Towards a device-infrastructure continuum in IoT and OT networks

HKUST Internet Research Workshop 2024, 2024-03-15

Carsten Bormann

Carsten Bormann

Universität Bremen TZI

IRTF T2T RG

cabo@tzi.org

- **IETF**
  - "Make the Internet work better"
  - **High-quality** standards that run the Internet
  - "Engineering"
  - **Open**
- **IRTF**
  - Research arm
  - No "standards"
  - Foster relevant research

Internet of Things (IoT): Thingness

- Thing:
  A **physical** item that is also made available in the **IoT**.
  - notable for their **interaction** with the physical world beyond
    - interaction with humans, and
    - its own physical internals.
- a temperature sensor or a light might be a Thing,
- but a router might exhibit less Thingness, even if it employs both temperature sensors and indicator lights, as the effects of its functioning are mostly on the digital side.

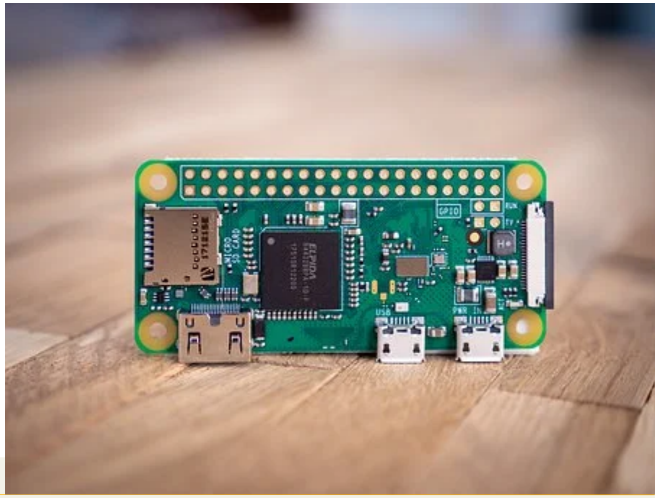
→ **Thingness**: the Thing is interesting because of its interactions with the physical world.

Constraints of IoT nodes

- Need to work with little power (energy)
  - RFC 7228 “constrained node”
  - Microcontroller, not full computer
- Need to be inexpensive in TCO

scaling

- Need to be in strange places ➔ physical distribution
  - ➔ mostly remote management
- Need to work with little **attention**
- Have limited user interfaces



- Need to run for decades
- Need to run continuously
- ➔ Are hard to bug-fix and upgrade

Pets vs. Cattle

Pet

- Treat as a unique item
  - Individual attention during network design and operation
- Individual configuration

Cattle

- Treat as a herd
  - Individual attention during installation only
- Individual identity, common configuration

Security

⛈⚡ **Disclosure**:

- … of personal information → **privacy** violations
- … of industrial information → **espionage**, **reconnaissance**

⛈⚡ **Falsification:**

- → Cannot rely on data (possibly regulatory consequences)

⛈⚡ **Malicious take-over:**

- Node no longer reliable
- Vehicle for DDoS **attacks** on others/other things

→ Modern architectures (e.g., zero-trust):

- Protection of Data is end-node responsibility
- Protection of Meta-Data is **hard**
- Protection of nodes needs software updates, **attestation**

Job of the IoT **network**

- Help in initialization/setup (e.g., assign IP address)
- Connectivity to **desirable** partners
  - Service parameters: Latency, bandwidth
  - **Protect** IoT node from irrelevant events
    (e.g., routing changes, hardware failover)
- **Protect** node from undesirable access (battery depletion)
- **Protect** others from malfunction/attack

- Focus for today's talk:
  - **Cattle**
  - **no personal data (PII) → industrial, building control, …**

What information can we build on?

- Instance information
  - Operational data (e.g., IP address)
  - Purpose in Life
    - e.g., installed where/for what
    - Instance-level communication partners, communication parameters (e.g., MQTT broker, topic)
  - Individual software state; attestation, …
  - Class (see below)
- Class information
  - Physical interfaces/capabilities
  - IoT Affordances (interaction patterns)
  - Class-level communication partners (e.g., update server)

What does the network see from this?

- First hop:
  - MAC address (now often randomized)
  - Potentially: Association info (802.1X etc.) ➔ ~ node identity
- Following hops:
  - Source address, destination address, protocol (TCP/UDP)
  - More information by peeking into the packet (ports, etc.)
  - Potentially: path-level negotiation (RSVP/integrated services)
    - Based on 5-tuple (SA, DA, protocol, SP, DP)
  - Intentional traffic classification by sender
    - DSCP (was: type of service ToS): differentiated services only 6 bits, bleached on domain boundaries
    - VLAN ID
    - **Semantic Addressing**

„ Do you know what’s on your **network**?

Surprisingly, this is often not well-defined, even in OT environments

The explosion of variety

- Over time, the number of **classes** of IoT Things increases
  - New kinds of devices
  - New suppliers, new product lines
  - New software versions
  - New usages
- Even factories are now multi-stakeholder environments
  - Compare airplanes, where the engines are islands of control
  - **Who** is responsible for a node?
- Desirable communication changes with new classes, instances
  - Which devices and IT nodes are the peers?
  - What are the performance needs?

Device **Classes** are Cattle

- We no longer can manually react to each new Thing species
- Information about device classes needs to be **machine-readable**

- Devices need to offer **self-descriptions,** provided by
  - Manufacturers (ODMs, OEMs, … through supply chain)
  - Integrators
  - Application operators

- Important for Thing software security:
  - **SBOM** (Software “Bill of Materials”; supply chain), SWID/**CoSWID**, **CoRIM** (Reference Integrity Measurements)
  - Manifests for Secure Software Update (IETF **SUIT**)

Keeping control

- Devices can be
  - Misconfigured, reacting badly to environmental changes
  - Attacked and compromised
    - Possibly after detection of vulnerability (zero-day)
- Is the device still healthy, i.e., behaving as **desired**?
  - **Things** have small number of purposes, are simple
  - Generally can define behavior tightly
  - If behavior leaves that envelope → problem?!
- IoT device **manufacturer** may know some of this
  - Class information
- “Purpose in life" information also needed
  - Class, instance information

RFC 8520: Manufacturer's Usage Description

- **Trustworthy** class information about intended behavior
- **Actionable**, can be translated into network control

- Manufacturer provides **MUD file** (simple JSON format)
- Makes it available under trustworthy URL (https://)
- Device declares its **MUD URL** (LLDP, DHCP, …)
- **MUD controller** picks up MUD information
  - Authorization — is a device like this even acceptable?
  - Derives network control information
  - Relays it to policy decision points, enforcers

MUD limitations

- **MUD =** Manufacturer's Usage Description
  - Class-level only

- MUD Information cannot be adapted to purpose in life
  - Actually desired peers are defined by application, not available at time of manufacture
- "Intended behavior" limited to ACL
  - No dynamic information — DNS indirection only
  - No quantitative information, no AI/ML

- Manufacturers have a hard time generating MUD files
  - Limited expressibility, too easy to open all barn doors
  - Incentive still low

MUD for legacy devices

- Manufacturer may no longer be around
  - Or not interested/qualified to provide MUD info

- Approach: Observe instances (classical, AI/ML)
- Generate non-manufacturer MUD files from observed behavior
- Can check behavior for covert call-home or other infractions

- Users of device class can collaborate via open-sourcing:
  ➔ Curated repositories of reverse-engineered MUD files

Beyond MUD

- Benefit from other types of self-description
  - W3C Web of Things “Thing description/model” TD/TM
  - IETF “Semantic Definition Format” SDF
  - OpenAPI API definitions
  - …
- Integrate with “Purpose in Life” information
  - Generated from network planning and design, applications
  - E.g., based on “Internet Ontology”
  - Needs to enable merging/inference with self-description and operational information
- Obtain attestations about device health
  - More self-description information from Attestation Verifiers

SDF: Semantic Definition Format

- Describe nodes beyond their network behavior
- Based on json.schema.org-like data model
- Interactions: Property, Action, Event

- Designed as a hub for **ecosystem models**
- Converters from/to OMA, OCF, … models exists
- Ecosystem specific mappings into protocols complement SDF

- draft-ietf-asdf-sdf-18 agreed by WG, IESG step next

- Compare W3C Thing Description

” KNOW MORE

What is on the network; what is the desirable traffic, not all traffic is the same

Where we want to be:

Well-Informed Networking (WIN)

” Ask not
what your network
can do for you –
ask
what you can do
for your network!

Inspired by John F. Kennedy’s famous inaugural address, 1961-01-20