The ATM Forum

Technical Committee

PNNI Augmented Routing (PAR) Version 1.0

AF-RA-0104.000

January, 1999

The ATM Forum
Worldwide Headquarters
2570 West El Camino Real, Suite 304
Mountain View, CA 94040-1313
Tel: +1-650-949-6700
Fax:+1-650-949-6705

Acknowledgments

Much work went into the development of the PAR specification. Numerous contributions from the ATM Forum and comments from participants of the RA working group enabled the specification to be completed. The editor would like to recognize the following members (in alphabetical order) for their significant contributions to this work:

Vivek Bansal
Ravi Raj Bhat
Ross Callon
Rao Cherokuri
Leslie Collica
David Cypher
Tim Dwight
Douglas Dykeman
Gérard Gastaud
Robert Haas
Joel Halpern
Atsushi Iwata
Jason Jeffords
Tony Przygienda
Tricci So
Mickey Spiegel
Balaji Srinivasan
Gauthaman Vasudevan
Colin West

Moreover a general recognition is due to the former PNNI and current RA working group in which this work was carried out.

Patrick Droz, Editor

Contents

Figures

1. Introduction

[Informative]

1.1 Overview

PNNI Augmented Routing (PAR) is an extension to PNNI routing to allow information about non-ATM services to be distributed in an ATM network as part of the PNNI topology. The content and format of the information is specified by PAR but is transparent to PNNI routing.

A PAR-capable device, one that implements PNNI and the PAR extension, is able to create PAR PTSEs that describe the non-ATM services located on or behind that device. Because this information is flooded by PNNI routing, PAR-capable devices are also able to examine the PAR PTSEs in the topology database that were originated by other nodes to obtain information on desired services reachable through the ATM network.

An important example of how PAR can be used is provided by overlay routing on ATM backbones. If the routers are PAR-capable, they can create PTSEs to advertise the routing protocol supported on the given interface (e.g., OSPF, RIP, or BGP), along with their IP address and subnet, and other protocol-specific details.

The PAR-capable routers can also automatically learn about "compatible" routers (e.g., supporting the same routing protocol, in the same IP subnet) active in the same ATM network. In this manner the overlay routing network can be established automatically on an ATM backbone. The mechanism is dynamic, and does not require configuration.

One potential drawback of PAR is that a device must implement PNNI in order to participate. Therefore, an additional set of optional protocols called Proxy PAR has been defined to allow a client that is not PAR-capable to interact with a server that is PAR-capable and thus obtain the PAR capabilities. The server acts as a proxy for the client in the operation of PAR. The client is able to register its own services, and query the server to obtain information on compatible services available in the ATM network.

A key feature of PAR and Proxy PAR is the ability to provide VPN support in a simple yet very effective manner. All PAR information is optionally tagged with a VPN ID and can therefore be filtered on that basis. This can be used for example, in a service provider network. Each customer can be provided with a unique VPN ID that is part of all Proxy PAR registrations and queries. Usage of the correct VPN ID can easily be enforced at the Proxy PAR server. In this way the services of a given customer will be available only to clients in that customer's network.

1.1.1 Overview of PNNI Augmented Routing (PAR)

PNNI Augmented Routing (PAR) is an extension to PNNI to allow the flooding of information about non-ATM devices. PAR uses a new PTSE type to carry this non-ATM-related information. The current version of PAR specifies IGs for the flooding of IPv4-related protocol information such as OSPF or BGP. In addition, PAR also allows the use of the System Capabilities IG, which can be used to carry proprietary or experimental information.

PAR supports extensive filtering possibilities, which allow the implementation of virtual privat networks (VPN). As PAR is a PNNI extension, it can reuse existing PNNI routing level scopes. In addition, PAR provides filtering in terms of a VPN ID, IP address, including a subnet mask, as well as protocol flags. The correct filtering according to these parameters is part of a PAR implementation.

1.1.2 Overview of Proxy PAR

Proxy PAR is a protocol that allows for different ATM attached devices to interact with PAR-capable switches and obtain information about non-ATM services without executing PAR themselves. The client side is much simpler in terms of implementation complexity and memory requirements than a complete PAR instance and should allow easy implementation in, for example, existing IP routers. Clients can use Proxy PAR to register different non-ATM services and protocols they support. This protocol has deliberately **not** been included as part of ILMI owing to the complexity of PAR information passed in the protocol and the fact that it is intended for integration of non-ATM protocols and services only. A device executing Proxy PAR does not necessarily need to execute ILMI or UNI signaling, although this will normally be the case.

The protocol does not specify how the distributed service registration and data delivered to the client are supposed to drive other protocols. For example, OSPF routers finding themselves through Proxy PAR could use this information to form a full mesh of P2P VCs and communicate using RFC1483 [RFC1483] encapsulation. In terms of the discovery of other devices such as IP routers, Proxy PAR is an alternative to LANE [LANE] or MARS [RFC2022]. It is expected that the guidelines defining how a certain protocol can make use of Proxy PAR and PAR should come from the group or standardization body that is responsible for the particular protocol.

PAR and Proxy PAR have the ability to provide ATM address resolution for IP attached devices, but such resolution can also be achieved by other protocols under specification in IETF. However, the main purpose of the protocol is to allow the automatic detection of devices over an ATM cloud in a distributed fashion, not relying on a broadcast facility. Finally, it should be mentioned that the protocol complements and coexists with server detection via ILMI extensions.

1.2 Reference Model

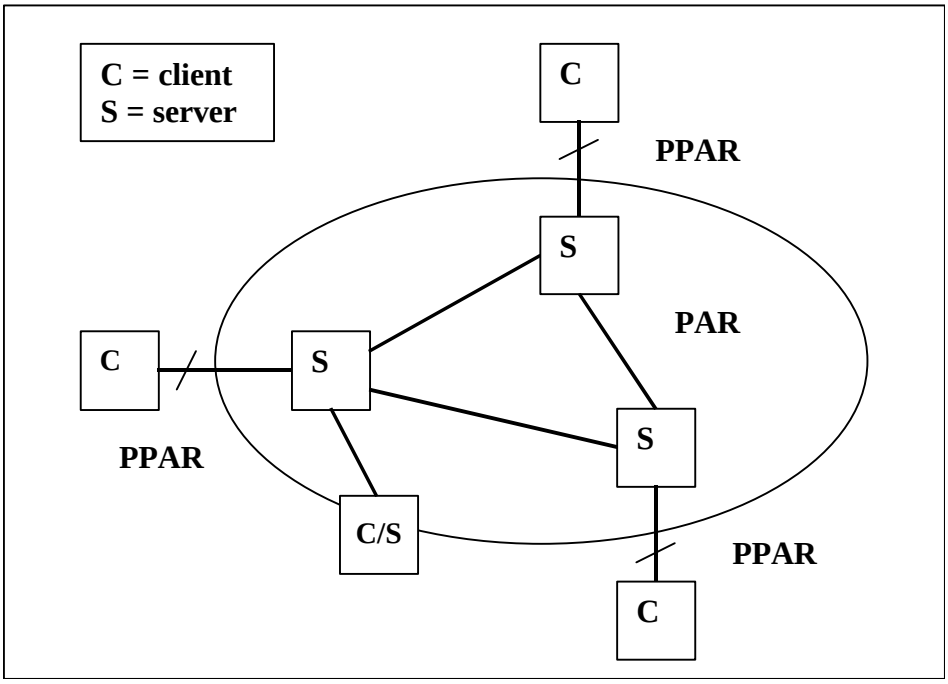

Figure 1-1: PPAR/PAR Reference Model

The device C/S is not using Proxy PAR for the communication between client and server. Such a device could be an IP router with full PAR capability but using an internal way to exchange non-ATM information with PAR.

1.3 Document Organization

The PAR Specification is organized into five main sections:

- *Chapter 2 – Terminology* provides a listing of Abbreviations and Definitions used in this specification.
- *Chapter 3 – Overview of PAR Mechanisms* describes the basic operation of PAR.
- *Chapter 4 – PAR Specific IGs and PTSEs* provides the detailed definition of the Packet and Information Group formats used by PAR.
- *Chapter 5 – Proxy PAR Protocol Specification* provides the detailed definition of the Proxy PAR protocol.
- *Chapter 6 – Proxy PAR Specific Packet Formats* provides the detailed definition of the Proxy PAR packets.
- *Chapter 7 – Architectural Variables* provides the values of the architectural variables.
- *Chapter 8 – References*.

- *Annex A – PAR PICS* provides the PICS for PAR.
- *Appendix A – OSPF Example* provides a simple way to run OSPF over PAR and Proxy PAR.
- *Appendix B – VPN Support* provides a description of possible filtering in a VPN environment.

Annexes are formal clarifications of material in the document, and are part of the specification. Appendices are included for clarification, but are not part of the formal PAR protocol.

2. Terminology

[Informative]

2.1 Acronyms

ARA	Address Resolution Advertisement
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
DR	Designated Router
ID	Identifier
IE	Information Element
IG	Information Group
ILMI	Integrated Local Management Interface
IP	Internet Protocol
LANE	LAN Emulation
LGN	Logical Group Node
LSA	Link State Advertisement
MARS	Multicast Address Resolution Server
MOSPF	Multicast OSPF
MPOA	Multi-protocol over ATM
NHRP	Next Hop Resolution Protocol
OSPF	Open Shortest Path First
OUI	Organizational Unit Identifier
PAR	PNNI Augmented Routing
PG	Peer Group
PGL	Peer Group Leader
PIM-DM	Protocol Independent Multicast Dense Mode
PIM-SM	Protocol Independent Multicast Sparse Mode
PNNI	Private Network-to-Network Interface
PPAR	Proxy PNNI Augmented Routing
PTSE	PNNI Topology State Element
PTSP	PNNI Topology State Packet
RIP	Routing Information Protocol
TLV	Type, Length, Value
UNI	User Network Interface
VCI	Virtual Channel Identifier
VPI	Virtual Path Identifier
VPN	Virtual Private Network

2.2 Normative Statements

The normative sections of this specification are Chapter 4 to 7, and all Annexes. Throughout these normative sections, normative statements are used as follows:

Table 2-1: Normative Statements

Statement	Verbal Form
Requirement	must/must not
Recommendation	should/should not
Permission	may

The term “may” is used to indicate that a particular procedure is allowed but not required. It is an implementation choice. The term “may” is also used to indicate allowed behaviors that must be accommodated.

3. OVERVIEW OF PAR MECHANISMS

[Informative]

3.1 PAR

3.1.1 General Mechanisms

PAR can be regarded as a standardized way to distribute and replicate non-ATM-related information via PNNI. One key application for such a service is the automatic detection and configuration of non-ATM devices such as IP routers. PAR-capable devices can generate and interpret such IP-related information. Non-PAR-capable switches can simply flood the information according to PNNI scoping rules.

3.1.2 PAR Extensions to PNNI

PAR requires a new PTSE to carry non-ATM information. The new PTSE is passed transparently by PNNI nodes that do not support PAR. In addition, some new functions have to be provided in PAR to access, construct, and filter non-ATM information.

The IGs inside the new PTSE carry information such as the OSPF area ID or router priority. The distributed information of a certain protocol is relevant for the bring-up of the protocol. For instance the OSPF area ID is needed so that the correct set of routers (in the same area) will find each other. The router priority is used for the designated router (DR) election, it influences whether a certain router will try to contact its neighbors.

A PAR-capable device must be able to generate and understand PTSEs that carry PAR-related IGs including the System Capabilities IG that can be used for proprietary and experimental implementations.

In addition, a PAR device must be able to do the additional filtering based on IP-related information contained in some IGs. The filtering is based on a VPN ID, IP address, including a subnet mask, as well as a protocol flag. With the help of this filter function, selective discovery can be implemented.

3.1.2.1 PAR PTSE

A PTSE that carries PAR IGs. The coding and structure is exactly the same as in PNNI. PAR PTSEs must have all information group tag bits set to 0. In particular this is the Mandat, D-Sum, Trans, as well as the Reserv bit.

3.1.2.2 PAR IGs

The set of additional IGs to carry PAR-related information.

3.1.2.3 Propagation across the hierarchy

The PAR extensions rely on appropriate flooding of information by the PNNI protocol. Nodes within the PNNI network take into account the associated scope of the information when it is flooded. It is thus possible to exploit the PNNI routing hierarchy by announcing different protocols on different levels of the hierarchy, e.g. OSPF could run inside certain peer groups, whereas BGP could run between the set of peer groups running OSPF. Such an alignment or mapping of non-ATM protocols to the PNNI hierarchy can drastically increase the scalability of PAR service. As the scope of the PAR information can indicate that a distribution beyond the boundaries of the peer group is necessary, the PGL of a peer group has to collect such information and propagate it into a higher layer of the PNNI hierarchy. Additionally, changes received at a lower layer have to trigger appropriate changes at higher PNNI layers as well.

As no assumptions except scope values can be made about the information distributed (e.g. IP addresses bound to AESAs are not assumed to be aligned with them in any respect such as encapsulation or functional mapping), such information cannot be summarized. This makes a careful handling of scopes necessary to preserve the scalability of the approach as described above.

3.2 Proxy PAR

3.2.1 Client Interaction Description

The protocol is asymmetric and consists of a discovery and query/registration part. The discovery is very similar to the existing PNNI Hello protocol and is used to initiate and maintain communications between adjacent clients and servers. The registration and update part execute after a Proxy PAR adjacency has been established. The client can register its services by sending registration messages to the server. The client obtains information of interest by sending query messages to the server. When the client needs to change its set of registered protocols it has to re-register its services. The client can withdraw all registered services by registering the null set. A similar result can be obtained by withholding the Hello packets to drop the adjacency formed. It is important to note that the server side does not push new information to the client, neither does the server keep any state describing which information the client received. It is the responsibility of the client to update and refresh its information and to discover new clients or update its stored information about other clients by issuing queries and registrations at appropriate time intervals. This simplifies the protocols, but assumes that the client will not store and request large amounts of data. The main responsibility of the server is to flood the registered information through the PNNI cloud such that potential clients can discover each other. It is assumed that services advertised by Proxy PAR will be advertised by a small number of clients and will be fairly stable, so that polling and refreshing intervals can be relatively long.

3.2.2 Interaction of Proxy PAR with PAR

When the client side registers or re-registers a new service through Proxy PAR, it associates a PNNI membership scope with the service that restricts the flooding of the service definition within the PNNI network. The Proxy PAR server translates the membership scope into a PNNI routing level. The Proxy PAR server may limit the flooding based on policies.

The Proxy PAR Server is also responsible for the translation of Proxy PAR registration messages into PTSEs and constructing appropriate service descriptions out of PTSEs. The Proxy PAR protocol is simpler than PNNI with PAR and therefore, has less functionality.

3.2.3 Extensions to ILMI and/or UNI for Detection of Proxy PAR Devices

In case Proxy PAR and PAR detection via ILMI becomes available it will be specified in the framework of ILMI. The detection via ILMI can also be used to obtain a non-default VPI/VCI on which the Proxy PAR protocol is running.

In case no ILMI detection is present the client can start to send Hellos on the predefined VCI to the server. A Proxy PAR-capable server will respond by sending Proxy PAR Hellos back to the client.

4. PAR Specific IGs and PTSE

[Normative]

This chapter describes the Information Groups and PTSE necessary for communication among PAR-capable devices.

4.1 Information Group Summary

The following table summarizes the information group types used. The list includes only the IG types used by PAR. The types of IGs defined for PNNI can be found in the PNNI specification.

Table 4-1: Information Group Summary

Type	IG Name	Nested in
768	PAR Service IG	PTSE (64)
776	PAR VPN ID IG	PAR Service IG (768)
784	PAR IPv4 Service Definition IG	PAR VPN ID IG (776) / PAR Service IG (768)
800	PAR IPv4 OSPF Service Definition IG	PAR IPv4 Service Definition IG (784)
801	PAR IPv4 MOSPF Service Definition IG	PAR IPv4 Service Definition IG (784)
802	PAR IPv4 BGP4 Service Definition IG	PAR IPv4 Service Definition IG (784)
803	PAR IPv4 DNS Service Definition IG	PAR IPv4 Service Definition IG (784)
804	PAR IPv4 PIM-SM Service Definition IG	PAR IPv4 Service Definition IG (784)

System Capabilities IGs can appear on every level of the nesting.

4.2 PAR Service IG

This Information Group permits services to be registered for a specific AESA. A scope controls the distribution of information. All services for the same AESA with the same scope must appear in the same PAR Service IG. This Information Group can appear multiple times and in multiple different PTSEs, with different AESAs and/or scopes. The PAR Service IG is a restricted IG.

Table 4-2: PAR Service IG

Offset	Size (Octets)	Name	Function/Description
0	2	Type	Type = 768 (PAR Service IG)
2	2	Length	
4	1	Scope	A PNNI routing level. The scope of advertisement applies to all services included in the information group.
5	3	Reserved	
8	20	ATM Address	AESA for which the service has been registered

Inside of the PAR Service IG, different optional PAR VPN ID or PAR Service Definition Information Groups may be present.

4.2.1 PAR VPN ID

This Information Group contains the virtual private network (VPN) identifier (ID). The VPN ID is structured into two parts. The first three bytes contain the OUI whereas the second 4 bytes represent the VPN index. A VPN index alone is normally not unique only together with the OUI part.

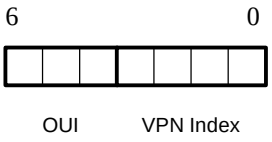

Figure 4-1 VPN ID Format

The VPN ID IG has the following structure.

Table 4-3 PAR VPN ID IG

Offset	Size (Octets)	Name	Function/Description
0	2	Type	Type = 776 (PAR VPN ID IG)
2	2	Length	
4	1	Reserved	
5	7	VPN ID	Virtual private network identifier consisting of OUI and VPN index.

The IG can appear multiple times with different VPN IDs. If the same VPN ID IG appears more than once the first must be taken. The IG can be absent completely.

4.2.1.1 PAR Service Definition IGs

These Information Groups can only occur inside a PAR VPN ID IG or directly in a PAR Service IG, and each describes a specific service with associated data for the specified AESA and scope and VPN ID contained in the enclosing PAR Service IG and VPN ID IG. At present, only IPv4 specific information is supported.

4.2.1.1.1 PAR IPv4 Service Definition IG

In order to distribute information about IPv4-related protocols and services, the IPv4 Service Definition IG has been defined. The IG contains all necessary information about the IPv4 address (IP address, and subnet mask) and a bitmask to indicate the protocols and services bound to the IP address. The only service currently defined is the registration of IPv4 interfaces with associated services.

Table 4-4: PAR IPv4 Service Definition IG

Offset	Size (Octets)	Name	Function/Description
0	2	Type	Type = 784 (PAR IPv4 Service Definition IG)
2	2	Length	
4	4	IP address	The IP address or IP address prefix for which the protocols are registered or queried.
8	4	IP address mask	Contiguous mask of the IP address or IP address prefix. A mask length of 0 specifies any IP address.
12	8	Service Mask	Bitmask of registered services.
			Bit[1] Protocol 01 OSPF 02 RIP 03 RIPv2

[1] Bit 1 is the least significant bit in network order.

			04	BGP3
			05	BGP4
			06	EGP
			07	IDPR
			08	MOSPF
			09	DVMRP
			10	CBT
			11	PIM-SM
			12	IGRP
			13	IS-IS
			14	ES-IS
			15	ICMP
			16	GGP
			17	BBN SPF IGP
			18-58	Reserved, must be set to 0 at the originator and ignored on reception.
			59	PIM-DM
			60	MARS
			61	NHRP
			62	ATMARP
			63	DHCP
			64	DNS

For some of the services indicated in the bitmask field, additional IGs have been defined to distribute more information about the protocol or service. It is expected that more IGs maybe defined in the future that can be embedded in the PAR IPv4 Service Definition IG. They will optionally contain information per service registered to allow clients to differentiate between registrations based on service specific data.

4.2.1.1.1.1 PAR OSPF IPv4 Service Definition IG

OSPF [RFC2178] Service Definition IG contains the Area ID of the interface registering with the service. It allows routers querying for their OSPF peers to distinguish between routers sharing the same IP network but operating in different areas. Although a small optimization in terms of packets, this can result in significant savings in terms of SVCs to be established. The OSPF router priority used in the designated router election is included as well. In addition, the interface type is contained which influences the operation of OSPF. Additional information contained in the IG allows a more efficient bring up of OSPF.

Table 4-5: PAR OSPF Service Definition IG

Offset	Size (Octets)	Name	Function/Description
0	2	Type	Type = 800 (PAR OSPF IPv4 Service Definition IG)
2	2	Length	
4	4	Area ID	OSPF Area ID running on the registered IPv4Address.
8	1	Router Priority	DR election priority
9	1	Interface type	Value Type 0 Unspecified 1 point-to-point 2 broadcast 3 NBMA 4 Point-to-MultiPoint 5 Virtual link

			all other values are reserved, and must be set to 0 at the originator and ignored on reception.
10	2	Reserved	

This IG can appear only once in a PAR IPv4 service definition IG. In case of multiple instances the first one found must be taken.

A simple example of how to run OSPF is given in Appendix A.

4.2.1.1.1.2 PAR MOSPF IPv4 Service IG

MOSPF IPv4 IG is identical to the OSPF IPv4 Service Definition IG except that the type is set to PAR MOSPF IPv4 Service Definition IG. The Type of the IG is set to 801.

This IG can appear only once in a PAR IPv4 service definition IG. In case of multiple instances the first one found must be taken.

4.2.1.1.1.3 PAR BGP4 IPv4 Service IG

BGP4 [RFC1771] Service Definition IG indicates the autonomous system number, which routers can use, for example, to enforce peering policies. In addition, the BGP identifier can be set. For route reflector configuration [RFC1966], the necessary configuration information can be included as well.

Table 4-6: PAR BGP4 Service Definition IG

Offset	Size (Octets)	Name	Function/Description
0	2	Type	Type = 802 (PAR BGP IPv4 Service Definition IG)
2	2	Length	
4	4	AS	Autonomous System Number
8	4	BGP Identifier	Identifier of BGP4 router
12	4	RR Cluster ID	Route Reflector Cluster ID
16	4	RR Type	Value Function 0 Non-Client Peer 1 Client Peer 2 Route Reflector all other values are reserved and must be set to 0 at the originator and ignored on reception.
20	4	Route Reflector ID	ID of the Route Reflector, if RR Type different from Route Reflector and Client Peer, can be ignored.

The included fields are BGP configuration parameters described in the BGP and route reflector specifications.

This IG can appear only once in an PAR IPv4 service definition IG. In case of multiple instances the first one found must be taken.

4.2.1.1.1.4 PAR DNS IPv4 Service IG

The DNS Service Definition IG is optional and indicates the domain the client can use to contact DNS servers. This Information Group may occur multiple times in PAR IPv4 Service Definition IG if multiple primary domain names are to be registered. The content provided can be used in many ways. For example, it enables clients to find their resolving servers easily without specifying their addresses in configuration files, or primary and secondary servers to detect each other over an ATM cloud.

Table 4-7: PAR DNS Service Definition IG

Offset	Size (Octets)	Name	Function/Description
0	2	Type	Type = 803 (PAR DNS IPv4 Service Definition IG)
2	2	Length	
4	DomainLength	Domain	Domain Name for which the server is acting as primary name server as an ASCII string terminated by a 0x00 octet.
DomainLength+4	0-3	Padding	0 to 3 octets of 0x00. The size of the padding field is calculated using the following formula: (4 – (DomainLength modulo 4)) modulo 4

4.2.1.1.1.5 PAR PIM-SM IPv4 Service Definition IG

The PIM-SM [RFC2117] Service Definition IG is optional. The IG is used for automatic discovery of PIM-SM routers. It contains the router priority used in the DR election.

This IG can appear only once in an PAR IPv4 Service Definition IG. In case of multiple instances the first one found must be taken.

Table 4-7: PAR PIM-SM Service Definition IG

Offset	Size (Octets)	Name	Function/Description
0	2	Type	Type = 804 (PAR PIM-SM IPv4 Service Definition IG)
2	2	Length	
4	4	Priority	Priority for DR election.

4.2.1.1.1.6 Additional IPv4 Service IG

For many other services such as PIM-DM or IS-IS [IS-IS] it may prove beneficial to define specific Service Definition IG to provide more configuration information.

4.2.2 System Capabilities IG

In order to allow experimental and proprietary extensions to Proxy PAR, the PNNI System Capability IG (PNNI 5.14.13 and Errata) can be included. The packet format and scope of the System Capability IG is the same as that specified in the PNNI specification 1.0 chapter 5.14.13 plus Errata.

4.3 PAR PTSE

The PAR Service IG is a restricted information group, and therefore PNNI PTSEs of this type must be created. The correct nesting of IGs in such a PTSE as well as the format of these IGs is specified above. A PAR-capable device must not create empty PAR PTSEs In addition, a PAR PTSE must contain all registered information for the included AESA and scope. A Proxy PAR capable server is responsible for aggregating information with the same AESA and scope as registered over multiple packets into one PTSE.

All PAR IGs defined in this specification must have all information group tag bits set to 0. In particular this is the Mandat, D-Sum, Trans, as well as the Reserv bit.

4.3.1 TLV Encoding

PTSE encoding is specified by PNNI v 1.0. The encoding of the PAR specific IGs is specified above. All IG encoding rules in PNNI apply to PAR IGs as well. In particular this includes the information group tags specified in section 5.14.2 of PNNI v1.0.

4.4 PGL Behavior

When processing PAR PTSEs a PAR capable PGL must look at the PAR Service IG that carries the PNNI relevant information in order to do appropriate flooding. The scope in the PAR Service IG determines the level to which the information has to be flooded inside the PNNI hierarchy. The interpretation of the PAR Service IG is required since the appropriate scope is not part of the PTSP or PTSE header, but rather included in the PAR-specific IG..

If the scope of the PAR information indicates that a distribution beyond the boundaries of the peer group is necessary, the PGL of a peer group has to collect such information and propagate it into a higher layer of the PNNI hierarchy. For purpose of summarization the PGL must pass up an entire PAR service IG intact including all IGs contained therein. When a new instance of a PAR PTSE is received by the PGL, the parent LGN must re-originate all PTSEs containing information from the received PTSE.

The transitive tag must be set to 0 in the PAR service IG to prevent PAR related information from being advertised beyond its scope. Therefore, a non-PAR capable PGL will not pass PAR related information up the hierarchy.. This is in accordance with the standard behavior of PNNI for PTSPs with the transitive tag set to zero.

5. Proxy PAR Protocol Specification

[Normative]
This chapter provides Proxy PAR extension which allows non-PAR-capable devices to communicate with PAR-capable devices in a simplified manner.

5.1 Protocol Packet Processing

Proxy PAR packets must by default be sent and received only on the reserved VPI/VCI values 0/18. Built-in auto-detection mechanisms may allow the invocation of the protocol once the interface supports it, but implementations are possible where the interface on the server side “snoops” for incoming Proxy PAR Hello packets on the default VPI/VCI and starts the protocol when they are received.

Each packet has a PNNI header so that all PNNI header processing rules, such as version negotiation, are applied. Internal format errors, such as length field of TLVs being shorter than the number of fixed fields defined for the specific TLV, should be handled by simply discarding packets unless specified otherwise.

As Proxy PAR is a PNNI extension, all general rules and restrictions that apply to PNNI hold for Proxy PAR as well. For example the maximum packet size for Proxy PAR is restricted to 8192 in the same fashion as for PNNI.

5.1.1 Receiving Proxy PAR Packets

Messages associated with the query and registration protocols should only be passed to the appropriate FSMs when the Hello state machine is in the *PPAR 2-Way* state and should be discarded otherwise.

5.2 Proxy PAR Hello Protocol

The Proxy PAR Hello Protocol is closely related to the Hello protocol specified in [PNNI]. It uses the same packet header and version negotiation methods. For the sake of simplicity, states that are irrelevant to Proxy PAR protocol have been removed from the PNNI Hello protocol. The purpose of the Proxy PAR Hello protocol is to bring up and maintain a Proxy PAR relation between the client and server that supports the exchange of registration and query messages. If the protocol is executed across multiple, parallel links between the same server and client pair, individual registration and query sessions are associated with a specific link. It is the responsibility of the client and server to assign registration and query sessions to the different communication instances. Proxy PAR can be run in the same granularity as ILMI to support virtual links and VP tunnels.

In addition to the PNNI Hello, the Proxy PAR Hellos from the server to the client inform the client about the life time the server assigns to the registered information. The client has to retrieve this interval from the Hello and set its refresh interval to a value below the obtained time interval in order to avoid purging unchanged information by the server.

5.2.1 Proxy PAR Hello Protocol Server and Client Side

The Proxy PAR server and client execute the Hello Protocol until they have established a 2-Way adjacency indicated by the state *PPAR 2-Way*. As mentioned above, this state is required prior to the exchange of registration and query messages. The initial state is *PPAR Down.*

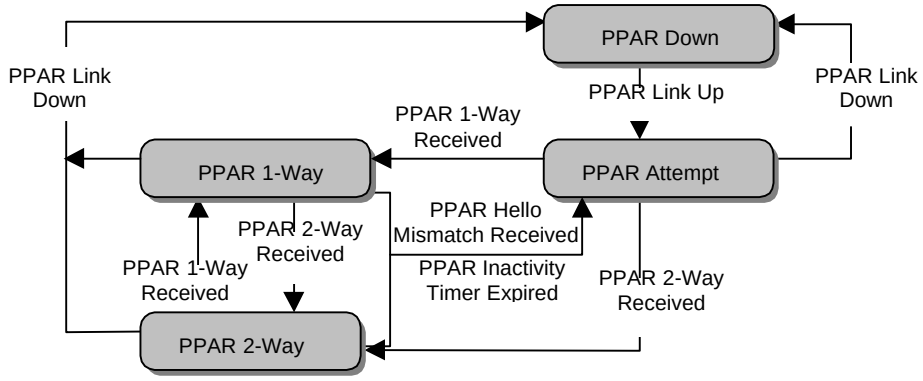

Figure 5-1: Proxy PAR Hello State Machine

The detailed packet format of the Hello Protocol is given in section 6.1.1. Hello packets are sent regularly to keep track of the state at the other end of the link. In case of a link or server failure the server and client have to reinitiate the Hello protocol. Whenever the server exits the *PPAR 2-Way* state it must purge all stored information that was registered by the associated client. Whenever the client exits the state *PPAR 2-Way* it must purge all information received in response to a query previously issued on that link.

5.2.2 The Hello Data Structure

There is a single hello data structure for each of this client/server's physical ports running Proxy PAR protocol and for each logical port (each Virtual Path Connection for which this client/server is an endpoint). Each hello data structure consists of the following items:

State
: The operational status of a link with neighboring client/server. This is described in more detail in Section 5.2.3.

Port ID
: A number assigned by the client/server that identifies which physical port and which virtual path connection, if any, is described by this hello data structure.

Remote ATM End-system Address
: The ATM End-system address of the neighboring client/server on the other end of the link. The Remote ATM End-system address is learned when Hellos are received from the neighboring client/server.

HelloInterval
: The amount of time, in seconds, between Hellos that the client/server sends out over this link, in the absence of event-triggered Hellos.

Hello Timer
: An interval timer that fires every HelloInterval seconds. Whenever the timer fires, the client/server transmits a Hello packet over this link.

InactivityFactor
: The amount of time, in multiples of the HelloInterval declared by the neighboring client/server in its Hellos, before the client/server will consider the link not active, if the neighboring client/server's Hellos cease to arrive.

Inactivity Timer
: A single shot timer whose firing indicates that no Hellos have been received from this neighboring client/server recently. The initial value of the Inactivity Timer must be set to the InactivityFactor times the HelloInterval from the most recent Hello received from the neighboring client/server.

Protocol Version

The current version of the Hello protocol being used for communication with this neighboring client/server. If no acceptable version number has been derived, this field will be zero.

5.2.3 Proxy PAR Hello FSM States

The states that PPAR Hello FSM may attain are described in this section. Figure 5-1 shows a diagram of the possible state changes. The arcs are labeled with the events that cause each state change. These events are described in Section 5.2.4. For a detailed description of the state changes and the actions involved with each state change, see Section 5.2.6.

PPAR Down
The initial state of the PPAR Hello FSM. This state is also reached when lower-level protocols have indicated that the link is not usable. No PPAR registration/query packets will be sent or received over such a link.

PPAR Attempt
This state indicates that either no Hellos or Hellos with mismatch information have been received recently from the neighboring client/server. In this state, attempts are made to contact the neighboring client/server by periodically sending Hello packets to the neighboring client/server, with period HelloInterval.

PPAR 1-Way
In this state, Hellos have recently been received from the neighboring client/server, but the remote ATM End-system address in the recently received Hello packet was set to zero.

PPAR 2-Way
In this state, Hellos have recently been received from the neighboring client/server, in which the remote ATM End-system address contains this client/server's ATM End-system address. When this state is reached, it indicates that bi-directional communication over this link between the client and server has been achieved. Registration/Query packets can only be transmitted over links that are in the 2-Way state.

5.2.4 Proxy PAR Hello FSM Events

State changes can be brought about by several possible events associated with operation of the Hello protocol. These events are shown as the labeled arcs in Figure Figure 5-1. A detailed explanation of the state changes and actions taken after an event occurs is given in Section 5.2.6.

PPAR LinkUp
Lower layer protocols have indicated that the link is operational.

PPAR 1-WayReceived
A Hello has been received from the neighboring client/server, in which the remote ATM End-system address is set to zero. Additionally, if the Protocol Version and Remote ATM End-system address fields in the hello data structure are not equal to zero, they must match, respectively, the Protocol Version and ATM End-system address from the received Hello.

PPAR 2-WayReceived
A Hello has been received from the neighboring client/server, in which the remote ATM End-system address correctly identifies this client/server's ATM End-system address. Additionally, if the Protocol Version and Remote ATM End-system address fields in the hello data structure are not equal to zero, they must match, respectively, the Protocol Version and ATM End-system address from the received Hello.

PPAR HelloMismatchReceived

A Hello has been received from the neighboring client/server in which at least one of the Protocol Version and ATM End-system address is different from the Protocol Version or remote ATM end-system address, respectively, in the hello data structure.

Alternatively, a Hello has been received in which the remote ATM end-system address is different from this client/server's own ATM End-system address, and is not set to zero.

In the state Attempt, only the second alternative is considered. The HelloMismatchReceived event takes precedence over the other events.

PPAR HelloTimerExpired
The Hello Timer has expired.

PPAR InactivityTimerExpired
The Inactivity Timer has expired. This means that no Hellos have been received recently from the neighboring client/server.

PPAR LinkDown
Lower layer protocols indicate that this link is not usable.

5.2.5 Receiving Hellos

This section describes the processing of Proxy PAR client or server Hello packets received. Before Proxy PAR processing is performed, version negotiation and checking as defined by the PNNI protocol specification are performed. Then the type is checked and, if it does not match the one expected, the packet is disregarded. In addition, the client has to retrieve the Registration Expiration Interval. The Hello Interval from the most recently received Hello packet is used to compute the initial value of the Inactivity Timer.

The remainder of the Hello packet is examined, whereby events are generated that are given to the appropriate Hello state machine.

The detailed processing of a received Hello packet depends on the state of the FSM. If the state is *PPAR Down*, the packet must be ignored. Otherwise, do the following:

```
if the Protocol Version and Remote Node ATM Address in the Hello data structure are equal to zero
   or
   the Protocol Version and Remote Node ATM Address in the Hello data structure match,
   respectively, the Protocol Version and ATM End System Address from the received Hello packet
   then
{
        if the Remote Node ATM End System Address from the received Hello packet is all 0 then
        {
                Hello FSM is executed with the event PPAR 1-Way Received;
        }
        else if the Remote Node ATM End System Address from the received Hello packet field
                is set to the ATM end system address of the receiving entity then
        {
                Hello FSM is executed with the event PPAR 2-Way Received;
        }
        else
        {
                Hello FSM is executed with the event PPAR Hello Mismatch;
        }
}
else
{
        Hello FSM is executed with the event PPAR Hello Mismatch;
}
```

5.2.6 Description of the Proxy PAR Hello FSM

Table 5-1: PPAR Hello FSM

	PPAR Down	PPAR Attempt	PPAR 1-Way	PPAR 2-Way
PPAR Link Up	PPHp1 PPAR Attempt	PPHp0 PPAR Attempt	PPHp0 PPAR 1-Way	PPHp0 PPAR 2-Way
PPAR 1-Way Received	FSM_ERROR	PPHp2 PPAR 1-Way	PPHp5 PPAR 1-Way	PPHp6 PPAR 1-Way
PPAR 2-Way Received	FSM_ERROR	PPHp3 PPAR 2-Way	PPHp4 PPAR 2-Way	PPHp5 PPAR 2-Way
PPAR Hello Mismatch	FSM_ERROR	PPHp0 PPAR Attempt	PPHp8 PPAR Attempt	PPHp7 PPAR Attempt
PPAR Hello Timer Expired	FSM_ERROR	PPHp10 PPAR Attempt	PPHp10 PPAR 1-Way	PPHp10 PPAR 2-Way
PPAR Inactivity Timer Expired	FSM_ERROR	FSM_ERROR	PPHp8 PPAR Attempt	PPHp7 PPAR Attempt
PPAR Link Down	PPHp0 PPAR Down	PPHp9 PPAR Down	PPHp9 PPAR Down	PPHp11 PPAR Down

FSM_ERROR Represents an internal implementation error.

PPHp0

Action: Do nothing.

PPHp1

Action: A client must send a Hello over the link and start the Hello Timer, enabling the periodic sending of Hellos. A server may take no action.

PPHp2

Action: Start the Inactivity Timer for the link. Set the *Remote Node ATM Address* in the Hello data structure to the end system address listed in the received Hello. Calculate the lower of the received newest version supported and the local newest version supported. Record this as the *Version Number*. Send a Hello to the neighbor and restart the Hello Timer.

PPHp3

Action: Start the Inactivity Timer for the link. Set the *Remote Node ATM Address* in the Hello data structure to the address listed in the received Hello. Calculate the lower of the received newest version supported and the local newest version supported. Record this as the *Version Number*. Send a Hello to the neighbor and restart the Hello Timer. Invoke the registration and query state machines with the event *PPAR Adjacency Up*.

PPHp4

Action: Restart the Inactivity Timer. Invoke the registration and query state machines with the event *PPAR Adjacency Up*.

PPHp5

Action: Restart the Inactivity Timer for the link because a Hello has been received from the neighbor.

PPHp6

Action: Restart the Inactivity Timer. Send a Hello to the neighbor and restart the Hello Timer. Invoke the registration and query state machines with the event *PPAR Adjacency Down*.

PPHp7

Action: The Inactivity Timer is disabled and the *Version*, *Remote Node ATM Address*, in the Hello data structure are cleared. Send a Hello to the neighbor and restart the Hello Timer. Invoke the registration and query state machines with the event *PPAR Adjacency Down*.

PPHp8

Action: The Inactivity Timer is disabled and the *version, Remote Node ATM Address,* in the Hello data structure are cleared. Send a Hello to the neighbor and restart the Hello Timer.

PPHp9

Action: The Hello and Inactivity timers are disabled and the *Version*, *Remote Node ATM Address,* fields in the Hello data structure are cleared.

PPHp10

Action: Send a Hello to the neighbor and restart the Hello Timer.

PPHp11

Action: The Hello and Inactivity timers are disabled and the *Version*, *Remote Node ATM Address* fields, in the Hello data structure are cleared. Invoke the registration and query state machines with the event *PPAR Adjacency Down*.

5.3 Proxy PAR Registration/Query Protocol

The registration and query protocols, respectively, enable the client to announce and learn which protocols are supported. All query/register operations are initiated by the client. The server never tries to push any information to the client. It is the client's responsibility to register and refresh the set of protocols supported and re-register them when changes occur. In the same sense, the client must query the information from the server at appropriate time intervals if it wishes to obtain the latest information. It is important to note that neither client nor server is supposed to keep any state information about the information stored by the other side. Both protocols use sequence numbers to control the flow of messages.

The protocol does not provide graceful recovery or explicit diagnostics, particularly during registration and query. Its typical behavior in the case of error is either to send a packet with initialize bit set or to restart the Hello FSM. Another way is to send a packet with a sequence number of zero.

It is important to note that every new registration session unconditionally discards any information registered previously between the same pair of AESAs that belong to the client and server. Within a query or registration session multiple different AESAs can be used. In case the same information appears more than once within the same session the first occurrence must be taken.

5.3.1 Proxy PAR Registration Protocol

The registration protocol enables a client to register the protocols and services it supports. All protocols are associated with a specific AESA and scope in the PNNI hierarchy. The client uses an abstract representation of the PNNI routing level called membership scope. The definition of membership scope is given in UNI signaling 4.0 Annex A5.3. As the default scope, implementations must choose the local scope in the absence of manual configuration. The server side of Proxy PAR maps the membership scopes to the correct PNNI routing level. The server is responsible for the initiation of flooding of the information within the peer group. Flooding will proceed throughout the hierarchy within the bounds of the specified scope.

The registration protocol is aligned as far as possible with the standard initial topology database exchange protocol used in link-state protocols and hence uses a window size of one. One information element is registered at a time and must be acknowledged before a new registration can be sent. The protocol uses "initialization" and "more" bits in the usual manner. The detailed packet format can be found in section 6.2.1 for the registration, and in section 6.2.2 for the registration acknowledgment. Any registration on a link unconditionally overwrites all registration data previously received on the same link.

5.3.1.1 Proxy PAR Registration Protocol Data Structure

There is a single PPAR Registration Protocol data structure for each of this client/server's physical ports running Proxy PAR protocol and for each logical port (each Virtual Path Connection for which this client/server is an endpoint). Each PPAR Registration protocol data structure consists of the following items:

State

The state of the PPAR Registration protocol FSM. This is described in more detail for the Server Side and Client Side in Section 5.3.1.2.1 and 5.3.1.3.1, respectively.

Remote ATM End-system address

The ATM address used to identify the neighboring peer client/server. As there is a one-to-one mapping between the PPAR Registration protocol FSM and Hello protocol FSM (refer section 5.2), one could instead use the Remote ATM End-system address in PPAR Hello protocol data structure.

Port ID
A number assigned by the client/server that identifies which physical port and which virtual path connection, if any, is described by this PPAR Registration protocol data structure. As there is a one-to-one mapping between the PPAR Registration protocol FSM and PPAR Hello protocol FSM (refer section 5.2), one could instead use the Port ID in PPAR Hello protocol data structure.

Sequence Number
An unsigned 32-bit number identifying individual PPAR Registration packets. Server's sequence number is initialized to the sequence number value in the very first PPAR Service Registration packet received by the server from the neighboring client. On the client side, the initial sequence number should be defined when the registration session is started, taking into account the following constraints:

- The protocols provide no support for sequence number wrap.
- A sequence number of zero is a special value used to support error recovery during the protocol and should not be used otherwise.
- The sequence number upon initiating the PAR Service Registration Client Side FSM must be randomized.

RegisterInterval
Each unacknowledged PPAR Service Registration packet is retransmitted every RegisterInterval seconds. This is applicable only to the PPAR client.

Register Timer
An interval timer that fires every RegisterInterval seconds. Whenever the timer fires, the client retransmits the last PPAR Service Registration packet over the corresponding link. This timer is stopped on receiving the PPAR Service Registration Acknowledgment packet from server in response to the last PPAR Service Registration packet sent by the client. This timer is applicable only to the PPAR client.

Registration Expiration Interval
The interval at which the server will purge the information registered by a client if it does not get refreshed. This is the life time the server assigns to the registered information. The server indicates the interval to the client in the Hello packets.

Registration Expiration Timer
An interval timer that fires every RegistrationExpirationInterval seconds. Whenever the timer fires, the server purges all information associated with this client. The timer is started (re-started) when the client registers information

5.3.1.2 Proxy PAR Registration Protocol Server Side

As soon as the server reaches a *PPAR 2-Way* state in the Hello protocol, it enters the *state PPAR Server Registration Wait* in which it can receive registration messages. Every registration packet received by the server has to be acknowledged by the server with the same sequence number.

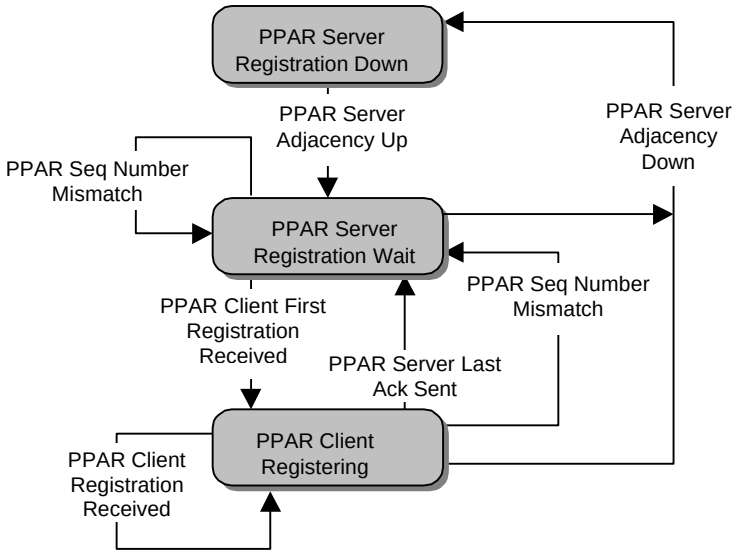

Figure 5-2: Proxy PAR Server Side of Registration Protocol

5.3.1.2.1 Proxy PAR Registration Protocol States

The states that PPAR Registration protocol FSM on the server side may attain are described in this section. Figure 5-2 shows a diagram of the possible state changes. The arcs are labeled with the events that cause each state change. These events are described in Section 5.3.1.2.2. For a detailed description of the state changes and the actions involved with each state change, see Section 5.3.1.2.4.

PPAR Server Registration Down

The initial state of a Registration Protocol FSM on the server side. This state indicates that the corresponding link to the neighboring client is not active (i.e., not in PPAR 2-Way state).

PPAR Server Registration Wait

This state indicates that the corresponding link to the neighboring client is active (i.e., in PPAR 2-Way state). In this state the server waits for a PPAR Service Registration packet from the neighboring client.

PPAR Client Registering

This state indicates that the neighboring client has started the registration process on the corresponding link. The server enters this state after receiving the first PPAR Service Registration packet from neighboring client. If the first PPAR Service Registration packet from the neighboring client had the "more" bit set to 0, then the Server immediately moves back to PPAR Server Registration Wait state after sending a PPAR Service Registration Ack packet. Otherwise, the server remains in this state till it receives a PPAR Service Registration packet from the neighboring client with the "more" bit set to 0.

5.3.1.2.2 Proxy PAR Registration Protocol Events

State changes can be brought about by several possible events associated with operation of the PPAR Registration protocol. These events are shown as the labeled arcs in Figure 5-2. A detailed explanation of the state changes and actions taken after an event occurs is given in Section 5.3.1.2.4.

PPAR Server Adjacency Up

This event is generated whenever the PPAR Hello protocol FSM of the corresponding link enters PPAR 2-Way state (i.e., link becomes active).

PPAR Client First Registration Received

This event is generated whenever the server receives a PPAR Service Registration packet with "initialize" bit set to 1.

PPAR Client Registration Received
This event is generated in PPAR Client Registering state whenever the server receives a PPAR Service Registration packet with "initialize" bit set to 0.

PPAR Server Last Ack Sent
This event is generated after the server responds with PPAR Service Registration Acknowledgment packet to a PPAR Service Registration packet, in which "more" bit was set to 0.

PPAR Server Adjacency Down
This event is generated whenever the PPAR Hello protocol FSM of the corresponding link goes out of PPAR 2-Way state (i.e., link becomes inactive).

PPAR Seq Number Mismatch
This event is generated whenever the "initialize" bit in the received PPAR Service Registration packet is not set to 1 and the sequence number in it is:

- not equal to the sequence number in the PPAR Registration protocol data structure, and
- not one greater than that of the sequence number in the PPAR Registration protocol data structure.

5.3.1.2.3 Receiving Proxy PAR Service Registration Packets

This section explains the detailed processing of a received service registration packet. Processing depends on the state of the FSM. If the state is *PPAR Server Registration Down* the packet should be ignored. Otherwise, if the state is:

PPAR Server Registration Wait:

```
if the Init bit is set then
{
    Server Registration FSM is executed with the event PPAR Client First Registration
    Received;
    if the More bit is not set then
    {
        the Server Registration FSM is executed with the event PPAR Server Last Ack Sent;
    }
}
else if the sequence number is the same as that of the last registration packet received
    then
{
    /* the registration packet is a duplicate */
    send an acknowledgment with that sequence number and discard the duplicate packet;
}
else
{
    Server Registration FSM is executed with PPAR Seq Number Mismatch;
}
```

PPAR Client Registering:

```
If the Init bit is set then
{
    the Server Registration FSM is executed with the event PPAR Client First Registration
    Received;
    if the More bit is not set then
    {
        the Server Registration FSM is executed with the event PPAR Server Last Ack Sent;
    }
}
else if the sequence number is one higher than the last received sequence number then
{
    the Server Registration FSM is executed with the event PPAR Client Registration
    Received;
```

```
    if the More bit is not set then
    {
        the Server Registration FSM is executed with the event PPAR Server Last Ack Sent;
    }
}
else if the sequence number is the same as that of the last registration packet received
    then
{
    /* the registration packet is a duplicate */
    send an acknowledgment with that sequence number and discard the duplicate packet;
}
else
{
    Server Registration FSM is executed with PPAR Seq Number Mismatch;
}
```

5.3.1.2.4 Description of Proxy PAR Registration FSM

Table 5-2: PPAR Service Registration FSM (Server Side)

	PPAR Server Registration Down	PPAR Server Registration Wait	PPAR Client Registering
PPAR Server Adjacency Up	SRp1 PPAR Server Registration Wait	FSM_ERROR	FSM_ERROR
PPAR Client First RegistrationReceived	FSM_ERROR	SRp2 PPAR Client Registering	SRp2 PPAR Client Registering
PPAR Client Registration Received	FSM_ERROR	FSM_ERROR	SRp3 PPAR Client Registering
PPAR Server Last Ack Sent	FSM_ERROR	FSM_ERROR	SRp4 PPAR Server Registration Wait
PPAR Server Adjacency Down	FSM_ERROR	SRp5 PPAR Server Registration Down	SRp5 PPAR Server RegistrationDown
PPAR Seq Number Mismatch	FSM_ERROR	SRp6 PPAR Server Registration Wait	SRp6 PPAR Server Registration Wait

SRp1

Action: Clear the stored sequence number.

SRp2

Action: Reset any information pertaining to prior registration sequences. Save the sequence number of the incoming registration. Save the data in the registration. Send an acknowledgment containing the sequence number of the current request.

SRp3

Action: Save the sequence number of the request. Save the data and send an acknowledgment containing the sequence number on the current request.

SRp4

Action: Advertise the services specified in the requests by flooding the service data in PTSEs with the appropriate scope.

SRp5

Action: Flush all PTSEs describing the registered services.

SRp6

Action: Clear stored sequence number. Send an acknowledgment with special sequence number 0.

5.3.1.3 Proxy PAR Registration Protocol Client Side

When the client detects PPAR 2-Way adjacency it can start to register its services. Before a client is allowed to send a registration packet other than the first one, it must wait for an acknowledgment for the previous one. In the case that the acknowledgment is lost, the client must continue to resend the packet until it is confirmed. The client manages the sequence of registrations by means of an initialize bit as well as a more bit. This permits the server to keep track of the registration process. A new series of registration messages must always be initialized by a packet with the initialize bit set. When the client has to split the registration across multiple packets it indicates this by setting the more bit in all but the last registration sent. With the help of the sequence number the completeness of a registration session can be enforced.

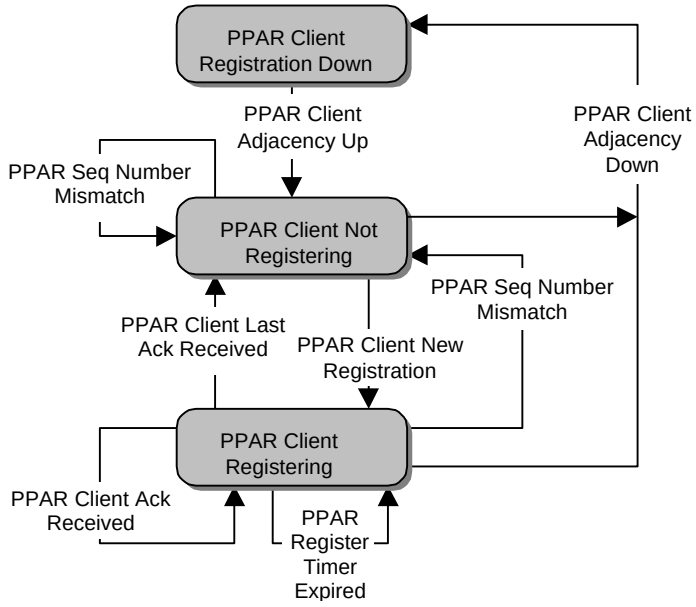

Figure 5-3: Proxy PAR Client Side of Registration Protocol

5.3.1.3.1 Proxy PAR Registration Protocol States

The states that, PPAR Registration protocol FSM on PPAR Client side may attain are described in this section. Figure 5-3 shows a diagram of the possible state changes. The arcs are labeled with the events that cause each state change. These events are described in Section 5.3.1.3.2. For a detailed description of the state changes and the actions involved with each state change, see Section 5.3.1.3.4.

PPAR Client Registration Down

The initial state of a Registration Protocol FSM on client side. This state indicates that the corresponding link to the neighboring server is not active (i.e., not in PPAR 2-Way state).

PPAR Client Not Registering

This state indicates that the corresponding link to the neighboring server is active (i.e., in Hello 2-Way state). In this state the client can send PPAR Service Registration packet to the neighboring server to register the services supported by it.

PPAR Client Registering

This state indicates that the client has started the registration process on the corresponding link. The Client enters this state after sending PPAR Service Registration packet to the neighboring server. The

Client remains in this state till it receives a PPAR Service Registration Ack packet in response to a PPAR Service Registration packet sent by it, with the "more" bit set to 0.

5.3.1.3.2 Proxy PAR Registration Protocol Events

State changes can be brought about by several possible events associated with operation of the PPAR Registration protocol. These events are shown as the labeled arcs in Figure 5-3. A detailed explanation of the state changes and actions taken after an event occurs is given in Section 5.3.1.3.4.

PPAR Client Adjacency Up
This event is generated whenever the PPAR Hello protocol FSM of the corresponding link enters PPAR 2-Way state (i.e., link becomes active).

PPAR Client New Registration
This event is generated in PPAR Client Not Registering state, whenever client sends a PPAR Service Registration packet, to register services supported by it with the server.

PPAR Client Ack Received
This event is generated in PPAR Client Registering state, whenever the client receives a PPAR Service Registration Acknowledgment packet in response to a PPAR Service Registration packet sent by it with the "more" bit set to 1.

PPAR Client Last Ack Received
This event is generated in PPAR Client Registering state, whenever the client receives a PPAR Service Registration Acknowledgment packet in response to a PPAR Service Registration packet sent by it with the "more" bit set to 0.

PPAR Register Timer Expired
Generated on the expiry of Register Timer.

PPAR Client Adjacency Down
This event is generated whenever the PPAR Hello protocol FSM of the corresponding link goes out of PPAR 2-Way state (i.e., link becomes inactive).

PPAR Seq Number Mismatch
This event is generated whenever the sequence number in the received PPAR Service Registration Acknowledgment packet is:

- not equal to the sequence number in the PPAR Registration protocol data structure, and
- not one less than that of the sequence number in the PPAR Registration protocol data structure

5.3.1.3.3 Receiving Proxy PAR Service Registration Acknowledgment Packets

This section explains the detailed processing of a received service registration acknowledgment packet. If the state of the FSM is *PPAR Client Registration Down or PPAR Client Not Registering* the packet should be ignored. Otherwise, if the state is:

PPAR Client Registering:

```
If the sequence number in the Service Registration Acknowledgment packet is the same as
    the sequence number in the PPAR Registration Protocol data structure then
{
    if the More bit in the last Registration Packet was not set then
    {
        the Client Registration FSM is executed with the event PPAR Client Last Ack
    Received;
    }
    else
    {
        the Client Registration FSM is executed with the event PPAR Client Ack Received;
```

```
    }
}
else if the sequence number in the Service Registration Acknowledgment packet is one less
    than the sequence number in the PPAR Registration Protocol data structure then
{
    /* the Ack packet is a duplicate */
    discard the duplicate packet;
}
else
{
    the Client Registration FSM is executed with the event PPAR Seq Number Mismatch;
}
```

5.3.1.3.4 Description of Proxy PAR Registration Protocol FSM

	PPAR Client Registration Down	PPAR Client Not Registering	PPAR Client Registering
PPAR Client Adjacency Up	CRp1 PPAR Client Not Registering	FSM_ERROR	FSM_ERROR
PPAR Client New Registration	FSM_ERROR	CRp2 PPAR Client Registering	FSM_ERROR
PPAR Client Ack Received	FSM_ERROR	FSM_ERROR	CRp3 PPAR Client Registering
PPAR Client Last Ack Received	FSM_ERROR	FSM_ERROR	CRp4 PPAR Client Not Registering
PPAR Register Timer Expired	FSM_ERROR	FSM_ERROR	CRp5 PPAR Client Registering
PPAR Client Adjacency Down	FSM_ERROR	CRp6 PPAR Client Registration Down	CRp4 PPAR Client Registration Down
PPAR Seq Number Mismatch	FSM_ERROR	CRp6 PPAR Client Not Registering	CRp4 PPAR Client Not Registering

Table 5-3: PPAR Service Registration FSM (Client Side)

CRp1

Action: Initialize the registration sequence number.

CRp2

Action: Increment the registration sequence number. Send an initial registration request with the initialize bit set and the more bit set if the data is not all contained in the first registration request. Start the Registration Timer.

CRp3

Action: Increment the registration sequence number. Send the next request with the initialize bit not set and the more bit set if the remaining data is not all contained in the current registration request, otherwise cleared. Start the Registration Timer.

CRp4

Action: Disable the Registration Timer.

CRp5

Action: Resend the last Registration Request. Start the Registration Timer.

CRp6

Action: Do nothing.

5.3.2 Proxy PAR Query Protocol

The client uses the query protocol to obtain information about services registered by other clients. The client requests services registered within a specific scope an optional VPN ID, and address. It is always the client's task to request information, the server never makes an attempt to push information to the client. If the client needs to filter the returned data based on service specific information, such as BGP AS, it must parse and interpret the registration information groups. The server never looks beyond the PAR IPv4 Service Definition IG.

5.3.2.1 Proxy PAR Query Protocol Data Structure

There is a single PPAR Query Protocol data structure for each of this client/server's physical ports running Proxy PAR protocol and for each logical port (each Virtual Path Connection for which this client/server is an endpoint). Each PPAR Query protocol data structure consists of the following items:

State
: The state of the PPAR Query protocol FSM. This is described in more detail for Server side and Client side in Section 5.3.2.2.1 and 5.3.2.3.1, respectively.

Remote ATM End-system address
: The node ID used to identify the neighboring peer client/server. As there is one to one mapping between the PPAR Query protocol FSM and Hello protocol FSM (refer section 5.2), one could instead use the Remote ATM End-system address in PPAR Hello protocol data structure.

Port ID
: A number assigned by the client/server that identifies which physical port and which virtual path connection, if any, is described by this PPAR Query protocol data structure. As there is one to one mapping between the PPAR Query protocol FSM and PPAR Hello protocol FSM (refer section 5.2), one could instead use the Port ID in PPAR Hello protocol data structure.

Sequence Number
: An unsigned 32-bit number identifying individual PPAR Service Request, Service Description and Service Description Acknowledgment packets. Server's sequence number is initialized to the sequence number value in the very first PPAR Service Request packet received by the server from neighboring client. On the client side, initial sequence number should be defined when query session is started, taking into account the following constraints:
  - The protocols provide no support for sequence number wrap.
  - A sequence number of zero is a special value used to support error recovery during the protocol and should not be used otherwise.
  - The sequence number upon initiating the PPAR Service Description Client Side FSM must be randomized.

ClientQueryInterval
: Each unacknowledged PPAR Service Request is retransmitted every ClientQueryInterval seconds. This is applicable only to the PPAR client.

Client Query Timer
: An interval timer that fires every ClientQueryInterval seconds. Whenever the timer fires, the client retransmits the last PPAR Service Request packet over the corresponding link. This timer is stopped on receiving the PPAR Service Description packet from server in response to the last PPAR Service Request packet sent by client. This timer is applicable only to the PPAR client.

ServerQueryInterval
: Each unacknowledged PPAR Service Description is retransmitted every ServerQueryInterval seconds. This is applicable only to the PPAR server.

Server Query Timer

An interval timer that fires every ServerQueryInterval seconds. Whenever the timer fires, the server retransmits the last PPAR Service Description packet over the corresponding link. This timer is stopped on receiving the PPAR Service Description Acknowledgment packet from client in response to the last PPAR Service Description packet sent by the server. This timer is applicable only to the PPAR server.

5.3.2.2 Proxy PAR Query Protocol Server Side

After the server has reached *PPAR 2-Way* adjacency it can respond to queries from the client. Upon reception of a query packet the server transmits all information associated within the specified scope to the client. As the information found can span multiple packets, a protocol similar to the registration protocol is executed across the connection.

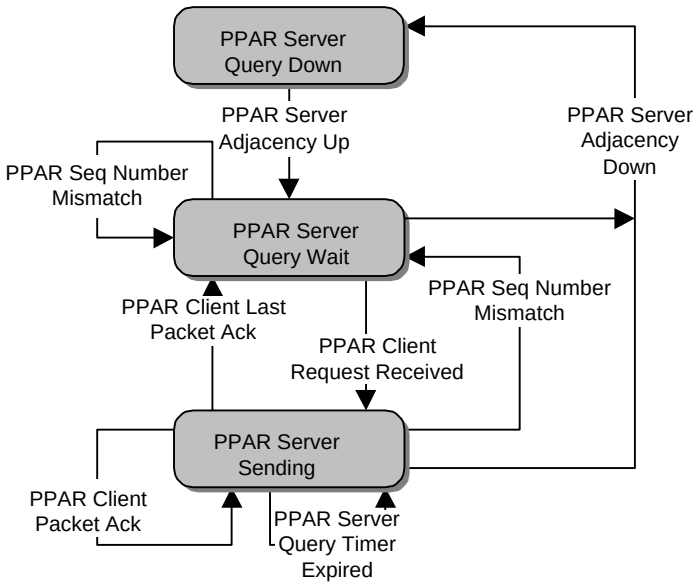

Figure 5-4: Proxy PAR Server Side of Query Protocol

5.3.2.2.1 Proxy PAR Query Protocol States

The states that PPAR Query protocol FSM on the server side may attain are described in this section. Figure 5-4 shows a diagram of the possible state changes. The arcs are labeled with the events that cause each state change. These events are described in Section 5.3.2.2.2. For a detailed description of the state changes and the actions involved with each state change, see Section 5.3.2.2.5.

PPAR Server Query Down

The initial state of a Query Protocol FSM on the server side. This state indicates that the corresponding link to the neighboring client is not active (i.e., not in PPAR 2-Way state).

PPAR Server Query Wait

This state indicates that the corresponding link to the neighboring client is active (i.e., in PPAR 2-Way state). In this state the server waits for PPAR Service Request packet from the neighboring client.

PPAR Server Sending

This state indicates that the neighboring client has started the query process on the corresponding link. The server enters this state after receiving the first PPAR Service Request packet from the neighboring client.

5.3.2.2.2 Proxy PAR Query Protocol Events

State changes can be brought about by several possible events associated with operation of the PPAR Query protocol. These events are shown as the labeled arcs in Figure 5-4. A detailed explanation of the state changes and actions taken after an event occurs is given in Section 5.3.2.2.5.

PPAR Server Adjacency Up
This event is generated whenever the PPAR Hello protocol FSM of the corresponding link enters PPAR 2-Way state.

PPAR Client Request Received
This event is generated whenever the server receives a PPAR Service Request packet.

PPAR Client Packet Ack
This event is generated in PPAR Server Sending state whenever the server receives a PPAR Service Description Acknowledgment packet in response to the last PPAR Service Description packet sent by the server with the "more" bit set to 1.

PPAR Client Last Packet Ack
This event is generated in PPAR Server Sending state whenever the server receives a PPAR Service Description Acknowledgment packet in response to last PPAR Service Description packet sent by server with the "more" bit set to 0.

PPAR Server Query Timer Expired
Generated on the expiry of the Server Query Timer.

PPAR Server Adjacency Down
This event is generated whenever the PPAR Hello protocol FSM of the corresponding link goes out of PPAR 2-Way state (i.e., link becomes inactive).

PPAR Seq Number Mismatch
This event is generated whenever the sequence number in the received Service Description Acknowledgment packet is:

- not equal to the sequence number in the PPAR Query protocol data structure, and
- not one less than that of the sequence number in the PPAR Query protocol data structure.

5.3.2.2.3 Receiving Proxy PAR Service Request Packets

This section explains the detailed processing of a received service request packet. If the state of the FSM is *PPAR Server Query Down*, the packet should be ignored. Otherwise, if the state is:

PPAR Server Query Wait:

```
the Server Query FSM is executed with the event PPAR Client Request Received;
```

PPAR Server Sending:

```
if the received sequence number in the Service Request packet is equal to the sequence
    number in the PPAR Query Protocol Data Structure then
{
    /* the Service Request packet is a duplicate */
    discard the packet;
}
else
{
    the Server Query FSM is executed with the event PPAR Client Request Received;
}
```

5.3.2.2.4 Receiving Proxy PAR Service Description Acknowledgment Packets

This section explains the detailed processing of a received service description acknowledgment packet. If the state of the FSM is *PPAR Server Query Down* or *PPAR Server Query Wait* the packet should be ignored. Otherwise, if the state is:

PPAR Server Sending:

```
If the sequence number in the received Ack packet is the same as that in the Query
    Protocol Data Structure then
{
    if the more bit was set in the last Service Description packet sent then
    {
        the Server Query FSM is executed with the event PPAR Client Packet Ack;
    }
    else
    {
        the Server Query FSM is executed with the event PPAR Client Last Packet Ack;
    }
}
else if the sequence number in the received Ack packet is one less than that in the Query
    Protocol Data Structure then
{
    /* the Description Ack packet is a duplicate */
    discard the packet;
}
else
{
    the Server Query FSM is executed with the event PPAR Seq Number Mismatch;
}
```

5.3.2.2.5 Description of Proxy PAR Query Protocol FSM

Table 5-4: PAR Service Description FSM (Server Side)

	PPAR Server Query Down	PPAR Server Query Wait	PPAR Server Sending
PPAR Server Adjacency Up	SQp1 PPAR Server Query Wait	FSM_ERROR	FSM_ERROR
PPAR Client Request Received	FSM_ERROR	SQp2 PPAR Server Sending	SQp2 PPAR Server Sending
PPAR Client Packet Ack	FSM_ERROR	FSM_ERROR	SQp3 PPAR Server Sending
PPAR Query Timer Expired	FSM_ERROR	FSM_ERROR	SQp4 PPAR Server Sending
PPAR Client Last Packet Ack	FSM_ERROR	FSM_ERROR	SQp5 PPAR Server Query Wait
PPAR Server Adjacency Down	FSM_ERROR	SQp7 PPAR Server Query Down	SQp5 PPAR Server Query Down
PPAR Seq Number Mismatch	FSM_ERROR	FSM_ERROR	SQp6 PPAR Server Query Wait

SQp1

Action: Initialize the query sequence number.

SQp2

Action: Store the received sequence number. Send the first description packet with the Init flag set, and also the More flag set if more data is to follow. Start the query timer.

SQp3

Action: Increment the sequence number. Send a further description packet with the Init flag not set, and with the More flag set if more data is to follow. Start the query timer.

SQp4

Action: Resend the last description packet and restart the query timer.

SQp5

Action: Disable the query timer.

SQp6

Action: Disable the query timer. Clear the sequence number. Send an empty description packet with the Init flag set, the More bit cleared, the sequence number and scope fields both set to zero.

SQp7

Action: Do nothing.

5.3.2.3 Proxy PAR Query Protocol Client Side

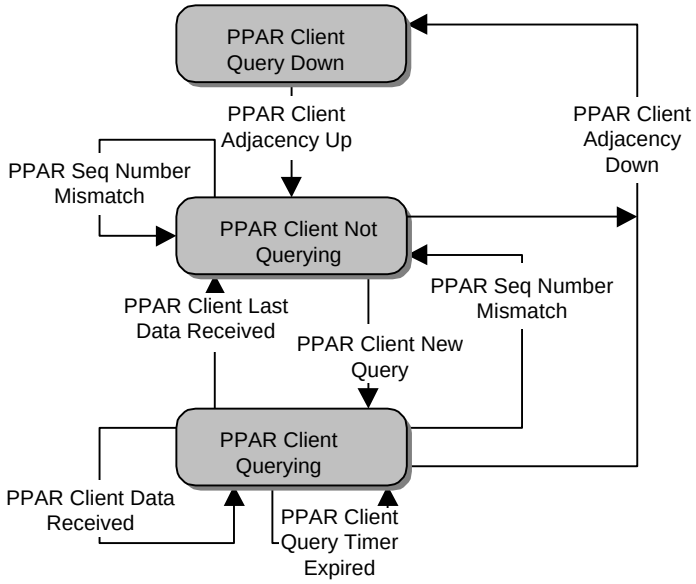

Figure 5-5: Proxy PAR Client Side of Query Protocol

Upon the establishment of *PPAR 2-Way* adjacency the client may query the server. It is up to the client to decide whether to register its services before it issues a query. A client may use an initial query to determine which protocols it has to register, but registration will typically precede the query.

The client queries the server for a specific scope, an optional VPN ID, IP prefix and set of services. In order to stay up-to-date the client should repeat the query periodically because the server does not automatically send information to the client. Failure to do so may result in degradation of services dependent on the received registration, but has no implications on the behavior of Proxy PAR. It should be possible to set the query interval to be relatively long, as the registration data is expected to be relatively stable over time.

5.3.2.3.1 Proxy PAR Query Protocol States

The states that the PPAR Query protocol FSM on the PPAR Client side may attain are described in this section. Figure 5-5 shows a diagram of the possible state changes. The arcs are labeled with the events that cause each state change. These events are described in Section 5.3.2.3.2. For a detailed description of the state changes and the actions involved with each state change, see Section 5.3.1.3.4.

PPAR Client Query Down

The initial state of a Query Protocol FSM on client side. This state indicates that the corresponding link to the neighboring server is not active (i.e., not in PPAR 2-Way state).

PPAR Client Not Querying
This state indicates that the corresponding link to the neighboring server is active (i.e., in PPAR 2-Way state). In this state the client can send a PPAR Service Request packet to the neighboring server to query service information.

PPAR Client Querying
This state indicates that the client has started the query process on the corresponding link. The client enters this state after sending a PPAR Service Request packet to neighboring server.

5.3.2.3.2 Proxy PAR Query Protocol Events

State changes can be brought about by several possible events associated with operation of the PPAR Query protocol. These events are shown as the labeled arcs in Figure 5-5. A detailed explanation of the state changes and actions taken after an event occurs is given in Section 5.3.2.3.4.

PPAR Client Adjacency Up
This event is generated whenever the PPAR Hello protocol FSM of the corresponding link enters PPAR 2-Way state.

PPAR Client New Query
This event is generated in PPAR Client Not Querying state, whenever the client sends a PPAR Service Request packet, to query service information from the server.

PPAR Client Data Received
This event is generated in PPAR Client Querying state, whenever the client receives a PPAR Service Description packet in response to a PPAR Service Request packet sent by it with the “more” bit set to 1.

PPAR Client Last Data Received
This event is generated in PPAR Client Querying state, whenever the client receives a PPAR Service Description packet in response to a PPAR Service Request packet sent by it with the “more” bit set to 0.

PPAR Client Query Timer Expired
Generated on the expiry of the Client Query Timer.

PPAR Client Adjacency Down
This event is generated whenever the PPAR Hello protocol FSM of the corresponding link goes out of PPAR 2-Way state (i.e., link becomes inactive).

PPAR Seq Number Mismatch
This event is generated in PPAR Client Querying state, whenever in the received PPAR Service Description packet, the “initialize” bit is set to 1 and the sequence number is:

- not equal to the sequence number in the Service Request packet.

or the initialize bit is set to 0 and the sequence number is:

- not equal to the sequence number of the previous Service Description packet, and
- not one greater than the sequence number of the previous Service Description packet.

5.3.2.3.3 Receiving Proxy PAR Service Description Packets

This section explains the detailed processing of a received service description packet. If the state of the FSM is *PPAR Client Query Down*, the packet should be ignored. Otherwise, do the following depending on the state:

PPAR Client Not Querying:

```
if the sequence number in the Service Description packet received is the same as the
    sequence number of the previous Service Description packet received then
{
    /* the Service Description packet is a duplicate */
    send a Service Description Ack packet with a sequence number of the packet received
    and discard the packet;
}
else
{
    send a Service Description Ack packet with a sequence number of 0 and discard the
    packet;
}
```

PPAR Client Querying:

```
if the init bit is set then
{
    if Sequence Number in the Service Description packet is the same as the one in the
    Service Request packet then
    {
        if the received Service Description Packet is the very first packet received after
    sending the  Service Request packet then
        {
               if the more bit is set then
               {
                 the Client Query FSM is executed with the event PPAR Client Data
                 Received;
               }
               else
               {
                 the Client Query FSM is executed with the event PPAR Client Last Data
                 Received;
               }
        }
        else
        {
             /* the Service Description packet is a duplicate */
             send a Description Ack packet with that sequence number and the packet is
             discarded;
        }
    }
    else
    {
        the Client Query FSM is executed with the event PPAR Client Seq Number Mismatch;
    }
}
else
{
    if the Sequence Number in the Service Description packet is one more than the sequence
    number of the previous Service Description packet then
    {
        if the more bit is set then
               {
                  the Client Query FSM is executed with the event PPAR Client Data
                  Received;
               }
               else
               {
                 the Client Query FSM is executed with the event PPAR Client Last Data
                 Received;
               }
    }
    else if Sequence Number in the Service Description packet is the same as the one in
    the previous Service Description packet then
    {
            /* the Service Description packet is a duplicate */
            send a Description Ack packet with that sequence number and the packet is
            discarded;
    }
    else
    {
```

```
            the Client Query FSM is executed with the event PPAR Client Seq Number Mismatch;
        }
    }
```

5.3.2.3.4 Description of Proxy PAR Query Protocol FSM

Table 5-5: PPAR Service Description FSM (Client Side)

	PPAR Client Query Down	PPAR Client Not Querying	PPAR Client Querying
PPAR Client Adjacency Up	CQp1 PPAR Client Not Querying	FSM_ERROR	FSM_ERROR
PPAR Client New Query	FSM_ERROR	CQp2 PPAR Client Querying	FSM_ERROR
PPAR Query Timer Expired	FSM_ERROR	FSM_ERROR	CQp3 PPAR Client Querying
PPAR Client Data Received	FSM_ERROR	FSM_ERROR	CQp4 PPAR Querying
PPAR Client Last Data Received	FSM_ERROR	FSM_ERROR	CQp5 PPAR Not Querying
PPAR Client Adjacency Down	FSM_ERROR	CQp0 PPAR Client Query Down	CQp6 PPAR Client Query Down
PPAR Client Seq Number Mismatch	FSM_ERROR	FSM_ERROR	CQp7 PPAR Client Not Querying

CQp0:

Action: Do nothing.

CQp1:

Action: Initialize the Query Sequence number.

CQp2:

Action: Increment the Query Sequence Number by 1. Send the service request and start the query timer.

CQp3:

Action: Resend the service request and start the query timer.

CQp4:

Action: Send an acknowledgment for the current data and disable the query timer if it is running. Increment the Query Sequence Number.

CQp5:

Action: Send an acknowledgment for the current data and disable the query timer if it is running. The information concerning the registered services may be used.

CQp6:

Action: Clear all information received concerning the registered services and disable the query timer.

CQp7:

Action: Set the Query Sequence number to zero, and send an Ack for this sequence number. Clear all information received concerning the registered services and disable the query timer.

5.4 Example of a Client-Server Exchange

The multiplicity of state machines introduced above tend to mask the overall simplicity of the protocol. Hence a simple example is included here to show a typical exchange between Proxy PAR server and client. Figure 5-6:

Example of Proxy PAR, shows an initial Hello exchange, followed by registration and query sequences. In general, the protocol permits registration and query to proceed concurrently.

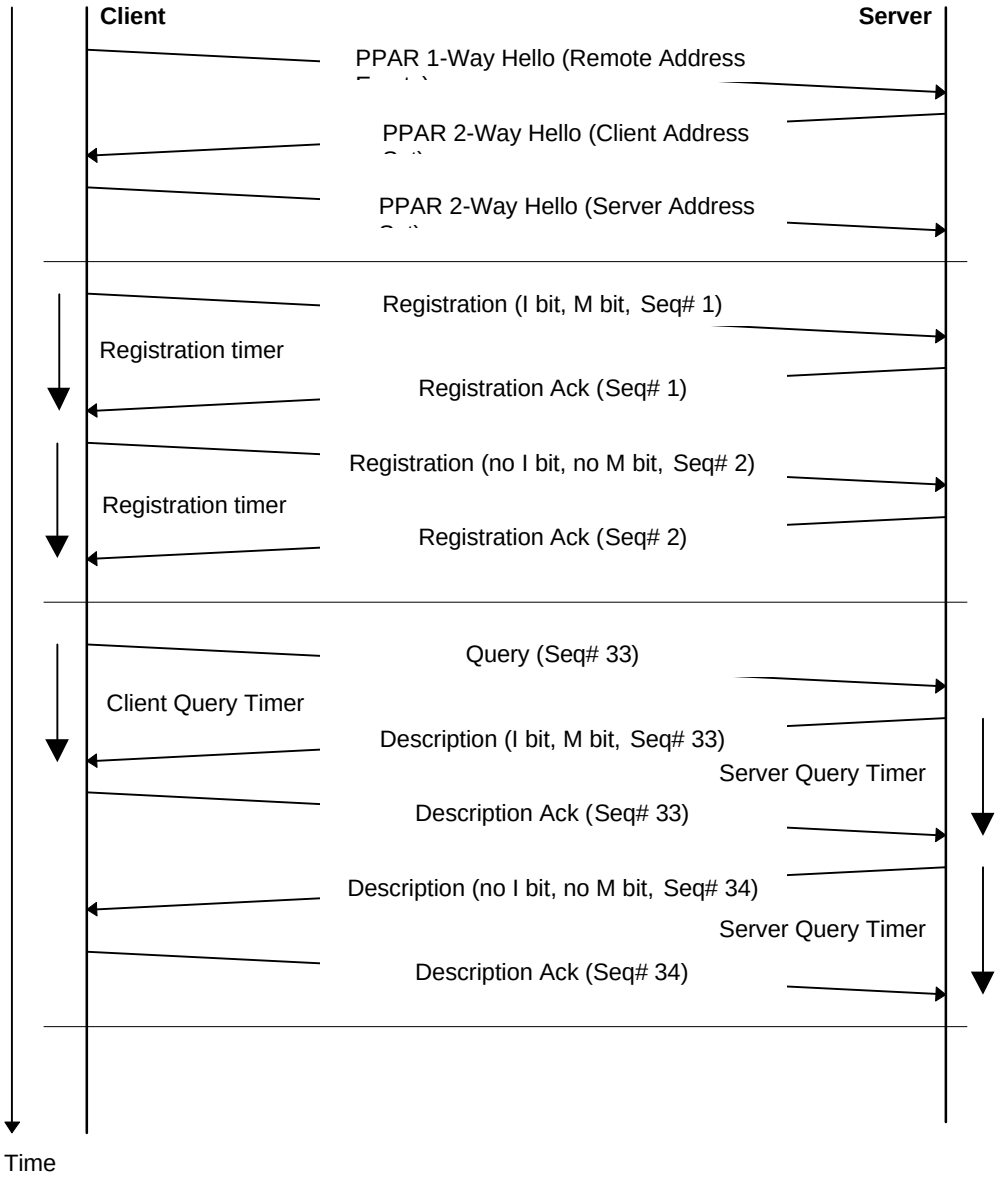

Figure 5-6: Example of Proxy PAR

6. Proxy PAR Specific Packet Formats

[Normative]

In order to simplify the implementation of the client side, the packet format for the communication between the client and server is more restrictive than a full PNNI suite. The ensuing lack of generality of packet formats may require separate formats for new types of services that will be introduced in the future. The PNNI packet header is taken from PNNI as specified in table 5-20 in [PNNI] with Proxy PAR version set to 1. In addition to the existing packet types of table 5-21 in [PNNI], the following list of packet types are necessary for Proxy PAR.

Table 6-1: Proxy PAR Packet Types

Packet Type	Packet Name
32	Proxy PAR Client Hello
33	Proxy PAR Server Hello
34	Proxy PAR Service Registration
35	Proxy PAR Service Registration Acknowledgment
36	Proxy PAR Service Request
37	Proxy PAR Service Description
38	Proxy PAR Service Description Acknowledgment

6.1 Hello Protocol

Packet formats follow the Hello protocol part of the PNNI specification [PNNI] as far as possible with necessary deviations for the server-client nature of the protocol and additional capabilities added. Instead of node identifiers, ATM addresses are used to verify that two-way communication has been established. Server and client indicate their respective functions in the Hello packet. In case of non-agreement on the type of functionality assumed on the remote side, Hello packets are dropped.

6.1.1 Client and Server Side Hello

Table 6-2: Proxy PAR Hello Message

Offset	Size (Octets)	Name	Function/Description
0	8	PNNI Header	A PNNI packet header structure with Packet type = 32 or 33 (client or server side Hello). See Table 5-20 in [PNNI].
8	2	Flags	reserved, must be 0.
10	20	ATM End System Address	A system that implements Proxy PAR must have a unique ATM End System Address.
30	20	Remote Node ATM End System Address	set based on value of remote node's address in received Hellos on same link, set to all zeros if not yet known.
50	2	Hello Interval	Hello Interval indicates how frequently the Hello packet must be received. If no Hello is received within a time-out period of (Inactivity Factor times Hello Interval) then the link is assumed to have failed.
52	2	Registration Expiration Interval	Registration Expiration Interval indicates the life time the server assigns to the registered information. The server sets the value to the interval used. In Hello packets coming from the client the value have to be set to 0. It is the client's duty to refresh its registered information in an appropriate time interval. A typical value for a refresh interval used by a client

			would be half of the registration expiration interval.
54	2	Reserved	

6.2 Service Registration

Service registration is a protocol initiated by the client that indicates which services PAR should distribute across the ATM cloud for this UNI. Client registration encompasses a simple protocol very similar to the initial topology database exchange including a sequence number to detect connectivity losses. The server acknowledges the registration packets. When a registration has to be changed, the protocol must be re-executed and the server must delete all client registration information received so far. The server always has to receive a complete new set of information from this client.

6.2.1 Client Side IPv4 Service Registration

Service registration packet encompasses any number of PAR IPv4 Service Definition IGs with the same scope. In addition, System Capabilities IG may be included.

Table 6-3: Service Registration Packet

Offset	Size (Octets)	Name	Function/Description
0	8	PNNI Header	A PNNI packet header structure with Packet type = 34 (Proxy PAR Service Registration).
8	4	Registration packet sequence number	
12	2	Flags	See Table 6-4.
14	20	ATM Address	AESA for which the service is to be registered.
34	1	Scope	Membership scope
35	1	Reserved	
Any number of PAR IPv4 Service Definition IGs optionally nested withing PAR VPN ID Igs, specifying the registered information with the fields having distinct meanings: • IP address specifies the host address at which the service is available. • IP address mask specified is used to indicate the subnet on which the IP address resides. It must be contiguous, starting at the MSB and have at least one bit set. • The service mask specifies a request for registrations for all of the services specified by the bits set. For each of the registered services, inside of the PAR IPv4 Service Definition IG an appropriate service specific IG can be included, e.g. if the BGP service bit is set, the PAR IPv4 Service Definition IG can include a PAR BGP4 IPv4 Service Definition IG with the appropriate AS. Any number of System Capabilities IGs			

Table 6-4: Service Registration Packet Flags

Bit ID:	bit 16 (MSB)	bit 15	bits 14..1 (LSB)
Bit Name:	'Initialize' (I) bit	'More' (M) bit	Reserved
Description:	Set to one during initialization of the registration process. Otherwise set to zero.	Set to one if the client has additional registration packets to transmit, or to zero if this is the last registration packets to be transmitted.	

6.2.2 Server Side Registration Acknowledgment

The server side of the protocol receives the registration packets and must confirm them with the sequence numbers received. In addition, the server indicates to the client whether the registration was successful via the included return code.

Table 6-5: Service Registration Acknowledgment Packet

Offset	Size (Octets)	Name	Function/Description
0	8	PNNI Header	A PNNI packet header structure with Packet type = 35 (Proxy PAR Registration Acknowledgment).
8	4	Registration packet sequence number	
12	1	Return code	Return code from the server to the client with the meaning: Value — Function 0 — Successful registration. 1 — Registered information not accepted 2 — Database Overflow 3 — Invalid VPN ID 4 — Invalid IPv4 IG 5 — Invalid Scope 6 — Invalid ATM Address 7-255 — Reserved
13	3	Reserved	

6.3 Service Request Protocol

The Service Request Protocol is similar to the registration protocol in terms of state machines. However, the packet contents are different. It is important to note that both protocol flows are always initiated by the client but the data transfer in the registration sequence is from the client to the server, whereas in the request case the packets flow from the server to the client.

6.3.1 Client Side IPv4 Service Request

This packet is issued by the client every time it initiates a query for a description of specific registered services. It includes query restrictions (scope) and an initial sequence number that the server must use for the first packet of its answer.

Table 6-6: Service Request Packet

Offset	Size (Octets)	Name	Function/Description
0	8	PNNI Header	A PNNI packet header structure with Packet type = 36 (Proxy PAR Service Request Packet).
08	4	Request sequence number	Initial sequence number for the server. The server starts the response packets with this number and increases it with every service description packet sent.
12	1	Scope	Membership scope of the request. Any service description returned must have a scope smaller than or equal to this scope. After translating the membership scope to the PNNI routing level, the server must look at the level of the node that created the PTSE in order to do correct filtering. The field to inspect is the originating node ID (first byte) in the PTSP header.
13	3	Reserved	
Zero or more PAR IPv4 Service Definition IGs, together with zero or more PAR VPN ID IGs each containing zero or more PAR IPv4 Service Definition IGs, specifying the requested information with the			

fields having the following meaning:

- The server should return information belonging to the given VPN ID(s) if PAR VPN ID IGs are present.
- IP address 0.0.0.0 specifies any IP address.
- IP address mask specified is used to mask out the given IP address before matching registered IP addresses and prefixes. Any registered address with the same or longer registered mask and significant bits matching the bits given in the masked out IP address are considered to be matched by the request.
- An IP address mask of length 0 means a request for services with any IP address.
- The service mask specifies a request for registrations with any of the services specified by the bits set.
- No mechanisms are defined to request services with specific service data such as the area inside the of PAR OSPF IPv4 Service Definition IG.

The server must return all System Capabilities IGs at the top level of the PAR Service IGs with matching scope that have a matching IEEE OUI.

It is the server's duty to perform the appropriate filtering. For the IPv4 Service Definition IG this is done according to the IP address information and Service Mask. For the System Capabilities IG the server has to filter according to the IEEE OUI for which the client sent the query.

6.3.2 Server Side IPv4 Service Description

The service description is sent by the server when a new request from the client comes in. It is a single packet or a sequence of packets describing registered services that match the query specification sent by the client. Note that in version 1 this packet format is identical (except for the type) to the service registration packet format.

Table 6-7: Service Description Packet

Offset	Size (Octets)	Name	Function/Description
0	8	PNNI Header	A PNNI packet header structure with Packet type = 37 (Proxy PAR Service Description).
8	4	Service description packet sequence number	
12	2	Flags	See table 6-4.
14	20	ATM Address	AESA for which the service has been registered
34	1	Scope	Scope of the registration.
35	1	Reserved	
Any number of PAR IPv4 Service Definition IGs nested in PPAR VPN ID IGs if they were specified in the request, describing the registered information. The service mask describes registrations for all of the services by the bits set. For each of the registered services, inside of the PAR IPv4 Service Definition IG an appropriate service-specific IG can be included, e.g. if the BGP service bit is set, the PAR IPv4 Service Definition IG can include a PAR BGP4 IPv4 Service Definition IG with the appropriate AS. In addition, System Capabilities IG can be included if the server was queried.			

In case there is no IG matching the client's request, the server must send an empty Service Description Packet with the ATM Address and the Scope set to zero.

6.3.3 Client Side IPv4 Service Description Acknowledgment

Client side of the protocol receives the registration packets and confirms those with sequence numbers received.

Table 6-8: Service Description Acknowledgment Packet

Offset	Size	Name	Function/Description

	(Octets)		
0	8	PNNI Header	A PNNI packet header structure with Packet type = 38 (Proxy PAR Service Description Acknowledgment).
8	4	Description packet sequence number	

7. Architectural Variables

[Normative]

The following list contains architectural variables used by Proxy PAR and PAR.

Table 7-1: Architectural Variables

Variable Name	Default Value	Range
Hello Timer Interval:	15 seconds	1 to 100 seconds
Inactivity Factor:	5	2 to 10
Register Timer Interval:	3 seconds	1 to 100 seconds
Server Query Timer Interval:	3 seconds	1 to 100 seconds
Client Query Timer Interval:	3 seconds	1 to 100 seconds
Registration expiration interval	1800 seconds	100 to 10000 seconds

8. References

[ILMI] ATM-Forum 95-0417R8, "Interim Local Management Interface (ILMI) Specification 4.0", June 1996

[IS-IS] ISO 10589, Information Technology - Telecommunications And Information Exchange between Systems - Intermediate System to Intermediate System Routing Exchange Protocol for Use in Conjunction with the Protocol for Providing the Connectionless-Mode Network Service, ISO 90

[LANE] ATM-Forum, “LAN Emulation over ATM 1.0”, af-lane-0021.000, Jan 1995

[MIB] IETF, “IP Forwarding Table MIB”, F. Baker, RFC1354, July 1992

[PNNI] ATM Forum, “Private Network-Network Interface Specification Version 1.0”, March 1996, af-pnni-0055.000

[RFC1483] IETF, “Multiprotocol Encapsulation over ATM Adaptation Layer 5” Juha Heinanen, July 1993

[RFC1577] IETF, “Classical IP and ARP over ATM”, M. Laubach, January 1994

[RFC1771] IETF, “A Border Gateway Protocol”, Y. Rekhter, and T. Li, March 1995

[RFC1793] IETF, “Extending OSPF to Support Demand Circuits”, J. Moy, RFC 1793, April 1995

[RFC1966] IETF, “BGP Route Reflection”, T. Bates, June 1996

[RFC2022] ETF, “Support of Multicast over UNI 3.0/3.1 based ATM Networks”, G. Armitage, May 1996

[RFC2117] IETF, “Protocol Independent Multicast-Sparse Mode (PIM-SM)”, D. Estrin, D. Farinacci, A. Helney, D. Thaler, S. Deering, M. Handley, V. Jacobson, C Liu, P. Shama, L. Wei, June 1997

[RFC2178] IETF, “OSPF Version 2”, J. Moy, July 1997

Annex A PAR PICS[2]

A.1 Introduction

To evaluate conformance of a particular implementation, it is necessary to have a statement of which capabilities and options have been implemented for a given specification. Such a statement is called a Protocol Implementation Conformance Statement (PICS).

A.1.1 Scope

This Annex provides the PICS proforma for the ATM Forum PNNI Augmented Routing (PAR) Specification Version 1.0 in [3] in compliance with the relevant requirements, and in accordance with the relevant guidance, given in Recommendation X.296 (and/or ISO/IEC 9646-7).

A.1.2 Normative References

[1] ISO/IEC 9646-1:1994, *Information technology - Open systems interconnection - Conformance testing methodology and framework - Part 1: General Concepts.* (See also ITU Recommendation X.290(1995)).

[2] ISO/IEC 9646-7:1994, *Information technology - Open systems interconnection - Conformance testing methodology and interconnection - Part 7: Implementation conformance statements*. (See also ITU Recommendation X.296(1995)).

[3] ATM Forum af-ra-0104.000: November 1998, *PNNI Augmented Routing (PAR) Version 1.0.*

[4] ATM Forum af-uni-0010.001: September 1993, *ATM User-Network Interface Specification V3.0.*

[5] ATM Forum af-uni-0010.002: 1994, *ATM User-Network Interface Specification V3.1*.

[6] ATM Forum af-sig-0061.000: July 1996,*UNI Signaling 4.0.*

[7] ATM Forum af-ilmi-0065.000:September 1996, *ILMI 4.0.*

[8] ATM Forum af-pnni-0055.000: March 1996, *P-NNI V1.0.*

A.1.3 Definitions

This specification uses the following terms defined in ISO/IEC 9646-1[1]):

- A Protocol Implementation Conformance Statement (PICS) is a statement made by the supplier of an implementation or system, stating which capabilities have been implemented for a given protocol.
- Protocol Implementation Conformance Statement (PICS) is a document, in the form of a questionnaire, designed by the protocol specifier or conformance test suite specifier, which when completed for an implementation or system becomes the PICS.

A.1.4 Acronyms

M Mandatory
N/A Not applicable
O Optional (may be selected to suite the implementation, provided that any requirements applicable to the options are observed)

---

[2] **Copyright release for PICS:**
This PICS Proforma may be freely reproduced, this so that it may be used for its intended purpose.

O.n Optional, but support is required for either at least one or only one of the options in the group labelled with the same number “n”
PICS Protocol Implementation Conformance Statement
X eXcluded or prohibited - there is a requirement not to use this capability in a given context

A.1.5 Conformance

The supplier of a protocol implementation which is claimed to conform to the ATM Forum PNNI Augmented Routing (PAR) Specification Version 1.0 is required to complete a copy of the PICS proforma provided in this annex and is required to provide the information necessary to identify both the supplier and the implementation.”

A.2 Instructions

A.2.1 Instructions for completing the PICS proforma

The supplier of a protocol implementation which is claimed to conform to the ATM Forum PAR 1.0 specification shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.
The PICS proforma is a fixed format questionnaire. The supplier of the implementation shall complete this questionnaire, in particular identify the implementation, complete the global statement of conformance, and provide answers in the rows of the tables in clauses 5 and following. The structure of the tables is explained in subclauses 2.5 and 2.6. For each row in each table, the supplier shall enter an explicit answer (i.e. by ticking the appropriate "yes", "no", or "N/A" in each of the support column boxes provided. Where a support column box is left blank, or where it is marked "N/A" without any tick box, no answer is required. If a "prerequisite line" (see 2.6 below) is used after a subclause heading or table title, and its predicate is false, no answer is required for the whole sublause or table, respectively.
A supplier may also provide - or be required to provide - further information, categorized as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further subclause of items labelled
"a.<i>" for additional information,
"x.<i>" for exceptional information
for cross-referencing purposes, where <i> is any unambiguous identification of an item (e.g., simply a numeral); there are no other restrictions on its format and presentation.

A.2.2 Additional Information

Items of Additional Information allow a supplier to provide further information intended to assist the interpretation of the PICS. It is not intended or expected that a large quantity will be supplied, and a PICS can be considered complete without any such information. Examples might be an outline of the ways in which a (single) implementation can be set up to operate in a variety of environments and configurations.
References to items of Additional Information may be entered next to any answer in the questionnaire, and may be included in items of Exception information.

A.2.3 Exception Information

It may occasionally happen that a supplier will wish to answer an item with mandatory or prohibited status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No pre-printed answer will be found in the Support column for this. Instead, the supplier is required to write into the support column an x.<i> reference to an item of Exception Information, and to provide the appropriate rationale in the Exception item itself. An implementation for which an Exception item is required in this way does not fully conform to this Standard; and the answer to the global statement of conformance (see clause 4) cannot be "yes". A possible reason for the situation described above is that a defect in the Standards has been reported, a correction for which is expected to change the requirement not met by the implementation.

A.2.4 Legend for the columns of the PICS proforma tables

The questionnaire in clauses 5 and beyond is structured as a set of tables in accordance with the guidelines presented in ISO/IEC 9646-7. The columns of the tables shall be interpreted as follows:

"Item"
The item column contains a unique reference (a mnemonic plus a number) for each item within the PICS proforma. Items need not always be numbered sequentially.

"Item Description"

The item description column contains a brief summary of the static requirement for which a support answer is required. This may be done by a question or a reference to a specific feature.

"Predicate"
The “predicate” column contains a specification of a conditionnal status, if appropriate. The indication of an item reference in this column indicates a simple-predicate condition (support of this item is dependent on the support marked for the referenced item).
Within the "predicate" column, the logical symbol "]" is used to indicate a logical negation ("NOT").

"Status"
The following notations, defined in ISO/IEC 9646-7, are used for the status column:

M	Mandatory - the support of this capability is required for full conformance to the standard.
N/A	Not Applicable - in the given context, it is impossible to use the capability. No answer in the support column is required.
O	Optional - the capability is not required for conformance to the protocol and may or may not be supported. However, if the capability is implemented, it is required to conform to the protocol specifications.
O.<n>	Qualified optional - in this case, <n> is an integer that identifies a unique group of related optional items. If no additional qualification is indicated, the support of at least one of the optional items is required for conformance to the standard. Otherwise, the qualification and logic of the selection among the optional items is defined below the table explicitly.
X	eXcluded or prohibited - there is a requirement not to use this capability in a given context.

"Reference"
Except where explicitly stated, the reference column refers to the appropriate subclause(s) of the ATM Forum PAR Vs. 1.0 specification describing the particular item. The reference merely indicates the place(s) where the core of a description of an item can be found; additional information on this item may be contained in other parts of the PAR Vs. 1.0 specification, and has to be taken into account when making a statement about the conformance to that particular item.

"Support "
In the support column, the supplier of the implementation shall enter an explicit answer. The following notation is used:

[ ] Yes	[ ] No	Tick "yes", if item is supported; tick "No", if item is not supported.
[ ] N/A		Tick "N/A", if the item is "not applicable".

In specific cases, the indication of explicit values may be requested. Where a support column box is left blank, or where it is marked "N/A" without any tick box, no answer is required.

A.2.5 Legend for further indications of the PICS proforma tables

In addition to the columns of a table, the following information may be indicated:

"Prerequisite line"
A prerequisite line after a subclause heading or table title indicates that the whole subclause or the whole table is not required to be completed if the predicate is false. The prerequisite line takes the form:
Prerequisite:<predicate>.

"Qualification"
At the end of a table, a detailed qualification for a group of optional items may be indicated, as specified in the description of the status "qualified optional" in subclause 2.5.

"Comments"
This box at the end of a table allows a supplier to enter any comments to that table. Comments may also be provided separately (without using this box).

A.3 Identification of the implementation

Identification of the implementation and the system in which it resides should be filled in to provide as much detail as possible regarding version numbers and configuration options.
The implementation about which this PICS proforma asks questions is an ATM Forum PAR 1.0 (btd-pnni-par-01.03) implementation.
The contact person indicated (see subclause 3.6) should be able to answer queries regarding information supplied in the PICS.
As specified in clause 5 of ISO/IEC 9646-7, it is required for all implementations to provide at least the identification of the implementation (3.2), product supplier information (3.4), identification of a contact person (3.6), and detailed identification of the protocol for which the PICS applies (3.7). Identification of the system in which the implementation resides (3.3) is recommended in order to facilitate full identification of the system, and to avoid possible problems during conformance testing. The customer information (3.5) needs to be filled in only if it is relevant and different from the product supplier information.

A.3.1 Date of statement

_____

A.3.2 Identification of the implementation

The terms "name" and "version" should be interpreted appropriately to correspond with a supplier's terminology (e.g. Type, Series, Model).
Name of the implementation:

_____
_____

Implementation version:

_____
_____

A.3.3 Identification of the system in which it resides

Name of the system:

_____
_____

Hardware configuration:

_____
_____
_____

Operating system:

_____
_____

A.3.4 Product supplier

Name:

_____

Address:

_____
_____
_____

Telephone number:

_____

Facsimile number:

_____

E-Mail address:

_____

Additional information:

_____
_____
_____

A.3.5 Customer

Name:

Address:

Telephone number:

Facsimile number:

E-Mail address:

Additional information:

A.3.6 PICS contact person

Name:

Address:

Telephone number:

Facsimile number:

E-Mail address:

Additional information:

A.3.7 Identification of the Protocol

This PICS Proforma applies to the following:
ATM Forum PNNI Augmented Routing (PAR) Specification
Version 1.0, af-ra-0104.000: November 1998

Errata Implemented (if applicable):

Addenda Implemented (if applicable):

Amendments Implemented (if applicable):

A.4 Global Statement of Conformance

Does the implementation described in this PICS meet the mandatory requirements of the referenced standard:

[ ] Yes

[ ] No

Note: Answering "No" to this question means that full conformance to the protocol specification must not be claimed. In this case, an explanation shall be given of the nature of nonconformance either below or on a separate sheet of paper. Furthermore the instructions outlined in subclause 2.4 ("Exception Information") shall be followed when completing the PICS proforma tables.

Nature of nonconformance (if applicable):

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

A.5 Roles

Roles are used in PICS proformas as predicates for the "predicate" column if an item is conditional upon the role(s) supported.

Item	Roles Is the implementation capable of ...	Status	Predicate	Reference	Support
Role1	functioning as a Proxy-PAR client?	O.1		3.2	[ ]Yes [ ]No
Role2	functioning as a PAR device?	O.1		3.1	[ ]Yes [ ]No
Role2.1	functioning as a PAR-capable Proxy-PAR server?	O N/A	Role2] (Role2)	3.2	[ ]Yes [ ]No [ ]N/A
Comments: O.1: support of at least one of these is mandatory.					

Table A-1: Roles

A.6 Additional Specifications implemented to support ATM Forum PAR V. 1.0

Note: It is strongly recommended that for the supported specifications, separate PICS be provided according to the PICS proforma specified for those specifications, and that they be attached to this PICS proforma. If not available, it is recommended to at least provide an informal sheet responding to the questions of clauses 3 and 4 above for that specification.

Item	Additional Specifications Supported Does the implementation support ...	Status	Predicate	Reference	Support
AddS1.1	ATM Forum UNI 3.0 specification?	O		[4]	[ ]Yes [ ]No
AddS1.2	ATM Forum UNI 3.1 specification?	O		[5]	[ ]Yes [ ]No
AddS1.3	ATM Forum UNI 4.0 specification?	O		[6]	[ ]Yes [ ]No
AddS1.4	ILMI extensions for Proxy-PAR detection?	O		[7]	[ ]Yes [ ]No
AddS2	ATM Forum PNNI 1.0 specification?	M	Role2	[8]	[ ]Yes [ ]No
Comments:					

Table A-2: Additional Specifications Supported

A.7 General Proxy-PAR

Prerequisite: Role1 or Role 2.1

Item	Proxy-PAR Is the implementation capable of...	Status	Reference	Support
PPAR-1	Using the default VCI 0:18 for Proxy-PAR Messages between the Proxy-PAR client and Proxy-PAR server?	M	5.1	[ ]Yes [ ]No
PPAR-1.1	Supporting which value/range different than the default VPI?	O	5.1	Value/Range:
PPAR-2	Creating a new instance of the Proxy-PAR client/server pair for each link where Proxy-PAR is configured?	M	3.2	[ ]Yes [ ]No
Comments:				

Table A-3: General Proxy-PAR capabilities

A.8 Proxy-PAR Hello Protocol

Prerequisite: Role1 or Role 2.1

Item	Proxy-PAR Hello Protocol Is the implementation capable of...	Status	Reference	Support
H-PPAR-1	Transmitting Hello Messages as long as the link is up?	M	5.2.1	[ ]Yes [ ]No
H-PPAR-2	Transmitting a Hello Message every Hello Interval seconds?	M	5.2.1	[ ]Yes [ ]No
H-PPAR-3	Using the Hello Interval value retrieved from the latest received Hello Message in order to reset the Inactivity Timer to Hello Interval times Inactivity Factor?	M	6.1.1	[ ]Yes [ ]No
H-PPAR-4	Using the default value of 5 for the Inactivity Factor?	M	7	[ ]Yes [ ]No
H-PPAR-5	Supporting a range of 2 to 10 for the Inactivity Factor?	O	7	Value/Range:
H-PPAR-6	Using the default value of 15 sec for the Hello Interval ?	M	7	[ ]Yes [ ]No
H-PPAR-7	Supporting a range of 1 to 100 for the Hello Interval?	O	7	Value/Range:
H-PPAR-8	Supporting all Hello Message parameters?	M	6.1.1	[ ]Yes [ ]No
H-PPAR-9	Discovering the ATM address of the client or server by sending Hello Messages with a Remote Node ATM End-System Address set to the null address?	M	5.2	[ ]Yes [ ]No
Comments:				

Table A-4: Proxy-PAR Hello Protocol capabilities

Prerequisite: Role1

Item	Client Proxy-PAR Hello Protocol Is the implementation capable of...	Status	Reference	Support
HC-PPAR-1	Transmitting the first Hello Message after initialization or link failure?	M	5.2.1	[ ]Yes [ ]No
HC-PPAR-2	Retrieving the Registration Expiration Interval from the Hello Message sent by the Proxy-PAR server?	M	5.2.5	[ ]Yes [ ]No
HC-PPAR-3	Establishing an adjacency with a new server after a link failure or a change in the topology?	M	5.2.1	[ ]Yes [ ]No
HC-PPAR-4	Supporting exclusively a value of 33 for the Hello Message PNNI Packet Type?	M	6.1.1	[ ]Yes [ ]No
Comments:				

Table A-5: Client Proxy-PAR Hello Protocol capabilities

Prerequisite: Role2.1

Item	Server Proxy-PAR Hello Protocol Is the implementation capable of...	Status	Reference	Support
HS-PPAR-1	Starting to send Hello Messages to a client only after receiving the first Hello Message from that client, after initialization or link failure?	O	5.2.6	[ ]Yes [ ]No
HS-PPAR-2	Inserting the Registration Expiration Interval into the Hello Message sent to the Proxy-PAR client?	M	5.2.6	[ ]Yes [ ]No
HS-PPAR-3	Supporting exclusively a value of 32 for the Hello Message PNNI Packet Type?	M	6.1.1	[ ]Yes [ ]No
Comments:				

Table A-6: Server Proxy-PAR Hello Protocol capabilities

A.9 PAR

Prerequisite: Role2 or Role2.1

Item	PAR **Is the implementation capable of...**	Status	Reference	Support
PAR-1	Supporting all parameters of the PAR Service IG?	M	4.2	[ ]Yes [ ]No
PAR-2	Supporting all parameters of the PAR VPN ID IG?	M	4.2.1	[ ]Yes [ ]No
PAR-3	Supporting all parameters of the PAR IPv4 Service Definition IG?	M	4.2.1.1.1	[ ]Yes [ ]No
PAR-4	Recognizing the PAR OSPF Service Definition IG?	O	4.2.1.1.1.1	[ ]Yes [ ]No
PAR-5	Recognizing the PAR MOSPF Service Definition IG?	O	4.2.1.1.1.2	[ ]Yes [ ]No
PAR-6	Recognizing the PAR BGP4 Service Definition IG?	O	4.2.1.1.1.3	[ ]Yes [ ]No
PAR-7	Recognizing the PAR DNS Service Definition IG?	O	4.2.1.1.1.4	[ ]Yes [ ]No
PAR-8	Recognizing the PAR PIM-SM Service Definition IG?	O	4.2.1.1.1.5	[ ]Yes [ ]No
PAR-9	Setting all information group tag bits of PAR PTSEs to 0?	M	4.3	[ ]Yes [ ]No
Comments:				

Table A-7: PAR capabilities

A.10 Proxy-PAR Registration/Query Protocol

Prerequisite: Role1 or Role2.1

Item	Proxy-PAR Registration/Query Protocol Is the implementation capable of...	Status	Reference	Support
RQ-PPAR-1	Running a r/q session (registration/query) instance for each client/server pair instance (identified by a pair of AESAs)?	M	5.3	[ ]Yes [ ]No
RQ-PPAR-2	Receiving all r/q messages when the adjacency is up?	M	5.3	[ ]Yes [ ]No
RQ-PPAR-3	Running registration sessions only one after the other between the same client/server pair instance?	M	5.3	[ ]Yes [ ]No
RQ-PPAR-4	Using the default value of 3 sec for the Registration Timer ?	M	7	[ ]Yes [ ]No
RQ-PPAR-5	Supporting a range of 1 to 100 sec for the Registration Timer?	O	7	Value/Range:
RQ-PPAR-6	Using the default value of 3 sec for the Query Timer?	M	7	[ ]Yes [ ]No
RQ-PPAR-7	Supporting a range of 1 to 100 sec for the Query Timer?	O	7	Value/Range:
RQ-PPAR-8	Running query sessions only one after the other between the same client/server pair instance?	M	5.3	[ ]Yes [ ]No
RQ-PPAR-9	In a Service Registration or Service Description packet, setting the appropriate Service Mask bit of the IPv4 IG for each Service Definition IG present in that packet?	M	6.2.1	[ ]Yes [ ]No
RQ-PPAR-10	Considering only the first occurrence of an IG if the same IG within the same nesting IGs and same AESA/scope occurs multiple times during the same registration or query session?	M	5.3	[ ]Yes [ ]No
Comments:				

Table A-8: Proxy-PAR Query/Registration Protocol capabilities

Prerequisite: Role1

Item	Client Proxy-PAR Registration/Query Protocol Is the implementation capable of...	Status	Reference	Support
RQC-PPAR-1	Registering and querying services to/from the Proxy-PAR server?	M	5.3	[ ]Yes [ ]No
RQC-PPAR-2	Using as default scope the local network membership scope?	M	0 A.5.3 in UNI 4.0	[ ]Yes [ ]No
RQC-PPAR-4	Registering one or many new services by registering this or these new services together with all previously registered services?	M	5.3	[ ]Yes [ ]No
RQC-PPAR-5	Unregistering one or many services by registering all previously registered services but this or these services?	M	5.3	[ ]Yes [ ]No
RQC-PPAR-6	Unregistering all services by registering the null service?	M	3.2.1	[ ]Yes [ ]No
RQC-PPAR-7	Unregistering all services by withholding Hello messages until the adjacency goes down?	O	3.2.1	[ ]Yes [ ]No
RQC-PPAR-8	Initially querying the server before registering any services?	O	5.3.1.3	[ ]Yes [ ]No
RQC-PPAR-9	Registering all services again before they expire at the server?	M	3.2.1	[ ]Yes [ ]No
RQC-PPAR-10	Using the Registration Expiration Interval received in the latest Hello Message from the server to compute the value of the Refresh Timer?	M	5.2	[ ]Yes [ ]No
RQC-PPAR-11	Using a value less than the Registration Expiration Interval for the Refresh Timer?	M	5.2	[ ]Yes [ ]No
RQC-PPAR-12	Supporting which value/range for the Refresh Timer?	M	5.2	Value/Range:
RQC-PPAR-13	Purging all information learned from a server when the adjacency goes down?	M	5.2.1	[ ]Yes [ ]No
RQC-PPAR-14	Picking an initial packet sequence number different than 0 and significantly smaller than the highest sequence number?	M	5.3	[ ]Yes [ ]No
RQC-PPAR-15	Sending repeatedly Registration, Query and Description Ack Messages until they are acknowledged by the appropriate message from the server?	M	5.3.1.3& 5.3.2.3	[ ]Yes [ ]No
RQC-PPAR-16	Supporting all parameters in the Service Registration Acknowledgment Message?	M	6.2.2	[ ]Yes [ ]No
RQC-PPAR-17	Supporting all parameters in the Service Description Message?	M	6.3.2	[ ]Yes [ ]No
Comments:				

Table A-9: Client Proxy-PAR Query/Registration Protocol capabilities

Prerequisite: Role2.1

Item	Server Proxy-PAR Registration/Query Protocol Is the implementation capable of...	Status	Reference	Support
RQS-PPAR-1	Receiving registrations from the client and acknowledging them?	M	5.3	[ ]Yes [ ]No
RQS-PPAR-2	Translating the registered services into PTSEs, passing them to the PNNI database and initiating the flooding?	M	3.2.1 & 3.2.2	[ ]Yes [ ]No
RQS-PPAR-3	Filtering PAR VPN ID IGs and PAR IPv4 IGs from the PTSEs in the PNNI database based on the arguments received in a query (scope, IP address and mask, VPN ID)?	M	5.3.2.3	[ ]Yes [ ]No
RQS-PPAR-4	Filtering PAR IPv4 IGs contained in PAR VPN ID IGs from the PTSEs in the PNNI database based on the arguments received in a query (scope, IP address and mask, VPN ID) if the query does not contain the same VPN ID IGs?	X	5.3.2.3	[ ]Yes [ ]No
RQS-PPAR-5	Returning the Service Descriptions with the filtered IGs?	M	3.2.2	[ ]Yes [ ]No
RQS-PPAR-6	Maintaining a mapping table of membership scopes (transmitted to/from the client) to PNNI scopes (used in PTSEs)?	M	5.3.1& A.5.3 in UNI 4.0	[ ]Yes [ ]No
RQS-PPAR-7	Purging all appropriate PSTEs from the PNNI database when the adjacency with the client who registered the corresponding services goes down?	M	5.2.1 & 5.2.6	[ ]Yes [ ]No
RQS-PPAR-8	Purging all appropriate PTSEs from the PNNI database when receiving a registration for the null service from the same client who registered the corresponding services?	M	5.2.1 & 3.2.1	[ ]Yes [ ]No
RQS-PPAR-9	Purging all appropriate PTSEs from the PNNI database when the Registration Expiration Timer expires?	M	5.3	[ ]Yes [ ]No
RQS-PPAR-10	Supporting all parameters in the Service Request Message?	M	6.3.1	[ ]Yes [ ]No
RQS-PPAR-11	Supporting all parameters in the Service Registration Message?	M	6.2.1	[ ]Yes [ ]No
RQS-PPAR-12	Supporting all parameters in the Service Description Acknowledgment Message?	M	6.3.3	[ ]Yes [ ]No
RQS-PPAR-13	Using a default value of 1800 sec for the Registration Expiration Interval?	M	7	[ ]Yes [ ]No
RQS-PPAR-14	Supporting a range of 100 to 10000 sec for the Registration Expiration Interval?	O	7	Value/Range:
RQS-PPAR-15	Inserting the appropriate Return Code in the Service Registration Acknowledgment packet?	M	6.2.2	[ ]Yes [ ]No
RQS-PPAR-16	Putting together IGs under the same nesting IGs if these nesting IGs are identical, as well the corresponding ASEA/scope, before inserting them in the database?	O		[ ]Yes [ ]No
Comments:				

Table A-10: Server Proxy-PAR Query/Registration Protocol capabilities

Appendix A. OSPF Example

A.1 Introduction

The PAR baseline provides the general definitions of the information to be exchanged using PAR. This includes the elements required to support OSPF, but does not describe how these are to be used. This appendix provides an informative description of how PAR information can be used by OSPF.

A.2 Overview

Proxy PAR provides a mechanism by which Proxy PAR clients that provide application services (Servers) attached to ATM may advertise the services they provide. PNNI and PAR are then used by the switches to propagate the information. Proxy PAR provides the means to fetch this information by a client. The OSPF Information Group provides the information about a router using OSPF. The information includes the Area supported, the router priority, and the interface type. If a router supports several areas over ATM, it would have several service advertisements.

The following text describes a simple way of using this information. It assumes that a full mesh of routers is desired, and that the router priority information in the Proxy PAR advertisement is usable.

A.3 Implementation

A router running OSPF may make use of Proxy PAR. Although the details are to be thoroughly defined by the IETF [PPOSPF], this document provides an example of how this may be done in a reasonably sized environment.

A.3.1 Advertisement Generation

For each area in which the router participates in over ATM, it will produce a PAR OSPF IPv4 Service Definition IG. This IG will include the Area ID and the router's priority in that area. The upper bit of the router priority is the DR indication, as per OSPF. The interface type will be advertised as NBMA.

A.3.2 Advertisement Usage

The OSPF router will get the service definitions from the switch. For each area in which it is participating, it will select from the services those routers that are participating in the same Area.

A.3.2.1 Designate Router Election

The router will use the router priorities from the service advertisements to determine who should be the designated router.

A.3.2.2 Router VCs

The router will peer with all of the other routers advertising service in the same area. It will attempt to bring up a VC to each router if this router's ATM address is numerically lower than the advertising router. For all routers advertising service to the same OSPF area, if no VC has been established after 30 seconds, the router will attempt to establish a VC to the advertising router. This attempt will be retried every 60 seconds thereafter. If two VCs are established between two routers, the one established by the lower addressed device shall be used by both devices. Idle VCs should be released. These VCs are used for OSPF exchanges and for forwarding IPv4 packets. In the absence of configuration, these VCs shall be of class UBR.

Appendix B. VPN Support

In order to implement virtual private networks all information distributed via PAR has a corresponding VPN ID. Based on this ID, individual VPNs can be separated. Inside a certain VPN further distinctions can be made according to IP address related information and/or protocol type.

In most cases the best VPN support can be provided when Proxy PAR is used between the client and server because in this way it is possible to hide the real PNNI topology from the client. The PAR capable server performs the translation from the abstract membership scope into the real PNNI routing level. In this way the real PNNI topology is hidden from the client and the server can apply restrictions in the PNNI scope. The server can for instance have a mapping such that the membership scope “global” is mapped to the highest level peergroup to which a particular VPN has access. Thus the membership scopes can be seen as hierarchical structuring inside a certain VPN. With such mappings a network provider can also change the mapping without having to reconfigure the clients.

For more secure VPN implementations it will also be necessary to implement VPN ID filters on the server side. In this way a client can be restricted to a certain set (typically one) of VPN IDs. The server will then allow queries and registrations only from the clients that are in the allowed VPNs. In this way it is possible to avoid an attached client from finding devices that are outside of its own VPN. There is even room for further restriction in terms of not allowing wildcard queries by a client.

In terms of security, some of the protocols have their own security methods, so PAR is only used for the discovery of the counterparts. For instance OSPF has authentication which can be used during the OSPF operation. So even in the case where two wrong partners find each other, they will not communicate because they will not be able to authenticate each other.