Association for Computing Machinery
Advancing Computing as a Science & Profession

*Ubiquity,* an ACM publication
December 2015

**Ubiquity Symposium**

# The Internet of Things

## Evolution and Disruption in Network Processing for the Internet of Things

***By Lorenzo Di Gregorio***

### Editor's Introduction

*Between prophecies of revolutions and inertiae of legacies, the Internet of Things (IoT) has already become the brand under which light processing units communicate over complex networks. Network processing is caught between demands for computation, raised by the growing complexity of the networks, and limitations imposed by performance of lightweight devices on processing. In this contribution the potential for disruptive changes against the scaling of existing technologies is discussed, specifically three main aspects of the IoT that impact network protocols and their processing: the reversal of the client/server architectures, the scavenging of spectral bands, and the federation of Internet gateways.*

**Ubiquity Symposium**

# The Internet of Things

## Evolution and Disruption in Network Processing for the Internet of Things

***By Lorenzo Di Gregorio***

Network processing emerged as a discipline in the mid-1990s prior to the dot-com bubble, driven by expectations of ever increasing network connectivity services and data rates. While much of that hi-tech rush was focused on silicon for network processors and has vanished over the subsequent decade, some of its main characteristics have turned into lasting traits of many mainstream products for networking beyond the sole Internet core—materializing mostly around functional decomposition and offloading of protocol processing workloads to specialized search and packet processors.

While the architecture of the Internet in the post new-economy decade has evolved to sustain rapid growths at its edges through intelligent aggregators in DSLAMs (digital subscriber line access multiplexers) and cellular base stations, cost pressure on carriers and service providers in the core areas has pushed consolidation of processing equipment toward centralized routers and cloud servers. Between core and edges, metropolitan area networks have grown large and populated by fast auto-configuring link-layer switches, mostly agnostic to upper layer protocols.

The Internet of Things (IoT) threatens this state of affairs by promising large added values for breaking the existing paradigms with processing platforms based on low-performance, low-power monolithic controllers; proliferation of autonomous systems at the Internet edges; decentralization of processing capabilities toward these edges; exposure of multitudes of massively heterogeneous status indicators; and heterogeneous control of common coordinating applications.

This article will focus on the disruptive potential of three further aspects of the IoT, which are deeply related to network protocols and their processing: the reversal of the client/server architecture, the scavenging of spectral bands, and the federation of Internet gateways.

**All-IP Reversed Client/Server Architectures**

There is little rationale in challenging the foundations of the ubiquitous Internet Protocol (IP); the computing resources demanded for supporting the IP network layer are not large. Implementations of the popular lwIP stack have shown TCP/IPv6 connectivity is feasible in lightweight embedded systems within about 20 KB ROM / 10 KB RAM [1]. Actual implementations range from 29 KB ROM / 17 KB RAM (Atmel SAM4N series) to about 33KB ROM / 34 KB RAM (NXP LPC177x_8x series), the latter including some additional buffer to prevent packet drops and optional 10 KB RAM for DHCP and UDP. On the MSP430 instruction set, Contiki delivers drivers for IP routing through the RPL protocol (RFC 6550) within about 5 KB ROM / 0.5 KB RAM and for the application layer through the CoAP protocol (RFC 7252) within about 8.5 KB ROM / 1.5 KB RAM. These figures are well below the capacities of lightweight cyber-physical systems available on the market nowadays, and obviously not all systems need all drivers, e.g. sensor endpoints within a wireless sensor network will not act as routers.

Though, the performance of IP transport over low power wireless and noisy links is known to be poor, it is the need for remedies like fragmentation, header compression, and mesh forwarding that has geared 6LoWPAN up for establishment as an adaptation layer between standard IPv6 and the PHY / MAC layers of IEEE 802.15.4.

The combination of IPv6 and 6LoWPAN adaptation is largely established as an evolutional network layer for the IoT. In contrast and despite of proven feasibility, upper layers see far less convergence toward demands from the network management communities for reliable transport and virtual private networks (VPN).

Reliable transport has been tackled both by compressing TCP headers over 6LoWPAN [2] and by extending ARQ (Automatic Repeat Request) protocols for 6LoWPAN to end-to-end transport [3]. Reliability has also been delegated to application layers through the CoAP protocol, geared toward the “Web of Things” and designed for ease of translation to HTTP. The solutions space for VPN is less clearly defined. It is widely perceived as related to encryption, while it is actually a set of tunneling techniques employed as segregation measure, largely driven by demands of IT managers to guarantee incumbent network functionality against the novel IoT.

The evolutionary path of “miniaturizing” Internet technologies to implement the IoT clashes with a fundamental ongoing change: the reversal of client-server architectures. Hesitancy in

TCP and VPN deployment can be interpreted as side effects of this change and as an approximation to the limits of the incumbent paradigm.

When a new device connects to the Internet, it produces data that must be routed to some consumers. Many device vendors are inclined to solve the problem of their devices autonomously finding remote consumers or producers by turning every device into a small server, hence pushing the problem away from said devices and implicitly reversing the widespread Internet access use case, which sees the multitude of endpoints on the Internet edge acting as clients of centralized server. This reversal shifts the network processing burden of servicing requests from high-end hardware to a scattered multitude of lightweight edge devices, which must get discovered despite intermittent network presence and be dimensioned for stateful connection management within a wide range of scenarios including virtualization. For example, supporting several applications might require supporting multiple legacy protocols as well as retaining several contexts with TCP sockets open and several packets ready for retransmission to carry out anti-spoofing and anti-flooding procedures, to cross legacy NAT gateways through explicit port forwarding or UDP hole punching, or to maintain a large number of VPN tunnels. All this is costly, and largely unnecessary, if all that needs to be transported is just a bunch of messages.

The alternative to turning every device into a small server, in need of being polled for discovery, consists of local network architectures sustained by publish/subscribe protocols like MQTT [4], which envisions agents between producers and consumers to operate as brokers in charge of collecting and passing messages—in fact aggregating the server functionality of multiple endpoints.

Such agents are not specific to MQTT and can actually be conceived for many protocols. However, they are generally regarded with skepticism in the mainstream network paradigms because they act as dedicated routing gateways, which break the end-to-end principle that application specific functions should reside in end hosts and not in intermediate ones. Software-defined networking (SDN) primitives and QoS (quality-of-service) policies often clash with incompatible features or lacking functionality in such intermediate hosts. On the other hand, in the emerging paradigm for the IoT, such agents provide a fundamental decoupling between producer and consumer: This decoupling enables not only asynchronous message passing across sleep phases of components, but also proxying device discovery and status poll responses for sleeping devices or subnets. Intermediate agents further enable routing functionality, e.g. as root nodes for RPL as well as for fast ad-hoc sensor-to-sensor routes, and

also enable compliance with legacies, e.g. translating CoAP to HTTP over TCP (messages with reliability) or UDP (messages without reliability). In fact, experience with SCTP (RFC4960) and SPDY [5] shows established protocols are supported by masses of equipment with tailored optimizations (TCP acceleration and web front-end optimization), and are hence extremely resilient to displacements even beyond the availability of full software functionality within deployed operating systems and applications.

The functionality as intermediate agent is the one that mediates the reversal of client/server architecture and is ultimately shaping the vision of myriads of interconnected nodes in the IoT. The question is whether this functionality shall be evolutionally supported by established protocols and equipment, like home gateways, or whether a disruption will take place. For example, opening possibilities for self-organization and cooperation schemes among everyday appliances, or enabling novel families of resilient control and calibration algorithms over deterministic networks.

While cloud computing has been driven by the reduction of operating costs for computing infrastructures, the introduction of intelligent intermediate agents represents a move of computation resources back toward the network edges, for example because additional computation resources will be demanded to support route definitions from software-defined networking controllers. Such a vision of computation migrating back from the centralized cloud toward the network edges is known as “fog computing” and will incur increased total costs. A contrast becomes visible against the cost pressure and energy constraints that drive the ongoing evolution of the IoT, characterized by the definition of cost- and energy-convenient topologies at the network layer as rooted graphs with nodes relaying packets across many-to-root and root-to-many routes (e.g. RPL protocol). Intelligent intermediate agents incur additional costs, which are justified only if substantial value is generated when data can be routed to computing resources closer to the endpoints.

The vision of computation and routing being unified, with network nodes that carry out partial computation and route outcomes to a next node for further computation, dates back to the 1980’s. When it was represented at Sun by John Gage’s catchphrase “the network is the computer.”

In the subsequent decades since this vision has not materialized. Classical disruption according to the definition given by Christensen [6] would now take place if such a model—not attractive in the incumbent paradigm—would gain traction under some novel burgeoning paradigms driven by novel applications in the IoT.

### Scavenging Spectral Bands

Proliferation of short-range radio devices bears a straightforward effect: Unlicensed spectral bands get overcrowded. The capacity of a band to accept communication channels depends obviously on the width of the band, but also on the widths of the channels and on the gaps between channels, called white spaces and left unused to reduce interference. In order to accommodate sufficient channels, bands are defined and assigned by public authorities. The current width of total wireless broadband spectrum consistently available across Europe sums up to about 990 MHz, and the European Commission targets overall harmonization of 1200 MHz across Europe by 2015, with ongoing investigations to extend the unlicensed bands above 5 GHz and ease the adoption of UWB (ultra-wide band) technologies [7]. Though, the total bandwidth available to the IoT is actually far narrower. Table 1 reports an overview of some widespread short-range radio technologies along with the most commonly achieved data rates, used bands, typical sensitivity, demanded transmit power, and achieved link budget as a measure of robustness.

| Technology | Data rate | Band | Sensitivity | Tx power | Link budget |
|---|---|---|---|---|---|
| **Zigbee** | 250 kb/s | 2400 MHz | -98 dBm | 8 dBm | 106 dB |
| **Bluetooth** | 1 Mb/s | 2400 MHz | -85 dBm | 7 dBm | 92 dB |
| **Z-Wave** | 40 kb/s | 900 MHz | -101 dBm | up to 0 dBm | 101 dB |
| **DECT** | 1 Mb/s | 1900 MHz | -98 dBm | 25 dBm | 123 dB |

**Table 1. Comparison of short-range radio technologies [8].**

Spectral overcrowding is not yet an everyday issue and some time will go by before it becomes one, but it is not unlikely to run into such situations nowadays. For example, Zigbee operates through 16 channels, numbered 11 to 26, each of 2 MHz spaced by 3 MHz in the same 2.4 GHz band as Bluetooth does with 79 channels of 1 MHz spaced by 2 MHz. In the same band, Wi-Fi bears with 14 channels separated by 5 MHz and whose spectrum is spread over 20 MHz with 5 MHz of white spaces. While these standards are able to coexist in practice, weak signals get easily shut down in proximity of stronger ones. For example, Wi-Fi devices, according to the IEEE 802.11 standards, bear a much stronger signal than Zigbee ones: three Wi-Fi devices

allocated at channels 1, 6, and 11, leave only the four Zigbee channels (15, 20, 25 and 26) of weak signals free of interference when in proximity of stronger Wi-Fi devices.

While there is wide convergence for short-range radio toward Wi-Fi, low power battery backed operations still rely on several short-range radio technologies. Table 2 gives an overview of the main performance figures achieved on widespread short-range air interfaces. This table disregards improved standard extensions like the Bluetooth Smart, because it has not yet reached a comparable level of adoption. Instead DECT ULE (ultra low energy) is used for comparison, because it builds on the large legacy of the DECT band and equipment.

Over the last years, these technologies have seen improvements in modulations and duty cycles to increase data rates per energy and distance units. While adoption of UWB has gone far on a long and troubled road to minimal market acceptance, the novel ULE extension of DECT has considerably reduced the duty cycles and promises market acceptance by building on its own worldwide, commonly regulated, royalty-free band, with a link budget that allows for much wider coverage than its competing technologies.

Further evolutions might see smarter channel management. For example, frequency hopping might avoid known busy channels and devices might initiate connections over narrow channels for low-power, long-range connectivity, widening these channels for performance [9] when demanded and feasible. Cooperative ARQ protocols might get employed on links where energy and memory are sufficient for buffering and retransmitting packets subject to bit errors.

| | **Bluetooth** | **ZigBee** | **Z-Wave** | **DECT ULE** |
|---|---|---|---|---|
| **Reach (in- / outdoor)** | 10 m | 10 / 75 m | 10 / 30 m | 50 / 300 m |
| **Antenna Power** | 100 mW | 30 mW | 25 mW | 250 mW |
| **Gross bit rate (depending on protocol)** | 9.6 - 80 kbps | 40 - 250 kbps | 9.6 - 40 kbps | 2.1 - 307.2 kbps |
| **Typical Battery life** | 3 - 6 months | 24 - 46 months | 24 - 60 months | 24 - 60 months |

**Table 2. Performance of royalties-free technologies, with Bluetooth V1.x and DECT ULE extension.**

In situations where all channels are occupied, evolutional channel management techniques cannot provide solutions: Overcrowded environments are a challenge in particular for devices, which have to wake up and autonomously obtain initial connectivity to pull setup instructions

or merely issue a warning that remedies are needed. Ongoing research is tackling so-called secondary channels, which are channels obtained by sensing of short free transmission gaps within active channels [10]. Since most devices in the IoT are expected to transmit only sporadic messages, a large number of secondary channels are expected to be identifiable. These techniques disrupt the existing MAC layers by demanding several different variations on common MAC functionality. For example, to negotiate parameters and policies to decide, in case the primary channel interrupts transmission of the secondary one, whether to back off or switch to a different channel. This demand for flexibility calls for multiplicities of network processing algorithms to be executed on programmable MAC controllers, which are also motivated by further application domains like protocol stack virtualization in software-defined wireless networking.

Next to overcrowding, the bursty nature of several applications, like Internet video surveillance, is likely to demand higher gross data rates and might push disruptive approaches. In the attempt to scavenge free bands, wireless carriers from multiple stations could be aggregated to improve data rates of a single interface when peak traffic arises. In such cases, channel bonding algorithms along the lines of segmentation and reassembly procedures born in Ethernet trunking, Multilink PPP, or IEEE link aggregation will find application. In such cases, large data flow is expected from sensors to a collector, but a feedback mechanism toward the sensors must exist to avoid some unreliable trunks increasing the packet jitter to a level not tolerable by a reassembly buffer.

While carrier aggregation tackles utilization of existing communication channels, research in wireless sensor networks has recently tackled a raise in the capacity of communication channels. Two techniques can be employed for exploiting spatial and temporal correlation of many natural signals: Compressive sensing [11] can lead to reconstruction of signals sampled below the theoretical limit of Nyquist, and matrix completion [12] can lead to the reconstruction incomplete data sets collected from multiple sources. These techniques demand more energy for computation on the receiver side. But in return for this cost in energy, they enable transmitters to operate at higher error rates or lower power levels. Hence they are attractive for implementation in root nodes of sensor networks, because sensors are strongly power limited but they predominantly transmit data to a common collector, which is likely to be powered for acting as small server for the IoT.

Since compressive sensing and matrix completion exploit, respectively, temporal and spatial correlations of sensed natural signals, the network can deliver reconstruction of the original

signal only at nodes where sufficient sparse signals are collected. Up to that collection point, signals need to be transported, but protocols are intolerant of bit errors and drop affected packets. Depending on the characteristics of the signal being sensed and the size of the samples, resilient compressed packet headers might be employed to provide increased forward error correction only on few protocol fields, preferring to deliver sensed data in payloads subject to bit errors rather than dropping packets (an example of a related proposal is given in RFC 5109).

**Federated Gateways for Smart Relaying Objects**

With the hype machine of the IoT on tilt, several vendors have taken the route of advertising their tiny microcontrollers as the next generation engines of novel networking.

Though, in a monolithic processing system "small" is not the same as "low power," a device which is merely relaying traffic does not need to power up the same processor core it uses for image recognition. No matter how small and efficient this core can get. Transformational processing like data compression can benefit from speed, because this speed increases the time periods in which most of the device can be powered down. In contrast, reactive processing like protocol stack drivers are event driven and can benefit from slower operations, which consume less dynamic power because their powered down phases would be anyway interrupted frequently by data arrivals.

An evolutionary path for devices in the IoT is moving from tiny controllers to multiple chip—and will keep expanding into multicore chips—with the goal of employing coarse-grained power gating to selectively activate only components required for specific workloads.

A reasonable conjecture against the increase of programmability is that next-generation systems-on-chip might deliver energy saving through the acceleration of widespread standards (e.g., for video encoding) by hardwiring functionality into application-specific circuits. Though, the IoT challenges many such paradigms and calls again for programmability. For example, within a visual sensor network of smart cameras, lighter encoding increases the energy necessary to drive the antenna, but decreases the one necessary to video processing.

Concepts of computing platforms suitable for smart objects have been devised also in the past under the umbrella terms of ubiquitous computing, pervasive computing, or ambient intelligence. The characteristic trait of the IoT is given by the networking capabilities of such

smart objects and a disruption in the existing landscape of product design could be driven by cooperative networking. If smart objects in the IoT, next to carrying out their own task, operate also as relay servers for others, then a smart object's architecture with decoupled data and control planes might become remarkably more efficient than one based on monolithic or symmetric multi-core controllers. In such architectures—conceptually well known from network processor design although in a much different context—most of the chip is available for being powered down, while established traffic flows are switched through a dedicated and independent processing unit (the data plane). A control-processing unit (control plane) is required to intervene for reprogramming the data plane only in case incoming traffic does not match to the programmed flows.

Relaying network nodes are prospected by the RPL protocol, which enforces the definition of routes across nodes as a rooted graph, whose root is the actual router while all other nodes relay traffic toward and from this node. Nodes may belong to multiple rooted graphs and nodes in best positions must act as federated roots toward backbone connectivity. Despite of known weaknesses in RPL (see an analysis in [13, 14] the concept that network nodes cooperate in relaying messages from many endpoints to few collectors is well established in sensor networks. This cooperative paradigm calls for a disruption against existing router products: Any smart object can, in principle, be eligible as router and routing becomes a functionality embedded in other products rather than implemented on a dedicated device. In typical sensor networks, lightweight sensor endpoints will not post messages for their availability as routers, but any other networked device which will do so, will be eligible as A router, and root for its subgraph. Within home or factory environments, there will be several possibilities for WAN connectivity over Wi-Fi or wired connections, e.g., PLC (power line communication). If a device bears a WAN (wide area network) connection, then it can be a root node and must be able to federate with other root nodes.

The requirement that root nodes must be able to federate implies advanced network processing must be carried out to discover and administer services across a WAN: Every device advertising routing functionality must be able to learn whether traffic flows shall be processed in a remote cloud or in a local "fog computing" unit, and this unit can be the root node itself or a federated node. For example, a root node might either terminate 6LoWPAN and carry out compressed sensing of the measurements to forward data to the cloud via TCP sockets, or it might not bear such processing capabilities and must decompress 6LoWPAN packets (and compress return traffic) to forward them to another node over a federated IPv6 WAN. More conveniently it might obtain a VPN tunnel to route 6LoWPAN over it.

WAN functionality is commonly supported by Internet gateways. Simple message relay is considerably less demanding than federation, so processing platforms for the IoT might materialize in three distinguished families: sensors and actuators, smart objects, and WAN gateways. Sensors and actuators are endpoints that bear an ultra-lightweight protocol stack to connect to relaying agents through a wireless interface. Smart objects can act as relaying agents, supporting application-specific software, but bearing on top a data plane to relay packets at low power and a software suite to operate as a microserver when addressed. These microservers are reachable over the WAN through gateways, which are able to translate protocols and federate to reach resources like data mining servers.

Smart relaying objects must support peculiar network processing algorithms to deal with two fundamental aspects of sensor networks specific of the IoT: heterogeneity of air interfaces and node-to-node communications. Air interfaces are heterogeneous because they can present features largely outside of mainstream Internet technologies, like secondary channels and unidirectional links, which require variations on MAC layers and on routing technologies. Node-to-node shortcut routes fall outside of the paradigm consisting of rooted communication graphs with many-to-one and one-to-many communication, which is shaping the IoT. However, there are applications like calibration procedures where fast node-to-node communication is necessary. In such cases relaying nodes are required to configure direct routes as defined in SDN fashion by a central software instance.

Sensed information about private environments or health, transported across several nodes, all installed at widespread easily accessible locations and potentially eligible as routers, will turn security into a primary concern. A novel aspect in this area is the fact that intermediate nodes are simple designs exposed to easy physical access. A proven concept against manumission of intermediate nodes is onion routing [15], though it requires the initiator of a transmission to learn the whole route until its destination, retrieve public keys, and carry out one encryption pass per hop. This effort is likely to be excessive for practical applications. Since DTLS is already widespread in lightweight devices (e.g., through the Contiki OS), and approaches to network security appear largely scattered because of the plethora of energy-security trade-off points, a most likely evolution could be a SOA (service-oriented architecture) design pattern for deployment of DTLS.

Another rather advanced evolutionary path is driven by smallest devices, not battery-backed or battery-assisted at all: RFID transponders get commonly activated at 100 µW and operate at 500 µW, as long as they are within a few meters from the reader antenna. This power budget is

sufficient for some processing and for storing some data into non-volatile (FRAM) memories, but not for common workloads. Evolution can be expected in localization of RFID tags and beamforming by resonant phased arrays of antennas to increase the power budget, as well as low-threshold and low-capacity transistors to reduce load presented by the device to the rectifier of the coupled RF signal. Though, a more disruptive change in this area could be driven by near-threshold design techniques [16], which largely impact design flow and design methodologies. Such techniques enable scaling voltage supply from 0.8 - 1 V down to 0.2 - 0.3 V, leading to typical five times reduction in power consumption for about ten times reduction in operating frequency [17]. The effect of ultra-low power processing in RFID on network processing is that buffering and stateful operations must be minimized. In fact, as long as the transponder is being read, energy is scavenged to obtain a stable power supply. When the read phase is terminated and power is not available anymore, retention demands costly non-volatile memories and the timing for the next active phase is mostly not determined.

## Conclusions

This contribution has discussed the disruptive potential of three aspects of the IoT with respect to network protocols and their processing: the reversal of the client/server architecture, the scavenging of spectral bands, and the federation of Internet gateways. The overall outcome of this discussion could be lumped in the slogan that smart objects are more than lightweight Internet-connected objects and less than objects with an Internet gateway attached. Ongoing evolution for the IoT orientates toward smart objects operating as Internet-connected peers of data mining servers in the cloud, while an upmarket development exists in concentrating functionality and cutting costs. The potential for disruption consists in some classes of smart objects operating as a microserver toward Internet gateways, as aggregators toward lightweight endpoints like sensors and a software-definable ultra-low power packet relaying data plane for routing. If the dominating paradigm for the IoT will shift from Internet-connected objects to Internet-connected solutions consisting of several objects, then such disruptions might gain momentum.

## References

[1] Dunkels, A. and Vasseur, J. P. IP for smart objects. IPSO White Paper #1

[2] Ayadi, A. et al. TCP header compression for 6LoWPAN. IETF Draft. http://tools.ietf.org/html/draft-aayadi-6lowpan-tcphc-00

[3] Ayadi, A et al. Energy-efficient fragment recovery techniques for low-power and lossy networks. IN *2011 7th International Wireless Communications and Mobile Computing Conference (IWCMC)* (4–8 July 2011) 2011, 601–606,

[4] Locke, D. MQ telemetry transport (MQTT) V3.1 protocol specification. IBM. 2010. http://www.ibm.com/developerworks/webservices/library/ws-mqtt

[5] SPDY: An experimental protocol for a faster web. http://www.chromium.org/spdy/spdy-whitepaper

[6] Christensen, C. M. *The Innovator's Dilemma*. HarperBusiness, New York, 1997.

[7] European Commission. Report from the commission to the European Parliament and Council on the implementation of the radio spectrum policy programme. Document 52014DC0228, http://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1401178255384&uri=CELEX:52014DC0228

[8] Leussink, S. and Kohlmann, R. Wireless standards for home automation, energy, care and security devices. DECT ULE White Paper. Dialog Semiconductor B.V. http://www.dialog-semiconductor.com/sites/default/files/dect_ule_whitepaper.pdf

[9] Chandra, R. et al. A case for adapting channel width in wireless networks. *ACM SIGCOMM Computer Communication Review* 38, 4 (2008), 135–146.

[10] Penna, F., Garello, R., and Spirito, M. A. Distributed inference of channel occupation probabilities in cognitive networks via message passing. In *IEEE Symposium on New Frontiers in Dynamic Spectrum* (April 6–10, Singapore). IEEE, Washington D.C., 2010.

[11] Candès, E. J. and Wakin, M. B. An introduction to compressive sampling. *IEEE Signal Processing Magazine* 25, 2 (2008), 21–30.

[12] Candès, E. J. and Plan, Y. Matrix completion with noise. *Proceedings of the IEEE* 98, 6, (2010), 925–936.

[13] Clausen, T., Herberg, U., and Philipp, M. A critical evaluation of the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL). In *IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob).* IEEE, Washington D.C., 2011, 365–372.

[14] Clausen, T. et al., Observations of RPL: IPv6 protocol for low power and lossy networks. IETF Draft. 2014. http://tools.ietf.org/html/draft-clausen-lln-rpl-experiences-08

[15] Reed M. G., Sylverson P. F., and Goldschlag D. M. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications* 16, 4 (1998), 482–494

[16] Dreslinski, R. G., Wieckowski, M., Blaauw, D., Sylvester, D. and Mudge, T. Near-threshold computing: Reclaiming Moore's Law through energy efficient integrated circuits. *Proceedings of the IEEE* 98, 2 (Feb. 2010), 253–266.

[17] Jain, S. et al. A 280mV-to-1.2V wide-operating-range IA-32 processor in 32nm CMOS. In *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC).* IEEE, Washington D.C., 2012, 66–68.

## About the Author

Lorenzo Di Gregorio joined Intel Mobile Communications in 2012 as principal engineer for mobile SoC architectures, after 14 years of experience in technical positions as designer, project leader, architect and scientific coordinator for communication systems with Siemens, Infineon and Lantiq. With the planned consolidation of Intel Mobile Communications in May 2015, he has joined Intel Corporation as a senior member of its technical staff.

**DOI:** 10.1145/2822877