Implementation of Fujitsu's Auto-Configuration for IP over ATM

Matthew B. Doar
© 1997 Fujitsu Network Communications, Inc.
mdoar@nexen.com

Abstract

Configuring the IP addresses and other information necessary for a Logical IP Subnet (LIS) is performed automatically in Fujitsu's ATM switches and IP routers which support RFC 1577 Classical IP over ATM. This approach is easily extensible to other protocols.

IP over ATM - Background

One of the key overlays used in networks is that of IP over ATM. This permits IP services to use an ATM network as a data link layer in the manner that IP frames over Ethernet or token Ring networks are carried. The specifications for IP over ATM [RFC1577] have evolved in the IP over ATM 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 Ethernet address of the destination host, so that the destination host can recognize that the packet is destined for it. This address is obtained by the Address Resolution Protocol (ARP) [RFC826] to inform the source host which Ethernet address is associated with the destination host's IP address. Once the destination host's Ethernet address is known, the source host can send packets 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 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 packets 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 many types of LANs such as Ethernet uses the ability to broadcast packets as part of the normal 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 the 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 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, 289 Great Road, Acton, MA 01720. T: (508) 266-3468, F: (508) 266-2300

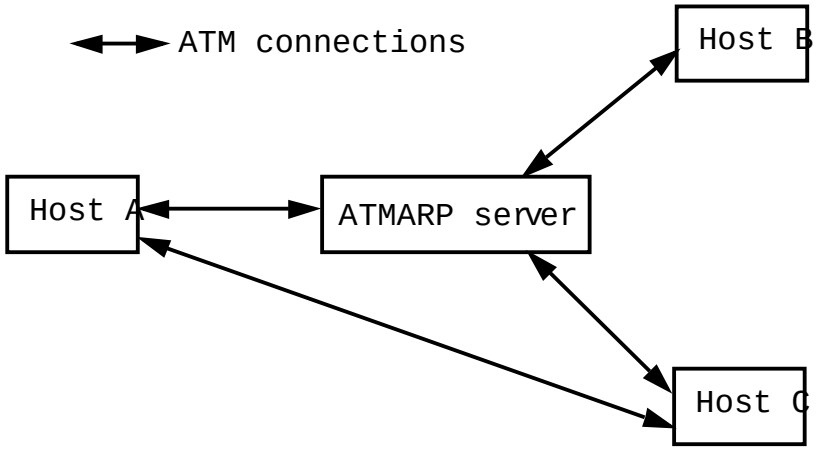

Figure 1: A Logical IP Subnet (LIS) with Three Hosts

The way that address resolution is provided in an IP over ATM overlay is through the use of an address server called an 'ATMARP server'. All hosts which wish to send data in the IP overlay 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 IP and ATM addresses of the hosts which have connected to it. The collection of hosts 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. Two of the hosts, A and C, directly connected to the ATMARP server, use 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 of IPv4 networks is a tedious, 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 LISs has similar issues to that of the configuration of traditional nets. Some work has gone on the autoconfiguration of IPv6 LISs, which should be able to use the autoconfiguration capabilities of IPv6 for LIS configuration.

Aims of Autoconfiguration

The practical need for autoconfiguration arose when IPv4 LISs were deployed within a company (Fujitsu, Ltd.) to manage ATM switches across the ATM network made up of these switches. The standard network management protocol SNMP uses IP to carry management information between entities being managed. So one solution is to create a LIS called a "management LIS" to carry the IP (SNMP) traffic over the ATM network. Since this is a necessary part of the network configuration, it seemed appropriate that the management LIS should

cally created and that each switch should be automatically connected to the
The information needed to configure a single host onto an Ethernet can be mi

- 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 hosts 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 ATM address
Maximum Transmission Unit (MTU) for the LIS, the encapsulation type used on
various timers; these can often be set to default values, or can also be distri
system.

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

- There should be a central place for entering information, so that the adm
  enter information in one place.
- Minimal manual information should be necessary; ideally, no 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 introduces a
  chicken-and-egg problem, since the LIS must be created before SNMP can be used to configure
  the autoconfiguration system which creates the managed LIS. This can be solved by creating
  the autoconfiguration system using a local management interface, and configuring it remotely only
  after the LIS has been created.
- Finally, the configuration system should scale well as the number of switches in the net-
  work increases.

Client/Server vs. Distributed Configuration Architecture

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

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. The Domain Name System (DNS), does just this. One disadvantage of the client/server approach to replicating information is that of reliability in the face of failure. This is because the failure of a small number of hosts does not stop the needy host from using the information which it has locally, or from obtaining more information from the remaining servers which are functioning. However, protocols which involve exchanging large amounts of information between each host and a number of servers do not tend 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 reliance on the client/server approach is to add redundancy to the servers. 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 then leads to questions of how to ensure that the information is consistent between servers, which in turn leads to more distributed architectures for the servers. Thus any architecture for the dissemination of information can be evaluated in terms of the potential for a distributed architecture at its core, possibly with a client/server approach on the outside the core for ease of deployability.

Existing Configuration Protocols

There are almost as many configuration protocols as there are protocols which need to be configured. An example is RARP [RFC903], where a host can broadcast, at layer 2, its MAC address in the hope of finding an IP address for itself. In the LAN, a RARP server is listening and will reply to the requesting host with a suitable IP address. BOOTP [RFC951] is similar to 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 and has the capability of automatic reuse and allocation of IP addresses when needed to be reconfigured.

All of these protocols expect that either a shared medium exists so that layer 2 broadcast is easy or that IP configuration data is in the network. In an ATM network, neither case is true, which is a major problem as well as the fact that no protocols have been 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 dedicated servers hold information, as in the configuration protocols mentioned above. Stateless autoconfiguration is where a host can autonomously construct an address using its layer 2 address and information common to all the hosts on the subnet. The IPv6 work also introduces the idea of lifetime for an address, after which it is deprecated automatically. These ideas have been adopted in the IPv6 work as 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 Wolfe and others, lead to the IP Autoconfiguration (IPAC) architecture shown in Figure 2. As can be seen, there are three entities: DCS (Information Server), CSS (Configuration Server) and CC (Configuration Clients).

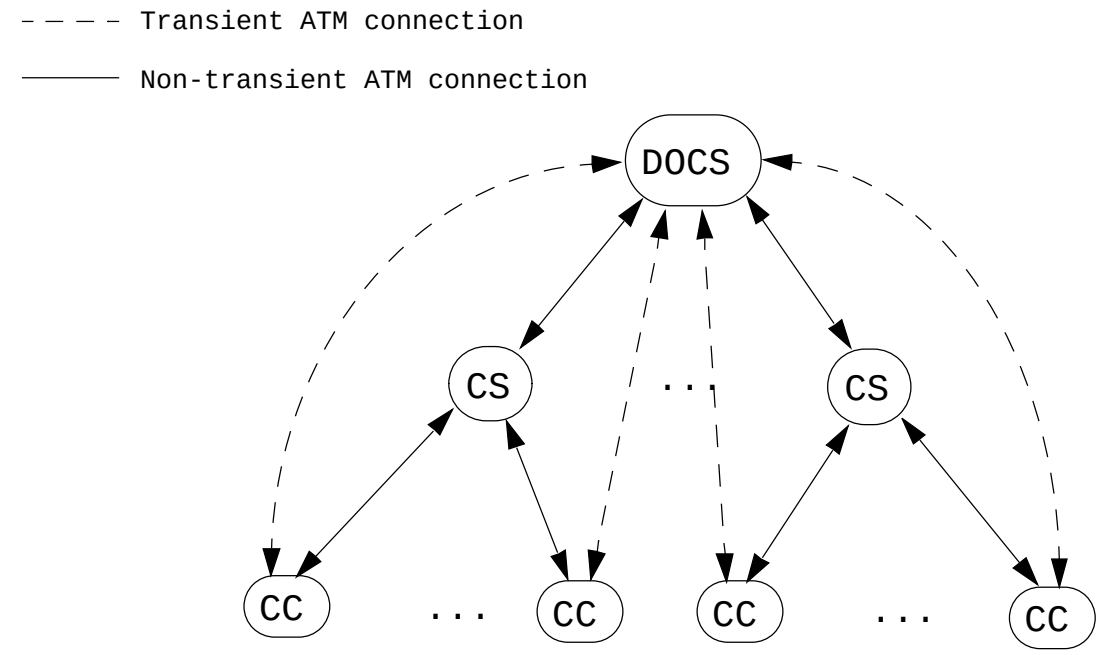

Figure 2: ATM Architecture

The purpose of the DOCS is to inform CSs which CSs they should use. The DOCS does not k
any permanent connection to CSs, which means that the configuration system can sca
out the DOCS becoming a bottle-neck (assuming that the signalling implement
large numbers of connection requests). The DOCS can act as the central point
uration information. It passes this information on to the appropriate CS's.

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

The purpose of a CC is to obtain the necessary configuration information from a DOCS to create and enter a LIS, or to join one already in progress for some other piece of software in the Network Management application in the host to use to create a LIS. Each CC is only connected to one CS. The connection from the CC to the DOCS is transie CC has found out from the DOCS which CS it should use, it has no more use f can disconnect from it.

A typical implementation of the architecture in Figure 2 might have one DOCS per ATM switch in the network, one CS per PNNI peer group or geographical cluster of switches an DOCS at a geographically central and well-connected switch, and the network expect less than 100 CCs per CS and less than 1000 CCs per DOCS. The address of the DOCS can either be a well-known ATM address, or it can be configured at each CC and C

start-up, possibly through ILMI. Manual configuration of IP addresses at the our implementation, and is used in the DOCS server, to check that the IP addresses in question have not already been assigned elsewhere in the LIS.

Redundancy

The purpose of the provision of redundancy in the autoconfiguration system is to remove single points of failure such as the DOCS and the CS (for each group of CC). 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 drafts such as the Server Cache Synchronization Protocol (SCSP) or one of a number of simple reliable multicast protocols.

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 uses a simple integer at each DOCS, CS and CC, restored from persistent storage after a reboot and increm a comparison between server and client instances and updating of the 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 networks closer to the ideal of “plug and play”. The scalability 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 autoconfigurat for configuring IP over ATM Logical IP Subnets, which are used for networking ATM.

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