

PNNI Augmented Routing (PAR) and Proxy-PAR

Robert Haas, Patrick Droz, Daniel Bauer

*IBM Research Division, Zurich Research Laboratory, Säumerstrasse 4,
8803 Rüschlikon, Switzerland*

Abstract

ATM networks are often used to carry IP traffic, but IP over ATM techniques suffer from complex and error-prone configuration. PAR (PNNI Augmented Routing) is an extension to PNNI to simplify IP support. In addition, a very lightweight interface called Proxy-PAR enables ATM-attached IP devices to operate without a full PAR implementation. PAR and Proxy-PAR provide a service discovery system for IP devices attached to ATM-PNNI networks. PAR and Proxy-PAR are standards from the ATM Forum, and are referenced by the IETF. This paper shows how PAR and Proxy-PAR can complement solutions such as ILMI-based Server Discovery, Classical IP and ARP over ATM, and NHRP, or even replace existing solutions, for instance where a distributed address resolution mechanism is preferable to a centralized one. This is then described in the context of three different types of networks: a campus, a backbone, and a mobile network.

Key words: PNNI Augmented Routing, IP/ATM integration, service discovery, automatic configuration, address resolution.

1 Introduction

ATM technology is now widely deployed in the campus and service provider environments thanks to its maturity and value-added features such as the inherent quality-of-service (QoS) support. In addition, ATM supports a wide range of link bandwidths. It is clear that in most cases ATM is deployed to carry IP traffic because ATM applications are not commonplace. For that reason, efficient methods to run IP over ATM are necessary.

This can be achieved in two radically different ways: integration or emulation. Integration tries to combine certain information generated by IP and ATM,

whereas emulation aims at hiding the ATM particularities to IP and deploying IP as an overlay over ATM.

The methods presented here, PAR (PNNI Augmented Routing) [8,29] together with Proxy-PAR, belong to the integration family. As we will show later, emulation methods can also benefit from PAR.

PAR is a successful result of common work between the ATM Forum and the IETF [18,28]. It is an extension to PNNI, which is the standard protocol used by ATM switches for routing and signaling. PAR is used to distribute information about external services across the ATM backbone network. This information is therefore not related to ATM itself, but rather to the IP nodes attached to the ATM network and the services they support. Thus, IP nodes will automatically learn about the various services available from other IP nodes connected to the same ATM backbone.

The paper is structured as follows: Section 2 gives an overview of various IP-ATM integration strategies, and positions PAR and Proxy-PAR in this context. Section 3 shows the details of PAR and Proxy-PAR. Section 4 shows where PAR and Proxy-PAR can be useful. Section 5 presents several examples where PAR and Proxy-PAR ease configuration tasks and offer higher robustness. Section 6 presents related work and a comparison of various methods including PAR and Proxy-PAR. Section 7 concludes the paper and describes future work.

2 Integration of IP and ATM

Integration of IP and ATM can be achieved at various levels, ranging from a tight to a loose integration. We describe below where and how integration can be achieved, but before we can do this, some details on the routing protocol itself are required. It should then become clear what we mean by integration, compared to emulation methods such as Classical IP and ARP over ATM [23] or LANE [3], and their respective shortcut-support extensions NHRP [24] and MPOA [6], described in Section 4.

In the following, we use the generic term *IP service* to describe any kind of network service, such as a particular routing protocol or name resolution service, that is available at a certain IP address in the network. Examples of such services can be found in Table 5.

2.1 Link-state routing protocols

PNNI [9] is the routing protocol used in ATM, and has key features like support for QoS and hierarchy. As in any other link-state routing protocol such as OSPF [26] and ISIS [12], two types of information are exchanged between nodes running a link-state routing protocol:

- *topology* information for nodes and links that describes how nodes are interconnected, and
- *reachability* information describing the address prefixes that can be reached at each node.

Integration of IP and ATM routing can be achieved at various levels: by integrating topology information, reachability information or both. The pros and cons of the different approaches are described below.

2.2 Topology and reachability integration

The most extreme integration is to make routers and switches appear in a common routing topology, where only the PNNI routing protocol distributes reachability information generated by both types of nodes, as shown in Figure 1 (for illustration purposes, we show routers interconnecting Ethernet networks through the ATM backbone). This was, for instance, the attempt of Integrated-PNNI (I-PNNI) [2].

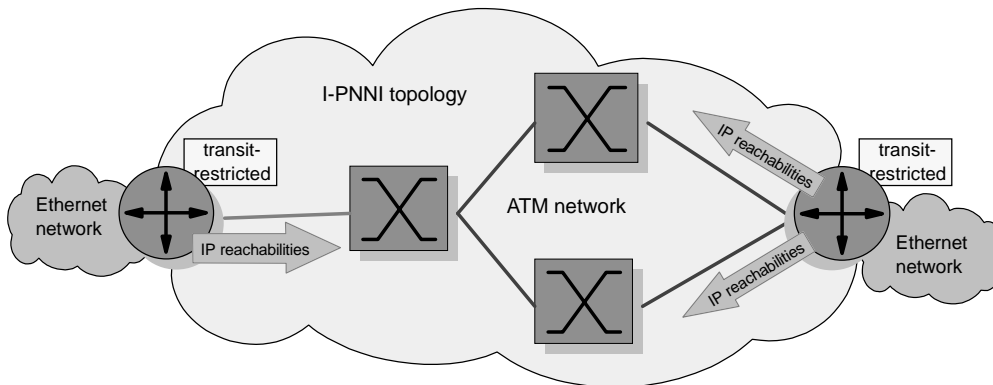


Fig. 1. Topology and reachability integration.

In I-PNNI, a single routing protocol is used that computes routes for ATM and IP destinations. Routers and switches generate reachability information, and they appear together in the topology derived by the routing protocol, with the slight difference that routers are tagged as *transit-restricted* because it is

clear that they cannot switch virtual circuits (VCs). Unfortunately, having two different types of devices creates difficulties in the case of a hierarchical network, where information about groups of nodes has to be summarized by a single node, the Peer Group Leader (PGL). This induces some constraints on the physical topology itself, for example, in how switches and routers are interconnected, that are not easily overcome.

2.3 Reachability integration

Another integration alternative is to have ATM switches generate IP reachability information on behalf of their attached routers, without having the routers appear in the topology at all, as shown in Figure 2. Here the routers are shown with dashed outlines to emphasize that they are no longer part of the PNNI topology.

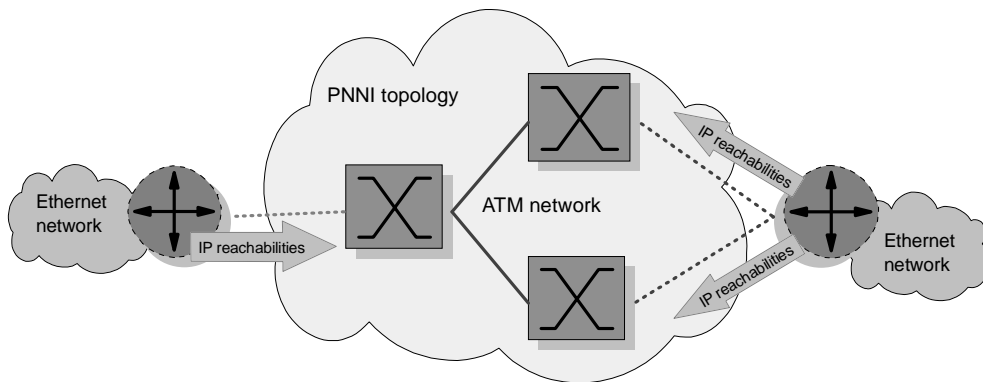


Fig. 2. Reachability integration.

To our knowledge, there is no proposal that pursues this direction. This level of integration would considerably simplify certain aspects compared to I-PNNI, as in this case only switches appear in the topology instead of a mix of routers and switches. A drawback is that an additional protocol would be needed between the switch and its attached router to perform the exchange of reachability information.

2.4 Topology and service integration

The next level of integration is when an IP router shows up in the PNNI topology, but does not advertise any reachability information. In order to do so, the IP router runs a full PNNI as well as an IP stack. This is shown in the router on the left-hand side of Figure 3. Topology integration means that

the router distributes topology information throughout the PNNI network. Being an extensible protocol, PNNI allows the router to advertise information that is not related to the topology, but to IP services that are external to PNNI. In that sense, PNNI can be viewed as a generic information distribution mechanism, not exclusively dedicated to pure routing protocol information. By “service integration” we mean here that PNNI is used to distribute information about external services, and that PNNI itself makes no use of this information.

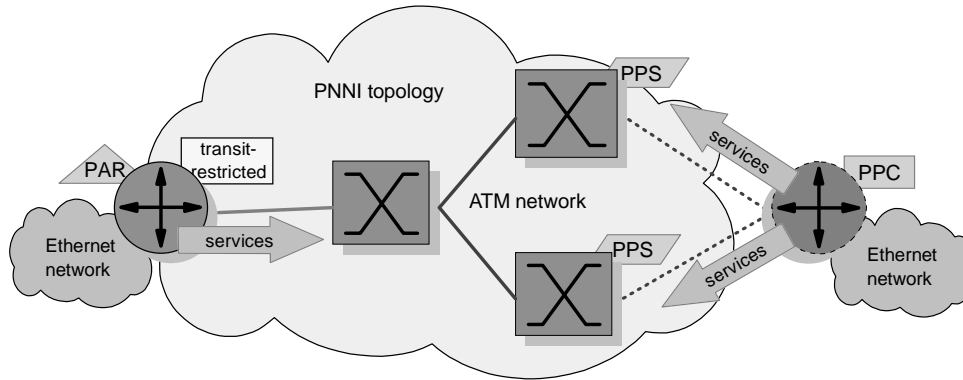


Fig. 3. Service integration.

Topology integration is costly in terms of processing resources required at the router, as a dual stack is needed. On the other hand, it is then theoretically possible for IP routing protocols to see the complete underlying ATM network topology, and make better-informed decisions on how they should operate.

The following subsection shows how these high costs can be avoided.

2.5 Service integration

The loosest integration level consists of keeping routers outside of the PNNI routing topology, and not let PNNI handle IP reachability information, as shown in the router on the right-hand side of Figure 3. Instead, PNNI only distributes information about external services advertised by the router. It is then up to the routers to create their own overlays and exchange reachability information (IP routes) independently of PNNI, for instance, using the OSPF routing protocol.

Similarly to the reachabilities exchange protocol mentioned in Section 2.3, an additional protocol is needed so that the switch can generate information on external services on behalf of its attached router. Unlike reachabilities, information regarding services does not need to be handled at a high time granularity, because service changes are not expected to occur as frequently

as reachability changes. Moreover, as the PNNI protocol does not utilize the information about these external services, there is no need here to deal with routing loops and other annoying phenomena within PNNI. A very simple exchange protocol can therefore be used for that purpose.

This alternative is the least intrusive of all: IP routing protocols are not replaced by PNNI, nor do they need to cooperate with PNNI, unlike the approach given in Section 2.3. Only the availability of external services is handled through PNNI, something that is not at all done in the emulation approaches such as Classical ARP and IP over ATM or LANE.

2.6 Service integration with PAR

PAR (PNNI Augmented Routing) [8], as the name suggests, refers to the PNNI routing protocol augmented with information on non-ATM services. PAR takes the integration approach shown in Section 2.4 to allow IP to benefit from the advantages of ATM and PNNI in a non-intrusive way.

Table 1 summarizes the points made above and illustrates why we consider PAR to be a non-intrusive integration of IP and ATM as compared to I-PNNI. We use the example in which routers use OSPF as their routing protocol. It is not necessary for routers to run PAR, which corresponds to an extended PNNI stack implementation. As the only purpose for a router to run PAR is to be able to register and query for IP services, a proxy solution, appropriately called Proxy-PAR, can be used. Proxy-PAR is the additional exchange protocol mentioned in Section 2.5. Proxy-PAR requires considerably fewer resources at the router than PAR. Note that not even all switches need to run PAR and Proxy-PAR: only switches connected to routers with Proxy-PAR are required to run PAR and Proxy-PAR, and all other switches that can become PGLs are required to run PAR. This is explained in Section 3.1.

Table 1

Comparison of IP-ATM integration methods.

	I-PNNI		PAR		PAR+Proxy-PAR	
	switches	routers	switches	routers	switches	routers
reachability topology	I-PNNI	I-PNNI	PNNI	PNNI, OSPF, etc.	PNNI	OSPF, etc.
services	-	-	-	PAR	PAR+Proxy-PAR	Proxy-PAR

3 PNNI, PAR and Proxy-PAR

This section describes the main concepts of PNNI on which PAR relies: the data structures where pieces of information are recorded (Information Groups,

IGs); the containers that carry these pieces of information throughout the network (PNNI Topology State Elements, PTSEs); and finally peer groups, levels and scoping, which are the elements used to build the PNNI hierarchy. The interested reader is referred to [9] for an in-depth description of PNNI. We then describe the main aspects of PAR, the specific PAR IG and the other IGs nested within this one, and illustrate how these IGs are used with a simple example. Finally, we provide an overview of Proxy-PAR. These aspects are of course covered in greater detail by the standard specification [8].

3.1 PNNI information distribution mechanism

As mentioned, PAR uses PNNI for the distribution of information. PNNI is a hierarchical link-state routing protocol that clusters the network into domains called peer groups. Within such a peer group, a flooding mechanism distributes link-state information reliably among all switches. The network is described in IGs of various types that are encapsulated into PTSEs. Each PTSE contains a single type of root IG, and optional System Capabilities IGs which contain proprietary information. All these PTSEs are stored in the *PNNI database* within each switch. Each IG describes a well-defined part of the topology, such as a node (switch), a link, or a set of reachable addresses, according to its type. All IGs are TLV-encoded (Type-Length-Value) and certain IGs can be nested inside others. PAR has its own IG type. Although switches that are not equipped with PAR cannot process PAR IGs, they will nevertheless forward all PTSEs to their neighbors (in the same peer group), including those containing PAR IGs. Thus, flooding inside a peer group works even if not all of the switches are equipped with the specific PAR extensions to PNNI.

Peer groups in PNNI are arranged in a hierarchical way, i.e., a peer group might be part of another, higher-layer peer group called *parent peer group*. Switches belonging to different peer groups do not exchange the complete topology information. Instead, each peer group elects a *Peer Group Leader* (PGL). The PGL is responsible for summarizing reachable addresses and forwarding those to its parent peer group. PNNI reachability IGs contain a scope that denotes the highest level to which the address is forwarded. This mechanism is extended to PAR IGs. The PGL has to be PAR-capable in order to interpret PAR IGs correctly. Upon receiving a PAR IG, the PGL investigates its scope and forwards it to the higher-layer peer group if the scope allows it. Otherwise, the PAR IG is only distributed within the peer group. As opposed to reachabilities, it is not possible for the PGL to summarize PAR IGs. This mechanism of scoping is illustrated later in Figure 9, where two peer groups are defined at PNNI level 80 and their parent peer group at level 60, and certain PAR IGs have a scope of 60, others of 80. In PNNI, levels are indicated by the length of the ATM address prefix. Therefore a level of 60 has a more

global scope than a level of 80, i.e., an IG originated with a scope of 60 will be flooded into larger parts of the network than an IG with a scope of 80.

3.2 PAR Information Groups

The PAR specification introduces in total eight new IGs, listed in Table 2. The PAR Service IG (type 768) always comes at the root. The seven other IGs are nested within this root IG. IGs are nested within each other to structure the information in a hierarchical manner: nested IGs contain additional and more specific information referring to their nesting IG. In the second column the table gives the possible nesting of the IGs. In what follows, we describe the PAR Service IG, the PAR *virtual private network identifier* (VPN ID) IG, and the PAR IPv4 Service Definition IG. These three IGs are usually found nested in that order provided there is a VPN ID IG. We also describe the PAR IPv4 OSPF Service Definition IG, and the PAR IPv4 BGP Service Definition IG, which are both nested into the PAR IPv4 Service Definition IG.

Table 2
PAR IGs and their nesting.

IG Name	Nested in
PAR Service IG	PTSE
PAR VPN ID IG	PAR Service IG
PAR IPv4 Service Definition IG	PAR VPN ID IG <i>or</i> PAR Service IG
PAR IPv4 OSPF Service Definition IG	PAR IPv4 Service Definition IG
PAR IPv4 MOSPF Service Definition IG	PAR IPv4 Service Definition IG
PAR IPv4 BGP4 Service Definition IG	PAR IPv4 Service Definition IG
PAR IPv4 DNS Service Definition IG	PAR IPv4 Service Definition IG
PAR IPv4 PIM-SM Service Definition IG	PAR IPv4 Service Definition IG

3.2.1 PAR Service IG

The PAR Service IG shown in Table 3 is the root IG for all Service Definition IGs referring to the same *ATM End System Address* (AESA) and scope. The scope restricts the distribution of information, and the combination of AESA and scope must lead to a unique PAR Service IG. The services defined in the IGs nested within the PAR Service IGs are available at the AESA defined in

this IG. As the VPN ID IG is not a Service IG, it is meaningless to have it alone (without any Service IG nested inside) nested within a PAR Service IG. A PAR Service IG may contain several different PAR VPN ID IGs or PAR IPv4 Service Definition IGs.

Table 3
PAR Service IG.

Offset (bytes)	Size (bytes)	Name	Function/Description
0	2	Type	Type: 768 (PAR Service IG)
2	2	Length	
4	1	Scope	PNNI Routing level
5	3	Reserved	
8	20	ATM Addr.	AESA of registered service

3.2.2 VPN ID IG

The PAR VPN ID IG shown in Table 4 contains the VPN ID. The seven bytes of the VPN ID are structured into two parts. The first three bytes contain the OUI (Organization Unique Identifier), the next four the VPN index. A VPN index alone is unique only together with the OUI part. The VPN ID is similar to the one used in ATM Forum standards and the one given by the IETF [19].

Table 4
PAR VPN ID IG.

Offset	Size	Name	Function/Description
0	2	Type	Type: 776
2	2	Length	
4	1	Reserved	
5	7	VPN ID	OUI and VPN index

The VPN ID IG can appear multiple times with different VPN IDs within the same nesting IG. It can be completely absent if no use of VPN support is made.

3.2.3 IPv4 Service Definition IG

The next IG type can only occur nested within a PAR VPN ID IG or directly in a PAR Service IG. It describes a specific service (or protocol) with associated data for the specified AESA, scope and VPN ID (if present) contained in the nesting PAR Service IG and VPN ID IG. At present, only the IPv4 protocol is supported by PAR, and therefore only the IPv4 Service Definition IG has been defined (Table 5).

The IPv4 Service Definition IG contains all necessary information about the

IPv4 address (IP address, and subnet mask) and a bitmask to indicate the IPv4 protocols and services bound to that IP address.

Table 5
PAR IPv4 Service Definition IG.

Offset	Size	Name	Function/Description	
0	2	Type	Type: 784	
2	2	Length		
4	4	IP addr	Address of Service	
8	4	IP addr mask	Subnet mask	
12	8	Service mask	Bitmask of Services	
			Bit Protocol	
			01	OSPF
			02	RIP
			03	RIPv2
			04	BGP3
			05	BGP4
			06	EGP
			07	IDRP
			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
59	PIM-DM			
60	MARS			
61	NHRP			
62	ATMARP			
63	DHCP			
64	DNS			

For some of the protocols or services indicated in the bitmask field, specific IGs have already been defined to hold additional information about that protocol or service. It is expected that more IGs will be defined in the future which can be nested within the PAR IPv4 Service Definition IG. These IGs will contain additional information deemed useful for the users of the PAR information, such as the routers running a Proxy-PAR client.

3.2.4 OSPF Service Definition IGs

As examples we give here two specific IGs for the OSPF and BGP4 [30] routing protocols.

The OSPF Service Definition IG (Table 6) 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 only a small optimization in terms of packets, this can result in significant savings in terms of connections (Switched Virtual Circuits, 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.

Table 6
PAR OSPF Service Definition IG.

Offset	Size	Name	Function/Description
0	2	Type	Type: 800
2	2	Length	
4	4	Area IAD	OSPF Area
8	1	Priority	Router Priority
9	1	Interface	Interface type
			Value Type
			0 unspecified
			1 point-to-point
			2 broadcast
			3 NBMA
			4 point-to-multipoint
5 virtual link			

The BGP4 Service Definition IG (Table 7) 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 [11], the necessary configuration information can also be included.

3.2.5 Example of OSPF Service Definition IG

Figure 4 shows an example of how two OSPF routers use PAR for OSPF service discovery. The routers are connected to an ATM network and participate in PNNI, in the “topology and service integration” way described in Section 2.4. The router with IP address 192.168.1.1 injects a PAR Service IG into PNNI to advertise its OSPF capabilities, as shown in Table 8.

Analogous PAR IGs are injected by the routers with IP addresses 192.168.1.2 and 192.168.1.3. The ATM switches simply flood these IGs throughout the

Table 7
 PAR BGP4 Service Definition IG.

Offset	Size	Name	Function/Description
0	2	Type	Type: 802
2	2	Length	
4	4	AS	Autonomous System Number
8	4	BGP ID	BGP Identifier
12	4	RR Cluster	RR Cluster ID
16	4	RR Type	Route Reflector type
			Value Function
			0 non-Client Peer
		1 Client Peer	
		2 Route Reflector	
20	4	RR ID	ID of Route Reflector

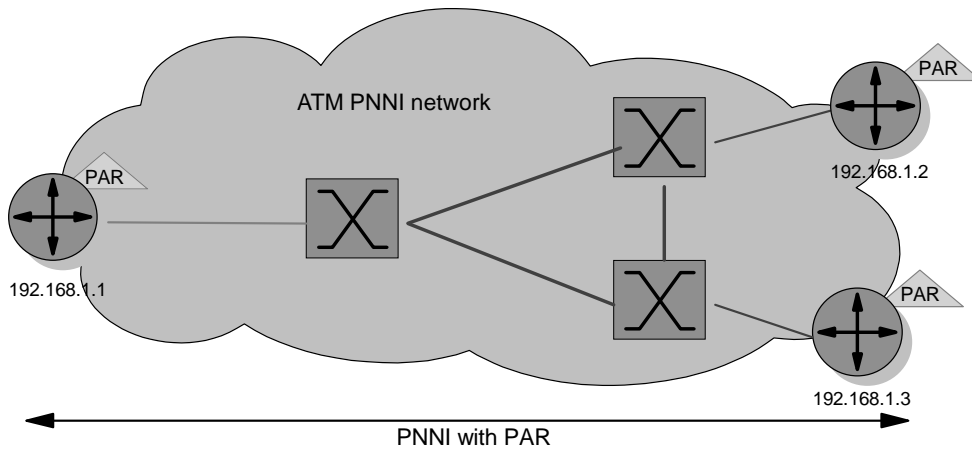


Fig. 4. PAR example.

PNNI network. Eventually, the PAR IGs will have been distributed to all routers, and the appropriate information contained in the OSPF Service Definition IGs is forwarded to the OSPF routing module. The interpretation of this information and the resulting action taken depends on OSPF and is beyond the scope of PAR. In this example, the OSPF routers will establish a full mesh of ATM SVCs. To preclude having two SVCs between each router pair, the router with the higher IP address will typically set up a bi-directional SVC. The router with the lower IP address only initiates a connection setup if no bi-directional connection has been established after a definable time-out. More details on the operation of OSPF with PAR and Proxy-PAR are given in [28].

Table 8
Example PAR OSPF Service Definition.

PAR Service IG	768
Scope	96
ATM addr.	0x39.75.01....00
IPv4 Service Definition	784
IP addr.	192.168.1.1
IP mask	255.255.255.0
Service mask	OSPF
OSPF Service Definition	800
Area ID	0.0.0.0
Router priority	10
Interface type	point-to-multipoint

3.3 Proxy-PAR

Inside routers, it often is neither desirable nor feasible to implement a full PNNI stack. In this case, a lightweight protocol called Proxy-PAR, briefly introduced in Section 2.6, is used to register and query for services.

Proxy-PAR is a client-server protocol in which a Proxy-PAR Client (PPC) runs inside the router, and a Proxy-PAR Server (PPS) inside the switch to which the router is directly attached. The reserved VPI:VCI (0:18) is used for the client-server communication. This protocol is actually composed of three different protocols, namely the *Proxy-PAR Hello Protocol*, the *Proxy-PAR Registration Protocol*, and the *Proxy-PAR Query Protocol*. The Hello protocol is similar to the PNNI Hello protocol, and monitors the state of the connection between the client and the server. It is also used by the server to inform the client of the Registration Expiration Interval, which is the lifetime assigned to information registered by the client into the PNNI database. The Proxy-PAR Registration Protocol allows the client to send the information it wants the server to register on its behalf in the PNNI database. Finally, the Proxy-PAR Query Protocol allows the client to obtain information extracted by the server from the PNNI database. The Registration and the Query protocols guarantee reliable transfers between the client and the server.

The protocol state machines of these three protocols are described in details in [8].

Proxy-PAR clients use the VPN ID IGs, IPv4 Service Definition IGs, and the various specific Service IGs nested within (OSPF, BGP4, etc) when they register the services they support to their server. This facilitates the parsing at the server, as the same format is used to store this information in the PNNI database. Likewise, the server uses the same IGs in any response to queries

from the client.

For their queries, clients use VPN ID IGs and IPv4 Service Definition IGs. In a query, the IP address and subnet mask of the IPv4 Service Definition IG are set to the subnet for which the client is querying. This can be the same subnet as the client's own IP interface or any other subnet. A special value of 0.0.0.0 in the IP address indicates that the client queries for services located in any subnet. The client also sets the appropriate bits in the Service Mask of the IPv4 Service Definition IG to obtain information about the corresponding services.

The formats of the packets that contain these IGs can be found in [8].

Figure 5 shows the same scenario as the example in Figure 4, but now using Proxy-PAR in the routers and the attached switches.

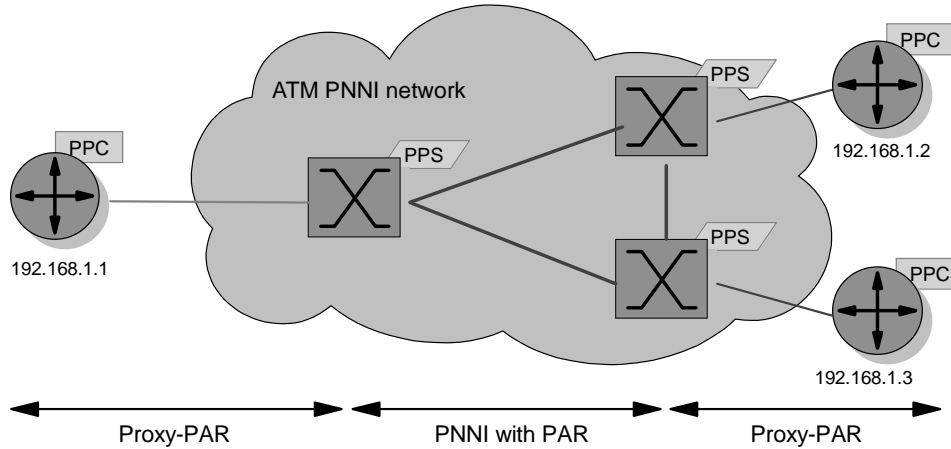


Fig. 5. Proxy-PAR example.

Note that the Proxy-PAR protocol is completely client-driven. In the case of registrations, it is the responsibility of the client to periodically re-register its services before they expire in the PNNI database of the server. Also, the client has to query regularly for the services it is interested in, which makes the protocol follow a *pull* rather than a *push* approach. The server does not keep any state of the information sent in previous queries: there is no *memory* in the server, which prevents incremental updates to the client. This has been a major protocol design decision in order to keep the Proxy-PAR protocol as simple as possible.

Extending PAR to support protocols other than IPv4 would only require the definition of one or more new Service Definition IGs. As PAR switches only need to understand the PAR Service IG to perform the correct flooding across peer groups, they require no change. Only Proxy-PAR servers and clients need to be extended with the new IG so as to process registrations and queries

correctly.

4 PAR and Proxy-PAR benefits in IP over ATM

This section provides a short overview of today's IP over ATM technologies, and what PAR and Proxy-PAR can bring in each case. Practical scenarios showing the deployment of some of these technologies are given in Section 5.

4.1 *ILMI-based Server Discovery*

ILMI-based Server Discovery [1,14–16] has recently been proposed as a mechanism to be used by an ATM-attached host to find the ATM address of the server it needs to contact, for instance an ATM ARP server. Unlike Proxy-PAR, ILMI-based server discovery can only be used to retrieve information, but not to store information on the switch where the host is connected. Thus, the addresses of the servers have to be stored on the switches using a network-management tool. Also, the host must specify explicitly which services it is looking for, and no support for VPNs is provided.

The ILMI-based server discovery mechanism can directly benefit from PAR as a means to distribute its information through the PNNI network. Instead of having to manually configure this information inside each ATM switch, it can be entered in a single switch as PAR data. This information will then be flooded to the other switches in the network. The ILMI 4.0 Service Registry MIB (Management Information Base) in each switch can then interface with PAR to retrieve the appropriate information needed by the hosts. This solution offers a considerable gain in speed and ease of network configuration.

4.2 *Classical IP and ARP over ATM*

Classical IP and ARP over ATM (CLIP) [23] is a well-established protocol that implements the IP address resolution service over an ATM network. It mimics the ARP service [27] with a centralized ATM ARP server. This server provides the IP to ATM address mappings of the hosts or routers in the Logical IP Subnet (LIS) it administrates. Each host in the LIS has to be configured with the address of the ATM ARP server present in the same LIS (in the case of SVCs). Packets destined to a host in another LIS have to be routed, even if direct connectivity is available at the ATM layer. Also, unlike LAN Emulation, CLIP requires that each host be directly connected to the ATM network.

PAR provides a very attractive feature in the context of CLIP: the configuration phase of the ATM ARP clients (the hosts) can be fully automated: ATM ARP servers first need to register via Proxy-PAR on their switch (this can also be done manually); then PAR distributes this information across the PNNI network to all ATM switches. Finally, the ATM ARP clients retrieve the address of their ATM ARP server automatically from their switch using either ILMI-based server discovery or Proxy-PAR.

Note that it is technically feasible to use Proxy-PAR and PAR as substitutes of CLIP. Although Proxy-PAR and PAR are designed for service discovery rather than for plain large-scale host address resolution, it appears that in some cases, the distributed characteristics of PAR compared to the centralized ATM ARP server can be a strategic advantage.

Support for multicast is provided by MARS (Multicast Address Resolution Server) [10], which can benefit from PAR in a very similar way as CLIP (see Section 5.1).

4.3 Next-Hop Resolution Protocol

The Next-Hop Resolution Protocol (NHRP) [24] allows the establishment of direct ATM connections between source and destination even when they are not located in the same LIS, therefore solving a major shortcoming of CLIP. Requests for shortcuts issued by NHCs (NHRP Clients) are routed hop-by-hop by NHSes (NHRP Servers) between LISes towards the NHS serving the destination LIS, where the ATM address of the destination is then retrieved and returned to the source to allow the setting up of a direct connection.

As in the CLIP case, PAR can be advantageously used to propagate configuration information of the IP and ATM addresses of the NHSes across the PNNI network: NHCs will learn which is their serving NHS.

Compared to the CLIP case, the advantage of using Proxy-PAR in the case of the NHRP, particularly for the NHSes, becomes much clearer: the NHSes can register themselves and automatically learn about neighboring NHSes going up or down. Otherwise manual configuration in this environment becomes very cumbersome.

Even though PAR could technically provide the address resolution functionality itself as well, as in the CLIP case, PAR and Proxy-PAR are not designed to replace all NHRP functionalities. NHRP can for instance require specific conditions to be met on the traffic characteristics before a direct shortcut is provided.

4.4 LAN Emulation and Multi-Protocol over ATM

LAN Emulation (LANE) [3–5] emulates an IEEE 802.3 (Ethernet) or IEEE 802.5 (token-ring) local-area network over ATM. This emulated local-area network is called Emulated LAN (ELAN). The main element is an address resolution service that maps a MAC address to an ATM address. Similar to CLIP, address resolution is done using a server called LANE Server (LES). Broadcasting and multicasting are emulated on the Broadcast and Unknown Server (BUS), which connects to all LANE Clients. Each LANE client can connect to the LECS (LANE configuration server) in order to receive appropriate configuration information.

Multiprotocol over ATM (MPOA) [6] offers to ELANs what NHRP offers to LISes, namely the ability to open shortcut connections across ELANs. MPOA puts the router concept, NHRP and LANE v2.0, into the same framework. In this framework, the NHS is a component of the MPOA router. Similar to LANE, the NHCs and the NHSes use the LECS to retrieve configuration information.

PAR provides no support for either LANE or MPOA, as configuration is solved in these cases by other means: the LECS ATM address can be manually configured in the LANE Client or ILMI can be used to discover this address, or a well-known VPI:VCI (0:17) can be used to reach the LECS. It is also possible to configure the address of the LECS or LES as a well-known anycast address. In the latter case no prior configuration of the LANE Client is required.

In certain cases, such as an ELAN interconnecting routers, it is interesting to replace LANE with PAR and Proxy-PAR. Such an example is described in more detail in Section 5.1.

4.5 Multi-Protocol Label Switching

Multi-Protocol Label Switching (MPLS) [32] is gaining momentum as a new technology for use in backbone environments, originally because of the high data-forwarding rates it allows. New gigabit routers now question the need for high-speed label-swapping technologies such as MPLS.

But besides throughput, MPLS offers traffic engineering features for IP networks that may prove to be very valuable in dense Internet Service Providers (ISPs) environments. In PNNI, automated traffic engineering is available in the form of soft-PVCs (Permanent VCs) and QoS-sensitive hierarchical routing. In terms of scalability, one advantage of MPLS over PNNI is the merging capability: in the case of routers interconnected through an ATM backbone,

MPLS requires fewer Virtual Circuits (labels) to be set up as it can merge LSPs (label-switched paths) at any given node in the network if both LSPs have the same destination and only require best-effort QoS. ATM does not allow this, not even for best-effort connections.

With a network running PNNI and PAR, IP and other network protocols (like IPv6) can coexist in a “ships in the night” fashion. These network protocols then benefit “for free” from all the automated traffic engineering capabilities of PNNI. At the same time, the network supports full service integration, for instance data and voice traffic.

PAR and Proxy-PAR only work together with PNNI, thus MPLS cannot benefit from the auto-configuration features they offer. Instead, MPLS uses its own mechanisms to create adjacencies between neighbor MPLS routers.

Generally, the choice between ATM-PNNI and IP technologies boils down to following aspect: while IP does not yet offer the same sophisticated QoS support that ATM does, it has the advantage of being the de facto interconnection protocol at a worldwide level. PAR and Proxy-PAR significantly ease the interoperability of IP and ATM-PNNI networks, and thus make the choice between IP/MPLS and IP/ATM-PNNI based on specific technical needs rather than interoperability issues.

5 PAR and Proxy-PAR deployment scenarios

Having described the principles of the main IP over ATM technologies used today as well as how PAR and Proxy-PAR could help in terms of auto-configuration, we now illustrate in more detail how this is realized in practice. The first example is the case of a campus network running PNNI, where IP is supported through Classical IP over ATM for unicast and through MARS for multicast traffic. As a second example, we take the case of a backbone ATM network interconnecting IP routers. The final example stresses the auto-configuration feature even more strongly in the case of a mobile IP network.

5.1 PAR and Proxy-PAR in the campus network

In terms of network architecture, the campus environment is relatively homogeneous and under the control of a single authority, so that usually there is no policy barrier against creating shortcuts between LISes within the network. The campus network is logically partitioned so that users with common interests (servers, printers, etc.) are assigned to the same LIS. Thus a higher

proportion of switched traffic (that remains in the same LIS) can be retained compared to routed traffic (which originates and terminates in different LISes), in order to have a more efficient network. The exact proportion strongly depends on administrative policies of how users are assigned to LISes. CLIP covers the most frequent cases to efficiently support IP traffic when the underlying network is ATM. For larger campus-network sizes (of more than 1000 hosts), with large numbers of LISes, there is a need for NHRP to allow shortcuts between LISes, and therefore avoid costly router hops and degradation in the QoS such a routing usually induces.

In both cases, PAR can be used to avoid manual configuration of ATM ARP or NHRP servers. Moreover, PNNI allows a better routing of NHRP shortcuts through the ATM network (than static routing tables do), and scoping can be used to limit the flooding of PAR PTSEs up to a certain PNNI hierarchy level.

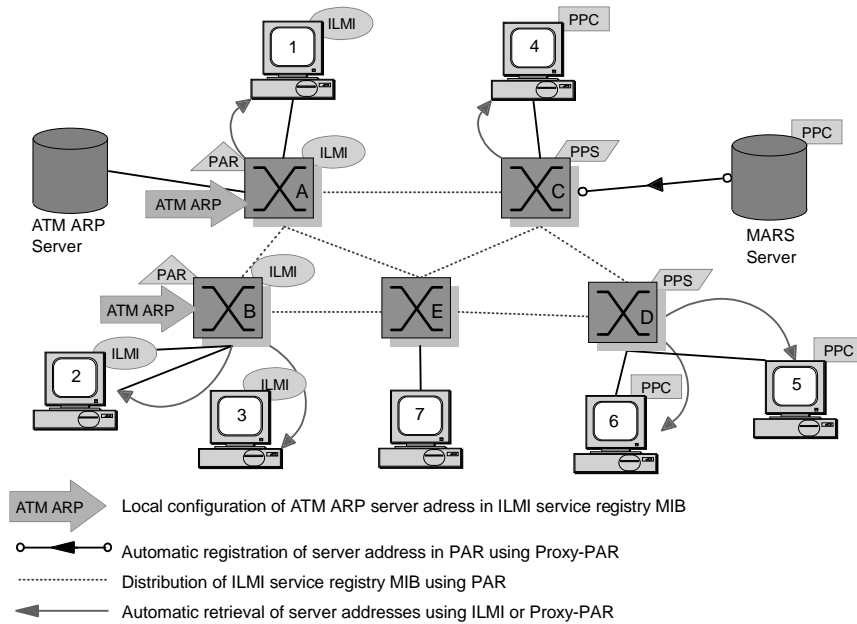


Fig. 6. Automatic configuration using ILMI-based server discovery and Proxy-PAR.

An example of a CLIP network is shown in Figure 6. This network consists of five ATM switches labeled A to E, an ATM ARP and a MARS server, and seven CLIP clients belonging to the same LIS. In this example, we illustrate all possible methods for obtaining the ATM ARP or MARS servers' addresses.

The ATM ARP server address is configured in the ILMI service registry MIB on switches A and B, allowing only the clients attached directly (clients 1, 2 and 3) to retrieve the ATM ARP server's address using ILMI. On all other clients, the ATM ARP server address needs to be configured manually.

A fully automatic configuration is achieved using the MARS server. It is

equipped with Proxy-PAR, which is used to automatically register the MARS server address on the local Proxy-PAR server on switch C. This address is then distributed in the network using PAR. Clients 4 to 6 fetch this information using Proxy-PAR. The PAR-enabled switches A and B store the MARS address in the ILMI registry MIB, where it is fetched by clients 1 to 3.

Client 7, which is equipped with neither Proxy-PAR nor ILMI, needs to be configured manually with both server addresses.

As the example shows, MARS configuration can be done completely automatically if the servers are equipped with Proxy-PAR. PAR can also be used in conjunction with ILMI, such that the clients do not need to be equipped with Proxy-PAR.

In campus networks where multiple LISes or ELANs need to be interconnected, often a backbone LIS or ELAN is created. This backbone ELAN interconnects the routers from each ELAN, as shown in Figure 7 for the LANE case only. In this solution, however, robustness and availability of the entire backbone mainly depend on the correct operation of the backbone's LANE servers (LECS, LES and BUS).

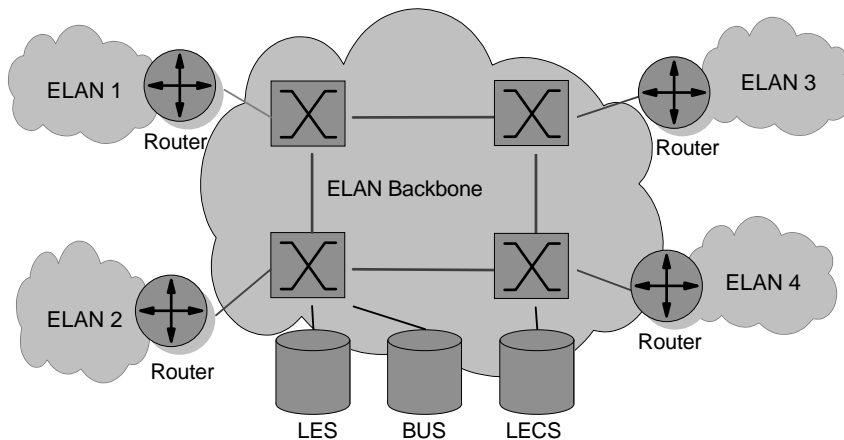


Fig. 7. ATM LAN Emulation.

To increase the robustness of the backbone LIS, Proxy-PAR can be used as shown in Figure 8. Each router acts as a Proxy-PAR client and registers the service it provides with Proxy-PAR, for instance OSPF, its IP address and its ATM address. This information on the available services is then distributed over the ATM backbone, and to each router with Proxy-PAR. Each router uses this information to establish ATM SVCs with the other routers. The Proxy-PAR solution is ideally suited for an ATM-based router backbone, as there no longer is a single point of failure. Moreover, as there no longer is a LANE server, routers do not need to be configured with the LECS address.

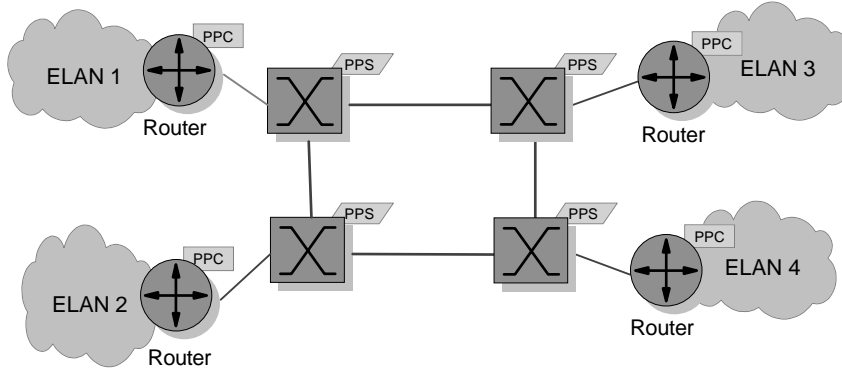


Fig. 8. PAR and Proxy-PAR in an ELAN environment.

5.2 PAR and Proxy-PAR in the backbone network

Backbone networks interconnect networks of different organizations in a scalable way. Backbone networks therefore are most often built in a hierarchical manner. Also, backbones support basic means to build virtual private networks for carrying traffic of possibly competing companies. PAR supports the configuration and operation of backbone networks in two ways. Firstly, it simplifies the configuration of routers that are interconnected through a hierarchical ATM backbone. Secondly, it has built-in support for VPNs.

Figure 9 shows a hierarchical backbone network. The network consists of an ATM backbone that interconnects a number of IP routers. The switches in the ATM network are arranged in a two-layer hierarchy. Two peer groups at level 80 form the lower layer. They are interconnected with a peer group at level 60. Switches that connect to clients are equipped with Proxy-PAR and PAR. Those switches that do not connect to clients but can become PGLs at least need to be equipped with PAR. The IP hierarchy consists of two Autonomous Systems (AS) running OSPF and interconnected with BGP. Routers R2 and R3 run both OSPF and BGP. The IP hierarchy is in alignment with the ATM hierarchy: IP routers of an AS connect to ATM switches of the corresponding peer group. All routers use Proxy-PAR to register their OSPF service capabilities at the corresponding Proxy-PAR server. The OSPF services are registered with local scope, i.e., with PNNI level 80. Consequently, the service announcements are distributed within the local peer group only. Because of the alignment of the ATM and the IP hierarchies, only routers within the same AS receive each other's OSPF announcements. Routers R2 and R3 in addition register their BGP service capabilities with scope "one above local", which the Proxy-PAR server translates into PNNI level 60. Hence, router R2's BGP service announcements are flooded throughout the entire network and eventually reach R3 (and vice versa), allowing the two routers to form a BGP adjacency.

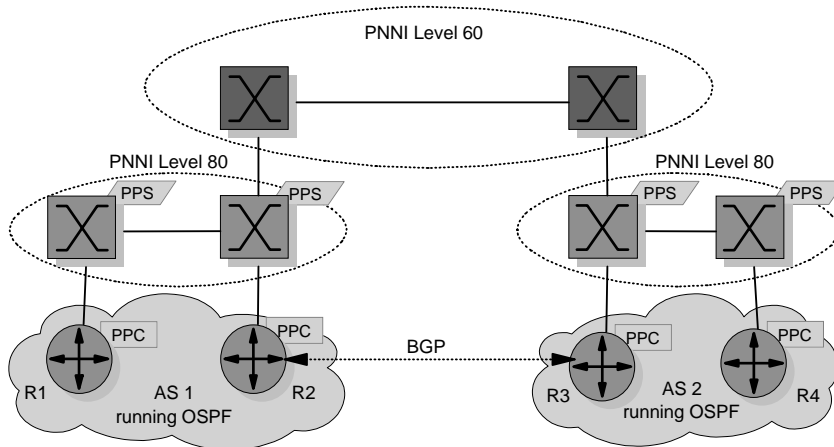


Fig. 9. Auto-configuration of a hierarchical backbone network with PAR.

Often backbone networks carry traffic of competing companies. Therefore, separating the traffic of different customers as a basic means to build virtual private networks is a must. The implementation of VPNs on top of ATM can easily be done using Virtual Path Connections (VPCs). For each router pair that should be connected, a VPC needs to be defined. Figure 10 shows an example backbone with three VPNs. Each VPN interconnects three routers using a total of three VPCs. The disadvantage of using VPCs is the high configuration effort, which must be done at each ATM switch as well as at the routers.

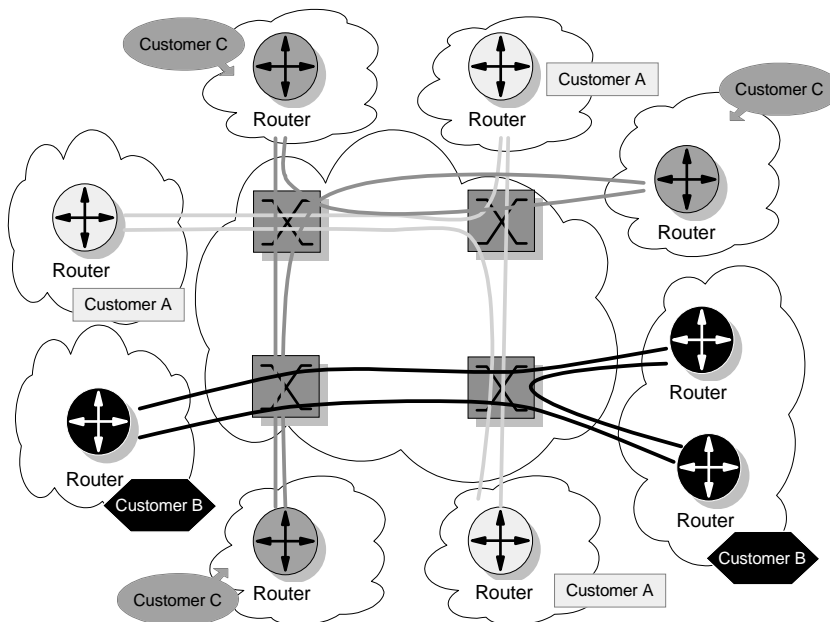


Fig. 10. Building VPNs using ATM PVCs.

PAR and Proxy-PAR allow automatic configuration of the routers with built-

in support for VPNs. The routers, configured as Proxy-PAR clients, register their service (such as OSPF) and ATM addresses with their Proxy-PAR server. Each Proxy-PAR client “tags” the PAR registrations with its VPN ID (using a PAR VPN ID IG). These VPN IDs are checked by the Proxy-PAR server, as client and server in this case belong to different administrative domains. Packets with invalid or unknown VPN IDs are ignored and not distributed by PAR. When the Proxy-PAR client queries the Proxy-PAR server, it puts the same VPN ID into its PAR query. The Proxy-PAR server checks again whether the client is allowed to query for this VPN. If not, an empty response is sent. Thus, routers inside a VPN will only receive service and address information of routers inside their same VPN. This allows, for instance, the same IP address space to be reused across different VPNs without generating a conflict.

Figure 11 shows an example of PAR VPN support with two customers sharing a common backbone network. Customer A and B configure their routers with VPN ID “A” and “B”, respectively. Also, the service provider configures VPN ID filters on the ports that lead to customer equipment. These filters, indicated by smaller grey circles, remove invalid and unknown PAR IGs (PAR VPN ID IGs and/or PAR IPv4 Service Definition IGs) from incoming Proxy-PAR registration packets. Valid service advertisements are flooded throughout the PNNI network. Filtering is again applied when the PAR service advertisements are forwarded to the customer equipment.

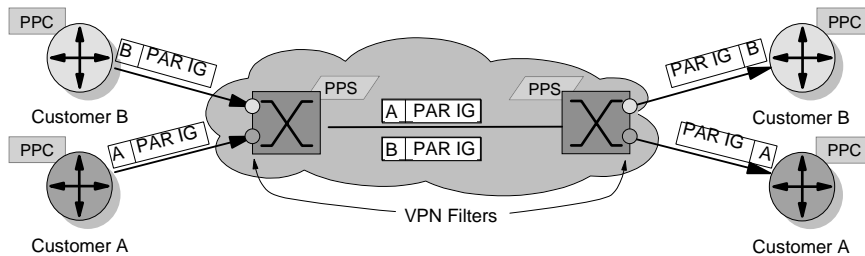


Fig. 11. Filtering of PAR IGs based on VPN IDs.

5.3 PAR and Proxy-PAR in the mobile network

PAR and Proxy-PAR are of great help when it comes to auto-configuration of IP routers running in a mobile network. While they are roaming, the IP routers automatically learn who the new neighbors are and with which they should form a peer relationship.

Figure 12 shows a fixed network composed of ATM switches running PNNI and IP routers, all equipped with PAR. A mobile-ATM network [7], shown with a single ATM switch and an IP router, changes its access point to the fixed network while roaming. Once the mobile-ATM network has moved to

the new access point, the IP router learns through Proxy-PAR which services are available in this new location. It then has the choice to establish SVCs to some or all of the neighbors it discovers through Proxy-PAR (this is shown by a dotted line). Without Proxy-PAR, the alternative is to manually look up which services are available at the new access point, followed by the appropriate manual reconfiguration of the router.

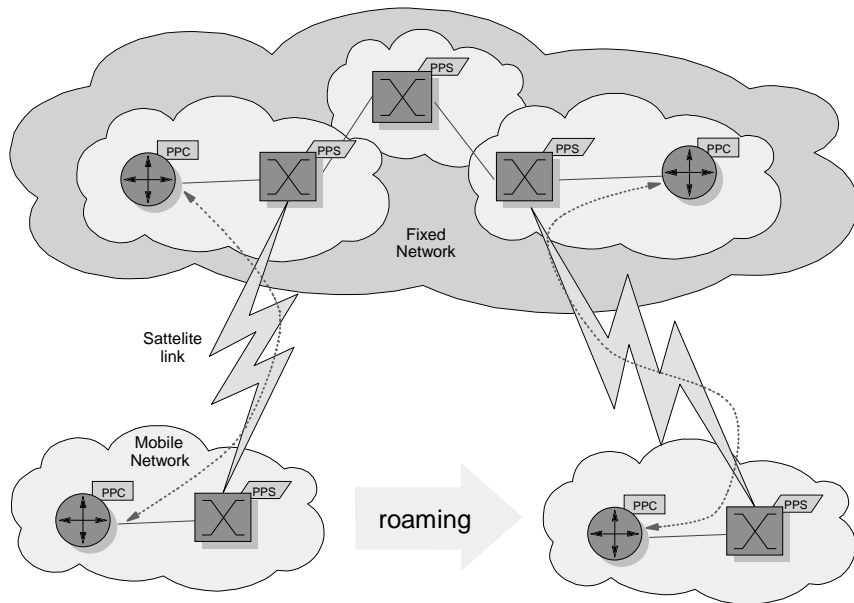


Fig. 12. Auto-configuration of a roaming mobile network with PAR.

The appropriate use of scope is determinant in this case, as it is in the example presented in Figure 9. Routers of the fixed network attached to the access points are visible to roaming routers only if they become physically close. This allows one to restrict with which neighbors a router is able to establish adjacencies.

6 Comparison and related work

Service discovery is more general than address resolution. For instance, routers first need to find their neighbors, and then need to obtain the IP-to-ATM address mapping for each neighbor. Service discovery fulfills both needs simultaneously: the neighbors, together with their IP address and the corresponding ATM address, are returned to the requesting client. This eliminates the need for a unicast or multicast address resolution mechanism. Table 9 summarizes the benefits of PAR and Proxy-PAR.

It is interesting to note how *multicast* can be used to perform service discovery, similarly to PAR or ILMI-based Server Discovery. In the example of Figure 7,

Table 9
Comparison of service discovery methods

	PAR+Proxy-PAR	PAR+ILMI	ILMI-based Serv. Disc.	IP Multicast
configuration	automatic	automatic	manually per switch	automatic
prerequisites	PNNI	PNNI	-	MARS or LANE
robustness	distributed	distributed	N/A	centralized servers
IP services supported	see Table 5	ARP, MARS, NHRP	ARP, MARS, NHRP	see [31,22]
registration and query	both	only query	only query	N/A
VPN support	yes	no	no	yes

OSPF routers use a well-known multicast address to send Hello packets to their neighbors. Likewise, RIP routers [25] use a well-known multicast address to send requests to all their neighbors. RIP and OSPF routers therefore do not have to be configured with the IP addresses of each of their neighbors, these addresses will be learned during protocol execution. Multicast is supported by MARS or LANE, and we have shown how PAR automates the configuration of the MARS Server in each MARS Client.

An alternative is to run OSPF or RIP in unicast mode only (or NBMA, short for Non-Broadcast Multiple Access), and configure the neighbors manually. While MARS or LANE will no longer be needed in this case, PAR can still be used to avoid manual configuration of the OSPF or RIP neighbors, as shown for OSPF in Figure 8.

We have not included the Dynamic Host Configuration Protocol (DHCP) [17] in our comparison because this protocol is destined specifically for the automatic configuration of *hosts*' IP stacks. PAR and Proxy-PAR allow automatic configuration of routers as well. Other service discovery protocols or, more generally, directory services, such as the Service Location Protocol (SLP) [21], DNS SRV [20], or the Lightweight Directory Access Protocol (LDAP) [33], stand at a higher layer than PAR and Proxy-PAR: these protocols require the network routing to be operational before they can execute and provide location of user services. PAR and Proxy-PAR therefore act as a bootstrap for all these additional mechanisms.

7 Conclusion and future work

In this paper, we have presented and discussed various IP over ATM technologies and some of their deficiencies, including the manual configuration of server addresses and lack of robustness of centralized servers solutions. The PAR extensions to PNNI, which allow IP information to be distributed across an ATM network by PNNI, are shown to be an important mechanism towards

solving these problems, when combined with a protocol such as ILMI-based Server Discovery or Proxy-PAR. This allows various kind of services, such as OSPF routers or even DNS servers, to be advertised across an ATM network, to the attached IP hosts or IP routers.

The use of PAR and Proxy-PAR has been shown to be the least intrusive approach for integrating IP and ATM, where only information about the availability of IP services is advertised by PNNI. That is why we call this “service integration”. Integration does not take place, for instance, at the routing level: the overlay approach is conserved (for instance, OSPF runs on top of PNNI) unlike in Integrated-PNNI, where IP and ATM routing are integrated.

Routers therefore need not necessarily run PNNI themselves, but only the client side of a simple exchange protocol (Proxy-PAR). This allows them to both register and query for other neighbor routers, unlike other ILMI-based solutions that only allow queries to be performed.

PAR and Proxy-PAR support VPNs by filtering, on each switch port, which services are being advertised to the attached client. It is then possible to safely reuse the same IP address space across different VPNs.

Because PAR and Proxy-PAR only have a minimal impact on the routers, it can be viewed as a very *IP-friendly* solution.

Once an ATM-PNNI network is equipped with PAR together with Proxy-PAR, it becomes straightforward to support new services, such as IPv6. PAR and Proxy-PAR therefore allow the rapid deployment of new protocols over an ATM-PNNI network.

The PAR extensions to PNNI were implemented in the IBM ATM-PNNI switch stack, and the Proxy-PAR protocol in both the switch and the IBM router stacks. In the router stack, we have concentrated mainly on the OSPF protocol, and our testbed demonstrated automatic configuration of OSPF routers over an NBMA (ATM SVCs) network, in a VPN environment.

In certain situations, the creation of a full mesh of VCs between all neighbor routers visible through Proxy-PAR is inefficient. Also, centralized server solutions, such as CLIP or MARS, are not suitable in the context of mobile ad-hoc networks [13]. Future work aims at extending Proxy-PAR to enable routers to make smarter choices when deciding with which neighbors they should peer, and using PAR and Proxy-PAR in mobile ad-hoc networks.

References

- [1] ATM Forum, “*Integrated Local Management Interface (ILMI) Specification Version 4.0*”, af-ilmi-0065.000, September 1996.
- [2] ATM Forum, “*Issues and Approaches for Integrated PNNI*”, atm96-0355, April 1996.
- [3] ATM Forum, “*LAN Emulation over ATM 1.0*”, af-lane-0021.000, Jan 1995.
- [4] ATM Forum, “*LANE v2.0 LUNI Interface*”, af-lane-0084.000, July 1997.
- [5] ATM Forum, “*LAN Emulation over ATM Version 2 - LNNI Specification*”, af-lane-0112.000, Feb. 1999.
- [6] ATM Forum, “*Multi-Protocol Over ATM Specification v1.0*”, af-mpoa-0087.000, July 1997.
- [7] ATM Forum, “*PNNI Addendum for Mobility Extension Version 1.0*”, af-ra-0123.000, April 1999.
- [8] ATM Forum, “*PNNI Augmented Routing (PAR) Version 1.0*”, af-ra-0104, January 1999.
- [9] ATM Forum, “*P-NNI V1.0*”, af-pnni-0055.000, March 1996.
- [10] G. Armitage, “*Support for Multicast over UNI 3.0/3.1 based ATM Networks*”, Internet RFC 2022, November 1996.
- [11] T. Bates and R. Chandrasekeran, “*BGP Route Reflection: An alternative to full mesh IBGP*”, Internet RFC 1966, June 1996.
- [12] R. Callon, “*Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*”, Internet RFC 1195, December 1990.
- [13] S. Corson and J. Macker, “*Mobile Ad hoc Networks (MANET): Routing Protocol Performance Issues and Evaluation Considerations*”, Internet RFC 2501, January 1999.
- [14] M. Davison, “*ILMI-Based Server Discovery for ATMARP*”, Internet RFC 2601, June 1999.
- [15] M. Davison, “*ILMI-Based Server Discovery for MARS*”, Internet RFC 2602, June 1999.
- [16] M. Davison, “*ILMI-Based Server Discovery for NHRP*”, Internet RFC 2603, June 1999.
- [17] R. Droms, “*Dynamic Host Configuration Protocol*”, Internet RFC 2131, March 1997.
- [18] P. Droz and T. Przygienda, “*Proxy-PAR*”, Internet RFC 2843, May 2000.

- [19] B. Fox and B. Gleeson, “*Virtual Private Networks Identifier*”, Internet RFC 2685, September 1999.
- [20] A. Gulbrandsen, P. Vixie and L. Esibov, “*A DNS RR for specifying the location of services (DNS SRV)*”, Internet RFC 2782, February 2000.
- [21] E. Guttman, C. Perkins, J. Veizades and M. Day, “*Service Location Protocol, Version 2*”, Internet RFC 2608, June 1999.
- [22] R. Hinden and S. Deering, “*IPv6 Multicast Address Assignments*”, Internet RFC 2375, July 1998.
- [23] M. Laubach and J. Halpern, “*Classical IP and ARP over ATM*”, Internet RFC 2225, April 1998.
- [24] J. Luciani, D. Katz, D. Piscitello, B. Cole and N. Doraswamy, “*NBMA Next Hop Resolution Protocol (NHRP)*”, Internet RFC 2332, April 1998.
- [25] G. Malkin, “*RIP Version 2*”, Internet RFC 2453, November 1998.
- [26] J. Moy, “*OSPF Version 2*”, Internet RFC 2178, July 1997.
- [27] D. Plummer, “*An Ethernet Address Resolution Protocol - or - Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware*”, Internet RFC 826, November 1982.
- [28] T. Przygienda, P. Droz and R. Haas, “*OSPF over ATM and Proxy-PAR*”, Internet RFC 2844, May 2000.
- [29] T. Przygienda, P. Droz and C. West, “*Proxy PNNI Augmented Routing (Proxy PAR)*”, ICATM 98 (1998 IEEE International Conference on ATM) Colmar, France, June 1998.
- [30] Y. Rekhter and T. Li, “*A Border Gateway Protocol 4 (BGP-4)*”, Internet RFC 1771, March 1995.
- [31] J. Reynolds and J. Postel, “*Assigned Numbers*”, Internet RFC 1700 (current IPv4 multicast addresses assignment: <ftp://venera.isi.edu/in-notes/iana/assignments/multicast-addresses>), October 1994.
- [32] E. Rosen, A.Viswanathan and R. Callon, “*Multiprotocol Label Switching Architecture*”, Internet Draft, work in progress, draft-ietf-mpls-arch-06.txt, August 1999.
- [33] M. Wahl, T. Howes and S. Kille, “*Lightweight Directory Access Protocol (v3)*”, Internet RFC 2251, December 1997.