

FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

FIPA Nomadic Application Support Specification

Document title	FIPA Nomadic Application Support Specification		
Document number	XI00014G	Document source	FIPA TC Nomadic Application Support
Document status	Experimental	Date of this status	2002/11/01
Supersedes	FIPA00062, FIPA00063, FIPA00065, FIPA00066		
Contact	fab@fipa.org		
Change history	See <i>Informative Annex A — ChangeLog</i>		

© 1996-2002 Foundation for Intelligent Physical Agents
<http://www.fipa.org/>
Geneva, Switzerland

Notice

Use of the technologies described in this specification may infringe patents, copyrights or other intellectual property rights of FIPA Members and non-members. Nothing in this specification should be construed as granting permission to use any of the technologies described. Anyone planning to make use of technology covered by the intellectual property rights of others should first obtain permission from the holder(s) of the rights. FIPA strongly encourages anyone implementing any part of this specification to determine first whether part(s) sought to be implemented are covered by the intellectual property of others, and, if so, to obtain appropriate licenses or other permission from the holder(s) of such intellectual property prior to implementation. This specification is subject to change without notice. Neither FIPA nor any of its Members accept any responsibility whatsoever for damages or liability, direct or consequential, which may result from the use of this specification.

20 **Foreword**

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

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

31 The FIPA specifications are developed through direct involvement of the FIPA membership. The status of a
32 specification can be either Preliminary, Experimental, Standard, Deprecated or Obsolete. More detail about the process
33 of specification may be found in the FIPA Document Policy [f-out-00000] and the FIPA Specifications Policy [f-out-
34 00003]. A complete overview of the FIPA specifications and their current status may be found on the FIPA Web site.

35 FIPA is a non-profit association registered in Geneva, Switzerland. As of June 2002, the 56 members of FIPA
36 represented many countries worldwide. Further information about FIPA as an organization, membership information,
37 FIPA specifications and upcoming meetings may be found on the FIPA Web site at <http://www.fipa.org/>.

38 Contents

39	1	Scope.....	1
40	2	General Analysis.....	2
41	2.1	Overview	2
42	2.2	Monitoring and Controlling Quality of Service	2
43	2.3	Negotiation of Message Transport Requirements.....	4
44	2.3.1	Negotiation about Message Transport Protocols	4
45	2.3.2	Negotiation about Message Representation	4
46	3	Nomadic Application Support Ontology.....	5
47	3.1	Object Descriptions	5
48	3.1.1	Transport Protocol Selection.....	5
49	3.1.2	Message Representation Description	5
50	3.1.3	Message Representation Selection	6
51	3.2	Function and Predicate Descriptions.....	7
52	3.2.1	Transport Selection	7
53	3.2.2	Message Encoding Selection.....	7
54	3.2.3	Open Communication Channel	8
55	3.2.4	Close Communication Channel.....	8
56	3.2.6	Activate a Message Transport Protocol	8
57	3.2.7	Deactivate a Message Transport Protocol.....	8
58	3.2.8	Select a Message Transport Protocol.....	9
59	3.3	Exceptions.....	9
60	3.3.1	Not Understood Exception Predicates.....	9
61	3.3.2	Refusal Exception Predicates.....	9
62	3.3.3	Failure Exception Propositions.....	10
63	4	Registration with the Directory Facilitator	12
64	5	Examples	13
65	5.1	Registration with a Directory Facilitator	13
66	5.2	Negotiating Message Transport Protocols	14
67	5.3	Negotiating Message Representations	18
68	6	Paramedic Scenario	20
69	6.1	Overview	20
70	6.2	Seamless Roaming	22
71	6.2.1	Disconnection and Reconnection of an Message Transport Connection	22
72	6.2.2	Example Negotiation of a Message Transport Protocol.....	26
73	6.2.3	Example Negotiation of a Message Representation	28
74	7	References	31
75	8	Informative Annex A — ChangeLog.....	32
76	8.1	2001/10/17 - version E by TC Gateways.....	32
77	8.2	2002/09/13 - version F by TC X2S	32
78	8.3	2002/11/01 - version G by TC X2S	33

79 **1 Scope**

80 This document is part of the FIPA specifications and deals with agent middleware to support applications in nomadic
81 environment. The environment of mobile computing is very different compared to today's environment of traditional
82 distributed systems in many respects. Bandwidth, latency, delay, error rate, interference, interoperability, computing
83 power, quality of display, among other things may change dramatically as a nomadic end-user moves from one location
84 to another. All these cause new demands for adaptability of data services.

85
86 Adaptability to the changes in the environment of nomadic end-users is an important issue. A nomadic end-user
87 confronted with these circumstances would benefit from having the following functionality provided by the infrastructure:
88 information about expected performance, agents controlling over the transfer operations, a condition-based control
89 policy, capability provided by agents to work in a disconnected mode, advanced error recovery methods, and
90 adaptability.

91
92 This specification gives an overview of the nomadic application support area and contains informative specifications for:

- 93
- 94 • Monitor Agent (MA) functionality, and
- 95
- 96 • Control Agent (CA) functionality.
- 97

98 In addition, three other FIPA specifications are related to nomadic application support: [FIPA00069], [FIPA00088] and
99 [FIPA00094].

100

101 2 General Analysis

102 2.1 Overview

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

114
115 Confronted with these circumstances, the nomadic end-user would benefit from having the following functionality
116 provided by the infrastructure: information about expected performance, agent monitoring and controlling the transfer
117 operations, and adaptability.

118
119 The ability to automatically adjust to changes in a transparent and integrated fashion is essential for *nomadicity*;
120 nomadic end-users are usually professionals in areas other than computing. Furthermore, today's mobile computer
121 systems are already very complex to use as productivity tools. Thus, nomadic end-users need all the support that a
122 FIPA agent-based distributed system can deliver and adaptability to the changes in the environment of nomadic end-
123 users is an important issue.

124
125 The adaptation of applications to various nomadic computing environments is an important area. There are several
126 tasks that agents need to carry out during application adaptation:

- 127
128 1. Selection of Message Transport Protocol (MTP) and Message Transport Connection (MTC) to be used for agent
129 communication.
- 130
131 2. Selection of an ACL and content language representation to be used for agent communication.
- 132
133 3. Provision of support for application agents to carry out adaptation of application data, such as still images, video
134 and audio, XML, etc. Today's Internet application data (such as multimedia content) are designed with high
135 performance desktop PCs and high quality displays in mind. Therefore, the application data is frequently unsuitable
136 for nomadic computing using wireless wide-area networks and low performance mobile devices, and hence
137 requires modification.
- 138
139 4. Communication between agents performing adaptation.

140
141 The FIPA Nomadic Application Support specifications define agent middleware to monitor and control an MTP and the
142 underlying MTC. In addition, this specification gives examples of the use of the above scenarios.

144 2.2 Monitoring and Controlling Quality of Service

145 The functions required to carry out monitoring and controlling for Quality of Service (QoS) can be split into several
146 specific tasks:

- 147
148 1. Observing the QoS of MTPs and MTCs,
- 149
150 2. Measuring (if there are no other means to obtain the required information) the QoS of an MTP and MTC,
- 151
152 3. Collecting information from the observing and measuring sources,

153
154
155
156
157
158
159
160
161
162
163

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 MA which carries out tasks 1 through 4, and,
- A CA which carries out task 5.

164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188

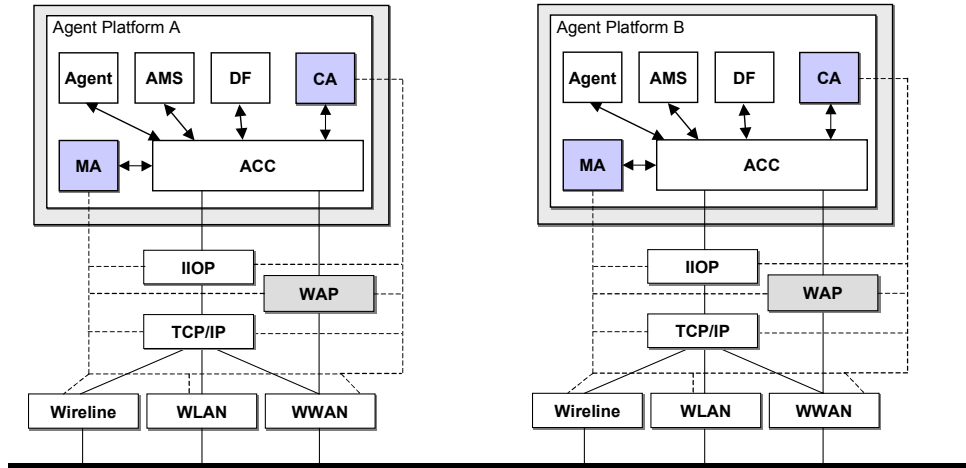


Figure 1: Reference Model of Agent based Adaptation

The most appropriate configuration of MAs and CAs is that there is at least one pair in each AP involving adaptation. The MA may measure the actual QoS of an MTC, if the network running an MTC does not provide users with required performance data¹.

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

A CA may:

- Manage (establish, close, suspend, activate, etc.) an MTC².

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.

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

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

189 2.3 Negotiation of Message Transport Requirements

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

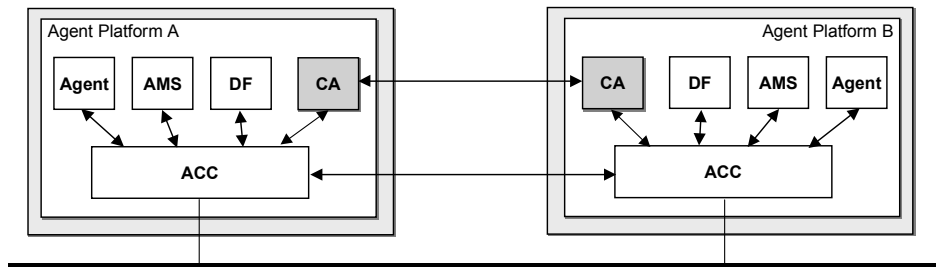
- 192
- 193 • Communicating entities know a peer entity's preferences beforehand and use them.
- 194
- 195 • The activating entity tries to use a method and if the peer entity is not capable of using the suggested method, then
 196 the activating entity may try another one (and so on).
- 197
- 198 • The communicating entities negotiate about a method to be used.
- 199

200 2.3.1 Negotiation about Message Transport Protocols

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

204
 205 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
 206 CA to either accept or reject the proposal based on whether it is possible to use the proposed MTP. CAs negotiate with
 207 peer CAs to use proposed MTPs which is illustrated in *Figure 2*.

208



209

210

211

212

213

214

215

Figure 2: Control Agents Negotiating About a Message Transport Protocol

213 CAs use the `fipa-propose` interaction protocol [FIPA00036] and the `use` action to negotiate about an MTP. An
 214 example negotiation is given in Section 5.2.

216 2.3.2 Negotiation about Message Representation

217 In the environment of nomadic applications, it may be necessary to switch from one ACL representation to another; for
 218 example, when a mobile host roams from a wire line network to a wireless network. Application agents may use the
 219 `fipa-propose` interaction protocol and the `use` action to negotiate about the representation of ACL. Examples of this
 220 negotiation are given in Section 5.3.

221

222 3 Nomadic Application Support Ontology

223 3.1 Object Descriptions

224 This section describes a set of frames, that represent the classes of objects in the domain of discourse within the
 225 framework of the `fipa-nas` ontology. The `fipa-nas` ontology extends the `fipa-qos` ontology defined in
 226 [FIPA00094].

227
 228 The following terms are used to describe the objects of the domain:

- 230 • **Frame.** This is the mandatory name of this entity that must be used to represent each instance of this class.
- 231
- 232 • **Ontology.** This is the name of the ontology, whose domain of discourse includes the parameters described in the
 233 table.
- 234
- 235 • **Parameter.** This is the mandatory name of a parameter of this frame.
- 236
- 237 • **Description.** This is a natural language description of the semantics of each parameter.
- 238
- 239 • **Presence.** This indicates whether each parameter is mandatory or optional.
- 240
- 241 • **Type.** This is the type of the values of the parameter: Integer, Word, String, URL, Term, Set or Sequence.
- 242
- 243 • **Reserved Values.** This is a list of FIPA-defined constants that can assume values for this parameter.
- 244

245 3.1.1 Transport Protocol Selection

246 This type of object represents a selection of transport protocol.

247

Frame Ontology	transports fipa-nas	Parameter	Description	Presence	Type	Reserved Values
send	A list of transport protocols supported for sending messages.	Mandatory	Sequence of transport-protocol ³			
recv	A list of transport protocols supported for receiving messages.	Mandatory	Sequence of transport-protocol			

248

249 3.1.2 Message Representation Description

250 This type of object represents an ACL message representation.

251

Frame Ontology	msg-representation fipa-nas	Parameter	Description	Presence	Type	Reserved Values
name	The name of the message representation.	Mandatory	word			
options	A list of parameters for the message representation.	Optional	Set of property ⁴			

³ See [FIPA00094].

⁴ See [FIPA00023].

252

253 **3.1.3 Message Representation Selection**

254 This type of object represents a selection of message representations.

255

Frame	msg-encoding			
Ontology	fipa-nas			
Parameter	Description	Presence	Type	Reserved Values
send	A list of message representations supported for sending messages.	Mandatory	Sequence of msg-representation	
recv	A list of message representations supported for receiving messages.	Mandatory	Sequence of msg-representation	

256

257 3.2 Function and Predicate Descriptions

258 The following tables define usage and semantics of the functions and the predicates that are part of the `fipa-nas`
 259 ontology.

260
 261 The following terms are used to describe the functions of the `fipa-nas` domain:
 262

- 263 • **Function.** This is the symbol that identifies the function in the ontology.
- 264
- 265 • **Predicate.** This is the symbol that identifies the predicate in the ontology.
- 266
- 267 • **Ontology.** This is the name of the ontology, whose domain of discourse includes the function or the predicate
 268 described in the table.
- 269
- 270 • **Supported by.** This is the type of agent that supports this function or predicate.
- 271
- 272 • **Description.** This is a natural language description of the semantics of the function or the predicate.
 273
- 274 • **Domain.** This indicates the domain over which the function predicate is defined. The arguments passed to the
 275 function or predicate must belong to the set identified by the domain.
- 276
- 277 • **Range.** This indicates the range to which the function maps the symbols of the domain. The result of the function is
 278 a symbol belonging to the set identified by the range.
- 279
- 280 • **Arity.** This indicates the number of arguments that a function or a predicate takes. If a function or a predicate can
 281 take an arbitrary number of arguments, then its arity is undefined.
 282

283 3.2.1 Transport Selection

Predicate	<code>transport-selection</code>
Ontology	<code>fipa-nas</code>
Supported by	CA
Description	An agent specifies the transport protocols that it is willing to use. The predicate is true, when the values of the <code>transports</code> parameter contain the transport protocol descriptions that the agent is willing to use. Otherwise, the predicate is false
Domain	<code>transports</code>
Arity	1

284

285 3.2.2 Message Encoding Selection

Predicate	<code>msg-encoding-selection</code>
Ontology	<code>fipa-nas</code>
Supported by	CA
Description	An agent specifies the message encoding choices that it is willing to use. The predicate is true, when the values of the <code>msg-encoding</code> parameter contain the message encoding choices that the agent is willing to use. Otherwise, the predicate is false
Domain	<code>msg-encoding</code>
Arity	1

286

287

291 **3.2.3 Open Communication Channel**

Function	open-comm-channel
Ontology	fipa-nas
Supported by	CA
Description	An agent can request that a CA open a communication channel. The communication channel description should contain enough information for a CA to be able to choose the right communication channel, that is, either the <code>name</code> parameter or the <code>target-addr</code> parameter must be present. The agent may also supply additional communication channel information by using the <code>options</code> parameter.
Domain	comm-channel
Range	The execution of this function results in a change of the state, but it has no explicit result. Therefore there is no range set.
Arity	1

292

293 **3.2.4 Close Communication Channel**

Function	close-comm-channel
Ontology	fipa-nas
Supported by	CA
Description	An agent can request that a CA close a communication channel. The communication channel description should contain enough information for a CA to be able to choose the right communication channel, that is, either the <code>name</code> parameter or the <code>target-addr</code> parameter must be present.
Domain	comm-channel
Range	The execution of this function results in a change of the state, but it has no explicit result. Therefore there is no range set.
Arity	1

294

295 **3.2.6 Activate a Message Transport Protocol**

Function	activate
Ontology	fipa-nas
Supported by	CA
Description	An agent can request that a CA activate a Message Transport Protocol (MTP). The transport protocol description should contain enough information to allow the CA to identify the correct transport protocol. Additionally, the agent may supply address information to where the transport protocol connection should be opened. It is possible to give the address of the gateway and/or the address of the destination AP. If the action is successful, the CA will return the object description of activated MTP.
Domain	Sequence of <code>transport-protocol</code>
Range	<code>transport-protocol</code>
Arity	1

296

297 **3.2.7 Deactivate a Message Transport Protocol**

Function	deactivate
Ontology	fipa-nas
Supported by	CA
Description	An agent can request that a CA deactivate an MTP.
Domain	<code>transport-protocol</code>

Range	The execution of this function results in a change of the state, but it has no explicit result. Therefore there is no range set.
Arity	1

298

299

3.2.8 Select a Message Transport Protocol

Function	use
Ontology	fipa-nas
Supported by	CA
Description	An CA can request another CA to select an MTP or message encoding for use between Agent Communication Channels (ACCs). The requesting CA shall provide enough information to establish a working MTP connection or message encoding. The direction of communication (either send, receive or both) and the list of choices must be present. The list of choices is an ordered list where the highest priority is the first item and the lowest priority is the last item in the list. The receiving CA shall select at most one choice for the proposed direction of communication (either send, receive or both)
Domain	transports / ⁷ msg-encoding
Range	transport-selection / ⁸ msg-encoding-selection
Arity	1

300

3.3 Exceptions

301

302

The exceptions for the fipa-nas ontology follow the same form and rules as specified in [FIPA00023].

303

304

3.3.1 Not Understood Exception Predicates

Communicative Act Ontology	not-understood fipa-nas	
Predicate Symbol	Arguments	Description
unsupported-act	string	The receiving agent does not support the specific communicative act; the string identifies the unsupported communicative act.
unexpected-act	string	The receiving agent supports the specified communicative act, but it is out of context; the string identifies the unexpected communicative act.
unsupported-value	string	The receiving agent does not support the value of a message parameter; the string identifies the message parameter name.
unrecognised-value	string	The receiving agent cannot recognise the value of a message parameter; the string identifies the message parameter name.

305

306

3.3.2 Refusal Exception Predicates

Communicative Act Ontology	refuse fipa-nas	
Predicate symbol	Arguments	Description
unauthorised		The sending agent is not authorised to perform the function.

⁷ Where '/' is "exclusive or".

⁸ Where '/' is "exclusive or".

unsupported-function	string	The receiving agent does not support the function; the string identifies the unsupported function name.
missing-argument	string	A mandatory function argument is missing; the string identifies the missing function argument name.
unexpected-argument	string	A mandatory function argument is present which is not required; the string identifies the function argument that is not expected.
unexpected-argument-count		The number of function arguments is incorrect.
missing-parameter	string string	A mandatory parameter is missing; the first string represents the object name and the second string represents the missing parameter name.
unexpected-parameter	string string	The receiving agent does not support the parameter; the first string represents the function name and the second string represents the unsupported parameter name.
unrecognised-parameter-value	string string	The receiving agent cannot recognise the value of a parameter; the first string represents the object name and the second string represents the parameter name of the unrecognised parameter value.
already-open	string	The specified communication channel is already open; the string identifies the communication channel.
not-open	string	The specified communication channel is not open; the string identifies the communication channel.
already-activated	string	The specified transport protocol is already activated; the string identifies the transport protocol.
not-active	string	The specified transport protocol is not active; the string identifies the transport protocol.
unrecognised-comm-channel	string	The specified communication channel is not recognised; the string identifies the communication channel.
unsupported-protocol	string	The specified transport protocol is not supported; the string identifies the transport protocol.

307

308

3.3.3 Failure Exception Propositions

Communicative Act Ontology	Arguments	Description
failure fipa-nas		
internal-error	string	An internal error occurred; the string identifies the internal error.
open-failed	string	The opening of a communication channel failed; the string identifies the failure reason.
transient-failed	string	The opening/closing of a communication channel or the activation/deactivation of a transport protocol failed; the string identifies the failure reason.

close-failed	string	The closing of a communication channel failed; the string identifies the failure reason.
activation-failed	string	The activation of a transport protocol failed; the string identifies the failure reason.
deactivation-failed	string	The deactivation of a transport protocol failed; the string identifies the failure reason.

310 4 Registration with the Directory Facilitator

311 In order for a CA and MA to advertise its willingness to provide its services to an agent domain, it must register with a
 312 DF (as described in [FIPA00023]. As part of this registration process, the following of constant values are introduced
 313 that universally identify the services the agent provides:

- 314
- 315 • The name parameter in service-description frame of a CA must be declared as a constant `fipa-mts-`
 316 `control`.
- 317
- 318 • The type parameter in service-description frame of a CA must be declared as a constant `fipa-ca`.
- 319
- 320 • The ontology parameter in service-description frame of a CA should be declared as a constant `fipa-`
 321 `nas`.
- 322
- 323 • The type parameter in service-description frame of a MA must be declared as a constant `fipa-mts-`
 324 `monitor`.
- 325
- 326 • The type parameter in service-description frame of a MA must be declared as a constant `fipa-ma`.
- 327
- 328 • The ontology parameter in service-description frame of a MA should be declared as a constant `fipa-`
 329 `qos`.
- 330

331 Below is given an example content of a `df-agent-description` frame which provides both MA and CA functionality:

```

332 (df-agent-description
333   :name
334     (agent-identifier
335       :name monitor&control_agent@iiop://foo.com/acc
336       :addresses (sequence iiop://foo.com/acc))
337   :protocols (set fipa-request fipa-propose)
338   :ontology (set fipa-nas)
339   :language (set fipa-sl)
340   :services (set
341     (service-description
342       :name fipa-mts-control
343       :type fipa-ca
344       :ontology fipa-nas)
345     (service-description
346       :name fipa-mts-monitor
347       :type fipa-ma
348       :ontology fipa-qos))
349   :ownership (set Sonera))))
350
351
```

352 5 Examples

353 5.1 Registration with a Directory Facilitator

354 1. A CA registers with a DF (see [FIPA00023]):

```

355 (request
356   :sender
357     (agent-identifier
358       :name ca@foo.com
359       :addresses (sequence http://foo.com/acc))
360   :receiver (set
361     (agent-identifier
362       :name df@foo.com
363       :addresses (sequence http://foo.com/acc)))
364   :language fipa-sl
365   :protocol fipa-request
366   :ontology fipa-agent-management
367   :content "(
368     (action
369       (agent-identifier
370         :name df@foo.com
371         :addresses (sequence http://foo.com/acc))
372       (register
373         (df-agent-description
374           :name
375             (agent-identifier
376               :name ca@foo.com
377               :addresses (sequence http://foo.com/acc))
378           :services (set
379             (service-description
380               :name fipa-mts-control
381               :type fipa-ca
382               :ontology (set fipa-nas))))))))))")
383

```

384 2. An MA registers with a DF.

```

385 (request
386   :sender
387     (agent-identifier
388       :name ma@foo.com
389       :addresses (sequence http://foo.com/acc))
390   :receiver (set
391     (agent-identifier
392       :name df@foo.com
393       :addresses (sequence http://foo.com/acc)))
394   :language fipa-sl
395   :protocol fipa-request
396   :ontology fipa-agent-management
397   :content "(
398     (action
399       (agent-identifier
400         :name df@foo.com
401         :addresses (sequence http://foo.com/acc))
402       (register
403         (df-agent-description
404           :name
405             (agent-identifier
406               :name ma@foo.com
407               :addresses (sequence http://foo.com/acc))
408           :services (set
409             (service-description
410               :name fipa-mts-control
411               :type fipa-ca
412               :ontology (set fipa-nas))))))))))")
413

```



```

411         (service-description
412           :name fipa-mts-monitor
413           :type fipa-ma
414           :ontology (set fipa-nas)))))))))")
415

```

5.2 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*.

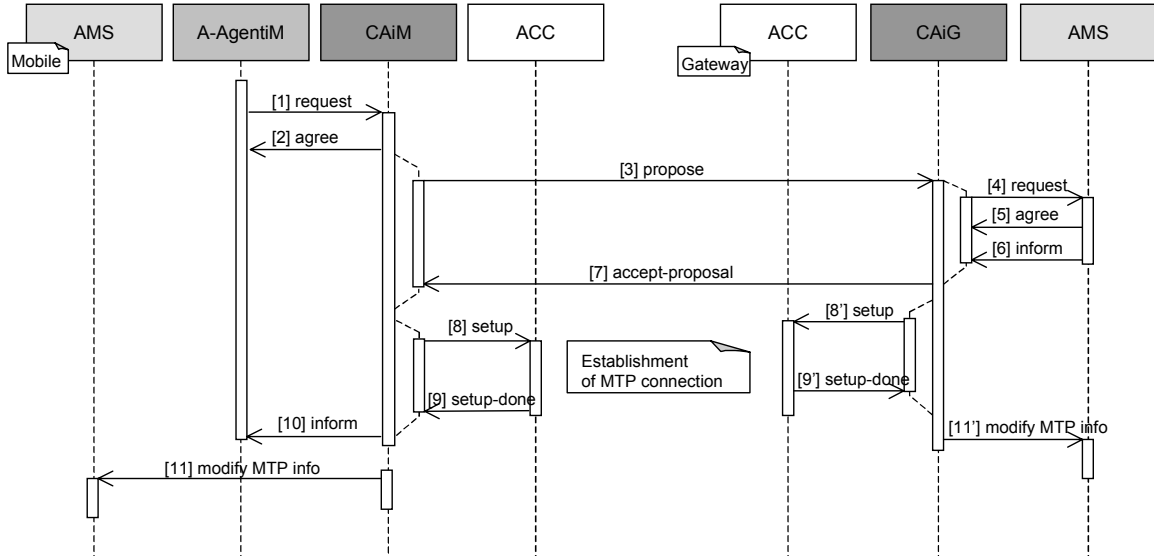


Figure 3: Flow of Message Transport Protocol Negotiation

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.

```

420
421
422
423
424 1. Message 1 request: An application agent issues a request to the CA to activate either the
425 fipa.mts.mtp.wap.std or x.uh.mdcP MTPs.
426
427 (request
428   :sender
429     (agent-identifier
430       :name A-AgentiM@mobile.com9)
431   :receiver (set
432     (agent-identifier
433       :name CaiM@mobile.com))
434   :ontology fipa-nas
435   :language fipa-sl
436   :protocol fipa-request
437   :content "(
438     (action
439       (agent-identifier
440         :name CAiM@mobile.com)
441       (activate (sequence
442         (transport-protocol
443           :name x.uh.mdcP)
444         (transport-protocol
445           :name fipa.mts.mtp.wap.std
446           :dest-addr wap://gateway.com:1234/acc))))))")
447

```

⁹ 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.

- 448 2. Message 2 agree: The CA agrees to activate an MTP. The decision to agree or disagree to activate an MTP might
 449 be based on the internal state of the CA (that is, the CA knows whether a requested MTP can be activated or not)
 450 or the CA might ask for an AP description from an AMS.

```

451 (agree
452   :sender
453     (agent-identifier
454       :name CAiM@mobile.com)
455   :receiver (set
456     (agent-identifier
457       :name A-AgentiM@mobile.com))
458   :ontology fipa-nas
459   :language fipa-sl
460   :protocol fipa-request
461   :content "(
462     (action
463       (agent-identifier
464         :name CAiM@mobile.com)
465       (activate (sequence
466         (transport-protocol
467           :name x.uh.mdcp)
468         (transport-protocol
469           :name fipa.mts.mtp.wap.std
470           :dest-addr wap://gateway.com:1234/acc))))
471     true))"
472
473

```

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

```

476
477 <?xml version="1.0"?>10
478
479 <envelope>
480
481   <params index="1">
482
483     <to>
484       <agent-identifier>
485         <name>CAiG@gateway.com</name>
486       </agent-identifier>
487     </to>
488     <from>
489       <agent-identifier>
490         <name>CAiM@mobile.com</name>
491       </agent-identifier>
492     </from>
493
494     <acl-representation>fipa.acl.rep.string.std</acl-representation>
495
496     <date>20000606T100900000</date>
497
498   </params>
499 </envelope>
500
501 (propose
502   :sender
503     (agent-identifier
504       :name CAiM@mobile.com)
505   :receiver (set
506     (agent-identifier
507

```

¹⁰ In most of the examples, the envelope part has been omitted for clarity.

```

508         :name CAiG@gateway.com))
509 :ontology fipa-nas
510 :language fipa-sl
511 :protocol fipa-propose
512 :content "(
513   (action
514     (agent-identifier
515       :name CAiM@mobile.com)
516     (use
517       (transports
518         :send (sequence
519           (transport-protocol
520             :name x.uh.mdcp)
521           (transport-protocol
522             :name fipa.mts.mtp.wap.std))
523         :recv (sequence
524           (transport-protocol
525             :name x.uh.mdcp)
526           (transport-protocol
527             :name fipa.mts.mtp.wap.std))))))
528   true)")
529

```

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 QoS requirements of the MTC.

```

535 (request
536   :sender
537     (agent-identifier
538       :name CAiG@gateway.com)
539   :receiver (set
540     (agent-identifier
541       :name ams@gateway.com))
542   :ontology fipa-agent-management
543   :language fipa-sl
544   :protocol fipa-request
545   :content "(
546     (action
547       (agent-identifier
548         :name ams@gateway.com)
549       get-description))")
550
551 (agree
552   :sender
553     (agent-identifier
554       :name ams@gateway.com)
555   :receiver (set
556     (agent-identifier
557       :name CAiG@gateway.com))
558   :ontology fipa-agent-management
559   :language fipa-sl
560   :protocol fipa-request
561   :content "(
562     (action
563       (agent-identifier
564         :name ams@gateway.com)
565       get-description)
566     true)")
567
568 (inform
569   :sender
570     (agent-identifier

```

```

571         :name ams@gateway.com
572         :addresses (sequence http://gateway.com/acc))
573 :receiver (set
574   (agent-identifier
575     :name CAiG@gateway.com
576     :addresses (sequence http://gateway.com/acc)))
577 :ontology fipa-agent-management
578 :language fipa-sl
579 :protocol fipa-request
580 :content "(
581   (result
582     (action
583       (agent-identifier :name ams@gateway.com)
584       get-description)
585     (ap-description
586       :name sonera-platform
587       :transport-profile
588       (ap-transport-description
589         :available-mtps
590         (set
591           (mtp-description
592             :profile fipa.profile.mts.alpha
593             :mtp-name fipa.mts.mtp.iiop.std
594             :addresses (sequence iiop://gateway.com/acc))
595           (mtp-description
596             :profile fipa.profile.mts.beta
597             :mtp-name fipa.mts.mtp.wap.std
598             :addresses (sequence wap://gateway.com:1234/acc))
599           (mtp-description
600             :profile x.uh.profile
601             :mtp-name x.uh.mdcp
602             :addresses (set mdcp://gateway.com/acc)))))))))")
603

```

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.

```

607 (accept-proposal
608   :sender
609     (agent-identifier
610       :name CAiG@gateway.com)
611   :receiver (set
612     (agent-identifier
613       :name CAiM@mobile.com))
614 :ontology fipa-nas
615 :language fipa-sl
616 :protocol fipa-propose
617 :content "(
618   (action
619     (agent-identifier
620       :name CAiM@mobile.com)
621     (use
622       (transports
623         :send (sequence
624           (transport-protocol
625             :name x.uh.mdcp)
626           (transport-protocol
627             :name fipa.mts.mtp.wap.std))
628       :recv (sequence
629         (transport-protocol
630           :name x.uh.mdcp)
631         (transport-protocol
632           :name fipa.mts.mtp.wap.std))))))
633

```

```

634     (transport-selection
635       (transports
636         :send (sequence
637           (transport-protocol
638             :name fipa.mts.mtp.wap.std))
639         :recv (sequence
640           (transport-protocol
641             :name fipa.mts.mtp.wap.std))))))")
642

```

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.

```

650 (inform
651   :sender
652     (agent-identifier
653       :name CAiM@mobile.com)
654   :receiver (set
655     (agent-identifier
656       :name A-AgentiM@mobile.com))
657   :ontology fipa-nas
658   :language fipa-sl
659   :protocol fipa-request
660   :content "(
661     (result
662       (action
663         (agent-identifier
664           :name CaiM@mobile.com)
665         (activate (sequence
666           (transport-protocol
667             :name x.uh.mdcp)
668           (transport-protocol
669             :name fipa.mts.mtp.wap.std
670             :dest-addr wap://gateway.com:1234/acc))))
671     (transport-protocol
672       :name fipa.mts.mtp.wap.std)))")
673

```

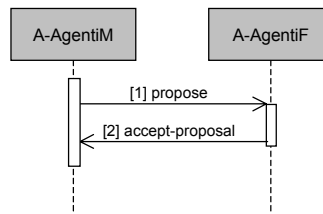
9. Message 11 and 11' set-description: CAiM (/CAiG) modifies the AP description to show that the `fipa.mts.mtp.wap.std` is now active.

677

678 5.3 Negotiating Message Representations

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

682



683

684

685

686

Figure 4: Flow of Message Representation Negotiation

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

```
689 (propose
690   :sender
691     (agent-identifier
692      :name A-AgentiM@mobile.com)
693   :receiver (set
694     (agent-identifier
695      :name A-AgentiF@fixed.com))
696   :ontology fipa-nas
697   :language fipa-sl
698   :protocol fipa-propose
699   :content "(
700     (action
701      (agent-identifier
702       :name A-AgentiM@mobile.com)
703     (use
704      (msg-rep-selection
705       :send (sequence
706         (msg-representation
707          :name fipa.acl.rep.bitefficient.std))
708       :rcv (sequence
709         (msg-representation
710          :name fipa.acl.rep.bitefficient.std))))))
711     true) ")
712
```

713

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

```
715 (accept-proposal
716   :sender
717     (agent-identifier
718      :name A-AgentiF@fixed.com)
719   :receiver (set
720     (agent-identifier
721      :name A-AgentiM@mobile.com))
722   :ontology fipa-nas
723   :language fipa-sl
724   :protocol fipa-propose
725   :content "(
726     (action
727      (agent-identifier
728       :name A-AgentiM@mobile.com)
729     (use
730      (msg-encoding
731       :send (sequence
732         (msg-representation :name fipa.acl.rep.bitefficient.std))
733       :rcv (sequence
734         (msg-representation :name fipa.acl.rep.bitefficient.std))))))
735     (msg-encoding-selection
736      (msg-encoding
737       :send (sequence
738         (msg-representation :name fipa.acl.rep.bitefficient.std))
739       :rcv
740        (sequence
741         (msg-representation :name fipa.acl.rep.bitefficient.std))))))")
742
```

743

744 6 Paramedic Scenario

745 This section illustrates some of the important issues of nomadic application support, using a paramedic application as
746 an example.
747

748 6.1 Overview

749 A paramedic team has several working environments:

- 750 • An emergency dispatch centre, which is covered by the hospital ATM network,
- 751 • A geographical area, which is wireless wide-area network (for example, GPRS), and,
- 752 • One or more hospitals, which are provided with a wireless local-area network.

753
754
755
756
757 When in transit, the paramedic computers are attached to docking stations residing in ambulances. At the dispatch
758 centre, the docking stations are connected to the ATM network. The paramedic application comprises the following
759 services:

- 760 • Retrieval of a patient's personal information, such as name, address, phone, and relatives,
- 761 • Retrieval of the patient's medical histories,
- 762 • Support for paramedic workers, and,
- 763 • Informing the hospital receiving the patient about the patient's current injury or illness and medical care given so far.

764
765
766
767
768
769 There are several application agents: Paramedic Support Agents (PSAs) working in the paramedic computers,
770 Dispatching Support Agent (DSA) working at the dispatch centre system, and the Hospital First Aid Support Agent
771 (HFASA) working at the hospital system.
772

773 The dispatch centre receives a call regarding a man who has severe chest pain; the symptom of an acute myocardial
774 infarct. The caller identifies the man and gives his personal identification number to the dispatcher. The dispatcher
775 alerts the paramedic team and informs the DSA about the address where the patient is located and his personal
776 identification number. The DSA simultaneously informs the PSA about the address of the attack (and possibly some
777 additional information about the environment of the heart attack) and queries the patient's medical history. Since the
778 results of the query to a local hospital are received before the paramedic unit is dispatched, the DSA (in co-operation
779 with the PSA) begins to load the patient's personal information and medical history into the paramedic computers. The
780 medical history includes several items of text-based information. The transmission time to load the information via the
781 ATM network to the paramedic computers (which are currently docked at the dispatch centre) is less than a second.
782 Before the ambulance leaves the dispatch centre, the docking station is detached from the ATM network and is
783 connected to the wireless wide-area network.
784

785 While the ambulance is approaching the location of the incident, the DSA receives more relevant results of the query of
786 the medical histories such as the latest heart operation of the patient. The medical history comprises several parts of
787 textual information and several images and the DSA begins loading the information. As the loading takes place when
788 the ambulance is in motion, the DSA finds out that the quality of transport service is too low for loading some textual
789 parts and any of the images of the medical history. It would take at least 40 minutes to download the images. Therefore,
790 the DSA informs the PSA that images are not required for the paramedic unit. During downloading, the ambulance
791 drives into a tunnel that causes the wireless link to be disconnected. After the tunnel, a CA re-establishes the
792 connection and downloading continues.
793

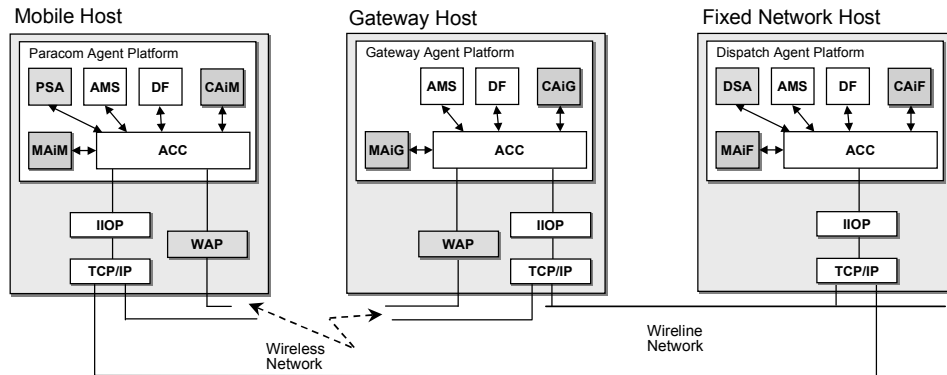
794 At the scene, the ambulance is stationary and the quality of transmission service increases to a level at which the DSA
795 is able to load the most relevant images (the ECGs) using an efficient compression method which is negotiated

796 between the DSA and the PSA to the paramedic computer. The paramedic team detaches the computers from the
 797 docking station and carries them to the patient.
 798

799 The paramedic team realises that they need the assistance of a medical expert located at the university hospital to
 800 stabilise the patient's condition. Therefore, they attach electrodes to the patient and the PSA starts transmitting the data
 801 of measurement such as SpO2 (oxygen saturation), cardiac rhythm, ECG, end tidal CO2 and temperature to the
 802 hospital. After successfully stabilising the patient's condition, the paramedic team moves the patient to the ambulance
 803 and sets off for the hospital. As the quality of the transport service decreases because of the motion, the PSA finds out
 804 that not all the on-going measurement data can be transmitted on-line to the hospital. Therefore, the PSA decides to
 805 transmit the most relevant data (SpO2 and cardiac rhythm). The PSA stores the rest of the data (ECG, end tidal CO2
 806 and temperature) into a cache of the paramedic computer.
 807

808 After the ambulance arrives at the hospital, the patient is transferred immediately to an operating room. Simultaneously,
 809 the paramedic team connects their paramedic computer to the wireless LAN of the hospital and the PSA transmits (in
 810 co-operation with the HFASA) all the measurement data to the hospital's system. A surgeon retrieves and analyses the
 811 measurement data before surgery.
 812

813 This example illustrates a future agent-based distributed system that offers its services at the best obtainable QoS in a
 814 wide variety of environments. A possible agent architecture is illustrated in *Figure 5* which refers to three separate APs:
 815 *Dispatch*, *Gateway* and *Paracom*. In addition, there are several hospital APs which are not illustrated.
 816



817
 818
 819 **Figure 5: Paramedic Scenario Architecture**
 820

821 The agents in the scenario are:

- 822
- 823 • *MAiM*, *MAiG* and *MAiF* are MAs which monitor the quality of the communication service, and,
 - 824
 - 825 • *CAiM*, *CAiG* and *CAiF* are CAs which manage the establishment, teardown, suspension, activation, etc. of the
 826 connection between the PAs. The MA informs application agents about the status and changes of the network
 827 services.
 828

829 When the mobile host is connected either to the ATM network or to the wireless LAN, the *fipa.mts.mtp.iio.p.std*
 830 MTP is used directly between the *Paracom* AP and the *Dispatch* AP. When the mobile host is connected to the wireless
 831 WAN, all agent message communication takes place through the gateway host. The *fipa.mts.mtp.wap.std* MTP is
 832 primarily used between the *Paracom* AP and the *Gateway* AP. The *fipa.mts.mtp.iio.p.std* MTP is used between
 833 the *Gateway* AP and the *Dispatch* AP.
 834

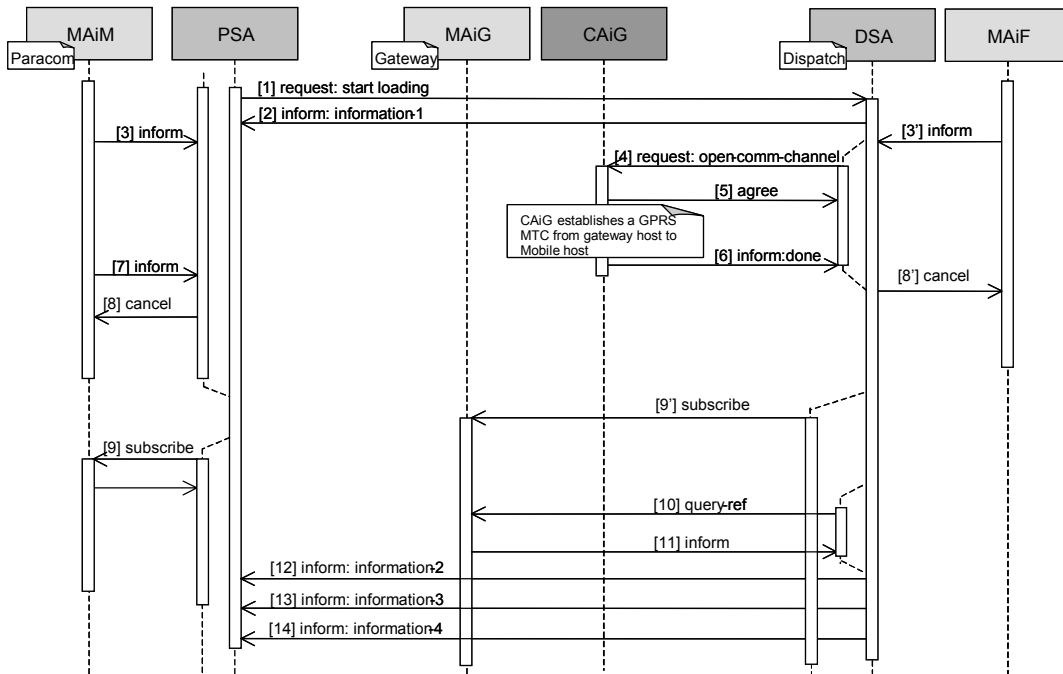
835 **6.2 Seamless Roaming**

836 The Seamless Roaming scenario describes the process, when the paramedic computer roams from the ATM network
 837 to the UMTS network. The scenario is split into following events:
 838

- 839 • Disconnection and reconnection of MTCs,
- 840
- 841 • Negotiation of MTPs, and,
- 842
- 843 • Negotiation of message representations.
- 844

845 **6.2.1 Disconnection and Reconnection of an Message Transport Connection**

846 The message exchange between the agents is illustrated in *Figure 6*.
 847



848 **Figure 6: Disconnection and Reconnection of a Message Transport Connection**
 849

- 850 1. Message 1 *request*: The PSA starts loading data from the DSA by sending a *request* message. This message is
 851 application specific and thus not shown here.
- 852 2. Message 2 *inform*: The DSA starts sending information by first sending an *inform* message.
 853
- 854 3. Messages 3 and 3' *inform*: MAiM (/MAiF) informs the PSA (/DSA) that the ATM connection has broken.
 855

```

856 (inform
857   :sender
858     (agent-identifier
859       :name MAiM@paracom.com)
860   :receiver (set
861     (agent-identifier
862       :name PSA@paracom.com))
863   :ontology fipa-nas
864   :language fipa-sl
865   :protocol fipa-subscribe
866   :conversation-id subscription-3105
867 )
868
869

```

```

870     :content "(
871         (qos-information
872           (comm-channel
873             :name ATM
874             :target-addr iiop://dispatch.com/acc)
875           (qos
876             :status disconnected))))")
877

```

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

```

880 (request
881   :sender
882     (agent-identifier
883       :name DSA@dispatch.com)
884   :receiver (set
885     (agent-identifier
886       :name CAiG@gateway.com))
887   :ontology fipa-nas
888   :language fipa-sl
889   :protocol fipa-request
890   :content "(
891     (action
892       (agent-identifier
893         :name CAiG@gateway.com)
894       (open-comm-channel
895         (comm-channel
896           :name GPRS
897           :target-addr iiop://paramedic.com/acc))))")
898

```

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

```

900 (agree
901   :sender
902     (agent-identifier
903       :name CAiG@gateway.com)
904   :receiver (set
905     (agent-identifier
906       :name DSA@dispatch.com))
907   :ontology fipa-nas
908   :language fipa-sl
909   :protocol fipa-request
910   :content "(
911     (action
912       (agent-identifier
913         :name CAiG@gateway.com)
914       (open-comm-channel
915         (comm-channel
916           :name GPRS
917           :target-addr iiop://paramedic.com/acc))))
918   true)")
919

```

Next CAiG establishes a GPRS MTC from the gateway host to the mobile host (his is an implementation issue).

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

```

924 (inform
925   :sender
926     (agent-identifier
927       :name CAiG@gateway.com)
928   :receiver (set
929     (agent-identifier
930       :name DSA@dispatch.com))
931

```

```

932 :ontology fipa-nas
933 :language fipa-sl
934 :protocol fipa-request
935 :content "(
936   (done
937     (action
938       (agent-identifier :name CAiG@gateway.com)
939       (open-comm-channel
940         (comm-channel :name gprs :target-addr iiop://paramedic.com/acc))))))"
941

```

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

```

943 (inform
944   :sender
945     (agent-identifier
946       :name MAiM@paracom.com)
947   :receiver (set
948     (agent-identifier
949       :name PSA@paracom.com))
950 :ontology fipa-nas
951 :language fipa-sl
952 :protocol fipa-subscribe
953 :conversation-id subscription-3105
954 :content "(
955   (qos-information
956     (comm-channel
957       :name GPRS
958       :target-addr wap://paramedic.com:1234/acc)
959     (qos
960       :status disconnected))))"
961
962

```

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

```

963 (cancel
964   :sender
965     (agent-identifier
966       :name PSA@paracom.com)
967   :receiver (set
968     (agent-identifier
969       :name MAiM@paracom.com))
970 :ontology fipa-nas
971 :language fipa-sl
972 :protocol fipa-subscribe
973 :content "(
974   (iota ?x
975     (exists ?y
976       (and
977         (qos-matches ?x
978           (qos-information
979             (comm-channel
980               :name gprs
981               :target-addr wap://paramedic.com:1234/acc)
982             (qos :status ?y))))
983         (or (= ?y connected) (= ?y disconnected))))))"
984
985
986

```

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

```

987 (subscribe
988   :sender
989     (agent-identifier
990       :name DSA@dispatch.com)
991   :receiver (set
992

```

```

995     (agent-identifier
996       :name MAiG@gateway.com))
997 :ontology fipa-nas
998 :language fipa-sl
999 :protocol fipa-request
1000 :content "(
1001   (iota ?x
1002     (and
1003       (time-constraint (time-type :value every) (time-value :value 10 :unit s))
1004       (qos-matches ?x
1005         (qos-information
1006           (comm-channel
1007             :name gprs
1008             :target-addr iiop://paramedic.comm/acc))))))")

```

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

```

1011 (query-ref
1012   :sender
1013     (agent-identifier
1014       :name DSA@dispatch.com)
1015   :receiver (set
1016     (agent-identifier
1017       :name MAiG@gateway.com))
1018   :ontology fipa-nas
1019   :language fipa-sl
1020   :protocol fipa-query
1021   :content "(
1022     (iota ?x
1023       (qos-information
1024         (comm-channel
1025           :name gprs)
1026         (qos
1027           :throughput ?x))))")

```

11. Message 11 inform: MAiG informs the DSA the current QoS of the GPRS MTC.

```

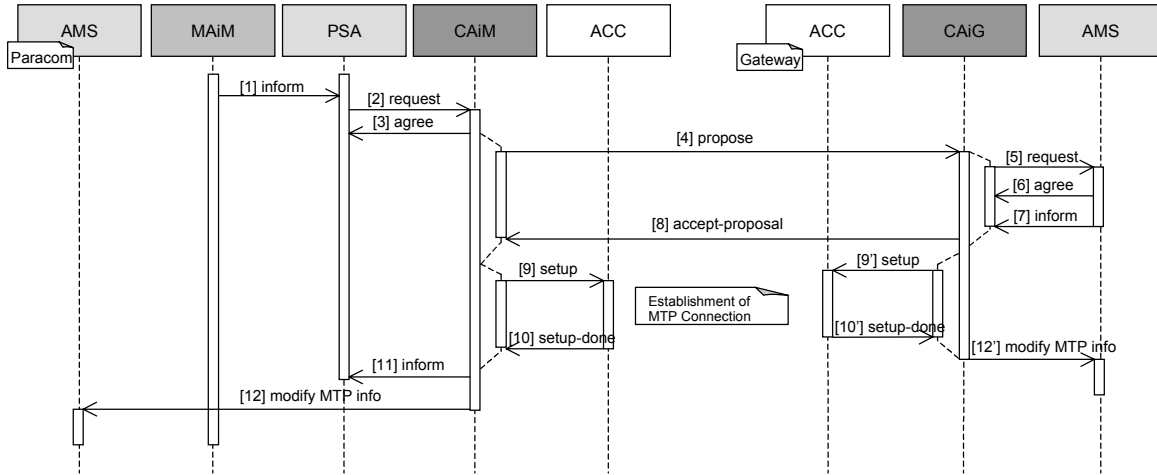
1031 (inform
1032   :sender
1033     (agent-identifier
1034       :name MAiG@gateway.com)
1035   :receiver (set
1036     (agent-identifier
1037       :name DSA@dispatch.com))
1038   :ontology fipa-nas
1039   :language fipa-sl
1040   :protocol fipa-query
1041   :content "(
1042     (= (iota ?x
1043       (qos-information
1044         (comm-channel
1045           :name gprs)
1046         (qos
1047           :throughput ?x)))
1048       (rate-value
1049         :direction outbound
1050         :unit kbits/s
1051         :value 20))))")

```

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

1056 **6.2.2 Example Negotiation of a Message Transport Protocol**

1057 When the mobile host roams from the ATM network to the GPRS network – after the reconnection – the PSA receives
 1058 the information from MAiM that the *Paracom* AP is now connected to the GPRS MTC. The PSA reasons that the
 1059 *fipa.mts.mtp.wap.std* MTP is better in that environment and it requests the CAiM to establish this MTP between
 1060 ACCiM and ACCiG. Also, CAiM proposes the establishment of this MTP to CAiG, which accepts the proposal, and they
 1061 command their respective ACCs to set it up. As a last action, both CAiF and CAiG modify the AP descriptions of their
 1062 APs. The message flow is illustrated in *Figure 7*.
 1063



1064 **Figure 7: Example Negotiation of a Message Transport Protocol**

- 1065
- 1066 1. Message 1 *inform*: MAiM informs the PSA that the *Paracom* AP is now connected to the GPRS network.

```

1067
1068
1069
1070 (inform
1071   :sender
1072     (agent-identifier
1073       :name MAiM@paracom.com)
1074   :receiver (set
1075     (agent-identifier
1076       :name PSA@paracom.com))
1077   :ontology fipa-nas
1078   :language fipa-sl
1079   :protocol fipa-subscribe
1080   :conversation-id subscription-3106
1081   :content "(
1082     (qos-information
1083       (comm-channel
1084         :name gprs
1085         :target-addr wap://paramedic.com:1234/acc)
1086       (qos
1087         :status connected)))")
1088

```

l089 2. **Message 2 request and message 3 agree:** The PSA requests CAiM to establish the fipa.mts.mtp.wap.std
 l090 MTP between ACCiM and ACCiG.

```
l091 (request
l092   :sender
l093     (agent-identifier
l094       :name PSA@paracom.com)
l095   :receiver (set
l096     (agent-identifier
l097       :name CAiM@paracom.com))
l098   :ontology fipa-nas
l099   :language fipa-sl
l100   :protocol fipa-request
l101   :content "(
l102     (action
l103       (agent-identifier
l104         :name CAiM@paracom.com)
l105       (activate (sequence
l106         (transport-protocol
l107           :name fipa.mts.mtp.wap.std
l108           :gw-addr wap://gateway.com:1234/acc))))))")
l110
```

l111 3. **Message 4 propose:** CAiM sends a propose message to the CAiG.

```
l112 (propose
l113   :sender
l114     (agent-identifier
l115       :name CAiM@paracom.com)
l116   :receiver (set
l117     (agent-identifier
l118       :name CAiG@gateway.com))
l119   :ontology fipa-nas
l120   :language fipa-sl
l121   :protocol fipa-propose
l122   :content "(
l123     (action
l124       (agent-identifier
l125         :name CAiM@paracom.com)
l126       (use
l127         (transports
l128           :send (sequence
l129             (transport-protocol
l130               :name fipa.mts.mtp.wap.std))
l131           :recv (sequence
l132             (transport-protocol
l133               :name fipa.mts.mtp.wap.std))))))
l134     true)")
l135
```

l136
 l137 4. **Message 5 request, message 6 agree and message 7 inform:** CAiG requests the local AP description to find
 l138 out if the fipa.mts.mtp.wap.std MTP is supported (see [FIPA00023]).
 l139

l140 5. **Message (8) accept-proposal:** CAiG accepts CAiM's proposal to use the fipa.mts.mtp.wap.std MTP.

```
l141 (accept-proposal
l142   :sender
l143     (agent-identifier
l144       :name CAiG@gateway.com)
l145   :receiver (set
l146     (agent-identifier
l147       :name CAiM@paracom.com))
l148   :ontology fipa-nas
l149   :language fipa-sl
l150
```

```

l151 :protocol fipa-propose
l152 :content "(
l153   (action
l154     (agent-identifier :name CAiM@paracom.com)
l155     (use
l156       (transports
l157         :send (sequence (transport-protocol :name fipa.mts.mtp.wap.std))
l158         :recv (sequence (transport-protocol :name fipa.mts.mtp.wap.std))))))
l159   (transport-selection
l160     (transports
l161       :send (sequence (transport-protocol :name fipa.mts.mtp.wap.std))
l162       :recv (sequence (transport-protocol :name fipa.mts.mtp.wap.std))))))")
l163

```

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.

```

l169 (inform
l170   :sender
l171     (agent-identifier
l172       :name CAiM@paracom.com)
l173   :receiver (set
l174     (agent-identifier
l175       :name PSA@paracom.com))
l176   :ontology fipa-nas
l177   :language fipa-sl
l178   :protocol fipa-request
l179   :content "(
l180     (result
l181       (action
l182         (agent-identifier :name CAiM@paracom.com)
l183         (activate
l184           (sequence
l185             (transport-protocol
l186               :name fipa.mts.mtp.wap.std
l187               :gw-addr wap://gateway.com:1234/acc))))
l188         (transport-protocol
l189           :name fipa.mts.mtp.wap.std :gw-addr wap://gateway.com:1234/acc))))")
l190
l191

```

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.

6.2.3 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*.

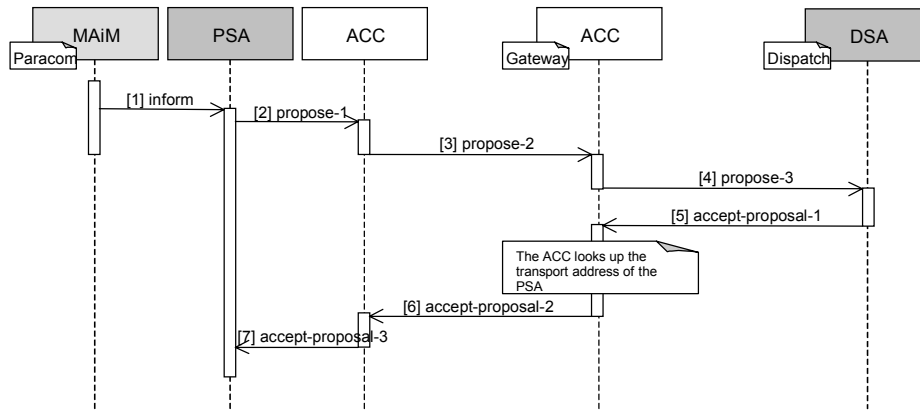


Figure 11: Example Negotiation of a Message Representation

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

```

l200 (inform
l201   :sender
l202     (agent-identifier
l203       :name MAiM@paracom.com)
l204   :receiver (set
l205     (agent-identifier
l206       :name PSA@paracom.com))
l207   :ontology fipa-nas
l208   :language fipa-sl
l209   :protocol fipa-subscribe
l210   :conversation-id subscription-3106
l211   :content "(
l212     (qos-information
l213       (comm-channel name gprs)
l214       (qos :throughput
l215         (rate-value :unit Kbits/s :direction Outbound :value 0.96))))"
l216
l217
l218
l219
l220
l221
l222
    
```

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

```

l226 (propose
l227   :sender
l228     (agent-identifier
l229       :name PSA@paracom.com)
l230   :receiver (set
l231     (agent-identifier
l232       :name DSA@dispatch.com))
l233   :ontology fipa-nas
l234   :language fipa-sl
l235   :protocol fipa-propose
l236   :content "(
l237     (action
l238       (agent-identifier
l239         :name PSA@paracom.com)
l240       (use
l241         (msg-encoding
l242           :send (sequence
l243             (msg-representation
l244               :name fipa.acl.rep.bitefficient.std))
l245           :recv (sequence
l246             (msg-representation
l247               :name fipa.acl.rep.bitefficient.std))))))
l248     true)"
    
```


- 1249
- 1250 3. Message 3 propose-2: The ACC at the mobile host forwards the same message to the ACC at the gateway host.
- 1251
- 1252 4. Message 4 propose-3: The ACC at the gateway host forwards the same message to the PSA.
- 1253
- 1254 5. Message 5 accept-proposal-1: The PSA accepts the proposal and sends a message back to the DSA.
- 1255
- 1256 (accept-proposal
- 1257 :sender
- 1258 (agent-identifier
- 1259 :name DSA@dispatch.com)
- 1260 :receiver (set
- 1261 (agent-identifier
- 1262 :name PSA@paracom.com))
- 1263 :ontology fipa-nas
- 1264 :language fipa-sl
- 1265 :protocol fipa-propose
- 1266 :content "(
- 1267 (action
- 1268 (agent-identifier :name PSA@paracom.com)
- 1269 (use
- 1270 (msg-encoding
- 1271 :send (sequence
- 1272 (msg-representation :name fipa.acl.rep.bitefficient.std))
- 1273 :recv (sequence
- 1274 (msg-representation :name fipa.acl.rep.bitefficient.std))))))
- 1275 (msg-encoding-selection
- 1276 (msg-encoding
- 1277 :send (sequence
- 1278 (msg-representation :name fipa.acl.rep.bitefficient.std))
- 1279 :recv (sequence
- 1280 (msg-representation :name fipa.acl.rep.bitefficient.std))))))")
- 1281
- 1282 6. Message 6 accept-proposal-2: The ACC at the gateway host forwards same message to the ACC at the
- 1283 mobile host.
- 1284
- 1285 7. Message 7 accept-proposal-3: The ACC at the mobile host delivers the same message to the PSA.
- 1286

7 References

- 1287
- 1288 [FIPA00023] FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2000.
1289 <http://www.fipa.org/specs/fipa00023/>
- 1290 [FIPA00036] FIPA Propose Interaction Protocol Specification. Foundation for Intelligent Physical Agents, 2000.
1291 <http://www.fipa.org/specs/fipa00036/>
- 1292 [FIPA00069] FIPA ACL Message Representation in Bit-Efficient Encoding Specification. Foundation for Intelligent
1293 Physical Agents, 2000.
1294 <http://www.fipa.org/specs/fipa00069/>
- 1295 [FIPA00075] FIPA Agent Message Transport Protocol for IOP Specification. Foundation for Intelligent Physical
1296 Agents, 2000.
1297 <http://www.fipa.org/specs/fipa00075/>
- 1298 [FIPA00076] FIPA Agent Message Transport Protocol for WAP Specification. Foundation for Intelligent Physical
1299 Agents, 2000.
- 1300 [FIPA00094] FIPA Quality of Service Specification. Foundation for Intelligent Physical Agents, 2000.
1301 <http://www.fipa.org/specs/fipa00094/>
- 1302 [ITUE800] Recommendation E.800 - Telephone Network and ISDN, Quality of Service, Network Management and
1303 Traffic Engineering, Terms and Definitions Related to Quality of Service and Network Performance
1304 Including Dependability. International Telecommunication Union, International Telecommunication
1305 Union, 1995.
- 1306 [ITUX135] Recommendation X.135 - Speed of Service (delay and throughput), Performance Values for Public
1307 Data Networks when Providing Packet-Switched Services. International Telegraph and Telephone
1308 Consultative Committee, 1993.
- 1309 [WAP99] Wireless Application Protocol Specification Version 1.2. WAP Forum, 1999.
1310 <http://www.wapforum.org/what/technical.htm>
1311

8 Informative Annex A — ChangeLog

8.1 2001/10/17 - version E by TC Gateways

1314	Page 8, lines 290-291:	Added a new frame <code>subscription-identifier</code> which is used to map subscriptions and subsequent cancel by the <code>subscribe-notification</code> and <code>cancel-notification</code> functions
1315		
1316		
1317	Page 12, lines 340-341:	Replaced predicate <code>qos-notification</code> with function <code>subscribe-notification</code> ; the <code>qos-notification</code> predicate was used as content for <code>subscribe act</code> , which is not used in this specification anymore, thus there is no need for this predicate, and, the <code>subscribe-notification</code> function replaces the <code>subscribe act</code> (in this spec), that is, it is used to subscribe changes in QoS
1318		
1319		
1320		
1321		
1322	Page 12, lines 341-342:	Added new function <code>cancel-notification</code> which replaces the <code>cancel act</code> (in this spec), that is, it is used to cancel previously subscribed notification(s)
1323		
1324	Page 13, lines 346-347:	Added sentence describing the return value of the function
1325	Page 14, lines 364-365:	Added a new refuse reason which is needed by the <code>cancel-notification</code> function
1326	Page 15, line 398:	Removed <code>fipa-subscribe</code> protocol from advertised protocols
1327	Pages 22-27, lines 799-1014:	“Message Exchange over WAP MTP” section removed because: (1) the example uses dynamic registration, and, (2) the functionality can be better implemented using FIPA messaging interoperability specification and FIPA message buffering specification
1328		
1329		
1330	Page 30, lines 1117-1119:	Figure 9 updated
1331	Page 30, lines 1127-1145:	Example ACL message updated to follow new subscription method
1332	Page 32, line 1216-1234:	Example ACL message updated to follow new subscription method
1333	Page 32, lines 1236-1268:	The <code>cancel</code> method replaced with the new one which includes replacing the <code>cancel ACL message</code> with <code>request</code> , <code>agree</code> and <code>inform</code> messages (<code>fipa-request</code>)
1334		
1335	Page 34: lines 1268-1290:	The <code>subscribe</code> method replaced with the new one which includes replacing the <code>subscribe ACL message</code> with <code>request</code> , <code>agree</code> and <code>inform</code> messages (<code>fipa-request</code>)
1336		
1337		
1338	Page 34, line 1290:	Updated message number
1339	Page 34, line 1310:	Updated message number
1340	Pages 34-35, lines 1312-1332:	Example ACL message updated to follow new subscription method
1341	Page 35, line 1334:	Updated message numbers
1342	Page 35, lines 1350-1368:	Example ACL message updated to follow new subscription method
1343	Page 38, lines 1496-1534:	Example ACL message updated to follow new subscription method
1344	Page 41, lines 1599-1600:	Removed reference to <code>fipa-subscribe</code> [FIPA00035]
1345		

8.2 2002/09/13 - version F by TC X2S

1346	Entire document:	Changed all ontology terms to lowercase
1347	Entire document:	Ontology name changed from <code>FIPA-Nomadic-Application</code> to <code>fipa-nas</code>
1348	Entire document:	Examples updated according to other modifications
1349	Entire document:	Examples updated according to other modifications
1350	Page 1, lines 102–103:	Removed reference to QoS ontology from the list of specification contents
1351	Page 1, lines 105–107:	Removed reference to WAP MTP and added references to bit-efficient message envelope and to QoS ontology specifications
1352		
1353	Page 2, lines 133–139:	Removed paragraph about WAP MTP
1354	Page 2, lines 160–161:	Removed reference to QoS ontology
1355	Page 5, lines 266–268:	Removed the <code>qos</code> frame (moved to [FIPA00094])
1356	Page 6, lines 269–272:	Removed the <code>rate-value</code> frame (moved to [FIPA00094])
1357	Page 7, lines 273–276:	Removed the <code>time-value</code> frame (moved to [FIPA00094])
1358	Page 7, lines 277–280:	Removed the <code>probability-value</code> frame (moved to [FIPA00094])
1359	Page 8, lines 281–284:	Removed the <code>change-constraint</code> frame (moved to [FIPA00094])
1360	Page 8, lines 285–288:	Removed the <code>time-constraint</code> frame (moved to [FIPA00094])
1361	Page 8, lines 289–292:	Removed the <code>subscription-id</code> frame (moved to [FIPA00094])
1362	Page 8, lines 293–297:	Removed the <code>comm-channel</code> frame (moved to [FIPA00094])

|363 **Page 9, lines 297–300:** **Removed the transport-protocol frame (moved to [FIPA00094])**
|364 **Page 11, lines 340–341:** **Removed the qos-information predicate (moved to [FIPA00094])**
|365 **Page 11, line 340:** **Added a transport-selection predicate**
|366 **Page 11, line 340:** **Added an msg-encoding-selection predicate**
|367 **Page 12, lines 343–344:** **Removed the subscribe-notification function (moved to [FIPA00094])**
|368 **Page 13, lines 345–346:** **Removed the cancel-notification function (moved to [FIPA00094])**
|369 Page 14, lines 362–364: Replaced the reference to the `fipa-agent-management not-understood`
|370 exception predicates with actual predicates
|371 Page 15, lines 366–368: Replaced the reference to the `fipa-agent-management refusal` exception
|372 propositions with the actual propositions
|373

|374 **8.3 2002/11/01 - version G by TC X2S**

|375 Entire document: Updated subscription examples to use `fipa-subscribe` protocol
|376