Low-Power Interoperability for the IPv6 Internet of Things

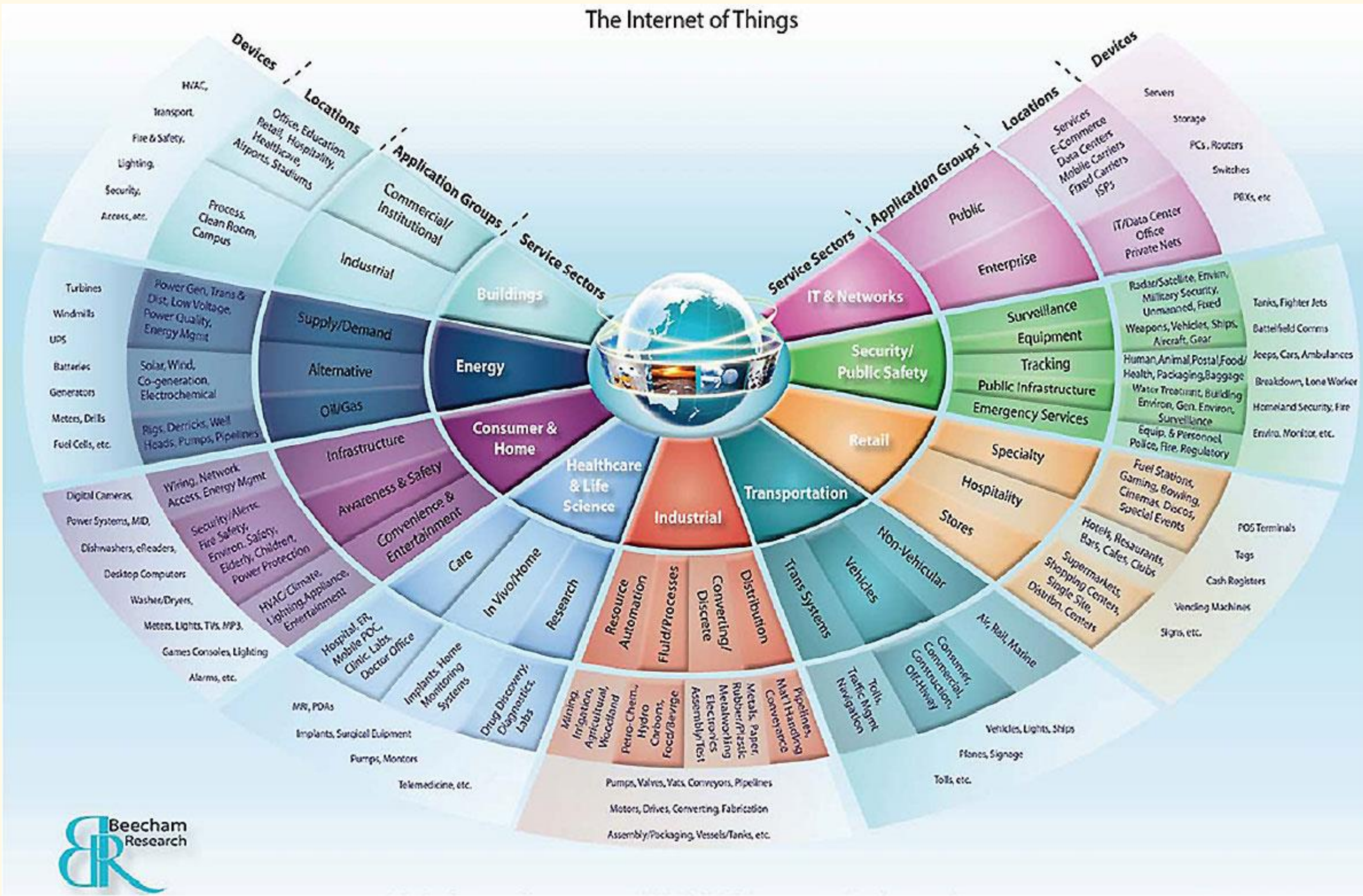
Adam Dunkels, Joakim Eriksson, Nicolas Tsiftes
Swedish Institute of Computer Science

Presenter - Bob Kinicki

Introduction

- The Internet of Things is a current 'buzz' term that many see as the direction of the "Next Internet".
- This includes activities such as Smart Grid and Environmental monitoring.
- This is a world of ubiquitous sensor networks that emphasizes energy conservation!
- This paper provides an overview of the low-power IPv6 stack.

Internet of Things (IoT)



Steps for IoT Interoperability

1. Interoperability at the IPv6 layer
   - Contiki OS provides IPv6 Ready stack.
2. Interoperability at the routing layer
   - Interoperability between RPL implementations in Contiki and TinyOS have been demonstrated.
3. low-power interoperability
   - Radios must be efficiently duty cycled.
   - Not yet done!!

Low-Power IPv6 Stack

Layer	*Example protocol*
Application	HTTP, CoAP
Transport	TCP, UDP
Network	IPv6, RPL, 6lowpan
MAC	CSMA
Radio duty cycling	X–MAC/ContikiMAC
Link	IEEE 802.15.4

focus of this paper

Figure 1. The low-power IPv6 stack consists of the standard IPv6 protocols at the network layer and transport layers, and of new protocols from the network layer and down.

CoAP versus HTTP

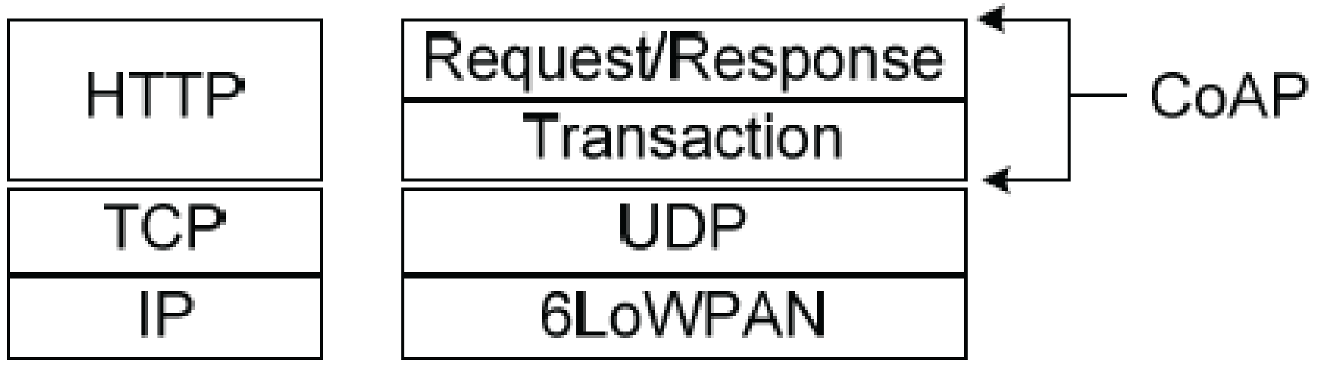


Figure 1. HTTP and CoAP protocol stacks

Colitti et al.

CoAP Background [Colitti]

- IETF Constrained RESTful environments (CoRE) Working Group has standardized the web service paradigm into networks of smart objects.
- In the Web of Things (WOT), object applications are built on top of the REpresentationl State Transfer (REST) architecture where resources (objects) are abstractions identified by URIs.
- The CORE group has defined a REST-based web transfer protocol called Constrained Application Protocol (CoAP).

CoAP

- Web resources are manipulated in CoAP using the same methods as HTTP: GET, PUT, POST and DELETE.
- CoAP is a subset of HTTP functionality re-designed for low power embedded devices such as sensors.
- CoAP's two layers
  - Request/Response Layer
  - Transaction Layer

CoAP

- Request/Response layer :: responsible for transmission of requests and responses. This is where REST-based communication occurs.
  - REST request is piggybacked on Confirmable or Non-confirmable message.
  - REST response is piggybacked on the related Acknowledgement message.

CoAP

- Transaction layer handles single message exchange between end points.
- Four message types:
  - Confirmable – require an ACK
  - Non-confirmable – no ACK needed
  - Acknowledgement – ACKs a Confirmable
  - Reset - indicates a Confirmable message has been received but context is missing for processing.

CoAP

- CoAP provides reliability **without** using TCP as transport protocol.
- CoAP enables asynchronous communication.
  - e.g, when CoAP server receives a request which it cannot handle immediately, it first ACKs the reception of the message and sends back the response in an off-line fashion.
- The transaction layer also supports multicast and congestion control.

COAP Efficiencies

- CoAP design goals:: small message overhead and limited fragmentation.
- CoAP uses compact 4-byte binary header with compact binary options.
- Typical request with all encapsulation has a 10-20 byte header.
- CoAP implements an observation relationship whereby an "observer" client registers itself using a modified GET to the server.
- When resource (object) changes state, server notifies the observer.

Accessing Sensor from Web Browser

Table 1. Comparison between CoAP and HTTP

	Bytes per-transaction	Power	Lifetime
CoAP	154	0.744 mW	151 days
HTTP	1451	1.333 mW	84 days

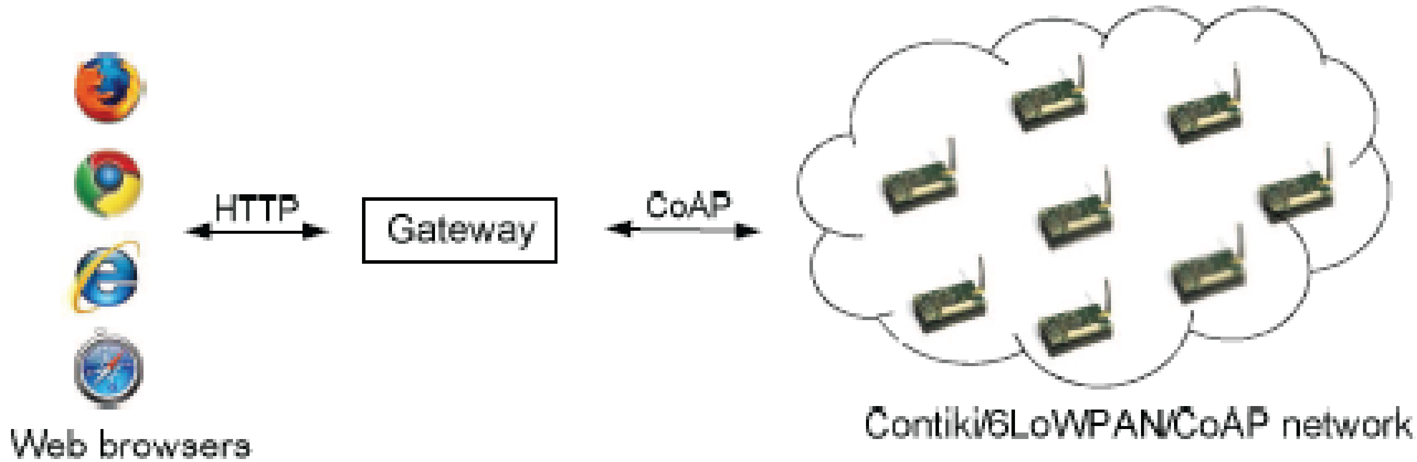


Figure 2. Integration between WSNs and the Web

Colitti et al.

IPv6 for Low-Power Wireless

- IPv6 stack for low-power wireless follows IP architecture but with new protocols from the network layer and below.
- 6LowPAN adaptation layer provides header compression mechanism based on IEEE 802.15.4 standard to reduce energy use for IPv6 headers.
  - Also provides link-layer fragmentation and reassembly mechanism for 127-byte maximum 802.15.4 frame size.

IPv6 for Low-Power Wireless

- IETF ROLL (Routing over Low-power and Lossy networks) group designed RPL (Routing Protocol for Low-power and Lossy networks) for routing in multi-hop sensor networks.
- RPL optimized for many-to-one traffic pattern while supporting any-to-any routing.
- Supporting different routing metrics, RPL builds a directed acyclic graph from the root node for routing.
- Since CSMA and 802.15.4 are most common, the issue becomes the radio duty cycling layer.

Radio Duty Cycling Layer

- To reduce idle listening, radio transceiver must be switched off most of the time.
- Figures show ContikiMAC for unicast and broadcast sender {similar to X-MAC}.
- ContikiMAC sender "learns" wake-up phase of the receivers.
- Performance relationship between RPL and duty cycling layer yet to be studied.

ContikiMAC Unicast

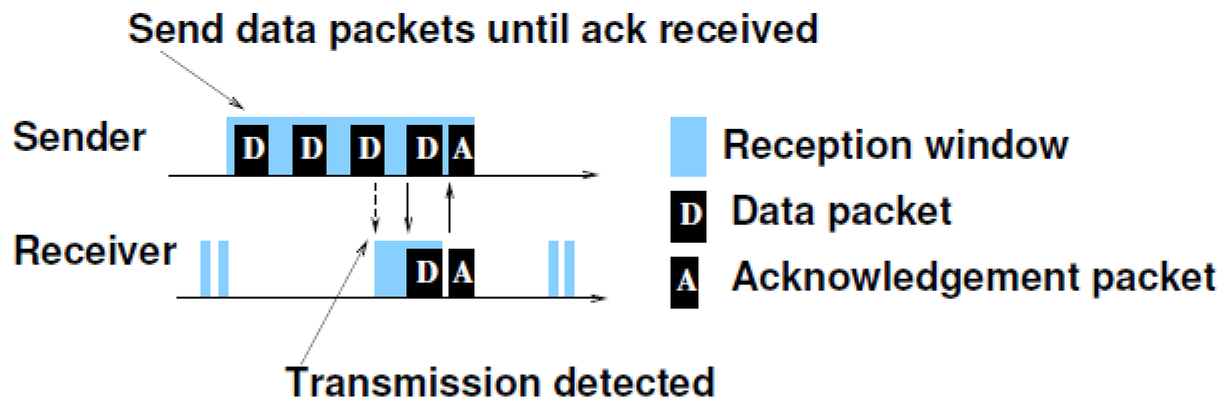


Figure 2. ContikiMAC, from Dunkels et al. [2].

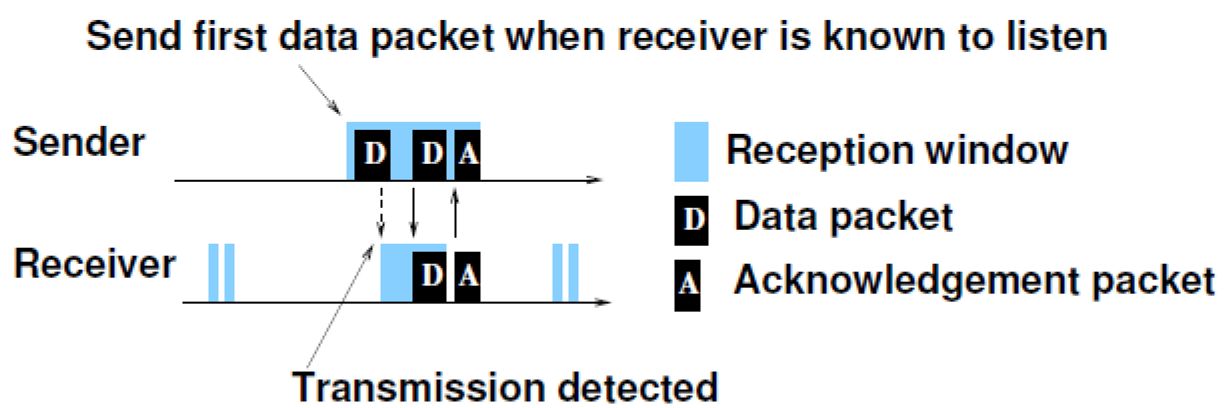


Figure 3. ContikiMAC sender phase-lock.

ContikiMAC Broadcast

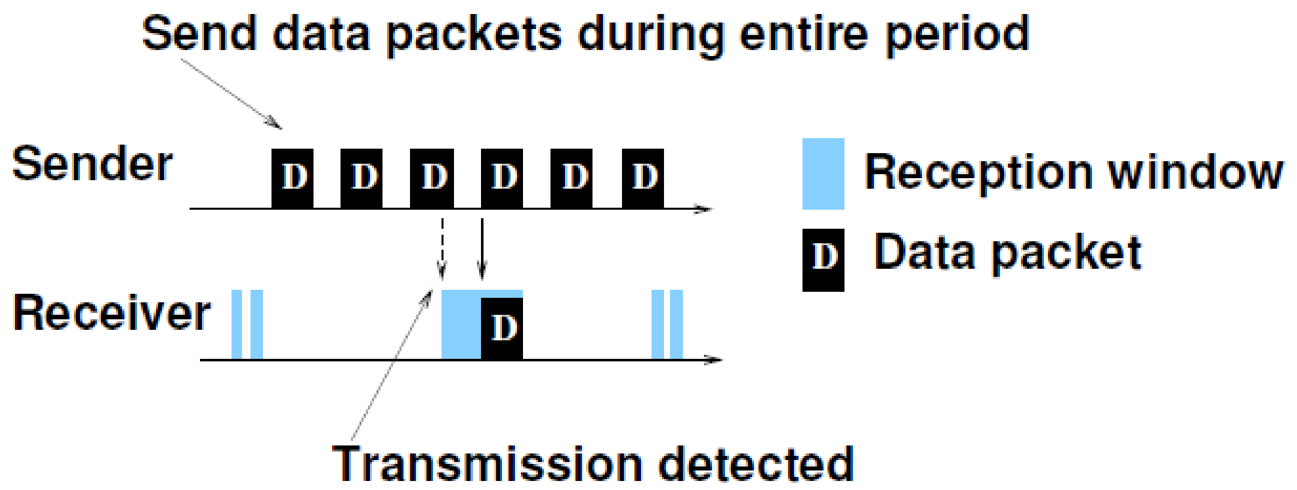


Figure 4. ContikiMAC broadcast.

ContikiMAC broadcast is the same as the *A-MAC* broadcast scheme.

Interoperability

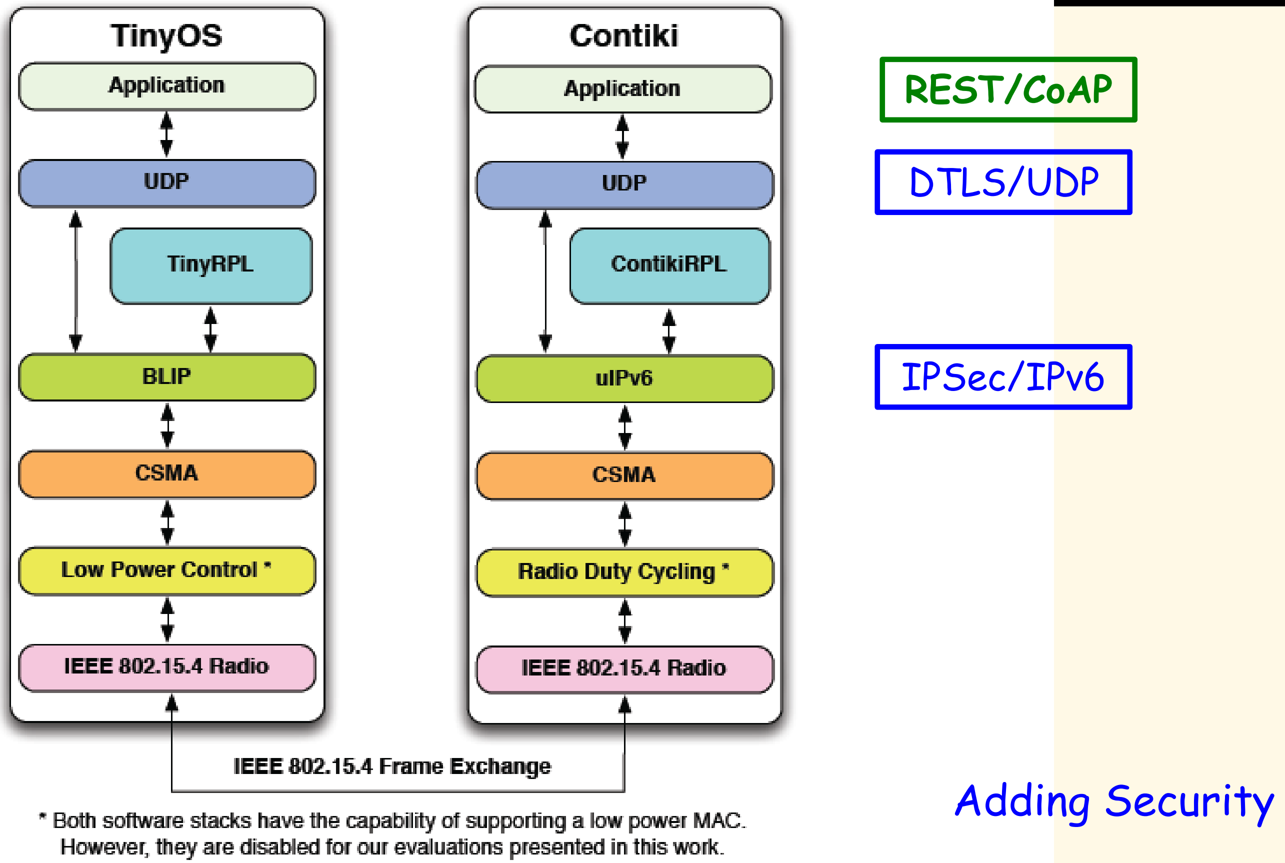


Figure 5. Contiki and TinyOS IPv6 interoperability, from Ko et al. [6]. We demonstrated interoperability at the network layer, the MAC layer, and the link layer, but without radio duty cycling.

Low-Power Interoperability

- Interoperable radio duty cycling is essential!
- Thus far interoperability demos have ONLY been with always-on radio layer.
- Contiki simulation tool can be used to study challenges of low-power IPv6 interoperability.

Low-Power Interoperability

Three challenges:

1. Existing duty cycle mechanisms NOT designed for interoperability.
   - e.g., ContikiMAC and TinyOS BoX-MAC have no formal specifications.
2. Duty cycling is timing sensitive.
   - Makes testing of interoperability difficult.
3. Current testing done via physical meetings of separate protocol developers.

Conclusion

- Attaining low-power interoperability for the Internet of Things is still an open problem because:
  - Existing protocols are not designed for duty cycling.
  - Existing duty cycling protocols are NOT designed for interoperability.

References

[Colitti] W. Colitti,K. Steenhaut and N. DeCaro, *Integrating Wireless Sensor Networks with the Web*, *from Extending the Internet to Low Power and Lossy Networks (IP+SN 2011)*, Chicago, April 2011.