Robert Hancock
Internet Draft   Eleanor Hepworth
Document: draft-hancock-nsis-framework-00.txt   Siemens/Roke Manor
                                                Research
Expires: August 2002

                                        Cornelia Kappler
                                        Hannes Tschofenig
                                            Jochen Eisl
                                          Jorge Cuellar
                                           Mehmet Ersue
                                            Siemens AG

                                           Xiaoming Fu
                                           Holger Karl
                                            TU Berlin

                                         Marcus Brunner
                                                    NEC

                                        Andreas Kassler
                                       University of Ulm

                                       February 22, 2002

Towards a Framework for QoS Signaling in the Internet
            <draft-hancock-nsis-framework-00.txt>

Status of this Memo

Abstract

   This Internet Draft presents a framework for further development of
   QoS signaling in the Internet. We give a basic model for the
   entities involved in QoS signaling, which is intended to be
   applicable to a very wide range of networking environments, while
   still retaining the flexibility to allow lightweight implementations
   in particular environments and incremental deployment in the
   Internet as a whole.

   As well as the details of the framework itself, we also relate it to
   the NSIS requirements work by mapping the framework to the
   requirements themselves. We also present an initial assessment of
   the applicability of existing QoS mechanisms to be used within the
   framework. Security, scalability, and resilience are considered as
   special issues. The framework leaves open a number of questions
   relating to tradeoffs between simplicity and flexibility, and these
   are summarized in the conclusions.

Table of Contents

 1 Introduction

    This Internet Draft presents a framework for further development of
    QoS signaling in the Internet. We give a basic model for the
    entities involved in QoS signaling, which is intended to be
    applicable to a very wide range of networking environments, while
    still retaining the flexibility to allow lightweight implementations

    Hancock et al.   Informational - Expires August 2002

in particular environments and incremental deployment in the
Internet as a whole.

As well as the details of the framework itself, we also relate it to
the NSIS requirements work [1], by comparing the framework to the
requirements themselves. We present an initial assessment of the
applicability of existing QoS mechanisms to be used within the
framework. Security, scalability, and resilience are considered as
special issues.

1.1 Fundamental Approach

Our basic approach is to define a set of entities which represent
the QOS signaling and associated functions within and 'around' a
single node. With a minor generalization, they can also represent a
group of nodes which act as a unit for NSIS QoS signaling purposes
(the 'virtual router'.)

This approach is similar to that taken in the Policy area, where the
abstract definition of Policy Enforcement and Policy Decision
concepts allows their use in a wide variety of network scenarios.
The same motivation applies in this case: the full variety of ways
in which QoS signaling might get implemented in real networks is
still unclear, and the more abstract the component definitions, the
more deployment possibilities are enabled. In particular, we have
avoided building the framework on concrete assumptions about
possible network topologies and hierarchies, instead building these
up from the basic components as a validation exercise.

A second analogy can be drawn with the way routing is now handled in
the Internet. The full routing problem is probably comparable in
complexity to the full QoS signaling problem, and a modular routing
framework has evolved which allows this problem to be solved
piecemeal. For example, the host and network parts are quite
decoupled, and the inter- and intra-domain areas are handled
separately. Also, multicast is nowadays considered as a function to
be built alongside or on top of unicast routing rather than as an
integrated part of it. We believe that the same level of
decomposition is both necessary and appropriate in considering QoS
signaling solutions, which need to be both widely applicable without
imposing the burden of a single full solution on all participating
nodes.

It is our intention that the framework is used to derive more
concrete requirements for QoS signaling protocols and data, and
possibly QoS extensions to existing protocols. This could be done by
a more formal analysis of NSIS requirements in the context of the
framework, and development of additional implementation requirements
on the protocols themselves. The framework can also serve as a

context for evaluating existing QoS and other mechanisms, either to
be parts of the NSIS solution, or for interactions with it in areas
such as accounting or application layer interactions.

Security is considered an integral problem within the framework
itself. The reason for this is that when any differential QoS
mechanism is available in a network, it may immediately become a
target for abuses such as theft and denial of service; the
assumption is commonly made that this is controlled by requiring
some kind of authentication and accounting relationship between
entities in the network. However, this only works if the
corresponding security relationships are consistent with the way
that the threats of QoS abuse propagate from peer to peer within the
network. In other words, the QoS framework must be underpinned with
a compatible security framework.

1.2 Scope and Design Principles

The fundamental goals of the framework presented here are three-
fold:
*) Applicability
*) Adaptability
*) Re-use

'Applicability' means that minimal assumptions are made about the
environment within which the framework can be used. It is supposed
to be applicable in both access and core contexts, for fixed and
wireless and mobile networks; also, it can operate over various
boundary types, as administrative domain boundaries and also address
space boundaries (including IPv4-IPv6 boundaries), and does not
assume any single style of QoS provisioning paradigm. (A consequence
of this wide applicability is that the core framework itself must be
rather minimal, which is also a desirable characteristic.)

Because we believe that network structures will continue to evolve
into progressively more complex and nested relationships, we have
avoided assuming any particular type of network hierarchy or
classification of node types. The fundamental framework contains a
single type of node that processes QoS signaling, which can be
decorated with particular selections of signaling protocols and
upper/lower layer interactions without modifying the overall
operation.

'Adaptability' means that the way the framework is instantiated in
any particular node or network type must be adaptable to the special
needs of that environment. For example, it must be possible to make

it lightweight for hosts at the edge of the network while making it
scalable in the core; security requirements are also likely to be
network scenario dependent. In particular, the framework must be
able to adapt to the fact that large parts of the Internet will at
least initially be NSIS-unaware, so incremental, minimal-pain
deployment must provide benefits even in this case.

The main method for achieving this is to make the framework modular,
so different parts can be adapted relatively independently: in

particular, the protocols used to carry QoS reservation information
are considered independent of the that information itself, and QoS
provisioning is treated strictly as a local matter, independent of
these (allowing any QoS provisioning paradigm).

'Re-use' means that the framework must be able to incorporate
existing QoS solutions in a natural way, otherwise networks could be
forced to deploy multiple QoS technologies in parallel. In this
draft, we have considered the applicability of both existing
architectural approaches to QoS such as DiffServ, RMD, and Bandwidth
Brokering (section 6) and made an initial assessment of possible
signaling protocols such as RSVP (section 7).

A second aspect of re-use is that we have attempted to minimize the
problem space of NSIS by having the framework hook into other
functions - such as user administration, accounting and especially
upper layer applications - in a well defined way, with a clear
function split between these functions and NSIS. A secondary benefit
of this is that these functions can be implemented with consistent
interactions with QoS elements of the network layer, without having
to be adapted to multiple QoS technology choices.

 1.3 Document Structure

    The document is structured as follows:
    - Section 2 introduces the additional terminology used within the
    draft.
    - Section 3 describes the framework, providing an overview of the
    entities and signaling concepts. The different signaling options
    that should be considered for support by the framework are discussed
    and a brief overview of accounting and security considerations is
    included.
    - Section 4 explores in more detail the framework components, and
    discusses interactions with higher layer functions. The interaction
    with local QoS provisioning mechanisms and routing are also
    highlighted.
    - Section 5 discusses various generic scenarios to illustrate the

use of the functions and definitions described in the previous
sections.
- Section 6 describes other QoS frameworks and protocols and
analyses how aspects of these solutions could be re-used within the
scope of the NSIS framework.
- Section 7 considers existing and potential future QoS signaling
proposals and evaluates their suitability as an NSIS signaling
protocol.
- Section 8 provides on overview of existing QoS parameter
descriptors, and analyses their applicability to the framework
- Section 9 details the security aspects related to the framework.
- Section 10 analyses the framework with regard to resilience and
scalability concerns.
- Section 11 describes the conclusions that can be drawn from the
framework and highlights open issues that need to be addressed.

- Appendix A analyses the framework against the relevant
requirements provided in [1].

2 Terminology

Where possible, this draft uses the terminology defined in [1].
Exceptions and additions are stated here.

Administrative Domain: a region of the network whose boundaries are
defined by the point where common administrative control ends. Also
called a QoS Domain or simply Domain in [1].

Edge router: router at the boundary of a domain or QoS subdomain
boundary; may be responsible for aggregation, shaping/policing, or
similar QoS functions.

NSIS Agent: any entity that takes part in NSIS QoS signaling. QoS
Controllers and QoS Initiators (as defined in [1]) are particular
types of NSIS Agents.

Proxy NSIS Agent: an NSIS agent that acts on behalf of an end host
(which might or might not be NSIS aware).

QoS Provisioning Signaling: the signaling messages that are used to
communicate provisioning commands from a QoS controller to its
routers; part of the overall QoS provisioning mechanism. QoS
provisioning signaling only exists where the QoS controller is
physically separate from the routers it controls.

Virtual Router: all routers provisioned under the control of a
single QoS controller, and seen as a unit by the NSIS signaling.

3 Framework Overview

3.1 Fundamental Building Blocks

   This section introduces the building blocks used within the
   framework and the motivation for their definition.

   The QoS Initiator and QoS Controller concepts introduced in [1] can
   be placed at many different locations within a network, and made to
   interact with many other entities. However, their QoS signaling
   attributes are not altered by differences in location, so it is
   possible to simplify the QoS initiator and QoS controller concepts
   to a single case and define how it supports QoS signaling.

   This single entity can then be taken and used to build more
   complicated scenarios by:
   -     linking entities together in various different ways, such as
     allowing two entities in neighboring domains to exchange
     information or allowing an entity to initiate QoS signaling for an
     aggregate.

   -     giving the entities additional non-NSIS functions such as the
     ability to interact with applications, the ability to know about
     routing in the local network, or the ability to know about domain
     wide resource utilization etc.

   Figure 1 illustrates the framework entity identified above.

```
                          ^         ^            ^
                          .         .            .
                      Accounting    .     User Admin
                      Interaction   .     Interaction
                          .         .            .
                          .  Application    .
                          .  Interaction    .
                          .         .            .
                          V         V            V
                      +-----------------+
                      |    NSIS Agent   |
                      |                 |
                      | +-------------+  |
                      | |              +=====================+\
                      | |              | Outgoing QoS requests   \
                      | |              | "I want QoS X2 for       >
                      | |     QoS      | packets described by Y" /
       +====================+ |    Flow      +=====================+/
       |Incoming QoS requests  \| Interworking|  |
```

```
 |"He wants QoS X for      >                   | /+....................+\
 |packets described by Y"/|                    |/   Interaction with     \
 +======================+ |                     <     routing, traffic      >
                        | |                    |\    engineering etc.     /
                        | |                    | \+....................+/
                        | |                    |  |
                        | |                    |  |
                        | |                    +======================+\
                        | |                    | Outgoing QoS requests   \
                        | |                    |"I want modified QoS Z for>
                        | |                    | packets described by Y" /
                        | |                    +======================+/
                        | +-------------+   |
                        |               |
                        +-----+-----+------+
                              .      .
                             . QoS .
                           Provisioning
                            (local or
                             remote)
                             .      .
                             +      +
                              \    /
                               \  /
                                v
                        Figure 1: Basic NSIS Agent
```

This framework entity is referred to as an NSIS agent. NSIS agents
communicate with each other using peer-to-peer QoS protocols, which
carry the QoS information between NSIS Agents to provision resources
for a traffic flow.

Therefore, the NSIS Agent peer-to-peer protocol can be sub-divided
into two parts:
-     the NSIS signaling protocol that carries data including the QoS
  parameters, and which has to carry out operations such as peer
  discovery, and that should support features such as reservation
  timeouts.
-     the NSIS signaling data that represents the flow and associated
  QoS requirements.
These aspects are discussed further in sections 4.3 and 4.4.

Different configurations of NSIS Agent can be identified based on
their interactions with surrounding functions and optional
capabilities in processing of QoS signaling messages. These are as
follows:

- If the NSIS Agent does not receive input signals from peer agents concerning QoS requirements, it probably receives QoS request information from the higher layers (applications).
- NSIS Agents can support N inputs and M outputs corresponding to a given flow, but 1:1 and n:1/1:n (to reflect the aggregation/ deaggregation case) are the only common ones.
- QoS provisioning can be treated as a black box (invoked in an implementation dependent way for example via a technology-specific convergence layer) if the QoS provisioning signaling used when carrying it out cannot be fitted into the framework. In this sense, there may be QoS signaling protocols that do not come under the NSIS 'umbrella'; our general intention is to fit existing protocols into the framework if this can be done simply, but there is no aim to generalize the framework to cover all possible QoS-related protocols.
- Conversely, if the NSIS signaling is being propagated along the traffic path within the network, it might be used directly to control the local QoS provisioning, and no additional provisioning actions are needed from the QoS controller.
- Some NSIS agents might simply do protocol or address boundary interworking, or gather and forward accounting/authorization information. In this case, they wouldn't perform any QoS provisioning or modify the flow signaling at all.
- Specialised NSIS Agents may interact with routing protocols or traffic engineering protocols etc. to support features such as sophisticated path capability discovery. See Section 4.5
- NSIS Agents can assume the role of proxy, and in this capacity can initiate and terminate signaling on behalf of a QoS initiator. Further details of this are provided in Section 3.7.1.

3.2 Fundamental Network Structures

In this section, we introduce some representative network structures which can be used to describe the range of actual allowable deployments of the fundamental NSIS agent building blocks of section 3.1. At the highest level, we can picture the networks across which QoS is signaled with the following structure. The complete end to end path covers a number administrative domains, each carrying a single path segment. (Within each domain there may be further subdomains corresponding to QoS technology boundaries.)

```
[+]------------------------ Path ------------------------------[+]

        [+]------- Path Segment ---------[+]

                    --------
```
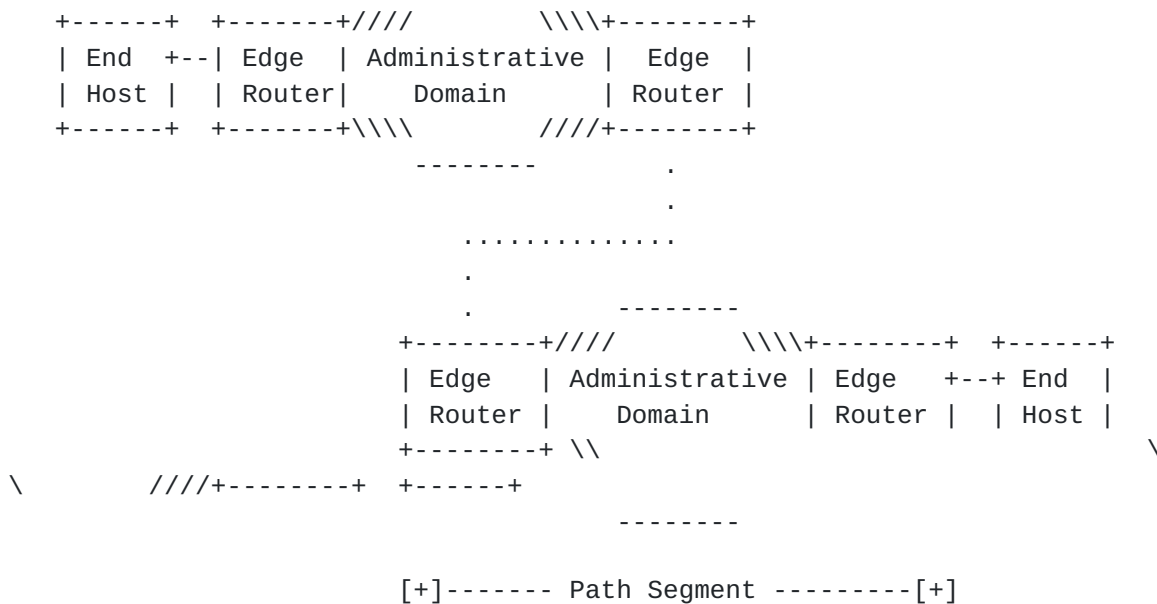
```
        +------+  +-------+////            \\\\+--------+
        | End  +--| Edge  | Administrative |  Edge  |
        | Host |  | Router|    Domain       | Router |
        +------+  +-------+\\\\          ////+--------+
                            --------         .
                                             .
                              ...............
                              .
                              .        --------
                       +--------+////           \\\\+--------+  +------+
                       | Edge   | Administrative | Edge    +--+ End  |
                       | Router |    Domain      | Router |  | Host |
                       +--------+ \\                                          \
  \       ////+--------+  +------+
                                            --------

                       [+]------- Path Segment ---------[+]

                    Figure 2: Top Level Network Structure
```

We believe that it is crucial to consider the complete end to end
path and then work down in detail to individual hops: this is partly
because QoS is meaningful primarily as an end to end concept, and
also because several critical technicalities (such as asymmetric
routing) are emphasized by this 'macro' view.

At this level, we see the network foremost at the level of
administrative domains and QoS subdomains rather than individual
routers (except in the special case that the domain contains one
link). Because administrative domains are defined as administrative
entities, we can expect that special security requirements apply to
the signaling between them. Subdomains are introduced to allow the
fact a given QoS provisioning mechanism may only be used within a
part of a domain, typically for a particular subnetwork technology
boundary. Another example might be that a subdomain uses some
special routing mechanisms, e.g. to support mobile hosts, and that

this may indirectly force the use of special QoS provisioning
methods. End hosts may be connected through one or more domains;
this is indicated by the dotted line in Figure 2.

Note that although the simple idea of a sequence of domains with one
level of subdomain covers many basic scenarios, it is certainly not
rich enough to encompass the possible configurations that could
arise in the real-world. For example:
*) Administrations could be nested (reflecting internal
organizational boundaries, where some subset of typical accounting

or security requirements might be applicable).
*) An end-user (as seen by the network) might support multiple end-
hosts (as seen by the user). Examples might be a dial-up user
supporting a home network, or a mobile cellular user supporting a
PAN; in each case, the full weight of a single 'network-network'
interface would be inappropriate.
*) There may be address space or technology boundaries within or
between networks, whose problems need to be addressed specially.

For these reasons, we have not tried to identify a specific
hierarchy of protocols; the framework is supposed to be more
generally applicable. Once consistency of the framework at the
overall level is understood, detailed requirements for protocols can
be considered in terms of particular scenarios.

Aggregation typically takes place at both domain and subdomain
boundaries, where edge routers are located. Edge routers may have
more responsibility than other routers, for example to carry out
this aggregation, or perform admission control. Where these
functions involve interactions with QoS signaling, there will be an
NSIS agent performing this signaling role.

The next figure shows the general structure of a domain or
subdomain, which is simply a network of connected routers.

```
   +----------------------------------------------------------------+
   |                    QoS (sub)domain using any                   |
   |                       QoS technology                           |
   |                                             +------+           |
   |                                             |Router|           |
   |                  +------+                    |      |          |
   |                  |Router|     +------+       |      |          |
+------+              |      |     |Router|------|       |----+------+
| Edge |--------------|      |-----|      |      +------+     | Edge |
|router|              +------+     |      |       /          |router|
|      |\             /   \    +------+    /           /|     |
+------+ \           /     \      \       /          / +------+
   |      \         /       \      \     /          /       |
   |       \       /     +------+    \   /         /         |
   |        \  +------+   |Router|    +------+    /          |
   |         \ |Router|   |      |    |Router|   /           |
   |          \|    +----|      |----+    | /             |
   |           \|       |     +------+    |    |/           |
   |          +------+                +------+             |
   +----------------------------------------------------------------+
```

Figure 3: General Domain/Subdomain Structure

One particular scenario is that the resources of these routers may
be governed by the decisions of a bandwidth broker, as shown in
Figure 4.

```
   +--------------------------------------------------------------+
   |                     Domain/Subdomain                         |
   |                                                              |
   |                      +-----------+                           |
   |.....................| Bandwidth |.......................... |
   |.             .......|  Broker   |............             . |
   |.             .      +-----------+            .             . |
   |.             .          .     .     .         +------+     . |
   |.             .    +------+   .    .           |Router|     . |
   |.             .    |Router|   .  +------+       |      |     . |
   +------+       .    |      |   . |Router|------|       |----+------+
   | Edge |------------|      |----|      |       +------+     | Edge |
   |router|       .    +------+  . |      |          /         |router|
   |      |       .       /   \   . +------+        /          /|      |
   +------+\       .      /     \  .         \     /          / +------+
      |     \      .     /       \ .          \   /          /      |
      |      \     .    /         \ .          \ /          /       |
      |       \  +------+       +------+        \ /         /        |
      |        \ |Router|       |Router|     +------+      /         |
      |         \|      |       |      |     |Router|     /          |
      |          \|      +----|       |----+      | /           |
      |           \|      |    +------+      |     |/            |
      |            +------+                  +------+             |
   +--------------------------------------------------------------+
```
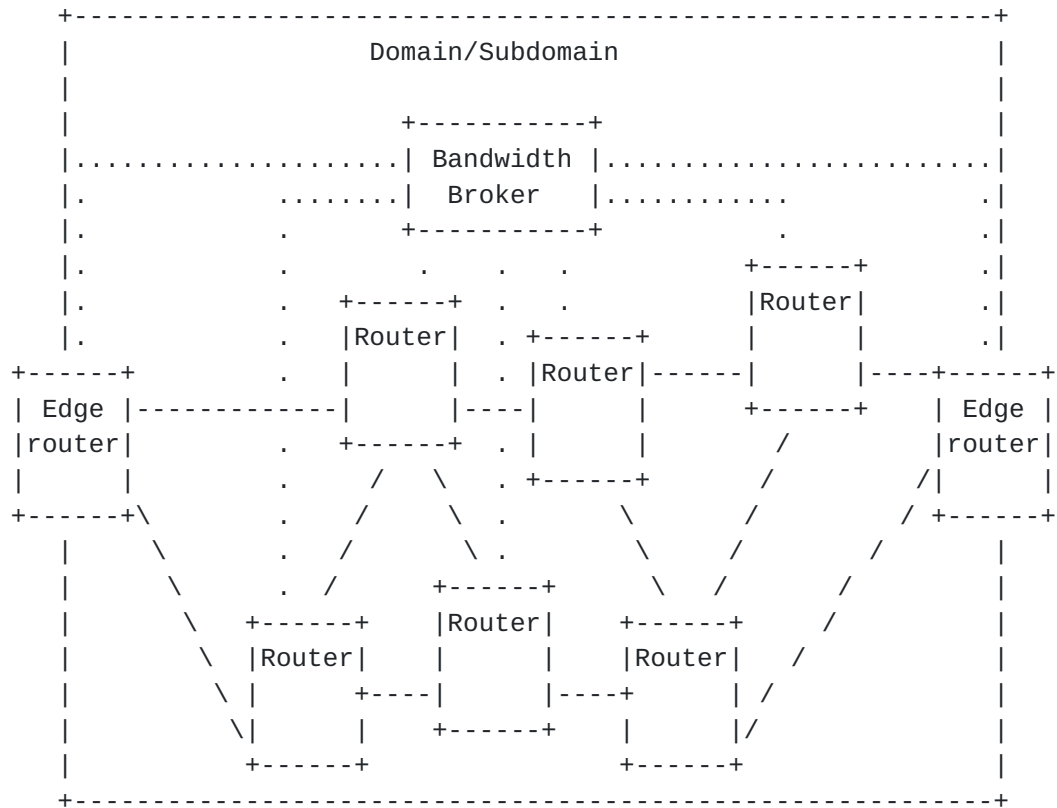
              Figure 4: Bandwidth Broker in a Domain or Subdomain

   Bandwidth broker or similar solutions can be expected to be common
   mechanisms where large or complex domains need to be QoS aware, and
   it is therefore important to consider how to fit them into the
   overall NSIS framework. At a minimum, we need to be able to
   propagate NSIS signaling between end hosts transparently through
   such a domain; ideally, there should be some interaction with the
   bandwidth broker itself to ensure that the provisioning it carries
   out reflects the QoS requirements of the underlying flows. The NSIS
   framework needs to define what mediates this interaction and where.

   Note that we don't consider the BB-router signaling as a fundamental
   part of NSIS, since it is essentially a local mechanism. However,
   there is an option to exploit work done in defining NSIS for use to
   carry out this type of remote provisioning. This possibility is
   discussed in section 5.4.

A particularly crucial point to note about the macro-scale network
structure is that end to end packet flows are likely to be
asymmetric, and this applies to both signaling and data packets. A
characteristic situation is shown in Figure 5. This simple fact has
significant impact on the signaling possibilities even within a
single edge domain, since only in special cases can the entry point
for incoming traffic be predicted or controlled, in which case QoS
signaling for this traffic has to be carefully routed. More
implications of this situation are considered in section 3.4.

```
>> = traffic flow H1->H2
<< = traffic flow H2->H1        +--------+
H = Host                        |  QoS   |
R = Router       +-----+        | domain |
                 |     |        +-+       |
                 |   +-+>>>>>>>|R|>       |
                 |   >|R|        +-+ >     |
                 |   > +-+       |    >   +-+
                 | >    |        |    >>|R|
                 |>     |        |      +-+
                 >      |        +------ >
  +--+     +-+     +-+  |                >
  |H1|><><|R|<><<>|R|   |                >
  +--+     +-+     +-+<   \               +-+
                 |   <   |       +--|R|>----+
                 |    <  |       |  +-+ >    |
                 |    <+-+        |       > +-+
                 |     |R|<<<<<<<<    |         >|R|>>>>>+--+
                 |     +-+          < +-+         +-+     |H2|
                 | QoS  |           <|R|<<<<<<<<<<<<<<<+--+
                 |domain|          +-+            |
                 +------+          |QoS domain|
                                   +----------+
```

                 Figure 5: Asymmetric Routing at the Domain Level

 3.3 Abstract Entities in QoS Signaling paths

   The NSIS requirements are phrased in terms of two abstract entities,
   the QoS Initiator, and the QoS Controller.

   The QoS initiator starts the request for QoS for a user flow. It
   might be located in the end system or within some part of the
   network, such as the edge router for an access network, or even an
   application layer proxy. The distinguishing feature of the QoS
   initiator is that it acts on triggers coming (directly or

indirectly) from the higher layers in the end systems, mapping the
QoS requested by them. It also provides feedback information to the
higher layers which might be used by transport layer rate management
or adaptive applications.

In our model, the QoS initiator is a particular instance of the
generic NSIS agent shown in Figure 1, which will have
*) Triggers from and feedback to upper layer applications
*) Typically, no incoming QoS requests
*) Typically, one outgoing QoS request per application data flow
*) Local QoS provisioning functions for the first hop (for both the
IP and link layers)

Likewise, the QoS controller manages and enforces QoS further along
the path. It might be located in some or all routers, or in a
separate network element, e.g. in a bandwidth broker (note therefore
that QoS controllers are not constrained to lie on the traffic
path). If QoS is to be provisioned on a path segment, there must be
at least one QoS controller to do this (but this would not be needed
for example in overprovisioned QoS subdomains). The QoS controller
does not interact with higher layers, but interacts with the QoS
initiator and possibly more QoS controllers further along the path.

In our model, the QoS controller is another particular instance of
the generic NSIS agent, which will have:
*) Incoming and outgoing QoS requests. Incoming QoS requests are
interworked to outgoing requests. The interworking might consist of
managing additional flows (e.g. aggregation or disaggregation), or
interpreting the QoS requested by the user in terms of the QoS
allowed by some SLA (e.g. on inter-domain links).
*) If the local QoS cannot be provisioned by sending the appropriate
outgoing QoS request, subdomain-specific QoS provisioning signaling
can be invoked. In this case, the provisioning mechanism is opaque
to NSIS and can be locally implementation specific.
*) Accounting and authentication information may be exchanged with
local AAA nodes. The precise protocol used to do this is again
outside the scope of NSIS signaling.

The QoS initiator and controller(s) interact with each other. This
interaction involves the exchange of data (QoS control information)
over some signaling protocol. In terms of our framework, this is
simply considered as a peer-peer protocol exchange between NSIS

agents (i.e. there is no client-server concept, or differences
between initiator-controller and controller-controller protocols
built into the framework, although the protocol selected may still
depend on the local environment).

A simple layer model covering a single path segment with a single
QoS controller is shown in Figure 6. The scope of NSIS within the
context of this diagram is therefore the protocol between the NSIS
agents (initiator and controller), including selection of signaling
protocols to carry the QoS information, and the syntax/semantics of
the information that is exchanged. The provisioning is being
carried out using non-NSIS mechanisms.

```
.............    ................
.  request/  .   .  response/    .
.trigger from.   . feedback to  .
.higher layer.   .higher layers .
.............    ................
        |            ^
        |            |
        |            |       ...............
        |            |       . QoS Control .
       V             |       . Information .
   +----------------+        ...............      +----------------+
--->|                |        |------------------------>|                |       |->
   |                |        |       QoS signaling     |                |       |
   | QoS Initiator  |        |      (request/query,    | QoS Controller |       |
   |                |        |     response/error etc.) |                |       |
<---|                |        |<------------------------|                |       |<-
   +----------------+        +----------------+        +----------------+
      ^              |                              ^              |
      | ........... |                              | ........... |
      |    QoS      |                              |    QoS      |
      | provisioning |                             | provisioning |
      |  commands &  |                             |  commands &  |
      |  responses   |                             |  responses   |
      | ........... |                              | ........... |
      |              |                              |              |
      |             V                               |             V
   +-------------------------------------------------------------+
   |        Administrative domain or QoS domain using any        |
   |                     Qos technology                          |
   |                                                             |
   |     +------+        +------+                  +------+      |
   |     | Edge |        |Router|                  | Edge |      |
   ========|Router|======|        |==================|Router|======
   |     |      |   |    |      |   |    flow path    |      |   |
   |     +------+        +------+                  +------+      |
   +-------------------------------------------------------------+
```

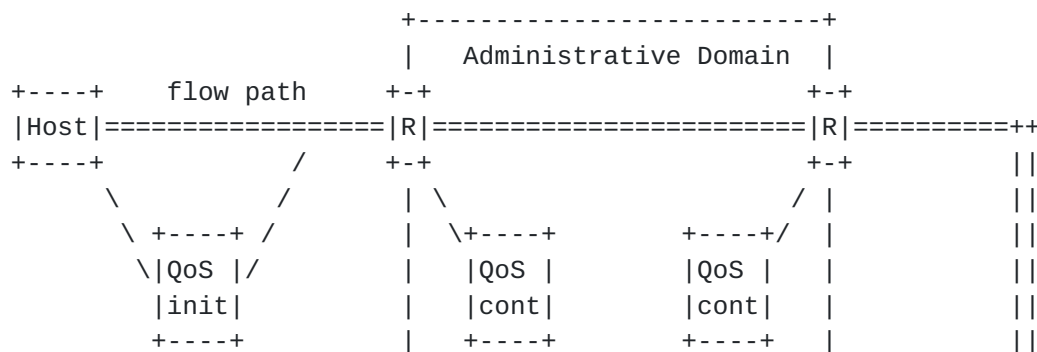   Figure 6: Generic scope of signaling

A second diagram, Figure 7, concentrates more on the overall end to
end (multiple QoS domains) aspects, in particular:

1. The QoS initiator need not be located at the end system (for
example, it might be an application signaling server), and the QoS
controllers are not assumed to be located on the flow path. However,
they must be able to identify the next hop QoS controller, to do
which might require knowledge about the path's egress point; more
detailed path information might be required to carry out the QoS
provisioning correctly.

2. Only a unicast flow is shown, with the QoS initiator at or near
one end. However, we do not exclude bi-directional flows with the
QoS requested by either end. Multicast or anycast flows or flows
with variable paths within a subdomain (e.g. to a mobile end system)
are also logically possible.

3. Any domain may contain QoS administration functions (e.g. to do
with traffic engineering, admission control, policy and so on).
These are assumed to interact with the QoS initiator and controllers
(and end systems) using standard mechanisms.

Although the figures show QoS controllers at a very limited number
of locations in the network (e.g. at domain or subdomain borders, or
even controlling a complete domain), this is only one possible case.
In general, we could expect QoS controllers to become more 'dense'
towards the edges of the network, but this is not a requirement. An
overprovisioned domain might contain no QoS controllers at all (and
be NSIS transparent); at the other extreme, QoS controllers might be
placed at every router. In the latter case, QoS provisioning can be
carried out in a local implementation-dependent way without further
signaling.

```
                         +--------------------------+
                         |  Administrative Domain   |
    +----+    flow path      +-+                          +-+
    |Host|================|R|========================|R|=========++
    +----+          /      +-+                          +-+        ||
        \          /        | \                        / |        ||
         \ +----+ /         |  \+----+        +----+/  |        ||
          \|QoS |/          |   |QoS |        |QoS |   |        ||
           |init|           |   |cont|        |cont|   |        ||
           +----+           |   +----+        +----+   |        ||
```

```
                        |        ^            ^      |            ||
                        |        |            |      |            ||
                        |        V            V      |            ||
   +-+                  |    +------------------+    |            ||
   |R| = Edge router    |    |QoS administration|    |            ||
   +-+                  |    |     functions    |    |            ||
                        |    +----------------+ |    |            ||
                        +--------------------------+ |            ||
                                                     |            ||
              +---------------------------------------------+    ||
              |            +--------------------+         |    ||
              |       +-----|   QoS controller   |--+      |    ||
              |      /      +--------------------+   \     |    ||
              |     /                                 \    |    ||
              |    /      +--------------------------+  \  |    ||
              |  /        |      QoS Subdomain        |  \ |    ||
   +----+        +-+     +-+                          +-+   +-+  ||
   |Host|========|R|======|R|***********************|R|===|R|====++
   +----+        +-+     +-+     aggregate path     +-+   +-+
                  |       | \                       / |    |
                  |       |  \+----+       +----+  / |    |
                  |       |   |QoS |       |QoS |  |    |
                  |       |   |cont|       |cont|  |    |
                  |Admin. |   +----+       +----+  |    |
                  |Domain  +-------------------------+    |
                  +------------------------------------------+
```

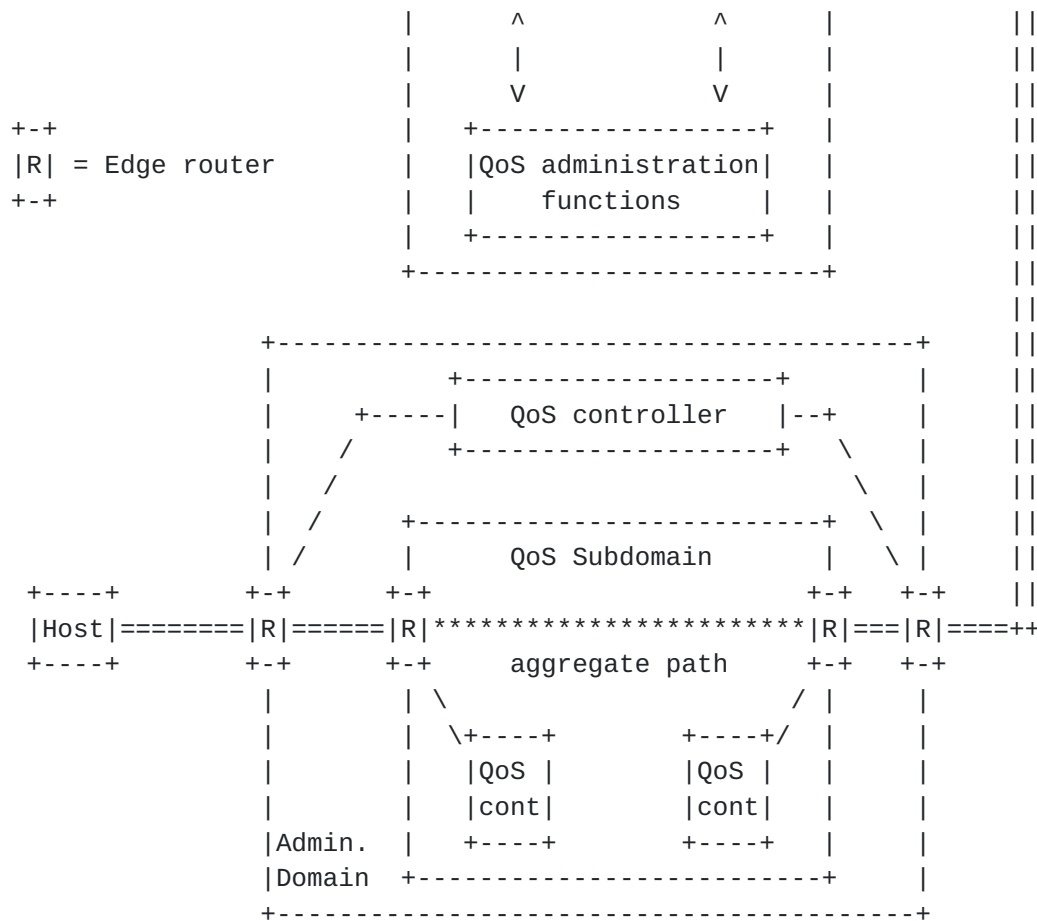   Figure 7: Signaling in Multiple (QoS) Domains

 3.4 Basic Signaling Paths

    It has been a long term question in NSIS whether and how to support
    different reservation models, sender initiated, receiver initiated,
    or bi-directional. (Here, 'sender/receiver' refers to the direction
    of user traffic flow, while 'initiated' refers to the role of the
    QoS initiator. A bi-directional reservation is logically a
    combination of a sender and receiver initiated reservation carried
    out by a single QoS initiator.)

    There are several models for how this might take place at the macro-
    level (i.e. at the level of whole domains). Which model is used must

    Hancock et al.   Informational - Expires August 2002
17   Towards a Framework for QoS Signaling   February 2002

    be fixed at this level level, since this cannot be decided locally
    without harming interoperability, especially taking into account
    that asymmetric routing is possible even at the domain level as
    discussed earlier.

3.4.1 Sender Initiated

Considering a sender initiated reservation for a single
unidirectional flow, the eventual setup must converge to the
situation shown in Figure 8. In the figure, each 'QC' represents a
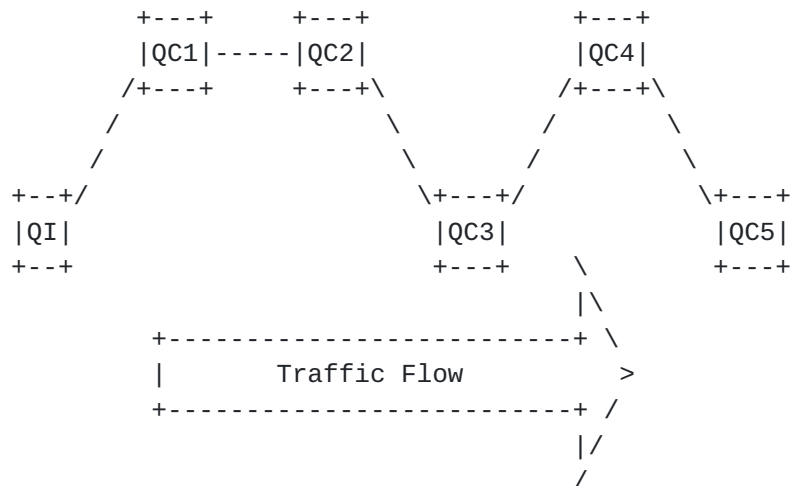single 'virtual router' which could be a complete domain.

```
            +---+      +---+              +---+
            |QC1|-----|QC2|              |QC4|
           /+---+      +---+\            /+---+\
          /                 \          /        \
         /                   \        /          \
      +--+/                   \+---+/           \+---+
      |QI|                     |QC3|             |QC5|
      +--+                     +---+     \        +---+
                                          |\
       +------------------------+  \
       |         Traffic Flow         >
       +------------------------+  /
                                          |/
                                          /
```

                 Figure 8: Basic Sender Initiated Reservation

Here, it is natural to assume that the actual reservation message is
generated at the QoS initiator QI, and then propagated sequentially
through the QoS controllers QC1...QC5 to the endpoint. Since this
runs in the same direction as the traffic flow, the underlying IP
routing of the signaling packet towards the flow destination can
usually be exploited to find the next hop QoS controller, although
other mechanisms might also be allowed for particular inter-domain
NSIS protocols. The use of the underlying routing protocol to reach
the next QoS controller (shorthand: the 'routing method') is
discussed in more detail in section 4.5.

We therefore make the assumption that basic signaling message flows
follow this pattern in the sender initiated case. Note that no
special forwarding state is needed in the QoS controllers to route
the signaling messages.

3.4.2 Receiver Initiated Reservations

In this case, the basic picture is very similar, except that the
traffic flow is in the opposite direction. Because of asymmetric
routing, QC2/3/4 have been replaced by QCa/b/c for the reverse
direction. However, we cannot assume that propagation of the
signaling messages from the QoS initiator is possible in the same

way as in the sender initiated case, because the underlying IP
routing cannot be used to route the signaling packets backwards.

Even if some QoS controllers can be configured to know their
upstream neighbor for a particular flow, if even a single one (e.g.
at an inter-domain boundary) needs the routing method to discover
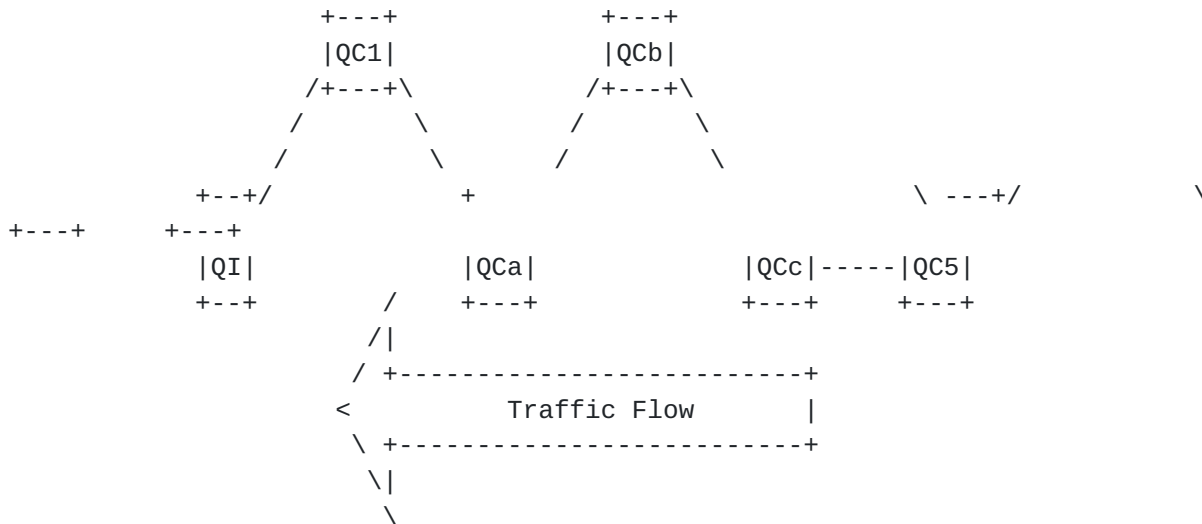its neighbor, the whole signaling procedure will fail.

```
                +---+                +---+
                |QC1|                |QCb|
               /+---+\              /+---+\
              /      \            /      \
             /        \          /        \
         +--+/          +        /          \ ---+/          \
 +---+      +---+
        |QI|              |QCa|            |QCc|-----|QC5|
        +--+        /     +---+            +---+      +---+
                  /|
                 / +--------------------------+
                <            Traffic Flow       |
                 \ +--------------------------+
                  \|
                   \
```

                 Figure 9: Basic Receiver Initiated Reservation

There appear to be two basic solutions to this problem:

*) "RSVP style": Here, a message must be sent from the far end
(QC5), which can use the routing method to work along QCc/b/a/1 back
to QI. At each stage, per-flow forwarding state must be installed in
the QoS controllers and QI so that each knows its upstream neighbor.
(This message can also be used to probe for resources.) Then, the
actual reservation can be initiated from QI in the same way as the
sender initiated case. (We can regard the first message, analogous
to an RSVP PATH, as part of a registration phase, see section 4.3.)
Note that there must be some message to stimulate QC5, which might
be application layer or part of the QoS signaling.

*) "Reflection style": Here, the reservation message is generated by
QI and sent directly to Q5 using normal routing. Q5 then sends the
reservation request on behalf of QI, and the routing method can be
used to discover the 'previous' hop QoS controllers along the path.
This does not require reverse path forwarding state to be installed
along the path and can save one end to end transmission time, but
requires more careful consideration of security and accounting
issues, since the reservation is now being set up in the reverse
direction. Also, resource probing and exception handling may be more
complex in such a approach.

The tradeoffs between these two techniques essentially relate to

1. The amount of state stored at intermediate QoS controllers, and
the necessity to maintain this state in the event of routing
changes.
2. The number of end to end transmissions required.
3. The complexity of handling admission control and accounting
policy information securely when the reservation starts from the
'wrong' end.
4. Resource query requirements.
5. The necessity to retain backwards compatibility with end to end
RSVP signaling (harder with reflection).

Selection between these options requires further analysis of the
scenarios and requirements. Fortunately, it seems that the
implications of this decision for the other parts of the framework
are limited (there is a potential impact on QoS violation reporting,
section 5.7).

3.4.3 Bi-Directional Reservations

In this case, QI wants to initiate the reservation for both
directions of the flow.

It is important first to consider exactly what benefit a special bi-
directional reservation might provide over independent sender and
receiver reservations made by QI in parallel. If the routing is
totally asymmetric, the resulting configurations will be identical;
therefore, bi-directional reservations are at most an optimization
to exploit regions of symmetric routing, typically towards the edges
of the network, e.g. starting at QI. In this symmetric region, it
may be possible to provision the QoS for the two flows more
efficiently when they are considered together (see also section
4.2). We assume that once the bi-directional reservation splits it
will be very hard to correlate the two parts at the other end, and
to enable the two reservations to be treated together near QI
without requiring complex correlation in the network, they should be
signaled in a single message.

```
        +---+    +---+    +---+    +---+
        |QC1|----|QC2|    |QCb|    |QC4|
        /+---+\   +---+\  /+---+\  /+---+\
       /      \       \/       \/       \
```
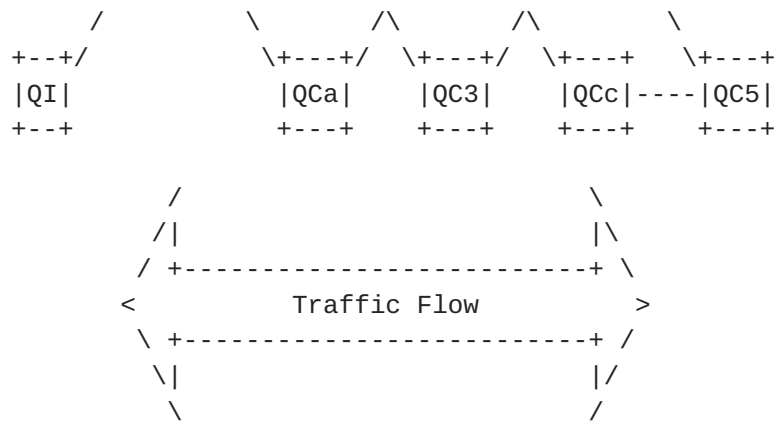
```
        /           \          /\          /\          \
   +--+/           \+---+/  \+---+/  \+---+   \+---+
   |QI|           |QCa|     |QC3|     |QCc|----|QC5|
   +--+           +---+     +---+     +---+     +---+

        /                            \
       /|                            |\
      / +--------------------------+ \
     <           Traffic Flow           >
      \ +--------------------------+ /
       \|                            |/
        \                            /
```

               Figure 10: Basic Bi-directional Reservation

   In the case shown in Figure 10, a bi-directional reservation could
   be set up between QI and QC1, but would then have to be split out
   into independent sender and receiver reservations for the remainder
   of the path. There appear to be two possibilities for how this could
   be done.

   1. The "RSVP style" is used to discover the upstream chain of QCs
   from QI. When QI notices that the first and last hop QoS controller
   are the same, it can send the combined reservation message to QC1,
   which then splits them for the uplink and downlink directions. The
   message flow then looks like a direct combination of a sender
   initiated reservation and RSVP style receiver initiated reservation,
   optimized over the first hops until the paths split.

   2. QI knows somehow (out of band) that it has a symmetric route to
   the first QoS controller (for example, as a consequence of the
   access technology it is using), and sends the bi-directional
   reservation directly to QC1. QC1 can then initiate a standard sender
   initiated reservation for the uplink direction, and use either style
   for the downlink direction.

   In neither case does the rest of the network (between QC1 and QC5)
   know that a bi-directional reservation has taken place.

   The first technique is clearly more general, but enforces the use of
   the RSVP style for the receive path everywhere. The second technique
   introduces a distinct third reservation type and can only apply
   where the QI knows about the uplink neighbor in this way. However,
   it is potentially more suited to operation in an environment where
   QC1 acts as a proxy initiator for QI (see section 3.7.1), and makes
   the signaling task for QI very simple indeed. It may cover the bulk
   of scenarios where bi-directional reservations are actually

valuable. Such a scenario is considered in more detail in .

Therefore, again, a choice between the two methods requires further analysis of the critical scenarios and the tradeoffs between the two methods.

3.5 Impact of Accounting Considerations

The traditional way of collecting accounting information is the following: after the successful authentication and authorization of the user, appropriate filters and meters are installed at the edge devices to collect information about resource consumption. Those data are merged together and sent to the home domain of the user. In our case, also, the edge devices monitor also the QoS treatment and generate corresponding accounting data, and, moreover, each edge router of each administrative domain collects resource consumption information, including again the different QoS treatment.

This distributed accounting data may be linked or merged into a single record, and periodically transmitted to the entity that takes care of the billing of the end entities. The charging entity could be for instance the home domain of the user. But also other business and payment models are possible, without changing the metering procedures sketched. Perhaps the domains pay to each other using their own measurements at their edge routers, based on service level agreements, and the user has to pay only what he is accounted for at his own access device. This also allows that QoS traffic of different users be aggregated. Obviously there is a need for different domains to be able to understand the collection of accounting data and the charging of them.

Note that there is in general no need to collect and report the accounting data in real-time to the home network. Accounting data may be collected as a batch-job and transmitted via the existing infrastructure, for example COPS or DIAMETER. However within a single service provide it may still be necessary to collect all accounting data relatively quickly to determine whether a user has reached a particular limit or not.

The above described mechanism should not only work with a subscription with a network provider or some form of network-based pre-payment, but also for a larger variety of forms of payment based on e.g. local cash payments, pre-paid cards, credit cards, electronic purses or micropayments. The form of payment may have some influence on the security architecture and on the accounting procedure.

It is currently an open issue how the price for a QoS service reservation should be determined. Some work has be done in the academic community but it is still an issue whether the user should

pay to the first service provider only (for the entire end-to-end

reservation) or whether it is necessary to somehow involve all other
providers to determine the final price of the reservation.
Those billing aspects (as opposed to accounting) are left out of the
scope of this draft.

3.6 Security Overview

This section gives an overview of the security issues related to the
described framework. The security architecture is divided into three
categories: The first category raises security issues for the user
to network communication. The second category discusses intra-domain
and network-to-network issues. Finally the last category deals with
end-to-end security issues.

The main concern of security for the user-to-network communication
deals with the separation of the initial authentication and key
agreement step and the security protection of the QoS signaling
messages originated from the user's host. Issues like the discovery
of the entity to which the user has to authenticate, user identity
confidentiality and different authentication and key agreement
mechanisms are critical, and are discussed in detail in section 9.1.
Signaling initiated by the user usually receives the highest
attention since authorization and accounting procedures strongly
depend on a successful authentication procedure.

To secure the messages that travel within an administrative domain
hop-by-hop security is applied. It is obvious that such a hop-by-hop
security protection of signaling messages aims to be fast and is
based on the assumption that the main threat originates from
adversaries not residing on the path of the QoS signaling messages.
However there is still a need to establish the security associations
required to secure this communication. Because of the static network
structure and the user-independence there is more flexibility for
security association establishment. The communication between
different networks is the next issue that needs investigation.
Authentication, authorization and accounting that are also executed
between different networks is assumed to be secure to guarantee that
the traffic conforms to service level agreements.

Finally there is the notion of end-to-end security. Two fields of
further investigation have been identified which are usually not
addressed within the context of QoS signaling protocols. A QoS
signaling protocol may exploit existing security association
(possibly established by preceding protocols and already available
to the application) to protect parts of the QoS signaling message

that are not modified in transit. Furthermore the QoS protocol may
be used to carry key management protocol messages, which are likely
to be opaque to the signaling protocol itself. If no end-to-end
security association is available then message exchange of the
signaling protocol may help reduce the latency for an application
setup.

The mutual implications QoS and security have for each other have
received relatively little consideration at system level up to now,
so there is a large set of potential open issues about the correct
way(s) to secure QoS signaling. These open issues are gathered
together in the conclusions section 11.

 3.7 Refinements and Extensions

This section discusses various ways in which the framework can be
extended, or deployed to achieve additional functions compared to
the core set introduced so far.

 3.7.1 Proxy NSIS Agents

A Proxy NSIS Agent acts as a NSIS agent in lieu of an end host. It
has a mandate from the end host to handle all NSIS-related signaling
and processing in its place. In particular, it initiates (as QoS
initiator) and terminates (as QoS controller) NSIS signaling on
behalf of the end host. It is the proxy's responsibility to relay
all relevant NSIS signaling information to the end host, possibly in
a condensed or otherwise optimised form. Normally, the signaling
between the proxy and end host should be considered a form of NSIS
signaling and within the scope of the framework; however, a proxy
with special functionality might also be used to isolate NSIS-aware
and NSIS-unaware networks. This use of the term 'proxy' is analogous
to the use in RSVP [22].

The proxy is located upstream from the end host. For all other NSIS
agents further upstream, the proxying can be considered transparent,
that is, they do not need to be aware that they are talking to the
proxy rather than to the end host directly (unless they happen to
know the true IP address of the end host). The proxy inserts its own
identification into the NSIS signaling to take the place of that of
the end host (see also section 4.4). Note that this requires the
flexibility that the network allows the QoS signaling to be managed
from a point which is not the end point of the traffic flow, which
is a fundamental assumption in our framework.

The reasons for employing a proxy can be manifold. The end host
might not have the processing capability necessary for acting as a

NSIS agent and therefore uses a proxy to carry out this function for him. From the network operator's perspective, using a proxy is desirable when the connection between end host and network is either of low bandwidth (expensive) or error prone, or shouldn't be burdened with signaling for some other reason. A good example of such a connection is the air interface in UTMS. In this case, the network operator might even prescribe the use of this type of NSIS proxy agent. A proxy may also be useful for hiding micro-mobility from the network, and thus simplifying QoS reservation re-setup after handoff, it might be deployed at an addressing boundary where deep translation of signaling message content was required.

For scenarios illustrating the use of signaling proxies see Section 5.1.

3.7.2 Multicast

It is currently an open question within NSIS whether support for signaling QoS for multicast applications is actually required. There are several reasons why it might be preferable to consider a unicast-only NSIS framework as the initial step: firstly, that multicast support is a source of considerable complexity in any network layer signaling and that one of the goals of NSIS is to allow lightweight signaling solutions; and secondly, that we already have a solution for the protocol for multicast signaling, namely RSVP.

We must therefore ask how much commonality there is between signaling for unicast and multicast applications, and how much benefit there is to gain in having a single signaling solution.

In terms of the framework presented so far, which has been developed mainly with the unicast case in mind, it is clear that several components could be re-used in a multicast scenario. For example, the NSIS signaling data (including QoS classes) should be mainly common, and likewise the basic concepts of NSIS agents inside QoS controllers and initiators (including issues like their peer-peer relationships, and authentication protocols and so on).

On the other hand, some parts of the framework would be totally different. In particular, for multicast, it probably only makes sense to consider receiver initiated reservations, whereas for unicast a particularly simple sender-only case is possible (one solution, based on RSVP with the multicast capabilities removed, is outlined in section 7.2). Also, the accounting implications for multicast are not well understood which will have an impact on

security analysis also. Finally, any QoS implementation that is
routing aware (e.g. bandwidth brokers) would have to be multicast
routing aware, not unicast routing aware.

Because of these concerns, it makes sense to ask what benefit there
is to integrate unicast and multicast QoS signaling, and at what
level. The basic consideration is that both sets of signaling are
for traffic that share the same physical resources. Therefore, it
would be possible to have unicast and multicast QoS signaling that
interacted only indirectly, via the QoS provisioning mechanisms,
therefore having no effect on the NSIS parts of the framework. (This
could be considered close to a 'ships in the night' approach,
comparable to what is often done in multiprotocol routing).

The main limitation of this approach is that there is no scope for
joint negotiation of unicast and multicast flows. Without more
detailed scenario analysis, it is hard to tell if this is a major
concern or not. Therefore, at the current time, we have continued to

consider multicast as a second stage issue. If it becomes a concern
for NSIS (for example, if RSVP is judged insufficient for this
purpose), consideration should be given to what components of the
unicast framework could be re-used to build a complete multicast QoS
signaling solution.

4 Fundamental Framework Components

4.1 Interactions with Application Layers

The application or higher layers can be seen as a service consumer
for the services provided by the NSIS agent on behalf of the NSIS
framework (the service provider). The task of the NSIS agent is to
provide QoS to higher layers/application. The task of an application
is to provide a QoS specification to describe the Quality of Service
that the applications wants for its network traffic. Adaptive
applications may want to adapt to changes in QoS delivery which may
be the result of network failure or hand-over. Therefore, they need
feedback information from the NSIS agent in form of monitoring or
adapt requests.

A typical interaction sequence is described below. The interaction
with applications/higher layer is bi-directional as the consumer may
(re-)negotiate for QoS and the NSIS agent may issue adapt requests.

A consumer requests a certain QoS with a QoS specification that
specifies the amount of network resources or the treatment of the
application's data flow. Together with the desired QoS the consumer
provides an identifier (e.g. one possibility would be port numbers,

IP addresses, protocol IDs) for the data flow and a direction (e.g.
send, receive, send/receive).

The NSIS agent may in some cases invoke an admission control test to
check if the QoS is available. In that phase, typically signaling is
invoked. If the request is granted, the NSIS agent notifies the
consumer about the success. If the request cannot be granted, the
consumer is notified via an error message.

Once the QoS is established, the consumer may re-negotiate
dynamically with the NSIS agent for better or lower QoS at any time.
This is necessary since the traffic characteristics may change over
time (e.g. a scene break in the video may lead to a higher network
resource consumption). The NSIS agent can again invoke an admission
control test and notify the consumer about success or failure. If
QoS delivery is not within the acceptable range, the consumer may
request better QoS or adapt to the situation.

Finally, the consumer releases the established QoS and the service
provider releases the allocated resources, if any.

At any time, network resource availability may change due to
admission of new consumers, departure or re-negotiation of existing

consumers. This may lead to overload/underload situations detected
by the NSIS agent (either directly, or notified to it by upstream
peer agents - see section 5.7 for a discussion of the implications
of this for the signaling protocols). In that case, the NSIS agent
may force some or all consumers to adapt, i.e. downgrade the QoS
requirements. The consumer can then restructure its internal
operation and adapt to the new situation by sending an optional
adapt response.

There may be situations where an initial request could not be
granted by the agent, e.g. if the requested QoS is not available.
Then, the application/higher layer may at any time restart the
process of requesting QoS (possibly at a lower level). This may lead
to trial and error situations and can be avoided by introducing
notifications. In that case the application would request to be
notified by the NSIS agent, when the desired QoS is available. In
addition, the application can first start to request QoS at a low
level and ask the NSIS agent to provide notification when the full
QoS is available.

The discussion here has focused on the interaction between the NSIS
agent and application layers in the case of the QoS initiator. The
only other place in the network where the application layers are

active is in the correspondent host at the other end of the traffic
flow. Here, the interaction is much simpler: it appears that all
that is needed is a notification capability that a QoS reservation
has taken place, and this can be managed using the session
identification information that must be carried in the signaling
data anyway.

It is also mentioned in the NSIS requirements that it may be helpful
to be able to carry additional application layer messaging opaquely
within the NSIS signaling messages. This has no other impact on the
framework other than to require the existence of such a container,
which is noted in section 4.4.

4.2 Interactions with QoS Provisioning

From the IP layer perspective, QoS provisioning techniques can
implement virtual circuit style provisioning schemes like IntServ
architecture or MPLS trunks etc. Alternative solutions are based on
a hop by hop concept like the Diffserv architecture. Each
provisioning scheme relies on router specific resource allocation
mechanisms. Also, link layer specific characteristics may have major
impacts on the performance of a QoS provisioning system. For
example, in some link layers, there may be very efficient ways to
allocate physical resources for a bidirectional flow, or more
generally multiplex flows together. On the other hand wireless link
layers may suffer from channel fading etc. These effects have to be
taken into account for allowing efficient operation of the QoS
provisioning system.

For flexible integration with various link layer technologies and
router platforms it is suggested that NSIS agents interact with the
QoS provisioning system on an abstract basis. Hence NSIS agents
should not be involved in interpretation of signaling parameters to
control a QoS provisioning system. Instead a generic interface is
required to exchange parameters for various purposes between NSIS
agents and the QoS provisioning system. As stated earlier in the
document the QoS provisioning system may be co-located with NSIS
agents or realized on one or more separate platforms. Following the
principle of an open internet architecture a resource allocation
protocol is required between NSIS agents and the QoS provisioning
system.

Adaptation to specific link layer characteristics is achieved by
introduction of a link layer specific convergence sublayer for the
QoS provisioning system. Support for a variety of NSIS compliant
provisioning systems with specific link layer convergence mechanisms

is a prerequisite for successful introduction of NSIS solutions.
Accordingly routing and switching hardware need to come along with
support for an NSIS based resource allocation protocol. This way
signaling parameters together with ISP policies rules define a
specified packet treatment behavior in a routing fabric. Finally,
packet treatment is defined by architectural components like packet
classifiers, queue buffers, schedulers and traffic conditioners.
Some of these components may be directly controlled by signaling
parameters.

The QoS provisioning system should map status indications, hardware
alarms and notifications into NSIS compliant reporting, which can be
passed to NSIS agents for subsequent processing. Furthermore it is
assumed that resource monitoring is performed by the QoS
provisioning system independently. Status information that is
generated by the QoS provisioning hardware and requires mapping to
be compliant with NSIS status reporting includes resource violation
events, results of reservation requests and records about available
resources.

4.3 NSIS Signaling Protocols

The NSIS Signaling Protocol supports the actual communication of QoS
requirement information for a traffic flow/aggregate between NSIS
agents. In order to support this, a number of network related
actions must be carried out, namely:

- Peer Agent Discovery: the NSIS Agents should be able to discover
peer NSIS Agents, and optionally establish a trust relationship with
them. One NSIS Agent may discover one or more peer NSIS Agents in a
number of different domains. NSIS Agents can also exchange SLA and
the many types of policy information required for intra/inter-domain
management. This refers both to the QoS initiator discovering its
first hop QoS controller, and also to have QoS controllers discover
each other.

- Agent Selection: the NSIS Agents that will be provisioning
resources for the traffic flow or aggregate are identified, i.e.
each NSIS Agent selects the next hop NSIS Agent from the peers with
which it has established a relationship with during peer agent
discovery. Policy information associated with the user, such as
whether realtime accounting must be supported, can be exchanged if
necessary. Agent selection will not be as dynamic in the core of the
network than at the edges, and user policy will probably not
propagate too far from the edges of the network.

- Path Capability Discovery: the capabilities and resource availability of the nodes along the data path are determined. There are open issues associated with whether this occurs locally or cumulatively and on a hop-by-hop or end-to-end basis.

- Reservation Request: the actual request for resources that triggers the QoS provisioning, also carrying QoS parameters and possibly per flow/aggregate policy information.

The potential transfer of authentication, authorisation and user information places additional confidentiality and integrity security requirements on the protocol (see section 3.6). Also, the information transferred within and between domains is variable, implying a need for an easily extensible set of parameters that can be carried, possibly opaquely, by the protocol (see section 4.4). It is the responsibility of the NSIS signaling protocol to detect failure conditions along the data path, and to initiate recovery mechanisms.

A single NSIS signaling protocol solution could address the aspects outlined above, however, since a goal of NSIS is to make the actual reservation signaling as lightweight as possible, it may be desirable to address the first two actions using a separate family of protocols, which do not necessarily need to be used on a per flow basis, that are initiated at some stage before the generation of the reservation request. We therefore potentially have a two-phase approach:

1. 'Registration' phase - discovery, authentication, overall policy aspects and so on.
2. 'Reservation' phase - any signaling associated with a single flow.

The registration phase may depend strongly on other protocols that already exist, and may be entirely optional in some environments. The protocol for the reservation phase should be of minimal complexity. Indeed, note that in several significant environments, the registration phase could be omitted altogether. For example, in a cellular network, the host could know (on the basis of what link type it was using) that the first hop QoS controller was always to be reached via its default router, and that authentication and

policy aspects were managed by the network on the basis of access control rather than an explicit message exchange.

4.4 NSIS Signaling Data

The NSIS signaling protocol should be able to carry a variety of

signaling data. The majority of the information is related to QoS,
but some other parameters are required to support protocol
operation, accounting, etc.

Different parameters may have relevance in different parts of the
network. End-to-end parameters carry application QoS requirements,
and may additionally carry extra information such as
- user policy information
- session ID to identify the traffic flow
- reservation ID to identify the reservation independently from the
  flow identifier
- initiator ID to identify the requestor of the reservation. This
  necessity for this parameter is dependent on the decision made
  about some aspects of the framework. The initiator id can be
  over-ridden be a proxy so that the proxy receives feedback
  messages etc. on behalf of the real initiator.
- opaque transport of application layer signaling

Other parameters may be exchanged between peer NSIS Agents to
support, for example, inter-domain QoS and accounting. This could
include parameters such as:
- charging policy: indicating how the domain expects to be charged
  for the traffic flow
- security parameters
- aggregation parameters: may include preferred mapping of the
  aggregate exiting one domain onto an aggregate in the neighboring
  domain.

Finally, some parameters may only have significance within a QoS
domain, and should not propagate outside that domain. Such
parameters may be:
- wireless parameters: such as those outlined in [2]
- optional router technology specific parameters: to support the use
  of NSIS as a QoS provisioning mechanism.

Note: the scoping of the requirements above is intended mainly for
demonstration purposes, there will be scenarios for which the scope
of the parameters will not be the same as listed above. Framework
level decisions may also effect the data that is carried.

As mentioned previously, the information carried by the NSIS
Signaling protocol may have different scoping characteristics
depending on its type. The concept of scope of parameters plays an
important part in the framework. At a minimum, the NSIS signaling
must carry the QoS parameters concerning the QoS requested by the

application end-to-end. This could be considered as a 'global'

scope.

The end-to-end per-flow parameters may not be interpreted by every
QoS controller along the data path and may pass transparently
through domains or NSIS agents that do not maintain per-flow state.
If per-flow information is interpreted by a QoS controller, these
parameters may need to be translated into a local format to support
legacy QoS mechanisms.

End-to-end parameters may not only consist of unidirectional, per-
flow information. In some cases, additional information could be
added to support bi-directional flows or to communicate QoS
requirements for multiple flows. In the former case, information
about both directions of the traffic flow would be included to
optimize reservation establishment in segments of the network where
bi-directional flows could be supported. Further details are
available in Section 3.4.3. In the latter case, multiple QoS
parameters sets could be provided for different flows from the same
sender to the same receiver to try and optimize the installation
time of multiple reservations. The various merits of doing this
compared to simply initiating multiple reservation requests
simultaneously still need to be investigated.

In addition to transporting the end-to-end QoS parameters, NSIS
Agents have the ability to signal new parameters either across their
administrative domain with a 'local' scope, or to peer NSIS Agents
in other administrative domains with a 'one-hop inter-domain' scope.
These parameters will be derived from parameters with greater scope,
e.g. the end-to-end parameters. These parameters can also be
modified or deleted from the signaling messages, but only if the
NSIS Agent has the right to do so by either belonging to the same
administrative domain as, or by having a peering relationship with,
the NSIS Agent who inserted the parameter. The indication that an
NSIS Agent has the right to modify or delete a parameter could be
indicated by a scope ID. The QoS Initiator is at liberty to include
local scope parameters in the reservation request, for example, to
provide information regarding how it wishes its sessions to be
handled during handover if it is attached to a mobile network.

Two options for the transport of 'local' scope and 'one-hop inter-
domain' scope parameters can be identified:

     1.   A separate signaling message could be initiated to carry
the
          parameters, which would also share the same scope as the
          parameters. The end-to-end parameters could be signaled
          directly to the next NSIS Agent either at the egress point of
          the current domain, or to a peer NSIS Agent in a neighboring
          domain.

     2.   Parameters can be 'stacked' within the single NSIS

signaling

message travelling end-to-end. NSIS Agents are allowed to

append parameters only to the top of the stack. When an NSIS
Agent determines that the scope ID is not valid beyond this
point, it should remove all parameters with that scope ID.
This is illustrated in Figure 11.

```
+----------+
| Peer QoS |
|Controller|
+----------+
      |    +------------+                  +------------+
      |    |+----------+|   +----------+   |+----------+|   +----------+
    +--->|    QoS     ||-->|    QoS    |-->|    QoS     ||->| Peer QoS |
+------+ ||Controller||   |Controller|  ||controller||   |Controller|
| user | |+----------+|   +----------+   |+----------+|   +----------+
|params| |            |                 |            |
+------+ |Ingress     | +------+------+ |        Egress|
         |router      | | user |domain| |         router|
         +------------+ |params|params| +------------+
                        +------+------+
        +------+------+               +------+------+   +------+
        | user |domain|   Signaled    | user |domain|   | user |
        |params|params|   Parameters  |params|params|   |params|
        +------+------+               +------+------+   +------+

        +-----------------QoS Domain----------------+
```
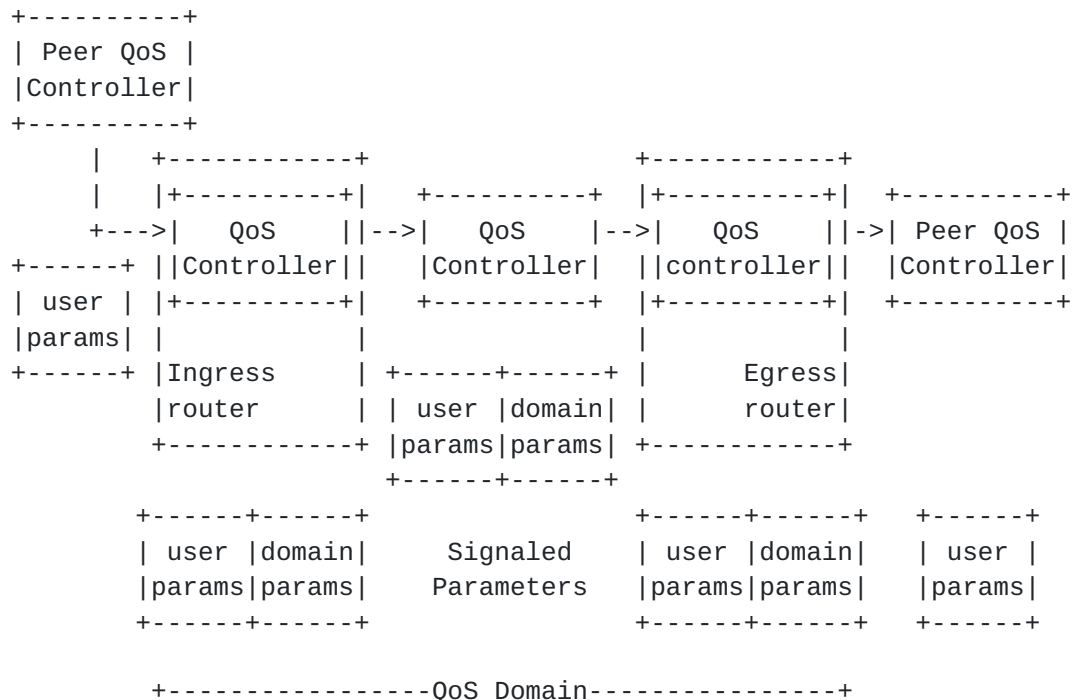
Figure 11: Domain Local Signaling Parameters

There are advantages and disadvantages associated with each
approach. The first approach minimizes the complexity of the end-
to-end signaling protocol and allows an independence of message rate
for the sets of signaling. However, some synchronization between
the signaling is required that could be quite complex to support.

The second approach increases the complexity and size of the data
carried by the signaling, but simplifies protocol interactions
within the network.

4.5 Routing Aspects

4.5.1 Implicit Routing of Signaling Packets

As described in sections 3.4 and 4.3, an NSIS agent needs to find
out about its nearest neighbors, and which of them is responsible to
handle signaling packets for which destinations. Additionally, it
needs to know whether it is able to initiate provisioning of the

path segment that is adjoining to the path segment provisioned by
the last-hop NSIS agent. (In other words it needs to know whether it
is the right QoS controller for this signaling packet.) When the
next-hop QoS controller is known, signaling packets can be addressed
directly to it.

However, there might be situations when a QoS controller doesn't
know an appropriate next-hop QoS controller. This might be either
because the adjacent domain is NSIS-unaware, or it is
overprovisioned, or even because of some failure or other, or simply
that the next hop has not yet been discovered. Along the same line,
an end host initiating NSIS signaling might not now the appropriate
QoS controller to address. Hence, there should be some simple
'boostrap' ways for finding the next NSIS agent when its address is
not yet known. We consider two basic approaches here, both relying
on the underlying routing layer to forward some signaling packet
towards the other end of the flow and have it intercepted. We call
this type of approach generically 'implicit routing' of signaling
messages.

The first possibility is to forward the signaling packet with the
signaling destination as the destination address (this requires that
the signaling destination is coded somewhere in the signaling
packet), so it can be intercepted by a router 'closer' to the next
hop QoS controller and handled by it. The natural way to implement
this with least impact on router efficiency is to use the IP Router
Alert Option [3]. The signaling message will be read and forwarded
by all routers on the path until it arrives at one which knows the
next QoS controller. This provides the discovery mechanism
(subsequent messages may turn off the IP Router Alert Option and be
addressed directly). Note that even this case still requires some
router to implement some special processing of the router alert
option, although it may be possible to re-use the processing already
defined for RSVP for this purpose.

Obviously, for this default scenario to work, at least some NSIS
agents need to be located in the data path. An end host (QoS
initiator) might just send the NSIS signaling packet with the IP
Router Alert Option set, addressed to the signaling end point. Then
it would be sensible for the end host's Access Router to be able to
forward this packet to the right QoS controller. Note that this
method is sometimes referred to as 'proxying', although it is
conceptually quite different from the proxying described in section
3.7.1. In this case here the router is acting as a proxy, forwarding
signaling packets to the 'real' QoS controller, but is doing no QoS
signaling on its own behalf.

A second possibility considers the case where QoS is required for
inbound traffic to some end host, and the far end doesn't support
any QoS (NSIS) signaling, but local bi-directional reservations are
not possible (e.g. because of asymmetric routing). This can be
achieved with a single inbound message from the remote end which is
intercepted by a specialized router at the local domain boundary.
The requirement here is that the message should use a lightweight
protocol (preferably one which did not trigger exception processing
in the rest of the network), and was 'firewall friendly' in using
well known protocols and carrying minimal data to minimize security
concerns. Some type of ICMP request/response exchange might be

appropriate here. Note that the response does not have to be
returned from the ultimate endpoint, just any router which knew it
was on the return path - so a site border firewall could
legitimately do this.

These techniques can be considered complementary. The adoption of
either of them requires a consideration of deployment considerations
for them. The first is naturally a generalisation of RSVP, while the
second provides part of the solution to the problem of QoS in multi-
homed networks having minimal impact on the rest of the network.

4.5.2 Impact of Multi-Field Routing

Classically, routing is destination based, that is, all packets from
one source with the same destination IP address usually travel along
the same route. Therefore, a flow usually will follow the same route
as the signaling packet that signaled QoS for that flow if they are
addressed to the same destination. This allows the signaling packet
to distribute QoS information locally (i.e. to the routers actually
on the flow path), where it will be needed. This is the set of
circumstances where implicit routing functions correctly.

However, routing may also be more complicated, and it is
increasingly deployed that way. It might be constraint based, or it
may be based on other fields in addition to the destination
(multifield routing), e.g. the TOS/DS field or higher layer
information. In such a case, some effort specific to the routing
applied in a particular area may be necessary for making the
signaling packet travel along the same route as the flow it is
signaling for - e.g. the signaling and flow need the identical TOS
Byte (and hence, as a possibly unwanted side effect, this signaling
packet would receive the same QoS as the signaled flow). Since the
fields or constraints on which routing is based might change in each
QoS subdomain, this might be difficult to achieve.

The problem is somewhat different when QoS controllers are not
located on every hop. They might for example be located on edge
routers only. In this case each QoS controller obviously controls
(or at least is aware of) the resources of a number of routers.
Depending on how finely-grained it controls these resources, it
needs to know the routing table, including the path taken by the
flow a particular signaling packet is signaling for. Additionally it
needs to know the boundary of its control over this path, and the
identity of the QoS controller for the next segment.

It is interesting that by incorporating the functionality that
supports interworking with multi-field and constraint-based routing,
we automatically must consider signaling traveling off the data
path. In that case, integrating QoS Controllers off the data path,
such as bandwidth brokers, does not require any extra effort.

5 Application to Generic Signaling Scenarios

5.1 Network / Proxy / Edge / End Signaling Scenarios

One important requirement for QoS signaling [1] is that the NSIS
protocol work in various scenarios end-to-end, edge-to-edge, end-to-
network etc. i.e. the location of QoS initiator and signaling end
point can be chosen freely.

The QoS initiator usually can be assumed to know the termination
point, e.g. when a reservation for aggregates is to be established
within a domain, the QoS initiator could be the ingress router, and
signaling is to be terminated at the egress router. In this case,
the signaling packet carries as its termination address (as opposed
to destination address, which typically is the address of the next-
hop QoS controller, see 4.5) the egress router. A slightly special
case is when the signaling end point is a signaling proxy. In this
case, the signaling end point addressed is thought to be a
particular end host, whereas in reality it is a signaling proxy
impersonating the end host.

It is clear that most combinations of QoS initiator and signaling
end point that can be composed of the section heading are addressed
straight forwardly by the framework. The only combinations which are
less evident are those involving the "network". This is discussed in
the next section, including defining more precisely what in this
context is meant by "network", see subsequent section.

5.2 End-to-Network Signaling and Interworking with Higher-Layer QoS
     Signaling

In some cases when NSIS signaling is to be used for reserving QoS
along a particular path, all relevant QoS parameters might already
have been exchanged by, or can be derived from, another form of
signaling, e.g. SIP /SDP [4]. If, additionally, the backbone is
overprovisioned, it is desirable to confine NSIS signaling to those
areas where it is needed, i.e. the (access) network up to the
backbone. Particularly, it would be desirable to not NSIS signal
end-to-end, but just (from both sides) end-to-network.

The problem with the above optimisation is that the use of upper-
layer end-to-end QoS exchange should in principle be transparent to
NSIS agents and the NSIS protocol. It is of course possible to build
hooks into the protocol to carry this information. However, this is
a clear case of layer-violation, would complicate the NSIS protocol
and is not recommended. Alternatively, one could use the fact that
the end host might know end-to-end QoS information has already been
exchanged. But on the other hand it wouldn't know whether the
network is overprovisioned or not.

A possible scenario for solving the problem is the employment of
Proxy NSIS agents at the edge to the overprovisioned domain, acting

as proxies for the end host on the other side. These proxies would
be configured such that they _always_ block NSIS signaling from
going any further into the network. Obviously, analogous proxies are
needed on the other side of the network. Such an approach can only
work however, when all operators involved agree (via SLAs) that
typically end-to-end exchange of QoS information has taken place
before NSIS signaling is invoked, and that in all other cases, end-
to-end QoS exchange is unnecessary. Further complication arises when
asymmetric paths through the overprovisioned domain are considered,
such that the ingress for upstream data is not the same as the
egress for downstream data.

A solution as described above is conceivable for relatively isolated
networks with a well-defined service structure, such as e.g.
commercial mobile networks.

5.3 Transparent path traversal

One of the requirements in [1](5.2.4) states that it should be
possible for NSIS signaling to traverse path segments transparently,
i.e. without interpretation by some QoS controllers. This ability
can e.g. be useful when a local QoS provisioning protocol, or
DiffServ, is employed in a subdomain. There is no need for NSIS
signaling to be interpreted in this region (However, in this case

there simply might not be any QoS controllers in this region). It is
also useful for tunnelling per-flow or per-sub-aggregate NSIS
signaling through aggregation regions.

Within the NSIS framework, such scenarios can easily be realised. As
described in 4.5, it is possible to directly address a particular
QoS controller. Thus, when the signaling is to enter the transparent
region, the adjacent controller (typically the ingress router to the
subdomain) would choose the next QoS controller beyond the
transparent region (typically the egress router) as a next-hop QoS
controller.

5.4 Use of NSIS Signaling in QoS Provisioning

The following scenario describes how QoS provisioning can be fitted
within the NSIS framework. The framework has the flexibility to
allow NSIS signaling to be used as part of the intra-domain QoS
provisioning mechanism with no additional complexity.

When NSIS is used in this way, it carries additional domain specific
parameters indicating the configuration requirements for the packet
engine, e.g. the schedulers. This sort of specific information is
not envisaged as being carried by 'normal' NSIS signaling. In
addition, the NSIS Agents in the routers pass configuration
parameters direct to the packet engine.

(Note that this is not the only type of QoS provisioning option
supported by the framework. Section 4.2 considers the implications

of using a 'black box' QoS provisioning mechanism, whose details are
opaque to the framework, which may be the more usual case.) If
desired, NSIS for QoS provisioning can be supported by the framework
in any of the following ways:

1. The NSIS protocol can be interpreted hop-by-hop along the data
path by placing an NSIS agent in every router. This would use NSIS
signaling to trigger the QoS provisioning mechanism locally; some
form of RSVP would be an appropriate protocol for this purpose,
provided RSVP can be fitted into the framework.

2. A central NSIS agent can initiate NSIS signaling to agents on
each router to configure resources directly. Used in this way,
protocols such as COPS-PR can be fitted within the framework.

```
                    +--------------+
                    |+-----------+|
        ------------->||          ||-------------->
```

```
       'Normal'<-------------|| NSIS Agent ||<--------------
     NSIS Signaling          |+------------+|
                             /| centralised |\
                            / | controller  | \
                           /  +-------------+  \
                          /         |           \
                         /          |            \
                        /       NSIS Signaling     \
                       /     for QoS Provisioning    \
                       |          |           |
                       V          V           V
       +-----router-----+  +-----router-----+  +-----router-----+
       |+-------------+|   |+-------------+|   |+-------------+|
       ||             ||   ||             ||   ||             ||
   =====||  NSIS Agent ||===||  NSIS Agent ||===||  NSIS Agent ||===
   flow |+-------------+|   |+-------------+|   |+-------------+|
   path | |             |   | |             |   | |             |
        | |configuration |  | |configuration |  | |configuration |
        | | parameters   |  | | parameters   |  | | parameters   |
        | |             |   | |             |   | |             |
        | | +----------+ |  | | +----------+ |  | | +----------+ |
        | +>|pkt engine| |  | +>|pkt engine| |  | +>|pkt engine| |
        |   +----------+ |  |   +----------+ |  |   +----------+ |
        +---------------+   +---------------+   +---------------+
```

           Figure 12: Domain-Local NSIS Signaling for QoS Provisioning

      In either case, domain specific information concerning the
      configuration of routers etc. would need to be included in the
      parameters carried by the NSIS protocol, but would be limited in
      scope to the area of the network where NSIS was being used as part
      of a provisioning mechanism. For some protocols, such as COPS-PR,
      the signaling has local scope and will not propagate outside the

      domain. For others, such as RSVP, indication must be provided to
      edge routers to indicate that the message must be terminated. This
      could be done using, for example, a scope identifier (see section
      4.4)

    5.5 Aggregation and Hierarchical Reservations

      Aggregation of per-flow signaling (or - recursively - per-
      subaggregate signaling) is a technique contributing to scalability
      of both QoS signaling and QoS provisioning. It results in
      hierarchical reservation set-up.

      Aggregation / deaggregation of flow state information can be

described by a functional component which may be located in NSIS
agents. However, this task needs to be carried out by dedicated NSIS
agents - not all NSIS agents need to implement aggregation and
deaggregation functions. Aggregation is expected to be in deployment
at all ingress routers of a particularly administrative domain or
subdomain. Therefore deaggregation would happen at an egress router,
but not necessarily within the same network domain.

An aggregation region as defined by [5] may stretch along several
domains with some level of nesting in aggregated domains. Therefore
we assume that these tasks may represent an interdomain problem.
Proper operation of aggregation mechanisms among several (sub-)
domains need to make sure that all entities are provided with
sufficient aggregation context. Correct interpretation of
aggregation context may be handled by establishment of service level
specifications (SLS's) among adjacent network domains.

Prominent existing solutions for aggregating flow state information
in a QoS signaling enabled network are namely the framework for
IntServ operation over Diffserv networks [6] and the RSVP
aggregation approach [5]. For the IntServ over Diffserv concept,
aggregation usually is performed with coarse granularity, since RSVP
style flow state information is mapped to a Diffserv transport
service class, which defines a specific per hop behaviour (PHB).
Aggregated RSVP represents a flexible solution to define the
granularity of aggregation. Flow aggregation can however also be
performed based on the NSIS signaling. In the following, all these
possibilities are considered for the NSIS framework.

5.5.1 NSIS Aggregation Techniques

An NSIS agent should be able to either initiate aggregate signaling
itself, or to interwork with network domains supporting their own
flow aggregation techniques. These two approaches are considered in
the following.

In any event, in order to realise aggregation, usually two things
are necessary:

(i) per-flow or per-subaggregate NSIS signaling must pass the
aggregation region between ingress and egress (or aggregator and
deaggregator) transparently. How this is done was handled in the
previous section on Transparent Path Traversal.

(ii) Per-aggregate signaling (NSIS-based or other QoS provisioning
signaling) must be initiated by the aggregator and terminated at the
deaggregator. The QoS information in the aggregate signaling message

is somehow derived from the total of per-flow or per-subaggregate
QoS information. However, the relation between them may be described
by a rather complicated algorithm. Furthermore, they may work on
very differently time scales. This corresponds to edge-to-edge
signaling described in 5.1.

NSIS-based Aggregation

The following figure describes a possibility of mapping this
scenario to the framework:

```
              +------------------+                  +-------------+
              |    +-------------+|                  |             |
              |    |Aggregation  ||                  |             |
              |    |algorithm    ||                  |             |
              |    |(application)||                  |             |
              |    +-------------+|                  |             |
              |        ^       |  |                  |             |
              |        |       |  |   Per-flow or    |             |
  +---------+ |+----------+    |  | per-subaggregate |+----------+ |
  |QoS      | ||QoS       |    |  |     signaling    ||QoS       | |
  |Initiator|-->|Controller|--|--|------------------->|Controller|-->
  +---------+ |+----------+    |  |                  |+----------+ |
              |              V   |                  |  ^          |
              |        +---------+|   +----------+   |  |          |
              |        |QoS      ||   |QoS       |   |  |          |
              |        |Initiator|--->|Controller|-------+         |
              |        +---------+|   +----------+   |            |
              |                   |                  |            |
              |     Aggregator    |                  |De-aggregator|
              +------------------+                  +-------------+
```
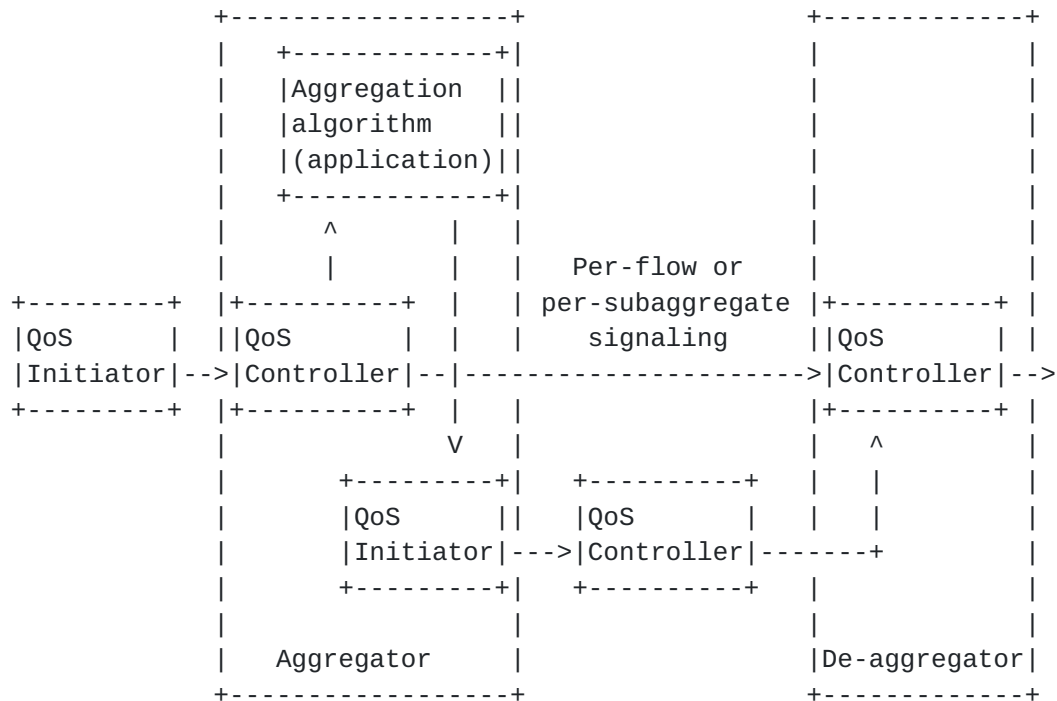
Figure 13: Aggregate Signaling

In Figure 13, the QoS controller on the Aggregator communicates the
per-flow or per-subaggregate QoS information to some kind of
aggregation algorithm. The output of this algorithm are instructions
to the QoS Initiator function on this same aggregator on how
existing aggregates are to be modified, or what new aggregates are
to be initiated. Sticking to the definition of QoS Initiator, the
aggregation algorithm it communicates with resides in the
application layer.

In an alternative scenario, the aggregation algorithm resides in a
separate administrative entity. In this case, this administrative
entity would trigger aggregate signaling in the Aggregator.

Interpreting the framework definition strictly, in this case the
administrative entity is the QoS initiator.

Non-NSIS based Aggregation

In this approach NSIS signaling data is forwarded to an adjacent
domain which supports non-NSIS compliant aggregation techniques
only, i.e. the considered domain is not capable of aggregating NSIS
signaling state information, but supplies its own QoS provisioning
signaling. In that situation the gateway NSIS agent must provide
specific mechanisms for allowing the aggregation. A function is
required to provide filter rules for classifying NSIS signaling
state information and apply aggregation mechanisms to support
interworking with the adjacent domain. This function may be co-
located with an NSIS agent as an aggregation algorithm as described
for NSIS-based Aggregation above.
Aggregation of NSIS data into Diffserv classes for example requires
an aggregation function to provide filter rules for mapping NSIS
flow state information into predefined PHB transportation classes
and coordinate DSCP codepoint marking to comply with service level
specifications (SLS) for domain interworking.

5.5.2 Aggregation Context

An administritative entity is required to decide about NSIS agent
responsibility to perform aggregation / deaggregation and to
determine the appropriate rules for performing the tasks.
Furthermore, participating entities should be identified and
entitled to carry out aggregation and deaggregation tasks. In case
of traffic trunks for example, tunneling endpoints might be
responsible for performing aggregation and deaggregation. It is
assumed that the administrative entity controls the distribution of
required information, identifies potential aggregation entities
(e.g. NSIS agents) and entitles them to execute the tasks. The term
aggregation context represents the structured information that is
required to enforce consistent execution of aggregation and
deaggregation tasks. It is distributed by the administrative entity
to entitled aggregation entities. Policy decision points (PDP's)and
network administration servers represent such administrative
entities. With appropriate extensions COPS and SNMP are envisioned
candidate protocols to carry aggregation context information and
related status and context messages.

As described above, the actual algorithm for determining whether
changes to aggregates or new aggregates are necessary might reside
either in the administrative entity, or in the aggregator.

5.6 Operation over Addressing and Other Boundaries

The ideal operational paradigm for the Internet is that end to end
addressing transparency is preserved; in other words, the IP
addresses in packets are unchanged between the sending and receiving
end hosts. This is recognised as an optimal situation from the point
of view of network simplicity and resilience [7].

Unfortunately, this paradigm is broken in today's Internet for
several reasons. The major current reason is NAT [8], although
address translation techniques are also one method considered for
IPv4-IPv6 transition [9]. In addition, some recent network layer
protocol developments change or add to the addressing capabilities
of the network layer header, including the HAO of Mobile IPv6 [10]
and HIP [11]. Finally, tunneling mechanisms or application layer
gateways can also be considered as falling into this category, but
are less of a concern here: tunnels can be modeled (and are often
implemented) relatively transparently as virtual links, while NSIS
would see an application layer gateway as a signaling endpoint in
its own right, not needing special consideration.

The implications of addressing non-transparency for NSIS are two-
fold.

Firstly, the signaling messages themselves must be capable of
passing through addressing boundaries and might have to use (or be
able to exploit) these more recent addressing approaches. This
requires consideration of the way in which possible NSIS signaling
protocols can be treated in this way. This is particularly
important, since it is likely that the NSIS signaling data will
contain addressing information about signaling endpoints which will
also need to be translated consistently.

Secondly, the NSIS signaling data will contain packet classification
information that is used to identify the user packets for which QoS
is being requested. On the assumption that both the user packets and
signaling packets have to be treated or interpreted consistently,
the treatment (e.g. translation) applied to the user packets must be
matched by updates to the signaling data carried in the NSIS
packets. This may have implications for the security treatment
applied to these packets.

In our framework, it seems that the various boundary cases should be
treated individually. Classic NAT is stateful and typically has to
run over a single physical device. In this case, it is natural to
locate a Proxy NSIS agent at the NAT device. In order to do
consistent message processing on the user and signaling data, this
will have to be closely integrated with the NAT functionality,
probably according to a proprietary behavior specification; however,
the rest of the network should continue to be unaware of its

presence. On the other hand, SIIT being stateless, it should ideally
also be possible to translate NSIS messages statelessly, which would
require extensions to the current definition, comparable to those
needed to handle IP options or extension headers. This requires

defining a NSIS data format which could have semantically identical
v4 and v6 forms, but NSIS proxies should not be enforced for this
case.

The situation of modified network layer identifiers may require
extended address format capabilities for NSIS signaling data to
include the new formats, and also a precise definition of the
semantics of addressing information elements (e.g. whether or not an
address in a classifier refers to the HAO if present).

It is certain that these addressing considerations will need to be
re-evaluated when a more concrete NSIS solution is ready to be
considered. At that stage, it might even be questioned whether
support for all possible network scenarios (e.g. NAT) needs to be
maintained.

5.7 Support for Adaptive Applications

Adaptive applications require feedback about the ongoing resource
availability in the network, or the occurrence of QoS violations.
The actual frequency and detail required in the feedback messages
any vary depending on the scenario, e.g. mobile vs. fixed network.

In the following discussion, it is assumed that the feedback
information should be passed back to the initiator of the session,
and then on to the application to whom the session belongs. The
application can choose to take adaptive action if required, which
may result in a re-negotiation of resources along the data path.

There are a number of ways that QoS feedback information to the
Initiator.

1.   Information can be sent direct to the QoS initiator: the NSIS
   Agents maintain the address of the initiator to allow the QoS
   violation notification to be signaled directly. When a proxy is
   present in the path and has modified the initiator identifier to
   identify itself, the message will be sent direct to the proxy
   instead of the real initiator. There may be trust issues
   associated with this approach concerning the fact that the
   initiator will be receiving information on which it is basing re-
   negotiation decisions from an untrusted node. However, the
   signaling does not have to be processed by all intermediate NSIS
   Agents.

2.   Information can be sent hop-by-hop back towards the initiator: the
   feedback information is signaled backwards between NSIS Agents on
   the data path until it reaches the QoS Initiator. Each NSIS Agent
   must maintain previous hop information and process the messages in
   part to forward the messages to the correct agent. However, the
   Initiator will receive information from a peer NSIS Agent with
   whom it has established a trust relationship.

3.   Information can be sent downstream in the data path to destination
   domain and reflected back to the Initiator: the violation
   information is simply sent along the signaling path to the
   destination node that then reflects this information back to the
   receiver. No state information is required within the NSIS
   Agents, but the propagation time for the signaling to reach the
   Initiator is increased

Only the entities that know about violations need to support
violation reporting. When the QoS for a flow that is being
transported as an aggregate is violated, it is a local matter as to
whether this information is propagated, and implies some per-flow
knowledge about the flows in the aggregate. This is to allow the
information to be propagated to the correct Initiator and is may be
undesirable for scalability reasons.

The decision about how feedback to applications is supported by the
framework will impact the QoS signaling data that has to be carried
by the protocol, and also the state information maintained by NSIS
Agents depending on the chosen option.

6 Applicability of Other QoS Frameworks and Protocols

6.1 Incremental Deployment in an NSIS-Unaware Internet

Introduction of NSIS compliant network components most likely will
follow a step-by-step process where deployment in few networks has
to be considered at the beginning. A three stage introduction
process is described assuming few NSIS compliant domains at the
beginning. If NSIS technology could gain more acceptance after the
introductory phase a heterogeneous infrastructure with NSIS capable
domains and non NSIS capable domains intermixed arbitrarily is
assumed.

For the non NSIS compliant domains in operation two cases can be
thought of.

a. Heterogeneous QoS Signaling Solutions

A domain offers QoS provision mechanisms but QoS signaling is not
compliant with NSIS protocols, e.g. an ISP's proprietary solution is
in operation. Application of appropriate interworking mechanisms
have to be considered then. NSIS signaling data traversing a non
NSIS compliant domain should not be subject to loss or delay caused
by low priority handling. To avoid loss, NSIS signaling information
either may be encapsulated or may be forwarded transparently in non
NSIS aware regions. Additionally non NSIS compliant domains have to
provide mechanisms to forward NSIS signaling information with the
required reliability and priority, which has to be defined by
service level specifications (SLS's).

b. Best Effort Forwarding Paradigm

In this case end to end signaling and QoS provision is broken by non
QoS aware domain(s). Consequently delivery of end to end services is
not possible with satisfactory service level guarantees.

6.1.1 Step 1: NSIS compliant Islands

In this situation deployment of NSIS technology is in its initial
phase. The technology can be used for usage of intradomain QoS
sensitive services. Then NSIS signaling can be applied e.g. for
accessing services from a local host. This approach can be
associated with the walled garden model [12]. A more complex
scenario deals with application flows that may extend along non NSIS
aware domains. A potential option for signaling between remote NSIS
agents can be accomplished by using a leased line service for
bridging the NSIS islands. Transport characteristics of the leased
line service have to be taken into account when deciding about
admission of an NSIS resource reservation request. Therefore gateway
node interfacing with the leased line service are responsible to
perform AAA functions on single flows and flow aggregates, that
require resources for bridging to a remote NSIS domain.

6.1.2 Step 2: Heterogeneous Infrastructure

With the incremental deployment of further NSIS technology a mixed
infrastructure is assumed comprising NSIS compliant and non NSIS
compliant domains. Some end to end NSIS signaling may be
accomplished without break in between even if several domains have
to be crossed. Still there remains some potential for unreliable
transport and signaling especially when the QoS enabled path extends
along several domains. This problem can be reduced by introducing
"trusted" NSIS regions. This concept assumes 'strong' SLA's between
several ISP's, extending a trusted NSIS region among several

domains. Further enhancement can be achieved by an approach, which
relies on an extended record route mechanism. Detection of next hop
NSIS agent requires the tracking of all NSIS agents along and end to
end path. The NSIS agent record can be used by all NSIS agent to
check whether the next hop agent is available in a network domain
with established SLA's for proper interworking. With the proposed
concepts an ISP may be aware with reasonable high reliability which
destination network will be reachable through NSIS signaling without
signaling and service break in between.

Destination networks may not be reachable without leaving an NSIS
compliant region. Reliable transport of NSIS signaling with
appropriate QoS provision could be enforced if there is no more than
one non-compliant NSIS domain between NSIS compliant domains with
appropriate SLA's in place.

In the context of heterogeneous infrastructure we consider a
scenario where a host or network node wants to act as QoS initiator
for NSIS signaling without actually being capable of doing so. Proxy

NSIS agents described in this document may take over the QoS
initiating part in this situation. ISP's could benefit from this
scenario by offering QoS enabled services to a larger number of
customers.

6.1.3 Step 3: Widespread deployment of NSIS

A widespread deployment of NSIS compliant domains represents an
optimistic case though it may be considered for specific countries
or federal states. Compared to step 2 of deployment there are no
further technical issues to consider here.

6.2 Basic Diffserv

DiffServ domains can be deployed in a variety of ways. Particularly,
they may work with or without admission control. If admission
control is included, it may work based on a variety of data. In this
context, DiffServ networks with the admission control based on the
flow characteristics and the flow's QoS requirement are interesting.
This information would be transferred by NSIS signaling. The
admission control typically is handled by edge routers or by
bandwidth brokers. Consequently, QoS controllers would be located on
the corresponding entity(entities), as applicable for a particular
network. The next QoS controller would be either the egress router,
or the bandwidth broker of the next domain. Inside the DiffServ
domain, usually no QoS controller is necessary.

The responsibility of the QoS controller also includes providing for

proper (re-)marking of data packets at the ingress router to the
DiffServ domain. RODA [16]defines a signaling and resource
reservation protocol for DiffServ domains. It can be regarded as a
QoS provisioning signaling protocol, outside but compatible with the
NSIS framework.

6.3 Basic Intserv

In an IntServ domain, each router is in charge of its own resources
and needs QoS signaling information. Hence, each router in fact is a
QoS controller.

If RSVP will be used as the / one NSIS signaling protocol, this
scenario fits in seamlessly. If on the other hand, NSIS decides on
using another signaling protocol, we need interworking of the NSIS
protocol and RSVP at the edge router of the IntServ domain. In this
case, the edge routers are QoS Controllers only, and RSVP is QoS
provisioning signaling.

6.4 RMD

Resource Management in Diffserv (RMD) is designed for edge-to-edge
resource reservation in a Differentiated Services domain [13]. The
RMD framework defines two architectural concepts:

      - the Per Hop Reservation (PHR)
      - the Per Domain Reservation (PDR)

Individual candidate resource reservation protocols are envisioned
for both concepts. A PHR protocol is used within a Diffserv domain
on a per-hop basis to augment the Diffserv Per Hop Behavior (PHB)
with resource reservation. It is implemented in all nodes in a
Diffserv domain. On the other hand, a PDR protocol manages the
resource reservation per Diffserv domain, relying on installed PHR
resource status in all nodes. The PDR is only implemented at the
boundary of the domain(at the edge nodes).

The RMD framework uses the Diffserv classes Expedited Forwarding(EF)
[14] and Assured Forwarding (AF) [15] as QoS classes. It is implies
that any network supporting the RMD framework is able to classify,
mark, police and schedule the traffic accordingly. A signaling
protocol is proposed as well for allocating resources hop by hop in
a DiffServ capable domain. An instantiation of a PHR protocol is
named RODA [16] (Resource Management in Diffserv On DemAnd), which
has been proposed recently. In contrast, the PDR can be supported
either by a new protocol or (one or more) existing protocols.
Examples of such existing protocols can be the Resource Reservation

Protocol(RSVP) [17], RSVP aggregation [5] etc.

For PHR the RMD Framework currently specifies two different groups:

The Reservation-Based PHR group

In this PHR group, each node in the communication path from ingress node to egress node keeps only one reservation state per PHB. The reservation is done in terms of resource units, which may be based on a single parameter, such as bandwidth, or on more sophisticated parameters. These resources are requested dynamically per PHB (i.e., per DSCP) and reserved on demand on all nodes in the communication path from an ingress node to an egress node. Furthermore, this PHR group has to maintain a threshold for each PHB that specifies the maximum number of reservable resource units.

The Measurement-based Admission Control (MBAC) PHR group

This PHR group is used to check the availability of resources before flows are admitted. That is, measurements are done on the real average traffic (user) data load. However, the measurement based PHR uses two states. One state per PHB that stores the measured user traffic load associated to the PHB and another state per PHB that stores the maximum allowable traffic load per PHB.

The RMD framework supports signaling to achieve load sharing among several paths. Interior nodes are able to observe when a load sharing situation occurs and know the number of multiple equal cost paths that a routing protocol will use to provide the load sharing

principle. Subsequently, for each arrived PHR message which is affected by the load sharing principle, the interior node will be able to create the appropriate number of PHR messages of identical type as the original one. "Appropriate" here means the number of paths participating in the load split.

In the following an assessment is done on fitting the RMD framework proposal with NSIS requirements [1]. The goal is to apply and reuse RMD concepts for NSIS based solutions where possible.

The RMD framework proposes a scalable solution and fits with the ideas of lightweight signaling. In particular, little state information is required for hop by hop signaling in the Diffserv domain. A major benefit is the possible reuse of existing signaling protocols (RSVP etc.) and QoS provisioning mechanisms (Diffserv). Independence of signaling and provisioning paradigm is achieved by the RMD framework. PDR allows hiding the internal structure of a QoS domain from end-nodes and from other networks. A useful feature not

explicitly part of the NSIS requirements is support for load
balancing.

RMD focus is on intradomain but does provide hooks for domain
interworking, and requirements that typically come from the end to
end or host to edge signaling cases are emphasised less. Resilience
aspects are not explicitly mentioned. RMD doesn't consider
interworking with QoS provisioning mechanisms other than Diffserv.
Processed state information for each PHB is restricted to a maximum
of two states, which are measured traffic load and maximum allowable
traffic load by name. Mobility support is not considered in the RMD
framework.

Though the RMD framework does not cover some crucial requirements to
be met by an NSIS solution it seems to fit into our overall NSIS
framework. RMD can be considered to provide a sub-solution for our
NSIS framework with beneficial extensions to an Diffserv network.

6.5   MPLS

In MPLS, the ingress router to a domain sets up LSPs e.g. edge-to-
edge. Particular LSPs may be set up for a particular QoS class. For
the ingress router it is necessary to know the QoS requirements of
an incoming flow in order to map it to the appropriate LSP. Hence,
in MPLS domains, ingress routers are QoS controllers. The next QoS
controller is on the egress of the LSP of the signaled flow. Note
that signaling inside the MPLS domain, e.g. for setting up LSPs, is
not in any way changed by NSIS signaling but can regarded as a form
of QoS signaling provisioning.

6.6   Bandwidth Broker

Bandwidth brokers are QoS controllers outside the data path. As
illustrated in section 4.5 on Routing Aspects, such QoS controllers

naturally integrate into the NSIS framework, as signalling cannot be
assumed to follow the data path anyways as soon as other than
destination-based routing is employed. Examples of bandwidth brokers
integration in NSIS signalling are given in the following.

The basic scenario is a bandwidth broker being the only QoS
controller in a domain. It makes its existence known to QoS
controllers in neighbouring domains, and directly receives NSIS
signalling from them. For provisioning QoS, the bandwidth broker may
signal to all routers in the domain, as illustrated in section 5.4.
Alternatively, the bandwidth broker may just monitor router loads in
the domain, and base admission control decisions on monitoring
results.

In a more involved scenario, additional QoS controllers are located
on edge routers. These edge routers are responsible for interworking
with outside domains. NSIS signaling in this case arrives at edge
routes first, which relay it (without any QoS provisioning action)
to the bandwidth broker(s). The bandwidth broker addresses the NSIS
signaling to the next-hop QoS controller, the egress router. It
needs to be investigated whether such a scenario is more robust,
since edge routers are in the data path and can always be discovered
by the default routing of NSIS signaling packets with Router Alert
Option on (cf. section 4.5). Additionally, it facilitates topology
hiding (at least the existence of the bandwidth broker is hidden)
required by some operators.

Of course, a bandwidth broker can also act as a QoS Initiator, for
example for setting up inter-domain aggregate reservations.

7 Possible NSIS Signaling Protocols

7.1 RSVP and its Extensions

RSVP has been extended in a number of directions, e.g., aggregation,
DiffServ, tunnel, policy, proxy, and mobility support. Together with
these extensions, RSVP serves for broader applicability of signaling
beyond the IntServ model only.

RSVPv1 provides an ability to communicate the application's
requirements to network elements along the path and to convey QoS
management information between network elements and the application.

RFC 2746 [18] extends RSVP to provide signaling support in IP
tunnels. The tunnel RSVP session views the two tunnel endpoints as
two end hosts with a unicast FF style reservation in between; the
e2e RSVP session views the tunnel as a single link on the path
between the source(s) and destination(s). Inside the tunnel, the
endpoint uses a new SESSION_ASSOC object in the Path message to map
between an e2e session and a tunnel session, and uses a tunnel Resv
to reserve resources.

RFC 2749 [19] supports COPS policy services in RSVP environments.
All objects carried in RSVP messages received by the PEP (if not
matching its cache) are encapsulated in a Client Specific
Information (CSI) object and sent to the PDP. COPS provides the PDP
with flexible policy controls upon receipt of the CSI, making
decision per reservation flow.

With RFC 3175 [5], individual flows can be aggregated in an

aggregated region. Aggregated Path/Resv messages are used in the
region to setup and maintain aggregation sessions. According to
DiffServ, DSCPs are used to identify the traffic belong to the
correspondent aggregation. The DSCP can be specified by a DCLASS
object per [20]. One or more Diff-Serv PHBs are used to offer the
required forwarding treatment to this traffic. The corresponding
PHB is determined by the Intserv to Diffserv mapping rules
recommended by [6].

With the addition of [22], RSVP is also used to proxy the RSVP
messages intended for one of the communication ends in an
intermediate node. The node running the rsvp-proxy takes the
responsibility of the other communication end. A RSVP receiver
proxy may generate a Resv message upon receipt of a Path message. A
RSVP sender proxy can initiates a proxy Path message upon some
policy-based triggers.

With various proposed mobility extensions, RSVP is used for
signaling application's requirements when hosts are mobile. For
examle, Shen et al [21] suggest a way to use the mobile node's home
address to identify the source or destination of a RSVP flow. In
[23] a "mobile-proxy" is put at the edge of the access domain on
behalf of RSVP messages: inside and outside the access domain, LCoA
(local CoA) and RCoA (RCoA) of the mobile node will be used to
identify the same RSVP session. The mobile-proxy will either change
the session information in Path/Resv message accordingly and forward
it (when it is an inter-domain handoff), or generate a Path toward
the mobile node / respond with a Resv message upon receipt of a
Path message from the mobile node (when it's an intra-domain
handoff).

7.2 RSVP ultra-lite

Per RSVPv1 [17], a unicast session is defined as session which has
one or multiple senders, one receiver. Multicast as well as multiple
senders unicast brings much difficulty e.g., in state merging and
maintenance. However there are many cases when a session has one
sender, one receiver. In this section a possible functionality set
of the simplified RSVP for one sender, one receiver unicast is
discussed. We use the abbreviation "RSVPvs" in the following
discussion.

- The session identification in RSVPvs. In RSVPv1, a SCOPE object
carries an explicit list of sender hosts, and (DestAddress,
ProtocolID[, DstPort]) is used to identify a session. A fileterspec,
together with the session information, defines the set of data

packets to receive the QoS specified in a flowspec and is used to set parameters in the packet classifier function. In RSVPvs, there will be only one FF style for reservation, hence the sender address with a port number may be added in the session information; the SENDER_TEMPLATE, FILTER_SPEC, STYLE and SCOPE objects can be omitted. Alternatively it is also possible to keep the same session identification and use the (FILTER_SPEC, session) pair to classify the packets belongs to a session.

- Message processing. The simplification as above may apply to all messages in RSVPvs. As a result of the one-to-one feature of sessions, in RSVPvs there will be no necessity for FLOWSPEC merging (or change) - and later, complexity in the state maintenance - when a session is reserved (or torn down). Path messages are reduced to transmit SENDER_TSPEC (and optional ADSPEC to find the predicted e2e QoS) and Resv messages are still used to reserve resources but no FLOWSPEC merging is required. It is possible for RSVPvs to simplify the sender-initiated reservation: senders are usually not able to predict the end-to-end service, thus it is possible to use only a Resv message in the direction which he/she wish to send. In this case, a SENDER_TSPEC will be required to be carried in the Resv message. An even simpler possibility is, a Resv message initiated by either the sender or the receiver, carrying a SENDER_TSPEC, a FLOWSPEC, and optionally an ADSPEC, may be sufficient for the NSIS signaling, and reduces to a pure "one-pass" approach.

- Application for RSVP aggregation and other extensions. In RFC 3175, it is suggested that reservation state is per multicast session inside the aggregation region, and multicast flows heterogeneous reservations increase the difficulty in RSVP aggregation. For RSVP proxy [22] and various proposed mobility extensions [23], it is extremely difficult to manage multicast sessions which are not supported for many applications. In contrast, RSVPvs allows a uniform method for flow aggregation in the aggregate region as well as in processing in proxy scenarios; thus largely decreases the overhead for multicast sessions.

7.3 In-band QoS Signaling

In some scenarios, it may be advantageous to include QoS related data within existing, non-QoS specific signaling messages such as routing signaling or mobility related signaling, or even within ordinary IP data packets.

As an example, if we consider a situation where a mobile node is attached to an access network, when the mobile changes its point of

attachment i.e. handsover, the routing in the network must be
updated so that traffic is sent to the mobile's new location. At the
same time, any QoS state stored along the old path must also be
moved to the new path. This process can be optimized by including
QoS information about a mobile's traffic flows in the routing update
messages, so that re-routing and resource allocation occur
simultaneously. INSIGNIA [24] proposes such an in-band signaling
approach that explicitly carries control information in the IP
packet header (so-called INSIGNIA IP option). This approach allows
one-pass check for the required resources along the path toward the
destination node.

A similar concept has been suggested for QoS support in Mobile Ipv6
[25]. The basic idea is to define a new hop-by-hop header containing
a so-called QoS object. The QoS object codes the QoS desired by a
flow and is attached to the mobile IP Binding Update. Each node
along the path reads the QoS object (acts as a QoS controller) and
provides QoS accordingly, if it has the resources to do so. This
way, QoS along the new path after handover is provided by one-pass
signaling simultaneously with the Binding Update. The QoS Initiator
may be either the end host, or the Access Router who received the
information on QoS desired by the end host e.g. by a SEAMOBY
mechanism, or the Correspondent Node who received a Binding
Update. The drawback of this one-pass signaling is of course that
the QoS Initiator doesn't receive any feedback on whether the QoS
desired is at all available along the new path.

This problem is solved in [26]. Here, the Binding Update is
conditionalized upon receiving the QoS desired along the new path.
If a router cannot provide adequate QoS it returns the Binding
Update as invalid. Furthermore, the QoS object does not need to be
read by every host, but could be obtained by the new access router
from the old access router e.g., by way of Context-Transfer [27].

Lightweight QoS signaling via QoS object has so far only been
investigated for providing QoS after hand-off. Whether it is a
viable option for QoS signaling in general needs further study, but
the mechanism is flexible enough to operate in many different
situations, not only those related to mobility scenarios.

8 Possible NSIS QoS Class Descriptors

It may be possible to re-use existing QoS class descriptors for use
as QoS signaling data options. Re-use of parameters provides
simplified interworking with existing QoS mechanisms such as IntServ
or COPS based networks, and also stops 're-invention of the wheel'

with existing work taken as a starting point for any extensions.

The following section provides a brief introduction to the possible descriptors that could be re-used in the framework, and a brief overview of where they might be useful. However, an in-depth analysis will need to be performed at a later stage.

- DiffServ PHBs/PDBs: the per-hop behavior (PHB) is a description of the externally observable forwarding behavior of a DiffServ node applied to a particular DiffServ behavior aggregate [28]. The Per Domain Behavior (PDB) describes the behavior that should be experienced by a particular set of packets as they cross the DiffServ domain by specifying how the forwarding path components work together [29]. These cannot be used to accurately describe the application requirements end-to-end, but may prove useful for aggregation purposes both within a domain, and potentially between domains if the appropriate administrative relationship is in place.

- UMTS bearer classes [30]: provides four distinct classes of parameter for different types of application traffic, conversational, streaming, interactive, and background. These parameters are fairly abstract and could be re-used to provide end-to-end application requirement descriptions.

- IntServ Tspec/Rspec [31, 32]: describes a traffic flow in terms of token bucket and other parameters such as the maximum datagram size and an indication of the desired service (controlled load or guaranteed service). The parameters exchanged in the Tspec and Rspec have been the result of intensive analysis work. These parameters are used to configure the schedulers within the routers. The IntServ Tspec/Rspec is appropriate for QoS provisioning since it indicates parameters for router configuration, but may also be suitable for end-to-end if it is flexible enough to meet all requirements.

  Additions to IntServ parameters have been proposed in [2] to improve the operation of Integrated Services in environments with wireless links.

Translation between domains using different QoS parameters can be provided by local 'stacking' of parameters, but these must be derived from a well defined and understood set of parameters that are transferred end-to-end. Since the QoS class descriptors are

primarily for end-to-end use, excessive flexibility and support for multiple options may lead to inter-operability problems.

9 Security Considerations

   This section tries to identify and describe possible security issues
   related to a QoS framework. We would like to focus on the security
   issues of the QoS signaling protocol and not to concentrate too much
   on the protocols supporting the QoS signaling and provisioning for
   example COPS, AAA, LDAP etc. For a security architecture it is
   useful to distinguish between user-related security and network
   domain security. The first part of this section concentrates on the
   user-related security, whereby an end-node (i.e. a user) issues a
   quality of service request that first reaches the network at the
   first hop router. That is, this section deals with the first stage
   of the QoS signaling. The next section discusses network-to-network
   and intra-domain signaling security. This addresses the network
   domain security part of the security architecture. Finally, the last
   section addresses end-to-end signaling issues which are also user-
   security related but usually not addressed in the context of QoS
   signaling. We, however, think that it is worth investigating issues
   concerning end-to-end security within a QoS framework.

9.1 End-Node to Network Signaling

   In order to authenticate to the network the user has to know (at
   least for the case of symmetric cryptography) to which entity he has
   to authenticate since he has to select the appropriate security
   association and the corresponding key. To support the generic
   roaming case there must be a way for the mobile node to learn the
   identity of this node. This can be the first hop router or another
   node in the network. If these two entities already share a security
   association, then an initial key management protocol was executed
   because manual key distribution is practically not feasible in a
   mobile environment. This established security association can be
   created using protocols like PANA or could be the result of a AAA
   protocol exchange that took place at the time the mobile node had to
   authenticate to the network. Since the mobile node is likely to be
   attached to a network different to his home network cross-realm
   authentication may very likely be required. However there is no need
   to solely create the QoS required security association with the help
   of a separate run of a key management protocol supporting cross-
   realm authentication. It is sufficient to derive the QoS required
   security association based on some available session key distributed
   by some other protocol. Hence the actual authentication and key
   distribution procedure executed with the visited network may already
   be finished at the time the QoS signaling is started. However, from
   this there is still the need to derive QoS related security
   association. Furthermore, it may be required to transfer the key to
   the appropriate entity in the network with which the security
   association is required. For the discovery of this entity several
   mechanisms could be used that are already used in other protocols.

The mobile node could learn this information via Router
Advertisements, address the entity with an anycast address, the
mobile node could retrieve information from a DNS server or query a
DHCP server. Additionally service location protocols could be used.
Note however that there is a strong need to execute this task
securely and as fast as possible since a real-time service would
suffer from performance degradation. The mechanism which results in
the lowest latency is probably to pre-configure a particular entity
within the network whose identity is already constructed in a
deterministic way. The first-hop router would be an example for such
a node since its IP address is supposed to be known to the mobile
node. The Kerberos principal name could be derived from the IP
address with a specific service prefix and the realm name would then
be taken from the Router Advertisement. Instead of having a pre-
shared security association of some kind it would be possible to
make use of Kerberos to dynamically derive one. Kerberos has the
main advantage that it obsoletes a separate, independent, previously
executed key management protocol since the purpose of the ticket
inside the AP_REQ request is to provide the session key to the other
party and to provide authentication. Note that the authentication
and the guarantee of freshness provided with the Kerberos
Authenticator can also be replaced by some other mechanisms similar
to those provided by the Integrity Object in RSVP. Usually the
Kerberos message exchange required to request the session ticket for
a particular principal is outside the scope of a QoS signaling
protocol but may require several roundtrips because of a cross-realm
authentication procedure. The total number of messages that need to
be transmitted depend on the type of inter-realm navigation that is
required to navigate from the home domain possibly via several
intermediate domains to the visited domain. This "navigation"
involves the request for cross-realm ticket granting tickets and the
corresponding response. Note that PKCROSS allows reducing the number
of round-trips for a cross-realm navigation and PKINIT allows a user
to use public key based mechanisms to request the initial ticket
granting ticket. PKINIT therefore also allows problems related to
dictionary attacks to be defended. The usage of passwords and
Kerberos has often caused concerns and some papers have been
published to address these issues and how to prevent them.

If we speak about authentication then we also have to answer the
question of what identities to use for identification. For the QoS
Initiator possible identification entities are the user, the host on
which the user resides and an entire network as described in more
detail in the next section. The type of identity used heavily
depends on the authentication protocol. Different identifiers are
used in Kerberos, AAA or as subject names inside certificates.
Additionally to the identity used by the user a host identity (i.e.
IP address) must be transmitted to the network and updated in case

of a movement since the accounting rules and firewall filters at the
edge of the network must reflect the fact that a particular user is
allowed to transmit data traffic that receives the desired QoS
treatment. Kerberos additionally complicates the authentication by

the fact that the mobile node needs to learn the principal name of a
node for which he has to request the session ticket.

There may also be a concern about the user identity confidentiality.
Hence it might be favorable to use a temporal identity for
subsequent sessions after a successful authentication. Note that the
question of user identity confidentiality might raise other
difficult questions that need to be answered. The initial
authentication protocol must provide some means of identity
protection which is not the case for current protocols like
Kerberos, Radius, Diameter etc. If the underlying authentication
protocol reveals the user identity then it might be difficult to
provide some useful protection at the QoS signaling level.
Furthermore it must be clear against whom these user identities need
to be protected i.e. what threat scenarios are applicable. For a
service provided at the visited network it might not be necessary to
know the real name of the user issuing the QoS request - a pseudonym
might be sufficient. The name must however provide enough
information to find and associate the corresponding home realm with
the user if traditional accounting with AAA protocols is used. Hence
there is a strong relationship with accounting and payment at this
point. Furthermore it might be necessary to protect the transmitted
identity against eavesdropper on the link between the mobile node
and for example the first hop router. These issues require a careful
investigation since there are strong interdependencies with other
protocols and hence this issue might be difficult to solve in
general.

To provide mutual authentication a second step has to be executed
after the user is authenticated to the network. In most cases it is
assumed that the initial key management protocol already provided
enough evidence to authenticate both parties mutually against each
other to prevent false base station like attacks. However note that
care must be taken if such an assumption does not hold. The fact
that most QoS signaling protocols use a message from the imitator to
the network and vice versa allows the initiator to be assured that
the other party knows the session key.

The authorization step executed after a successful authentication of
a QoS request may result in a token to be transmitted along the QoS
signaling path within the same administrative domain to provide
subsequent nodes with enough information to do admission control.

The authorization step usually requires a router that receives a
properly secured resource request to trigger other protocols to
retrieve the desired information. Such a token may have already been
handed over to the QoS initiator as part of a protocol executed
preceding the QoS signaling (e.g. a SIP signaling protocol run) and
used to provide a pointer to an already executed authorization step.

The issue of public key based authentication of the mobile node to
the visited network is somewhat more complex than the symmetric
case. If we assume that accounting (and the payment of the consumed

resources) is the main goal of the authentication of the mobile node
(i.e. of the user) then there must be a strong interrelationship
between a payment protocol and the public key based authentication
protocol. Otherwise it would be quite difficult to accomplish the
accounting task. This requires some investigations for alternative
means of access authentication and payment. The combination between
a public key based authentication and the traditional AAA
infrastructure may lead to the conclusion that two different
authentication mechanisms are used together although a single one
might be sufficient. Furthermore it should not be underestimated
that public key based authentication involves substantially more
computational operations than symmetric cryptography and then there
are issues related to certificate path validation, certificate
revocation, etc. This may imply that a digital signature used to
protect every QoS signaling message might not be an advantage in the
context of "lightweight" QoS signaling. If a digital signature is
used with every QoS signaling message then denial of service attacks
are more easy to launch since the verification of the signature
requires more performance demanding cryptographic computations than
the verification of keyed message digest.

Public key based mechanisms may therefore be more interesting for
initial key distribution (key transport or key agreement) where
several roundtrips are more likely to be accepted. The resulting
session key could then be used to protect subsequent QoS signaling
messages. User identity confidentiality however can be solved more
efficient using public key based techniques compared to symmetric
alternatives. The question of discovering the identity to which the
user or the user's host has to authenticate is also less difficult.

The main threats that must be prevented in this context are the
following. A malicious user must not be able to request resources on
behalf of another user. We can prevent this threat by requiring
every user to authenticate and to issue only authenticated,
integrity and replay protected QoS signaling messages. A subsequent
step of authorization ensures that each user can only request the

amount of resource he is entitled to. This step is executed as part
of the policy based admission control procedure. If every user
requesting resources is authenticated and the request is properly
integrity and replay protected then there is no danger of denial of
service attacks against the access network whereby an
(unauthenticated) adversary massively requests resources. Special
cases during the reservation setup like the RSVP Killer Reservation
are no security issue by itself. A different threat, which was
previously mentioned, is related to user identity confidentiality
whereby an adversary learns the user identity and the corresponding
QoS request issued. It is an open point of discussion whether this
threat should be addressed or not.

If we assume a preceding authentication and key agreement phase
before the QoS signaling is started then we have the following
advantages. First, the user can easily authenticate himself using

various types of protocols including public key based protocols. It
may therefore be possible to run any "administrative tasks" which
may be more time-consuming in advance to the actual QoS signaling.
Second, such an initial phase allows the network and the user to
establish the required QoS security association. Furthermore the
identities used within the initial authentication step may be
different to those used during the QoS signaling. For the benefit of
user identity confidentiality and shorter messages it would be
beneficial to use a short identifier with local meaning (independent
of the IP address of the user's host) to select security
associations and to identity the registered user. These messages
have typically lower time constraints and the message exchange may
involve several roundtrips. Please note that such an initial
authentication and key agreement phase does not necessarily need to
be part of the QoS protocol and may also be optional.

 9.2 Network to Network

As soon as the QoS signaling message is authenticated and the
request is authorized a token may be added to allow subsequent
routers participating in the QoS signaling to immediately have a
possibility to point to the already established authorization state
maintained at some nodes in the network. Note that this
authorization token may already be obtained from a different
protocol executed before the QoS signaling took place. This issue
was already mentioned in the previous section.

The QoS signaling messages then travel within the same
administrative domain and experience hop-by-hop protection. Note
that hop-by-hop protection does not necessarily mean that each node

along the path needs to participate in the QoS signaling. Hop-by-hop
security could also mean that the ingress router adds a security
object and the egress router verifies this object and removes it and
the nodes within the network may be transparent from the signaling
point of view. There may also be the case that a different QoS
mechanism is in place within the network and that a different
security mechanism may be used to protect these signaling messages.
Hop-by-hop security assumes a certain trust relationship between the
entities within the core network whereby protection against outsider
attacks and against nodes that do not participate in the QoS
signaling itself is guaranteed.

To provide fast processing of the messages within the core network
the common security mechanism deployed for this purpose is a keyed
message digest of the entire signaling message including a replay
protection indicator (for example a sequence number). To provide
protection against resynchronization failures countermeasures must
be in place. Additionally it is required to consider the case of
sequence number rollovers to provide rekeying in those cases.

The key used to provide this hop-by-hop protection is often
distributed with protocols different from the QoS protocol. Because

of the static nature of the network structure various means of
distributing these security associations are possible. The
possibilities range from manual distribution (for small networks) to
network management protocols that allow automatic distribution to
standard key management protocols. Note that a manually established
QoS security association does obviously not allow rekeying. In
larger networks public key based key management protocols may be
used to allow more flexibility for the network provider. One
important issue that has to be mentioned is that the key management
protocols used must be able to create security associations that can
be used with the QoS protocol. One possibility to secure a QoS
protocol independent of the protocol itself is to provide protection
with IPSec in a hop-by-hop manner. It must be noted that in such a
case it is obvious that the processing for policy aware and policy
ignorant nodes cannot be distinguished. Such a differentiation
between nodes that are QoS aware but policy ignorant and other nodes
that are QoS and policy aware cannot be accomplished at the network
layer. Furthermore there is little possibility for the application
to recognize the existence of the underlying (for the application
transparent) IPSec protection. Missing protection because of a
misconfiguration can therefore not be recognized by the QoS
signaling daemon.

In order to select the correct security association it is necessary

for the individual QoS aware network element (i.e. the network
element actively participating in the QoS signaling) to know the
next hop. Such information should also be available to the key
management protocol and to the entities distributing the security
associations.

From a threat scenario perspective it is obvious that the choice for
a hop-by-hop security nature also implies that insider attacks
cannot be prevented. Hence if an adversary is able to break the
security of a router participating in the QoS signaling then there
is no possibility to prevent this adversary from modifying the QoS
signaling messages or from mounting denial of service attack against
the network. But, anyway, if an adversary is able to break into a
router, he has the possibility of mounting all types of DoS attacks,
not only for QoS. He can just throw away some packets or change some
headers or whatever. Such a threat can only be reduced by end-to-end
security means whereby the end-to-end protection of QoS signaling
message parts is limited to those objects that do not change in
transit. Message parts that need to be modified hop-by-hop obviously
cannot be protected end-to-end. From this point-of-view it would be
appropriate to separate the message parts of the QoS signaling
message into those that change during transit and others that remain
unchanged (mutable and non-mutable message parts). More problematic
is the case where the QoS signaling messages traverse networks where
no security protection (hop-by-hop protection within the network) is
applied at all and the hop-by-hop security principle is violated.
Obviously, in such a case it is not possible to ensure appropriate
protection. We assume that every network requires that QoS signaling

messages transmitted by users always receive the necessary security
protection. The exact threat caused to the network depends on the
QoS signaling protocol used and on the scenario in which such a
signaling is used. QoS protocols with one roundtrip may have the
advantage that the first message establishes some state information
before the actual reservation takes place. It is therefore assumed
that a reservation message that is transmitted without a preceding
path message cannot result in fully successful end-to-end
reservation. A more detailed threat analysis may however be required
for a particular QoS signaling protocol.

If the QoS message leaves one administrative domain and enters
another then network-to-network authentication has to be executed.
Note that the above description assumes that the QoS Initiator is
the user and the subsequent QoS signaling messages traverse through
the network. However QoS signaling can also be initiated by the
network which has basically the same consequences for network-to-
network and intra-domain signaling. Usually it is assumed that two

domains already have service level agreements and have a trust
relationship established. This initial trust relationship then
allows a security association to be dynamically derived based on
some pre-shared secret or an existing public key infrastructure.
Since scalability issues are important and other QoS technologies
may be used that do not provide fine-grained reservations on a flow
level there is no notion of the user initially triggering the QoS
request at this point of the signaling that issued the resource
request. Hence admission control is executed based on policies
shared between the different network service providers.

Finally the QoS signaling message terminates at the destination and
the last hop must also be secured. If we assume a previous initial
authentication and key agreement step of the quality of service
supporting end-device then this existing QoS security association
can be used to protect the message at the last hop. If such a
security association is not available then the corresponding key
management protocol must be triggered to dynamically derive one. In
case of Kerberos the reversed authentication from the network to the
user - the so-called user-to-user authentication could be used. The
Kerberos case however is especially difficult since the network (or
some node within the network to be more precise) does not know the
identity of the end-node and hence it is difficult to request a
session ticket if the realm and the principal name is unknown. A DNS
lookup to determine the Kerberos realm and principal name may be
possible based on the mobile node's or the server's (home) IP
address. Furthermore the network does not know which protocols are
supported by the end-node (which may also be a mobile node) and
hence such a task can be difficult.

9.3 End-to-End

Traditionally there is little support for end-to-end security at the
QoS signaling level. However there are two reasons that might argue

for incorporating end-to-end security mechanisms into a QoS
protocol. First QoS signaling messages contain parts that do not
change during transit and are therefore subject to a possible end-
to-end protection. This circumstance is already described in the
previous section. Hence if a previous protocol already established
an end-to-end security association then such a security association
may be reused for protecting the QoS signaling messages end-to-end.
The second issue tries to address the case where other protocols are
not executed before the QoS signaling took place. In this case the
QoS signaling could be reused to transport authentication and key
agreement messages that are later used to provide protection of the
end-to-end data communication subsequent to the QoS signaling. Some

key management protocols have been proposed that try to establish a
session key with a single roundtrip. These protocols use timestamps
for replay protection. To provide independence against the
underlying transport protocol the data of the key management
protocol are embedded into SDP.

Finally, there is the issue of the subsequent protection of data
traffic between the two end-nodes along the pinned path. TLS and
other higher layer security protocols are independent and unaffected
of the underlying QoS established data path. But in case of network
layer protection as achieved with IPSec the relevant filters need to
be adjusted to be able to identify the encrypted data traffic
accordingly. Whatever QoS protocols are used care must be taken to
ensure to allow IPSec protected data traffic to be transmitted over
the QoS established path.

10 Resilience and Scalability Considerations

Resilience and scalability considerations may influence the NSIS
framework. Resilience usually implies that a network designer
strives hard to avoid single points of failure. At the same time
information needs to be replicated without impacts on performance.

10.1 Resilience

Resilience as considered in this draft deals with NSIS agent failure
and the consequences for the QoS architecture. If QoS control
components are located within the data path, when a node fails or
the data path changes due to re-routing both the signaling and data
paths are affected. Resilience can be achieved by redirection around
the point of failure, using for example, constrained based routing
schemes. However, any state information maintained by the failed
node must be transferred to another node, or re-discovered. If the
QoS enabled path, including the state information can be re-
established in a considerably short time an application would
experience service degradation only for a short time period.

Resilience has also to be considered when NSIS agent resides off the
data path. When there is a node failure or re-routing along the data
path, there is no need to move state information since it still

resides in the same NSIS Agent. However, if the NSIS Agent fails,
then the signaling path and state information must be recovered.
Redundancy is achieved if a substitute NSIS Agent can take over on
demand. Usually in case of failure a smooth change over with
traversal of all current state information is not possible. To
minimize the loss of signaled state information NSIS Agents within

the same network domain may periodically update each other with
context information. A substitute NSIS Agent in action needs to be
properly addressed to process upcoming resource requests. For this
purpose the new NSIS Agent should advertise its presence to
counterpart NSIS agents located in end terminals, adjacent network
domains or any other locations in the Internet.

Resilience issues are closely related to the location of state
information within the network. The locations where state
information is required must be determined, and the more globally
useful the state information is, the more state information will
have to be maintained.

10.2 Scalability

In the NSIS framework critical criteria for scalability can be
described by the amount of state information processing and by the
signaling overhead that is fed into the network.

The first issue deals with the signaling state information that has
to be processed by the NSIS agent. This will rely strongly on a
number of framework level decisions. The following discussion aims
to highlight the amount of state information that must be maintained
for the different options highlighted for a number of open issues in
the framework.

*)Basic Signaling Paths

The two options and their implications are as follows:
-    Sender Initiated: no state information is maintained in NSIS
Agents
-    Receiver Initiated: the 'previous' hop NSIS Agent address must
be maintained to support correct routing of the message if the 'RSVP
style' option is used.

*)NSIS Signaling Protocols

Identified the need for NSIS Agents to maintain information
regarding:
-    Peer NSIS Agents and authentication state and policy associated
with them
-    The next hop NSIS Agent associated with a particular flow or
aggregate
-    Timeout periods for reservations

*)NSIS Signaling Data

Identified the need for NSIS agents to maintain state about:
-    Identifying the flow or aggregate
-    Scope identification for scoping parameters

*)NSIS Aggregation Techniques

Aggregation techniques can have a large impact on the amount of per-
flow state information maintained within the NSIS Agents.

An open issue here concerns whether a domain that is performing
aggregation still needs to maintain per-flow state information.

*)Support for Adaptive Applications
Two options for providing feedback to applications were identified:
-    Send feedback directly to the initiator: this required the
initiator ID to be maintained in the NSIS Agents
-    Send feedback hop-by-hop back towards the initiator: the
previous hop NSIS Agent address must be maintained.
-    If admission control failed, and the application has indicated
that it would like to be informed when resources become available,
per-flow information and the initiator identity must be maintained.
-    Threshold values for when the level of service provided
requires that the application be informed via feedback messages

Signaling overhead is less critical but influences the throughput of
the network to some degree. The amount of signaling information is
influenced by the scale of flow aggregation, the amount of signaled
information per flow and the frequency of state information update.
The NSIS framework provides a flexible approach for transporting
signaling parameters. Though an NSIS protocol should allow the
transportation of a variety of signaling parameters it should at the
same time provide a flexible structure for carrying only signaling
information that is actually required for a specific purpose. For
example an NSIS signaling message chould carry MAC layer specific
parameters like bit error ratio (BER) etc. only if the considered
link layer is able to provide and process this information suitably.
Defining a flexible protocol avoids the transportation of "empty"
information fields significantly.

Another issue is the frequency of information state update by
refresh messages. High accuracy of resource allocation requires the
frequent exchange of refresh messages, which on the other hand
increases the amount of signaling overhead. While host to network
refresh frequency are controlled by the terminals, domain internal
frequency may be determined by ISP policies, e.g. terminal to
network refreshing could be delayed by an NSIS agent at the network
ingress before initiating a separate refresh message at a later
time. Generally an NSIS agent could consider policies for variable
definition of timer refreshing, but this is a matter for the
protocol.

Two outstanding decisions concerning the framework also impact on
signaling overhead:

*)Basic Signaling Paths

The two options and their implications on signaling overhead are as
follows:
-    Sender Initiated: only one end-to-end transmission is required
-    Receiver Initiated: additional round-trips may be required to
determine the correct routing path.

*)NSIS Signaling Data

Identified two different approaches to exchanging parameters that do
not have local scope:
-    Parameter 'stacking': more information is carried by the
signaling protocol, larger messages
-    Separate signaling message: the signaling is lighter-weight,
but more messages are required.

There will be a trade-off between the information carried in the
message as a parameter, and the amount of state information that is
maintained at the NSIS Agent.

11 Conclusion

This document has presented a framework for further discussion of
the requirements and possible implementation architectures of NSIS.
We believe that this framework can be developed into the skeleton of
a more concrete solution, that meets most if not all of the
requirements that were identified in [1].

It has been a major goal of the framework to be modular, so that it
can support a wide range of features without forcing their use
everywhere - lightweight but widely applicable. We believe that this
approach is important for the success of NSIS, since a QoS solution
that is optimised for only a single subset of the Internet will
never gain the wide acceptance it needs to acquire the 'critical
mass' necessary for pervasive deployment of QoS. Some of these
modular capabilities are listed here:
1. Use of bi-directional reservations is a local issue (section
   3.4.3). Outside the path segments subject to the bidirectional
   reservation, the rest of the network does not need to know that
   bidirectional reservations are being used.
2. Multipart reservations could likewise be handled as a local
   matter.

3. Proxy use (3.7.1 and 5.1) is similarly a local issue. The proxy
   simply hides the QoS initiator identification with its own; other
   parts of the network can ignore the fact that the proxy is not
   the real initiator.

4. Registration details are optional and can depend on the local
   environment, and the actual QoS reservation signaling can be made
   lightweight as a result.
5. The framework emphasises the use of different signaling protocols
   and QoS provisioning techniques.
6. Parameters can be scoped to allow local and inter-domain
   parameters to be transported, and the signaling protocol is not
   affected by their inclusion. QoS controllers are allowed to
   ignore all parameters that are out of their scope, and it should
   be trivial to organise packet formats to make this filtering a
   simple process.
Many other aspects of the modularity requirement would be refined
given a more detailed analysis of the QoS signaling data formats
themselves. This would be a second stage of refinement of the
networks.

In developing the framework, we have discovered a number of open
issues about the right way to proceed in more detail. Several of
these are related to the detailed interpretation of particular
requirements, while others are related to design tradeoffs between
flexibility, complexity, and applicability. To resolve these issues,
and to further validate the framework as it stands, a mapping of the
scenarios of [1] onto the framework would be very beneficial in
enabling a detailed comparison to be done.

1.   Should we have a defined set of hierarchical levels in the
     framework, or allow arbitrary nesting? (For example, is there
     a well defined level of 'edges' in the Internet, or can there
     be edges within edges?)
2.   Should receiver initiated reservations be propagated from the
     receiver against the traffic flow direction (requires reverse
     path forwarding state at all intermediate agents) or reflected
     via the sender (requires the QoS request to indicate 'I am
     sending you packets which you wanted'.)
3.   As a related point, should bi-directional reservations be
     treated as a special case mainly to support proxy NSIS agents,
     or as an add on to standard reservations but requiring RSVP
     style receiver oriented reservations?
4.   At what level does any multicast QoS framework need to be
     integrated with a unicast framework? Are common signaling
     protocols useful or necessary? Is a common set of traffic
     classes useful or necessary?

5.   Is there any value in integrating QoS provisioning protocols
     (router remote control) into the NSIS framework, as discussed
     in section 5.4? Do they interact with each other?
6.   Does the NSIS signaling security between the endpoints
     interact with any assumptions about how the application layers
     have associated with each other? Is security between the end
     hosts required at the NSIS level at all, or can all the
     necessary protection be propagated downwards from the higher
     (application) layers.

7.   Is it appropriate to assume that security associations to
     protect the signaling protocol itself are already available?
     If so, how are they hooked into the signaling protocol itself
     (since each end has to agree to use the same security
     association). If not, what is the overhead of developing an
     NSIS security protocol?
8.   Is there a need to protect non-mutable message parts end-to-
     end in addition to the hop-by-hop security protection? Is
     there anyone that argues against the assumptions raised with
     hop-by-hop security?
9.   Are public key based mechanisms appropriate for the protection
     of a lightweight signaling protocol? Please note that the
     answers might be different between the initial authentication
     and key agreement phase and the subsequent protection of
     signaling messages.
10. Are there further interrelationships between the signaling
     protocol and the process of accounting?
11. Which identity should be used for the purpose of
     authentication? Is it enough to have the user authenticate to
     the network or should also the host identity be involved?
12. Should end-to-end security mechanisms receive further
     investigation with a QoS signaling protocol?
13. Is user identity confidentiality a concern that needs to be
     addressed? In the scenarios where it is, can the proxy
     concepts of section 3.7.1 always be applied?
14. How does Path Capability Discovery operate? Are capabilities
     discovered locally or cumulatively, and on a hop-by-hop or
     end-toýend basis.
15. Do we need to support aggregate policy information, e.g.
     indicating between domains how one domain would like it's
     aggregate traffic to be treated, and if so, what does the
     policy information consist of?
16. How is the feedback of QoS violations and resource
     availability supported, and how does it interact with
     aggregate flows?
17. How do we address the issue of end-to-end QoS class
     descriptors to maintain flexibility but also avoid inter-

operability problems?
    18. Should the agent maintain total resource usage and request
         information, or should this be maintained by the local
         provisioning function? (May have implications for NSIS
         management, e.g. MIB design.)

  12 References

    1 Brunner, M. (ed.), "Requirements for QoS Signaling Protocols",
       draft-ietf-nsis-req-00.txt, Work in Progress, February 2002
    2 Fodor G., Persson F., Williams B., "Proposal on new service
       parameters (wireless hints) on the controlled load integrated
       service", draft-fodor-intserv-wireless-params-01, January 2002
    3 Katz, D, " IP Router Alert Option", RFC 2113, February 1997
    4 W. Marshall, K. Ramakrishnan , E. Miller, G. Russell, B. Beser, M.
       Mannette, K. Steinbrenner, D. Oran, F. Andreasen, M. Ramalho, J.
       Pickens, P. Lalwaney, J. Fellows, D. Evans, K. Kelly, A. Roach, J.
       Rosenberg, D. Willis, S. Donovan and H. Schulzrinne, "Integration
       of Resource Management and SIP", draft-ietf-sip-manyfolks-
       resource-03.txt, work in progress, November 2001
    5 Baker F., Iturralde C., Le Faucheur F., Davie B., " Aggregation of
       RSVP for IPv4 and IPv6 Reservations", RFC 3175, September 2001
    6 Bernet, Y. et al, "A Framework for Integrated Services Operation
       over Diffserv Networks", RFC 2998, November 2000
    7 Hain, T. "Architectural Implications of NAT", RFC 2993, November
       2000
    8 Egevang, K, Francis, P, "The IP Network Address Translator (NAT)",
       RFC 1631, May 1994
    9 Nordmark, E. "Stateless IP/ICMP Translation Algorithm (SIIT)",
       RFC2765, February 2000
    10 Johnson, D. B, Perkins, C, "Mobility Support in IPv6", draft-
       ietf-mobileip-ipv6-15.txt, Work in Progress, July 2001
    11 Moskowitz, R. "Host Identity Payload And Protocol", draft-
       moskowitz-hip-05.txt, Work in Progress, November 2001
    12 D. Mitzel "Overview of 2000 IAB Wireless Internetworking
       Workshop" RFC 3002, December 2000
    13 L. Westberg "Resource Management in Diffserv (RMD) Framework",
       draft-westberg-rmd-framework-01.txt, Work in Progress, February
       2002
    14 V. Jacobson, "An Expedited Forwarding PHB", RFC 2598, June 1999
    15 J. Heinanen, "Assured Forwarding PHB Group", RFC 2597, June 1999
    16 L. Westberg "Resource Management in Diffserv On DemAnd (RODA)
       PHR", draft-westberg-rmd-od-phr-01.txt, Work in Progress, February
       2002
    17 Braden, B., Ed., et. al., "Resource Reservation Protocol (RSVP) -
       Version 1 Functional Specification", RFC 2205, September 1997

18 Terzis, A., Krawczyk, J., Wroclawski, J. and L. Zhang, "RSVP
   Operation Over IP Tunnels", RFC 2746, January 2000
19 S. Herzog (ed), et al, "COPS usage for RSVP", RFC 2749, January
   2000
20 Bernet, Y., "Format of the RSVP DCLASS Object", RFC 2996,
   November 2000
21 C. Q. Shen et al. "An Interoperation Framework for Using RSVP in
   Mobile IPv6 Neworks", draft-shen-rsvp-mobileipv6-inerop-00.txt,
   Work in Progress, July 2001

22 Silvano Gai, Dinesh G Dutt, Nitsan Elfassy, Yoram Bernet, "RSVP
   Proxy", draft-ietf-rsvp-proxy-02.txt, Work in Progress, July 2001
23 S. Paskalis et al., "RSVP Mobility Proxy", draft-paskalis-rsvpmp-
   00.txt, Work in Progress, December 2001
24 The INSIGNIA Project, http://www.comet.columbia.edu/insignia/
25 Hemant Chaskar, Rajeev Koodli, "A Framework for QoS Support in
   Mobile IPv6", draft-chaskar-mobileip-qos-01.txt, work in progress,
   March 2001
26 X. Fu, H. Karl, et al, "QoS-Conditionalized Binding Update in
   Mobile IPv6", draft-tkn-nsis-qosbinding-mipv6-00.txt, work in
   progress, January 2002
27 Hamid Syed, et al, "General Requirements for Context Transfer",
   draft-ietf-seamoby-ct-reqs-03.txt, work in progress, January 2002
28 Blake S., Black D., Carlson M., Davies E., Wang Z., Weiss W., "An
   Architecture for Differentiated Services", RFC 2475, December 1998
29 Nichols K., Carpenter B., "Definition of Differentiated Services
   for Per Domain Behaviors and Rules for their Specification", RFC
   3086, April 2001
30 3GPP, "QoS Concept and Architecture", TS23.107 v5.3.0, January
   2002
31 Shenker S., Partridge C., Guerin R., "Specification of Guaranteed
   Quality of Service", RFC 2212, September 1997
32 Wroclawski J., "Specification of the Controlled-Load Network
   Element Service", RFC 2211, September 1997

 13 Acknowledgments

14 Author's Addresses

Robert Hancock (contact), Eleanor Hepworth
Roke Manor Research Ltd
Romsey, Hants, SO51 0ZN
United Kingdom
E-Mail: [robert.hancock|eleanor.hepworth]@roke.co.uk

Cornelia Kappler
Siemens AG
Berlin 13623
Germany
E-Mail: cornelia.kappler@icn.siemens.de

Hannes Tschofenig
Siemens AG
Otto-Hahn-Ring 6
81739 Munchen
Germany
E-mail: Hannes.Tschofenig@mchp.siemens.de

Jorge R Cuellar
Siemens AG
Otto-Hahn Ring 6
81730 Munich
Germany
E-mail: jorge.cuellar@mchp.siemens.de

Jochen Eisl
Siemens AG
Hofmannstr. 51
81359 Muenchen
Germany
E-Mail: jochen.eisl@icn.siemens.de

Mehmet Ersue
Siemens AG
Hofmannstr. 51
81359 Munich
Email: Mehmet.Ersue@icn.siemens.de

Xiaoming Fu, Holger Karl
Technical University Berlin
Sekr. FT 5-2, Einsteinufer 25
Berlin 10587
Germany
E-Mail: [fu|karl]@ee.tu-berlin.de

Marcus Brunner
NEC Europe Ltd.
Network Laboratories
Adenauerplatz 6
D-69115 Heidelberg
Germany
E-Mail: brunner@ccrle.nec.de

Andreas Kassler
Dept. Distributed Systems
University of Ulm
Oberer Eselsberg
89069 Ulm
Germany
E-Mail: kassler@informatik.uni-ulm.de

Appendix A. Mapping to Requirements

The following section examines the requirements outlined in [1] and
verifies that the framework can support the requirement and explains
how it does so. The numbering of the requirements is taken directly
from [1].

5.1 Architecture and Design Goals

5.1.1 Applicability for different QoS technologies.
5.1.2 Resource availability information on request
5.1.3 Modularity
5.1.4 Decoupling of protocol and information it is carrying
5.1.5 Reuse of existing QoS provisioning
5.1.6 Avoid duplication of [sub]domain signaling functions
5.1.7 Avoid modularity with large overhead (in various dimensions)
5.1.8 Option to use the protocol for existing local technologies
5.1.9 Independence of signaling and provisioning paradigm

Requirement	Supported by Framework
5.1.1	General feature of framework, examples of how QoS technologies can be used within the framework are provided in sections 5.4 and 6.
5.1.2	Supported by the Path Capability Discovery aspects of the QoS Signaling Protocol (section 4.3)
5.1.2	General feature of the framework, examples of this are

```
             | provided in sections 1 and 11
    ----------------------------------------------------------------
    5.1.4       | The NSIS Signaling Data (section 4.4) is independent
                | of the NSIS Signaling Protocol used in any particular
                | domain
    ----------------------------------------------------------------
    5.1.5       | Many types of different QoS provisioning techniques
                | can be fitted within the framework as illustrated in
                | sections 4.2 and 5.4
    ----------------------------------------------------------------
    5.1.6       | This must be considered in context of a more detailed
                | description of the QoS provisioning mechanisms and how
                | link layer aspects are managed here
    ----------------------------------------------------------------
    5.1.7       | Framework tries to achieve this in a number of ways,
                | see section 11
    ----------------------------------------------------------------
    5.1.8       | This is supported by the framework, see section 5.4
    ----------------------------------------------------------------
    5.1.9       | Fundamental characteristic of the framework to
                | separate signaling from actual QoS provisioning
    ----------------------------------------------------------------
```

5.2 Signaling Flows

5.2.1 Free placement of QoS Initiator and QoS Controllers functions
5.2.2 No constraint of the QoS signaling and QoS Controllers to be
in the data path.
5.2.3 Concealment of topology and technology information
5.2.4 Optional transparency of QoS signaling to network
5.2.5 Deal with IP fragmentation gracefully

```
    ----------------------------------------------------------------
    Requirement | Supported by Framework
    ----------------------------------------------------------------
    5.2.1       | The framework makes no assumptions about the location
                | of the NSIS Agents
    ----------------------------------------------------------------
    5.2.2       | NSIS Agents can be on or off the signaling path
    ----------------------------------------------------------------
    5.2.3       | Can be achieved using NSIS proxy agents and is also a
                | side effect of aggregation
    ----------------------------------------------------------------
    5.2.4       | The QoS signaling protocol supports the transparent
                | transport of parameters either by parameter 'stacking'
                | or by initiating multiple signaling flows (section
                | 4.4)
```

```
      -----------------------------------------------------------------------
      5.2.4        | This is an open issue related to possible packet
                   | classification rules. It cannot be solved purely
                   | as a signaling issue
      -----------------------------------------------------------------------


      5.3  Additional information beyond signaling of QoS information


      5.3.1  Explicit release of resources
      5.3.2  Ability to signal life-time of a reservation
      5.3.3  Possibility for automatic release of resources after failure
      5.3.4  Possibility for automatic re-setup of resources after
      recovery
      5.3.5  Prompt notification of QoS violation in case of error /
      failure to QoS Initiator and QoS Controllers
      5.3.6  Feedback about the actually received level of QoS guarantees
      5.3.7  Automatic notification on available resources not been
      granted before
```

```
      -----------------------------------------------------------------------
      Requirement | Supported by Framework
      -----------------------------------------------------------------------
      5.3.1        | This is an attribute of the NSIS Signaling Protocol
      -----------------------------------------------------------------------
      5.3.2        | This can be included in the NSIS Signaling Data
      -----------------------------------------------------------------------
      5.3.3        | Something can be implemented by the NSIS agent to
                   | support this provided the signaling protocol enables
                   | failure detection
      -----------------------------------------------------------------------
      5.3.4        | see 5.3.3
      -----------------------------------------------------------------------
      5.3.5        | Supported by NSIS signaling feedback mechanism (
                   | section 5.7)
      -----------------------------------------------------------------------
      5.3.6        | Supported as part of the NSIS Signaling reservation
                   | mechanism
      -----------------------------------------------------------------------
      5.3.7        | This could be implemented given an asynchronous
                   | signaling protocol and the appropriate monitoring
                   | functionality within the QoS controller with support
                   | from the provisioning mechanism
      -----------------------------------------------------------------------


      5.4  Layering


      5.4.1  The signaling protocol and QoS control information should be
```

application independent.
```
---------------------------------------------------------------
Requirement | Supported by Framework
---------------------------------------------------------------
5.4.1       | The framework makes no assumptions regarding
            | applications, and includes support for the opaque
            | transport of application information (section 4.4).
---------------------------------------------------------------
```

5.5 QoS Control Information

5.5.1 Mutability information on parameters
5.5.2 Possibility to add and remove local domain information
5.5.3 Simple mapping to lower-layer QoS provisioning parameters
5.5.4 Aggregation method specification
5.5.5 Multiple levels of detail
5.5.6 Ranges in specification
5.5.7 Independence of reservation identifier
5.5.8 Seamless modification of already reserved QoS
5.5.9 Signaling must support quantitative, qualitative, and
relative QoS specifications
5.5.10 QoS conformance specification

```
---------------------------------------------------------------
Requirement | Supported by Framework
---------------------------------------------------------------
5.5.1       | End-to-end parameters are transported unchanged end-to
            | -end, but domain specific parameters can also be
            | signaled (section 4.4)
---------------------------------------------------------------
5.5.2       | Supported by either parameter 'stacking' or by
            | initiating separate signaling message (section 4.4)
---------------------------------------------------------------
5.5.3       | Assumed to be a function of technology specific
            | convergence sublayer (inter provisioning) and is
            | dependent on the complexity of the QoS parameters (see
            | section 8)
---------------------------------------------------------------
5.5.4       | Any NSIS Agent is capable of inserting aggregation
            | parameters into the NSIS Signaling messages (section
            | 4.4)
---------------------------------------------------------------
5.5.5       | see 5.5.2
---------------------------------------------------------------
5.5.5       | Will be supported as part of the QoS descriptor
            | activities, transport of which is supported by the
```

```
             | framework.
       ---------------------------------------------------------------------
       5.5.7       | see 5.5.5
       ---------------------------------------------------------------------
       5.5.8       | NSIS signaling messages can be generated at any time
                   | to support service re-negotiation
       ---------------------------------------------------------------------
       5.5.9       | see 5.5.5
       ---------------------------------------------------------------------
       5.5.10      | see 5.5.5
       ---------------------------------------------------------------------
```

5.6 Performance

5.6.1 Scalability in the number of messages received by a signaling
communication partner (QoS initiator and controller)
5.6.2 Scalability in number of hand-offs
5.6.3 Scalability in the number of interactions for setting up a
reservation
5.6.4 Scalability in the number of state per entity (QoS initiators
and QoS controllers)
5.6.5 Scalability in CPU use (end terminal and intermediate nodes)
5.6.6 Low latency
5.6.7 Low bandwidth consumption

```
       ---------------------------------------------------------------------
       Requirement | Supported by Framework
       ---------------------------------------------------------------------
       5.6.1       | Number of messages depends on some high level
                   | framework decisions (section X)
       ---------------------------------------------------------------------
       5.6.2       | This depends on the detailed protocol design. The
                   | protocol design can be optimized for the environment
                   | for which this type of scalability is needed.
       ---------------------------------------------------------------------
       5.6.3       | see 5.6.2
       ---------------------------------------------------------------------
       5.6.4       | Amount of state maintained depends on some high level
                   | framework decisions (section X)
       ---------------------------------------------------------------------
       5.6.5       | see 5.6.2
       ---------------------------------------------------------------------
       5.6.6       | see 5.6.2
       ---------------------------------------------------------------------
       5.6.7       | see 5.6.2
       ---------------------------------------------------------------------
```

5.7 Flexibility

5.7.1 Aggregation capability, including the capability to select
and change the level of aggregation.
5.7.2 Flexibility in the placement of the QoS initiator
5.7.3 Flexibility in the initiation of re-negotiation (QoS change
requests)
5.7.4 Uni / bi-directional reservation

Requirement | Supported by Framework

5.7.1 | Supported by aggregation parameters carried by
            | NSIS Signaling Protocol

5.7.2 | Framework makes no assumption about the location of
            | NSIS Agents

5.7.3 | Framework supports the asynchronous generation of
            | signaling messages to support re-negotiation

5.7.4 | Both can be supported by the framework ([section 3.4](#))

5.8 Security

5.8.1 The QoS protocol must provide strong authentication
5.8.2 The QoS protocol must provide means to authorize resource
requests
5.8.3 The QoS signaling messages must provide integrity protection.
5.8.4 The QoS signaling messages must be replay protected.

5.8.5 The QoS signaling protocol must allow for hop-by-hop
security.
5.8.6 The QoS protocol should allow identity confidentiality and
location privacy.
5.8.7 The QoS protocol must prevent denial-of-service attacks
against signaling entities.
5.8.8 The QoS protocol may support confidentiality of signaling
messages.
5.8.9 The QoS protocol should provide hooks to interact with
protocols that allow the negotiation of authentication and key
management protocols.
5.8.10 The QoS protocol should provide means to interact with key
management protocols

All of these requirements are already allocation to either the
protocol or signaling data components of the framework. More

detailed analysis of these requirements should therefore be carried
out in the context of particular selected signaling protocols or
concrete definitions of the signaling data format.

5.10 Interworking with other protocols and techniques

5.10.1 Interworking with IP tunneling
5.10.2 The solution should not constrain either to IPv4 or IPv6
5.10.3 Combination with Mobility management
5.10.4 Independence from charging model
5.10.5 The QoS protocol should provide hooks for AAA protocols
----
Requirement | Supported by Framework

5.10.1 | This is an issue concerning the NSIS Signaling
            | protocol choice

5.10.2 | No assumption concerning the IP version is made.
            | Addressing issues of highlighted in section 5.6

5.10.3 | Framework does not preclude in-band signaling or
            | other such optimizations (section 7.3)

5.10.4 | NSIS supports charging and accounting functions by
            | allowing the exchange of pertinent information, but
            | does not assume any charging models (section 3.5)

5.10.5 | This is an issue concerning the NSIS Signaling
            | protocol choice