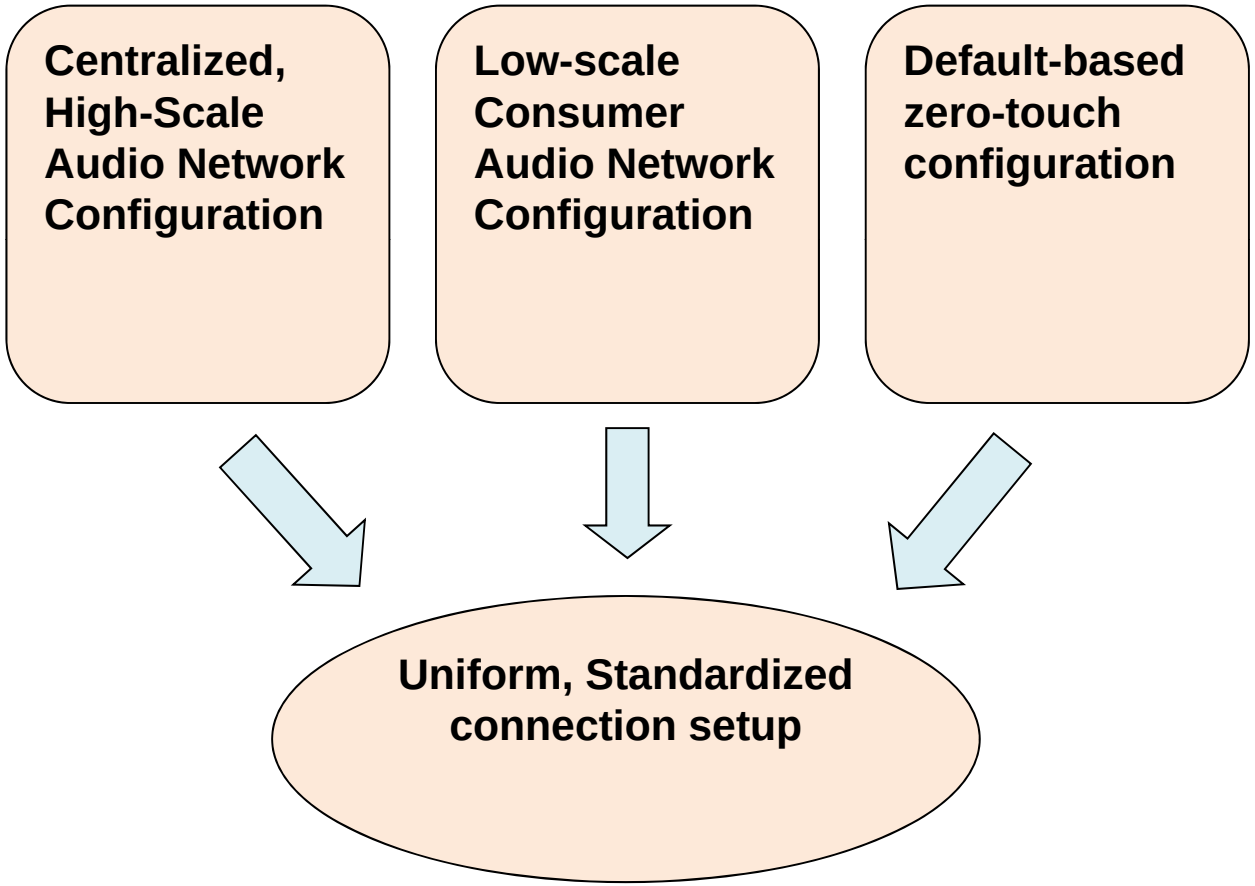
P1722.1 Connection Sequencing

Guy Fedorkow
Adamson Systems Engineering
guy (at) adamsonsystems.com
Feb 10, 2010

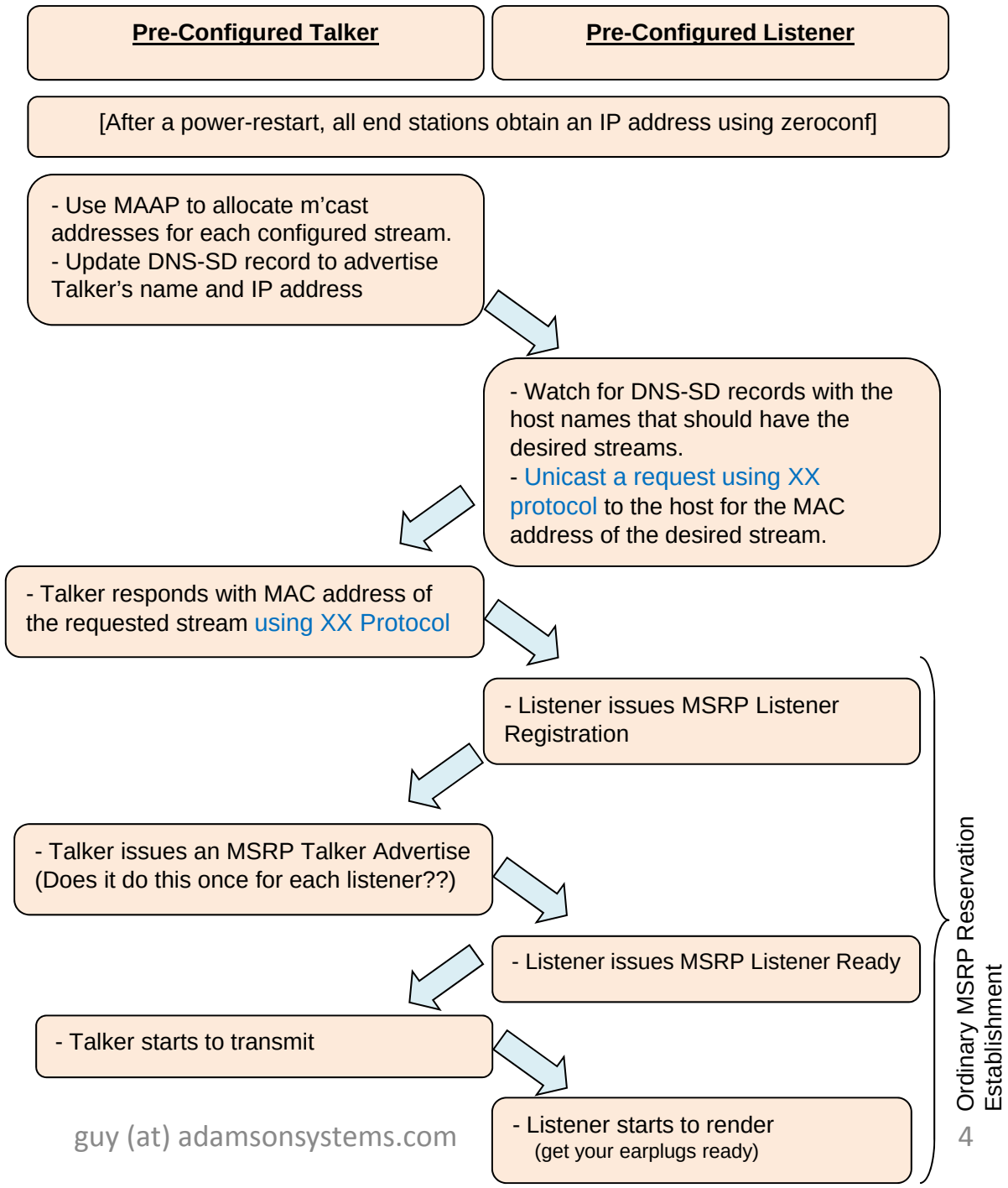
Agenda

- Connection Sequencing Recap
- Enumeration Protocol
  - Requirements
  - Candidates
  - Selection Criteria

Configuration vs Connection



Post-Configuration Restart



Endpoint Configuration

AVB Endpoint(s)	Connection Manager (System Controller?)
Use AutoIP or DHCP to obtain an IP address	
Use mDNS to register an endpoint device name	
	Collect a list of device names
Use DNS-SD to advertise basic AVB capability “I can do AVB” “I’m an Adamson Powered Speaker, Model YXXX”	
	Collect a list of all AVB-capable end points
	Issue unicast protocol XX commands to enumerate capabilities of each AVB-capable endpoint
Respond to unicast Protocol XX commands for enumeration information	
	Compile a table of channel counts, bit rates, coding, etc for all devices. Use proprietary techniques (rotary switches, LCD panels, cable-ID, etc) to match device names with physical devices. Read a list of desired connections from local storage / GUI / whatever.
	Issue proprietary commands to set gain, DSP params, delay compensation, etc using Protocol XX
Respond to proprietary configuration commands	
	Issue Protocol XX commands to configure names of desired connections to talkers and listeners
Respond to connection name configuration commands.	

Unicast Enumeration Protocol

AVB devices need a protocol that can be used for Enumeration:

- During the “connection” phase,
  - the listener needs to ask the talker for the MAC address / StreamID of the stream it wants.
- During the Configuration activity, a management device may:
  - ask Talkers for the list of names of streams they’re configured to transmit.
  - ask for a complete list of the parameters associated with a particular stream, i.e. bit rate, coding, etc.
- And during configuration, any authorized device may instruct:
  - a talker to talk a particular stream with particular parameters,
  - a listener to listen to a particular stream from a particular talker, also with the given parameters.
- The protocol could also report other “helpful stuff” as an assist to network management, (but that kind of reporting also could also be handled by a MIB.)

Protocol Candidates

- HTTP / Webforms of some sort
  - See http://microformats.org/wiki/introduction as a possible example
- AV/C encapsulated in IP (what's the proper name for the encapsulation?)
- SNMP (as described by Stephen Turner and John Grant)
- Protocol Buffers (Google's open-source extensible remote procedure call mechanism)
- Open Sound Control (see http://opensoundcontrol.org/introduction-osc)
- IETF CoRE (Constrained RESTful (Representational State Transfer) Environments, being developed for sensor networks)
- SOAP / XML (Simple Object Access Protocol)

Protocol Selection Criteria

Some criteria for selection:

- Small footprint in embedded devices
- Efficient enough for small processors
- Extensible as new connection parameters appear
- Clear ownership compatible with standardization
- Scalable enough for 1,000+ streams, hundreds of endpoints.
- Provision for security
- Interaction with routers and firewalls (e.g. should the protocol use, or avoid, Port 80?)
- Implementation Status


ADAMSON