

Always Online Requirements for Sleeping CoAP Nodes

draft-cao-core-aol-req-00.txt

Zhen Cao

China Mobile

July 2011, Quebec City

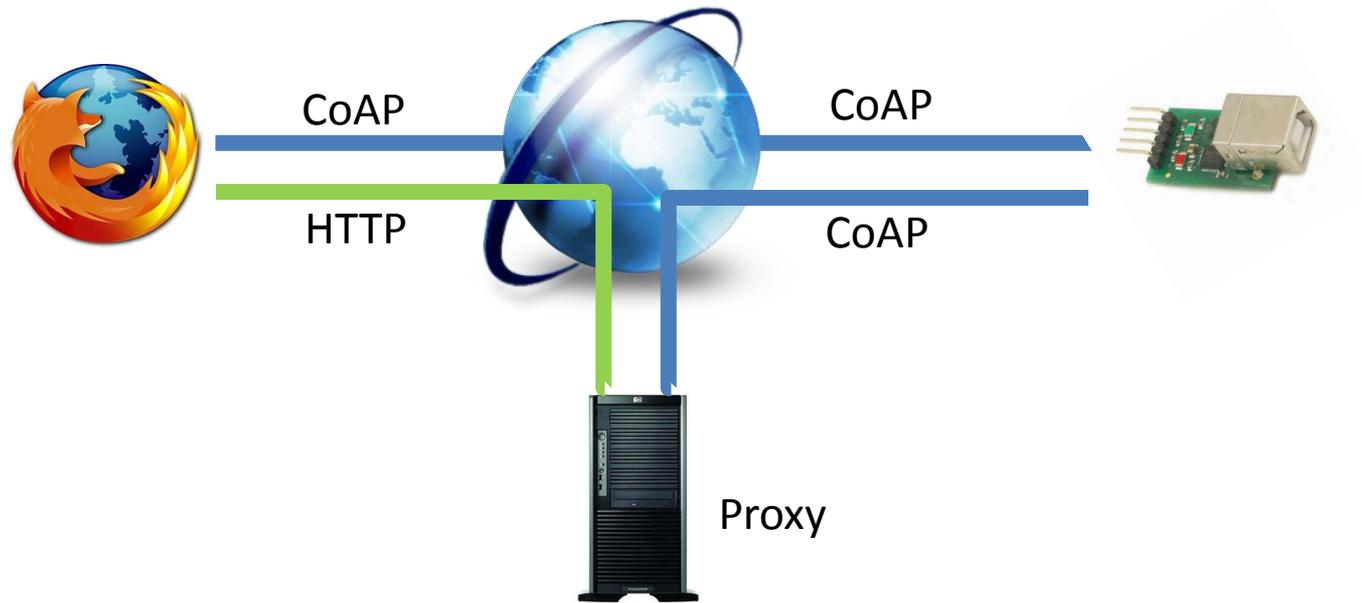
What's the Web alike



Web servers and data center

- In the Web environment built from HTTP, the web servers are
 - Highly reachable
 - Highly reliable
 - With careful attendance

Web of Things enabled by CoAP



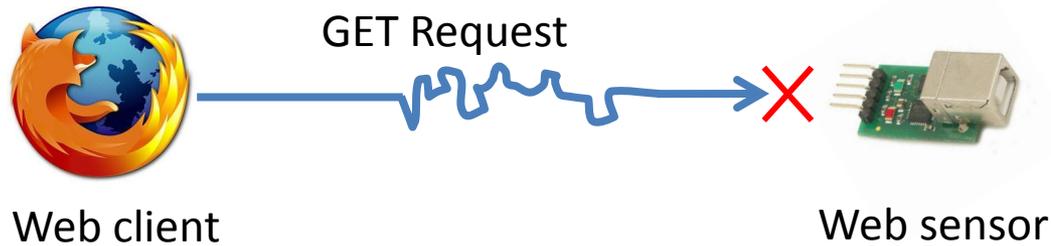
- RESTful
- Easy to proxy to/from HTTP
- Cachable
- Lightweight

What happened if sensor join the web?



- Sensors are “Drowsy”, not always reachable
 - Energy conservative
 - Unattended
 - Computing and storage constrained
 - Multihop constrained

Cases of Sleeping CoAP Nodes



Constraintful Sleepy

- Duty-circled MAC
- Higher layer energy saving mechanism

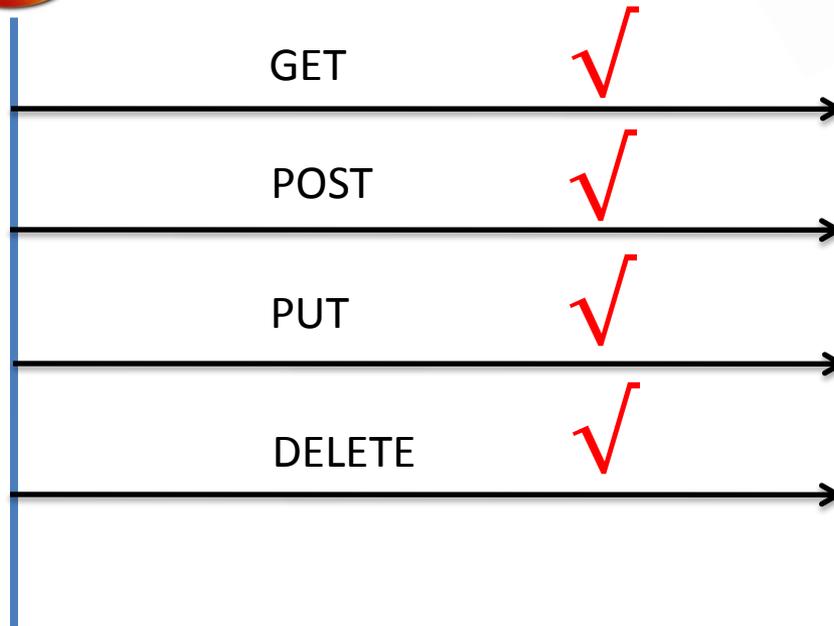
Interface-wise Sleepy

- For example, cellular interface
- IP addresses are occasionally released if there are not traffics for a while

Network-wise Sleepy

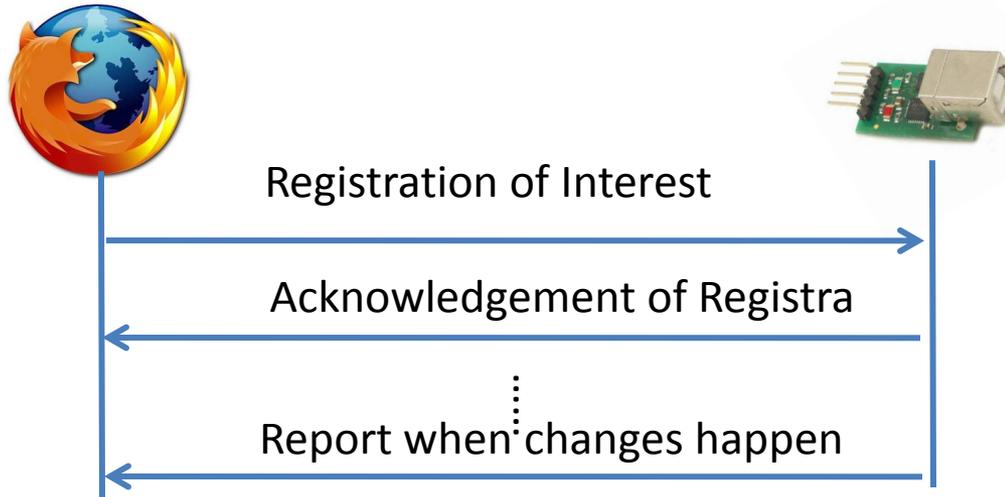
- In IPv4, sensors placed behind the NAT, UDP
- IP address changes, DNS not updated

Always online Requiriements



- Key: keep the connection state

Ways to Circumvent - Observe



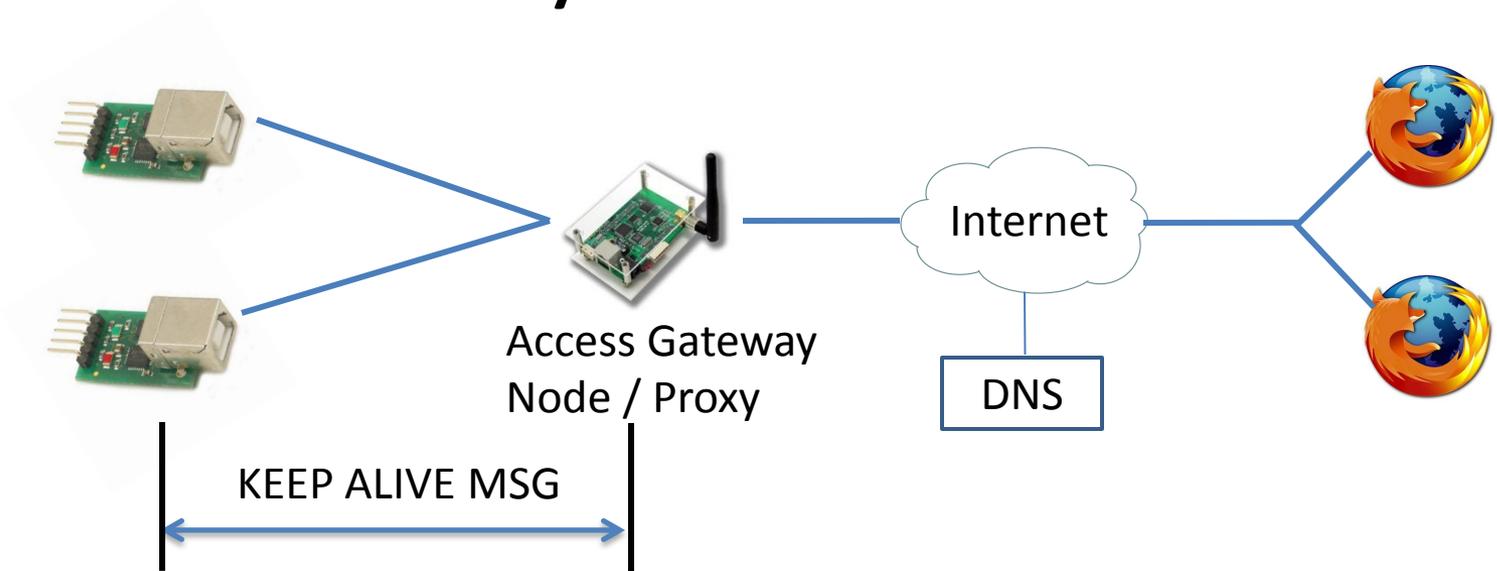
- Prons:
 - On-demand
 - Easy to support frequently changed state
- Cons:
 - Sensor should keep listening for the Reg request
 - Fail if non-eventful objects, e.g., POST request

Ways to Circumvent – Client-only Mode



- Prons:
 - Easy to implement and avoid the cost of serving content reactively
- Cons:
 - Repeated sending is energy consuming, use Well known URL?

Ways to Circumvent – Always-online Mode



- **Prons**
 - Application layer keep alive messages can keep the IP and lower layer connection states, so that web client can always reach them
- **Cons**
 - Energy needed, cost to be evaluated!

Purpose of this draft and presentation

- Do NOT intent to evaluate these different approaches at this point
- DO want to arouse interest and attention to this problem!