Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

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

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

This Internet-Draft will expire on December 21, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

The MIPSHOP working group is working on common functionality necessary for the support of media independent handover (MIH) protocols, initially focusing on the architectures being developed with the 802.21 working group of the IEEE. A problem statement draft describing the context and scope of the common functionality is already in progress, and a second draft has been prepared that describes the main protocol design issues that have to be taken into consideration. Although the common functionality (relating to
transport and security requirements) is mainly standard, there is still a significant element of protocol design to integrate them with the MIH service itself, including in particular secure peer discovery and capability negotiation. This draft presents a solution for the common protocol functionality based on the use of the General Internet Signaling Transport (GIST) protocol, which has been developed in the NSIS working group. GIST can already satisfy the bulk of the protocol requirements in its current form, and this draft explains how this existing GIST functionality fits into the MIH problem space. The main special requirements for MIH relate to signalling node discovery and message routing, and this draft outlines three options for supporting these requirements within the GIST framework.

Table of Contents

1. Introduction .................................................. 3
2. Functionality Overview ...................................... 4
  2.1. Discovery and Routing ..................................... 4
  2.2. Message Transport ........................................ 5
  2.3. State Management .......................................... 5
  2.4. Multiplexing and Extensibility ............................ 6
  2.5. Network Layer Interactions ................................. 6
  2.6. Message Security .......................................... 7
  2.7. Overload Management ....................................... 8
  2.8. Failure Handling .......................................... 9
3. Extended Message Routing Methods ................................. 9
  3.1. Overview ................................................ 9
  3.2. A Message Routing Method for MIH Services ................ 10
  3.3. Multi-MIH Service Discovery ............................... 12
4. MIH Service Development ........................................ 13
5. Security Considerations ......................................... 15
6. Conclusions .................................................. 16
7. Informative References ......................................... 17
Appendix A. Acknowledgements ...................................... 18
Authors’ Addresses .............................................. 20
Intellectual Property and Copyright Statements ................. 21
1. Introduction

The MIPSHOP working group is currently working on common functionality necessary for the support of media independent handover (MIH) protocols, initially focussing on the architectures being developed with the 802.21 working group of the IEEE. MIH protocols are used to exchange information between terminals and networks to convey information about radio conditions, available resources and other handover requirements in order to allow optimised handover decisions to be taken and executed in heterogeneous wireless environments (that is, environments where more than one type of radio access technology is available or in use at any one time). Thus, although the MIH protocols carry link layer information, the protocols themselves are not tightly bound to any specific link layer. Instead, they can be carried at the network layer, both within the radio access infrastructure and over the air.

A problem statement describing the context and scope of the common functionality is already in progress [1]. This draft describes a decomposition of the overall MIH protocol problem into a common part, and a set of specific users or MIH services which is layered over it. This structure is consistent with the architecture defined by 802.21 [2], where the services in question have been identified as the Event, Command and Information Services (ES/CS/IS). The problem statement itself outlines a division of functionality between the common part and the individual services, with the intention that the common part will be generally useful even in non-802.21 architectures. The expectation is that the MIPSHOP working group will develop a solution for the common part which meets the 802.21 requirements as expressed in the problem statement draft. A second draft outlining more detailed design considerations for the common part has also been prepared [3], which formalises some of the open issues the scope of the common functionality, and lists the key design choices that have to be made and the criteria that should be taken into account in doing so. Although [3] refers to components of possible solutions as examples, it is not a solution proposal in its own right.

While the common functionality (relating to transport and security requirements) is mainly standard, there is still a significant element of protocol design to integrate them with the MIH service itself, in particular secure peer discovery and capability negotiation. This draft presents an outline solution for the common protocol functionality, based on the use of the General Internet Signaling Transport (GIST) protocol [4]. GIST has been developed in the NSIS working group to provide common protocol support for a diverse set of signalling applications, such as resource reservation [5] and middlebox control [6]. Within the overall NSIS protocol
suite, GIST has responsibility for the discovery (implicit or explicit) of the signalling peers that should take part in a transaction, and the actual transfer of the signalling messages between them (including security and transport issues). In this sense, it is already aligned at a high level with the requirements for the MIH support, especially so far as message transfer functionality is concerned. For discovery, the initial use cases for GIST have followed the ‘path-coupled’ signalling paradigm, where the signalling peers are assumed to lie on the path taken by a flow in the network; however, GIST has the concept of modular message routing methods to implement alternative discovery paradigms, and this flexibility can be used to develop a simple extension of GIST for the MIH problem space if this turns out to be necessary.

This draft is structured as follows. Section 2 provides an overview of how GIST can be used to support the various functional requirements for the MIH support; in most cases, there is a single natural approach and only a brief discussion is required. The case of routing and discovery is more complex; this is discussed in more detail in Section 3, which outlines three possible design options and their advantages and disadvantages. Section 4 describes the process of implementing an MIH service within such a protocol framework. Security considerations are discussed in Section 5, and finally Section 6 summarises the open issues and provides conclusions and recommendations.

2. Functionality Overview

This section follows the structure of section 3 of [3] in addressing each design consideration in turn.

2.1. Discovery and Routing

Discovery and routing are the most MIH-specific parts of the signalling problem. The considerations of [3] imply that this functionality should be integrated as part of the GIST layer at least to some extent, for example because it interacts very strongly with the re-use of standard security and transport protocols which are the basis of GIST functionality. The current GIST specification includes one routing method that could be used to allow the MIH services to specify the signalling peer address directly, and this could be used as a fallback method. However, it is likely that a GIST extension to support a different type of message routing will provide better performance. This issue is therefore discussed in more detail in Section 3.
2.2. Message Transport

All the standard message transport requirements can be supported by GIST, by virtue of it being able to negotiate the use of standard transport protocols. The baseline GIST specification supports a minimal unreliable mode (using UDP), and reliable, congestion controlled and flow controlled transport over TCP, which is also the only mode to allow large messages. As an implementation and policy issue, GIST is allowed to create multiple TCP connections to avoid head of line blocking or to support differential transmission priorities. Between two peers, messages may be transported reliably, unreliable, or any mixture of the two. Note that the MIH services would not directly select between these various possibilities; they just indicate the transmission requirements for a particular message and rely on GIST to transport it accordingly.

It is not foreseen that additional transport protocols would be needed for the support of MIH services. Services that intermittently required a large volume of unreliably transported real-time data (e.g. for measurement reporting) could benefit from the use of DCCP, although careful service design and implementation would be needed to avoid saturating the radio links with reporting data. A GIST extension to use SCTP has already been proposed [7], which would allow a single transport connection to support multiple priority streams as well as mixing reliable and unreliable data, which would lead to efficiency savings; however, support for SCTP in mobile nodes is not currently widespread.

2.3. State Management

Conceptually, GIST provides a stateless service to the upper layers (MIH services). It is up to the MIH service to define its own transaction structure; whether or not transactions nest or overlap or how a transaction structure is even indicated at all is left open. Also, there is no hard binding or dependency between state at the GIST level and state at the MIH service level: they can be managed largely independently. For example, the opening or closing of a TCP connection within GIST has no significance to an MIH service — transport connection state can be managed separately. In particular, if the MIH service needs to refresh state periodically at a peer or tear it down, this is done with explicit messages at the service level rather than implicitly by configuration operations on GIST.

GIST does have a concept of signalling sessions; however, a signalling session is simply a group of signalling messages with a common session identifier (SID). It is left to the signalling application to generate SIDs and assign them to individual messages or groups of messages. The main significance of SID selection is
that messages for a single session that are delivered reliably will be delivered in order. (In addition, GIST maintains a separate copy of the routing state for each signalling session that it is aware of, although this is to handle certain security considerations rather than in the expectation that routing state will be SID dependent.) In the MIH context, it is likely that only a small number of sessions (e.g. 1) will need to be created by an MIH service while it is operating for any single mobile node.

GIST will automatically maintain its internal state to ensure that MIH messages can continue to be transported. MIH services can provide timing information on how long it will require such state to be kept for. GIST can provide hints to the MIH service that the state of peer node may have changed (see Section 2.8), but it is up to the MIH service itself to verify the exact situation and take appropriate recovery actions.

2.4. Multiplexing and Extensibility

GIST supports multiple signalling applications. Messages for different signalling applications are transported independently, and a single identifier distinguishes which application in a node should receive the message. The identifier is the NSIS Signalling Layer Protocol Identifier (NSLPID), which is a two byte field managed by IANA (see section 9 of [4]).

MIH services could use a single NSLPID, with further demultiplexers defined inside the signalling application payload, or a set of NSLPIDs could be allocated. Because the routing of a message depends on the NSLPID, the choice has implications for the design of the routing and discovery functionality, see Section 3 below. The minimal approach would be to request a new NSLPID for each MIH service, but more sophisticated options are possible. General guidelines on the extensibility of the NSIS protocol suite as a whole are contained in [8].

2.5. Network Layer Interactions

The signalling messages themselves have to pass through the network infrastructure, and thus are subject to the same problems with middlebox (NAT and firewall) traversal as all other types of data traffic. The signalling also has to operate in both IPv4 and IPv6 environments and also possibly in transition scenarios.

Most of these issues have to be handled in the GIST layer. Since this uses standard transport encapsulations (currently UDP and TCP) for all its data, NAT traversal should not be a problem. (The transport connections are initiated from the terminal, which in most
cases will be on the ‘private’ side of any NAT, which is the same scenario as typical host-initiated client-server interactions.) The UDP exchanges use a specific well-known port which firewalls would have to be configured to pass. The transport protocol negotiation allows agile port numbers to be used for the TCP connections, so this would either require GIST-awareness in the firewall to open the correct ports, or pinning the port to be offered in the GIST layers in the signalling nodes themselves. GIST is IP version neutral because it uses transport protocols which can operate equally over IPv4 or IPv6.

Special problems arise if the signalling application data, i.e. the messages defined for the MIH services, contain embedded IP addresses for third party nodes (i.e. other than the addresses of the signalling nodes themselves). If this is the case, these payloads may need to be translated in an application specific way as the message passes through a NAT. Whether this is actually necessary depends on a more detailed analysis of the MIH services themselves; if it is, it is potentially a source of considerable complexity because it requires the signalling to be aware of the address realm topology over a potentially wide area. It would be ideal if it was possible to express any IP addressing information at the MIH service level in terms of a single addressing realm (for example, the addressing as seen by the terminals themselves). The current 802.21 [2] draft does include payloads with such data, and further analysis would be needed to understand whether these need to be carried explicitly as addresses or simply as references to nodes in the signalling exchange.

2.6. Message Security

GIST is able to negotiate the use of channel security protocols to protect the messages between adjacent signalling peers. (Here, the term ‘adjacent’ means two nodes supporting the same signalling application, without any other node for that signalling application between them. There could be one or several IP hops between the peers.) In principle, any channel security protocol can be negotiated; in the initial GIST specification, the single mandatory-to-implement mechanism is TLS over TCP, with TLS authentication using certificates. Further security options could be added within the GIST framework with relatively simple extensions, such as:

- TLS over other transports (DTLS over UDP, TLS over SCTP)
- Other TLS authentication mechanisms
- IPsec using IKE or KINK for authentication and key exchange
The main motivation for defining additional security options would be to re-use existing security (authentication) infrastructures with existing deployed sets of credentials. It would be natural to carry out such extensions to GIST itself, rather than building something MIH specific into the signalling application level. In particular, GIST already includes a protocol negotiation capability which can support the addition of further options, and designing such a negotiation which is secure against bidding-down and denial of service attacks is not trivial.

The operation of the channel security mechanism chosen can be made largely invisible to the MIH services. For any specific message they simply have to indicate to GIST that security protection is required. The authenticated identifier of the peer can be checked before the message is sent, and this is also available at the receiver. The initial negotiation process (what to offer, what to accept and what to demand) is policy driven, and it may be appropriate for MIH services to require a particular security policy to ensure that appropriate security associations are actually created.

2.7. Overload Management

The basic requirement for overload management within GIST is to avoid causing network layer congestion in the Internet (between signalling nodes). This is achieved mainly by using congestion-controlled transport protocols for the bulk of signalling data transfer, coupled with rate control for the unreliable message transfer and exponential backoff for the initial discovery transactions. These also serve to handle processing overload within the GIST layer itself.

GIST simplifies the process of writing overload-controlled signalling applications by providing a flow controlled transport service between them. The flow control runs directly between the signalling application nodes; if the receiving node is not able to accept signalling data fast enough from GIST, the sending node will be flow controlled. Overload adaptation mechanisms are left entirely to the application: possibilities include discarding messages at the receiver after minimal processing, discarding messages at the sender when overload is detected, or adapting transaction timers (e.g. refresh rates) at the sender.

As well as overload caused by excessive volumes of (legitimate) signalling application data, there is a related problem of overload caused by malicious injection of signalling application data as part of a denial of service attack. GIST includes a two-stage defence against this type of attack. Firstly, the initial discovery process incorporates a cookie exchange which forces the initiator of the discovery process to create local state before a signalling
relationship can be created. Once this has been done, if channel security is used this is an efficient mechanism to reject messages from non-authenticated peers.

2.8. Failure Handling

As part of the discovery process, GIST is responsible for detecting when there is no signalling support for a particular service in the network. Detection would result from failure to receive a response to a query after some number of retries with an exponential timer backoff; the number of retries can be configured as part of local GIST policy, and has to be set as a tradeoff between resilience and timeliness. Once a signalling node has been discovered, GIST does not perform application liveness monitoring itself, but it is able to provide hints to the application that an error condition may have occurred if indications from the network or transport layer suggest this.

3. Extended Message Routing Methods

3.1. Overview

A key aspect of the operation of any signalling protocol is how the peers in the message exchange are defined and discovered. Distinct from typical application protocols, the correct peer will not be defined in terms of some global identifier (e.g. a specific DNS name or IP address), but will rather be a function of two variables:

- The type of signalling to be done (be it for resource reservation, middlebox control, MIH services, or whatever)
- The network context within which the signalling takes place

For example, for the MIH Event Service, the destination for a message from a terminal can be abstractly stated in some form like "the ES server for the network to which I am currently attached". In the NSIS protocol suite, the responsibility for routing messages is placed on GIST, which takes the abstract destination specification and turns it into message delivery towards a concrete node. This may be done either by transmitting the message directly with a special encapsulation so it is intercepted at the correct node without prior communication setup; or, by using the first technique to discover the peer node explicitly and then send messages point to point. A related element of functionality is that GIST is responsible for monitoring the local network state and determine when the peer node has or might have changed. (This is generically referred to as route change handling, because a change in the local topology is the
commonest cause for such a peer change; however, the same
functionality handles other situations, such as changes in the
capability of a signalling node itself.) Note that discovery is the
responsibility of the initiator of the signalling exchange, but once
it has taken place, further transactions can be started by either
peer (i.e. the association is symmetric); in the following we assume
that the initiator in MIH scenarios is always the terminal.

The combination of functions need to perform message routing
(including peer discovery and route change handling) in GIST is
referred to as a ‘Message Routing Method’ (MRM). An MRM defines a
specific set of data items on which the message is routed, the
Message Routing Information (MRI). Two MRMs are defined in the base
GIST protocol; it needs to be decided whether to base MIH support on
one of these, or whether a new MRM should be defined. The current
MRMs are:

Flow-coupled: The signalling message is defined to pass along the
same set of nodes as the packets of a specific flow; the flow is
defined by a header N-tuple, which makes up the MRI itself. This
MRM is not applicable in the MIH context, because the signalling
messages are not associated with particular flows. (This MRM is
primarily aimed at resource reservation signalling applications.)

Loose-end: The signalling message is sent in the direction of a
particular destination identified by its IP address, but is
expected to be intercepted at a network boundary on the way.

It would be possible to use the loose-end MRM for MIH support,
thereby essentially pushing the discovery responsibility into the
signalling application (MIH service). The signalling application
could use any of a number of IP layer discovery techniques to
determine the destination IP address for a particular MIH service,
such as DHCP [9][10], SLP [11], mDNS [12], or even link-specific
techniques, and then use that address directly as the signalling
destination in the loose-end MRM. Although such techniques are not
favoured (for reasons discussed more extensively in [3]), this
approach can always be used as a fall-back if required.

3.2. A Message Routing Method for MIH Services

An alternative approach is to create a new MRM, where the message
routing information includes precisely the identifiers that are
needed to select between the possible peers to take part in a
signalling exchange. Defining the set of identifiers requires
further analysis of the set of MIH services, but could include:
- The MIH service identifier (if this is not already provided by the NSLPID itself).

- Some information about the terminal, such as the realm id of the home network (for example if the information service is proxied back to nodes in the home network).

- Some information about the terminal type or capability.

Note that the information should be strictly limited to that which is necessary for message routing; any other MIH specific information can be carried in the service payload itself.

The terminal sends signalling messages with this routing information towards its first-hop IP access router (AR). Here we are assuming that IP connectivity is already available; the case where IP connectivity is not available is already handled by lower layer protocols in the 802.21 model [2], and would be out of scope for other MIH work in the IETF. An operator deploying MIH services in a wireless network is required to configure the ARs so they either support the MIH services directly, or know how to forward signalling messages to nodes that do. There are at least two options here:

1. The AR is able to parse the GIST message including the MRI, and consult local routing information to forward the message on to the actual MIH serving node. The routing information has to be installed by the network operator, which could typically be done using network management or network autoconfiguration techniques (as discussed above) - the actual method is not visible to the terminal. The end result will be that the terminal will peer directly with the correct signalling node. The same technique can be used to forward the signalling exchange upwards through a chain of proxies. This mode of operation is conceptually similar to the forwarding of AAA messages in AAA protocols such as RADIUS[13] or Diameter [14], where the analogue of the message routing information is the realm part of the NAI. A further refinement would be to define a default lookup mechanism in the AR which could be overridden by specific manual configuration.

2. A second technique which requires less GIST-specific processing in the AR is to configure the AR to intercept the discovery messages at the IP/UDP level and forward them directly to a dedicated node which performs the MIH-specific routing. The format of the GIST encapsulation of discovery messages means that this is possible on most standard routers (in an implementation specific way). Similar techniques are described for the purpose of carrying out a type of off-path signalling in [15].
The benefit of this approach compared to putting the discovery responsibility into the MIH service is that the terminal requirements are minimised and that considerable freedom is left for the internal configuration of the network. The terminal can issue signalling messages as soon as IP connectivity is available without waiting for a separate discovery process (e.g. DHCP or SLP) to complete; this is particularly valuable in the post-handoff stage where the terminal could verify whether the MIH serving nodes have changed within a single round trip. There is also no need to select a single discovery protocol, which would be mandatory-to-implement for all terminals and network environments, including those in which no such protocols are automatically expected to be used (e.g. IPv6 with stateless autoconfiguration). Indeed, it may be possible to generalise this method to the case where direct layer 2 transport is being used before full IP connectivity is available at all. If discovery is handled within the GIST layer in this way, GIST can also take responsibility for tracking the correctness of the routing state by periodically re-probing the network to find if conditions have changed. A further advantage is that the denial of service properties of the signalling connection setup can be analysed in an integrated way, rather than being a property of the discovery protocol selected by the MIH service, the transport protocol it uses, and the interaction between them.

3.3. Multi-MIH Service Discovery

The above approaches to signalling node discovery require a query-response transaction for each different MIH signalling node that the terminal wishes to communicate with. Because each MIH service could conceptually be implemented on a different node, GIST requires a query-response exchange for each MIH service. This would be the behaviour if the approach of using a single NSLPID for each MIH service was used, as discussed in Section 2.4. However, one requirement that has been discussed during various stages of the 802.21 developments is that it would be desirable to be able to carry out multiple discovery operations in a single message exchange.

One solution to this conflict is to reject the requirement. It is in any case a performance optimisation rather than a fundamental goal. Whether this is acceptable depends on whether there are many MIH services with strong time constraints on the discovery phase. If there are several such services, then there are two ways to aggregate them under a single NSLPID, which is an absolute requirement to be able to discover multiple service endpoints with a single GIST query. (Please note that familiarity with the GIST design [4] is a prerequisite for understanding the content of this section.)
1. To carry the required set of service identifiers as piggybacked NSLP data in the query message. The first receiver of the Query message can respond on behalf of the services that it supports, and forward a modified Query to another node or nodes for the remainder. Although this is conceptually possible, it would be very hard to fit into a standard GIST implementation, because the GIST stack in the initiating terminal would see multiple responses to a single Query, with no information visible at the GIST layer to distinguish between them. Also, for subsequent messages, there is no information visible at the GIST layer about which service is being invoked, so there would be no method to distinguish which peer to send the message to. Therefore, this method is not favoured.

2. To carry the required set of service identifiers as part of the message routing information (for example, as a capability bitmap). Again, the first receiver generates a response for the subset of services that it supports, and re-forwards the Query with a reduced set of service identifiers. Each response creates a separate item of GIST routing state for a specific subset of supported services. Subsequent messages would indicate a particular service identifier and be matched against the routing state for it.

The second technique is conceptually feasible and does fit into the GIST layer model relatively naturally. It would require some analysis of the GIST state machine behaviour to ensure that necessary modifications can be confined to the MRM specification. (The current definition of the routing state machine assumes that there is a single response to any given Query. One simple way to handle the extension is to require the initiating node to clone the state machine when it receives a partial response, which mimics the same behaviour as would occur if multiple Queries had been sent in the first place.) Analysis would also be needed of the NAT traversal behaviour; however, it is already a requirement that this has to be able to allow multiple responses for the same query (because of retransmissions).

It requires further analysis to decide whether the additional complexity of such a solution is merited by the performance improvement that is gained. Note that similar problems arise with any type of discovery mechanism, where there are potentially multiple (and an unknown number) of responses to a single message.

4. MIH Service Development

On the assumption that GIST is used to provide the common MIH support
functions, the process of formalising the way that an MIH service would use it is relatively simple. If the MIH service is already defined as a sequence of message exchanges, the formalisation mainly consists of a description of how the MIH service requires GIST to be configured and how it invokes GIST to transport each message in turn.

1. NSLPIDs for the MIH service need to be chosen; there may be one or several (see Section 2.4). Typically any single message would be associated with a single specific NSLPID unless there is some type of aggregation concept, which seems not to be applicable here.

2. The invocation of GIST to transfer a message is modelled in terms of an abstract API (section B of [4]). The originator invokes SendMessage() and the receiver processes it with RecvMessage(). The actual message payload is defined by the MIH specification and is opaque to GIST. Further parameters influence the message transfer process, and the MIH service needs to define how these will be used:

   A. The MIH service needs to decide how messages are to be assigned to signalling sessions (see Section 2.3).

   B. The MIH service needs to provide message routing information (see the discussion above in Section 3). This includes the option to require the message to be routed explicitly to the same peer that took part in a previous exchange (this can be used, for example, for explicit state teardown).

   C. The MIH service can specify additional security and transport attributes (whether channel security is required and whether the message needs to be reliably delivered).

   D. The MIH service may also provide hints about the message priority, timeouts on how long delivery should be attempted, and how to be notified about error conditions relating to the message, although all of these only affect local processing within the node rather than end to end behaviour.

3. If a new MRM is needed to support the MIH routing and discovery requirements, this also needs to be specified as a GIST extension in parallel to the MIH specification itself (see Section 3 above). Part of this includes the definition of how to detect and react to route change events, which can also result in notifications over the API between GIST and the MIH services.

4. If additional authentication methods or security protocols need to be defined (compared to the baseline of TLS + certificates) to
ease MIH deployment in particular environments, these should also be defined as GIST extensions. In addition, the criterial for security policies for the use of GIST for MIH will need to be developed, and a security analysis of the combined (MIH service + GIST) solution. These issues are discussed in the security considerations section below, Section 5.

5. Security Considerations

This document proposes to address the MIH requirements within a layered protocol model, within which some security functionality is provided in a generic way by a lower layer (GIST).

It is important to note that not all the MIH security requirements can be met by GIST, nor can the applicability of GIST security for MIH be analysed in isolation from other parts of the MIH solution. In particular, whether the end result is secure depends on how GIST is used (what features are used and how they are invoked). In addition, GIST is completely neutral to certain security requirements, specifically in the area of authorisation, and these must be handled by the correct implementation and configuration of the MIH services that run over it. A discussion of general threats to signalling (with a partial focus on QoS and on-path routing) can be found in [16]; most of these issues can be extended to the MIH problem space.

Therefore, the approach taken here is to describe the security functionality provided by GIST, and indicate how it could be used as part of a more complete security solution. The main functions are as follows (details are given in Section 2.6):

Message protection: GIST can negotiate the use of a standard channel security mechanism (initially TLS) to provide confidentiality and integrity protection for messages passing between adjacent signalling nodes. It cannot provide security for messages over greater scope than one signalling hop, nor can it assert whether the message originator has the right to request whatever actions the message implies (i.e. authorisation). These issues must be addressed within the MIH service itself.

Peer authentication: GIST uses the authentication mechanism within the channel security protocol to provide per-peer authentication. The default in the current version is limited to TLS authentication based on certificates; the authenticated peer identities are visible to the MIH service and may be used as part of an authorisation decision. It may be desirable to extend the set of authentication mechanisms to improve the applicability and
ease of deployment in the MIH problem space.

Security protocol selection: GIST provides a handshake which can negotiate one option from a set of security protocols, in a way which is secure against bidding-down attacks and denial-of-service attacks. (This is distinct from any option negotiation within the possible security protocols themselves.) Normally the details of this negotiation would be invisible to MIH services; however, there may be a need for specific security policies to ensure that the level of protection negotiated is appropriate.

Denial of service mitigation: GIST includes mechanisms to mitigate denial of service (flooding) attacks within the GIST layer, and can also negotiate channel protection to enable the rejection of signalling messages from unauthenticated peers. An MIH service using GIST in this way only needs to care about flooding or other DoS attacks launched by authenticated peers.

An MIH service designed to run over GIST needs to specify the required security policy that should be imposed at the GIST level, and also needs to define how authorisation is defined and denial of service attacks are prevented at the service level. These latter problems are clearly closely related.

6. Conclusions

This draft has presented an outline solution for supporting common MIH signalling functions using GIST. Most of the required functionality is already present in GIST; some options for GIST extensions were also identified.

Most of the functionality of GIST is obtained by the integration of existing security and transport protocols, and conceptually it would be possible to design MIH services so they just re-use these protocols directly. Compared to this, the advantages of a GIST-based approach include:

- integrated negotiation of which security and transport protocols to use, including protocol options and port numbers (no need to fix on a particular choice);

- negotiation which is resistant to denial of service and bidding down attacks;

- peer discovery and protocol negotiation take place in a single round trip, minimising latency;
Further development of a GIST-based solution depends on some more detailed analysis of the MIH problem space. Issues which have been noted in this draft include:

- whether additional security protocol support will ease deployment in typical MIH environments;
- what are the more detailed scenarios for signalling endpoint location (with implications for how much sophistication is needed in the discovery process); and
- how to fit the MIH service structure into the overall NSIS protocol suite framework, for example as one or several signalling applications.

7. Informative References


[7] Fu, X., "General Internet Signaling Transport (GIST) over


Appendix A. Acknowledgements

The authors would like to thank the participants in the NSIS and MIPSHOP working groups in the IETF and 802.21 in the IEEE for many discussions on signalling protocols in general, and MIH services in particular, which have informed the content of this draft.

Robert Hancock and Eleanor Hepworth are partly funded by Ambient Networks, a research project supported by the European Commission under its Sixth Framework Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Ambient Networks.
project or the European Commission.
Authors’ Addresses

Robert Hancock
Roke Manor Research
Old Salisbury Lane
Romsey, Hampshire  SO51 0ZN
UK

Email: robert.hancock@roke.co.uk

Eleanor Hepworth
Roke Manor Research
Old Salisbury Lane
Romsey, Hampshire  SO51 0ZN
UK

Email: eleanor.hepworth@roke.co.uk
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.