# A Framework for Overlay QoS Routing

Dragos Ilie and Adrian Popescu
Dept. of Telecommunication Systems
Blekinge Institute of Technology
Karlskrona, Sweden
E-mail:{dragos.ilie, adrian.popescu}@bth.se

***Abstract*— The emergence of new types of popular multimedia services requires in the long run a quality-of-service (QoS) solution better than the best-effort service provided by IP. Failure to widely deploy either one of the main architectures for IP QoS, integrated services (IntServ) or differentiated services (DiffServ), has fueled research into alternate solutions based on overlay networks on top of IP.**

**At Blekinge Institute of Technology (BTH) we are working towards an architecture for multimedia distribution in overlay networks. An important part of this architecture is QoS overlay routing.**

**This paper discusses Overlay Routing Protocol (ORP), a framework for unicast overlay routing, which will be used to test various QoS routing protocols and algorithms.**

## I. Introduction

During the last decade the Internet community has witnessed the proliferation of multimedia services such as voice over IP (VoIP), videoconference, live Internet TV/radio and video on demand (VoD). A common feature for these services is the requirement for network paths that satisfy constraints on specific QoS parameters *e. g.*, available bandwidth, maximum delay and delay jitter. The goal is to ensure that enough resources are available such that the end-user is satisfied with the quality of the received service.

Currently, there are two different QoS architectures for Internet Protocol (IP)-based networks: IntServ [1] and DiffServ [2]. IntServ defines a fine-grained system with *per-flow* management. DiffServ defines a course-grained system, where flows belonging to a specific QoS class are managed as a group. Neither architecture has been widely deployed to date due to reasons amply discussed in [3]–[5]. Some important issues include lack of a viable economical solution for network operators, poor backwards compatibility with existing technology and difficulties in the interaction between different network operators. Several research projects have therefore attempted to provide overlay-based QoS solutions on top of IP's best-effort service [6]–[9]. Current status of QoS routing research and existing challenges are discussed in detail in [10].

At BTH we are working towards an architecture for multimedia distribution in overlay networks. The work includes evaluation and enhancement of various parts required by the targeted architecture. One important part of the architecture is QoS routing in overlay networks.

The authors would like to thank Euro-NGI and the Swedish Internet Infrastructure Foundation (IIS) for supporting and granting the ROVER project during 2006 and 2007.

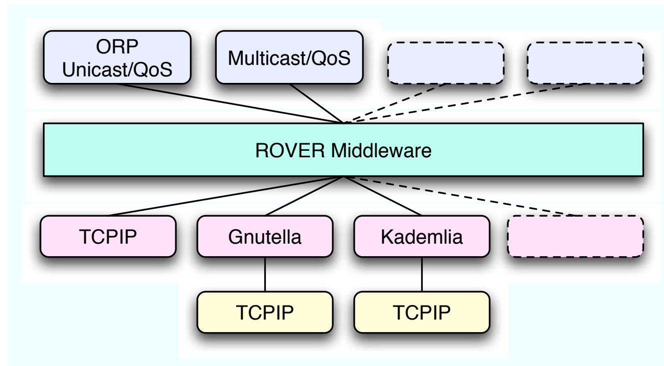


Fig. 1. ROVER architecture

## II. ROVER Architecture

The project Routing in Overlay Networks (ROVER) pursued by BTH aims at developing a platform to facilitate development, testing, evaluation and performance analysis of different solutions for overlay routing, while requiring minimal changes to the applications making use of the platform. The project aims to do this by implementing a middleware system, exposing two set of APIs – one for application writers, and one for interfacing various overlay solutions.

Overlay routing frameworks have been been the subject of much research in recent years. Systems such as Chord [11], *i3* [12], and Kademlia [13] have been proposed and studied from various aspects. The similarities in the functionality of these and other structured overlay routing systems have resulted in a suggestion for a common application programming interface (API) for structured overlays [14]. The ROVER research group uses this API as a starting point for the development of the ROVER middleware.

The common API [14] is designed to abstract structured overlays, *i. e.*, overlays whose topologies follow a specific geometry imposed by the distributed hash table (DHT) they use. These overlays are in contrast with unstructured overlays, in which there is no internal structure, and the system can be viewed as emergent. An important goal of the ROVER middleware is to abstract both structured and unstructured overlays.

The ROVER architecture is shown in Fig. 1. The top layer represents various protocols and applications using the ROVER API. The middle layer is the ROVER middleware with associated API. Finally, the bottom layer represents various transport protocols that can be used by the ROVER middleware. Only the left box, denoted ORP, in the top layer in the figure is within the scope of this paper. ORP is a framework that allows us to study various types of problems and solutions related to unicast QoS routing.

## III. Overlay QoS Routing

ORP is part of a larger goal to research and develop a QoS layer on top of the transport layer. The main idea is to combine an ORP together with additional QoS mechanisms such as resource reservation and admission control into a QoS layer. User applications that use the QoS layer can obtain soft QoS guarantees. These applications will run on a end-hosts without any specific privileges such as the ability to control the internals of TCP/IP stack, the operating system, or other applications that do not use the QoS layer. Nodes in the ORP overlay use the User Datagram Protocol (UDP) to transport application data, similar to the solution reported in [6]. In terms of the OSI protocol stack, the QoS layer is a sub-layer of the application layer. Applications may choose to use it or bypass it.

The QoS layer implements *per-flow* QoS resource management. In contrast to IP routing, we envision that it is mostly end-nodes in access networks that take part in the routing protocol. IP routers are not required to take part or be aware of the QoS routing protocol running on the end-nodes. In other words, we propose a QoS layer on top of the best-effort service provided by IP. Since a best-effort service leaves room for uncertainties regarding the resource allocation, we aim only for soft QoS guarantees.

ORP requires that nodes interested in performing QoS routing form an application-layer overlay. The overlay may be structured (*i.e.*, a DHT) or unstructured. The only requirements for it are the ability to forward messages and to address individual nodes through some form of universally unique identifier (UUID) [15].

The type of services considered for the QoS layer are currently restricted to those that require interactive and non-interactive live unicast multimedia streams. In the future, we will consider other service types as well (*e.g.*, multicast).

By a multimedia stream, we mean a stream containing audio, video, text (*e.g.*, subtitles or Text-TV), control data (*e.g.*, synchronization data), or a combination thereof. If an application chooses to use several media streams (*e.g.*, one stream per media type), the QoS routing protocol treats them independently of each other and assumes that the application is capable on its own of performing synchronization or any other type of stream merging processing.

The multimedia streams within the scope of ORP are of unicast type, *i.e.*, point-to-point (one-to-one). Multicast streams (one-to-many) are subject for later research. Furthermore, the streams we consider are live, which means that the receiver is not willing to wait until the whole stream data is received, but would rather start watching and listening to it as soon as enough data is available for rendering.

By interactive multimedia streams, we mean streams generated by user interaction as in a video conference or a VoIP call. Conversely, non-interactive multimedia streams do not involve any interaction between users as is the case of Internet TV or music streaming.

Applications on top of the QoS layer request overlay paths to certain destinations, along with specific constraints attached to each path (*e.g.*, minimum bandwidth required, maximum delay and delay jitter tolerated). We expect the source nodes to compute the feasible path for each flow that originates from them. The path information is later communicated to the nodes on the corresponding path as part of the ORP operation. Essentially, all source nodes compete among each other for overlay resources.

We assume that each node is capable of estimating its available host resources (*e.g.*, RAM, storage) as well as link properties (*e.g.*, residual bandwidth, round-trip time (RTT)) to its one-hop neighbors in the overlay. Nodes are expected to exchange this information using some form of link-state routing protocol implemented by ORP.

Furthermore, we assume that the QoS layer cannot interfere with the general resource usage (either in terms of host or network resources) other than those used by the QoS layer itself. In other words, the QoS routing protocol cannot perform resource reservation other than on residual resources (*i.e.*, resources not used by other applications running simultaneously on the node). Obviously, if *all* applications on a node run on top of the QoS layer, then true resource reservation can be performed for the host resources. Network resources will however always be fluctuating due to traffic streams outside the control of the QoS layer.

Some resource fluctuations may drive the node into resource starvation. During resource starvation the node is unable to honor some or all of the QoS guarantees. This type of events may lead to degradation in the quality of rendered media (*e.g.*, through MPEG frames that are lost, garbled, or arrive too late).

Applications on top of the QoS layer may be able to tolerate quality degradation for very brief periods of time or even recover from brief degradation by using forward error correction (FEC) codes or retransmissions.

However, prolonged quality degradation may eventually lead to user dissatisfaction with the quality of the service. Each node must therefore carefully monitor the link properties to each of its immediate neighbors. If resource starvation is detected (or anticipated) then a new feasible path must be found and traffic re-routed on it. This mechanism must be robust enough to avoid route flapping.

## IV. QoS Path Selection Algorithms

The path selection algorithms are an essential part of QoS routing. In general, the path selection problem is posed in the form of an optimization problem. The network is represented by a directed graph $G = (\mathbb{V}, \mathbb{E})$ where $\mathbb{V}$ is a set of $V$ nodes (vertices) and $\mathbb{E}$ is a set of $E$ directed links (edges).

Each link has a number of *additive* QoS metrics (*e. g.*, delay and delay jitter) as well as *non-additive* QoS metrics (*e. g.*, bandwidth). Problems involving constraints on non-additive metrics can be resolved by pruning the links of the graph that do not satisfy the constraints [16]. Additive metrics are more difficult to handle. For $i = 1, \ldots, m$ we denote by $w_i(u, v)$ the $i$-th additive metric for the link $(u, v)$ between nodes $u$ and $v$ such that $(u, v) \in \mathbb{E}$. Given $m$ additive constraints $L_i$ for the requested path, the multi-constrained path (MCP) optimization problem is finding a path that satisfies

$$w_i(P) \triangleq \sum_{(u,v)\in P} w_i(u, v) \leq L_i \quad (1)$$

for $i = 1 \ldots m$.

In some cases there is also an objective function $\mathcal{F}$ that needs to be minimized or maximized, *e. g.*, a global cost function. In these cases we have what is called the multi-constrained optimal path (MCOP) problem [16].

Our focus is on path selection algorithms that can work in the scenario described in Section III. In particular we are interested in algorithms that can handle the following issues:

- *multiple constraints:* most often we will be looking for paths that satisfy two or more constraints. We consider in general problems with linear constraints. In some problems there is an additional objective function that needs to be minimized or maximized. We do not consider paths with several objective functions, *e. g.*, simultaneous minimal delay, maximum bandwidth.

- *dynamic environments:* the algorithms must be able to cope with changes in the environment, *e. g.*, resource fluctuation, churn. If the changes are small, then a local search may prove to be less computationally expensive than if the optimization algorithm is run over the entire search space. Some algorithms can do this automatically, through time-dependent objective functions, others can be combined with local search methods and the remaining algorithms perform a full search every time the environment changes [17]. The computational overhead of running a full search compared to the other two algorithms types is highly dependent on the optimization problem and algorithms in question.

- *"realtime" performance demand:* the algorithm must be able to compute the feasible routes for each link-state update. This may be particularly difficult to achieve for optimization algorithms that perform a full search whenever the environment changes.

Below is a short presentation of the methods that we consider for path selection.

### A. SAMCRA

Self-Adaptive Multiple Constraints Routing Algorithm (SAMCRA) is based on the following four key concepts [18, 19]:

- a nonlinear definition of the path length improves the ability to satisfy all $m$ constraints.
- a $k$-shortest path approach, which first shortest, the second shortest, *etc*, up the $k$-shortest path.
- non-dominated paths, which is a technique that can dramatically increase the computational efficiency.
- look-ahead that reduces the search-space of possible paths.

### B. The Simplex Method

The simplex method is one of the most popular methods of mathematical programming for linear optimization problems with linear constraints [20, 21]. The method can be made more efficient by use of LU-decomposition [20].

LU-decomposition can also be used to efficiently solve a system of linear equations [22, 23]. This means that a well designed simplex method can be used for both MCP and MCOP problems.

### C. Gradient Projection Method

This method is related to the steepest descent method used for unconstrained optimization problems. The gradient projection method can be applied to general nonlinear constrained optimization problems [20]. Typically, the method converges rapidly to a neighborhood of the optimal solution. However, convergence speed to optimal solution inside the neighborhood depends on the problem parameters [24].

### D. Conjugate Gradient Method

The conjugate gradient method is one of the best general purpose methods for unconstrained optimization problems [20].

Our reason to consider this method is two-fold. First, this method can be used to solve a system of linear equations as shown in [23]. Second, many constrained optimization problems can be converted to unconstrained form through Lagrangian relaxation or through the introduction of penalty or barrier functions [17, 20, 25]. At that point, it may be possible to use the conjugate gradient method to solve the unconstrained problem.

### E. Particle Swarm Optimization

Particle Swarm Optimization (PSO) falls into the category of computational swarm intelligence [17]. It is based on emergent behavior of neighboring individuals. In PSO, each particle represents a potential solution to the problem. By altering the position and speed of each particle according to specific rules, the swarm searches the multidimensional space for the optimal solution [17].

The classical PSO can be applied to unconstrained optimization problems. However, there are several mechanisms that can be applied to PSO to allow it to solve constrained optimization problems [17].

## V. Future Work

We will evaluate diverse ORP solutions by using both simulation and live testing. The simulations will allow us to fine-tune the routing protocol and to test the performance of the routing algorithms. The next step will be to run several experiments on PlanetLab [26]. We are particularly interested in experiments to evaluate the protocol overhead, the success rate of the algorithms and the scalability of the entire solution.

We will also investigate solutions that allow ORP to use hierarchical routing for networks with a large number of hosts.

## References


[1] R. Braden, D. D. Clark, and S. Shenker, *RFC1633: Integrated Services in the Internet Architecture: an Overview*, IETF, June 1994, category: Informational.

[2] S. Blake, D. L. Black, M. A. Carlson, E. Davies, Z. Wang, and W. Weiss, *RFC2475: An Architecture for Differentiated Services*, IETF, Dec. 1998, category: Informational.

[3] G. J. Armitage, "Revisiting IP QOS," *ACM SIGCOMM Computer Communications Review*, vol. 33, no. 5, pp. 81–88, Oct. 2003.

[4] G. Bell, "Failure to thrive: QoS and the culture of operational networking," in *Proceedings ot the ACM SIGCOMM Workshops*, Karlsruhe, Germany, Aug. 2003, pp. 115–120.

[5] L. Burgsthaler, K. Dolzer, C. Hauser, J. Jähnert, S. Junghans, C. Macián, and W. Payer, "Beyond technology: The missing pieces for QoS success," in *Proceedings ot the ACM SIGCOMM Workshops*, Karlsruhe, Germany, Aug. 2003, pp. 121–130.

[6] D. G. Andersen, "Resilient overlay networks," Master's thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 2001.

[7] L. Subramanian, I. Stoica, H. Balakrishnan, and R. Katz, "OverQoS: An overlay based architecture for enhancing internet QoS," in *Proceedings of NSDI*, San Francisco, CA, USA, Mar. 2004.

[8] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 29–40, Jan. 2004.

[9] M. Castro, M. B. Jones, A.-M. Kermarrec, M. Theimer, H. Wang, and A. Wolman, "An evaluation of scalable application-level multicast built using peer-to-peer overlays," in *Proceedings of INFOCOM*, San Francisco, CA, USA, Apr. 2003.

[10] X. Masip-Bruin, M. Yannuzzi, J. Domingo-Pascual, A. Fonte, M. Curado, E. Monteiro, F. Kuipers, P. Van Mieghem, S. Avallone, G. Ventre, Aranda-Gutiérrez, M. Hollick, R. Steinmentz, L. Iannone, and K. Salamatian, "Research challenges in QoS routing," *Journal of Computer Communications*, vol. 29, no. 5, pp. 563–581, Mar. 2006.

[11] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of ACM ISGCOMM*, San Diego, CA, USA, Aug. 2001, pp. 149–160.

[12] D. Stoica, Ion an Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," in *Proceedings of ACM SIGCOMM*, Pittsburgh, PA, USA, Aug. 2002, pp. 73–88.

[13] P. Maymounkov and D. Maziéres, "Kademlia: A peer-to-peer information system based on the xor metric," in *Proceedings of IPTPS*, Berkeley, CA, USA, Mar. 2002.

[14] F. Dabek, B. Zhao, P. Druschel, J. Kubiatowicz, and I. Stoica, "Towards a common API for structured peer-to-peer overlays," in *Proceedings of IPTPS*, Berkeley, CA, USA, Feb. 2003.

[15] P. Leach, M. Mealling, and R. Salz, *RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace*, July 2005, category: Standards Track.

[16] F. Kuipers, P. Van Mieghem, T. Korkmaz, and M. Krunz, "An overview of constraint-based path selection algorithms for QoS routing," *IEEE Communications Magazine*, vol. 40, no. 2, pp. 50–55, Dec. 2002.

[17] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. Chichester, West Sussex, England: John Wiley & Sons, 2006, ISBN: 0-470-09191-6.

[18] P. Van Mieghem and A. Kuipers, Fernando, "Concepts of exact QoS routing algorithms," *IEEE/ACM Transactions on Networking*, vol. 12, no. 5, pp. 851–864, Oct. 2004.

[19] F. A. Kuipers, T. Korkmaz, M. Krunz, and P. Van Mieghem, "Performance evaluation of constrained-based path selection algorithms," *IEEE Network*, vol. 18, no. 5, pp. 16–23, Sept. 2004.

[20] D. G. Luenberger, *Linear and Nonlinear Programming*. Norwell, MA, USA: Kluwer Academic Publishers, 2004, ISBN: 1-4020-7593-6.

[21] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Mineola, NY, USA: Dover Publications, 1998, ISBN: 0-486-40258-4.

[22] W. H. Press, S. A. Teukolsky, W. T. Vettering, and B. P. Flannery, *Numerical Recipes in C++: The Art of Scientific Computing*, 2nd ed. Cambridge, UK: Cambridge University Press, 2002, ISBN: 0-521-75033-4.

[23] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: The Johns Hopkins University Press, 1996, ISBN: 0-8018-5414-8.

[24] D. P. Bertsekas and R. G. Gallager, *Data Networks*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1991, ISBN: 0132009161.

[25] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theories, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice Hall, 1993, ISBN: 0-13-617549-X.

[26] PlanetLab, "Planetlab," http://www.planet-lab.org, January 2007.