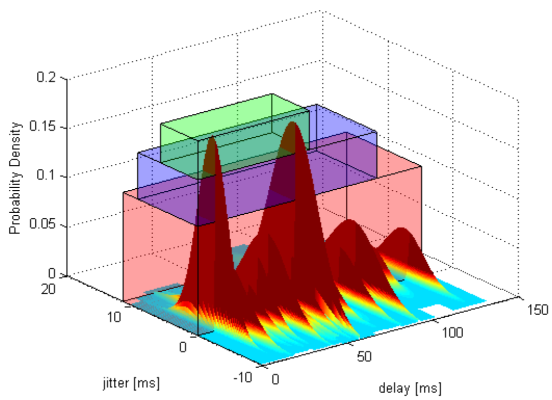

Fig. 5. Link classification for VoIP backbone

We have decided to define another set of thresholds – η_2, φ_2 and χ_2 which will be worse than the original set of threshold (Formula 5, 8 and 11). These thresholds will divide the cube model into 3 regions (Fig. 5). Green region contains voice-capable links (region Ω_1). For future processing, these links have metric of 1. Purple region (Ω_2) contains link which aren't suitable VoIP backbone but can be used for voice flows. All these links have metric of 2. Last region (Ω_3), pink region represents the rest of the links which aren't suitable for VoIP traffic and should be used only as a temporary solution. These links have metric of 5.

Based on these details, the whole network topology can be represented by a single table. This table will be used during the last phase of our proposed algorithm where the final single VoIP backbone will be created. This table will keep following details (Table 1):

- Link ID – unique identifier of the link.
- Start Node – identifier of the first of two routers which this link connects.
- End Node - identifier of the second of two routers which this link connects.
- Metric – metric based on the link's situation in the cube model with 3 regions.

B. *Minimal tree*

The process of fetching the minimal tree is based on graph theory. Graph is an ordered pair G = (V, E) comprising a set V of vertices or nodes together with a set E of edges or lines, which are 2-element subsets of V [11], [13] – our requirements best fits the undirected weighted graph.

This graph is represented with the dataset stored in table with structure as in Table 1. Together with this, it's necessary to define group of border routers *BORDR* (vertices) which needs to be included in the final backbone (tree). Next steps are as follows:

1. Select all voice-capable links (edges with *Metric = 1*) from the table and store them in variable *VC_LINKS*.
2. Create subgraphs from *VC_LINKS*. In case start vertex of one edge is end of another edge, these edges can be merged into vertex.
   Repeat this operation for every single edge from *VC_LINKS*. All created subgraphs will be stored in *SUBGR*.
3. Check if vertices in *BORDR* are present in *SUBGR*. In case vertex from *BORDR* is not present in *SUBGR*, create new subgraph in *SUBGR*. New subgraph will contain only 1 vertex from *BORDR*. Repeat for every vertex in *BORDR*.
4. Use Dijkstra's algorithm to find the best (cheapest) path between every pair of vertices in the main graph [11, 12]. Store information for each vertex in separate table.
5. Find the best path between every pair of subgraphs in *SUBGR*. As the graph is undirected, path between subgraph 1 and 2 is the same as path between subgraph 2 and 1. Take first pair of subgraphs (*S1, S2*) and first vertex (*VS1, VS2*) from each subgraph. Check the cost from VS1 to VS2 using tables calculated in step 4.
   Put all these values (*S1 as* ω_1, *VS1 as* θ_1, *S2 as* ω_2, *VS2 as* θ_2, *Cost, Use=0*) into table for paths with structure as in Table 2. Table for paths will have *(n.(n-1))/2* rows, where *n* stands for the number of subgraphs in *SUBGR*.
   Repeat for every combination of vertices of selected two subgraphs. If path cost is lower while using another pair of vertices, replace details about cost and vertices in the path table for specific pair of subgraphs.
   Repeat for each pair of subgraphs.
6. Complete table of path is representation of another full-mesh graph. To get the minimum spanning tree for this connected weighted graph use the Kruskal's algorithm [13]. Mark the edges of the minimum spanning tree in the table of paths with *Use=1*.
7. Find every path with *Use=1*. Find all vertices from θ_1 to θ_2 for that specific row in path table and mark this vertices with *Tree=1* in the table of the main graph.
8. Find every vertex with metric of 1 in the table of the main graph and mark it with *Tree=1*.
9. Links in the table of the main graph marked with Tree = 1 creates VoIP backbone. Part of these links with *Metric=1* is the main backbone. The other links are capable for upgrade.
10. Plot results into graph with highlighted border routers, VoIP backbone and temporary links.

Table 1. Data structure for main graph

Link ID	Start Node	End Node	Metric	Tree

Table 2. Data structure for paths

ω_1	θ_1	ω_2	θ_2	Cost	Use

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

IV. PRELIMINARY TESTING

A. *Functional verification*

To test our concept we gathered samples from active measurements performed in a real network environment. These measurements conducted over a period of 24 hours represent the average values of delay and jitter (averaged per hour) on 20 high-speed WAN links (Fig. 6). The gathered data was fed to our proposed cube model in Matlab and results were evaluated both statistically and graphically. The measured data are plotted in Fig. 7 and Fig. 8, were first 10 links are plotted in Fig. 7 and the remaining 10 links in the Fig. 8.

Based on the results only two links (#9, #11) are fully VoIP capable, another two links (#1 and #19) have a metric of 2 while all the other links have metric of 5. Links #5 and #11 create one subgraph BFH, others subgraphs are border routers A and J (border router B is already included in BFH). Following the algorithm variable *SUBGR* contains {BFH, A, J}. Following the Dijkstra's and Kruskal's algorithms and the last 3 steps of the algorithm we are able to create VoIP backbone with connection to all border routers. Final network topology (Fig.9) contains voice capable (solid), usable (dashed) and no temporary (dotted) lines.

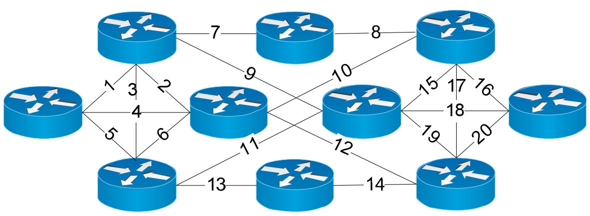

Fig. 6. Network topology

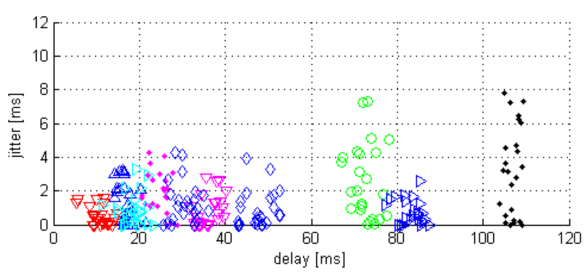

Fig. 7. Measured 24 hours delay and jitter on 20 WAN links (part 1/2)

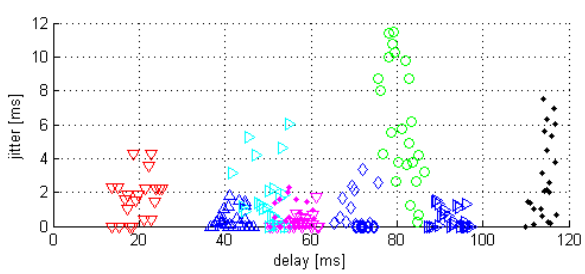

Fig. 8. Measured 24 hours delay and jitter on 20 WAN links (part 2/2)

B. *Performance verification*

During preliminary testing, besides verifying the correctness of our approach we also measured the computing performance of the model and the distribution of network links elected into the VoIP backbone. For these purposes we have generated 3 independent random sets of networks ranging in size from 10 to 1000 routers. For each such randomly generated network a distribution of network performance parameters was assigned to links which followed usual distribution of such parameters as measured previously for a 24 hour period.

Our experimental findings are summarized in Fig. 10 which represents the percentage of links elected into the VoIP backbone for a randomly generated network with a given number of routers. In Fig. 11 we list the high and low average distributions corresponding to Fig. 10. In Fig. 12 we list the time performance of our model on the same set of randomly generated networks. In Fig. 13 we present the high and low averages of time performance corresponding to Fig. 12.

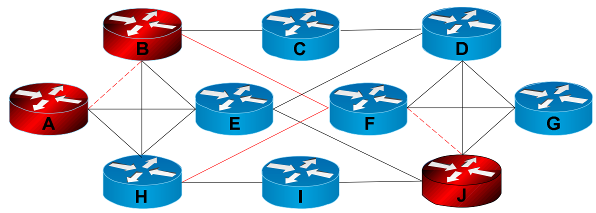

Fig. 9. VoIP backbone connected to border routers

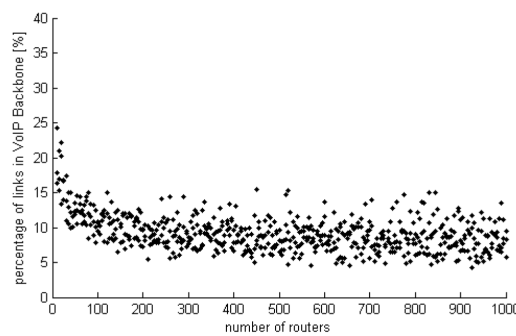

Fig. 10. Percentages of links inside the VoIP backbone

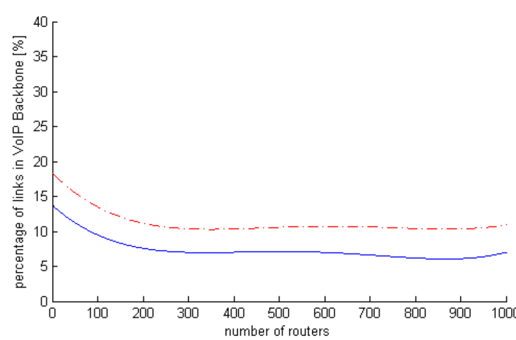

Fig. 11. High and low averages of links in the VoIP backbone

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

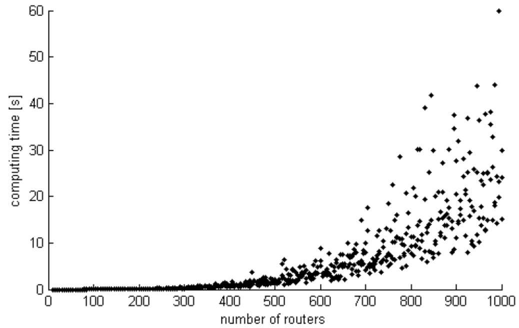

Fig. 12. Experimental computing time

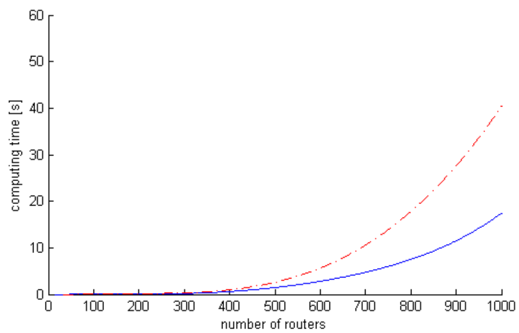

Fig. 13. High and low averages of the computing time

V. Conclusion

In this paper we proposed a novel method of routing VoIP traffic in large networks. By performing periodic measurements of network performance parameters (delay and jitter) we model the probability distributions of these parameters for each link and use the model as input for a path selection algorithm based on a modification of well known Dijkstra's and Kruskal's algorithms. In this way we establish a VoIP backbone which is a network overlay suitable for VoIP traffic deployment.

We list preliminary functional results proving the correctness of our approach and also performance results conducted on randomly generated networks with random statistically generated network performance parameters. Our modeling approach and path selection algorithm performed well, choosing an average of 10% of all available network links into the VoIP backbone. We believe that our approach presents a useful method for VoIP implementation in planning and production phases of large network deployment.

Acknowledgements

The support by Slovak Science Grant Agency (VEGA1/0676/12 "Network architectures for multimedia services delivery with QoS guarantee") is gratefully acknowledged.

References

[1] Gunnar, A.; Johansson, A.; Telkamp, T.: "Traffic Matrix Estimation on a Large IP Backbone: A Comparison on Real Data", *In Proceedings of the 4th ACM SIGCOMM conference on Internet measurement (IMC '04)*, October 25-27, 2004, pp.149-160

[2] Fortz, B.; Rexford, J.; Thorup, M.: "Traffic engineering with traditional IP routing protocols", *In IEEE Communications Magazine*, vol. 40, no. 10; December 10, 2002, pp.118-124

[3] Kun-chan Lan; Tsung-hsun Wu: "Evaluating the perceived quality of infrastructure-less VoIP," *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, 11-15 July 2011, pp.1-6

[4] Wang, X.; Wan, S.; Li, L.: "Robust Traffic Engineering Using Multi-Topology Routing", *In Global Telecommunications Conference*, 2009, pp.1–6

[5] Abrahamsson, H.; Bjorkman, M.: "Robust traffic engineering using l-balanced weight-settings in OSPF/IS-IS", *In Broadband Communications, Networks, and Systems (BROADNETS)*, 2009, pp.1-8

[6] Hyeongu Son; Youngseok Lee: "Detecting Anomaly Traffic using Flow Data in the real VoIP network," *Applications and the Internet (SAINT), 2010 10th IEEE/IPSJ International Symposium on* , 19-23 July 2010, pp.253-256

[7] Butcher, D.; Xiangyang Li; Jinhua Guo: "Security Challenge and Defense in VoIP Infrastructures," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol.37, no.6, Nov. 2007, pp.1152-1162

[8] Narayan, S.; Yhi Shi: "Application layer network performance evaluation of VoIP traffic on a test-bed with IPv4 and IPv6 LAN infrastructure," *Computational Technologies in Electrical and Electronics Engineering (SIBIRCON), 2010 IEEE Region 8 International Conference on*, 11-15 July 2010, pp.215-219

[9] Okech, J.M.; Hamam, Y.; Kurien, A.: "A Cross-Layer Adaptation for VoIP over Infrastructure Mesh Network," *Broadband Communications, Information Technology & Biomedical Applications, 2008 Third International Conference on*, 23-26 Nov. 2008, pp.97-102

[10] Feng Cao; Malik, S.: "Vulnerability analysis and best practices for adopting IP telephony in critical infrastructure sectors," *Communications Magazine, IEEE*, vol.44, no.4, April 2006, pp.138- 145

[11] Bing Liu; Siew-Hwee Choo; Shee-Ling Lok; Sing-Meng Leong; Soo-Chee Lee; Foong-Ping Poon; Hwee-Har Tan: "Finding the shortest route using cases, knowledge, and Djikstra's algorithm," *IEEE Expert* , vol.9, no.5, Oct 1994, pp.7-11

[12] Noto, M.; Sato, H.: "A method for the shortest path search by extended Dijkstra algorithm," Systems, Man, and Cybernetics, 2000 IEEE International Conference on, vol.3, 2000, pp.2316-2320

[13] Guttoski, P.B.; Sunye, M.S.; Silva, F.: "Kruskal's Algorithm for Query Tree Optimization," *Database Engineering and Applications Symposium, 2007. IDEAS 2007. 11th International*, 6-8 Sept. 2007, pp.296-302

[14] Goldschmidt, O.: ISP Backbone Traffic Inference Methods to Support Traffic Engineering. *In Internet Statistics and Metrics Analysis Workshop (ISMA'00)*; December 7-8, 2000.

[15] Tebaldi, C.; West, M.: Bayesian Inference on Network Traffic Using Link Count Data; Journal of the American Statistical Association; vol. 93, no. 442; June 1998; ISSN: 0162-1459; pp. 557-576.

[16] Cao, J.; Davis, D.; Vander Wiel, S.; Yu, B.: Time-Varying Network Tomography: Router Link Data; Journal of the American Statistical

[17] Hansen, T.J.; Morup, M.; Hansen, L.K.; , "Non-parametric co-clustering of large scale sparse bipartite networks on the GPU," *Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on* , vol., no., pp.1-6, 18-21 Sept. 2011

[18] Casas, P.; Vaton, S.; Fillatre, L.; Chonavel, L.; , "Efficient methods for traffic matrix modeling and on-line estimation in large-scale IP networks," *Teletraffic Congress, 2009. ITC 21 2009. 21st International* , pp.1-8, 15-17 Sept. 2009

[19] Haoxiang Wang; , "From a Mess to Graphic Maps: Visualization of Large-Scale Heterogeneous Networks," *Computer Modeling and Simulation, 2010. ICCMS '10. Second International Conference on* , vol.1, pp.531-535, 22-24 Jan. 2010

[20] Jin Xiao; Boutaba, R.;, "QoS-aware service composition in large scale multi-domain networks," *Integrated Network Management, 2005. IM 2005. 2005 9th IFIP/IEEE International Symposium on*, pp. 397- 410, 15-19 May 2005