IoT Operational Issues

Karsten Walther — Perinet
(Carsten Bormann — Uni Bremen TZI)

2024-03-20
— Simple networkable things: light, sensor, power-switch
  — in our case Single Pair Ethernet is used
  — no user interface, except a web page and a label
— devices, virtual devices and local services (on containers)
  — act autonomously together during normal operation
  — users occasionally use a browser for interaction
— Typical Applications:
  Operations level in industry, building and agriculture
  — often no internet connection nor name server or DHCP available
Distance

Time: --

- distance: 220 mm
- max: 300 mm
- min: 40 mm
— Typical User: Technician without IT knowledge
  — previously configured a PLC
  — has no administration rights on their computer
  — minimizes contact to network administrators

→ must support ad-hoc connection by technician
→ Zeroconf is a base requirement
Edge Computer with Lightweight VMs

Networked Sensors and Actuators (mdns-sd, IPv6LL only, https server, mqtt client)

User

MQTT broker
PKI2go
Application logic

Switch/AP

Ethernet

S

WiFi

S

A

S
Example Hardware For Web Server and MQTT Client

- Cortex R4 250MHz
- ~ 200KB RAM
  (for code and data)
Problem 1: misleading mDNS name resolution

- mDNS resolves to multiple addresses GA, ULA and LL
- often the requester cannot use all of them and has to select
- intransparent to the user because hidden in the SW stack
- no control possibility by the user (browser address field)
- non deterministic
- the problem is unnecessary, since requester defines a domain
  - .local
  - .my-intranet
  - .tld
- → just reply addresses matching to the domain
Problem 2: `.local` request sent to nameservers

— local nameserves may silently ignore `.local` requests
  — → timeout
    → requested web server inaccessible by browser
  — strange workarounds in the field (e.g., fritz.box)
  — how should an IoT device know what is a local scope?
— some name servers reply a special address for unknown host names
  — the actual device never will use this and is therefore inaccessible
Problem 3: IPv6 LL zone ID

- makes URIs unusable: information in the URL is locally valid only
  - Setup: Client (Browser), Management HTTP Service (on Edge Device), and IoT Web service (on the networked sensor) are connected in L2
  - Management Web page cannot provide an IPv6 LL address as link to the networked sensor
  - same applies in various other situations
- no support by many popular libraries (e.g., nodejs)
  → "non-working-experience" for users
- IMHO zone ID is unnecessary:
  it creates more problems now than it solves later
Problem 4: Support for offline environments

— Example Situation: Sensors on a plow attached to different pulling machines (or not at all during servicing)

— Webpage cannot be accessed by the user due to various reasons
  — some browsers deactivate IPv6 completely (also Link Local), when there is no Internet connectivity
  — Windows deactivates MDNS for unknown networks by default
  — user cannot type IPv6 addresses with zone ID in the address bar
  — browsers don't support local web server lookup via mdns, as printer dialogue does, and local device may be muddy or below covers, thus the user will have no address information
Problem 5: Short certificate lifetime

— Web PKI is moving to ever smaller certificate lifetimes
— devices are not online in many cases and cannot be updated automatically with certificates
  — during shelf storage
  — attached to non-powered machinery, different networks or no internet connectivity for months
— required fall back after factory reset to initial certificate
Problem 6: No standard simple role based access model

— Authorization is granted via a client certificate
— User levels are typically simple
  — normal authenticated users can read values or control actuators
  — privileged users: e.g., setting calibration values
  — application admins (technicians) can do everything: updates, connection settings, security
— so far we use an extension field in certificates in a proprietary way
  — works very well, but we would like to have a standard
  — support by standard PKI tools required in the mid term
Problem 7: Browsers disrespect Web server constraints

— our device tells the client the maximum packet size and number of connections
  — $\leq 6$ connections
  — memory restrictions $\rightarrow$ no Jumbo Frames
— is ignored by some browsers or web libraries
  — especially in browsers it looks like a stalled device, with MQTT, MDNS still working
Problem 8: Virtualization environments act as routers

— IoT heavily relies on ZeroConf mechanisms like MDNS
— e.g. a sensor has to find the responsible MQTT broker, which runs in a container
— user need to access user interface in browser
— Virtualization environments (docker, snap...) act as routers
— IPv6 LL and mdns cannot be used
— breaks fundamentals of local IoT
What next?

— Where can standardization help?
— Help me to identify the right working groups
— or link me to relevant groups outside of IETF
— Are there simpler ways than updating/making standards?