FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

FIPA Nomadic Application Support Overview Specification

Document title	FIPA Nomadic Application Support Overview Specification		
Document number	OC00066D	**Document source**	FIPA Nomadic Application Support
Document status	Obsolete	**Date of this status**	2001/08/10
Supersedes	None		
Contact	fab@fipa.org		
Change history			
2000/08/04	Document made obsolete by FIPA00014		
2001/08/10	Line numbering added		

Geneva, Switzerland

Foreword

The Foundation for Intelligent Physical Agents (FIPA) is an international organization that is dedicated to promoting the industry of intelligent agents by openly developing specifications supporting interoperability among agents and agent-based applications. This occurs through open collaboration among its member organizations, which are companies and universities that are active in the field of agents. FIPA makes the results of its activities available to all interested parties and intends to contribute its results to the appropriate formal standards bodies.

The members of FIPA are individually and collectively committed to open competition in the development of agent-based applications, services and equipment. Membership in FIPA is open to any corporation and individual firm, partnership, governmental body or international organization without restriction. In particular, members are not bound to implement or use specific agent-based standards, recommendations and FIPA specifications by virtue of their participation in FIPA.

The FIPA specifications are developed through direct involvement of the FIPA membership. The status of a specification can be either Preliminary, Experimental, Standard, Deprecated or Obsolete. More detail about the process of specification may be found in the FIPA Procedures for Technical Work. A complete overview of the FIPA specifications and their current status may be found in the FIPA List of Specifications. A list of terms and abbreviations used in the FIPA specifications may be found in the FIPA Glossary.

FIPA is a non-profit association registered in Geneva, Switzerland. As of January 2000, the 56 members of FIPA represented 17 countries worldwide. Further information about FIPA as an organization, membership information, FIPA specifications and upcoming meetings may be found at http://www.fipa.org/.

Contents

1 Scope

This document is part of the FIPA specifications and deals with agent middleware to support applications in nomadic environment. This specification also forms part of the FIPA Nomadic Application Support Specification and gives an overview of the Nomadic Application Support area.

2 General Analysis

2.1 Overview

The results of current developments in both wireless data communications and mobile computers are being combined to facilitate a new trend: *nomadic computing*. Compared to today's traditional distributed systems, the nomadic computing environment is very different in many respects. Bandwidth, latency, delay, error rate, quality of display and other non-functional parameters may change dramatically when a nomadic end-user moves from one location to another and thus from one computing environment to another, for example, from a wireline LAN to a UMTS network. The variety of mobile workstations, handheld devices and smart phones, which allow nomadic end-users to access Internet services, is increasing rapidly. The capabilities of mobile devices range from very low performance equipment (such as PDAs) up to high performance laptop PCs. All these devices create new demands for adaptability of Internet services. For example, PDAs cannot display properly high quality images and as nomadic end-users will be charged based on the amount of data transmitted over the GPRS-UMTS network, they will have to pay for bits that are totally useless to them.

Confronted with these circumstances, the nomadic end-user would benefit from having the following functionality provided by the infrastructure: information about expected performance, agent monitoring and controlling the transfer operations, and adaptability.

The ability to automatically adjust to changes in a transparent and integrated fashion is essential for *nomadicity*; nomadic end-users are usually professionals in areas other than computing. Furthermore, today's mobile computer systems are already very complex to use as productivity tools. Thus, nomadic end-users need all the support that a FIPA agent-based distributed system can deliver and adaptability to the changes in the environment of nomadic end-users is an important issue.

FIPA uses the Wireless Application Protocol (WAP) [WAP99] as its wireless Message Transport Protocol (MTP - see [FIPA00076]). The WAP Forum has developed industry-wide specifications for low bandwidth wireless services (such as GSM, GPRS, etc.) and wireless devices (such as mobile telephones and personal digital assistants). The WAP specifications address the characteristics of wireless networks by adapting low bandwidth wireless services and low-end mobile devices to the special requirements of information services. The WAP specification defines a set of standard components that can be used in agent message communication, such as standard data formats and standard data communication protocols.

The adaptation of applications to various nomadic computing environments is an important area. There are several tasks that agents need to carry out during application adaptation:

1. Selection of MTP and Message Transport Connection (MTC) to be used for agent communication.

2. Selection of an ACL and content language representation to be used for agent communication.

3. Provision of support for application agents to carry out adaptation of application data, such as still images, video and audio, XML, etc. Today's Internet application data (such as multimedia content) are designed with high performance desktop PCs and high quality displays in mind. Therefore, the application data is frequently unsuitable for nomadic computing using wireless wide-area networks and low performance mobile devices, and hence requires modification.

4. Communication between agents performing adaptation.

The FIPA Nomadic Application Support specifications define agent middleware to:

Monitor [FIPA00062] and control [FIPA00063] an MTP and the underlying MTC, and,

An ontology [FIPA00065] for representing the quality of service of the Message Transport Service (MTS - see [FIPA00065]) in the context of nomadic application support.

In addition, this specification gives examples of the use of the above scenarios in section *3, Scenarios*.

2.2 Monitoring and Controlling Quality of Service

The functions required to carry out monitoring and controlling for quality of service can be split into several specific tasks:

1. Observing the quality of service of MTPs and MTCs,
2. Measuring (if there are no other means to obtain the required information) the quality of service of an MTP and MTC,
3. Collecting information from the observing and measuring sources,
4. Analysing the information, and,
5. Controlling an MTC and selecting an MTP.

Based on this division, the agent middleware consists of the following logical agents (see *Figure 1*):

A **Monitor Agent** (MA - see [FIPA00062]) which carries out tasks 1 through 4, and,

A **Control Agent** (CA - see [FIPA00063]) which carries out task 5.

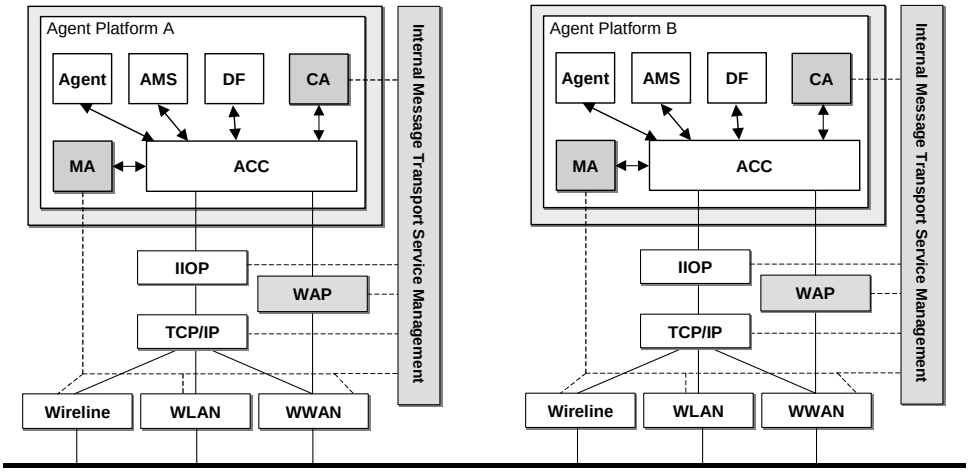

Figure 1: Reference Model of Agent Adaptation

The most appropriate configuration of an MA and a CA is that there is at least one pair in each AP involving adaptation. The MA may measure the actual quality of service of an MTC, if the network running an MTC does not provide users with required performance data[1].

An MA may:

Consist of network-service-specific components that collect raw performance data at fixed intervals,

Provide a repository for the measurement data collected,

Perform first level analysis of the collected data, and,

[1] The way this actual measurement is performed is not a subject of standardisation within FIPA.

Send the results of the analysis to CA, if requested to do so.

A CA may:

May manage (establish, close, suspend, activate, etc.) an MTC[2].

In some cases there is a need for MAs and CAs in heterogeneous APs to communicate with each other; therefore, interaction protocols and ontologies to achieve this are specified in this document.

2.3 Negotiation of Message Transport Requirements

There are several mechanisms that can determine the MTP, message representation and content language to use between communicating entities:

Communicating entities know a peer entity's preferences beforehand and use them.

The activating entity tries to use a method and if the peer entity is not capable of using the suggested method, then the activating entity may try another one (and so on).

The communicating entities negotiate about a method to be used.

2.3.1 Negotiation About Message Transport Protocols

Previous FIPA specifications have implicitly assumed that the MTC is operational all the time (meaning that the MTC has been established before the agent message exchange and that it is reliable). However, this is not always the case within a nomadic environment.

A CA can activate the selection of an MTP or an agent can propose an MTP to a CA and it is the responsibility of the CA to either accept or reject the proposal based on whether it is possible to use the proposed MTP. CAs negotiate with peer CAs to use proposed MTPs which is illustrated in *Figure 2*.

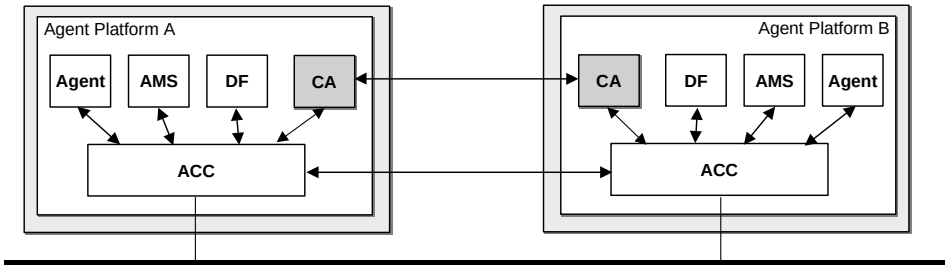

Figure 2: Control Agents Negotiating About a Message Transport Protocol

CAs use the `FIPA-Propose` interaction protocol [FIPA00036] and the `use` action [FIPA00063] to negotiate about an MTP. An example negotiation is given in section *3.1, Negotiating Message Transport Protocols*.

2.3.2 Negotiation About Message Representation

In the environment of nomadic applications, it may be necessary to switch from one ACL representation to another; for example, when a mobile host roams from a wireline network to a wireless network. Application agents may use the `FIPA-Propose` interaction protocol and the `use` action to negotiate about the representation of ACL. Examples of this negotiation are given in section *3.2, Negotiating Message Representation*.

[2] The way that management actions are executed is not a subject of standardisation within FIPA.

3 Scenarios

3.1 Negotiating Message Transport Protocols

This example shows a scenario, where an application agent requests the use of either the WAP MTP [FIPA00076] or a proprietary MTP (for example, `x.uh.mdcp`). The message flow of a successful negotiation is illustrated in *Figure 3*.

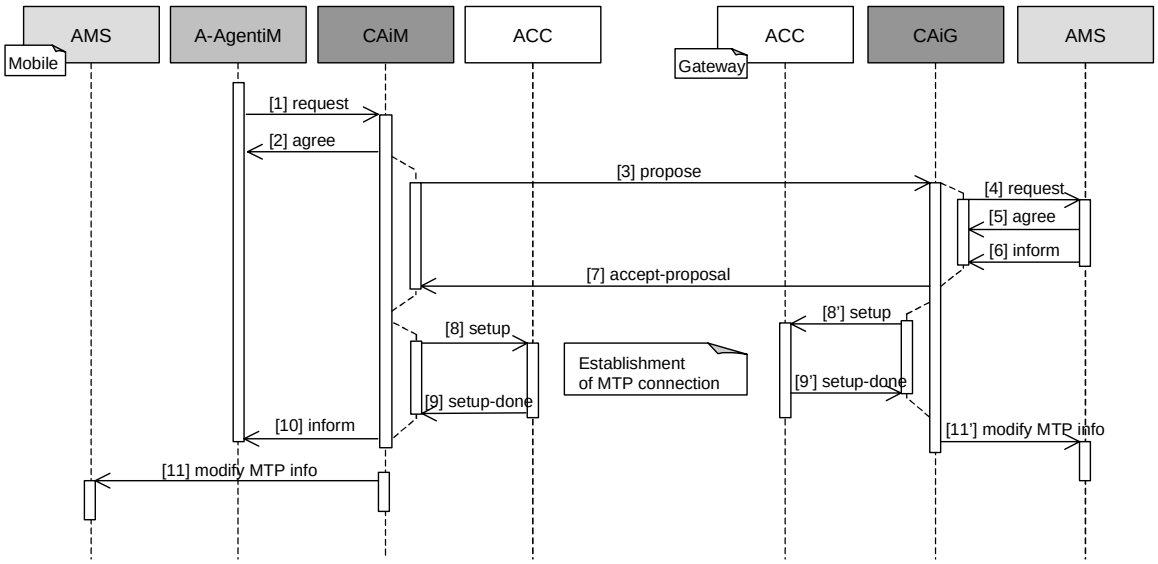

Figure 3: Flow of Message Transport Protocol Negotiation

1. Message 1 `request`: An application agent issues a request to the CA to activate either the `fipa.mts.mtp.wap.std` or `x.uh.mdcp` MTPs.

```
(request
  :sender
    (agent-identifier[3]
      :name A-AgentiM@mobile.com[4])
  :receiver (set
    (agent-identifier
       :name CaiM@mobile.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL0
  :protocol FIPA-Request
  :content
    (action
      (agent-identifier
        :name CAiM@mobile.com)
      (activate (sequence
        (transport-protocol
          :name x.uh.mdcp)
        (transport-protocol
          :name fipa.mts.mtp.wap.std
          :dest-addr wap://gateway.com:1234/acc)))))
```

[3] In most of the examples in this section, the `:address` parameters of AIDs have been omitted for clarity, except where absolutely necessary.

[4] In all of the examples in this specification, the suffix of `iM` in an agent's name represents a mobile host, that is, an agent that is located on a mobile AP. Similarly, the suffix `iG` represents a gateway host and the suffix `iF` represents a fixed network host.

2. Message 2 `agree`: The CA agrees to activate an MTP. The decision to agree or disagree to activate an MTP might be based on the internal state of the CA (that is, the CA knows whether a requested MTP can be activated or not) or the CA might ask for an AP description from an AMS (see messages 4, 5 and 6 as an example).

```
(agree
  :sender
    (agent-identifier
      :name CAiM@mobile.com)
  :receiver (set
    (agent-identifier
      :name A-AgentiM@mobile.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL0
  :protocol FIPA-Request
  :content
    ((action
      (agent-identifier
        :name CAiM@mobile.com))
     (activate (sequence
       (transport-protocol
         :name x.uh.mdcp)
       (transport-protocol
         :name fipa.mts.mtp.wap.std
         :dest-addr wap://gateway.com:1234/acc))))
    true))
```

3. Message 3 `propose`: The CA in the mobile host proposes to its peer CA in the gateway host that either the `fipa.mts.mtp.wap.std` or `x.uh.mdcp` MTPs should be used in communication between the APs.

```
To: (agent-identifier :name CAiG@gateway.com)
From: (agent-identifier :name CAiM@mobile.com)
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T100900000

(propose
  :sender
    (agent-identifier
      :name CAiM@mobile.com)
  :receiver (set
    (agent-identifier
      :name CAiG@gateway.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL0
  :protocol FIPA-Propose
  :content
    ((action
      (agent-identifier
        :name CAiM@mobile.com)
      (use
        (transports
          :send (sequence
            (transport-protocol
              :name x.uh.mdcp)
            (transport-protocol
              :name fipa.mts.mtp.wap.std))
          :recv (sequence
            (transport-protocol
              :name x.uh.mdcp)
            (transport-protocol
              :name fipa.mts.mtp.wap.std)))))
    true))
```

4. Message 4 `request`, message 5 `agree` and message 6 `inform`: The CA in the gateway host requests the AP description from the local AMS (see [FIPA00023]) to determine whether the `x.uh.mdcp` or `fipa.mts.mtp.wap.std` MTPs are supported. The AMS informs the CA that both MTPs are supported and the CA decides to use `fipa.mts.mtp.wap.std` MTP based on the current quality of service requirements of the MTC.

```
(request
  :sender
    (agent-identifier
      :name CAiG@gateway.com)
  :receiver (set
    (agent-identifier
      :name ams@gateway.com))
  :ontology FIPA-Agent-Management
  :language FIPA-SL0
  :protocol FIPA-Request
  :content
    (action
      (agent-identifier
        :name ams@gateway.com)
     get-description))

(agree
  :sender
    (agent-identifier
      :name ams@gateway.com)
  :receiver (set
    (agent-identifier
      :name CAiG@gateway.com))
  :ontology FIPA-Agent-Management
  :language FIPA-SL0
  :protocol FIPA-Request
  :content
    ((action
      (agent-identifier
        :name ams@gateway.com)
      get-description)
    true))

(inform
  :sender
    (agent-identifier
      :name ams@gateway.com
      :addresses (sequence http://gateway.com/acc))
  :receiver (set
    (agent-identifier
      :name CAiG@gateway.com
      :addresses (sequence http://gateway.com/acc)))
  :ontology FIPA-Agent-Management
  :language FIPA-SL0
  :protocol FIPA-Request
  :content
    (ap-description
      :name sonera-platform
      :transport-profile
        (ap-transport-description
          :available-mtps
            (set
              (mtp-description
                  :profile fipa.profile.mts.alpha
                  :mtp-name fipa.mts.mtp.iiop.std
                  :addresses (sequence iiop://gateway.com/acc))
              (mtp-description
```

```
                    :profile fipa.profile.mts.beta
                    :mtp-name fipa.mts.mtp.wap.std
                    :addresses (sequence wap://gateway.com:1234/acc))
                (mtp-description
                    :profile x.uh.profile
                    :mtp-name x.uh.mdcp
                    :addresses (set mdcp://gateway.com/acc))))))
```

5. Message 7 `accept-proposal`: The CA in the gateway host accepts the proposal to use the `fipa.mts.mtp.wap.std` MTP and sends the response to the CA in the mobile host informing it about the preferred MTP.

```
To: (agent-identifier :name CAiM@mobile.com)
From: (agent-identifier :name CAiG@gateway.com)
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T100900000

(accept-proposal
  :sender
    (agent-identifier
      :name CAiG@gateway.com)
  :receiver (set
    (agent-identifier
      :name CAiM@mobile.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL0
  :protocol FIPA-Propose
  :content
    (action
      (agent-identifier
        :name CAiM@mobile.com)
      (use
        (transports
          :send (sequence
            (transport-protocol
              :name x.uh.mdcp)
            (transport-protocol
              :name fipa.mts.mtp.wap.std))
          :recv (sequence
            (transport-protocol
              :name x.uh.mdcp)
            (transport-protocol
              :name fipa.mts.mtp.wap.std)))))
    (transports
      :send (sequence
        (transport-protocol
          :name fipa.mts.mtp.wap.std))
      :recv (sequence
        (transport-protocol
          :name fipa.mts.mtp.wap.std))))
```

6. Messages 8 and 8' `setup`: The CAs request their respective ACCs to setup the `fipa.mts.mtp.wap.std` MTP. This is an implementation issue.

7. Message 9 and 9' `setup-done`: The ACCs inform their respective CAs that the `fipa.mts.mtp.wap.std` MTP has been established between the mobile host and the gateway host.

8. Message 10 `inform`: The CA informs the application agent that the MTC is established.

```
(inform
  :sender
    (agent-identifier
      :name CAiM@mobile.com)
  :receiver (set
    (agent-identifier
      :name A-AgentiM@mobile.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL0
  :protocol FIPA-Request
  :content
    (result
      (action
        (agent-identifier
          :name CaiM@mobile.com)
      (activate (sequence
        (transport-protocol
          :name x.uh.mdcp)
        (transport-protocol
          :name fipa.mts.mtp.wap.std
          :dest-addr wap://gateway.com:1234/acc))))
    (transport-protocol
      :name fipa.mts.mtp.wap.std))
```

3.2 Negotiating Message Representations

This example shows a scenario where an application agent in a mobile host proposes to its peer application agent in a fixed host the use of the `fipa.acl.rep.bitefficient.std` representation of ACL [FIPA00069] for their communication. The message flow is illustrated in *Figure 4*.

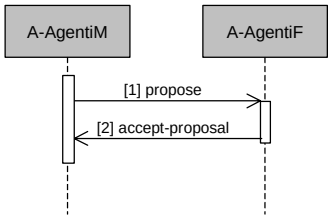

Figure 4: Flow of Message Representation Negotiation

1. Message (1) propose: The agent in the mobile host proposes the use of the `fipa.acl.rep.bitefficient.std` representation of ACL.

```
To: (agent-identifier :name A-AgentiF@fixed.com
                      :addresses (sequence iiop://fixed.com/acc))
From: (agent-identifier :name A-AgentiM@mobile.com
                        :addresses (sequence wap://mobile.com:1234/acc))
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T101000000

(propose
  :sender
    (agent-identifier
      :name A-AgentiM@mobile.com)
  :receiver (set
    (agent-identifier
      :name A-AgentiF@fixed.com))
  :ontology FIPA-Message-Representation
  :language FIPA-SL0
```

```
  :protocol FIPA-Propose
  :content
    ((action
      (agent-identifier
        :name A-AgentiM@mobile.com)
      (use
        (msg-rep-selection
          :send (sequence
            (msg-representation
              :name fipa.acl.rep.bitefficient.std))
          :recv (sequence
            (msg-representation
              :name fipa.acl.rep.bitefficient.std)))))
      true))
```

2. Message 2 `accept-proposal`: The agent in the fixed host accepts the proposal.

```
To: (agent-identifier :name A-AgentiM@mobile.com
                      :addresses (sequence wap://fixed.com:1234/acc))
From: (agent-identifier :name A-AgentiF@iiop://fixed.com
                        :addresses (sequence iiop://fixed.com/acc))
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T101000000

(accept-proposal
  :sender
    (agent-identifier
      :name A-AgentiF@fixed.com)
  :receiver (set
    (agent-identifier
      :name A-AgentiM@mobile.com))
  :ontology FIPA-Message-Representation
  :language FIPA-SL0
  :protocol FIPA-Propose
  :content
    (action
      (agent-identifier
        :name A-AgentiM@mobile.com)
      (use
        (msg-rep-selection
          :send (sequence
            (msg-representation
              :name fipa.acl.rep.bitefficient.std))
          :recv (sequence
            (msg-representation
              :name fipa.acl.rep.bitefficient.std))))
      (msg-rep-selection
        :send (sequence
          (msg-representation
            :name fipa.acl.rep.bitefficient.std))
        :recv (sequenc
          (msg-representation
            :name fipa.acl.rep.bitefficient.std))))
```

3.3 Message Exchange Over a WAP Message Transport Protocol

Figure 5 refers to a reference architecture for message exchange in context of nomadic applications. Messages between the mobile host and gateway host are delivered mainly using the `fipa.mts.mtp.wap.std` MTP and messages between gateway host and other APs in the fixed network are delivered using the `fipa.mts.mtp.iiop.std` MTP (see [FIPA00075]).

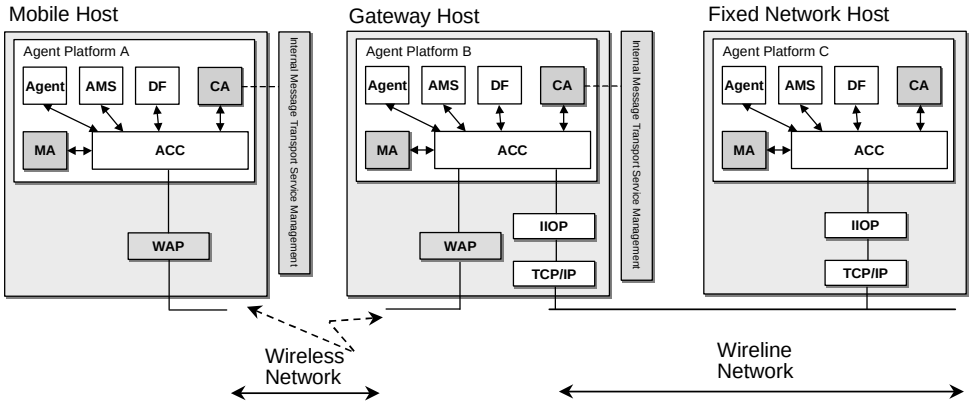

Figure 5: Gateway-Based Nomadic Application Architecture

3.3.1 Message Exchange Activation by an Agent in a Mobile Host

This example shows the scenario where an agent in a mobile host has a WAP address and an agent in fixed host has an IIOP address. In this example, there are three specific APs involved; one running in a mobile host, one running in a gateway host and the last one running in a host situated in a fixed network which represents the rest of the network. An example of the flow of a message exchange is illustrated in *Figure 6*.

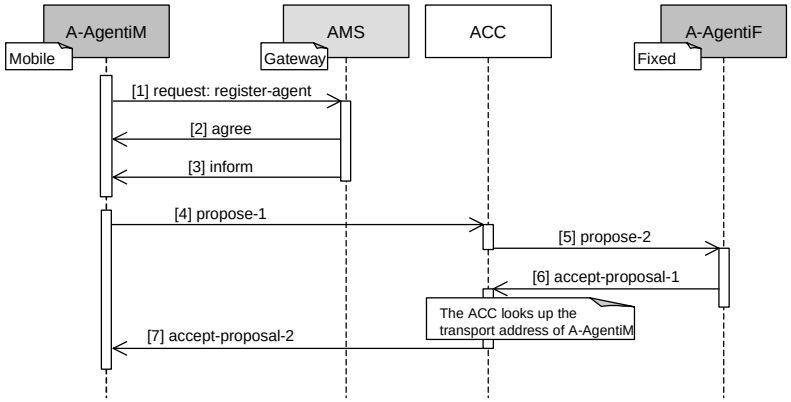

Figure 6: Mobile Originated Message Exchange Over Gateway Host

1. Message 1 `request`, message 2 `agree` and message 3 `inform`: In order to be reachable from an AP operating in a fixed network environment, an agent in the mobile host must register with the AP running in the gateway host. Subsequently, the ACC in the gateway host AP can forward messages intended for the agent operating in the mobile host to the ACC.

```
To: (agent-identifier :name ams@gateway.com
                      :addresses (sequence wap://gateway.com:1234/acc))
From: (agent-identifier :name A-AgentiM@mobile.com
                        :addresses (sequence wap://mobile.com:1234/acc))
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T101000000
```

```
(request
  :sender
    (agent-identifier
      :name A-AgentiM@mobile.com)
  :receiver (set
    (agent-identifier
      :name ams@gateway.com))
  :language FIPA-SL0
  :protocol FIPA-Request
  :ontology FIPA-Agent-Management
  :content
    (action
      (agent-identifier
        :name ams@gateway.com)
      (register
        (ams-agent-description
          :name
            (agent-identifier
              :name A-AgentiM@mobile.com
              :addresses (sequence wap://mobile.com:1234/acc))
          :state active))))
```

The AMS informs A-AgentiM that registration was completed successfully and after registration, A-AgentiM can be reached via the gateway host using, for example, the following To parameter:

```
To: (agent-identifier :name A-AgentiM@mobile.com
                      :addresses (sequence iiop://gateway.com/acc))
```

If the gateway host is not operational, then the direct WAP address (wap://mobile.com:1234/acc) could be used.

2. Message 4 propose–1: A-AgentiM sends a propose message to A-AgentiF. In the From parameter, A-AgentM informs A-AgentiF that its primary return address is its address in the gateway host.

```
To: (agent-identifier :name A-AgentiF@fixed.com
                      :addresses (sequence iiop://fixed.com/acc))
From: (agent-identifier :name A-AgentiM@mobile.com
                  :addresses (sequence iiop://gateway.com/acc wap://mobile.com:1234/acc)
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T101000000

(propose
  :sender
    (agent-identifier
      :name A-AgentiM@mobile.com)
  :receiver (set
    (agent-identifier
      :name A-AgentiF@fixed.com))
  :language FIPA-SL0
  :content
    (action
      (agent-identifier
        :name A-AgentiM@mobile.com)
      (compress-data (> object-size 1kb)))
```

The ACC in the mobile host forwards the message to the ACC in the gateway host using fipa.mts.mtp.wap.std MTP[5].

[5] The actual way in which the is achieved in not a subject of standardisation within FIPA.

607 3. Message 5 `propose-2`: The ACC in the gateway host forwards the message to `A-AgentiF` using
608 `fipa.mts.mtp.iiop.std` MTP. The ACC may change the encoding of the message.
609
610

4. Message 6 `accept-proposal–1`: `A-AgentiF` accepts `A-AgentiM`'s proposal by sending an `accept-proposal` message to `A-AgentiM` using its gateway host address.

```
To: (agent-identifier :name A-AgentiM@mobile.com
                      :addresses (sequence iiop://gateway.com/acc))
From: (agent-identifier :name A-AgentiF@iiop://fixed.com
                        :addresses (sequence iiop://fixed.com/acc))
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T101100000

(accept-proposal
  :sender
    (agent-identifier
      :name A-AgentiF@fixed.com)
  :receiver (set
    (agent-identifier
      :name A-AgentiM@mobile.com))
  :language FIPA-SL0
  :content
    ((action
      (agent-identifier
        :name A-AgentiM@mobile.com)
      (compress-data (> object-size 1kb)))
     true)
```

5. Message 7 `accept-proposal–2`: The ACC in the gateway host forwards the message to the ACC in the mobile host using the `fipa.mts.mtp.wap.std` MTP. The ACC may change the encoding of the message.

3.3.2 Message Exchange Termination to an Agent in a Mobile Host

This example shows the scenario where an agent in a fixed host activates a conversation. The message flow is illustrated in *Figure 7*.

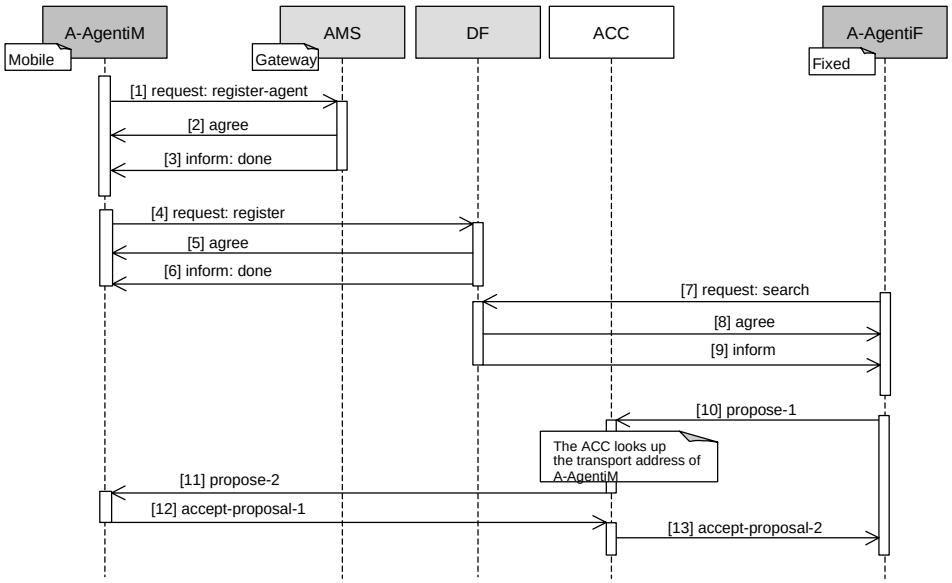

Figure 7: Mobile Terminated Message Exchange Over Gateway Hosts

1. Message 1 `request`, message 2 `agree` and message 3 `inform`: See section *3.3.1, Message Exchange Activation by an Agent in a Mobile Host*.

2. Message 4 `request`: `A-AgentiM` needs to register its services with the DF in the gateway host in order to be able to publicise its services even when the mobile host itself is disconnected from the fixed network.

```
To: (agent-identifier :name df@gateway.com)
From: (agent-identifier :name A-AgentiM@mobile.com)
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T101100000

(request
  :sender
    (agent-identifier
      :name A-AgentiM@mobile.com)
  :receiver (set
    (agent-identifier
      :name df@gateway.com))
  :ontology FIPA-Agent-Management
  :language FIPA-SL0
  :protocol FIPA-Request
  :content
    (action
      (agent-identifier
        :name df@gateway.com)
      (register
        (df-agent-description
          :name
            (agent-identifier
              :name A-AgentiM@mobile.com
              :addresses (sequence iiop://gateway.com/acc wap://mobile.com:1234/acc))
          :services (set
            (service-description
              :name Field-Warrior
              :type field-information
              :ontology (set field-service)
              :properties (set
                (property
                  :name availability
                  :value 24h))))
          :language (set FIPA-SL0))))
```

3. Message 5 `agree` and message 6 `inform`: The DF in the gateway host AP informs `A-AgentiM` that registration was successful.

```
(inform
  :sender
    (agent-identifier
      :name df@gateway.com)
  :receiver (set
    (agent-identifier
      :name A-AgentiM@mobile.com))
  :language FIPA-SL0
  :protocol FIPA-Request
  :ontology FIPA-Agent-Management
  :content
    (done
      (action
        (agent-identifier :name df@gateway.com)
      (register
        (df-agent-description
          :name
            (agent-identifier
              :name A-AgentiM@mobile.com
              :addresses (sequence iiop://gateway.com/acc wap://mobile.com:1234/acc))
          :services
```

```
            (service-description (set
              :name Field-Warrior
              :type field-information
              :ontology field-service
              :properties (set
                (property
                  :name availability
                  :value 24h))))
          :language (set FIPA-SL0)))))
```

4. Message 7 `request`, message 8 `agree` and message 9 `inform`: When `A-AgentiM` needs the `Field-Warrior` service, it searches the gateway host DF which informs it that `A-AgentiM` offers such a service (see [FIPA00023]).

5. Message 10, 11, 12 and 13: The messages used and the message flow are similar to the example in section *3.3.1, Message Exchange Activation by an Agent in a Mobile Host*.

4 Informative Annex A - Paramedic Scenario

This section illustrates some of the important issues of nomadic application support, using a paramedic application as an example.

4.1 Overview

A paramedic team has several working environments:

An emergency dispatch centre, which is covered by the hospital ATM network,

A geographical area, which is wireless wide-area network (e.g. GPRS), and,

One or more hospitals, which are provided with a wireless local-area network.

When in transit, the paramedic computers are attached to docking stations residing in ambulances. At the dispatch centre, the docking stations are connected to the ATM network. The paramedic application comprises the following services:

Retrieval of a patient's personal information, such as name, address, phone, and relatives,

Retrieval of the patient's medical histories,

Support for paramedic workers, and,

Informing the hospital receiving the patient about the patient's current injury or illness and medical care given so far.

There are several application agents: Paramedic Support Agents (PSAs) working in the paramedic computers, Dispatching Support Agent (DSA) working at the dispatch centre system, and the Hospital First Aid Support Agent (HFASA) working at the hospital system.

The dispatch centre receives a call regarding a man who has severe chest pain; the symptom of an acute myocardial infarct. The caller identifies the man and gives his personal identification number to the dispatcher. The dispatcher alerts the paramedic team and informs the DSA about the address where the patient is located and his personal identification number. The DSA simultaneously informs the PSA about the address of the attack (and possibly some additional information about the environment of the heart attack) and queries the patient's medical history. Since the results of the query to a local hospital are received before the paramedic unit is dispatched, the DSA (in co-operation with the PSA) begins to load the patient's personal information and medical history into the paramedic computers. The medical history includes several items of text-based information. The transmission time to load the information via the ATM network to the paramedic computers (which are currently docked at the dispatch centre) is less than a second. Before the ambulance leaves the dispatch centre, the docking station is detached from the ATM network and is connected to the wireless wide-area network.

While the ambulance is approaching the location of the incident, the DSA receives more relevant results of the query of the medical histories such as the latest heart operation of the patient. The medical history comprises several parts of textual information and several images and the DSA begins loading the information. As the loading takes place when the ambulance is in motion, the DSA finds out that the quality of transport service is too low for loading some textual parts and any of the images of the medical history. It would take at least 40 minutes to download the images. Therefore, the DSA informs the PSA that images are not required for the paramedic unit. During downloading, the ambulance drives into a tunnel that causes the wireless link to be disconnected. After the tunnel, a CA re-establishes the connection and downloading continues.

At the scene, the ambulance is stationary and the quality of transmission service increases to a level at which the DSA is able to load the most relevant images (the ECGs) using an efficient compression method which is negotiated

between the DSA and the PSA to the paramedic computer. The paramedic team detaches the computers from the docking station and carries them to the patient.

The paramedic team realises that they need the assistance of a medical expert located at the university hospital to stabilise the patient's condition. Therefore, they attach electrodes to the patient and the PSA starts transmitting the data of measurement such as SpO2 (oxygen saturation), cardiac rhythm, ECG, end tidal CO2 and temperature to the hospital. After successfully stabilising the patient's condition, the paramedic team moves the patient to the ambulance and sets off for the hospital. As the quality of the transport service decreases because of the motion, the PSA finds out that not all the on-going measurement data can be transmitted on-line to the hospital. Therefore, the PSA decides to transmit the most relevant data (SpO2 and cardiac rhythm). The PSA stores the rest of the data (ECG, end tidal CO2 and temperature) into a cache of the paramedic computer.

After the ambulance arrives at the hospital, the patient is transferred immediately to an operating room. Simultaneously, the paramedic team connects their paramedic computer to the wireless LAN of the hospital and the PSA transmits (in co-operation with the HFASA) all the measurement data to the hospital's system. A surgeon retrieves and analyses the measurement data before surgery.

This example illustrates a future agent-based distributed system that offers its services at the best obtainable quality of service in a wide variety of environments. A possible agent architecture is illustrated in *Figure 8* which refers to three separate APs: *Dispatch*, *Gateway* and *Paracom*. In addition, there are several hospital APs which are not illustrated.

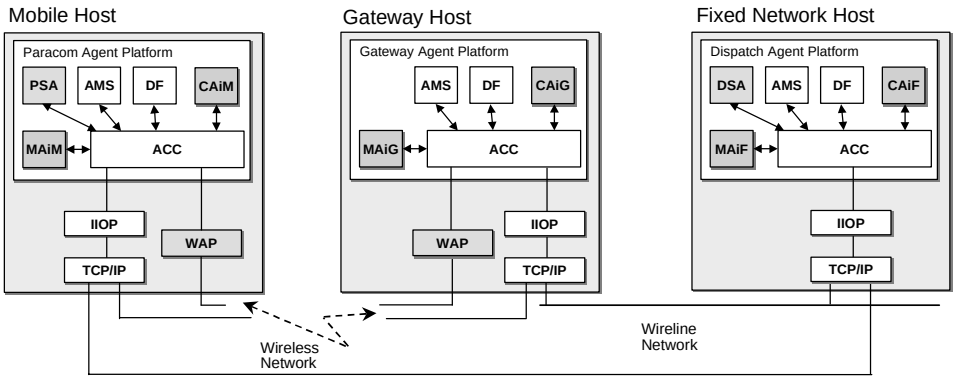

Figure 8: Paramedic Scenario Architecture

The agents in the scenario are:

`MAiM`, `MAiG` and `MAiF` are MAs which monitor the quality of the communication service,

`CAiM`, `CAiG` and `CAiF` are CAs which manage the establishment, teardown, suspension, activation, etc. of the connection between the PAs. The MA informs application agents about the status and changes of the network services.

When the mobile host is connected either to the ATM network or to the wireless LAN, the `fipa.mts.mtp.iiop.std` MTP is used directly between the *Paracom* AP and the *Dispatch* AP. When the mobile host is connected to the wireless WAN, all agent message communication takes place through the gateway host. The `fipa.mts.mtp.wap.std` MTP is primarily used between the *Paracom* AP and the *Gateway* AP. The `fipa.mts.mtp.iiop.std` MTP is used between the *Gateway* AP and the *Dispatch* AP.

4.2 Seamless Roaming

The Seamless Roaming scenario describes the process, when the paramedic computer roams from the ATM network to the UMTS network. The scenario is split into following events:

Disconnection and reconnection of MTCs,

Negotiation of MTPs, and,

Negotiation of message representations.

4.2.1 Disconnection and Reconnection of an Message Transport Connection

The message exchange between the agents is illustrated in *Figure 9*.

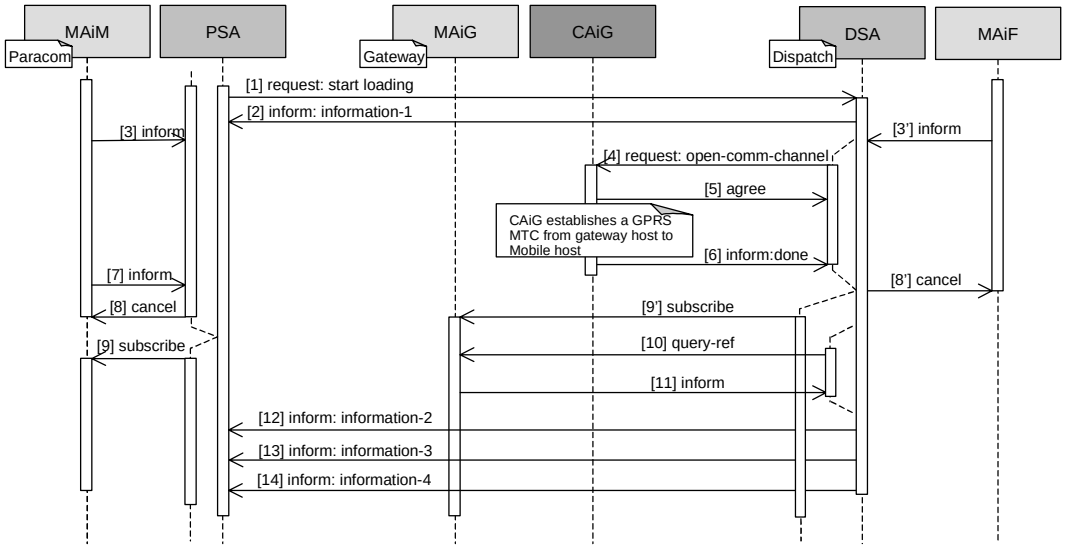

Figure 9: Disconnection and Reconnection of an Message Transport Connection

1. Message 1 `request`: The PSA starts loading data from the DSA by sending a `request` message. This message is application specific and thus not shown here.

2. Message 2 `inform`: The DSA starts sending information by first sending an `inform` message.

3. Messages 3 and 3' `inform`: `MAiM` (/ `MAiF`) informs the PSA (/DSA) that the ATM connection has broken.

```
(inform
  :sender
    (agent-identifier
      :name MAiM@paracom.com)
  :receiver (set
    (agent-identifier
      :name PSA@paracom.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL2
  :protocol FIPA-Subscribe
  :content
    (= (iota ?x
      (qos-information
        (comm-channel
          :name ATM
          :target-addr iiop://dispatch.com/acc)
      (qos
        :status ?x)))
     disconnected))
```

4. Message 4 `request`: The DSA requests `CAiG` to open a wireless wide-area MTC and `CAiG` agrees.

```
To: (agent-identifier :name CAiG@gateway.com)
From: (agent-identifier :name DSA@dispatch.com)
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T101100000

(request
  :sender
    (agent-identifier
      :name DSA@dispatch.com)
  :receiver (set
    (agent-identifier
      :name CAiG@gateway.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL0
  :protocol FIPA-Request
  :content
    (action
      (agent-identifier
        :name CAiG@gateway.com)
      (open-comm-channel
        (comm-channel
          :name GPRS
          :target-addr iiop://paramedic.com/acc))))
```

5. Message 5 `agree`: `CAiG` agrees that it will try to open the GPRS connection.

```
To: (agent-identifier :name DSA@dispatch.com)
From: (agent-identifier :name CAiG@gateway.com)
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T101200000

(agree
  :sender
    (agent-identifier
      :name CAiG@gateway.com)
  :receiver (set
    (agent-identifier
      :name DSA@dispatch.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL0
  :protocol FIPA-Request
  :content
    ((action
      (agent-identifier
        :name CAiG@gateway.com)
      (open-comm-channel
        (comm-channel
          :name GPRS
          :target-addr iiop://paramedic.com/acc))))
    true)
```

Next `CAiG` establishes a GPRS MTC from the gateway host to the mobile host. This is an implementation issue.

6. Message 6 `inform`: After successful establishment, `CAiG` informs the DSA.

```
To: (agent-identifier :name DSA@dispatch.com)
From: (agent-identifier :name CAiG@gateway.com)
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T101200000

(inform
  :sender
    (agent-identifier
      :name CAiG@gateway.com)
  :receiver (set
    (agent-identifier
      :name DSA@dispatch.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL0
  :protocol FIPA-Request
  :content
    (done
      (action
        (agent-identifier
          :name CAiG@gateway.com))
      (open-comm-channel
        (comm-channel
          :name GPRS
          :target-addr iiop://paramedic.com/acc)))))
```

7. Message 7 `inform`: `MAiM` informs the PSA that a new MTC has been established.

```
(inform
  :sender
    (agent-identifier
      :name MAiM@paracom.com)
  :receiver (set
    (agent-identifier
      :name PSA@paracom.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL2
  :protocol FIPA-Subscribe
  :content
    (= (iota ?x
      (qos-information
        (comm-channel
          :name GPRS
          :target-addr wap://paramedic.com:1234/acc)
      (qos
        :status ?x)))
     connected))
```

8. Message 8 and 8' `cancel`: The PSA (/DSA) cancels subscription notifications about the changes in the ATM MTC.

```
(cancel
  :sender
    (agent-identifier
      :name PSA@paracom.com)
  :receiver (set
    (agent-identifier
      :name MAiM@paracom.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL0
  :protocol FIPA-Subscribe
  :content
    (subscribe
      :sender
        (agent-identifier
          :name PSA@paracom.com)
      :receiver (set
        (agent-identifier
          :name MAiM@paracom.com))
      :ontology FIPA-Nomadic-Application
      :language FIPA-SL2
      :protocol FIPA-Subscribe
      :content
        (iota ?x
          (qos-information
            (comm-channel
              :name GPRS
              :target-addr wap://paramedic.com:1234/acc)
          (qos
            :status ?x)))))
```

9. Message 9 and 9' `subscribe`: The DSA (/PSA) subscribes to `MAiG` (/`MAiM`) for notifications about the changes in the GPRS MTC.

```
(subscribe
  :sender
    (agent-identifier
      :name DSA@dispatch.com)
  :receiver (set
    (agent-identifier
      :name MAiG@gateway.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL2
  :protocol FIPA-Subscribe
  :content
    (iota ?x
      (qos-information
        (comm-channel
          :name GPRS
          :target-addr iiop://paramedic.com/acc)
      (qos
        :status ?x))))
```

10. Message 10 `query-ref`: The DSA requests current quality of service of the GPRS MTC from `MAiG`.

```
(query-ref
  :sender
    (agent-identifier
      :name DSA@dispatch.com)
  :receiver (set
    (agent-identifier
      :name MAiG@gateway.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL2
  :protocol FIPA-Query
  :content
    (iota ?x
      (qos-information
        (comm-channel
          :name GPRS)
      (qos
        :throughput ?x)))
```

11. Message 11 `inform`: `MAiG` informs the DSA the current quality of service of the GPRS MTC.

```
(inform
  :sender
    (agent-identifier
      :name MAiG@gateway.com)
  :receiver (set
    (agent-identifier
      :name DSA@dispatch.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL2
  :protocol FIPA-Query
  :content
    (= (iota ?x
      (qos-information
        (comm-channel
          :name GPRS)
      (qos
        :throughput ?x)))
    (rate-value
      :direction Outbound
```

```
      :unit Kbits/s
      :value 20)))
```

12. Messages 12, 13 and 14 `inform`: The DSA sends the rest of the requested information to the PSA.

4.2.2 Example Negotiation of a Message Transport Protocol

When the mobile host roams from the ATM network to the GPRS network – after the reconnection – the PSA receives the information from `MAiM` that the *Paracom* AP is now connected to the GPRS MTC. The PSA reasons that the `fipa.mts.mtp.wap.std` MTP is better in that environment and it requests the CAiM to establish this MTP between `ACCiM` and `ACCiG`. Also, `CAiM` proposes the establishment of this MTP to `CAiG` which accepts and they command their respective ACCs to set it up. As a last action, both `CAiF` and `CAiG` modify the AP descriptions of their APs. The message flow is illustrated in *Figure 10*.

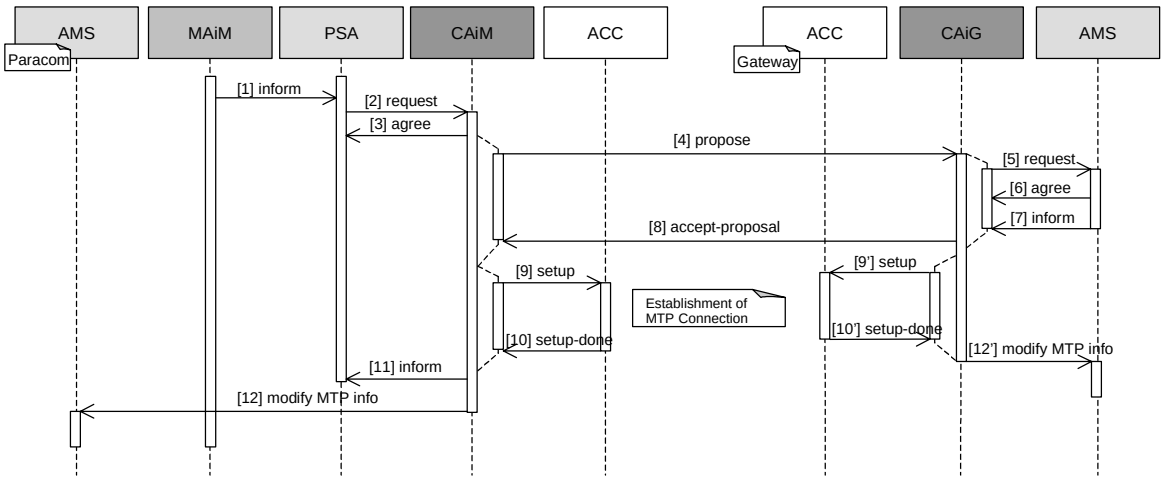

Figure 10: Example Negotiation of a Message Transport Protocol

1. Message 1 `inform`: `MAiM` informs the PSA that the *Paracom* AP is now connected to the GPRS network.

```
(inform
  :sender
    (agent-identifier
      :name MAiM@paracom.com)
  :receiver (set
    (agent-identifier
      :name PSA@paracom.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL2
  :protocol FIPA-Subscribe
  :content
    (= (iota ?x
      (qos-information
        (comm-channel
          :name GPRS
          :target-addr wap://paramedic.com:1234/acc)
      (qos
        :status ?x)))
    connected))
```

2. Message 2 `request` and message 3 `agree`: The PSA requests `CAiM` to establish the `fipa.mts.mtp.wap.std` MTP between `ACCiM` and `ACCiG`.

```
(request
  :sender
    (agent-identifier
      :name PSA@paracom.com)
  :receiver (set
    (agent-identifier
      :name CAiM@paracom.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL0
  :protocol FIPA-Request
  :content
    (action
      (agent-identifier
        :name CAiM@paracom.com)
      (activate (sequence
        (transport-protocol
          :name fipa.mts.mtp.wap.std
          :gw-addr wap://gateway.com:1234/acc))))
```

3. Message 4 `propose`: `CAiM` sends a `propose` message to the `CAiG`.

```
To: (agent-identifier :name CAiG@gateway.com)
From: (agent-identifier :name CAiM@paracom.com)
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T101200000

(propose
  :sender
    (agent-identifier
      :name CAiM@paracom.com)
  :receiver (set
    (agent-identifier
      :name CAiG@gateway.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL0
  :protocol FIPA-Propose
  :content
    ((action
      (agent-identifier
        :name CAiM@paracom.com)
      (use
        (transports
          :send (sequence
            (transport-protocol
              :name fipa.mts.mtp.wap.std))
          :recv (sequence
            (transport-protocol
              :name fipa.mts.mtp.wap.std)))))
    true))
```

4. Message 5 `request`, message 6 `agree` and message 7 `inform`: `CAiG` requests the local AP description to find out if the `fipa.mts.mtp.wap.std` MTP is supported (see [FIPA00023]).

5. Message (8) `accept-proposal`: `CAiG` accepts `CAiM`'s proposal to use the `fipa.mts.mtp.wap.std` MTP.

```
To: (agent-identifier :name CAiM@paracom.com)
From: (agent-identifier :name CAiG@gateway.com)
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T101200000

(accept-proposal
  :sender
    (agent-identifier
      :name CAiG@gateway.com)
  :receiver (set
    (agent-identifier
      :name CAiM@paracom.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL0
  :protocol FIPA-Propose
  :content
    (action
      (agent-identifier
        :name CAiM@paracom.com)
      (use
        (transports
          :send (sequence
            (transport-protocol
              :name fipa.mts.mtp.wap.std))
          :recv (sequence
            (transport-protocol
              :name fipa.mts.mtp.wap.std)))))
    (transports
      :send (sequence
        (transport-protocol
          :name fipa.mts.mtp.wap.std))
      :recv (sequence
        (transport-protocol
          :name fipa.mts.mtp.wap.std))))
```

6. Messages 9 and 9' `setup` and messages 10 and 10' `setup-done`: `CAiM` (`CAiG`) commands `ACCiM` (`ACCiG`) to setup the `fipa.mts.mtp.wap.std` MTP. As this is intra-platform communication between `CAiM` (`CAiG`) and `ACCiM` (`ACCiG`), this is an implementation issue.

7. Message 11 `inform`: `CAiM` returns the result to the PSA.

```
(inform
  :sender
    (agent-identifier
      :name CAiM@paracom.com)
  :receiver (set
    (agent-identifier
      :name PSA@paracom.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL0
  :protocol FIPA-Request
  :content
    (result
      (action
        (agent-identifier
          :name CAiM@paracom.com)
      (activate (sequence
        (transport-protocol
          :name fipa.mts.mtp.wap.std
          :gw-addr wap://gateway.com:1234/acc)))
      (transport-protocol
```

```
        :name fipa.mts.mtp.wap.std
        :gw-addr wap://gateway.com:1234/acc)))
```

8. Message 12 and 12' `set-description`: `CAiM` (`CAiG`) modifies the AP description to show that the `fipa.mts.mtp.wap.std` is now active.

4.2.3 Example Negotiation of a Message Representation

`MAiM` informs the PSA that the quality of the message transport connection has dropped significantly. The PSA reasons that the ACL representation needs to be changed to `fipa.acl.rep.bitefficient.std` and it proposes that to the DSA. The DSA accepts the PSA's proposal. The message flow is illustrated in *Figure 11*.

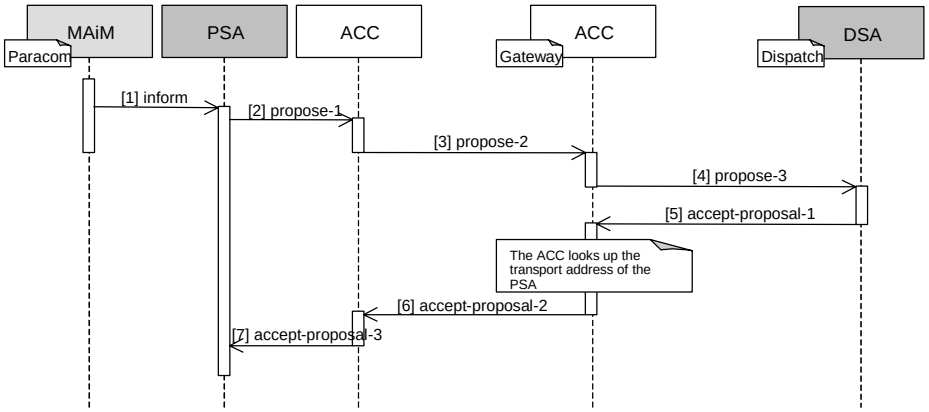

Figure 11: Example Negotiation of a Message Representation

1. Message 1 `inform`: The MA informs the PSA that the outbound throughput has changed.

```
(inform
  :sender
    (agent-identifier
      :name MAiM@paracom.com)
  :receiver (set
    (agent-identifier
      :name PSA@paracom.com))
  :ontology FIPA-Nomadic-Application
  :language FIPA-SL2
  :protocol FIPA-Subscribe
  :content
    (= (iota ?x (
      (qos-notification
        (comm-channel
          :name GPRS)
        (throughput
          (rate-value
            :unit Kbits/s
            :direction Outbound
            :value ?x))
        (change-constraint
          :value (<
            (qos
              :throughput
                (rate-value
                  :unit Kbits/s
                  :direction Outbound
                  :value 1)))))))
    (0.96)))
```

2. Message 2 `propose-1`: Based on the new throughput value, the PSA decides to change to the message representation.

```
To: (agent-identifier :name DSA@dispatch.com)
From: (agent-identifier :name PSA@paracom.com)
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T101200000

(propose
  :sender
    (agent-identifier
      :name PSA@paracom.com)
  :receiver (set
    (agent-identifier
      :name DSA@dispatch.com))
  :ontology FIPA-Message-Representation
  :language FIPA-SL0
  :protocol FIPA-Propose
  :content
    ((action
      (agent-identifier
        :name PSA@paracom.com)
      (use
        (msg-rep-selection
          :send (sequence
            (msg-representation
              :name fipa.acl.rep.bitefficient.std))
          :recv (sequence
            (msg-representation
              :name fipa.acl.rep.bitefficient.std)))))
     true))
```

3. Message 3 `propose-2`: The ACC at the mobile host forwards the same message to the ACC at the gateway host.

4. Message 4 `propose-3`: The ACC at the gateway host forwards the same message to the PSA.

5. Message 5 `accept-proposal-1`: The PSA accepts the proposal and sends a message back to the DSA.

```
To: (agent-identifier :name PSA@paracom.com)
From: (agent-identifier :name DSA@dispatch.com)
ACL-representation: fipa.acl.rep.string.std
Date: 20000606T101200000

(accept-proposal
  :sender
    (agent-identifier
      :name DSA@dispatch.com)
  :receiver (set
    (agent-identifier
      :name PSA@paracom.com))
  :ontology FIPA-Message-Representation
  :language FIPA-SL0
  :protocol FIPA-Propose
  :content
    (action
      (agent-identifier
        :name PSA@paracom.com)
      (use
        (msg-rep-selection
          :send (sequence
            (msg-representation
              :name fipa.acl.rep.bitefficient.std))
```

```
            :recv (sequence
              (msg-representation
                :name fipa.acl.rep.bitefficient.std))))
        (msg-rep-selection
          :send (sequence
            (msg-representation
              :name fipa.acl.rep.bitefficient.std))
          :recv (sequence
            (msg-representation
              :name fipa.acl.rep.bitefficient.std)))))
```

6. Message 6 `accept-proposal–2`: The ACC at the gateway host forwards same message to the ACC at the mobile host.

7. Message 7 `accept-proposal–3`: The ACC at the mobile host delivers the same message to the PSA.

5 References

[FIPA00023] FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00023/

[FIPA00036] FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00036/

[FIPA00062] FIPA Nomadic Application Support Monitor Agent Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00062/

[FIPA00063] FIPA Nomadic Application Support Control Agent Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00063/

[FIPA00065] FIPA Nomadic Application Support Ontology Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00065/

[FIPA00069] FIPA ACL Message Representation in Bit-Efficient Encoding Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00069/

[FIPA00075] FIPA Agent Message Transport Protocol for IIOP Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00075/

[FIPA00076] FIPA Agent Message Transport Protocol for WAP Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00076/

[WAP99] Wireless Application Protocol Specification Version 1.2. WAP Forum, 1999. http://www.wapforum.org/what/technical.htm