Constrained Node Networks

2014-03-05

Prof. Dr.-Ing. Carsten Bormann

TZI – Universität Bremen
Connecting:
Places → People → Things
Scale up:

Number of nodes

(50 billion by 2020)
Scale down:

node
Scale down:

- cost
- complexity
cent
kilobyte
megahertz
Constrained nodes: orders of magnitude

10/100 vs. 50/250

- There is not just a single class of “constrained node”

- Class 0: too small to securely run on the Internet
  - “too constrained”

- Class 1: ~10 KiB data, ~100 KiB code
  - “quite constrained”, “10/100”

- Class 2: ~50 KiB data, ~250 KiB code
  - “not so constrained”, “50/250”

- These classes are not clear-cut, but may structure the discussion and help avoid talking at cross-purposes
in constrained node/networks, Moore’s law barely applies

• In the low-power, low-cost area, gains from Moore’s law are used
  • to save power
  • to save cost
• Performance, ROM, RAM grow very slowly
Constrained networks

- **Node**: ... must sleep a lot (\(\mu W!\))
  - vs. “always on”

- **Network**: \(\sim 100\) kbit/s, high loss, high link variability
- May be used in an unstable radio environment
- Physical layer packet size may be limited (\(\sim 100\) bytes)
- “LLN low power, lossy network”
please re-calibrate your complexity meters

- **code** is expensive
- "class 1" = 100 KiB, "class 2" = 250 KiB

- **state** is expensive
- "class 1" = 10 KiB, "class 2" = 50 KiB

- **packets** are expensive

- **listening** is even more expensive
  - and multicast doesn’t work
Energy consumption on TelosB

Message exchange cost orders of magnitude more than symmetric crypto

Constrained Node Networks

Internet of Things (IoT)
Wireless Embedded Internet (WEI)
Low-Power/Lossy Networks (LLN)
IP Smart Objects (IPSO)
## Constrained Node Network Cluster

<table>
<thead>
<tr>
<th>INT</th>
<th>LWIG</th>
<th>Guidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT</td>
<td>6Lo</td>
<td>IP-over-foo</td>
</tr>
<tr>
<td>INT</td>
<td>6TiSCH</td>
<td>IP over TSCH</td>
</tr>
<tr>
<td>RTG</td>
<td>ROLL</td>
<td>Routing (RPL)</td>
</tr>
<tr>
<td>APP</td>
<td>CoRE</td>
<td>REST (CoAP)</td>
</tr>
<tr>
<td>SEC</td>
<td>DICE</td>
<td>Improving DTLS</td>
</tr>
<tr>
<td>OPS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(2) The Application
CoAP
Constrained Node/Networks ➔ Compressed HTTP?

- Saves some bytes
- Retains all the complexity
  - lots of historical baggage
  - still needs TCP below
- Adds the CPU requirements for compression
- Limited gain
  - compression only takes you so far
"Make things as simple as possible, but not simpler."

Attributed to Albert Einstein
The Constrained Application Protocol (CoAP)

- Implements HTTP’s REST model
  - GET, PUT, DELETE, POST; media type model
- While avoiding most of the complexities of HTTP

- Simple protocol, datagram only (UDP, DTLS)
- 4-byte header, compact yet simple options encoding

- Adds “observe”, a lean notification architecture
CoAP Examples

- **GET** coap://temp1.25b006.floor1.example.com/temperature
  - ASCII string: 22.5
  - could use JSON, e.g. as in draft-jennings-senml
- **PUT** coap://blue-lights.bu036.floor1.example.com/intensity
  - ASCII string: 70 %
- **GET** coap://25b006.floor1.example.com/.well-known/core
  - </temp>;n="TemperatureC",</light>;ct=41;n="LightLux"
  - see RFC 6690 (CoRE link format)

More in draft-vanderstok-core-bc-05
see also draft-ietf-core-interfaces
Example Interchange

C: CON + GET coap://server/resource

S: ACK, ct=application/cbor, payload: {"hlo":"World"}
Combining CoAP and HTTP

- CoAP is used in constrained environment
- CoAP and HTTP share proxy model based on REST
- Enables standard, application-independent proxy
Security is not optional!

- HTTP can use TLS (“SSL”)
- CoAP: Use **DTLS 1.2**
  - Add 6LoWPAN-GHC for efficiency
- Crypto: Move to **ECC**
  - P-256 curve
  - SHA-256
  - AES-128
- To do:
  - Commissioning models (Mother/Duckling, Mothership, …)
  - **Authorization format and workflow**
  - Performance fixes (DICE)

**128-bit security**

(≈ RSA 3072-bit)
Processes for **usably secure** lifecycle (changes of ownership, authorization, privacy, ...)

---

The lifecycle of a thing in the Internet of Things

[draft-garcia-core-security]