An Overview of Automatic Network Configuration for IPv4 Appliances

Denis Chalon, Yves Durand, Bruno Richard
HP Laboratories Grenoble
HPL-2001-235
September 28th , 2001*

E-mail: {Denis_Chalon, Yves_Durand, Bruno_Richard}@hp.com

appliance, configuration, IPv4, discovery

In the near future, appliances will be able to cooperate to provide useful services to consumers. Thus, common appliances such as Set-top boxes, PDAs, car positioning devices will require network connectivity. But still, the unacceptable complexity of network configuration hinders the acceptance of the market. The only way to overcome this is to rely on fully automated configuration mechanisms.

This document aims at providing a detailed view about auto configuration issues in IPv4 networks to software developers and architects who may not necessarily be networking experts. We restrict our analysis to appliance networks, and thus we focus on specific issues like the absence of central servers for functions such as Domain Name Service or routing.

This work is a foundation for the effort at HP Labs Grenoble to implement an "appliance ecosystem" [18] from existing technologies. Thus, even if IPv6 resolves most of the existing hurdles of auto configuation, we focus here on the mechanisms that are common to both IPv4 and IPv6, in order to provide a solution suitable for the transition time between both.

September 5, 2001

An Overview of Automatic Network Configuration for IPv4 Appliances

Denis Chalon, Yves Durand, Bruno Richard, HP Labs Grenoble

{Denis_Chalon, Yves_Durand, Bruno_Richard}@hp.com

Abstract. In the near future, appliances will be able to cooperate to provide useful services to consumers. Thus, common appliances such as Set-top boxes, PDAs, car positioning devices will require network connectivity. But still, the unacceptable complexity of network configuration hinders the acceptance of the market. The only way to overcome this is to rely on fully automated configuration mechanisms.

This document aims at providing a detailed view about auto configuration issues in IPv4 networks to software developers and architects who may not necessarily be networking experts. We restrict our analysis to appliance networks, and thus we focus on specific issues like the absence of central servers for functions such as Domain Name Service or routing.

This work is a foundation for the effort at HP Labs Grenoble to implement an “appliance ecosystem” [18] from existing technologies. Thus, even if IPv6 resolves most of the existing hurdles of auto configuration, we focus here on the mechanisms that are common to both IPv4 and IPv6, in order to provide a solution suitable for the transition time between both.

1 Introduction

As appliances are becoming communicating devices, the first step for enabling communications is to link them together. A second step is then to establish a connection with the outside world. IP is not the only connectivity solution today, but it has become the standard for enterprise networks. Thus, the strategy consists in leveraging the reliability and maturity of IP and apply them to a range of less powerful devices. Although big efforts have been made to simplify the configuration of IP networks, the IP structure is still mainly in enterprise networks. And these networks are managed by skilled engineers. Our objective is to create a *fully functional network* with no manual configuration at all. Within such a network a device is able to aggregate spontaneously to a pre-existing appliance community:

- Without any manual configuration, the device gains basic network connectivity: it is automatically aware of the existence of others.
- The device is able to use standard facilities already offered by the community: routing to Internet, proxies.
- The device will be able to discover and use specific services offered by the community, with no or minimal pre-installed software.

Section 2 of this document enumerates the IP network services that are pertinent to an appliance network. Then we list the necessary settings for these services, and a few issues that arise during configuration. Section 3 describes different methods to actually configure the appliance network: we distinguish the case where configuration is fully manual, from the case where the configuration benefits from infrastructure services provided by pre-existing servers. We also cover the case of server-less structures. Finally, we recommend and discuss a minimal solution for appliance networks, based on state-of-the-art solutions.

2 Configuration requirements of an appliance network

2.1 A minimal infrastructure for an IP network of appliances

We divide the basic requirements for our appliance network in three distinct areas:

- IP existence
- IP routing, namely towards the external world
- Service support: discovery and capability advertising

In the first place, it is necessary for devices to have an IP existence, in other words to have the proper IP settings that will allow the connection between themselves and the outside world. The problem here is that the set of IP addresses, Gateway addresses and Subnet masks are presumably unique to each network. Hence it is not possible to hardcode those settings without any *a priori* knowledge of the target environment.

Service between appliances is the second aspect to take into account. An appliance network is useful if the aggregation of devices is able to provide some service to the end-user. But it requires some synchronization and some resources sharing for appliances to cooperate.

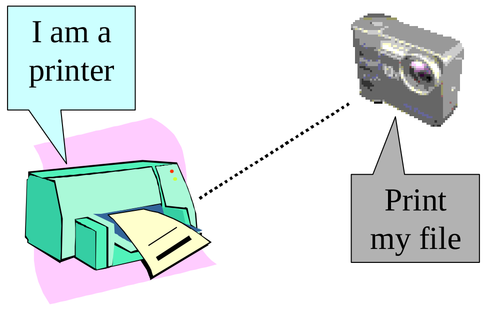

FIGURE 1. Camera and printer. They share a network connection. The printer exposes its capabilities, and the camera is able to invoke its printing service

Both aspects described above, namely connectivity and services, must comply with the very specific constraints of appliances:

- ***neutrality***: an appliance cannot rely on the presence of a specific network service to configure its own network parameters.
- ***auto-configurability***: the dynamic addition or removal of any appliance from the network should not involve any manual intervention from the user

Next paragraph lists the settings that are mandatory for an IP-based network of appliances to operate.

2.2 IP basics: IP address and subnets

Any IP device requires some specific parameters in order to operate on the network:

1. An IP Address. This is both a way to get a unique key at a given time and a necessary address to receive packets. Currently mobile users get different IP addresses whenever they move and connect to a different network. The IP address is a 32 bits integer, often represented as a sequence of 4 decimal byte integers separated by a dot. For example, 10.2.4.1 is such an address. The IP address is the only absolutely mandatory parameter, as the lack of an IP address will prevent any communication with peer devices (located on the same subnet). The IP address has to be unique on the subnet[1] and in line with other parameters to function.
2. A subnet mask, which is a bitmask-encoded filter determining which part of the IP address will be common to all the IP devices on the subnet. It should be noted that the Gateway address and subnet mask are necessary for routing purposes i.e. to allow network packets to reach IP entities which are outside the subnet.

1. in theory, the IP address has to be unique on internet. This is true for any device directly connected to internet. However, this point is addressed in paragraph 2.4 “Gateway to outside world”.

2.3 Discovery of local Services

Obtaining an IP address in the network is meaningless if the device cannot reach other devices, or more precisely the services they provide. In this paragraph, we will focus on the service discovery within the appliance network. The broader topic of Internet service discovery is beyond our scope: it has been analyzed in another HP Labs Report [6]

We focus our execution model on the concept of *service delegation.* One reason is that we target our appliance network on small devices with little computing power. In this model, the small device which is the initiator relies on the functionality of the other resource.

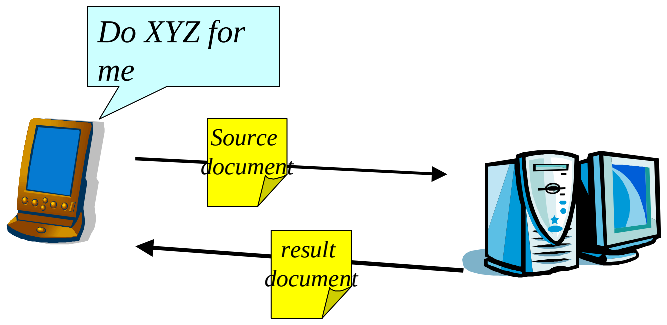

FIGURE 2. service invocation through delegation

The alternative model is the *shared execution*, the service receives and executes parts of initiator's code i.e. the initiator relies on computing power of the other resource.

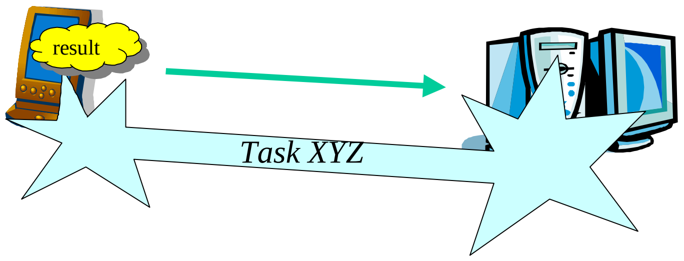

FIGURE 3. service invocation through shared execution

Coming back to the delegation model, it becomes crucial for the initiator to be aware of the existence of *service resources* among his neighbours. This allows him to filter the offered resources that best match its requirements. We will not detail this *filtering* (or *brokering*) feature.

On the other end, any appliance has to expose to other nodes the services that it offers. This feature is called *advertising*, and has to be somehow provided by the appliance network.

There are two ways to implement the *advertising* of capabilities: the first one is that each appliance is responsible of periodically announcing its characteristics the others.

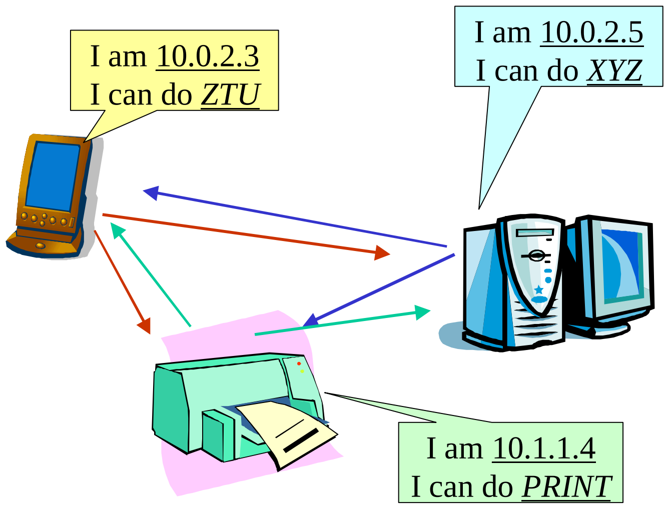

FIGURE 4. service advertising through broadcast

The alternative way to expose capabilities is to use a directory service. When joining the network, appliance register with the directory. Appliance requesting a service have to query the directory for references. The directory is itself a service (often called look-up service).

.

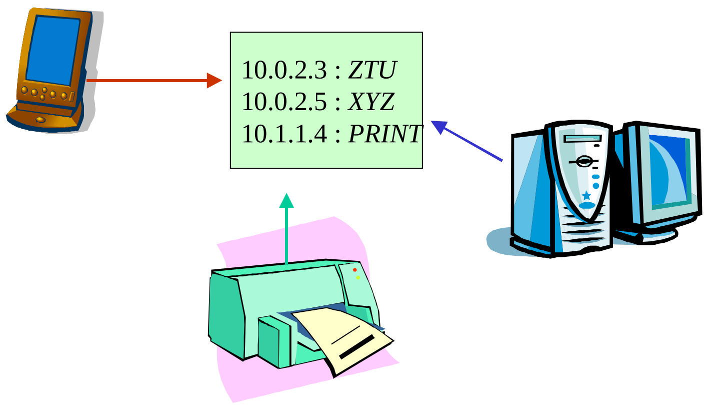

FIGURE 5. service advertising using a directory

The model of delegated services requires a mapping between IP addresses and services: this appears explicitly in figure 5. The section above discusses alternatives on how to implement the mapping. Another crucial issue is to maintain the stability of such a mapping. Unfortunately, there is no unique solution in the IPv4 world today. Enterprises use combinations of name servers and application solution (task brokers) to resolve it. Specialized middleware (e.g. CORBA solutions) implement elaborated solutions the applicative tier of 3-tier applications. But none of these solutions satisfies the criterion of "no configuration" for discovery of local services. On the other hand, Ramanathan et al. discuss in [6] a method based on discovery agents to discover services in a service provider network.

In paragraph 3.4, we list a few current alternatives that are envisageable for networked appliances.

2.4 Gateway to the outside world

Appliances may be able to provide useful services to the end user when they collaborate. The value of this network improves greatly if it provides Internet-wide connectivity. The first issue here is the routability of the IP addresses of our devices. There are two kind of IP addresses:

- *routable* addresses: a routable address is globally valid on the public Internet. Routable addresses are identified by specific ranges.
- *non-routable* addresses: a non-routable address is agreed upon locally, but since it is not unique within Internet, it can not be used outside the local cell. It is also possible to use a theoretically routable address inside a local cell.

At this point we also distinguish two types of connections: Outbound connection, where the initiating appliance is able to reach Internet, and inbound connection, where the Internet is able to reach an appliance inside the network. We consider that a network offering a reliable inbound access goes beyond the minimalism that we want to adopt here. In the remainder of this paragraph, we will focus on outbound access to the Internet.

Unmediated Outbound access. direct access to the Internet adds some constraints: each appliance on the network must have its own individual valid address (e.g. routable addresses).This requirement is almost impossible to fulfill in a dynamic environment, mainly because the limitation of the address space in IPv4. Thus will focus on mediated access to internet.

Mediated access to internet: the simplest way to connect multiple appliance to the internet is to connect them to a particular appliance which has a valid routable IP address. The particular appliance, that we call the mediator, will redirect the traffic in and out the local appliance network.

- Network address translation: the appliances need to know the router address which identifies the mediator, called the network router. This host routes network packets whose destination is outside current subnet. In case of shared Internet access, Network Address Translation (NAT) or IP masquerading are common mechanisms that enable sharing of an IP address[1].
- The alternative is to mediate at application level (such as Proxy server and socks server address). Mainly found in corporate environments, this kind of accesses will be used more and more in home environments to allow various home devices to access the Internet. It works well with *client/server* architectures, where the client is using proxy or socks servers to get access to the service[2].

Proxies, socks server on one end and router on the other are doing same kind of work: moving packet from one subnet to another. However routers are not handling specificities from application level: all packets are routed the same way (IP level routing). Proxies and socks servers filters some applications traffic (TCP or application level routing). However, from our point of view, both solutions share the same advantages and issues:

- they hide the internal address away from the external world: this simplifies the IP attribution problem
- the connection to internet is appears as another local service: thus, it shares the problematic of discovery that we described in paragraph 2.3.

3 Configuration Techniques

Our starting assumption was that manual configuration is not acceptable for a network of appliances. One good reason for this is that some appliances have reduced or no user interface: the configuration process then requires to tweak DIP switches, or   to connect a terminal to a serial port for entering the basic configuration parameters.

There are basically three possibilities left:

- manual pre configuration: delegate the manual configuration to the vendor.
- automatic client configuration: mimic the enterprise network configuration, by re-creating DHCP/DNS on the appliance network.

1. Note: NAT and IP masquerading are changing packet contents and headers. Cryptographic protocols such as IPSec and SSL ensure end to end confidentiality and integrity during transport. Then usage of both technologies is not possible as is: cryptographic protocols will detect changes in the packet structure and will reject it. The only solution consists in trusting the NAT router and giving it necessary keys to do the cryptography by it-self.
2. In *peer- to-peer* service architectures, connection could be initiated from both sides and classical “proxy” servers do not support it. The support of peer-to-peer applications requires specific proxy servers, as the connection could be either inbound or outbound.

- automatic configuration: use a specific solution.

3.1 Manual pre configuration

A side method for assigning the IP configuration parameters without requiring a technician to come at the device's deployment place is to configure the host at a central location, before the actual deployment. For example, this technique is used by "BeAtHome" [2] to pre configure home appliances.When the device reaches its final location the IP parameters are already assigned to it and it can be remotely accessed through TCP/IP.

Drawbacks . This method is inapplicable for dynamic networks. It does not fulfill the requirements defined in paragraph 2.1.

3.2 Automatic Client IP configuration

In the previous case, each appliance is a repository of its own network settings. Automatic client IP configuration factorize all these settings distributed in each appliance into a centralized location. This centralized location is then in charge of redistributing settings to appliances. We describe hereafter how this is done in enterprise networks.

Static IP address assignment through BOOTP or RARP servers. RARP is historically the first automated way to automatically retrieve IP addresses. In Ethernet networks, a MAC address is available and unique. This is how devices are identified. Server provides an IP address to a requesting host when MAC address of this host matches an entry in server directory. This mechanism was implemented first in a RARP server. But while RARP is simple to implement and to use, this configuration mechanism can only supply the IP address. The Gateway and Subnet mask have to be configured in another manner.

BOOTP [3] is an evolution of the RARP mechanism. It includes the ability to configure the Gateway and Subnet mask, and also suppresses the broadcast requirement. Moreover BOOTP supports wide spread networks through BOOTP Relay Agent mechanism supported by a router. A single BOOTP server is able to provide IP addresses for several subnets.

The BOOTP mechanism is simple: an IP appliance sends a UDP datagram to a BOOTP server. Then servers responds with the proper configuration parameters needed by the appliance. An advantage of BOOTP, besides its simplicity, is that it can include some vendor specific extensions, A typical use of this is to provide some proxy configuration information.

Dynamic IP address assignment with DHCP servers. The Dynamic Host Configuration protocol (DHCP) is a superset of the BOOTP protocol. The major benefit of DHCP over BOOTP is that IP addresses have a leasing capability by which the life cycle of the IP address can be inherently managed.Moreover, IP addresses can be dynamically assigned to devices and handled by the protocol helpers (client and server), hence the administrator just has to assign a range of addresses that are allowed to all the IP appliances.Then the DHCP server will pick addresses in the range and dynamically assign them to the requesting devices. The DHCP request and response packets follow the same format as BOOTP with some DHCP specific extensions.

Drawbacks. The problem is that there is no way to remotely determine the address picked up by a given device. This eventually makes some tasks difficult, like for example programmatic configuration of printers.

DHCP as a repository for Configuration data. DHCP server may alternatively be used as a repository for some network configuration options. A lot of extensions to DHCP have been added. They are used to store DNS, Gateway, subnet information among others.

DNS as a repository for mapping services to hosts. Once a DNS is located, it may also be used to obtain configuration data. Current DNS implementations are repository for network information: mail server addresses, server addresses. For example, the postfix “MX” appended to an IP address (like in 192.168.0.136 MX) in DNS configuration files denotes that 192.168.0.136 server is a mail server.

3.3 Auto-configuration in neutral Environment

The previous sections describe mechanisms that may help achieving automatic configuration of devices. At this stage, the configuration process is global for the network, instead of being local to every single device. Unfortunately, these mechanisms rely on infrastructure processes, which have to be configured too. These management and configuration tasks are still expensive and heavy. However our objective is more ambitious: no configuration at all.

This section details methods to retrieve parameters automatically. These methods are using either some characteristics or constraints of common network to solve configuration problems.

3.3.1 Automatic Private IP Assignment

Extensions to the standard DHCP protocol describes new ways to get an IP address even when no server is available. These mechanisms are known as *Autonet*, *Auto-IP Configuration*, *IP-Auto Configuration*, or *Automatic Private IP Addressing* (APIPA) by Microsoft.Ofiicially, they are referred to as “Dynamic Configuration of IPv4 link-local addresses”[14]. In the remainder of the document, we will denote it *AutoIP.*

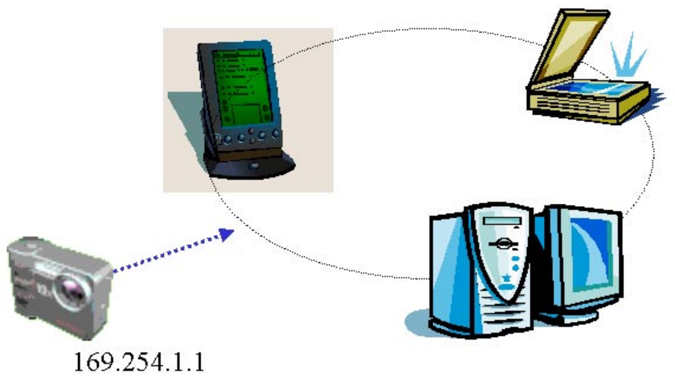

FIGURE 6. **Automatic IP address assignment**

With AutoIP, DHCP clients can automatically obtain an IP address and subnet mask when a DHCP server isn't available. When the DHCP client boots up, it looks for a DHCP server to obtain an IP address, subnet mask, and other possible DHCP options configured on the server. If the client is unable to obtain a lease from a DHCP server, the client uses AutoIP and automatically configures itself with an IP address from a reserved IP address range of 169.254.0.1 through 169.254.255.254 (range 169.254/16). The Internet Assigned Numbers Authority (IANA) has reserved this range for private IP addressing, so no one can use it on the Internet. The client also configures itself with a default class B subnet mask of 255.255.0.0. A client uses the self-configured IP address until a DHCP server becomes available.

Note: AutoIP is meant for non-routed small environments. AutoIP uses ARP to resolve potential IP address conflicts. When an AutoIP client boots up, it randomly picks an IP address in 169.254/16 range. To avoid duplication on the network, it sends a gratuitous ARP to ensure that no other clients have selected the address. This IP address range is non routable. To gain access to Internet in such a network, network address translation or IP masquerading are required.

3.3.2 Alternatives to AutoIP

Stealing an IP address . This method is similar to the technique used by hackers to enter a private network. It consists in listening to the network and in getting a maximum of information concerning this network: Free IP addresses, proxy information, gateway, routing parameters.

This method is highly dependent from the physical layer: a non switched ethernet network (hub network) replicates all incoming packets on all ports. It is possible to gather information on neighbor just by listening on the wire. On non-switched network, data is replicated only on the destination port. The listener only receives broadcast and multicast traffic. This is not sufficient to retrieve information about subnet mask, free IPaddresses. Moreover it is not possible to emit packets on this wire without getting first an IP address. This method is only suitable for non switched ethernet or 802.11 environments, but it is not suitable for a switched network (Gigabit, 100Tx Full Duplex, ATM,...)

However, sniffing network wires in promiscuous mode can allow IP appliances guessing some network configuration such as Web Proxy, DNS and other parameters, but cannot be safely used in real-world conditions.

Inferring the subnet. In some of the schemes described above, the IP appliance just obtains an IP address during the configuration phase. In order to be able to route packets outside of the subnet, it needs the Gateway address and subnet mask. To solve this problem, we have designed an algorithm, which will guess the subnet without any prior knowledge.

let's define as a notation the direct and indirect broadcast addresses for a given subnet:

$$BC1 = IP \text{ AND } SubnetMask \qquad \textbf{(EQ 1)}$$

$$BC2 = (IP \text{ AND } SubnetMask) \text{ OR } (\text{NOT } SubnetMask) \qquad \textbf{(EQ 2)}$$

Where *IP* is the IP address of device of the subnet, and *SubnetMask* is the mask associated to the subnet. Both are expressed as 32 bits unsigned integers.

We use a specificity of the Internet Protocol which imposes to all devices in a given subnet to reply to a message which is addressed to its subnet's broadcast address. More specifically, when an *ICMP Echo request* is sent to *BC1* or *BC2*, the emitting device will receive an echo packet from all the devices of the subnet.

Given a device with an IP address, it is possible to detect its subnet using the following heuristic:

Starting with a 30-bits subnet mask (255.255.255.252) down to a 8-bits mask (255.0.0.0), we will attempt each possible subnet mask value. For each considered mask, an *ICMP Echo Request* will be sent to the *BC1* and *BC2* IP addresses. If at least 2 answers are received from both of the considered *BC1* and *BC2*, the considered mask is the correct one, hence the subnet can be deduced from the mask.

3.3.3 Network configuration: summary

The table hereunder summarizes the alternatives for appliance network configuration. We mention explicitly the issue of persistence - both for addresses and for names - since it impacts the service discovery mechanism.

	Pre-configuration	**Client-based static configuration**	**Client-based dynamic configuration**	**AutoIP**
Advantages	controlled stable environment	persistence of addresses	persistence of names	
Drawbacks	no support for new devices	address server needs initial administration, and for every new device introduction	no persistence of addresses	no persistence problematic when 2 networks merge
Issues	not applicable for appliances	requires a server	requires a server	

TABLE 1. **summary comparison of network configuration models**

3.4 Service Discovery

As mentioned earlier, the service discovery issue brings two other issues: *advertising* and *filtering*. Advertising may be achieved in two different ways: either the appliance is in charge of exposing its capability to others, or it only exposes its capability to a process which will act as a directory. Both approaches have advantages and drawbacks, that are summarized in the following table:

	Individual advertising	**Centralized advertising**
Advantages	No infrastructure	Easy to filter
	No need for identified services	Consistency
	No configuration and directory maintenance	Efficiency
	Spontaneity	Lightweight client
	No bootstrap	
Examples	SLP, uPnP, SDP (bluetooth)	SLP, uPnP, Jini

TABLE 2. **Comparison between distributed and centralized service discovery**

3.4.1 Service Location Protocol

Service Location Protocol [12] has been defined by *Sun Microsystems* to address service discovery requirements. It is now a well documented IETF standard. Unlike other challengers, the most remarkable thing about SLP is that it is designed to operate both in server mode, or in peer mode. A SLP service does not require configuration to begin searching for a service or advertising himself against the network.

Each SLP service should advertise its own availability. All services are supposed to be well known from others (preexisting knowledge about how the service interferes is required) and no mechanism to exchange profile and methods is available.

If no SLP directory is available, network layer should provide multicast capabilities. Standard *relative multicast IP addresses* have been reserved and registered to IANA and are supposed to be well known from all SLP-aware services[1].

Standardized profiles for SLP advertisements are available from IANA Web site in SRVLOC section [17].

3.4.2 uPnP and SSDP

Universal Plug-and-Play (UPnP) and *Simple Service Discovery Protocol* (SSDP) are protocols driven by Microsoft [8]. SSDP provides a mechanism to advertise and discover services without any prior configuration. The implemented mechanism has been designed to be as simple as possible.

SSDP uses the HTTP protocol as a transport layer for discovery, announcement and query messages. To be able to work in neutral network environment, SSDP relies on a special multicast relative address standardized by IANA to reach all networked devices.

1. Some networks (PPP, Bluetooth) don't provide multicast mechanism.

Services appears as a couple of an *Universal Resource Identifier* (URI) and a *Unique Service Name* (USN). *"ssdp:"* is used as the URI protocol part and the service name is used to describe the given service. This naming scheme is similar to the one used in SLP.

3.5 Configuration in volatile environments

A *volatile environment* is a neutral network in which devices can enter or leave at any time. This adds constraints to the appliance network, compared to a more stable network. The most immediate additional requirements are[1]:

- the infrastructure should not rely on a dedicated appliance to be functional: this is called here "serverless infrastructure".
- no alea (e.g. disparition of a appliance) should damage the integrity of data.
- the cost of infrastructure should be minimal on the infrastructure resources. This aspect may appear as secondary, but the overhead due to the update of structural information is likely to impact the performance of the ecosystem.

Some of the current network solutions, mainly in home environments, already present characteristics that are useful for volatile networks: No server, infrastructure not always on, minimal cost on small appliances.

- **NetBios** (from IBM) and Wins (from Microsoft) have implemented election mechanisms to attribute special roles to elected hosts. These protocols resist to unattended shutdown and network issues.
- The **WINS** service is a name service protocol which implements some requirements for volatile networks: server less infrastructure, challenge to choose master browser, client and server in the same service. But WINS doesn't scale well compared to worldwide implementation of DNS. Moreover it isn't compatible from a protocol perspective with DNS (Microsoft operating systems integrates a gateway between DNS and WINS).
- **DHCP** is precious (as mentioned in paragraph 3.2) but it requires use of a dedicated server.mini-DHCP is a small DHCP server [5] implemented everywhere, able to be chosen by peer to take DHCP server role. This server would disappear when a real DHCP server becomes available. This would partially solve the IP attribution problem.

4 Conclusions & future work

We have identified the main issues that arise when IPv4 appliances connect together. We have pointed out the techniques used in "managed networks" that do not scale down for these appliance networks. We have presented a selection of viable solutions. However, none of those existing "standard" solutions does solve the problem as a whole.

AutoIP resolves the most basic issue of attribution of IP address and subnet. But the appliance networks requires a dedicated infrastructure service to update the other parameters: gateway, proxy, mail server, etc. We propose to handle these infrastructure services in the same way as the applicative services offered by the appliances themselves. We listed alternatives and current solutions for implementing a directory for these solutions, and we identified SLP as a valid choice.

On the other side, an appliance joining a network should become aware of the services already available from the infrastructure. Heuristic methods such as WPAD [10] are good examples of how it could be done.

Finally, the frequent connections and disconnections of the appliances in such a network raise stability issues: we have listed a few techniques - leader election, absence of dedicated server - that could be adopted to solve it.

1. [18] dedicates a chapter to the resilience of a ecosystem.

Currently, we have been exploring the alternative solutions individually. In the near future, we plan to prototype IP-based appliance networks, relying on heterogeneous transport layers such as Bluetooth or 802.11, and integrate the miscellaneous elements listed above.

A. Definitions

Manual: a manual network should be entirely configured by a human administrator

Friendly: a friendly network provides special server dedicated to infrastructure. These servers will provide configuration information.

Neutral: a neutral network contains no specialized devices able to answer other appliances configuration needs. Each device is able to get up, configured and running by it self.

Volatile . a volatile network is a neutral network in which every thing may append automatically: incoming of new hosts, departure of other hosts. In such a network, all appliance should support resilience.

Service . Services are abstract representations of functions implemented by appliances.

Server . It is a device dedicated to run one or several services. When ever a network is scaled, moving a service on a dedicated server improves network availability and responsiveness. In this case a server becomes part of the infrastructure.

B. References

[1] M. Hattig - Intel Corporation. Zeroconf Requirements. *IETF draft*, 20 November 2000.

[2] Be At Home - http://www.beathome.com

[3] Bill Croft - Standford University and John Gilmore - Sun Microsystems. Bootstrap Protocol. RFC 951, IETF, September 1985.

[4] R. Droms - Bucknell University. Dynamic Host Configuration Protocol. *IETF protocol*, March 1997.

[5] B. Aboba, “The mini-DHCP server”, IETF Internet-draft, October 2000

[6] S. Ramanathan, D. Caswell, S. Neal, Äuto-Discovery Capabilities for Service Management: An ISP Case Study”, HP Laboratories technical report HPL-1999-68, May 1999

[7] Enhancement to DHCP for automatic IP settings when no DHCP is available. *http://www.upnp.org/resources/UpnP-bkgnd.htm*.

[8] Yaron Y. Goland, Ting Cai, Paul Leach, Ye Gu, Microsoft Corporation, Shivaun Albright, and Hewlett-Packard Company. Simple Service Discovery Protocol/1.0. draft v1.03, IETF, 28 October 1999.

[9] Cuneyt Akinlar, David Braun, and Sarit Mukherjee - Panasonic Research. Zero-Configuration Routing Information Protocol. *IETF Draft*, 15 August 2000.

[10] Paul Gauthier - Inktomi Corporation, I. Cooper - Equinix, J. Cohen - Microsoft Corporation, Inc M. Dunsmuir - Realnetworks, and Inc. C. Perkins - Sun Microsystems. Web Proxy Auto-Discovery Protocol. Draft, 15 November 2000. Submission to WREC Working Group at IETF.

[11] Finlayson, Mann, Mogul, and Theimer - Stanford University. A Reverse Address Resolution Protocol. *IETF protocol*, June 1984.

[12] J. Veizades - @Home Network, E. Guttman - Sun Microsystems, C. Perkins - Sun Microsystems, and S. Kaplan. Service Location Protocol. RFC 2608, IETF, June 1999.

[13] Inc S. Alexander - Silicon Graphics and R. Droms - Bucknell University. DHCP Options and BOOTP Vendor Extensions. RFC 2132, IETF, March 1997.

[14] Suart Cheshire - Apple Computer. Dynamic Configuration of IPv4 link-local addresses. *IETF Draft*, 24 November 2000.

[15] C. Perkins - Sun Microsystems and E. Guttman - Sun Microsystems. DHCP Options for Service Location Protocol. RFC 2610, IETF, June 1999.

[16] M. Patrick - Motorola BCS. DHCP Relay Agent Information Option. RFC 3046, IETF, January 2001.

[17] Protocol Numbers and Assignment Services, SVRLOC Template, IANA web site http://www.isi.edu/in-notes/iana/assignments/svrloc-templates/

[18] HP Labs Grenoble staff- “HPLG research strategy” - HPL report, submitted September 2001