How does Zeroconf compare with Viiv/DLNA/DHWG/UPnP?

Taking the broadest view, the stories of Zeroconf and UPnP began with the same high-level observation: "Why is IP networking so hard? Why can't it be like AppleTalk was in 1986, where you just plug things together and it all works automatically?"

From this initial observation, the development of Zeroconf and UPnP then proceeded in different directions.

Zeroconf is a very precisely defined, compact trio of industry-standard network technologies that work in concert to provide a reliable foundation for effortless networking. The word *foundation* in that sentence is significant. What vendors choose to build on top of that reliable foundation is largely up to the vendor in question. Zeroconf is mature, stable, and in certain product segments it is already very widely adopted. For example, virtually every network printer sold today, from every printer manufacturer, implements the Zeroconf protocol trio. In fact, if you install Apple's free Bonjour for Windows on your Windows PC, you can discover for yourself how many of the network printers on your own company network already implement the Zeroconf protocols. In other product categories, like network-attached cameras, Zeroconf offers obvious benefits that leading vendors like Axis have been quick to recognize. There's no network product that doesn't benefit from incorporating the solid Zeroconf foundation, improving the usability and reliability of the product, resulting in simpler documentation, lower support costs, lower product returns, and higher customer satisfaction.

In contrast, UPnP is an open-ended collection of device-specific protocols. As each new device type comes along, the UPnP Forum creates a new working group to talk about how that device type should work. UPnP claims adoption in many network devices, but in reality the adoption is more in name than in spirit. For example, many printers claim to implement UPnP, and indeed examining the network with a packet sniffer will show that the printer is sending UPnP SSDP packets, but Windows doesn't actually use those packets to discover, configure, or use the printer. This lack of any actual useful customer benefit is a common phenomenon in the UPnP world.

A commonly asked question is, "Why should we adopt Zeroconf instead of UPnP?"

Phrased like this, the question is mis-stated in two ways.

The first is that it's not an either/or choice. There's no reason a product can't do both. There's no reason a product can't reap the tangible usability and reliability benefits of Zeroconf, while at the same time satisfying the marketing department's requirement to put a sticker on the box saying "Supports UPnP". In fact, as explained below, the self-assigned link-local addressing layer is identical in Zeroconf and UPnP, so any product doing UPnP link-local addressing has already implemented the first of the three Zeroconf layers anyway.

The second way the question is mis-stated is that "Zeroconf" and "UPnP" are not directly comparable concepts. Zeroconf is a three-layer foundation which provides dependable IP networking in the absence of configuration information and infrastructure. UPnP is an organization, actively involved in the ongoing process of producing protocol specifications. Talking about "adopting UPnP" is like proposing that it would be a good idea to "adopt ISO", or IEEE, or ANSI. When someone describes a product as implementing some protocol documented in an IETF RFC or Internet Draft, they say that it "implements RFC xxxx", not that it "implements IETF" or that the company is "adopting IETF."

When people talk about "adopting UPnP" they don't usually mean that their company has blindly committed to implement and support anything and everything that might come out of the UPnP organization. What they really mean is usually one of two things. Sometimes they mean that they have decided to adopt certain specific UPnP protocols, and they're assuming by context that it's clear which specific UPnP protocols they're talking about. Other times — and sadly this is all too common — the person making the statement about "adopting UPnP" is a high-level executive who knows nothing about network protocols. All he knows is that networking products are too hard to use, and he's been informed that saying "we adopt UPnP" will make the problem go away. The technical minutiae of deciphering what that statement actually means in concrete terms are beneath him — such boring details will be handled by some lowly engineer.

Some people in the Viiv/DLNA/DHWG/UPnP community encourage this impression of an "either/or" choice by portraying Zeroconf and UPnP as incompatible competing technologies, perhaps mistakenly, or perhaps intentionally to create confusion. The truth is that Zeroconf and Viiv/DLNA/DHWG/UPnP (which from now on I'll just refer to as

"UPnP") only appear to be competing if you view them at a very superficial level. It would be more accurate and helpful to describe them as complementary.

Zeroconf and UPnP are not really competing or contradictory; they're more orthogonal, in the sense of "unrelated". Zeroconf went in a horizontal direction, creating a solid foundation suitable for all IP-based networking protocols. The UPnP Forum went in a vertical direction, creating specific vertical solutions aimed at solving specific vertical problems. For example, while the UPnP Forum is currently working on defining a new printing protocol (a laudable and worthy goal) Zeroconf created a reliable foundation upon which you can run *any* past, present, or future IP-based printing protocol. Zeroconf automates the TCP/IP configuration aspects of today's network printers, making it easy to discover those printers and establish a TCP connection to print to them using the LPR protocol or IPP; Zeroconf itself does nothing to remedy any deficiencies in the LPR protocol or in IPP. In an ideal world, Zeroconf would provide the dependable TCP/IP networking foundation that all networked devices so desperately need; other organizations like the UPnP Forum could then work on designing new application-layer protocols to supplement or supplant today's application-layer protocols where necessary. For example, just about every network printer today implements the full three-layer Zeroconf foundation (IPv4LL + mDNS + DNS-SD), and supports LPR printing. However, the LPR printing protocol lacks feature negotiation and has no status back-channel from the printer to tell the user about paper jams and so forth, and hence the time is ripe for some other, more capable, printing protocol to come along and establish itself as the new de jure or de facto printing standard.

It's helpful at this stage to examine Zeroconf's three-layer foundation, and compare that with the equivalents UPnP offers:

	Zeroconf	The UPnP Forum's corresponding foundation layers
Addressing	The IETF Zeroconf Working Group defined RFC 3927, "IPv4 Link-Local Addressing".	**UPnP's link-local addressing is exactly the same.** To be precise, the UPnP Forum originally adopted an early draft of the RFC, draft-ietf-zeroconf-ipv4-linklocal-01.txt, as can be seen in this snapshot of the UPnP.org specification page as of April 2004 (courtesy of the Internet Archive WayBack Machine). The draft-01 version lacks some refinements, but is completely compatible with the final protocol standardized by the IETF. The UPnP Forum later pulled the link from their web site, so it's unclear whether they still feel they have any useful contribution to make regarding the crucial question of how a device is supposed to get itself an IP address.
Naming	Zeroconf uses Multicast DNS to translate host names to addresses in the absence of a conventional unicast DNS server.	**UPnP has no equivalent to mDNS.** Since the UPnP Forum was focussed on creating new vertical solutions, worrying about improving the usability of existing legacy tools that use host names (ping, telnet, ftp, ssh, etc.) was simply not a problem that UPnP sought to solve. In a UPnP world, if a developer wants to use ssh to connect to a UPnP device under development, the only way to do that is by typing in a raw dotted-decimal IP address, but since UPnP devices use randomly-assigned IPv4 Link-Local Addresses, and since you generally don't know what random address a device has picked for itself, trying to use raw addresses instead of a name lookup protocol is just not practical.
	Building on its Multicast DNS	**UPnP has no dependable Service Discovery protocol.** For UPnP Forum members, the issue of Service Discovery, like the other foundation layers, was not a high priority. It was merely a minor issue that had to be dispensed with before they could get to the more interesting work designing application-layer protocols. For this layer, as with the addressing layer, the UPnP Forum chose to adopt an unfinished IETF draft. However, unlike the addressing layer, where the design was eventually completed and published as a Standards-Track RFC, in the case of SSDP, the IETF draft was abandoned by its authors and never finished. The last update to the SSDP (Simple Service Discovery Protocol) draft was draft-03 in October 1999. A little technical analysis reveals some of the shortcomings that led to the proposal

Discovery	foundation, Zeroconf adds DNS Service Discovery to discover what services are available on the network, providing the dependable Service Discovery that's the vital for a good user experience. For Zeroconf, Service Discovery is its *raison d'être*.	being abandoned:
		- Zeroconf's mDNS-SD is more efficient on the network, using techniques like duplicate query suppression and known-answer suppression.
		- Zeroconf's mDNS-SD is resilient to extended periods of packet loss (often a severe problem on 802.11 wireless networks); SSDP is not.
		- Zeroconf's mDNS-SD is simple and doesn't take a lot of code — it's been implemented in devices with as little as 16 kilobytes of flash-based firmware, like the SitePlayer serial interface board.
		- Zeroconf's mDNS-SD uses multicast for both requests and responses, allowing it to function even in the face of many common home network misconfigurations, such as non-matching subnet numbers.
		Solutions to the many problems with SSDP were never found, and the abandoned October 1999 draft on the UPnP Forum web site is still full of little editorial "note to self" comments in square brackets. One of the problems that the authors recognized was that if you had more than about ten devices on a network using SSDP, it would rapidly become far too chatty, and one of these little "note to self" comments points out this deficiency:
		`6.1. Problem Statement`
		`A mechanism is needed to ensure that SSDP does not cause such a high level of traffic that it overwhelms the network it is running on.`
		`6.2. Proposed Solution`
		`[Ed. Note: We have a proposed solution but it is still a bit rough, so we will be publishing to the SSDP mailing list for further discussion before including it in the draft.]`
		It's also noteworthy that Yaron Goland, designer of SSDP and Microsoft's original lead UPnP architect, gave an endorsement for the "Praise for this Book" page in O'Reilly's Zero Configuration Networking: The Definitive Guide.

Application	Zeroconf is completely agnostic with regard to application-layer protocols. Whatever application-layer protocol you currently use today, or have in mind for the future, the three-layer Zeroconf foundation provides the reliable IP networking you want.	Application-layer protocols are UPnP's primary focus. If UPnP already defines the particular application-layer protocol your device needs, then you should evaluate that application-layer protocol and make an informed decision about whether you want to use it. If you plan to use some other existing application-layer protocol, or design your own new application-layer protocol, then UPnP has nothing to offer you (except perhaps as a possible venue for forming a new committee to talk about the problem).

The sensible conclusion is that device designers wanting to make reliable products should use Zeroconf as their foundation, and then on top of that, run whatever application protocols are appropriate (possibly including UPnP application-layer protocols, if UPnP offers any that are a sensible choice).

In summary:

- Implementing *IPv4 Link-Local Addressing* is an easy decision, because both Zeroconf and UPnP are identical at that layer.
- Implementing *Multicast DNS* is an easy decision, because UPnP doesn't have any alternative to offer.
- The only area of overlap is the service discovery layer, and it's important to remember that mDNS-SD and SSDP are not mutually exclusive. Device designers don't have to choose between one and the other. If a device offers certain services via certain application-layer protocols, it can advertise those services via both SSDP and via mDNS-SD, and let the marketplace determine which works most reliably. There's really no sense in making a great product, and then have it die in the marketplace, as so many have over the last five years, because SSDP — an insignificant part of what the product does — let it down.

The related conclusion here is that if you're not implementing some existing UPnP application-layer protocol, but instead you're building a new product or designing a new application-layer protocol of your own, then deciding to build your new application-layer protocol on a UPnP foundation would be a truly nonsensical decision, because UPnP isn't a foundation; it's an organization devoted to producing a few narrowly defined vertical solutions in specific problem spaces.

Page maintained by Stuart Cheshire