OSPF over ATM and Proxy-PAR

Status of this Memo

Copyright Notice

Abstract

   This memo specifies, for OSPF implementors and users, mechanisms
   describing how the protocol operates in ATM networks over PVC and SVC
   meshes with the presence of Proxy-PAR. These recommendations require
   no protocol changes and allow simpler, more efficient and cost-
   effective network designs. It is recommended that OSPF
   implementations should be able to support logical interfaces, each
   consisting of one or more virtual circuits and used either as
   numbered logical point-to-point links (one VC), logical NBMA networks
   (more than one VC) or Point-to-MultiPoint networks (more than one
   VC), where a solution simulating broadcast interfaces is not
   appropriate. PAR can help distribute across the ATM cloud
   configuration setup and changes of such interfaces when OSPF capable
   routers are (re-)configured. Proxy-PAR can in turn be used to
   exchange this information between the ATM cloud and the routers
   connected to it.

1 Introduction

   Proxy-PAR and PAR have been accepted as standards by the ATM Forum in
   January 1999 [1]. A more complete overview of Proxy-PAR than in the
   section below is given in [2].

1.1 Introduction to Proxy-PAR

Proxy-PAR [1] is an extension that allows different ATM attached
devices (like routers) to interact with PAR-capable switches and to
query information about non-ATM services without executing PAR
themselves. The Proxy-PAR client side in the ATM attached device is
much simpler in terms of implementation complexity and memory
requirements than a complete PAR protocol stack (which includes the
full PNNI [3] protocol stack) and should allow easy implementation,
e.g. in existing IP routers. In addition, clients can use Proxy-PAR
to register the various non-ATM services and protocols they support.
Proxy PAR has consciously been omitted as part of ILMI [4] due 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 that executes Proxy-PAR does not necessarily need to
execute ILMI or UNI signaling, although this normally will be the
case.

The protocol in itself does not specify how the distributed service
registration and data delivered to the client is supposed to drive
other protocols. Hence OSPF routers, for instance, that find
themselves through Proxy-PAR could use this information in a
Classical IP and ARP over ATM [5] fashion, forming a full mesh of
point-to-point connections to interact with each other to simulate
broadcast interfaces. For the same purpose, LANE [6] or MARS [7]
could be used. As a byproduct, Proxy-PAR could provide the ATM
address resolution for IP-attached devices, but such resolution can
be achieved by other protocols under specification at the IETF as
well, e.g. [8]. Last but not least, it should be mentioned here that
the protocol coexists with and complements the ongoing work in IETF
on server detection via ILMI extensions [9,10,11].

1.1.1 Proxy-PAR Scopes

Any information registered through Proxy-PAR is flooded only within a
defined scope that is established during registration and is
equivalent to the PNNI routing level. As no assumption can be made
about the information distributed (e.g. IP addresses bound to NSAPs
are not assumed to be aligned with them in any respect such as
encapsulation or functional mapping), it cannot be summarized. This
makes a careful handling of scopes necessary to preserve the
scalability. More details on the usage of scope can be found in [2].

1.2 **Introduction to OSPF**

   OSPF (Open Shortest Path First) is an Interior Gateway Protocol (IGP)
   and described in [12] from which most of the following paragraphs has
   been taken almost literally. OSPF distributes routing information
   between routers belonging to a single Autonomous System. The OSPF
   protocol is based on link-state or SPF technology. It was developed
   by the OSPF working group of the Internet Engineering Task Force. It
   has been designed expressly for the TCP/IP internet environment,
   including explicit support for IP subnetting, and the tagging of
   externally-derived routing information. OSPF also utilizes IP
   multicast when sending/receiving the updates. In addition, much work
   has been done to produce a protocol that responds quickly to topology
   changes, yet involves small amounts of routing protocol traffic.

   To cope with the needs of NBMA and demand-circuit-capable networks
   such as Frame Relay or X.25, [13] has been made available. It
   standardizes extensions to the protocol that allow efficient
   operation over on-demand circuits.

   OSPF supports three types of networks today:

     + Point-to-point networks: A network that joins a single pair of
        routers. Point-to-point networks can either be numbered or
        unnumbered. In the latter case the interfaces do not have IP
        addresses nor masks. Even when numbered, both sides of the link
        do not have to agree on the IP subnet.

     + Broadcast networks: Networks supporting many (more than two)
        attached routers, together with the capability of addressing a
        single physical message to all of the attached routers
        (broadcast). Neighboring routers are discovered dynamically on
        these networks using the OSPF Hello Protocol. The Hello
        Protocol itself takes advantage of the broadcast capability.
        The protocol makes further use of multicast capabilities, if
        they exist. An Ethernet is an example of a broadcast network.

     + Non-broadcast networks: Networks supporting many (more than
        two) attached routers, but having no broadcast capability.
        Neighboring routers are maintained on these nets using OSPF's
        Hello Protocol. However, due to the lack of broadcast
        capability, some configuration information is necessary for the
        correct operation of the Hello Protocol. On these networks,
        OSPF protocol packets that are normally multicast need to be
        sent to each neighboring router, in turn. An X.25 Public Data
        Network (PDN) is an example of a non-broadcast network.

OSPF runs in one of two modes over non-broadcast networks. The
first mode, called non-broadcast multi-access (NBMA), simulates
the operation of OSPF on a broadcast network. The second mode,
called Point-to-MultiPoint, treats the non-broadcast network as
a collection of point-to-point links. Non-broadcast networks
are referred to as NBMA networks or Point-to-MultiPoint
networks, depending on OSPF's mode of operation over the
network.

2 OSPF over ATM

2.1 Model

Contrary to broadcast-simulation-based solutions such as LANE [6] or
Classical IP and ARP over ATM [5], this document elaborates on how to
handle virtual OSPF interfaces over ATM such as NBMA, Point-to-
MultiPoint or point-to-point and allow for their auto-configuration
in the presence of Proxy-PAR. One advantage is the circumvention of
server solutions that often present single points of failure or hold
large amounts of configuration information.

The other main benefit is the capability of executing OSPF on top of
NBMA and Point-to-MultiPoint ATM networks, and still benefit from the
automatic discovery of OSPF neighbors. As opposed to broadcast
networks, broadcast-simulation-based networks (such as LANE or
Classical IP and ARP over ATM), and point-to-point networks, where an
OSPF router dynamically discovers its neighbors by sending Hello
packets to the All-SPFRouters multicast address, this is not the case
on NBMA and Point-to-MultiPoint networks. On NBMA networks, the list
of all other attached routers to the same NBMA network has to be
manually configured or discovered by some other means: Proxy-PAR
allows this configuration to be automated. Also on Point-to-
MultiPoint networks, the set of routers that are directly reachable
can either be manually configured or dynamically discovered by
Proxy-PAR or mechanisms such as Inverse ATMARP. In an ATM network,
(see 8.2 in [5]) Inverse ATMARP can be used to discover the IP
address of the router at the remote end of a given PVC, whether or
not its ATM address is known. But Inverse ATMARP does not return, for
instance, whether the remote router is running OSPF, unlike Proxy-
PAR.

Parallel to [14], which describes the recommended operation of OSPF
over Frame Relay networks, a similar model is assumed where the
underlying ATM network can be used to model single VCs as point-to-
point interfaces or collections of VCs as non-broadcast interfaces,
whether in NBMA or Point-to-MultiPoint mode. Such a VC or collection
of VCs is called a logical interface and specified through its type
(either point-to-point, NBMA or Point-to-MultiPoint), VPN ID (the

Virtual Private Network to which the interface belongs), address and
mask. Layer 2 specific configurations such as the address resolution
method, class and quality of service of circuits used, and others,
must also be included. As a logical consequence thereof, a single,
physical interface could encompass multiple IP subnets or even
multiple VPNs. Contrary to layer 2 and IP addressing information,
when running Proxy-PAR, most of the OSPF information needed to
operate such a logical interface does not have to be configured into
routers statically but can be provided through Proxy-PAR queries.
This allows much more dynamic configuration of VC meshes in OSPF
environments than, for example, Frame Relay solutions do.

Proxy-PAR queries can also be issued with a subnet address set to
0.0.0.0, instead of a specific subnet address. This type of query
returns information on all OSPF routers available in all subnets
within the scope specified in the query. This can be used for
instance when the IP addressing information has not been configured.

2.2 Configuration of OSPF interfaces with Proxy-PAR

To achieve the goal of simplification of VC mesh reconfiguration,
Proxy-PAR allows the router to learn automatically most of the
configuration that has to be provided to OSPF. Non-broadcast and
point-to-point interface information can be learned across an ATM
cloud as described in the ongoing sections. It is up to the
implementation to possibly allow for a mixture of Proxy-PAR
autoconfiguration and manual configuration of neighbor information.
Moreover, manual configuration could, for instance, override or
complement information derived from a Proxy-PAR client. In addition,
OSPF extensions to handle on-demand circuits [13] can be used to
allow the graceful tearing down of VCs not carrying any OSPF traffic
over prolonged periods of time. The various interactions are
described in sections 2.2.1, 2.2.2 and 2.2.3.

Even after autoconfiguration of interfaces has been provided, the
problem of VC setups in an ATM network is unsolved because none of
the normally used mechanisms such as Classical IP and ARP over ATM
[5] or LANE [6] are assumed to be present. Section 2.5 describes the
behavior of OSPF routers necessary to allow for router connectivity.

2.2.1 Autoconfiguration of Non-Broadcast Multiple-Access (NMBA)
Interfaces

Proxy-PAR allows the autoconfiguation of the list of all routers
residing on the same IP network in the same VPN by simply querying
the Proxy-PAR server. Each router can easily obtain the list of all
OSPF routers on the same subnet with their router priorities and
corresponding ATM addresses. This is the precondition for OSPF to

work properly across such logical NBMA interfaces. Note that this
member list, when learned through Proxy-PAR queries, can dynamically
change with PNNI (in)stability and general ATM network behavior.
Relying on an OSPF mechanism to discover a lack of reachability in
the overlaying logical IP network could alleviate the risk of
thrashing DR elections and excessive information flooding. Once the
DR election has been completed and the router has not been elected DR
or BDR, an implementation of [13] can ignore the fact that all
routers on the specific NBMA subnet are available in its
configuration because it only needs to maintain VCs to the DR and
BDR. Note that this information can serve other purposes, such as the
forwarding of data packets (see section 2.4).

Traditionally, router configuration for a NBMA network provides the
list of all neighboring routers to allow for proper protocol
operation. For stability purposes, the user may choose to provide a
list of neighbors through such static means but also enable the
operation of Proxy-PAR protocol to complete the list. It is left up
to specific router implementations to determine whether to use the
manual configuration in addition to the information provided by
Proxy-PAR, to use the manual configuration to filter dynamic
information, or whether a concurrent mode of operation is prohibited.
In any case it should be obvious that allowing for more flexibility
may facilitate operation but provides more possibilities for
misconfiguration as well.

2.2.2 Autoconfiguration of Point-to-MultiPoint Interfaces

Point-to-MultiPoint interfaces in ATM networks only make sense if no
VCs can be set up dynamically because an SVC-capable ATM network
normally presents a NBMA cloud to OSPF. This is for example the case
if OSPF executes over a network composed of a partial PVC or SPVC
mesh or predetermined SVC meshes. Such a network could be modeled
using the Point-to-MultiPoint OSPF interface and the neighbor
detection could be provided by Proxy-PAR or other means. In the
Proxy-PAR case the router queries for all OSPF routers on the same
network in the same VPN but it installs in the interface
configuration only routers that are already reachable through
existing PVCs. The underlying assumption is that a router knows the
remote ATM address of a PVC and can compare it with appropriate
Proxy-PAR registrations. If the remote ATM address of the PVC is
unknown, it can be discovered by such mechanisms as Inverse ARP [15].

Proxy-PAR provides a true OSPF neighbor detection mechanism, whereas a mechanism like Inverse ARP only returns addresses of directly reachable routers (which are not necessarily running OSPF), in the Point-to-Multi-Point environment.

2.2.3 Autoconfiguration of Numbered Point-to-Point Interfaces

OSPF point-to-point links do not necessarily have an IP address assigned and even if they do, the mask is undefined. As a precondition to successfully register a service with Proxy-PAR, an IP address and a mask are required. Therefore, if a router desires to use Proxy-PAR to advertise the local end of a point-to-point link to the router with which it intends to form an adjacency, an IP address has to be provided as well as a netmask set or a default of 255.255.255.252 (this gives as the default case a subnet with two routers on it) assumed. To allow the discovery of the remote end of the interface, IP address of the remote side has to be provided and a netmask set or a default of 255.255.255.252 assumed. Obviously the discovery can only be successful when both sides of the interface are configured with the same network mask and are within the same IP network. The situation where more than two possible neighbors are discovered through queries and the interface type is set to point-to-point presents a configuration error.

Sending multicast Hello packets on the point-to-point links allows OSPF neighbors to be discovered automatically. On the other hand, using Proxy-PAR instead avoids sending Hello messages to routers that are not necessarily running OSPF.

2.2.4 Autoconfiguration of Unnumbered Point-to-Point Interfaces

For reasons given in [14], the use of unnumbered point-to-point interfaces with Proxy-PAR is not a very attractive alternative because the lack of an IP address prevents efficient registration and retrieval of configuration information. Relying on the numbering method based on MIB entries generates conflicts with the dynamic nature of creation of such entries and is beyond the scope of this work.

2.3 Registration of OSPF interfaces with Proxy-PAR

To allow other routers to discover an OSPF interface automatically, the IP address, mask, Area ID, interface type and router priority information given must be registered with the Proxy-PAR server at an appropriate scope. A change in any of these parameters has to force a reregistration with Proxy-PAR.

It should be emphasized here that because the registration
information can be used by other routers to resolve IP addresses
against NSAPs as explained in section 2.4, the entire IP address of
the router must be registered. It is not sufficient to indicate the
subnet up to the mask length; all address bits must be provided.

2.3.1 Registration of Non-Broadcast Multiple-Access Interfaces

For an NBMA interface the appropriate parameters are available and
can be registered through Proxy-PAR without further complications.

2.3.2 Registration of Point-to-Multipoint Interfaces

In the case of a Point-to-MultiPoint interface the router registers
its information in the same fashion as in the NBMA case, except that
the interface type is modified accordingly.

2.3.3 Registration of Numbered Point-to-Point Interfaces

In the case of point-to-point numbered interfaces the address mask is
not specified in the OSPF configuration. If the router has to use
Proxy-PAR to advertise its capability, a mask must be defined or a
default value of 255.255.255.252 used.

2.3.4 Registration of Unnumbered Point-to-Point Interfaces

Owing to the lack of a configured IP address and difficulties
generated by this fact as described earlier, registration of
unnumbered point-to-point interfaces is not covered in this document.

2.4 IP address to NSAP Resolution Using Proxy-PAR

As a byproduct of Proxy-PAR presence, an OSPF implementation could
use the information in registrations for the resolution of IP
addresses to ATM NSAPs on a subnet without having to use static data
or mechanisms such as ATMARP [5]. This again should allow a drastic
simplification of the number of mechanisms involved in operating OSPF
over ATM to provide an IP overlay.

From a system perspective, the OSPF component, the Proxy-PAR client,
the IP to NSAP address resolution table, and the ATM circuit manager
can be depicted as in Figure 1. Figure 1 shows an example of
component interactions triggered by a Proxy-PAR query from the
Proxy-PAR client.

2.5 Connection Setup Mechanisms

   This section describes the OSPF behavior in an ATM network under
   various assumptions in terms of signaling capabilities and preset
   connectivity.

2.5.1 OSPF in PVC Environments

   In environments where only partial PVCs (or SPVCs) meshes are
   available and modeled as Point-to-MultiPoint interfaces, the routers
   see reachable routers through autodiscovery provided by Proxy-PAR.
   This leads to expected OSPF behavior. In cases where a full mesh of
   PVCs is present, such a network should preferably be modeled as NBMA.
   Note that in such a case, PVCs failures will translate into not-so-
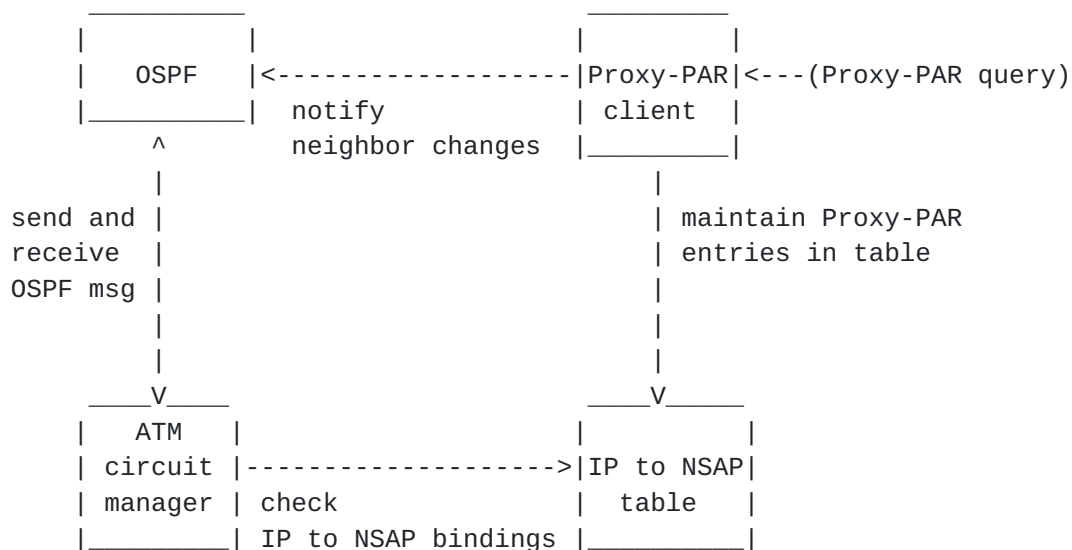   obvious routing failures.

        _____   _____
       | | | |
       | | | |
       | OSPF |<-------------------|Proxy-PAR|<---(Proxy-PAR query)
       |_____| notify | client |
            ^   neighbor changes |_____|
            | |
    send and | | maintain Proxy-PAR
    receive | | entries in table
    OSPF msg | |
            | |
            | |
        ____V_____   ____V_____
       | ATM | | |
       | circuit |-------------------->|IP to NSAP|
       | manager | check | table |
       |_____| IP to NSAP bindings |_____|

   Figure 1: System perspective of typical components interactions.

2.5.2 OSPF in SVC Environments

```
        +               +                           +
        |   +---+       |                           |
+--+    |---|RTA|---|            +-------+           |    +--+
|H1|---|    +---+     |           | ATM   |           |---|H2|
+--+    |             |   +---+   | Cloud |  +---+    |    +--+
        |LAN Y        |---|RTB|-------------|RTC|---|
        +             |   +---+   | PPAR  |  +---+    |
                      +           +-------+           +
```
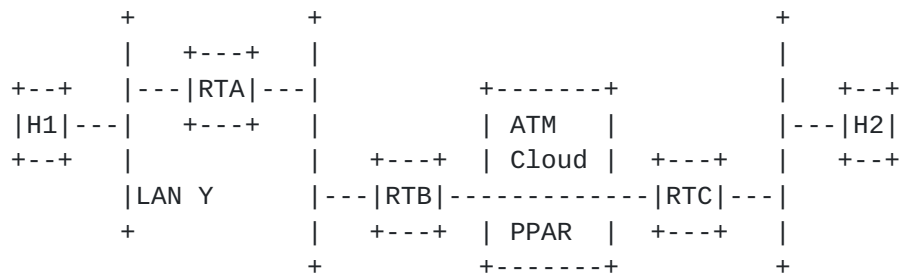
          Figure 2: Simple topology with Router B and Router C operating
                   across NBMA ATM interfaces with Proxy-PAR.

In SVC-capable environments the routers can initiate VCs after having
discovered the appropriate neighbors, preferably driven by the need
to send data such as Hello packets. This can lead to race conditions
where both sides can open a VC simultaneously. It is generally
desirable to avoid wasting this valuable resource: if the router with
lower IP address (i.e., the IP address of the OSPF interface
registered with Proxy-PAR) detects that the VC initiated by the other
side is bidirectional, it is free to close its own VC and use the
detected one. Note that this either requires the OSPF implementation
to be aware of the VCs used to send and receive Hello messages, or
the component responsible of managing VCs to be aware of the usage of
particular VCs.

Observe that this behavior operates correctly in case OSPF over
Demand Circuits extensions are used [13] over SVC capable interfaces.

Most of the time, it is possible to avoid the setup of redundant VCs
by delaying the sending of the first OSPF Hello from the router with
the lower IP address by an amout of time greater than the interval
between the queries from the Proxy-PAR client to the server. Chances
are that the router with the higher IP address opens the VC (or use
an already existing VC) and sends the OSPF Hello first if its
interval between queries is shorter than the Hello delay of the
router with the lower IP address. As this interval can vary depending
on particular needs and implementations, the race conditions
described above can still be expected to happen, albeit presumably
less often.

The existence of VCs used for OSPF exchanges is orthogonal to the
number and type of VCs the router chooses to use within the logical
interface to forward data to other routers. OSPF implementations are
free to use any of these VCs (in case they are aware of their
existence) to send packets if their end points are adequate and must
accept Hello packets arriving on any of the VCs belonging to the

logical interface even if OSPF operating on such an interface is not
aware of their existence. An OSPF implementation may ignore
connections being initiated by another router that has not been
discovered by Proxy-PAR. In any case, the OSPF implementation will
ignore a neighbor whose Proxy-PAR registration indicates that it is
not adjacent.

As an example consider the topology in Figure 2 where router RTB and
RTC are connected to a common ATM cloud offering Proxy-PAR services.
Assuming that RTB's OSPF implementation is aware of SVCs initiated on
the interface and that RTC only makes minimal use of Proxy-PAR
information, the following sequence could develop, illustrating some
of the cases described above:

  1. RTC and RTB register with ATM cloud as Proxy-PAR capable and
     discover each other as adjacent OSPF routers.

  2. RTB sends a Hello, which forces it to establish a SVC
     connection to RTC.

  3. RTC sends a Hello to RTB, but disregards the already existing
     VC and establishes a new VC to RTB to deliver the packet.

  4. RTB sees a new bidirectional VC and, assuming here that RTC's
     IP address is higher, closes the VC originated in step 2.

  5. Host H1 sends data to H2 and RTB establishes a new data SVC
     between itself and RTC.

  6. RTB sends a Hello to RTC and decides to do so using the newly
     establish data SVC. RTC must accept the Hello despite the
     minimal implementation.

[3](#) Acknowledgments

Comments and contributions from several sources, especially Rob
Coltun, Doug Dykeman, John Moy and Alex Zinin are included in this
work.

[4](#) Security Considerations

Several aspects are to be considered in the context of the security
of operating OSPF over ATM and/or Proxy-PAR. The security of
registered information handed to the ATM cloud must be guaranteed by
the underlying PNNI protocol. The registration itself through Proxy-
PAR is not secured, and are thus appropriate mechanisms for further
study. However, even if the security at the ATM layer is not
guaranteed, OSPF security mechanisms can be used to verify that

detected neighbors are authorized to interact with the entity
discovering them.

[5](#) Bibliography

[1]   ATM Forum, "PNNI Augmented Routing (PAR) Version 1.0." ATM
      Forum af-ra-0104.000, January 1999.

[2]   Droz, P. and T. Przygienda, "Proxy-PAR", [RFC 2843](#), May 2000.

[3]   ATM-Forum, "Private Network-Network Interface Specification
      Version 1.0." ATM Forum af-pnni-0055.000, March 1996.

[4]   ATM-Forum, "Interim Local Management Interface, (ILMI)
      Specification 4.0." ATM Forum af-ilmi-0065.000, September 1996.

[5]   Laubach, J., "Classical IP and ARP over ATM", [RFC 2225](#), April
      1998.

[6]   ATM-Forum, "LAN Emulation over ATM 1.0." ATM Forum af-lane-
      0021.000, January 1995.

[7]   Armitage, G., "Support for Multicast over UNI 3.0/3.1 based ATM
      Networks", [RFC 2022](#), November 1996.

[8]   Coltun, R., "The OSPF Opaque LSA Option", [RFC 2328](#), July 1998.

[9]   Davison, M., "ILMI-Based Server Discovery for ATMARP", [RFC 2601](#),
      June 1999.

[10] Davison, M., "ILMI-Based Server Discovery for MARS", [RFC 2602](#),
      June 1999.

[11] Davison, M., "ILMI-Based Server Discovery for NHRP", [RFC 2603](#),
      June 1999.

[12] Moy, J., "OSPF Version 2", [RFC 2328](#), April 1998.

[13] Moy, J., "Extending OSPF to Support Demand Circuits", [RFC 1793](#),
      April 1995.

[14] deSouza, O. and M. Rodrigues, "Guidelines for Running OSPF Over
      Frame Relay Networks", [RFC 1586](#), March 1994.

[15] Bradley, A. and C. Brown, "Inverse Address Resolution Protocol",
      [RFC 2390](#), September 1999.

Authors' Addresses

    Tony Przygienda
    Siara Systems Incorporated
    1195 Borregas Avenue
    Sunnyvale, CA 94089
    USA

    EMail: prz@siara.com

    Patrick Droz
    IBM Research
    Zurich Research Laboratory
    Saumerstrasse 4
    8803 Ruschlikon
    Switzerland

    EMail: dro@zurich.ibm.com

    Robert Haas
    IBM Research
    Zurich Research Laboratory
    Saumerstrasse 4
    8803 Ruschlikon
    Switzerland

    EMail: rha@zurich.ibm.com

Full Copyright Statement

Acknowledgement