QoS in mobile multimedia networks

Wolfgang Schönfeld,Ralf Steinmetz,Nicole Berier

GMD IPSI
Dolivostr. 15 • D-64293 Darmstadt • Germany
{wolfgang.schoenfeld,ralf.steinmetz,nicole.berier}@darmstadt.gmd.de

Lars Wolf

University of Karlsruhe • Computing Center
Zirkel 2 • D-76128 Karlsruhe• Germany
lars.wolf@rz.uni-karlsruhe.de

Abstract: Mobility highly affects quality of service in multimedia networks. This applies both to link layer mobility (radio link quality) and to network layer mobility (hand-over, roaming). We develop an abstract framework for different architectures of network layer mobility support like Mobile IP, Cellular IP, IPv6, GSM, UMTS etc. and discuss their appropriateness for QoS handling.

1 Introduction

Current Internet routers offer best-effort routing service, i.e. they forward data packets very fast, but they do not guarantee any service quality. Multimedia applications, on the other hand, require stable or at least predictable QoS.

In classical stationary networks, the basic impediment to QoS handling consists of the competing service requests of other nodes, resulting in buffer space exhaustion, congestion etc. With the advent of mobile computers, the situation changes: Moves of hosts will continuously change the transmission conditions of the links, up to complete losses and to sudden recoveries. Hence, routing in mobile networks has to cope not only with varying traffic load, but also with varying transmission conditions and frequent topology changes.

There are well-known solutions to handle these problems caused by mobility. Boot protocols like Dynamic Host Configuration (DHCP) support automatic configuration of links. In this case however, the mobile host is assigned another IP address which fits into the new LAN. On the other hand, a mobile host using Mobile IP may keep one and the same IP address wherever it attaches. A consequence is that TCP connections can persist even during a move. Similarly, mobile devices in a cellular network keep their phone number and maintain connections when handed over from cell to cell (even to one in another domain).

These approaches have to converge when future UMTS/3GPP networks will „bring IP to the base stations“. Existing proposals are Cellular IP [1], Hierarchical IP [3], Mobile IP Regional Registration [17], etc.

Within these approaches, QoS considerations currently do not play a central role. We at GMD IPSI and Karlsruhe University are filling this gap, see [5], [6], and [8]. In this talk, we investigate mobile QoS by developing a general model of routing and network layer mobility. We describe certain (special features of) mobility management architectures and their appropriateness for handling basic functions of QoS management.

2 Routing

Let the *topology* of a mobile network be given by a directed finite graph $T=(N,L)$ where the elements of N represent communicating entities. $L \subseteq N \times N$ is the set of communication links, i.e. $(X,Y) \in L$ means that Y can (directly) receive data sent by X. Because of the characteristics of radio communication, T may be non-symmetric, non-transitive, and even non-reflexive.

We assume *next-hop* routing, i.e. each node X maintains a table $next_X \subset (N \times N)$, its *routing table*. Here, $(Y,Z) \in next_X$ means that X sends („routes“) to Z

packets addressed to *Y*. For any node *Y*, let $route_Y = \{(Y,Z) \in next_X \mid X, Z \in N\}$ be the *sink graph* to *Y*. It is *correct* if it is a strict partial order contained[1] in *L* in which Y is a maximal („last") element and which contains, for each other „correspondent" node *C*, at least one path from *C* to *Y*. Routing in a network is *correct* if all nodes have correct sink graphs.

For the sake of presentation, we restrict mobility in the following way: A node without routing functionality (i.e. it is minimal in the sink graph for any node) is called an *end node*. A *base station* is a router loosing and gaining links over time. A *mobile node* is an end node with the same property. Hence, topology dynamics is restricted base stations and mobile nodes.

We consider both *multi-path* and *multi-cast* routing: Any router *X* receiving a packet addressed to a set *N* forwards it (after removing its own address, if necessary) to all neighbours Z (to which *X* is linked, obviously, and) for which there is some $Y \in N$ such that $(Y,Z) \in next_X$. In case of uni-path and -cast routing, the sink graph degenerates to the well-known *sink tree*.

Multi-path routing can serve different purposes. First, there may be the need to spread network load across a broader collection of network links, see [1]. Second, multiple paths may be used to set up a distribution network of *mixes* for security reasons, see [2]. And third, multi-path routing may help to solve the inherent problem of location uncertainty, see e.g. [16]. Fast hand-over in cellular networks applies a special case of multi-path routing: Packets addressed to the mobile node arrive, for a certain time interval, both at the current and at the next base station. In an ideal network with unbounded resources where node mobility and packet arrival/ delivery were exactly predictable, multi-path routing would not be needed.

1. Considering links as pairs of nodes excludes multiple links, but this does not cause any loss of generality and makes our framework easier to handle. However, it may be necessary in practice, e.g. when the mobile node uses more than one channel to communicate with the base station and only some of these channels vanish, say because of interference.

3 Routing Dynamics

The topology of a mobile network is dynamic, i.e. links vanish or arise over time, caused by node mobility. We make no assumptions on how topology changes are detected. Base stations and/or mobile nodes may periodically send messages by which a partner node can derive whether a link persists. Such messages may contain further information like service advertisements etc. We also leave open which node, when detecting a change, triggers the network to react appropriately.

Topology changes need not immediately affect correctness of routing tables, in particular if multi-path routing is applied. A router may decide to keep a route in its table even when the underlying link is lost. Likewise, routers need not immediately use a newly arosen link. We leave out here any discussions on the principles to be applied and consider only the basic functions of adding and removing routes, no matter in which way they are applied.

In any case, to use a new link means adding new paths via this link. For a new link from a base station to the mobile node, new paths have to be added to the mobile node's sink graph. For a new link in the reverse direction, paths have to be added to the sink graphs of any other node. However, we need not worry about the latter case since base stations do not change their links to other routers - paths from them to other nodes are already known.[2]

Suppose now that a new link from base station *B* to mobile node *Y* has **arisen**. Let $S \subseteq N$ be the set of nodes from which new paths to *Y* via *B* are to be added to *Y*'s sink graph. Methods to incorporate new routes into a network are well known. In the context of mobility, we propose to use B's sink graph to extend that of Y in the obvious way.

The converse case, i.e. that an existing link from base station *B* to mobile node *Y* is **lost**, is more difficult to handle. A quite common solution is to assign to any routing table entry a pre-determined life time („timeout"). This means that entries vanish independently of whether the corresponding links exist and that they have to be refreshed periodically, with a time interval fitting to each entry's timeout.

The other solution is to explicitly delete entries from routing tables. If, by this, a link used in a cer-

2. We do not consider adding new paths to the stationary part of the network (e.g. to spread network load).

tain sink graph vanishes, one has to decide whether routing should instantly be repaired or be left in a temporarily inconsistent state with the hope that the link will soon re-appear. Detailed answers to these problems will be given elsewhere.

The reader will recognize similarities to the garbage collection problem in run-time environments of some programming languages. Let us only mention here that, in the case of a tree-like topology, it makes sense to combine deletion of old entries with creation of new ones. This is the way how mobility is handled in standard cellular networks.

4 Tunnels

To handle Internet mobility solely by routing table updates as described in Chapter 3 is obviously not feasible. One reason is that IP routing havily relies on address masking by which large sets of routing table entries can be contracted to one single entry. When a node moves, its new address may fall out of a mask, and masking would have to be reworked. It is a common understanding that Internet routing must be protected from this instability (see [7]) caused by mobility.

On the other hand, changing routing tables is appropriate (and already in use) for small (parts of) networks, in particular if they have a tree-like topology. The key idea of bringing mobility also to large networks is to restrict mobility support to a reasonably small subset of routers.

Many proposals apply this principle. To elaborate a common view, let us go back to the basic mobility problem. When a data packet following a path in the sink graph arrives at a missing link, it has to be dropped. Or stated otherwise, we have an inconsistency between packet address and routing table. Chapter 3 describes how to remove this inconsistency by changing the routing tables, but we may also change the packet address.[3] Before the packet can arrive at the receiver, we have to restore the original address.

The standard way of temporarily hiding the receiver address by another address is *encapsulation* (see [9]). Since en-(or de-)capsulation of a data packet in general means to enter (or leave) a lower layer (in the sense of OSI), we extend our general model of Chapter 2 in the following way.

The higher layer topology $\underline{T} = (\underline{N},\underline{L})$ is derived from the given topology $T = (N,L)$ in the following way: Select a subset $M \subseteq N$ of nodes and take a copy $\underline{M}$ of M, i.e. to each $X \in M$, there corresponds exactly one $\underline{X} \in \underline{M}$. Encapsulation means that there is a link from $\underline{X}$ to X, and decapsulation is just the reverse.[4] Links within the upper layer (i.e. the set $\underline{L}$) are inherited from those in L: Define $\underline{L}$ to be the set of pairs $(\underline{X},\underline{Y})$ such that there is a path from X to Y within the lower layer T. We call such a link *virtual* and the corresponding path a *tunnel* realizing the virtual link. See picture below. Note that, in general, a virtual link is realized by more than one tunnel.

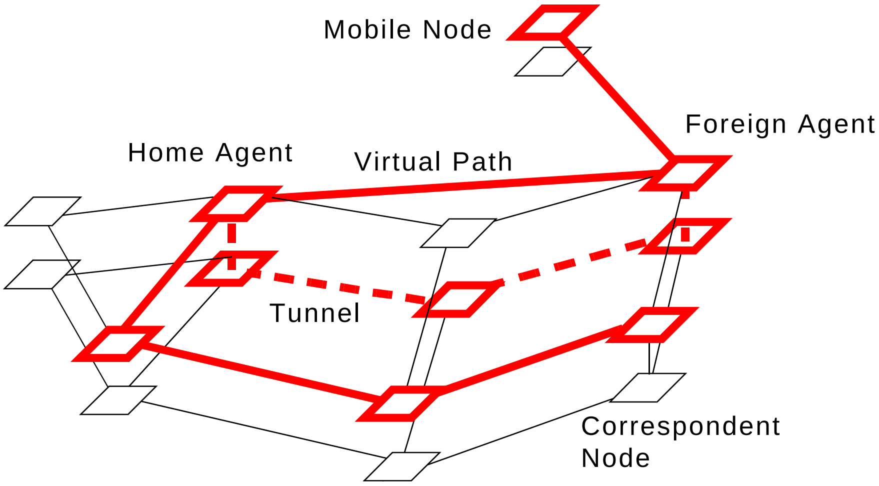
We restrict dynamics to the upper („mobility“) layer. Its routing derives from that of the lower („stationary“) layer. In addition, two routers performing en- and decapsulation X and Z („mobility agents“) may *register* a virtual link leading to a mobile node Y by extending their routing tables in such a way that $(Y,\underline{X}) \in next_X$ and $(Y,Z) \in next_{\underline{Z}}$. All the other notions of Chapter 2 and Chapter 3 directly apply.

This general model admits a remarkably high degree of freedom for the development of new mobility architectures. E.g. one may consider to have more than one virtual link to a mobile node be registered at a mobility agent, even in parallel to a „stationary“ link - resulting in a „multi-virtual-path“ routing to possible locations of a mobile node (see last paragraph in Chapter 2 for this motivation).

Mobile IP (see [10] for the current version) is the most well-known existing architecture applying virtual paths. When considering the case of a foreign agent care-of address (and reverse tunneling), we have the subsequent picture.

3. Note that, this way, the inconsistency is only temporarily (just for one packet) removed whereas routing update is a persistent (as long as the node does not move again) correction.

4. Readers with some background in graph theory will recognize this as „unfolding the loop-back“.



Virtual is the link between home and foreign agent. We get the case of a co-located care-of address by making the (former) foreign agent a normal router of the lower layer and adding a (non-virtual!) link to that layer leading from this router to the mobile node. As a consequence, the mobile node belongs to both layers.

A variation of this principle is to let the correspondent node be the starting point of the tunnel (see [11]) in order to avoid triangular routing (as in the above picture). Other mobility architectures like Cellular IP [1], Hierarchical IP [3], or Mobile IP Regional Registration [4] attach to the foreign agent a tree-like sub-network, thus combining the principles of routing updates and tunneling.

Any other combination is also feasible, e.g. spreading mobility agents all over the Internet. There are good reasons to expect from this a reduction of the control traffic necessary to handle mobility (agent advertisment, registration, etc.). Tree-like topologies for these architectures are discussed, but there are no discussions of non-tree topologies.

We do not exclude that an additional mobility layer is placed on top. This will be useful when not only single nodes, but whole pieces of the topology collectively move, as will be the case in large ship, trains, cars etc.

5 Quality of Service

To incorporate QoS into our model is now straighforward: We refine simple best-effort routing. Forwarding of a packet on a given link now depends on a *valuation* of that link (and, maybe, of other links). I.e. we have a function *b* which specifies for each link *l* the resources (bandwidth etc.) avaible for this link. Clearly, *b* will vary over time. Note that in best-effort routing, *b* degenerates to a Boolean function which assigns on of the values „exists“ and „does not exist“ to any link.

Router *X* bases its decisions not only on its routing table $next_X$, but also on *b*. It is out of the scope of this paper to discuss particular decision functions. See e.g. [6] or [8] for more details. We simply assume that *X* can compute a routing function r_X which, based on valuation *b* and routing table $next_X$, associates to each node Y a set of neighbours to which the packet in question has to be forwarded. Remember that (see Chapter 2) any packet is routed to all neighbours indicated by the routing table to which a link exists. Now, routing function r_X makes a selection among these neighbours according to valuation *b*.

Let us now approach the problem addressed in the title of this talk, i.e. how we can handle QoS in mobile multimedia networks. The key to a solution is given in Chapter 4 where we introduced a layered view of mobility. Obviously, both the underlying stationary layer and the mobility layer in a network can be equipped with QoS handling. Since QoS parameters of a virtual link derive from those of the corresponding tunnel, QoS guarantees within the upper layer clearly depend on those of the lower layer. But, to a certain extend, we can also influence from the upper layer the load in the lower layer. Let us consider an example.

In the COSMOS project (see [5]), mobile nodes within a construction side connect to those at the company’s headquarter via, among others, satellite or ISDN link. Both are expensive and, hence, should be set up only temporarily. By placing mobility agents on both sides of these links, we not only can handle mobility, but also can distribute load over these (and possibly other) links.

To handle the general case, consider a link from a node $\underline{X}$ to a node $\underline{Y}$ in the mobility layer. Seen from the underlying layer, it is a virtual link. Let S be the sub-topology of T defined by the set of tunnels realizing this link (see Chapter 4). We call it the *sub-topology* realizing the given link.

For an application of this construction, assume that, within the mobility layer, we can choose among two different links to set up a route between one and the same endpoints. In the underlying layer, this means that we choose among two different (maybe, disjoint) sub-networks.

6 Conclusions

In this talk, we developed a general model in which the appropriateness of mobility support architectures can be discussed. In particular, we showed how QoS handling within mobile network and within the underlying network depend on each other.

One result which can be derived is that, the smaller the tunnels realizing the links of the mobility layer are, the finer can QoS be handled. Or stated otherwise, architectures for mobility support which use single end-to-end tunnels (like Mobile IP) can handle QoS only by relying to that of the underlying stationary network.

Clearly, the model developed in this paper is too general to be implemented, but it may help to develop partial solutions which best fit to particular situations.

References

The Internet Documents mentioned here are available at http://ietf.org/.

[1] A. Campbell e.a., „Cellular IP“, draft-ietf-mobileip-cellularip-00.txt (2000).

[2] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", *Communications of the ACM 24/2* (1981), 84-88

[3] C. Castellucia, „Hierarchical Mobile IPv6 Proposal“, *Institut National de Recherche en Informatique et en Automatique*, Report 226 (1998).

[4] E. Gustafsson, A. Jonsson, C. E. Perkins, „Mobile IP Regional Registration“, *Internet Draft* draft-ietf-mobileip-reg-tunnel-02.txt (2000).

[5] H. Huang, A. Meissner, W. Schönfeld, R. Steinmetz, „QoS Policy Framework and its Implementation“, to appear in Proc. ICCT2000, Beijing (2000).

[6] M. Karsten, N. Berier, L. Wolf; R. Steinmetz, „A Policy-Based Service Specification for Resource Reservation in Advance“, *Proceedings of the International Conference on Computer Communications*, (1999.

[7] C. Labovitz, G. Robert Malan and F. Jahanian, „Internet routing instability“, *IEEE/ACM Transactions on Networking* 6 (1998), 515-528.

[8] A. Pajares, N. Berier, L. Wolf, R. Steinmetz, „An Approach to Support Mobile QoS in an Integrated Services Packet Network“, *First Workshop on IP Quality of Service for Wireless and Mobile Networks, Aachen, Germany* (1999).

[9] C. E. Perkins, „Minimal Encapsulation within IP“, *Internet Proposed Standard* RFC 2004(1996).

[10] C. E. Perkins, „IP Mobility Support for IPv4, revised“, *Internet Draft* draft-ietf-mobileip-rfc2002-bis-01.txt (2000).

[11] C. E. Perkins, D. B. Johnson, „Route Optimization in Mobile IP“, *Internet Draft* draft-ietf-mobileip-optim-09.txt (2000).

[12] Andrew S. Tanenbaum, „Computer Networks“, Prentice-Hall (1996).

[13] W. T. Teo and Y. Li, „Mobile IP extensions for Private Internets Support (MPN)“, *Internet Draft* draft-teoyli-mobileip-mvpn-02.txt (1999).

[14] W. Willinger and V. Paxson, „Where Mathematics meets the Internet“, *Notices of*

the American Mathematical Society, 45 (1998), 961-970.

[15] Hans Erikson. „MBONE: The Multicast Backbone“, *Communications of the ACM* 37 (1994), 54-60.

[16] A. O’Neill, G. Tsirtsis, and S. Corson, „Edge Mobility Architecture“, draft-oneill-ema-01.txt (2000).

[17] E. Gustafsson, A. Jonsson, C. E. Perkins, „Mobile IP Regional Registration“, draft-ietf-mobileip-reg-tunnel-02.txt

[18] Ellen W. Zegura, Kenneth L. Calvert, and Michael J. Donahoo, „A quantitative comparison of graph-based models for internet topology,“ *IEEE/ACM Transactions on Networking* 5(1997), 770-783.