

Automatic Service Discovery in IP Networks A Developer's Perspective

Brian Bloniarz (brian.bloniarz@streamunlimited.com)
Klaus Kuehnhammer (klaus.kuehnhammer@streamunlimited.com)

StreamUnlimited Engineering GmbH
High Tech Campus Vienna
A-1102 Wien, Gutheil-Schoder-Gasse 10
Austria
www.streamunlimited.com



Abstract

Nearly every network application needs to discover devices that inhabit the local network. Over the last years, several competing standards for enabling automatic service discovery for embedded devices have emerged to this end: SSDP (as used for UPnP), mDNS-SD (Apple's Bonjour) and, to some extent, Bluetooth pairing. Between them, these standards span a range of network technologies and privacy expectations. For certain applications, their scope must be confined to the home network, while for others it encompasses a corporate network or the entire Internet. This paper will give a comparative overview of these protocols from an embedded software developer's point of view: ease of implementation, existing reference designs, confidentiality and authentication, and scalability. We will also discuss the conflicting goals of universal networking and digital rights management considerations in the context of self-configuring home networks.

Introduction and Problem Definition

Building a networked device that's easy to use and easy to configure can be deceptively difficult. To many end users, a network is little more than a port in the back of the computer that brings in Internet. To a system implementer, networking is a jumble of technologies accumulated over the course of 30 years, encompassing an endless variety of possible network setups and host configurations. This problem is particularly noticeable for connected consumer devices – Ethernet and 802.11 wireless are quickly becoming the preferred interconnect for devices like printers, networked storage, audio/video extenders, even DVD players, digital cameras and mobile phones. Here, a typical consumer expects plug-and-play configuration and interoperability. As setups like these become more commonplace, there will be an increasing focus on bridging the gap between the consumer's usability expectations and the reality of TCP/IP networks.

Technologies for automatic networking fall under the heading of Zero Configuration Networking. Zero Configuration Networking is typically concerned with 3 things:

- 1. Addressing.** This concerns the acquisition of an IP address by the device for communication in the local network. In the absence of a DHCP server, link-local IP addressing can allow two hosts in the LAN to communicate without any configuration. Link-local address allocation has been standardized by the IETF Zeroconf Working Group [1] and has been widely implemented in the PC and embedded realm.
- 2. Name Resolution and Service Discovery.** Once a host has an address, it can communicate with other hosts through their IP addresses. Rather than requiring the user to type in an IP address, we require that our devices address other hosts using a human-readable friendly name – this is called name resolution. We also require that our devices can discover hosts that offer a given service – this is called service discovery. These two interconnected requirements are the focus of this paper.
- 3. Services.** This concerns the actual communication that occurs once devices can find and talk to each other. There are a wide variety of protocols at this level (HTTP,

UPnP, SMB, etc.), and they need not be zero-configuration specific. Discussing available service standards is a huge topic, and detailed information is outside the scope of this paper.

Examples

As motivation, we present some example scenarios which can benefit from automatic service discovery.

- **Network Printer.** A printer on the network should advertise itself to allow easy configuration on the PC. Users can scan for printers and choose from a list rather than typing in an IP address. Apple's DNS-SD standard (part of Bonjour, described later in this document) seems to be widely implemented by printer manufacturers to advertise printers.
- **Media Server/Media Client.** A device connected to the network has stored music, video, or other media. It wishes to allow clients to connect and stream media over the network. Universal Plug and Play (UPnP, described later) is the most common open standard here.
- **Network Attached Storage.** A small, network connected device offers hard drive storage on the network for all hosts to access. Microsoft's SMB protocol (Server Message Block) is the most common in practice. SMB is not publicly specified, but it has been reverse engineered and compatible implementations exist. SMB is also over complex because of legacy, having been originally developed for NetBIOS networking under MS-DOS.
- **Configurable network devices.** A network device (e.g. home router) has an embedded web server which serves a setup page. The user should be able to scan for these devices on the PC and open the configuration page in their web browser. Such support is built in to Microsoft Windows XP through Universal Plug and Play. Apple computers have built-in support for DNS-SD; free DNS-SD software is also available for Windows.
- **Embedded network devices in development.** A networked device is in the development or testing phase. It offers some "debug-only" services over TCP/IP, such as a trace log, command line, or software upgrade service. Such services are traditionally accessed by IP address, or with a custom in-house service discovery protocol. A simple standardized service discovery protocol like DNS-SD can be implemented in few lines of code, and has a preexisting base of interoperable software libraries on the PC side.

Technical Approach

All service discovery protocols share some technical details. Multicast UDP is usually used to send requests visible to multiple hosts on the network. Note that UDP packets have a small implementation-defined size limit, so fragmentation must be supported in the protocol. This can complicate protocols and lead to slower discovery.

All protocols must limit their discovery scope, so that neither the user nor the network is burdened with search results from thousands of kilometers away. There is unfortunately no a priori scope which is appropriate for all uses – the user might want to discover devices in the same room, in the same building, or something more esoteric. Most protocols pick an arbitrary scope limit. For example, the mDNS protocol uses a multicast address in the link-local multicast range 224.0.0.0 to 224.0.0.255. Packets from this address range will never be forwarded by a router, so they stay in the local LAN segment. The IP Time-To-Live (TTL) field is typically also set to a low value, to prevent packets from being forwarded to the Wide Area Network. UPnP SSDP uses a TTL of 4.

Because multicast traffic places an additional burden on the network, all protocols try to limit the amount of multicast traffic generated by the host. This takes the form of time delays and traffic throttling; this can slow down the discovery process. The amount of traffic generated in a home network is insignificant; scalability to larger networks is a major concern.

mDNS & DNS-SD

Multicast DNS (mDNS) enables name resolution for zero configuration networks. It uses standard DNS packet formats, sent to a link-local multicast address instead of a DNS server [2]. This provides device friendly names, even in the absence of a DNS server. mDNS operates in the special top level domain ".local." so it can peacefully coexist with an existing DNS server.

DNS Service Discovery (DNS-SD) is a complementary technology [3] which extends the standard DNS protocol (RFCs 1034, 1035). It describes a way of structuring normal DNS requests and replies to describe services that the host offers. Together with mDNS and Zeroconf link-local addressing, it forms the core of the family of protocols Apple promotes as Bonjour (formerly Rendezvous).

Bonjour allows IP address allocation without a DHCP server, translation between names and addresses without a DNS server, and discovery of named services on the local link without a directory server. It has become the de facto standard for automatic printer discovery and installation, and has seen adoption in other areas like streaming devices, network cameras, and Network Attached Storage.

DNS Service Discovery is not an independent protocol – it is simply a way of structuring normal DNS requests and replies. It can therefore be implemented on top of standard unicast DNS or mDNS. For efficient unicast DNS service discovery, 2 extensions to DNS are required. The first, DNS UPDATE [4], allows a host to dynamically alter the data stored in a DNS server. The second, DNS Long Lived Queries [5], a draft standard from Apple, adds notifications sent by the DNS server when records change. If these extensions see widespread implementation, they would allow DNS-SD to scale beyond small networks and beyond the link-local multicast domain. To the knowledge of the authors, comparable scalability has not been provided by any other zero configuration service discovery protocol.

mDNS & DNS-SD are available on practically all desktop operating systems. For embedded use, Apple provides an open source reference implementation under the Apple Public Source License [6].

mDNS & DNS-SD are not official Internet Engineering Task Force standards. They originated from an IETF standard-draft, and have been widely implemented by Apple and others. A related protocol for DNS over multicast, called Link-local Multicast Name Resolution (LLMNR), is currently being standardized by the DNSEXT working group of the IETF. LLMNR was intended solely as a replacement for DNS when a name server is not available, in contrast to mDNS which was designed to operate even when conventional DNS is available. For this and other reasons, LLMNR is not suitable for service discovery. The LLMNR standardization process seems to be stalled [7], and mDNS has seen much wider adoption in practice.

UPnP SSDP

Universal Plug and Play (UPnP) is a protocol suite focused on trouble-free connectivity of devices in a home network. It defines services for media servers & clients, media control points, automatic router/gateway configuration, home automation and more [8].

Service discovery in UPnP is handled by SSDP, the Simple Service Discovery Protocol [9]. SSDP is a competitor with mDNS, and can be used independently of UPnP [10]. SSDP uses HTTP as a base, sent over UDP multicast. This approach is disadvantageous for the highly embedded developer, where implementation of an RFC2616 compliant HTTP parser has proved to be memory intensive and error-prone when compared to the relative ease of mDNS parsing and implementation.

SSDP does not use link-local multicast. Instead, it uses an administratively scoped multicast address (239.255.255.250). Administratively scoped multicast packets will be forwarded by routers within a group or administration, but blocked by routers on the boundary of the administration [11]. Given properly configured routers, this allows SSDP can provide a wider

and more logical scope than protocols which use link-local multicast.

There are several high quality UPnP implementations on the market, as well as open-source offerings from Intel [12] and others. Independent implementations of SSDP are less common.

Security

Zero-configuration networking is at odds with security, almost by definition. Devices which connect to an unsecured network and automatically establish interconnectivity make easy targets for hijacking, man-in-the-middle attacks, and other forms of abuse. The popularity of wireless networks (which are often unsecured or undersecured) has made this a very real threat.

By security, we mean two things:

1. **Authentication.** Imagine the user connects a network printer to the network, and shortly thereafter sees a "New Printer Found" notification on her PC. The user needs to know that the device named "Alice's Printer" on her PC is really the printer she just connected, and not some hacker performing a man-in-the-middle attack.
2. **Confidentiality.** Alice also requires that no network eavesdropper can see the documents she's sending to the printer.

Authentication and confidentiality cannot be automatically guaranteed without user interaction. At some point, a human being must make an out-of-band confirmation of security. This can be done by showing a hash value on both devices, and requiring the user to confirm that they match. Alternately, the user can type a one-time shared secret into both devices – this approach is used in the Bluetooth pairing scheme [13]. It has the drawback that shared secrets short enough to be convenient for the user are easy to attack [14].

Security is not the concern of the service discovery protocol. Rather, it's usually added at the link level (IPSEC, WPA, etc.) or at the service level (HTTPS, SFTP, DNSSEC, etc.). However, a common service-independent security framework for zero configuration devices is sorely needed, seeing that the security requirements of (for example) a printer and a media client are quite similar. If security is left to the service, this would lead to an explosion of security standards and unnecessary complexity. For this reason, the authors hope that a common framework for service-level security can be standardized.

An additional security issue is Digital Rights Management (DRM). Here, the device must as a matter of policy only interoperate with other devices which have been compliance certified by a central body. This is done to guarantee control over the distribution of copyrighted material like films and music. Examples of DRM standards for connectivity are the Digital Transmission Content Protection (DTCP-IP) and Microsoft Windows Media DRM 10. DRM has the same two security concerns (authentication and confidentiality), but they're negotiated on behalf of the manufacturers rather than the user. For this reason, DRM can be done without user intervention, using shared secret keys or secret algorithms hidden in the device firmware.

Conclusion

Ethernet and 802.11 wireless are a very attractive way to interconnect devices – they're fast, general, and widespread. Their generality can also be a weakness, leading to configuration nightmares that more specialized standards like USB, IEEE1394, HDMI, Bluetooth and Wireless USB avoid. Automatic service discovery is one important step in bridging this usability gap. Simple, open protocols like mDNS and UPnP SSDP are necessary for true interoperability. New refinements will be necessary, especially in the area of scalability and security.

References

- [1] IETF Zeroconf Working Group. <http://www.zeroconf.org/>
- [2] Cheshire, S., Krochmal, M. Multicast DNS. Internet Draft, available from <http://files.multicastdns.org/draft-cheshire-dnsexext-multicastdns.txt>, 2005.

- [3] Cheshire, S., Krochmal, M. DNS-Based Service Discovery. Internet Draft, available from <http://files.dns-sd.org/draft-cheshire-dnsext-dns-sd.txt>, 2005.
- [4] Vixie, P., Thomson, S., Rekhter, Y., Bound, J. Dynamic Updates in the Domain Name System (DNS UPDATE). RFC 2136, available from <http://www.ietf.org/rfc/rfc2136.txt>, 1997.
- [5] Sekar, K., Cheshire, S., Krochmal, M. DNS Long-Lived Queries. Internet Draft, available from <http://files.dns-sd.org/draft-sekar-dns-llq.txt>, 2005.
- [6] Bonjour, <http://developer.apple.com/darwin/projects/bonjour/>
- [7] Wasserman, M., 2005. Summary of the LLMNR Last Call. <http://www1.ietf.org/mail-archive/web/ietf/current/msg37740.html>
- [8] UPnP Forum. <http://www.upnp.org/>
- [9] Golan, Y, Cai, T., Leach, P., Gu, Y., Albright, S. Simple Service Discovery Protocol/1.0, Operating without an Arbiter. http://www.upnp.org/download/draft_cai_ssdv_v1_03.txt, 1999.
- [10] Golan, Y., ZeroConf, SLP, SSDP & UPnP. <http://www.oreillynet.com/pub/a/wireless/2002/12/20/zeroconf.html>, 2003.
- [11] Meyer, D. Administratively Scoped IP Multicast. RFC 2365, available from <http://www.ietf.org/rfc/rfc2365.txt>, 1998.
- [12] Intel Software for UPnP Technology. <http://www.intel.com/technology/upnp/>
- [13] Bluetooth Core Specification v2.0 + EDR. <https://www.bluetooth.org/spec/>
- [14] Shaked, Y., Wool, A. Cracking the Bluetooth PIN. Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys 2005), available at <http://www.eng.tau.ac.il/~yash/shaked-wool-mobisys05/index.html>, 2005.