

Network Protocol Project Details

Expectations

- Learn the Protocols:
 - Basic history
 - How they are used today
 - Protocol design
 - Packet structure
 - Message Flow
- Learn the Toolsets
 - Install SDKs
 - Build, run, and debug samples
 - Learn network monitoring tool(s)
- Design Tests
 - Choose one or more aspects of efficiency
 - Design tests to examine chosen aspects of efficiency
 - Implement and run tests for all 3 protocols
- Report
 - Show comparison of efficiency among the protocols
 - Study results and suggest ways to improve efficiency

Learning

Network connected devices (NCDs) are growing in leaps and bounds and will soon become ubiquitous in our lives as consumers and workers. Everything from our appliances to our entertainment systems will be connected and controllable 24x7 from any client device anywhere on the planet. This explosion in connectivity requires that we think strategically about network bandwidth usage, security, power management, remote access, interoperability, and more. In this project we will explore 3 of the primary protocols used by NCDs today:

- Device Profile for Web Services (DPWS)
- Universal Plug and Play (UPnP)
- Bonjour

Below is a detailed list of sources for learning about the protocols and associated APIs and tools. The list may appear a bit overwhelming but actually most of it is fairly short specs and articles and you can often gloss over detail until you actually need it in practice. In short, read through it with the goal of understanding the concepts rather than every bit and byte.

To begin with there are some networking fundamentals that you will need to understand for all 3 protocols. I have included some Wikipedia links here as a starting point. If you have access to Andrew

Tannenbaum's excellent text book "[Computer Networks](#)" then I would recommend you read the sections on these base protocols. These sections of the book are clear, concise, and have exactly what you need. No need for the latest edition. The book also has a web site <http://authors.phptr.com/tanenbaumcn4/> that includes an excellent section entitled "Web Resources" that has links to lots of good tutorial information online.

- Internet Protocol (IP)
http://en.wikipedia.org/wiki/Internet_Protocol
- Transmission Control Protocol (TCP)
http://en.wikipedia.org/wiki/Transmission_Control_Protocol
- User Datagram Protocol (UDP)
http://en.wikipedia.org/wiki/User_Datagram_Protocol
- IP Multicasting
http://en.wikipedia.org/wiki/IP_multicast

For DPWS and UPnP you will need some more fundamentals:

- Hypertext Transfer Protocol (HTTP)
<http://en.wikipedia.org/wiki/Http>
- Extensible Markup Language (XML)
<http://en.wikipedia.org/wiki/Xml>
- Simple Object Access Protocol (SOAP)
<http://en.wikipedia.org/wiki/SOAP>

You will also need to download and install the Windows SDK:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=c17ba869-9671-4330-a63e-1fd44e0e2505&displaylang=en>

DPWS

For DPWS you will need some basic knowledge of Web Services:

http://en.wikipedia.org/wiki/Web_services

and some basic understanding of DPWS itself:

<http://en.wikipedia.org/wiki/Dpws>

and the core Web Services specs upon which DPWS is built:

DPWS: <http://specs.xmlsoap.org/ws/2006/02/devprof/devicesprofile.pdf>

Web Services Addressing: <http://www.w3.org/Submission/ws-addressing/>

Web Services Description Language: <http://www.w3.org/TR/wsdl>

Web Services Discovery: <http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf>

Web Services Metadata Exchange: <http://xml.coverpages.org/WS-MetadataExchange.pdf>

Next step is to understand the API and toolset on Windows. A good starting point is on one of our developer's blogs where he has a good introduction to the API:

<http://blogs.msdn.com/dandris/archive/2007/11/09/wsdapi-101-part-1-web-services-from-10-000-feet.aspx>

There is also a whitepaper on MSDN with some good intro material:

<http://msdn.microsoft.com/en-us/library/cc980021.aspx>

After that read through the official documentation on MSDN:

[http://msdn.microsoft.com/en-us/library/aa826001\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa826001(VS.85).aspx)

There is also information in there about the API samples and how to install them. These will be the starting point for your tests:

[http://msdn.microsoft.com/en-us/library/bb872383\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb872383(VS.85).aspx)

Note that for all the protocols the APIs and samples allow you to simulate a device on a PC so there is no need for any embedded programming.

UPnP

UPnP differs from DPWS in that it doesn't rely so heavily on an assembly of other specifications so the information is much more concentrated. This first step is the UPnP Device Architecture:

<http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf>

then read through the Windows API documentation on MSDN:

[http://msdn.microsoft.com/en-us/library/aa382303\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa382303(VS.85).aspx)

There is also information there on the samples:

[http://msdn.microsoft.com/en-us/library/aa382324\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa382324(VS.85).aspx)

[http://msdn.microsoft.com/en-us/library/aa382328\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa382328(VS.85).aspx)

The samples can be found at \Microsoft SDKs\Windows\v7.0\Samples\netds\upnp in your Windows SDK installation.

Bonjour

Bonjour is owned and published by Apple but they have a Windows API. You are welcome to do this part on a Mac if you prefer. Bonjour only deals with device discovery so in your comparison you will only be able to include it in the discovery portion of the protocols (i.e. not device control).

To begin with I highly recommend watching a Google Tech Talk given by the protocol inventor Stuart Cheshire. It will give you a good overview in an hour:

<http://video.google.com/videoplay?docid=-7398680103951126462&q=Google+techtalks>

Everything else you will need is at Apple's Bonjour page:

<http://developer.apple.com/networking/bonjour/index.html>

If you are interested there is also a book by Stuart Cheshire:

http://www.amazon.com/Zero-Configuration-Networking-Definitive-Guide/dp/0596101007/ref=sr_1_3?ie=UTF8&s=books&qid=1250554796&sr=8-3

Network Monitoring

Microsoft's NetMon is an excellent network monitoring tool that should do everything you need:

<http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=983b941d-06cb-4658-b7f6-3088333d062f>

There are other network monitoring and simulation tools out there and you are welcome to use whatever you'd like. Some examples:

<http://en.wikipedia.org/wiki/NS2>

<http://www.wireshark.org/>

<http://www.opnet.com/>

Designing Tests

There are numerous aspects of protocol efficiency. Here are some examples:

- Network bandwidth usage – when the protocol is sending messages does it do so efficiently?
- Chattiness – how often do the two sides of the protocol exchange messages?

- Multicasting usage – multicasting is very expensive on a wireless network. Protocols that multicast too much can be problematic in those environments and in fact some managed networks prohibit wireless multicast.
- Power usage – does the protocol's design prevent a device or host from powering down quickly enough when idle (or at all)?
- Implementation costs – most devices are commodity items. How does the protocol design affect the bill of materials on devices (e.g. CPU power, memory requirements)?
- Complexity – complexity can be measured in different ways but it adds to development and maintenance cost, reliability, and error-proneness of implementation.

Some of these can be tested by running code and others are pure research. Sometimes a test is a thought exercise that is written down for review by others and sometimes it is a program, possibly generated from a model. There is no requirement to choose a particular technique for this project. Pick one or more aspects of efficiency that interest you (there may be others not listed here that you find). Design and run the tests or do the research on all 3 protocols and produce a comparison report showing the relative efficiency of each.

As a final exercise, use your results and what you have learned of the protocols to suggest possible improvements to make them more efficient. There are no guidelines for this exercise. This is your chance to do some creative thinking.

Questions

I am always available for any questions you may have. The best way to contact me is by email: pshier@microsoft.com. If I find that an email conversation is getting too complex I'll set up a time to talk on the phone.