Distributed Control: Echelon’s view of the Internet of Things

Bob Dolin’s position paper

Fred Baker’s presentation

Distributed control in the "Internet of Things"

- Types of "real time" in communications
  - Normal Internet communications
    - Moving files, exchanging text, in a static network
    - Human time scales, nobody dies if something is late
  - Space: Command/Telemetry
    - Constant feedback, occasional configuration changes
    - Remote object time scales, nobody dies if something is late
  - Distributed Control – machine to machine
    - Continuously changing variables to control the behavior of an application
    - In some applications, people die, or are annoyed, or the purpose is not served if time scales of interest are not observed

Example of distributed control: ANSI/CEA 709.1, EN-14908-1, GB/Z 20177.1

Bellagio Fountain

http://www.5min.com/Video/Learn-About-the-Bellagio-Fountains-278834664

- **Numerous control points**
  - Sometimes addressed as a group
  - Sometimes addressed individually
- **Paradigm:**
  - Central controller sets variables
  - Actions are published as status
  - Status changes published on a node affect the actions of the nodes that are subscribed to the status change

- **NACKs**
  - May be appropriate for scaling when moving large amounts of data

- **Positive Acknowledgement**
  - Need for commands
  - Retransmission within loss variance

Network stack requirements
(assumptions made by application) #1

- Retransmission:
  - Network recovers from packet loss or informs application
  - Recovery is timely: on the order of RTTs, not seconds
- Network engineering meets application requirements
- Network independent of MAC/PHY
- Scale
  - Thousands of nodes
  - Multiple link speeds
- Multicast
  - Throughout network
  - Reliable (positive Ack)
- Duplicate suppression

Network stack requirements

(assumptions made by application) #2

- Emergency messages
  - Routed and/or queued around other traffic
  - Other traffic slushed as delivered
- Routine traffic delivered in sequence
- Separate timers by peer/message
- Polling of nodes
  - Sequential
  - Independent of responses
- Paradigm supports peer-to-peer
  - Not everything is client/server

Network stack requirements

(assumptions made by application) #3

- Capabilities:
  - Discover nodes
  - Discover node capabilities
  - Deliver multi-segment records (files)
- Exchange of multi-segment records
- Network and application versioning
- Simple Publish/Subscribe parsers
- Security
  - Strong encryption,
  - Mutual authentication,
  - Protection against record/playback attacks
  - Suite B ciphers