Enhancement of CoAP Packet Delivery Performance for Internet of Things

Hang Liu

Outline

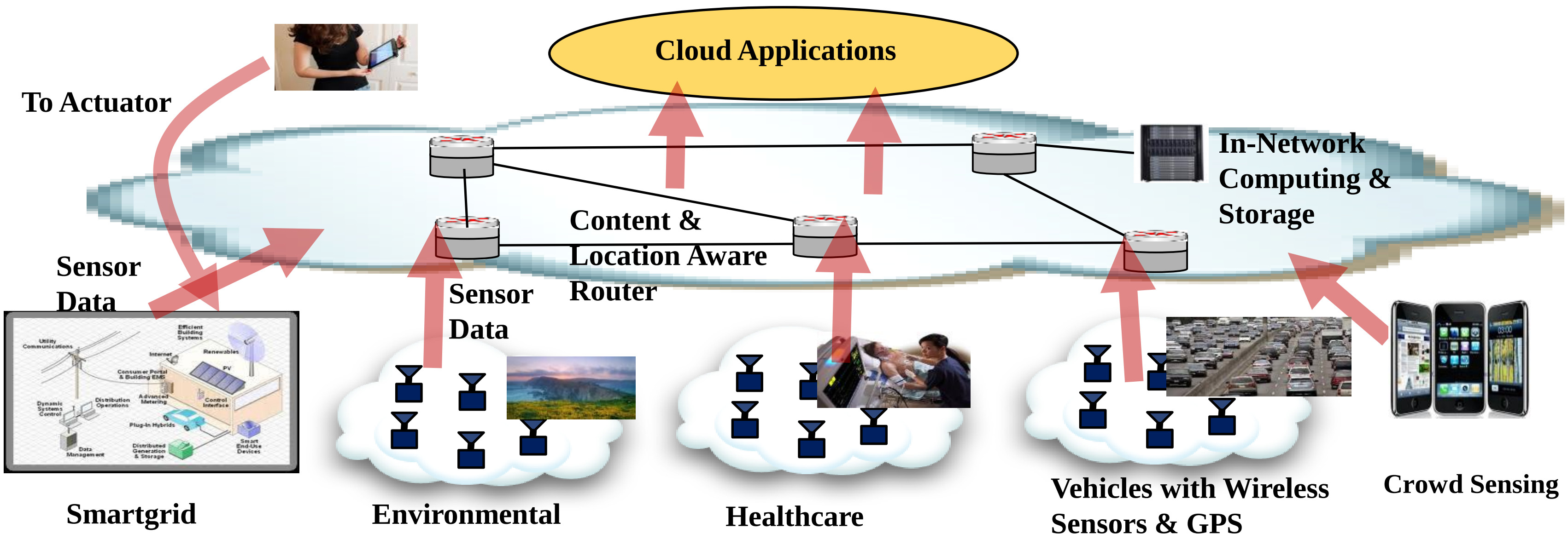
- Motivation and Industrial Relevance
- Project Objectives
- Approach and Previous Results
- Future Work
- Outcome and Impact

Motivation & Relevance

- Internet of Things (IoT) may be the next revolutionary technology in transforming the Internet,
  - Sensors and actuators blend seamlessly around us, interfacing human beings with the physical world.
  - IoT enables a wide range of applications including mobile health, industrial control, smart utilities, smart transportation, smart city, etc.,
- Industry forecast:
  - Currently there are 9 billion interconnected devices and it is expected to reach 24 billion devices by 2020.
  - Potentially generates $1.3 trillion revenue opportunities for mobile network operators alone.

Architecture Consideration

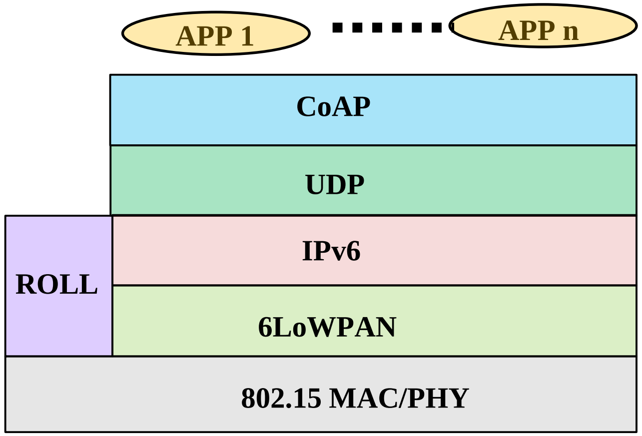
- We consider a cloud-based, content-centric IoT architecture, design and experiment with more robust and efficient protocols and algorithms.
  - Cloud computing with powerful data analysis and representation tools, on-demand storage, ubiquitous access, flexible information queries, and high reliability and scalability
  - Information is shared across platforms as services



[4]

Project Objectives

- Make standard IoT protocols more robust, scalable, power-efficient, and bandwidth-efficient
  - Standard IP-compliant network stack to support diverse IoT applications is critical for the future success of IoT
  - Proprietary systems impede wide adoption and interoperability of IoT
  - IEEE, IETF, ETSI, … defining IoT standards
  - Industry starts converging to a standard protocol stack
  - Enhancement and optimization needed given IoT network restrictions and diverse application scenarios



- Enhancement of standard IoT protocols
  - IETF Constrained Application Protocol (CoAP)
  - Routing over Low Power and Lossy Networks protocol (ROLL)

CoAP

- HTTP: high overhead, use TCP connection, not optimized for sensors.

- CoAP:
  - A generic application-layer protocol for constrained environments with a set of RESTful specifications,
  - Support publish/subscribe model.
  - Designed for low overhead, heterogeneous and low-power devices
  - Not a simple compression of HTTP, interoperable with HTTP
  - Run on the top of UDP with application layer unicast and best-effort support and asynchronous message exchanges.

[6]

CoAP

- Issues of CoAP:
  - Very simple stop-and-wait retransmission mechanism, involving packet delay/loss.
  - Current implementation utilized a single timer for all the retransmissions, either a fixed timer value or exponential backoff.
  - Requirement for better efficiency and reliability.
  - Can be further enhanced to support more features for new applications

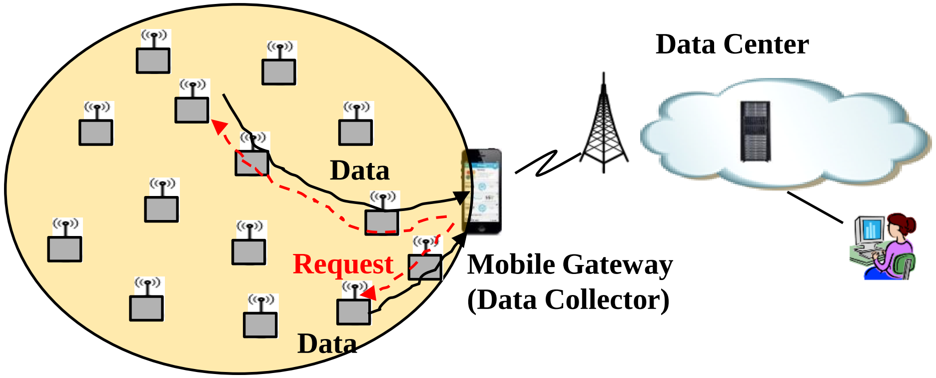
Approach

❖ Reliable message transport and *multipath* congestion control scheme for CoAP

– Based on the asymmetric characteristic of the system.

◆ More complex multi-destination Request rate control and scheduling at gateway with Request retransmissions

◆ Simple response scheme at sensors



[8]

Congestion Control

- Design and evaluate alternative congestion control algorithms for CoAP to understand the tradeoffs between performance and complexity/overhead.
  - Multi-timer, single window:
    - Separate RTT estimations and retransmission timer for different sensor sources at the data gateway.
    - Single window or queue for congestion control.
  - Multi-timer, multi-window

Request Transmission

- Retransmission of a Request once timeout
- Request window W changes:
  - Increased by $\frac{\mu}{W}$ at Data packet reception
  - Decreased by γW with a request timeout
- Pacing the transmission of "Request" (not like TCP)
  - Requests are transmitted at a rate W/RTT

RTO Estimation

- Compare several Request retransmission timeout (RTO) estimation algorithms to understand their performance and complexity in IoT networks.
  - Algorithm 1

$$RTT_i = (1-\alpha)RTT_i + \alpha RTT_i(k)$$

$$\tau_i = \delta \times RTT_i$$

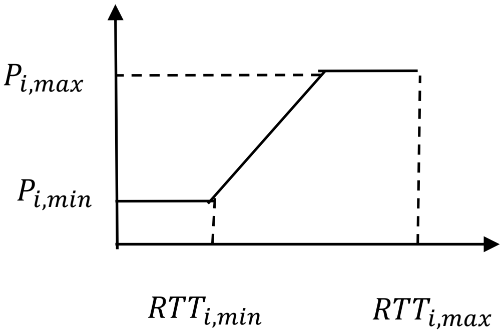
  - Algorithm 2

$$Dev_i = (1-\beta)Dev_i + \beta|RTT_i - RTT_i(k)|$$

$$\tau_i = RTT_i + \delta \times Dev_i$$

  - Algorithm 3: the probability of window change is

$$P_i(k) = P_{i,min} + (P_{i,max} - P_{i,min}) \frac{RTT_i(k) - RTT_{i,min}}{RTT_{i,max} - RTT_{i,min}}$$



- Ongoing work: OMNet
- Future work: prototyping and testbed evaluation

Outcome and Impact

- Protocols and mechanisms to enhance CoAP protocols, congestion control, request rate control and scheduling.
  - Improve CoAP reliability and efficiency
  - Standardization opportunities
- Simulation software
- Prototyping software developed
- Evaluation results

Thanks

Q & A