

Internet Engineering Task Force
Internet-Draft
Expires: November, 2002

J. Manner
T. Suihko
M. Kojo
M. Liljeberg
K. Raatikainen
May 14, 2002

Localized RSVP
<[draft-manner-lrsvp-00.txt](#)>

Status of this Memo

This document is a submission to the NSIS Working Group of the Internet Engineering Task Force (IETF). Comments should be submitted to the nsis@ietf.org mailing list.

Distribution of this memo is unlimited.

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#). Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire in November, 2002.

Copyright Notice

Copyright (C) The Internet Society (2000). All Rights Reserved.

Abstract

Guaranteed QoS for multimedia applications is based on reserved resources in each node on the end-to-end path. For stationary nodes this may be achieved but not so easily for mobile nodes. Many multimedia applications become less useful if the continuity of QoS is disturbed due to end-to-end signalling or slow re-reservations of resources after each handover. Additionally, if the correspondent node does not support QoS, it would be useful to be able to reserve

at least local resources, especially wireless link resources. This draft proposes small modifications to RSVP, so that initial resource

reservations and re-reservations due to host mobility can be done locally in an access network.

Table of Contents

1	Introduction	2
1.1	Related work	3
2	Local QoS Support	4
2.1	Upstream transfers	4
2.2	Downstream transfers	5
2.3	Additional functionality	5
3	Fast local repair	7
4	Issues with the localized signalling	8
5	Signalling diagrams	10
6	Security Consideration	10
7	Summary and Future Work	11
8	References	12
9	Author's Addresses	13

[1](#). Introduction

Future mobile networks will be based on IP technology. This implies that, on the network layer, all traffic, both traditional data and streamed data like audio or video, is transmitted as packets without knowledge of any higher layer data flows. At the same time different multimedia applications are becoming increasingly popular. These applications require better than best-effort service from the connecting network. They require strict Quality of Service (QoS) with guaranteed minimum bandwidth and maximum delay. Other applications, such as electronic commerce, network control and management, and telnet-type applications, would also benefit from a differentiated treatment.

The Internet Engineering Task Force (IETF) has proposed two main models for providing differentiated treatment of packets in routers. The Integrated Services (IntServ) model [\[1\]](#) together with the Resource Reservation Protocol (RSVP) [\[2\]](#) [\[3\]](#) provides per-flow guaranteed end-to-end transmission service. The Differentiated Services (DiffServ) framework [\[4\]](#) provides non-signaled flow differentiation that usually provides but does not guarantee proper transmission service. These architectures, however, have problems, for example, RSVP requires support from both communication end points and from the intermediate nodes. DiffServ, on the other hand, requires support from the sender of a stream and from the network nodes. The Internet Architecture Board has outlined additional issues related to these two architectures [\[5\]](#).

Let's consider a scenario, where a fixed network correspondent node (CN) would be sending a data stream to a mobile node behind a wireless link. If the correspondent node does not support RSVP it cannot signal its traffic characteristics to the network and request

specific forwarding services. Likewise, if the correspondent node is not able to mark its traffic with a DiffServ Code Point (DSCP) to trigger service differentiation, the multimedia stream will get only best-effort service which may result in poor visual and audio quality in the receiving application. Even if the connecting wired network is over-provisioned, reserving local resources is especially important in wireless access networks, where the bottleneck resource is most probably the wireless link. For illustration purposes, we will only consider a scenario with a mobile access network that is aware of the proposed signaling mechanism and has mobile nodes as its clients.

In the absence of end-to-end QoS support, a mobile node could still want to reserve resources from the access network for both outgoing and incoming traffic. We propose a signaling mechanism based on a slightly modified RSVP. Access network specific reservations would be distinguished from the end-to-end reservations. The mobile node does not need to know the access network topology or the nodes that will reserve the local resources. The reservation message itself identifies the intention and the access network will find the correct network node(s) to respond to the reservation. However, the scheme is not tied to only mobile networks but can be used in any network that needs flexible local resource allocations.

This would be very beneficial, even if the QoS support is only available in the access network. Furthermore, the same solution saves us from end-to-end signaling even if all intermediate nodes on the transmission path would support standard RSVP. This would allow, for example, the mobile node to reserve wireless resources separately from end-to-end resources.

1.1. Related work

Currently we can identify two ways to signal QoS requirements to an access network: DiffServ Code Points (DSCP) and RSVP with IntServ. In the DiffServ-based solution the mobile node can mark the upstream packets with proper DSCP values, but for the downstream the gateway on the edge of the access network must be able to identify and handle the preferred streams. This can be accomplished with default values for different micro flows in the Service Level Agreement negotiated between the client and the access network service provider.

A second way to make use of DiffServ could be through a Bandwidth Broker [6] [7] [8] that dynamically returns the proper code point for each flow: when the first packet of a flow arrives at the access network edge router, it requests the proper code point from the Bandwidth Broker. The router maintains a mapping from micro flow identification to the DiffServ code point (soft state) so that future

packets can be directly identified and labeled. Still, this would require a protocol that the mobile node could use to dynamically adjust the mapping information stored at the access network edge for the incoming traffic.

RSVP can provide the signalling mechanism for QoS requirements to the access network. For upstream reservations, the mobile node would send the PATH message to the access network edge router, which would return the RESV message and set up the reservations. The edge router would act as an RSVP proxy [9]. However, the reservation in the downlink direction is not as straightforward since the downlink reservation needs to be initiated by the RSVP proxy. We would need a way to trigger the proxy to initiate the RSVP signalling for a downlink flow.

Thus, these mechanisms do not seem to solve the problem entirely. The DiffServ mechanisms cannot provide explicit resource reservations. The problem with the RSVP proxy approach is that the proxy cannot automatically distinguish reservations that would be answered by the correspondent node and reservations that would require interception. Additionally, the RSVP proxy needs a way to know when to allocate resources for incoming flows. Mobile access networks also add to the problems, since mobile nodes can frequently change their point of attachment in the network and resource allocations need to be re-arranged, including changing the serving RSVP proxy.

2. Local QoS Support

Our solution to the problem of localized QoS co-ordination is based on proxies and the RSVP local repair mechanism [2]. The proxy is running on an RSVP node and is called the Localized RSVP Proxy server (LRSVP proxy). In addition, we need a way to differentiate reservations that are internal to the access network. We suggest using one bit of the eight flag bits in the RSVP Session object for this purpose. We use the bit 0x8 and call the flag a Local Indication (LI). We also add a new message type called "Path Request" message with an initial message type number 8 and a message called "Path Request Tear" with an initial message type number 9. The first sketch of this solution appeared in [10] and [11], although some implementation ideas have changed since.

When a mobile node wants to reserve resources in the local network, it uses the LI flag in RSVP messages to indicate a local reservation. The structure of the RSVP messages follow the RSVP standard. The LRSVP proxy that receives an RSVP message with the LI bit set will notice that the flag was set, does not forward the message further to the next hop and responds according to the following description.

2.1. Upstream transfers

Setting upstream reservations is straightforward and follows the standard RSVP functionality. The mobile node sends the usual Path

message, but sets the LI flag. The Session Object defines the destination of the flow that will eventually be transmitted from the mobile, and the Sender Template provides information about the mobile itself.

When the LRSVP proxy receives the Path message, it notes due to the LI bit that the reservation is meant to stay within the access network and responds with a Resv message. It will not forward the Path message further to the next hop. If a Resv message arrives at the mobile, the resources for the session defined in the Path message have been allocated.

2.2. Downstream transfers

Before setting a downstream resource reservation, the mobile needs to be aware of the data senders. For multimedia communications, a session is usually initiated up with application layer protocols like SIP or RTSP [[12](#)]. From that correspondence, the mobile knows the necessary information about the sender. Another, more coarse reservation could be set, for example, for browsing different audio servers; the mobile node would indicate in the RSVP messages that the sender will use UDP. It is thus possible to make resource reservations for any sender that wants to communicate with the mobile. However, in order to allow for more accurate QoS support, more information should be given.

In order to set up the downstream reservation we need a way to signal the LRSVP proxy to initiate the RSVP reservation setup on behalf of the sender(s), that is, to send a Path message. To achieve this, the mobile node sends the Path Request message with the LI flag set. The Path Request message is identical to a standard Path message apart from the message type field. The Session Object must include information about the recipient, the mobile in this case, and the Sender Template must define expected sender(s). The Traffic specification (TSpec) can either be based on the mobile users wishes or a best estimate of the incoming traffic characteristics, or on application level session signalling prior to the transfer.

When the LRSVP proxy receives this message, it detects that the message is meant to stay within the access network. The message type indicates that the proxy should initiate an RSVP reservation for a downstream flow and use the information in the message to fill the field in a Path message. The proxy can now change the message type from Path Request to Path, keep the LI flag, and send this Path message towards the mobile node. Since the Session Object and Sender Template were originally set "backwards", the proxy can copy these entries directly as-is to the Path message.

When the mobile node receives the Path message, it responds with a Resv message with the LI flag set. This reserves the downstream resources within the access network for the senders originally identified by the mobile.

2.3. Additional functionality

All the other features of RSVP are used with LRSVP in the standard way including the local repair mechanism and reservation tear down.

Downstream reservations are torn down with the Path Request Tear message. All messages used for local reservations must have the LI flag set in order to keep the signaling within the access network. Intermediate RSVP routers between the mobile node and the LRSVP proxy must not process the Path Request message and they must forward it as an ordinary IP packet.

The proposed scheme also allows RSVP to be used to signal DiffServ Code Points in a DiffServ access network using the RSVP DCLASS object [13]. The DCLASS object is used to represent and carry DiffServ code points within RSVP messages. The mobile node can use the DCLASS object to instruct the LRSVP proxy to mark incoming traffic with certain DiffServ code points to trigger different forwarding behavior within the DiffServ access network. The mobile node, however, needs to be aware of the different code point values and the related services, especially if other than standardized code points are used. This information can be part of host auto-configuration, for example.

In addition, the LRSVP proxy may need information on how to map RSVP requests to proper DiffServ classes if it wants to have closer control of downstream resource allocations. This can be accomplished by using standardized code point values for the DiffServ Assured Forwarding service. Thus, our signalling mechanism can also be used to give relative priority to specific flows without explicit resource reservations.

Furthermore, the proposed signalling can be used at both ends of a data stream. For example, if two mobile nodes in different access networks are communicating with each other, both of them could use the mechanism to allocate resources, for example on their own access networks, independently of each other. This could happen, if the two access networks had a different view of QoS, one uses only IntServ and RSVP, while the other also uses DiffServ. In such a scenario, however, it may be more practical to use RSVP end-to-end, even if the core network connecting the two access networks does not support RSVP.

If the reserved bits in the RSVP Session Object are deemed too valuable to be used for this kind of signaling, the RSVP CAP-object [14] could be used to carry the bit that identifies the signaling as local. The CAP object can be used in the RSVP Path message to convey end host/upstream node capabilities to the downstream network/nodes. This would, however, add another eight bytes of headers in order to carry a single bit of information. In addition, the processing of the messages is more time consuming due to the extra header. In any case, a new Path Request message is still needed, because it would complicate the message processing in routers if the "request to send a Path" was indicated as another bit in the CAP object. With the new message type intermediate routers on the uplink can forward the RSVP

packet to the LRSVP proxy faster, since the router does not need to examine the whole packet and the CAP object in order to find the same information, just the message type.

3. Fast local repair

The RSVP standard [2] defines that Path messages can perform a local repair of reservation paths. When the route between the communicating end hosts changes, a Path message will set the state of the reservation on the new route and a subsequent Resv message will make the resource reservation. Therefore, by sending a Resv message a host cannot alone update the reservation, and thus perform a local repair, before a Path message has passed. It is also said in the RSVP specification that in order to provide fast adaptation to routing changes without the overhead of short refresh periods, the local routing protocol module can notify the RSVP process of route changes for particular destinations. The RSVP process should use this information to trigger a quick refresh of state for these destinations, using the new route (Chapter 3.6, [RFC2205](#) [2]). However, not all local mobility protocols, or even Mobile IP, affect routing directly in routers, and thus mobility may not be noticed at RSVP routers.

When the mobile has moved, it can send a Path message for each upstream resource reservation and initiate the local repair mechanism. But for the downstream, the mobile will need to wait until it receives a Path message, refreshing the reservation states on the new route. Only after receiving the Path message, the mobile can send a Resv message to re-reserve the downstream resources. The Path Request message could potentially be used in mobile networks to initiate a faster local repair on behalf of a mobile node that changed access points during an ongoing RSVP session.

When the mobile node changes its access point in the network, it should send the Path Request message immediately after the handover. The Path Request message is forwarded through the intermediate RSVP routers until it arrives at the cross-over RSVP router that has the reservation for the mobile node stored on a different interface. The message would then instruct the cross-over router to initiate a local repair by sending the needed Path message.

The cross-over router may be located between the LRSVP proxy and the mobile node and will therefore respond to the message. Otherwise, the Path Request message will arrive at an LRSVP proxy, which will initiate a reservation refresh for the mobile. Thus, the closest node to the mobile, the LRSVP proxy or cross-over router, will respond to the routing change.

In some access networks, the access network gateways could also act as LRSVP proxies. If the movement of the mobile node results in packets to flow through a new gateway (and new proxy), the Path Request message would re-reserve the local resources for the new path.

However, the faster local repair scheme has the requirement that the RSVP daemon running on the mobile needs to get an indication when a handover has occurred. The change of access points is best noticed by

the link layer, through the actual wireless device. When the link layer address of the access point changes, this event could trigger a signal to the higher layers, including the RSVP-daemon that a handover has happened. The way the higher layers then take this signaling into account is not a concern of the link layer.

Initiation of the local repair must be done every time the access point changes, regardless of whether the access router changes or remains the same after the handover. If the access router remains the same, the access router itself is the crossover router. The Path Request message sent will be intercepted by this access router, which will reply with the Path message and therefore initialize the resource sharing through its new interface. (Note, that we make the assumption, that each access point has its own dedicated network interface on the access router.) If the access router changes, the local repair mechanism will eventually arrive at either the crossover router or at the access network gateway.

If the whole access network changes as a result of a handover, the situation becomes more complex, and may require a full authentication and admission control procedure to be initiated. From the QoS point of view, the situation does not differ from a situation, where both the access router and the network gateway changed.

The LI flag must be set in all RSVP refresh messages if the reservation is set for the local access network. This will prevent refresh message, like the Path Request message, to be routed out of the access network.

4. Issues with the localized signalling

An important question remains with the localized signalling mechanism: what destination address should the mobile use, when initiating a downstream reservation setup? This has implications on the network path, on which the reservation will be set up.

The Session object and the Sender Template define the parties involved in the reservation. Thus, the destination IP address is not needed in the reservation set up but has an effect on the routing of the packets. The issue concerns the situations, where there are several ingress routes to the access network. In such a scenario, LRSVP proxies might be situated further away from the access routers, closer to the edge of the access network, for example.

There are two fundamentally different options for the IP destination address. The first option is that the mobile node can use the IP address of the host that it intends to communicate with. This has the benefit that a Path message will be routed according to the usual IP

routing mechanisms. Thus, the Path message will be routed to the proxy that will eventually also receive the upstream data flow.

However, if the mobile wants to set up a reservation for the downlink, on behalf of the correspondent node, there is a potential

problem. If the access network has several ingress routes, and hence, most probably, several LRSVP proxies, the data flow may eventually arrive through a path different from the path that had the reservation in place. This can happen due to the basic property of asymmetry in IP routing.

The mobile node might set up a reservation on behalf of the correspondent node through a path using LRSVP proxy A but the data will actually arrive through a path using proxy B. A same problem arises if the mobile node wants to reserve resources without an exact sender IP address. The mobile might want resources for audio streams initiated while browsing the Internet without specifying all possible web servers that it may be using.

A solution to this problem is, that the mobile directs the localized RSVP messages to all LRSVP proxies. This can be achieved using a multicast address of all the LRSVP proxies. As a result, each LRSVP in the access network would receive the RSVP packets, a Path or a Path Request, and respond to the mobile. Since resource reservations are set up on several paths but only some are actually needed in the end, we need a way to remove unnecessary reservations. This can be accomplished through the RSVP soft state mechanism: Unused reservations are revoked using a timeout mechanism, no refresh messages are sent for those paths. This is possible if the reservation refresh is coupled with actual data transferred through that reservation. Reservations are only kept alive if data is actually sent through a particular path.

The multicast functionality can be further modified, so that a proxy will not even send the Path message if it does not receive packets from the specified sender within a timeout. Thus, no downstream reservations are initialized for paths that are not carrying packets belonging to the request.

It seems that there is no best solution to the destination problem. Both solutions have their benefits and drawbacks. The first solution has problems with asymmetric routing and undefined IP destinations (the data senders). The second solution can create a lot more signalling messages and a lot of unnecessary resource reservations. The second solution also requires that the access network supports multicast. Furthermore, one of the original design criteria for the localized signalling was to make the design to transparently co-operate with standard end-to-end RSVP. Now it seems that the localized signalling needs a separate mechanism although the changes required in standard RSVP routers and mobile nodes RSVP daemons are small.

Regardless of the exact solution, an important functionality is that the implementation should be transparent to the mobile node. The

mobile node would always operate in the same way when it wants to setup QoS for downstream flows. The access router can help in supporting the localized RSVP scheme.

Thus, when the mobile node wants to reserve resources for the

downstream, it will should as IP destination address either an LRSVP proxy address provided as part of host autoconfiguration or the default router address. The LRSVP proxy address can be a unicast or multicast address, and it is up to the access network to take care of removing unneeded reservations. If the mobile node does not have the LRSVP proxy address configured, it will use the default router address, which all IP hosts know. The access router can then perform routing lookup and address translation and forward the Path Request message to the correct proxy. Alternatively, it can just forward the message as any IP packet. Thus, the mobile node will always have an address to use as the recipient of the Path Request and the access network nodes will take care of proxy discovery.

Similarly, in an unlikely but still possible case that a mobile would want to set QoS parameters for several upstream paths, it can use the configured LRSVP proxy address or the default router address in the Path message. The access router can then forward the Path to the correct number of LRSVP proxies, and these can then respond with the Resv message. However, how the unneeded reservations are rewoeked is unknown. Thus, it seems, that an access network should consider allowing multiple local upstream reservations.

Furthermore, if the host is mobile and is using Mobile IP to acquire the address, it must use the Care-of-Address as the address in the Session Object address field. If the CoA changes after a handover, the mobile node must create a new Path message, and hence Path state, to set up new upstream reservations. A new Path state is needed, if the filters in the old Path state used the mobile node's CoA as partial information.

For local downstream reservations, the mobile node can send a new Path Request with the new CoA in the Session Object, and simultaneously send a Path Request Tear for the old reservation. The LRSVP proxy will thus create a new downstream reservation and must remove the old reservation state. For end-to-end reservations, an IPv6 correspondent node should use the MIPv6 Binding Update message as an indication to re-establish the Path state using a new destination address in the Session Object.

5. Signalling diagrams

<TBD>

6. Security Consideration

The localized signalling does not raise new securiy issues. The security considerations with standard RSVP apply.

7. Summary and Future Work

In summary, the Localized RSVP requires the following changes in the standard RSVP protocol:

- a) A new message type and number, named Path Request, number 8.
- b) A new message type and number, named Path Request Tear, number 9.
- c) One bit from the Session Object for the use as the Local Indication (LI), bit 0x8
- d) An RSVP router must keep the LI bit set in all messages belonging to that session, if it encounters packets with the bit set.
- e) An RSVP router that is not also a proxy, must forward an RSVP packet with a message type "Path Request" without storing state.
- f) An AN RSVP router should be able to use the Path Request as an indication of the need for a local repair. This can enable faster reservation set up following a handover. This affect point e), because the router that receives a Path Request must first check if it has the Path state stored on another network interface. If one is present, the Path Request message must not be forwarded to the next hop and instead the router must send a Path message towards the mobile node.
- g) Refresh of the soft state for downstream reservations should be tied to actual data flow. Alternatively, the soft state can be kept up by periodic Path Request messages sent by the mobile node.

To summarize, these changes are small and would make RSVP more appealing as a signalling protocol for IP access networks.

We still have to study the interactions between local and end-to-end reservations, for example, whether an existing local reservation could be upgraded into a full end-to-end reservation. Similary, we need to study whether an initial end-to-end reservation attempt could be changed flexibly into a local reservation in case the end-to-end reervation was not succesfull, for example, within the core network connecting the two access networks.

Furthermore, we need to study, whether there could be need for nested proxies, so more than one proxy per path. In addition, a possibility to manage the communication between the end host and the LRSVP proxies would be to use UDP and a wellknown port number. Thus, "local" messages would be sent encapsulated in UDP packets.

8. References

- [1] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview". Internet Engineering Task Force, Request for Comments 1633, June 1994.
- [2] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin. "Resource ReSerVation Protocol (RSVP), Version 1 Functional Specification. Internet Engineering Task Force, Request for Comments 2205, September, 1997.
- [3] J. Wroclawski, "The Use of RSVP with IETF Integrated Services. Internet Engineering Task Force, Request for Comments 2210, September, 1997.
- [4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", Internet Engineering Task Force, Request for Comments 2475, December, 1998.
- [5] G. Huston, "Next Steps for the IP QoS Architecture". Internet Engineering Task Force, Request for Comments 2990, November 2000.
- [6] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet". Internet Engineering Task Force, Request for Comments 2638, July 1999.
- [7] R. Yavatkar, D. Hoffman, Y. Bernet, F. Baker, M. Speer, "SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks". Internet Engineering Task Force, Request for Comments 2814, May 2000.
- [8] Phil Chimento and Ben Teitelbaum, "QBone Bandwidth Broker Architecture". The Internet2 initiative, June 2000. (<http://qbone.internet2.edu/bb/bboutline2.html>).
- [9] S. Gai, D. G. Dutt, N. Elfassy, Y. Bernet, "RSVP Proxy". Internet Draft (work in progress), July 2001 (draft-ietf-rsvp-proxy-02.txt).
- [10] Jukka Manner and Kimmo Raatikainen, "Extended Quality-of-Service for Mobile Networks". Proceedings of the 9th International Workshop on Quality of Service (IWQoS), June 2001, pp. 275-280. Published in the series Springer Lecture Notes in Computer Science (LNCS) 2092.
- [11] IST-BRAIN Project, "Deliverable D2.2: BRAIN architecture specifications and models, BRAIN functionality and protocol specification". March 2001, (available at: <http://www.ist-brain.org>).
- [12] H. Schulzrinne, A. Rao, R. Lanphier, "Real Time Streaming Protocol (RTSP)". Internet Engineering Task Force, Request for

Comments 2326, April 1998.

[13] Y. Bernet, "Format of the RSVP DCLASS Object". Internet Engineering Task Force, Request for Comments 2996, November 2000.

Manner et al

Expires November 2002

[Page 12]

[14] Hamid Sued, "Capability Negotiation: The RSVP CAP Object".
Internet Draft (work in progress), February 2001 ([draft-ietf-issll-rsvp-cap-02.txt](#)).

9. Author's Addresses

Questions about this document may be directed to:

Jukka Manner
Department of Computer Science
University of Helsinki
P.O. Box 26 (Teollisuuskatu 23)
FIN-00014 HELSINKI
Finland

Voice: +358-9-191-44210
Fax: +358-9-191-44441
E-Mail: jmanner@cs.helsinki.fi

Markku Kojo
Department of Computer Science
University of Helsinki
P.O. Box 26 (Teollisuuskatu 23)
FIN-00014 HELSINKI
Finland

Voice: +358-9-191-44179
Fax: +358-9-191-44441
E-Mail: kojo@cs.helsinki.fi

Kimmo Raatikainen
Department of Computer Science
University of Helsinki
P.O. Box 26 (Teollisuuskatu 23)
FIN-00014 HELSINKI
Finland

Voice: +358-50-483-6275
Fax: +358-9-191-44441
E-Mail: kraatika@cs.helsinki.fi

Tapio Suihko
VTT Information Technology
P.O. Box 1203
FIN-02044 VTT
Finland

Voice: +358-9-456-6078
Fax: +358-9-456-7028
E-Mail: tapio.suihko@vtt.fi

Mika Liljeberg
Nokia Research Center
P.O. Box 407
FIN-00045 Nokia Group
Finland

Voice: +358-50-4836791
Fax: +358-
E-Mail: Mika.Liljeberg@nokia.com

Acknowledgement

This work has been partially performed in the framework of the IST project IST-2000-28584 MIND, which is partly funded by the European Union. The authors would like to acknowledge the help of their colleagues in preparing this document, in particular Eleanor Hepworth and Alberto Lopez.

Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

