Multi-protocol Home Networking Applet for HTML5

September 19, 2011
Clarke Stevens c.stevens@cablelabs.com
Home Networking Goal for Cable Television

• Support distribution of commercial and personal media content to any capable device in the home
• Support other networked services as opportunities emerge (e.g. home security, energy management, home health and fitness, etc.)
Home Networking Requirements

• Support existing home networking protocols (e.g. DLNA/UPnP, Zeroconf, etc.)
  ◦ Control interface can be loaded from the Internet
  ◦ Discovery of devices and services on the home network
  ◦ Messaging between the control interface and the home networked devices and services
  ◦ Support asynchronous events
  ◦ Security is key (user can choose what to make accessible)
CableLabs’ Prototype Implementation

- Low-level API in signed Java Applet
- Home networking stacks and user interface code in JavaScript
- User Interface template in CSS
- Remainder of user interface in HTML with JavaScript APIs for interaction with devices and services on the home network
Demonstration

• HTML, CSS web page user interface
• User agent written as a signed Java Applet
  ◦ Applet allows for cross domain interaction
• UPnP and Zeroconf stacks written in JavaScript and calling generic APIs in the User Agent Applet
• User authorizes access for each discovered device
Discovery

• discoveryControl(JSONString protocols) //start discovery
  ◦ Protocols = ‘{
    “upnp”:”upnpDiscoveryCallback”,
    “zeroconf”:”zeroconfDiscoveryCallback”
  }’

• upnpDiscoveryCallback(jsonObject) {}
  ◦ JavaScript routine that is called whenever a UPnP device is discovered or lost

• zeroconfDiscoveryCallback(jsonObject) {}
  ◦ JavaScript routine that is called whenever a Zeroconf service is discovered or lost
Messaging and Events

- `sendRequest(jsonString, upnpCallback)`
  - `jsonString = '{
      "protocol":"upnp",
      "serviceType":"urn:schemas-upnp-org:service:AVTransport:1",
      "uuid":"00000000-0000-1010-8000-5442499C2FE3",
      "action":"#PLAY",
      "body":"…UPNP SOAP Command…"
  }'`

- `upnpCallback(jsonObject) {}`
  - `jsonObject = {
        "protocol":"upnp",
        "serviceType":"urn:schemas-upnp-org:service:AVTransport:1",
        "uuid":"00000000-0000-1010-8000-5442499C2FE3",
        "friendlyName":"BRAVIA XBR-52LX900",
        "response":"…UPnP SOAP Response…",
        "responseCode":"200"
    }`
Security

• User must authorize user agent to run (signed Java Applet)
• User must authorize access for any discovered device or service
• User agent may implement additional security and control measures
  ◦ Authorize high-value content
  ◦ Require link protection for sending content between devices
  ◦ Verify that user has a subscription and the type of subscription
  ◦ Verify that selected content can play on selected device or select an appropriate content format
Revised API with Opera

• Since developing and implementing the described API, CableLabs has worked with Opera to develop a joint API proposal.
• That proposal is now ready for public review and will shortly be submitted to the DAP WG.
• The CableLabs messaging API has been replaced with existing and WIP messaging in HTML5 (e.g. XMLHttpRequest with cross domain functionality).
• CableLabs has implemented this API in a Java Applet implementation as well and will soon release it for use in developing HTML5 support for home networking.
Next Steps

• Work with W3C on standardization and with browser vendors on implementation
  ◦ Opera and CableLabs will formally submit their joint API to the DAP WG and will work as editors of the document
  ◦ CableLabs is providing design information and source code it has created for the applet implementation