Implications for QoS provisioning based on traceroute measurements

Janic, M; Kuipers, FA; Zhou, X; Van Mieghem, PFA

DOI
10.1007/3-540-45859-X_1

Publication date
2002

Document Version
Accepted author manuscript

Published in
From QoS provisioning to QoS charging; Third COST 263 international workshop on quality of future internet services, QofIS 2002 and second international workshop on internet charging and QoS technologies, ICQT 2002

Implications for QoS provisioning based on traceroute measurements

Milena Janic, Fernando Kuipers, Xiaoming Zhou and Piet Van Mieghem

Delft University of Technology
Information Technology and Systems
P.O. Box 5031, 2600 GA Delft, The Netherlands
{M.Janic, F.A.Kuipers, Z.Xiaoming, P.VanMieghem}@its.tudelft.nl

Abstract. Based on RIPE NCC *traceroute* measurement data, we attempt to model the Internet graph with the graph G_1, which is the union of the most frequently occurring paths in the RIPE *traceroute* database. After analyzing the topological properties of this graph, a striking agreement with the properties of the uniform recursive tree has been found. This similarity allows us to compute the efficiency of multicast. After assigning polynomially distributed link weights to the edges of the graph G_1 and comparing the simulated shortest paths with the *traceroutes*, no clear conclusions on the link weight structure could be deduced. Finally, possible implications of the measurement data on the provisioning of Quality of Service are suggested.

1. INTRODUCTION

Many researchers have attempted to provide a map of the Internet as well on a router-level as on an Autonomous Systems (AS) level [11][7][3]. In 1995, Pansiot and Grad [11] created a router-level topology based both on *traceroutes* to 5000 geographically distributed destinations from a single node, as well as on *traceroutes* from a subset of 11 nodes chosen from the set of 5000 nodes to the rest of the destinations. Govindan and Tangmunarunkit [7] obtained a snapshot of the Internet topology by using a Mercator program. Mercator is designed to map the network from a single source location without an initial database of target nodes for probing.

Most of the researchers have used the *traceroute* utility [12] for acquiring a map of Internet. We have analyzed and categorized errors occurring in *traceroute* measurements obtained from RIPE NCC, and based on this *traceroute* data, we attempt to create a graph that will approximate a part of Internet. Moreover, by assigning link weights that are polynomially distributed, and comparing simulated paths with real *traceroute* data, we try to gain better knowledge of the Internet link weight distribution. Investigating the possible implications of *traceroute* measurements on the provisioning of Quality of Service (QoS) has, to our knowledge, never been done.

The remainder of the paper is organized as follows: The construction of the approximate Internet graph and its topological properties (hopcount, node degree) are described in detail in Section 2. The same Section presents the results of the analysis

of the Internet link weight structure and of the Internet path dominance. The possible implications of *traceroute* measurements for multicast deployment and end-to-end QoS are addressed in Section 3. Finally, we conclude in Section 4.

2. APPROXIMATING THE INTERNET GRAPH

In this section we try to model the Internet based on *traceroute* data provided by RIPE NCC (the Network Coordination Centre of the Réseaux IP Européen). At the moment, RIPE NCC performs *traceroute* measurements between 50 measurement boxes scattered over Europe (and few in the US and New Zealand). Every 40 seconds, probe-packets of fixed size have been sent between each pair of measurement boxes, and the data has been collected once a day in a central point in RIPE (for a further detailed description of the measurements and measurement configuration we refer to [13]). The graph G_1 is the union of the most frequently occurring *traceroute* paths, provided from measurements performed by RIPE NCC in the period 1998-2001. Based on G_1 and the *traceroute* database, we will investigate whether the Internet possesses a polynomial link weight structure. We conclude this section, by evaluating how stable Internet interface paths really are.

2.1. CONSTRUCTING G_1

For each pair of measurement boxes a large number of different paths have been distinguished in the database. Obviously, some of the *traceroute* records suffer from errors. For 14 test-boxes (out of 50) no record in the database could have been found. Moreover, *traceroute* records for 5 test-boxes have all been erroneous (temporary or persistent loops and unresponsive routers). Therefore, these 19 test-boxes have been excluded from further analysis. In the *traceroute* data from the remaining 31 boxes the most dominant path, i.e. the path occurring most frequently has been determined, resulting in totally 465 most dominant paths. We ascertained further that 17% of the those *traceroutes* suffer from errors mentioned above. The graph G_1 has been created by including every link belonging to each of the remaining 386 non-erroneous paths, resulting in a graph consisting of 1888 nodes and 2628 edges. Here we must make one important remark. The *traceroute* utility returns the list of IP addresses of routers along the path from source to destination. One router can have several interfaces, with several different IP addresses. To determine which IP addresses belong to one router is a rather difficult task, due to, among others, the security reasons (port snooping). As a consequence of this, the graph G_1 represents the approximation of the Internet interface map, not of the Internet router map.

In Subsection 2.3 we will show that the most dominant path alone does not cover the majority of paths between a particular source and destination. Intuitively, a much better approximation to the Internet graph would be a graph created as the union of k most frequently occurring paths. In our study, we have also considered the graph G_{10}, constructed as the union of ten most frequently occurring paths, and we have discovered that properties of G_{10} still resemble those of G_1. Therefore, due to the

higher complexity of creating G_k, in further analysis we will confine ourselves to G_1. Let us remark that although the graph G_1 differs from the real underlying Internet graph G_{INT}, (because it is an overlay network on G_{INT}) it seems to present a reasonable approximation to G_{INT} (or a part of G_{INT}). Moreover, nearly 85% of the total number of nodes is already spanned by the most dominant *traceroutes* from (any) 18 sources to all destinations. Including the most dominant *traceroutes* of a new source only adds 1.5% of new nodes to the topology.

2.2. THE TOPOLOGICAL PROPERTIES OF G_1

In Figure 1 and Figure 2, the probability density function of the hopcount and node degree in the graph by G_1 has been plotted, respectively. The data of the hopcount has been fitted with a theoretical law (4.1) derived in [15] based on the random graph with uniformly (or exponentially) distributed link weights. The essential observation in the modeling [15] is that the shortest path tree in the random graph with uniformly (or equivalently exponentially) distributed link weights is independent of the link density p. This implies that even the complete graph with uniformly distributed link weights possesses precisely the same type of shortest path tree as for any connected random graph with uniformly distributed link weights. In other words, to obtain a source-based shortest path tree, the link weight structure is more decisive than the underlying random topology. In [15] has been shown that the shortest-path problem in the class of random graph *Gp(N)* [2] with exponentially distributed link weights can be reformulated into a Markov discovery process with an associated uniform recursive tree [14]. A uniform recursive tree is defined by the following growth rule: given a uniform recursive tree with *N* nodes, a uniform recursive tree with *N*+1 nodes is deduced by attaching the *N*+1-th node uniformly (thus with probability 1/*N*) to each of the *N* other nodes in the tree. If the number of nodes $N \to \infty$, the probability density function of the hopcount in the uniform recursive tree is shown [15] to obey precisely

$$\Pr[h_N = k] = \frac{(1+o(1))}{N} \sum_{k=1}^{N-1} c_{m+1} \frac{\ln^{k-m} N}{(k-m)!} \qquad (\mathbf{1})$$

where c_k are the Taylor coefficients of $\Gamma(x)^{-1}$ listed in [1]. The law (1) can be approximated by a Poisson law,

$$\Pr[h_N = k] \approx \frac{(E[h_N])^k}{k!N} \qquad (\mathbf{2})$$

which implies that $E[h_N] \approx var[h_N] \approx lnN$.

Figure 1 illustrates that a fit with (1) is reasonable, but the quality remains disputable. A more striking agreement with the uniform recursive tree is shown by the degree law in Figure 2. We observe that the pdf of the node degrees in G_1 follows an exponentially decreasing function with rate –0.668 over nearly the entire range. The

ratio of the average of the number of nodes with degree k, denoted by D_N^k, over the total number of nodes in the uniform recursive tree obeys for large N [14]

$$\frac{E[D_N^k]}{N} = \frac{1}{2^k} + O\left(\frac{\log^{k-1} N}{N^2}\right) \tag{3}$$

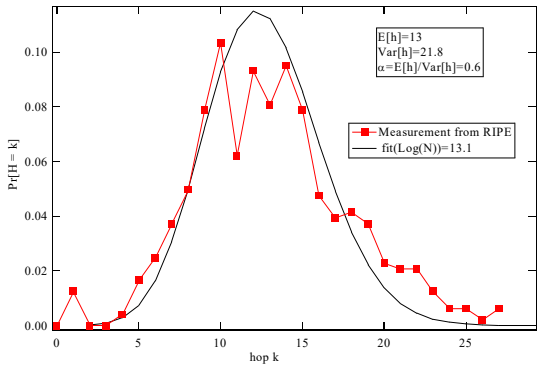

Fig. 1. *Both measurements and theory for pdf of hopcount in G_I with N = 1888*

which is, for large N, close to *Pr[degree = k]*, the probability that an arbitrary node has degree k. Hence, the decay rate of the pdf of the node degrees in the uniform recursive tree equals –ln2 = 0.693. In summary, the pdf of the node degrees in G_I (the union of the most dominant non-erroneous *traceroutes*) follows a same law as that in the uniform recursive tree. This intriguing agreement needs additional comments. It is not difficult to see that, in general, the union of two or more trees (a) is not a tree and (b) has most degrees larger than that appearing in one tree. Hence, the close agreement points to the fact that the intersection of the trees rooted at a measurement box towards the other boxes is small such that the graph G_I is 'close' to a uniform recursive tree. This conclusion is verified as follows. Starting with the first union $T = T_1 \cup T_2$ of the trees T_1 and T_2, we also have computed the intersection $T_1 \cap T_2$. Subsequently, we have added another tree T_3 to the union $T = T_1 \cup T_2 \cup T$ and again computed the intersection (or overlap) $(T_1 \cup T_2) \cap T_3$. This process was continued until all trees were taken into account. We found that (a) the number of nodes in the intersection was very small and (b) the common nodes predominantly augmented the degree by 1 and in very few cases by more than 1.

It is likely that the overlap of trees would be larger (in terms of common nodes and links) if we had considered the router level map instead the interface map. The similarity in properties of the graph G_I and the uniform recursive tree might be in that case smaller. This might explain the discrepancy with the results of Faloutsos et al. [5], who have reported a power law for the degrees in the graph of the Internet. This seems to suggest that G_I is not representing the graph of the Internet well. Or, put differently, the union of *traceroutes* seems to possess a different structure than the underlying graph G_{INT}. Another possible reason for this discrepancy is that the number of test boxes (and path trees) considered here is small compared to the number of

nodes. Finally, a source tree in Internet seems thus to be well modeled by imposing uniform or exponential link weight structure on the topology. Both uniform and exponential distributions have the same minimal domain of attraction [15]. Furthermore, this link weight structure models the end-to-end IP paths consisting of concatenations of shortest paths sections [intra-domain routing] "randomized" by BGP's policy-driven path enforcements [inter-domain routing].

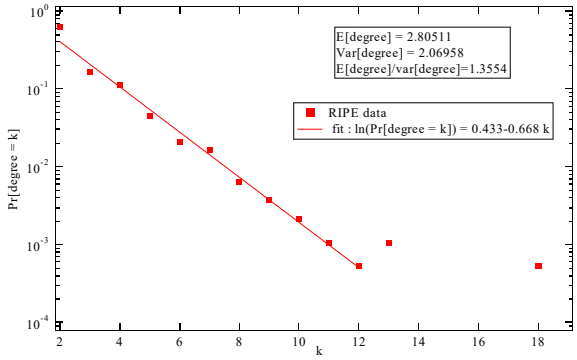

Fig. 2. *The pdf of the node degree (RIPE data) in G1 with N = 1888*

2.3. THE ANALYSIS OF THE LINK WEIGHT STRUCTURE

In addition to the network topology, knowledge on the link weight distribution is needed in order to solve a particular routing problem. In the Internet topology models proposed earlier, all links weights have unit weight (w=1). However, weights on Internet are assigned according to physical properties, such as link propagation delay or capacity, or in order to enable directing traffic along certain preferred paths, and hence, assigning equal value to all links is not justified. The next step in our analysis was to gain some more insight in the effective Internet link weight structure. In order to perform such an analysis, weights have been assigned to the edges in G_1, and subsequently, for each source-destination pair, the shortest path in G_1 has been compared to the most dominant *traceroute* path. For reasons explained in [15] we have chosen a polynomial distribution for the link weights. Hence, for each link between nodes i and j in G_1, the link weight obeys $P[w \le x] = x^{\alpha} 1_{0 \le x \le 1} + 1_{x \ge 1}$, for various values of the exponent α (α= 0.05, 0.2, 0.5, 1, 2, 5, ∞). If sufficiently many values of α are applied, we may assume that the value of a that leads to the minimum number of dissimilarities with Internet *traceroutes*, may represent the best polynomially distributed match for the Internet link weight structure. A minor secondary reason for choosing polynomially distributed link weights is motivated by [9]. The asymptotic analysis in [9] reveals that, for random graphs with uniformly (or equivalently exponentially) distributed, independent link weights, the hopcount in r dimensions behaves similarly as in the random graph in r=1 dimension with polynomially distributed link weights, where the polynomial degree a is equal to the dimension r.

For each value of the exponent α, 300 replicas of G_1 were generated, and in each of them, for each source-destination pair, the shortest path was computed using Dijkstra's algorithm and compared to the corresponding traceroute in the database.

Figure 3 shows the difference in hopcount $\Delta=h_{RIPE}-h_{SIM}$ as a function of the exponent α. This figure immediately reveals that a small α is undoubtedly not a good model for link weights on the Internet, since the hopcount of the shortest path can be considerable longer than the traces. With the increase of α, the negative tails decrease in the favor of positive values of Δ_h. Only when $\alpha=\infty$ (Figure 4), corresponding to all link weights being precisely equal to 1, the shortest paths are always shorter than or equal to the *traceroutes*. Even then *traceroutes* are minimal hopcount paths only in about 22% of the cases. These results seem to indicate that either a link weight structure in the Internet is difficult to deduce when assuming that link weights are polynomially distributed or that the latter assumption is not valid. Even the case where all link weights are equal to 1 does not perform convincingly.

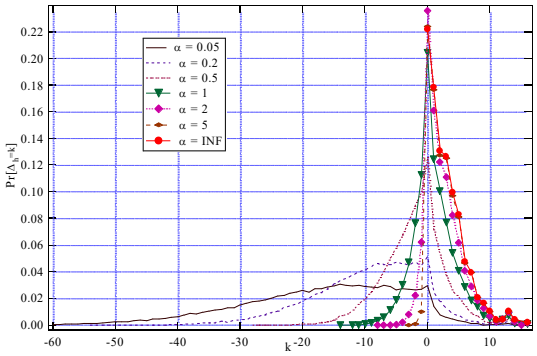

Fig. 3. *The probability density function of the different number of hops in RIPE traces and in simulated shortest paths.*

Inside AS, the intra-domain routing is based on a shortest path criterion, but routing between AS (BGP) does not clearly follow a minimization criterion [8]. Hence, e2e routing may result in paths that obey a different criterion, not necessarily a minimization. This may explain the poor agreement with our simulations.

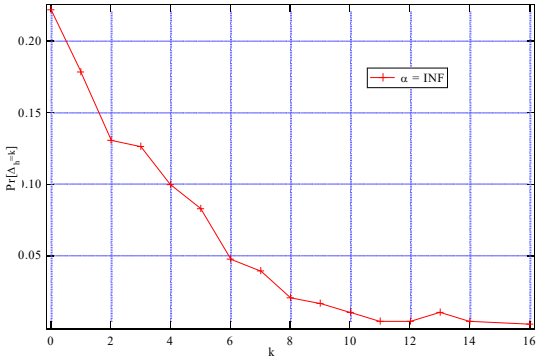

Fig. 4. *The probability density function of the different number of hops in RIPE traces and simulated shortest paths ($\alpha=\infty$)*

2.4. HOW DOMINANT ARE INTERNET PATHS?

For each source-destination pair (tt_i, tt_j) the number of different interface paths collected in the database is very large, rising up to an amazing number of 9032 different interface paths for some (tt_i, tt_j) pairs, in a period of roughly three years! With the goal of establishing the critical number of different paths that constitute the majority of paths, we evaluated the probability density function and the probability distribution function that a particular non-erroneous path P is the kth dominant path.

We observed that for most source-destination pairs, there is no clear dominant path, which has been illustrated in Figure 5, where we have plotted, for 546 source-destination pairs, the probability that the most dominant path will be chosen. On average, the probability of taking the most dominant path equals 0.13, with a variance of 0.01. This indicates that IP packets regularly follow different paths over the three years time period. This behavior may have several causes:

1. Growth of the Internet: Over the years, the Internet has grown with respect to the number of nodes and links. This growth and increased link-density has resulted in an increased number of paths between nodes. It is therefore likely that over time better paths emerge, becoming most dominant paths.
2. Load balancing: Through load balancing, Internet providers try to relieve the shortest most dominant paths in order to avoid overloading them.
3. Multi-homing: Multi-homing refers to having multiple connections or links between two parties. For instance, we can have multi-homing of hosts, networks and Internet service providers. The reason for doing this is to increase robustness (back-up paths) and for load sharing. A consequence of multi-homing is the increase in the size of the BGP routing tables.
4. Changing SLAs: Routing on the inter-domain level is subject to policy constraints (Service Level Agreements). These policies may change or new policies may appear, affecting traffic behavior and routing possibilities.

5. Failures: Our measurements seem to indicate that the Internet has lost in stability and robustness over the past five years. As failures occur more frequently, packets may more often deviate from the most dominant path.

It is debatable whether the behavior in Figure 5 is disturbing or not. If the behavior is mostly due to causes 1-4, then there are no big concerns. However, if the lack of dominant paths is caused by network failures, this would be an alarming situation. Currently the update of router tables only occurs at relatively large intervals of time (15 min.), when some node/link failure has occurred. With QoS routing, we will need to trigger flooding, based on dynamic QoS parameters, like bandwidth and delay. If there are many short-term QoS applications, this will lead to an even greater lack of dominant paths. On the other hand, it is conceivable that for instance a corporate network will request a certain QoS connection for a large period of time (e.g. 1 year) for its users. Such a connection/path will become very dominant.

In [10] it is argued that the lack of interdomain failover due to delayed BGP routing convergence will potentially become one of the key factors contributing to the "gap" between the needs and expectations of today's data networks. In the worst case this delay can grow exponentially with the number of ASs. This can seriously impact the provisioning of QoS, which assumes a stable underlying interdomain forwarding infrastructure and fast IP path restoral. With the emerging of QoS the update frequency will probably increase, leading to an even worse scenario. It is therefore important to have a "smart" update strategy and perhaps desirable to structure the Internet into more hierarchies, reducing the number of domains and updates per hierarchy level.

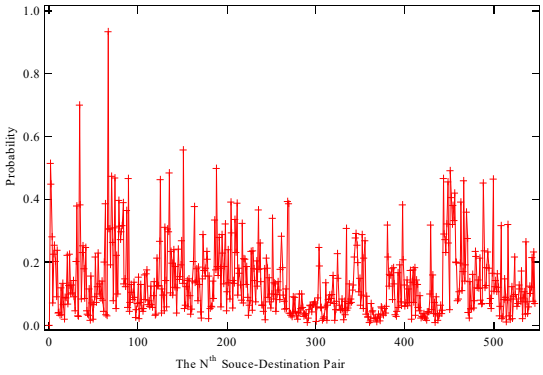

Fig. 5. *For each source-destination pair the probability that the most dominant path will be chosen*

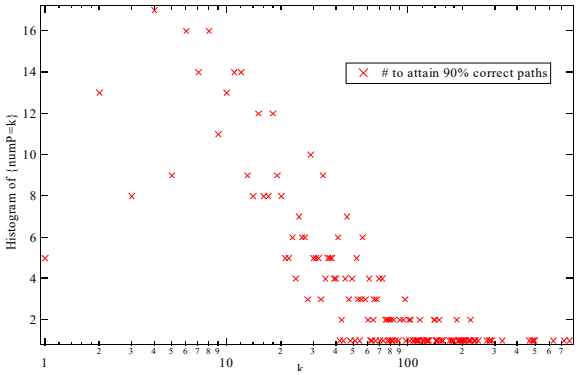

Fig. 6. *The histogram of the number of the non-erroneous paths per source-destination pair over about 3 years time of which k paths are used more than 90% of the time*

Figure 6 represents a histogram of the number of source-destination pairs of which the number of dominant paths, denoted by *numP*, necessary to satisfy the condition Pr[P≤ k]= 90% (i.e. which are used more than 90% of the time). We can see that the number of paths is relatively large and varies considerably. The reasons are similar to the 5 reasons listed above. Nonetheless, it would be desirable to evaluate whether those are truly the reasons and if so, which of them is the most dominant.

3. IMPLICATIONS FOR FUTURE INTERNET SERVICES

3.1. IMPLICATIONS FOR MULTICAST DEPLOYMENT

The indication that the graph G_1 is similar to a uniform recursive tree (see Section 2) has an implication to multicast deployment. The purpose of multicast is to provide savings in bandwidth consumption. However, the network cost for multicast protocols is larger than that of unicast. The computational and administrational overhead of multicast group management increases the deployment cost. Clearly, the deployment of multicast can only be justified if the netto gain defined as savings minus costs is larger than the netto gain for unicast. Therefore, in order to encourage the deployment of multicast on a larger scale, a break-even analysis in the number of users can determine when multicast is more profitable than unicast. To our knowledge, none of the business model proposals to this moment proposes a method for computing the threshold value for the number of users.

In order to define this break-even value, it is necessary to quantify the cost and the gain of multicast. Defining the gain of multicast has been initiated by Chuang and Sirbu [4], but Van Mieghem et al.[17] have presented the general framework, valid for any graph. Van Mieghem et al. derived in [17] the exact mathematical expression for the multicast efficiency over unicast, in terms of bandwidth utilization, in the random graph *Gp(N),* with independent exponentially distributed link weights w, and

m multicast users uniformly chosen out of the total number of nodes N. As already mentioned, the shortest path tree deduced from the Internet measurements was shown to be very close to a uniform recursive tree. For this uniform recursive tree, the corresponding multicast efficiency g_N, defined as the average number of hops in the shortest path tree rooted at a particular source to m randomly chosen destinations, for N large (and all m) is given by

$$g_N(m) \sim \frac{mN}{N-m} \log\left(\frac{N}{m}\right) - 0.5 \qquad \textbf{(4)}$$

Since we have shown that the graph G_1 or even better, the shortest path tree from a source to m destinations has properties similar to those of the uniform recursive tree, we were triggered to compare the multicast efficiency in shortest path trees in the graph G_1 with the one computed with the theoretical law (5). The simulations have been conducted as follows: one node, out of 1888 nodes in G_1, has been uniformly chosen to represent a source, and from this source a shortest path tree to m randomly chosen destinations has been constructed. For each number of users m, 105 different combinations of source and destinations have been chosen, and subsequently 105 different shortest path trees have been constructed. The number of users m has been chosen to be small, (the ratio $x=m/N$ is smaller than 2%).

In Figure 7, we plotted the simulation data together with the law (5), which shows a good agreement. It has been shown in [17] that the average hopcount in the r. g. obeys

$$E[h_N] = \log N + \gamma - 1 + o(1) \qquad \textbf{(5)}$$

where γ is Euler's constant (γ = 0.5772156…). Since unicast uses on average $f_N(m)=mE[h_N]$ links, the ratio $g_N(m)/f_N(m)$ could be used as a good estimate for gain of multicast.

The remaining problem in the break-even analysis of the number of users reduces to quantifying the cost. In other words, all factors that impact the cost of multicast (such as additional cost in construction of multicast trees and in the multicast protocol, maintaining multicast state at routers, etc.) need to be quantified. This not a trivial task is, however, a topic for further research.

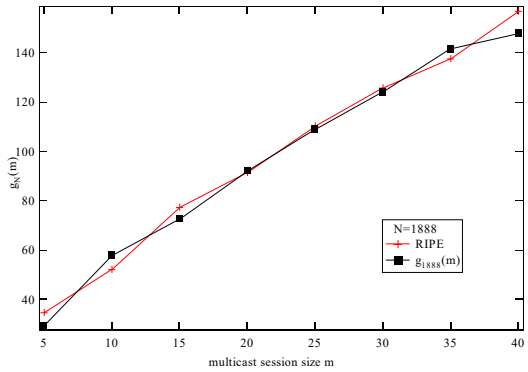

Fig. 7. *The efficiency of multicast*

3.2. IMPLICATIONS FOR E2E QOS

The traceroute measurements are end-to-end (e2e) measurements in a part of the current Internet. The fact that there is no QoS architecture implemented yet, prohibits us to measure a "QoS-behavior" in the Internet. Nonetheless, we will reflect on how the traceroute measurements may provide some insight into an important question: Is the current Internet suitable for delivering guaranteed end-to-end QoS?

In order to be able to answer this question, we must first determine the necessary requirements for delivering guaranteed end-to-end QoS. We will restrict ourselves to only enumerate the requirements relevant to this paper:

· As in current network routing, QoS routing must consist of two identities, namely (1) a QoS routing protocol and (2) a QoS routing algorithm. The QoS protocol should distribute the state of the network along with the relevant QoS parameters (e.g. available bandwidth and delay) as efficient as possible, such that each node in the network will have a consistent and up-to-date view of the network. Based on this view, and the requirements/constraints of a QoS-dependent application, the QoS routing algorithm should find a path through the network that can guarantee the set of constraints. Finding a path based on multiple additive constraints is a NP-complete problem [6][18]. Over the years much research was devoted to finding QoS protocols and algorithms. However, much of this effort was directed to protocols and algorithms on the intradomain level. On this level, it seems possible to guarantee QoS, but on the interdomain level with its many ASs, this will become extremely problematic. In order for end-to-end QoS to become reality, we will need to "QoS-enhance" our routing protocols and algorithms on both intra- and inter-domain level.

· Current Internet routing is hop-by-hop based. In [16] it has been proven that QoS constraints cannot be guaranteed in a hop-by-hop fashion. We need a connection-oriented approach, reserving the required resources along the path.

· In order for guarantees to be made, one must understand and be able to control the physical Internet. In other words, routers and switches must be dimensioned properly. The Internet topology must be stable and robust. If link and nodes frequently go down or some other failures occur, this will impact the level of QoS experienced by many users.

Section 2 has indicated that the choice of paths in Internet is currently very volatile. There is hardly ever a path between a source-destination pair that is used more than 50% of the time over the measured period of three years. This implies that routing behavior, due to its volatile nature, is very hard to predict. If predictions about routing or traffic behavior cannot be made, this impacts the ability of a provider to efficiently deliver its services. Predictability of network behavior is essential for QoS negotiation, QoS admission control and maintaining admitted QoS sessions.

Finally we would like to touch on the most difficult part in QoS provisioning, namely QoS routing on the interdomain level. Currently, BGP allows each AS to independently formulate its routing policies and allows these policies to override distance metrics. These policies can conflict, resulting in route oscillations or may not

be robust with respect to network failures. Routing on the interdomain level, is therefore not a simple shortest path problem, but could be seen as an instance of the stable path problem [8]. To determine whether such an instance is solvable is a NP-complete problem [8]. This clearly demonstrates that maintaining robustness on the interdomain level is more difficult than on the intradomain level. Moreover, the *traceroute* measurements indicated that the influence of BGP is tremendous. QoS path selection based on policy constraints as well as on other constraints is a task of the QoS routing algorithm, which can be vendor/provider specific. However, the stable paths problem illustrates that there is a need for some kind of control in creating policies, such that the stability and robustness of the Internet is maintained. We believe that the current Internet is still prone to too much failures and therefore, in order to attain QoS in a future Internet, much effort has to be put in making the Internet more robust. We conceive that MPLS based on controlled paths is likely the current best strategy towards providing guaranteed e2e QoS.

4. CONCLUSIONS

The main goal of this paper was to provide some more insight into possible QoS provisioning, based on RIPE NCC *traceroute* measurement data. For this purpose the graph G_I has been constructed as the union of most frequently occurring paths in *traceroute* measurements. A strong resemblance of the properties of this graph to those of the uniform recursive tree has been found. We showed that this analogy could enable network operators to compute the multicast efficiency over unicast and to determine eventually the break-even point in m to switch from unicast to multicast. After assigning polynomially distributed link weights to the edges of the graph G1 and comparing the simulated shortest paths with the *traceroutes*, no clear conclusions on the link weight structure of the Internet could be deduced. Finally, we analyzed the *traceroute* measurements from a QoS perspective. Two important properties were motivated: (1) QoS provisioning will operate better as the Internet evolves to a more interconnected graph, leading also to a better modeling/understanding of the Internet graph and (2) the Internet has to be made more robust with respect to decreasing failures of routers and links and with respect to controlling interdomain-level policies.

Acknowledgments

We would like to thank dr. H. Uijterwaal of RIPE NCC for providing us the RIPE data.

REFERENCES

[1] Abramovitz, M. and I.A. Stegun, "Handbook on Mathematical Functions" Dover Publications, Inc., N.Y.,1968.

[2] Bollobas, B., "Random Graphs", Cambrigde University Press, second edition, 2001.

[3] Burch H. and B. Cheswick. “Mapping the Internet”. In IEEE Computer, April 1999.
[4] Chuang J. and M. Sirbu, “Pricing multicast communication: A cost-based approach,” presented at the INET, 1998.
[5] Faloutsos, M. and P. Faloutsos and C. Faloutsos, "On power-law relationships of the Internet Topology", Proceedings of ACM SIGCOMM'99, Cambridge, Massachusetts, pp. 251-262, 1999.
[6] Garey, M.R., and D.S. Johnson, “Computers and Intractability: A guide to the Theory of NP-completeness”, Freeman, San Francisco, 1979.
[7] Govindan R. and H. Tangmunarunkit, “Heuristics for Internet Map Discovery”, In proceedings of IEEE INFOCOM’00, Tel Aviv, Israel, 2000.
[8] Griffin T. G., F. B. Shepherd and G. Wilfong, ”The stable paths problem and Inter domain Routing", IEEE/ACM Transactions on Networking, vol. 10, No. 2, pp. 232-243, April 2002.
[9] Kuipers F.A. and P. Van Mieghem, “QoS routing: Average Complexity and Hopcount in m Dimensions”, Proc. Of 2nd COST 263 International Workshop, QofIS 2001, Coimbra, Portugal, pp. 110-126, September 24-26, 2001.
[10] Labovitz C., A. Ahuja, A. Bose and F. Jahanian, “Delayed Internet Routing Convergence”, IEEE/ACM Transactions on Networking, vol. 9, no. 3, June 2001.
[11] Pansiot J. J. and D. Grad. “On routes and multicast trees in the internet”. In ACM Computer Communication Review, January 1998.
[12] Richard Stevens W., “TCP/IP Illustrated, volume 1, The Protocols”, Addision-Wesley, Reading, Massachusetts, 1994.
[13] RIPE Test Traffic Measurements, http://www.ripe.net/ripencc/mem-services/ttm/
[14] Smythe, R. T. and H. M. Mahmoud, "A Survey of Recursive Trees", Theor. Probability and Math Statist., No. 51, pp. 1-27, 1995.
[15] Van Mieghem P., G. Hooghiemstra and R. W. van der Hofstad, "Scaling Law for the Hopcount", Delft University of Technology, report2000125, 2000.
[16] Van Mieghem P., H. De Neve and F.A. Kuipers, “Hop-by-hop quality of service routing”, Computer Networks, vol. 37, pp. 407-423, 2001.
[17] Van Mieghem, P., G. Hooghiemstra and R. W. van der Hofstad, “On the Efficiency of Multicast”, IEEE/ACM Transactions On Networking, vol. 9, pp. , 2001.
[18] Wang Z. and J. Crowcroft, “QoS Routing for supporting Multimedia Applications”, IEEE J. Select. Areas Commun., 14(7):1188-1234, September 1996.