Fostering IoT Deployment

Challenges and Assets of SDN Techniques

C. Jacquenet

christian.jacquenet@orange.com

Outline

- Context
- On IoT networking, routing and service design
- Additional IoT challenges
- A software-defined approach to IoT networking
- SDN-based IoT service production chain
- IoT instantiation of SDN meta-functional blocks
- On virtualization
- Conclusion

Internet of Things

- Where physical items are connected
  - Remotely controlled
  - Access points to Internet services
- Things are computerized
  - From sensors and actuators to electric toothbrushes, washing machines and fridges
- Things network with Internet resources, communicate and cooperate with each other
  - Without any specific human interference

Context

- Foreseen tens of billions of objects (sensors, actuators, controllers, *etc*.) deployed for a plethora of usages
  - Objects communicate over a networking infrastructure by means of various, possibly service-inferred designs
- IoT infrastructure varies in scope (home, access, metro and beyond) and techniques (wired, wireless, a combination thereof)
  - Depending on the nature of the service

On IoT Networking

- Network scale is different
  - Up to several thousands of nodes, yielding path computation issues with current routing protocols
- Technology and environment are (often) constrained
  - Low powered devices, unstable communication links and outdoor installation raise new (security) issues
- Traffic patterns are very specific
  - Data are collected by sensors and forwarded to actuators and/or Internet “gateways” (*e.g*., CPEs), yielding N:1 group communication scheme
  - Control traffic (*e.g*., commands) is also very unidirectional, yielding 1:P group communication schemes
    - P2P traffic patterns remain somewhat marginal

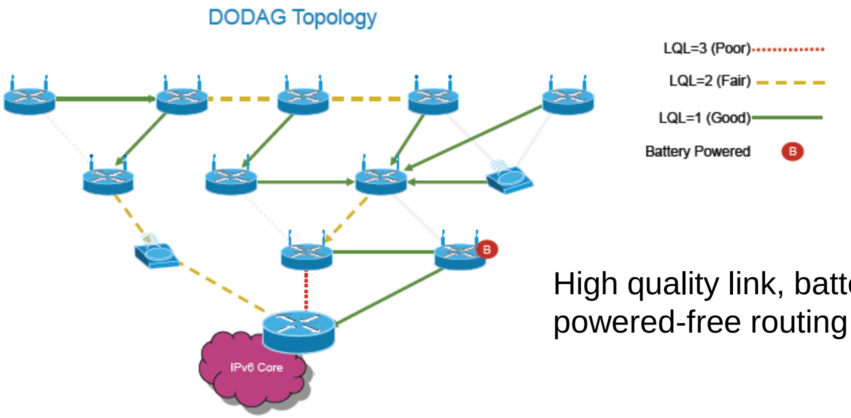
On IoT Routing

- Nature and scope of the service combined with device technology affect route computation schemes, *e.g.*,:
  - Privacy to be considered as a metric for e-health services
  - Specific constraints (like energy consumption status) likely to impact forwarding decisions

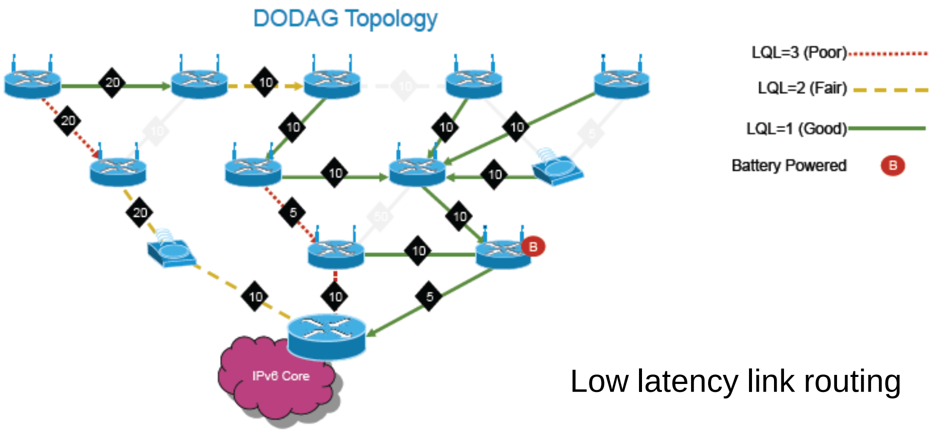
Internet Routing	IoT Routing
Nodes are routers	Nodes can be anything – sensors, actuators, routers, *etc*.
~100 node magnitude per network	Up to 1,000+ node magnitude per network
Links and nodes are stable overtime	Some links are highly unstable (lossy) and nodes fail more often (battery and CPU limitations, *etc*.)
No major routing constraint (so far)	Highly-constrained environment (energy, CPU, outdoor conditions, *etc*.)
Routing is not application-aware by default	Routing is application-aware

Need for Advanced IoT Routing Policies

- Objective Function (OF) of IoT routing can be manifold *e.g.*,:
  - Use high quality links and avoid battery-powered nodes
  - Use low latency paths only
- Use of combined metrics to address goals defined in OF, *e.g.*,:
  - Expected Transmission Count (ETX) metric combined with Hop Count metric to preserve energy and privilege traffic load balancing



High quality link, battery-powered-free routing



Low latency link routing

On IoT Service Design

- IoT service design assumes the combined and possibly ordered activation of elementary capabilities or Service Functions (SF)
  - Forwarding and routing, firewall, QoS, DPI, *etc*.
- IoT service complexity suggests robust mastering of chained SF activation and operation
  - For the sakes of optimized resource usage and reliable service delivery

A Few Additional IoT Challenges

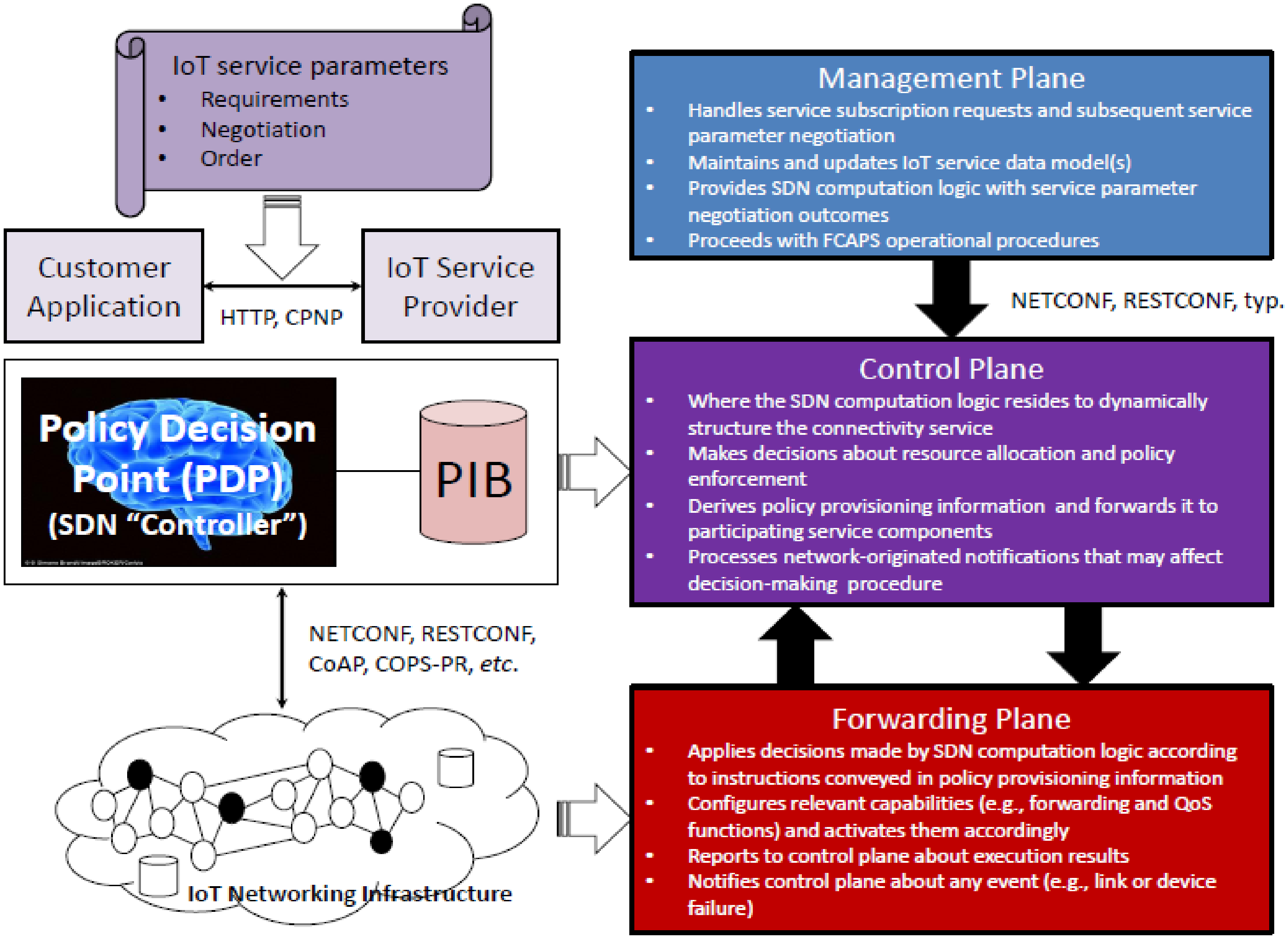
- Scalability and performance
  - Potential large scale requires efficient name resolution and forwarding paradigms
- Dynamic service discovery
  - Services for things must be identified for proper operation
  - Requires appropriate semantics to describe service capability
    - Can also be user-driven (*e.g*., human/machine interaction for dynamically geo-located parking facilities)
- Mobility
  - Locate things and the services they support while in motion
- Security and privacy
  - Selective access to specific services (*e.g*., monitoring of biometric data)

Rationale for Software-Defined IoT Networking

- Introduce *robust automation* in IoT service delivery for the sakes of cost optimization and improved service production times
  - Based upon a set of IoT service-specific policies
  - According to IoT customer/app requirements, possibly yielding a dynamic negotiation of IoT service parameters
- Use *dynamic resource allocation and policy enforcement schemes*
  - Likely based upon the use of various protocols and tools, given the broad heterogeneity of IoT technologies
- Activate *feedback mechanisms* to assess efficiency of IoT service delivery procedure
  - Verify that what has been delivered complies with what has been negotiated

SDN Landscape

SDN is defined as a "set of techniques used to facilitate the design, the delivery and the operation of network services in a deterministic, dynamic and scalable fashion" (RFC 7149)

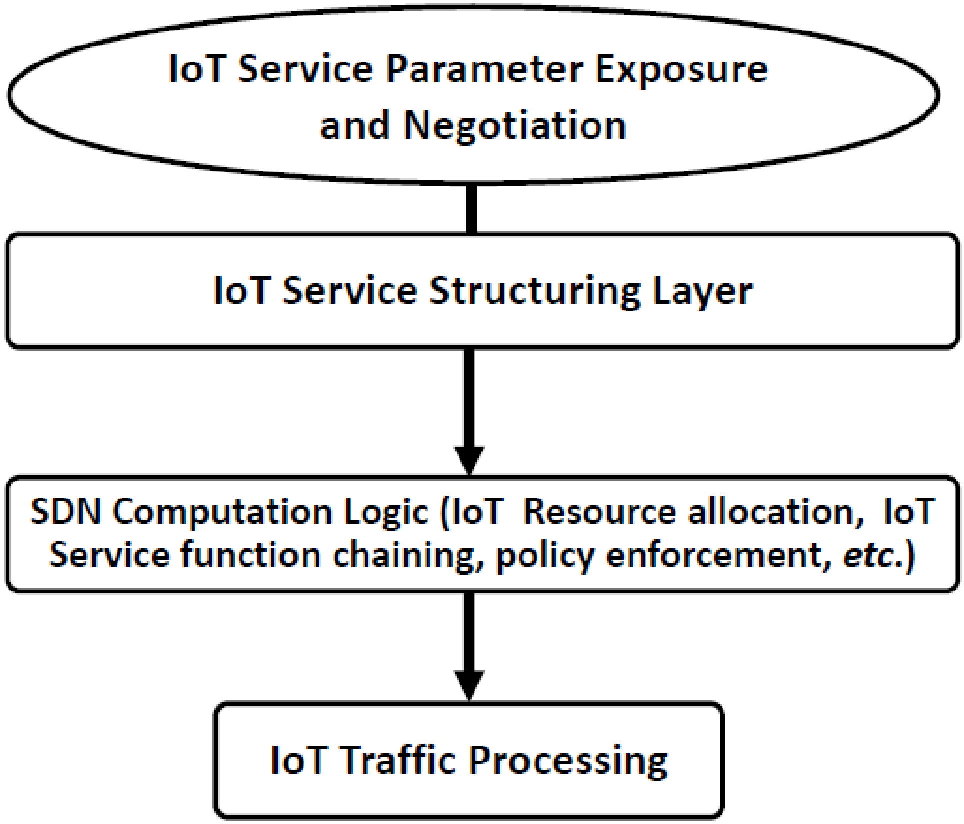


SDN Meta Functional Blocks

- Discovery of IoT network topology, devices and their capabilities
  - Acquired information stored in IoT service repositories according to (standard) data models
- IoT service exposure and parameter negotiation
  - By means of standard, commonly agreed, Connectivity Provisioning Profile-like (CPP, RFC 7297) templates
- Policy enforcement and resource allocation schemes
  - Based upon automated configuration procedures
- Feedback mechanisms
  - To assess how efficiently a given policy (or a set thereof) is enforced from a service fulfillment and assurance perspective

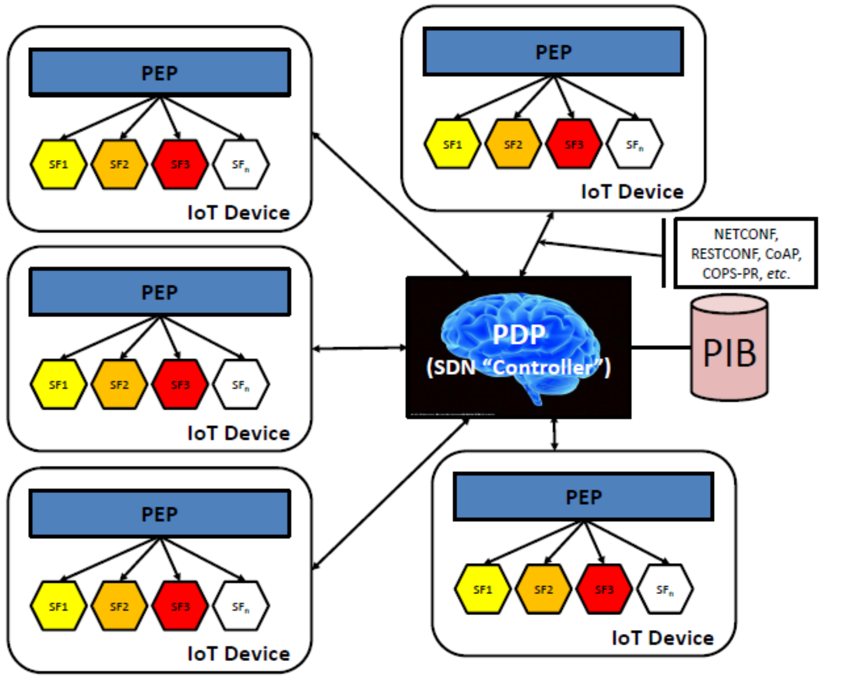
SDN-Based IoT Service Production Chain

- Outcomes of IoT service parameter negotiation feed SDN computation logic
  - Along with other inputs, like network-originated notifications and available resources
- IoT service is structured accordingly
  - Based upon abstract IoT service components depicted in (service-inferred) data models
- SDN computation logic then allocates IoT resources (network, storage, CPU)
  - Forwards policy decisions and configuration information to participating devices



Dynamic IoT Resource Allocation

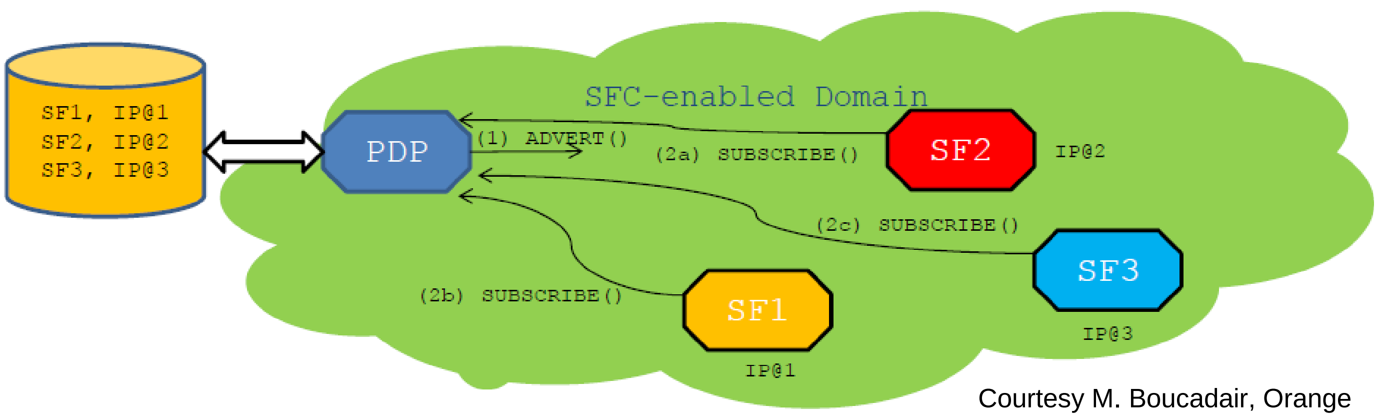
- A la Policy-Based Management
  - Cornerstone of SDN computation logic
- Policy Decision Point (PDP) makes decisions based upon various inputs
  - Outcomes of IoT service parameter negotiation
  - IoT network planning policies
  - IoT network-originated notifications
- PDP then derives IoT policy provisioning information
  - Based upon IoT service abstraction models stored in Policy Information Base (PIB)
  - Forwards corresponding configuration information to participating IoT service components for resource allocation and policy enforcement purposes
- Policy Enforcement Points (PEP) embedded in IoT devices apply PDP-made decisions
  - Configure and activate IoT service functions
  - Monitor IoT SF/device status and report to PDP



- Heterogeneous environments and technologies encourage standard IoT service data models
- Various protocols can convey information between IoT SDN controller (*a.k.a.* PDP) and PEPs
  - PCEP, NETCONF, CoAP, COPS-PR, *etc.*

Dynamic Discovery Capabilities

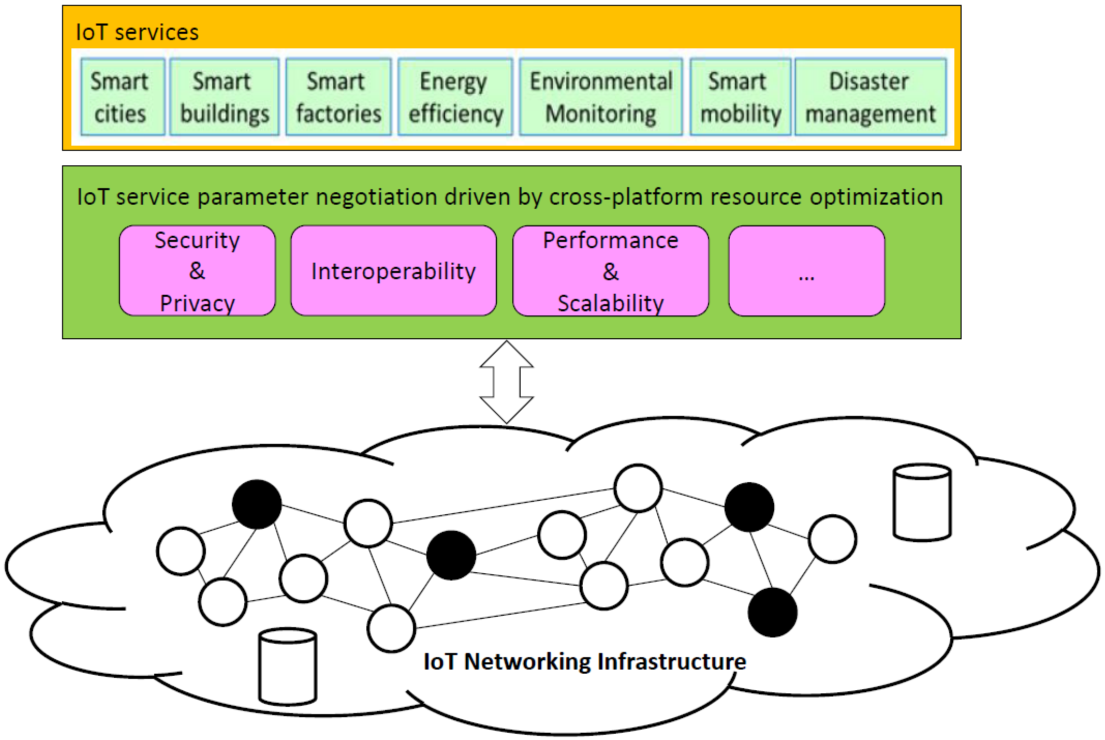
- IoT SDN bootstrapping procedure suggests dynamic acquisition of IoT SF (node) information
  - Can be advertised by means of specific ICMPv6 option or IGP
    - Each SF can be subTLV-encoded and the PDP belongs to the IGP control plane
  - Can also rely upon a kind of pub/sub procedure
    - PDP advertises its presence within the IoT networking infrastructure *e.g.*, by means of ICMPv6 option
    - IoT SF nodes then reply to the PDP with a description of their functional capabilities and other information, like reachability and SF-specific status



Courtesy M. Boucadair, Orange

IoT Service Parameter Exposure and Negotiation

- Multi-clause IoT service parameter template to
  - Accommodate specifics of a large variety of IoT services
  - Facilitate multiparty-operated resource integration
    - *E.g.*, cross-platform cooperation
- Clauses are manifold
  - Device geo-location
  - QoS requirements
  - Privacy requirements
  - Flow identification
  - *Etc.*



IoT Service Function Chaining

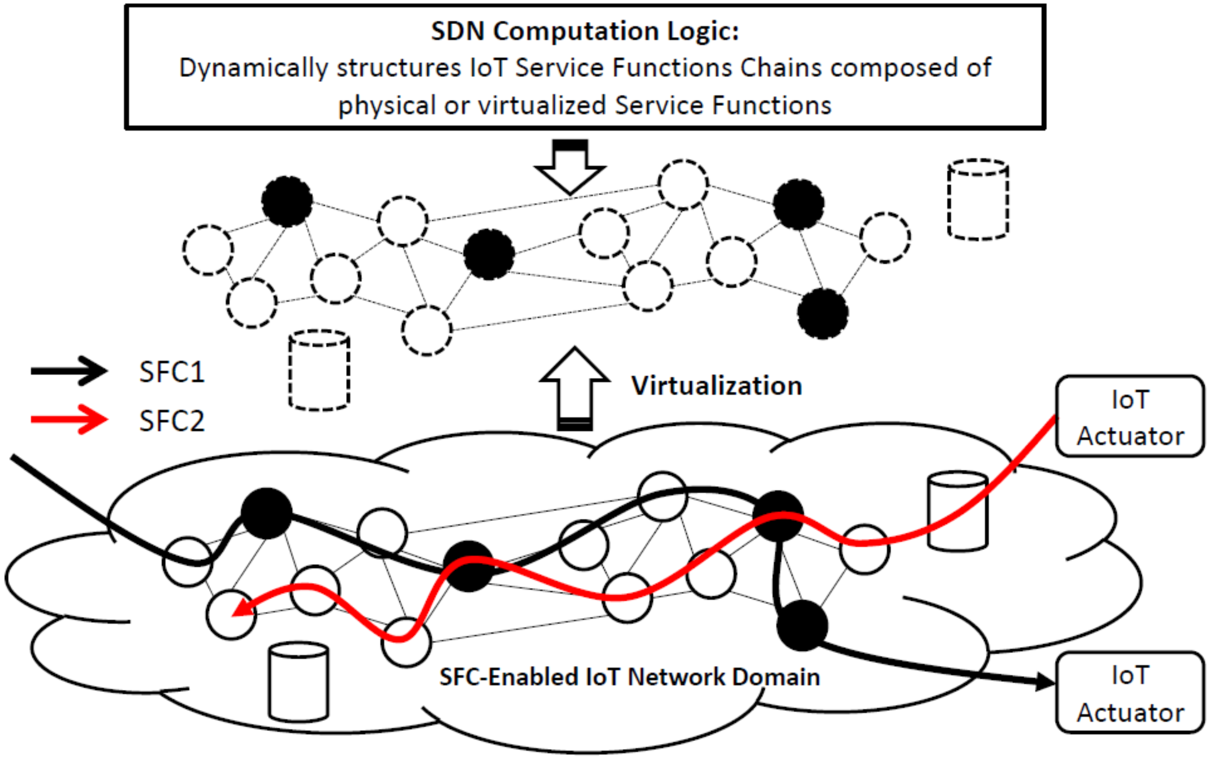
- Compute and establish IoT service-inferred forwarding paths
  - Thereby contributing to the optimization of overall service delivery and operation
- Master IoT Service Function (SF) chaining regardless of the underlying topology and routing policies
  - Yielding an IoT SF-based differentiated forwarding policy enforcement scheme
- Facilitate IoT SF operation while avoiding any major topology upgrade
  - Derive chronology of SF invocation according to the required service and associated parameters

IoT SFC Specifics

- Adaptation and mapping functions are required to accommodate a large variety of (proprietary) protocols and technologies
- A wide range of IoT-specific, tunable capabilities
  - Sleeping mode and duty cycle management (*e.g*., Report Period settings)
  - MTU settings
  - IoT service-inferred routing metric settings (*e.g*., ETX, latency, energy metric settings)
  - Objective Function (derived metric settings)
  - Security features
  - *Etc*.

IoT SFC Chaining Example

- Example chains reflect two different services (*e.g.*, e-health and home automation) where:
  - SFC1 = {DPI; 6lo (WPAN) encapsulation; ETX setting; 6lo de-capsulation}
  - SFC2 = {DPI; 6lo (NFC) encapsulation; CoAP-to-HTTP; 6lo de-capsulation}



SDN-Based IoT Service Management

- Diagnose and repair
  - Need for SDN-adapted OAM tools
    - Pinging an IoT function only means it's reachable (does not mean it's operational)
  - Detect and proceed with corrective actions
    - Verify completion of forwarding actions
    - Detect IoT SF liveliness
    - Assess status of IoT SF serviceability
    - Provide a collection of counters and statistics as part of abstract IoT service models (*a la* feedback PIB, RFC 3571)
- Determinism remains key
  - (Re-)programming must reflect predictable behavior
    - Allocated IoT resources and enforced policies are derived from completed IoT service parameter (re-)negotiation phase

IoT Virtualization: What For?

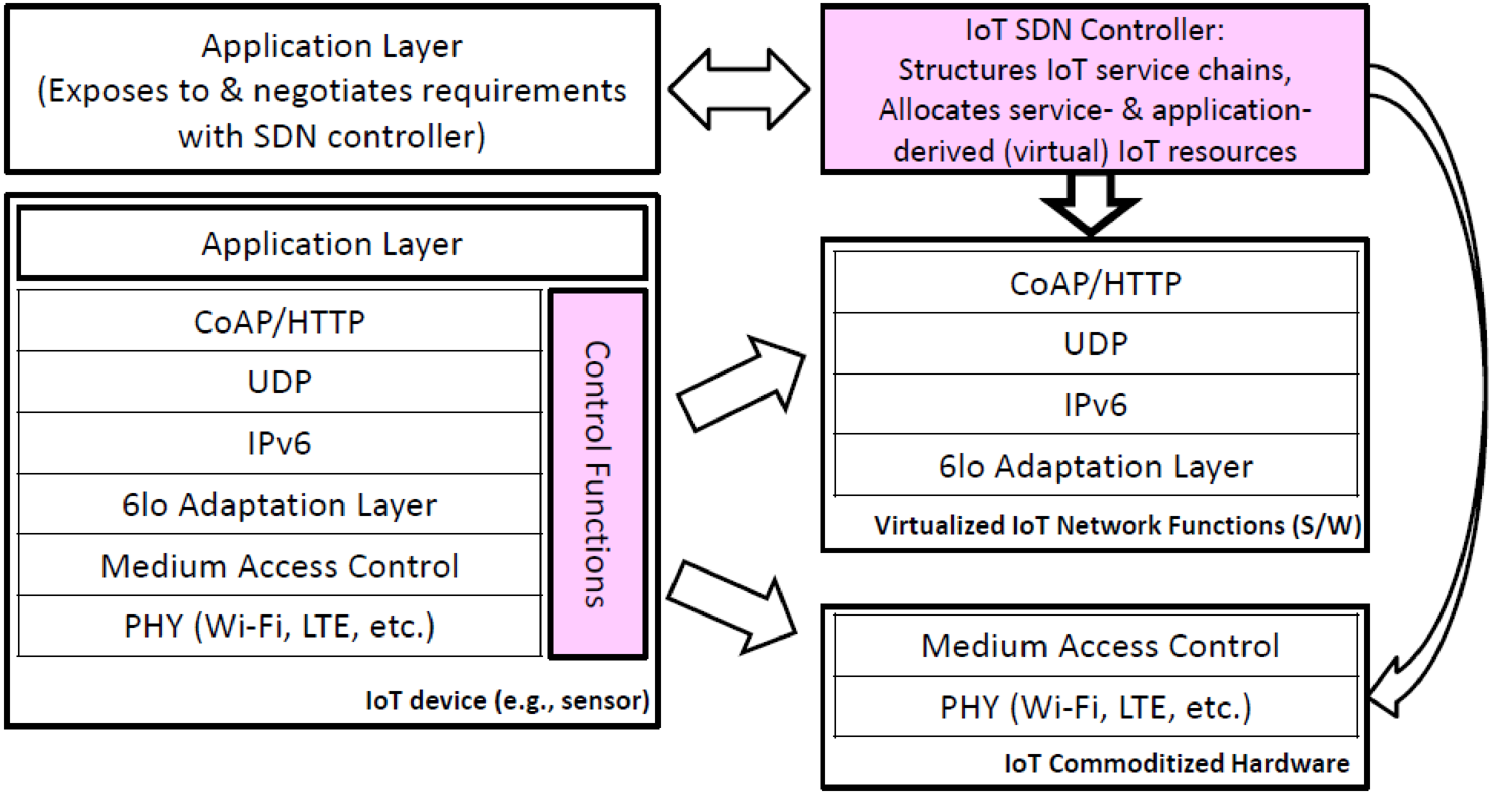
- Current IPv6-based technology footprint is a few tens of kilobytes
  - Including OS-level services like RIB management
- But some features are CPU-intensive although optimizable
  - IPv6-specific signaling traffic (NS/NA exchange)
  - RPL protocol machinery
- While others consume most of the energy
  - Radio transmission component
- There is therefore room for optimization where virtualization can help

Component	ROM	RAM
CC2420 Driver	3149	272
802.15.4 Encryption	1194	101
Media Access Control	330	9
Media Management Control	1348	20
6LoWPAN + IPv6	2550	0
Checksums	134	0
SLAAC	216	32
DHCPv6 Client	212	3
DHCPv6 Proxy	104	0
ICMPv6	522	0
Unicast Forwarder	1158	315
Multicast Forwarder	352	4
Message Buffers	0	2048
Router	2050	64
UDP	450	6
TCP	1674	48

Source: J. Hui et al., 2008, 2010, 2013

Virtualizing IoT Service Functions

- Example of an IoT device



Source: “A Software-Defined Networking Architecture for the internet of Things”, 2014

Takeaways

- SDN for IoT is not science-fiction
  - Some vendors already claim early SDN-labelled prototype implementations
  - Standardization effort is underway
    - Bootstrapping aspects, CoAP-based IoT resource management and data models are being specified by IETF and IPSO in particular
- Beneath SDN lies a complex combination of various techniques and protocols, let alone computation logics
  - This complexity is the key challenge, from service parameter exposure and negotiation to resource allocation and service fulfillment
  - Signaling traffic may also affect expected overall performance, flexibility and scalability
- IoT scale and figures suggest considerations on hierarchical SDN designs
  - Service- or geographically-driven

Thank You!