Lecture 6

Discovery services II
The Peer-to-Peer approach

Universal Plug and Play

- Microsoft initiative: enables *device and service discovery* and use in an ad-hoc environment. More than 1000 company members of the UPnP Forum.

- "UPnP™ technology defines an architecture for *pervasive peer-to-peer network connectivity of intelligent appliances, wireless devices, and PCs of all form factors."* UPnP Device Architecture 1.0

- Application areas include:
  - computing, printing and networking;
  - consumer electronics;
  - home appliances, automation, control and security;
  - mobile products and IoT.

UPnP features

- The network model is *peer-to-peer*; there is no central control.
- A UPnP device may join/leave a UPnP network, advertise its services, discover and use other devices' services at any time.
- It uses *auto-configuration*: initially, any new device searches for a DHCP server (it must run a DHCP client); if not found, it picks an IP address from the link-local range. An ARP request is issued to check for duplicates.

A. Networking

- Devices get IP addresses by using DHCP, if this service is available; otherwise, they use AutoIP.
- The transport protocol is UDP.
- A device may join and leave (bye-bye) the UPnP network at any moment.

Addressing in UPnP 1/2

- If there is no DHCP, the joining device choose an address using a pseudo-random algorithm, AutoIP, distributed over the entire address range from 169.254.1.0 to169.254.254.255, to minimize the likelihood of collisions when several devices join the network at the same time. This pseudo-random algorithm may be seeded with the MAC address.

- *Address collision detection is an ongoing process* that is in effect for as long as the device is using a link-local address. A device that has auto-configured an IP address must periodically check for the existence of a DHCP server. This is accomplished by sending DHCPDISCOVER messages. How often this check is made is implementation dependent, but checking every 5 minutes would maintain a balance between network bandwidth required and connectivity maintenance.

- While devices supporting dynamic DNS updates can register their DNS records directly in DNS, it is also possible to configure a DHCP server to register DNS records on behalf of these DHCP clients.

Addressing in UPnP 2/2

- IP packets whose source or destination addresses are in the 169.254/16 range must not be sent to any router for forwarding. Instead, the senders must ARP for the destination address and then send the packets directly to the destination on the same link.

- IP datagrams with a multicast destination address and an Auto-IP source address must not be forwarded off the local link.

- Devices and control points MAY assume that all 169.254/16 destination addresses are on-link and directly reachable.

B. Devices, services and control points

- UPnP Device:
  - not necessarily a real physical device;
  - it is a representation of a logical entity;
  - it presents a set of functions and state.

- UPnP Device examples:
  - Media Server
  - Media Renderer
  - Internet Gateway device
  - Printer

The UPnP service

- A service in a UPnP device consists of a state table, a control server and an event server.
  - The state table models the state of the service through state variables and updates them when the state changes.
  - The control server receives action requests (such as set_time), executes them, updates the state table and returns responses.
  - The event server publishes events to interested subscribers anytime the state of the service changes.
- Examples:
  - Connection Manager service: a set of functions that are used to negotiate which protocol to use for communication;
  - Rendering Control (service): a set of functions that change settings like volume, brightness, contrast etc.;
  - Media Renderer: a set of functions to control playback (via the network).

Service description

- The service type;
- The service ID;
- The Service Description file:
  - an action list with actions the service can carry on,
  - a service state table that contains the set of state variables – they represent the service state at runtime;
- The service URL for invoking it (SOAP);
- The event subscription URL.

The UPnP Control Point

- A control point in a UPnP network is a controller capable of discovering and controlling other devices. After discovery, a control point could:
  - retrieve the device description and get a list of associated services;
  - retrieve service descriptions of interest;
  - invoke actions to control the service;
  - subscribe to the service's event source; anytime the state of the service changes, the event server will send an event to the control point.

- It is expected that devices will incorporate control point functionality (and vice-versa) to enable true peer-to-peer networking.

C. Service discovery: 1. Initiated by the control point

- UPnP uses the Simple Service Discovery Protocol, SSDP;
- There are two interacting entities: the device and the control point (CP),

1. The CP sends a SSDP search request;
2. The device issues a unicast UDP notify answer with the URL to the device's XML description document;
3. The CP downloads the description document by HTTP – the web server running on the device returns the description document;
4. To receive automatic notifications of device changes, the CP subscribes to services it is interested in by HTTP;
5. The device acknowledges the subscription request and returns a unique Subscription Identifier (SID).

Service discovery: 2. A new device joins the network

- When a device joins a network, it advertises its devices and services to CPs – two messages for each embedded device, one message for each service.
- It does this by multicasting discovery messages (several times) to a standard address and port:

239.255.255.250:1900

- CPs listen to this address.
- Each message contains an *expiration time for the advertisement; if it remains available it should advertise again.*
- SSDP also provides a way for a device and associated service(s) to gracefully leave the network (bye-bye notification).

D. Using the service

- One CP can command the device to perform actions by changing one of the state variables:
  - the URL to send control requests is contained in the device's description document;
  - the CP issues a SOAP action over HTTP;

- The device changes the internal variable state and issues a SOAP response message;
- The result is expected in at most 30 sec; what happens after that depends on implementation.
- The device can notify clients of changes in its state because of explicit actions or implicit changes in the device itself – all subscribers are notified via a unicast NOTIFY message by HTTP.

Remote execution by SOAP - Simple Object Access Protocol

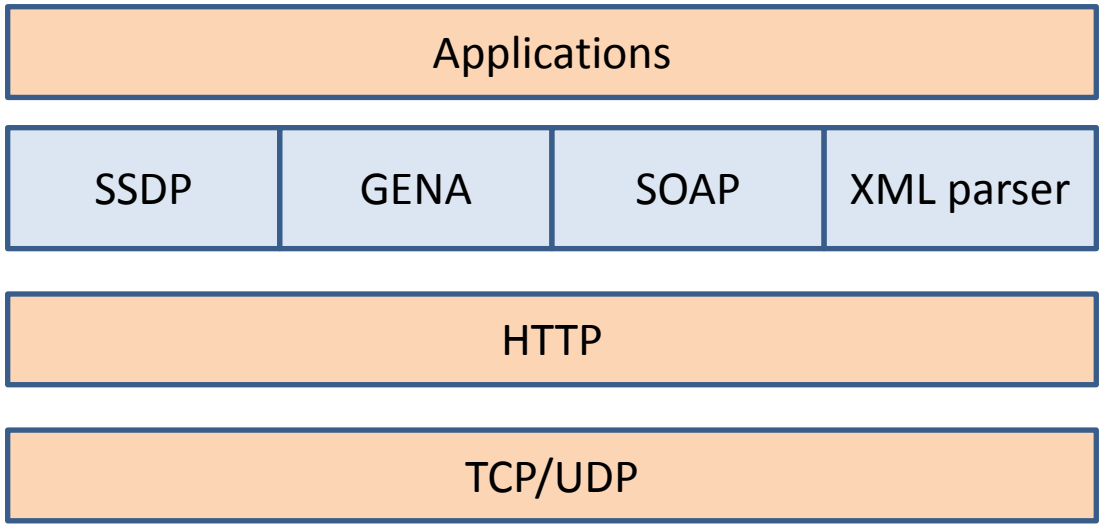
- SOAP defines the use of Extensible Mark-up Language (XML) and HTTP to execute remote procedure calls.
- Much like a remote procedure call, UPnP uses SOAP to deliver control messages to devices and return results or errors back to control points.
- Each UPnP control request is a SOAP message that contains the action to invoke along with a set of parameters. The response is a soap message as well and contains the status, return value and any return parameters.

Generic Event Notification Architecture (GENA)

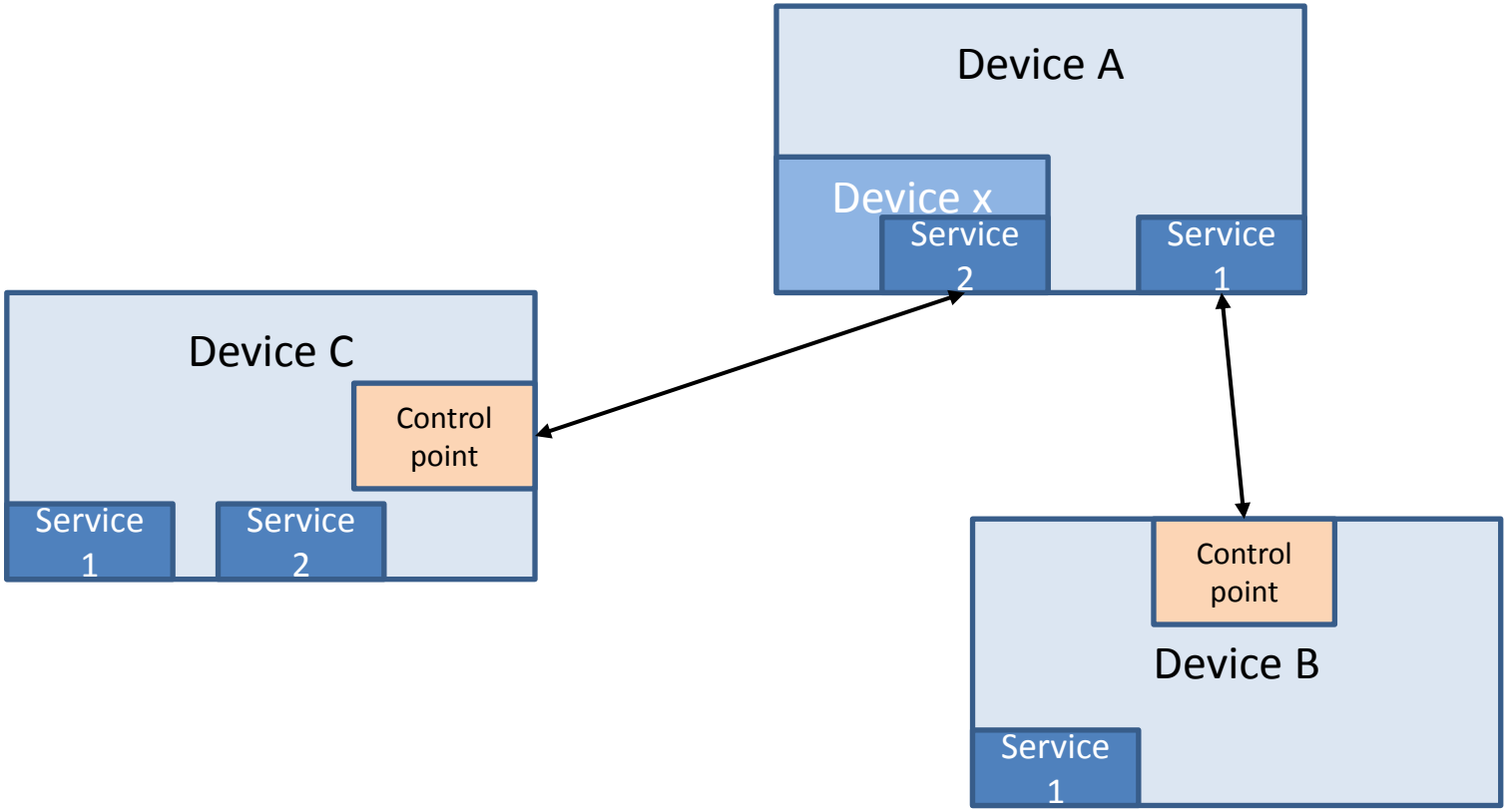
- A control point interested in receiving event notifications will subscribe to an event source by sending a request that includes:
  - the service of interest,
  - a location to send the events to and
  - a subscription time for the event notification.

- The subscription must be renewed periodically to continue to receive notifications, and can also be cancelled using GENA.

E. UPnP architecture



The UPnP model



Use case

- Steps needed to play an item from a Media Server on a Media Renderer :
  1. Select a Media server;
  2. Invoke Browse() to present content for selection for playback;
  3. Select a Media Renderer;
  4. Invoke GetProtocolInfo() on the Media Renderer;
  5. Match the ProtocolInfo from the content and the MediaRenderer;
  6. Invoke SetAVTransportURI() with the matched content;
  7. Invoke Play() to start the playback of the content.

Conclusions

- UPnP extends the model of PC plug and play of peripheral devices to networks and services.
- It is designed for “zero-configuration” ad-hoc networking; peers advertise and use services.
- Uses popular Internet protocols such as DHCP, SOAP, HTTP, TCP, IP, UDP, XML, XMPP.
- Too chatty, consuming a lot of network resources.
- Taken over by IoTivity.

Links

- http://upnp.org/resources/documents/UPnP_IoT_Overview_Dec2014.pdf
- http://openconnectivity.org/
- www.iotivity.org