Implementation of IP Autoconfiguration for IP over ATM

Matthew. B. Doar

mdoar@nexen.com

Abstract

Configuring the IP addresses and other information necessary for creating interfaces to a Logical IP Subnet (LIS) is performed automatically, improving the ease of configuration for ATM switches and IP routers which support RFC 1577 (Classical IP over ATM). The approach taken is easily extensible to other protocols.

IP over ATM - Background

One of the key overlays used in ATM networks is that of IP over ATM. This permits IP services to use an ATM network as a datalink layer, exactly in the manner that IP frames are carried over an Ethernet or a Token Ring network. The specifications for IP over ATM [RFC1577] have evolved in the ION working group (the current name) of the IETF during the past five years.

When a host on an Ethernet wishes to communicate with another host on the same Ethernet, the packet has to contain the correct layer 2, or MAC, address of the destination host, so that the destination host can recognize that the packet is destined for it. This address is obtained by using the Address Resolution Protocol (ARP) [RFC826] to inform the source host which MAC address is associated with the destination host's IP address. Once the destination MAC address is known, the source host can send the packet across the Ethernet to the destination host.

When running IP over an ATM network, one can view the ATM network as just another example of a layer 2 network, such as an Ethernet, with a similar need to again resolve IP addresses to layer 2 addresses, in this case, ATM addresses. Once the destination host's ATM address is known, the source host can set up a connection through the ATM network to the destination host's ATM address and send the packet on that connection, as Hosts A and C are doing in Figure 1.

The provision of IP over an ATM network is complicated by the fact that existing ARP in legacy LANs such as Ethernet uses the ability to broadcast packets to all destinations as part of the mechanism of address resolution. This is easy to do in shared media LANs, such as Ethernet, where packets can be easily broadcast to all hosts on a LAN. However, in an ATM network there is no equivalent mechanism for all hosts to be able to broadcast packets to all other hosts. This means that for IP to run successfully over an ATM network, some analogous mechanism to broadcast ARP packets needs to be created for address resolution.

`mdoar@nexen.com` or **`doar@pobox.com`**. Fujitsu Nexion, 289 Great Road, Acton, MA 01720. Tel: (508) 266-3468, Fax: (508) 266-2300

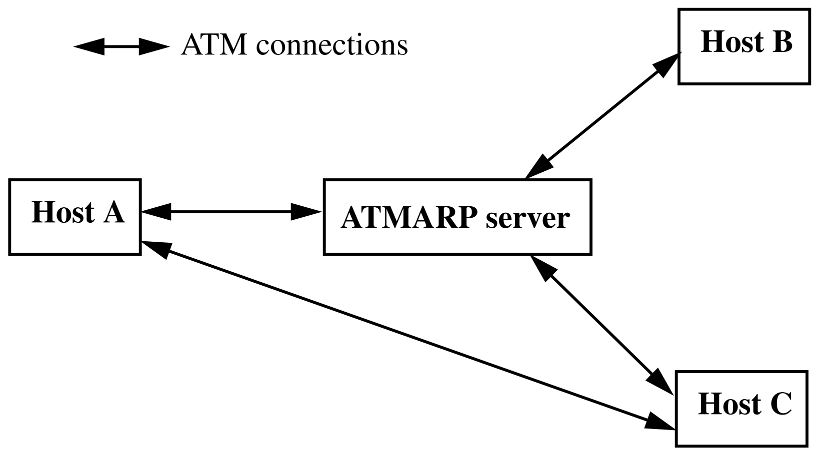

Figure 1: A Logical IP Subnet (LIS) with Three Hosts

The way that address resolution is provided in an IP over ATM overlay is through the use of an address server called the ‘ATMARP server’. All hosts which wish to send data in the IP network have to use the ATMARP server to get the ATM address associated with the IP address of the destination host. The ATMARP server keeps a table of all the known IP and ATM addresses of the hosts which have connected to it. The collection of hosts with interfaces connected to a particular ATMARP server and the ATMARP server itself is referred to as a Logical IP Subnet, or LIS. The LIS shown in Figure 1 comprises three hosts, all connected to the ATMARP server, and two of the hosts, A and C, directly connected after they used the ATMARP server to resolve one another’s ATM addresses.

The Need for Autoconfiguration

As has been documented in several places [RFC1900, RFC1971], manual configuration or reconfiguration of IPv4 networks is a tedious and error-prone activity, yet is a necessary part of permitting the Internet to grow exponentially. Autoconfiguration is also a crucial part of establishing connectivity in wireless networks.

The configuration of LIS’s has similar issues to that of the configuration of traditional IPv4 subnets. Some draft work has begun on the autoconfiguration of IPv6 LIS’s, which should be able to use the autoconfiguration capabilities of IPv6 for LIS configuration.

Aims of Autoconfiguration

The practical need for autoconfiguration of an IPv4 LIS was driven by the need within a company (Fujitsu Nexion) to manage its ATM switches across the ATM network made up of these switches. The standard network management protocol SNMP uses UDP over IP to carry management information between entities being managed. So one solution is to create a LIS called the “management LIS” to carry the IP (SNMP) packets over the ATM network. Since this is a necessary part of the network configuration, it seemed appropriate that the management LIS should be automatically created and that each switch should be automatically connected to the management LIS. The information needed to configure a single host onto an Ethernet can be minimally stated as:

- IP address of the interface
- Network mask
- Subnetwork mask

Similarly, the information needed to configure a LIS can be minimally stated as:

- IP addresses of the interfaces to be attached to the LIS
- Network mask
- Subnetwork mask
- ATM address of the ATMARP server for that LIS

There are other pieces of configuration information, such as the IP address of a default router, the Maximum Transmission Unit (MTU) for the LIS, the encapsulation type used on connections and various timers, but these can often use defaults, or can also be distributed by an autoconfiguration system.

The other aim of an autoconfiguration system is that it really should make configuring the switches in an ATM network easier for the network administrator. To this end, a number of other aims were defined:

- There should be a central place for entering information, so that the administrator only has to enter information in one place.
- Minimal manual intervention should be necessary, but total manual intervention should be possible.
- The information itself should be easy to enter, for example, a range of IP addresses, rather than every IP address in the range.
- The configuration system itself should be able to be managed using SNMP. This produces a chicken-and-egg problem, since the LIS must be created before SNMP can be used to manage the autoconfiguration system which creates the management LIS. This can be solved by creating the autoconfiguration system using a local craft interface, then managing it remotely only after the LIS has been created.
- Finally, the configuration system should scale well as the number of switches in the ATM network increases.

Client/Server *vs.* Distributed Configuration Architectures

There are two broad approaches for ensuring the distribution of information within a network. The first, the client/server approach, assumes that there is a central point for distributing the information (the server) and various clients which retrieve the information from the server. This approach is used by large file systems in LANs, for instance. The second, the distributed approach, maintains the configuration information at each place it is needed, such as at a host in a LAN. In this approach, the information is updated by sending each hosts' copy of the information to the other hosts for them to update their copies of the information. This is the approach used by

routing protocols, such as RIP and others.
One advantage of a client/server approach is that it can be made to scale well as the network grows in size by using hierarchies of servers, just as the heart of the Internet naming scheme, the Domain Name System (DNS), does. One advantage of the distributed approach to replicating information is that of reliability in the face of failure. This is because the failure of any number of hosts does not stop an unaffected host from using the information which it has locally and obtaining more information from the remaining hosts. However, protocols which involve exchanging large amounts of information between every involved host often fail to scale well as the number of hosts grows, so for information distribution to large numbers of hosts, a client/server approach is often used.

The obvious way to reduce the effects of server failure on the client/server approach is to add other servers to the network. Note that this is different from just making hierarchies of servers, since each server in the hierarchy is still a single point of failure for its clients. The replication of information contained by the servers in a network then leads to questions of how to ensure that the information is consistent between the servers, which in turn leads to a distributed architecture for the servers. Thus any architecture for the dissemination of information can be expected to have the potential for a distributed architecture at its core, possibly with a client/server approach outside the core for improved scalability.

Existing Configuration Protocols

There are almost as many configuration protocols as there are protocols which need to be configured. An early example is RARP [RFC903], where a host can broadcast, at layer 2, a request for an IP address for itself. Somewhere in the LAN, a RARP server is listening and will reply to the host with a suitable IP address. BOOTP [RFC951] is like RARP but uses UDP, rather than a layer 2 broadcast, and consequently can be routed. BOOTP can also be used to obtain more than just an IP address for a host. DHCP [RFC2131] is based on BOOTP with the capability of automatic reallocation of IP addresses when a network needs to be reconfigured.

All of the above protocols expect that either a shared medium exists so that layer 2 broadcast is easy, or that IP connectivity already exists in the network. In an ATM network, neither case is true, which is why a novel protocol was developed for the autoconfiguration of IPv4 LIS's.

Work currently in progress for IPv6 discusses the use of stateful and stateless autoconfiguration [RFC1971]. Stateful autoconfiguration is where a server is used to distribute information, as in the configuration protocols mentioned above. Stateless autoconfiguration is where a host can construct its own address using its layer 2 address and information common to all the hosts on a subnet. The IPv6 work also introduces the idea of lifetime for an address, after which its use becomes deprecated automatically. The ideas being investigated in the IPv6 work are useful and will no doubt be introduced to future configuration protocols for ATM networks.

The IP Autoconfiguration (IPAC) Architecture

Work at Ascom Nexion (now Fujitsu Nexion) by the author, together with Jim Luciani, Hal Rosenstock, Bill Willcox and others, lead to the IP Autoconfiguration (IPAC) architecture shown in Figure 2. As can be seen, there are three tiers of information servers; DOCS (Directory Of Co figuration Servers), CS's (Configuration Servers) and CC's (Configuration Clients).

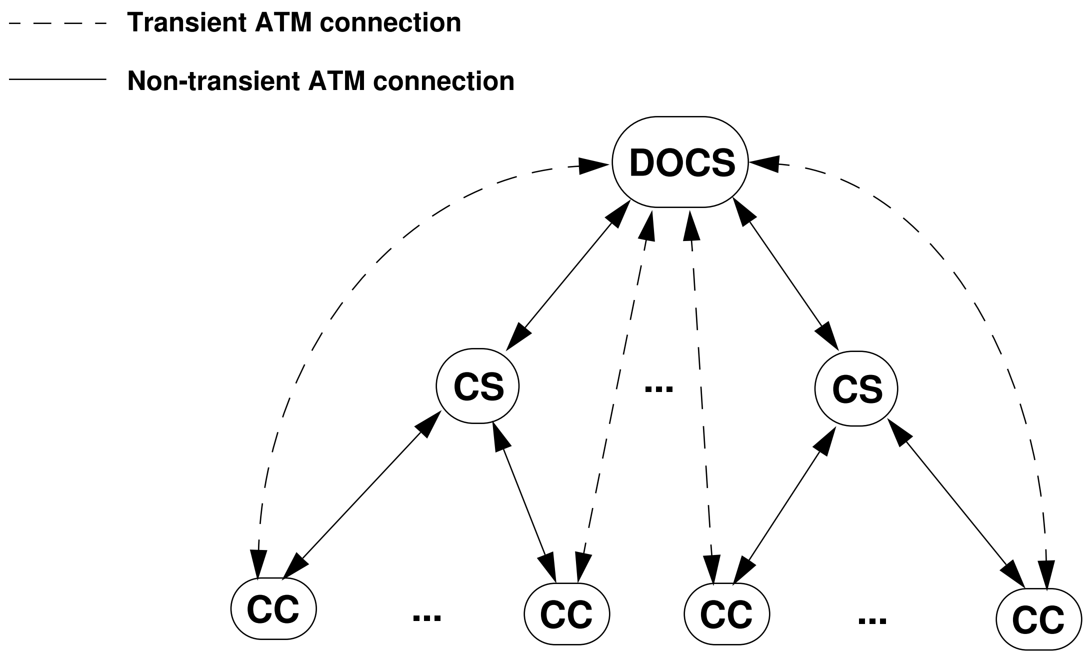

Figure 2: IPAC Architecture

The purpose of the DOCS is to inform CC's which CS they should use. The DOCS does not keep any permanent connection to any CC's, which means that the configuration system can scale without the DOCS becoming a bottle-neck (assuming that the signalling implementation can handle large numbers of connection requests). The DOCS can act as the central point for entering configuration information. It passes this information onto the appropriate CS's.

The purpose of a CS is to provide CC's with the configuration information which they require, such as IP addresses, network and subnetwork masks and the ATM address of the ATMARP server for the management LIS. Each CS need only store the information relevant to the group of CC's which might potentially attach to it.

The purpose of a CC is to obtain the necessary configuration information and either use it actively to create an interface to a LIS, or to provide it passively for some other piece of software such as a Network Management application in the host to use to create an interface to a LIS. Each CC is only connected to one CS. The connection from the CC to the DOCS is transient in that after the CC has found out from the DOCS which CS it should use, it has no more use for the DOCS and can disconnect from it.

A typical implementation of the architecture in Figure 2 might be to have a CC on each ATM switch in the network, one CS per PNNI peer-group or geographical cluster of switches and the DOCS at a geographically central and well-connected place in the network. Typically, one might expect less than 100 CC's per CS and less than 100 CS's per DOCS. The ATM address of the DOCS can either be a well-known ATM address, or it can be configured at each CC and CS at start-up, possibly through ILMI. Manual configuration of IP addresses at the CC is permitted in our implementation, and in fact uses the DOCS ‘in reverse’, to check that the IP addresses in question have not already been assigned anywhere else in the LIS.

Redundancy

The purpose of the provision of redundancy in the autoconfiguration system is to remove the single points of failure such as the DOCS and the CS (for each group of CC’s). The first point of failure, the DOCS, can be avoided by having several DOCS’s in the network. The multiple DOCS’s can be kept in synchronization by using other IETF draft work on the Server Cache Synchronization Protocol (SCSP) or any other of a number of simple reliable multicast protocols[CAP].

The other point of failure, the CS for a group of CC’s, can be removed by permitting each CC to recontact an available DOCS for the ATM address of another CS, should the one originally provided fail to perform properly. The alternate CS may be either a CS from a different group of CC’s, thus merging the two groups of CC’s, or it may be a replicated CS, in the same sense that the DOCS can be replicated.

There are several extensions to the simple hierarchical client/server approach which need to be made for redundancy, not least a mechanism for tracking when clients and servers are out of synchronization after reboots. The current implementation of IPAC used a simple integer at each DOCS, CS and CC, restored from persistent storage after a reboot and incremented. This permits a comparison between server and client instances and updating of the relevant information, as necessary.

Conclusion

The hierarchical client/server approach described in this paper provides the basis of a scalable solution to the problem of distributing configuration information, which is necessary to bring real networks closer to the ideal of “plug and play”. The availability issues which come with the client/server approach can be solved, particularly if the number of servers which need to be replicated is small. The work described here has been used to implement an autoconfiguration protocol for configuring IP over ATM Logical IP Subnets, which are used for managing ATM networks.

References

[RFC1577] "Classical IP and ARP over ATM", M. Laubach, January 1994

[RFC826] "An Ethernet Address Resolution Protocol", D. C. Plummer, November 1982

[RFC1900] "Renumbering Needs Work", B. Carpenter, Y. Rekhter, February 1996

[RFC1971] "IPv6 Stateless Address Autoconfiguration", S. Thomson, T. Narten, August 1996

[RFC903] "A Reverse Address Resolution Protocol", R. Finlayson, T. Mann, J. Mogul, M. Theimer, June 1984

[RFC951] "BOOTSTRAP PROTOCOL (BOOTP)", W. Croft, J. Gilmore, September 1985

[RFC2131] "Dynamic Host Configuration Protocol", R. Droms, March 1997