Chapter 7

Energy-Efficient Communication in Wireless Networks

David Boyle, Roman Kolcun and Eric Yeatman

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/65986

Abstract

This chapter describes the evolution of, and state of the art in, energy-efficient techniques for wirelessly communicating networks of embedded computers, such as those found in wireless sensor network (WSN), Internet of Things (IoT) and cyberphysical systems (CPS) applications. Specifically, emphasis is placed on energy efficiency as critical to ensuring the feasibility of long lifetime, low-maintenance and increasingly autonomous monitoring and control scenarios. A comprehensive summary of link layer and routing protocols for a variety of traffic patterns is discussed, in addition to their combination and evaluation as full protocol stacks.

Keywords: Internet of Things, sensor networks, energy efficiency, communications protocols

1. Introduction

The promise of the 'fourth industrial revolution' relies on the development and integration of the so-called Internet of Things, cyberphysical systems and associated services and process improvements. The basis of the promise is the ability to instrument, connect, automate and remotely manage the majority of industrial systems and processes. The underlying assumption is that cheap, wirelessly communicating sensors and actuators can contribute to providing this capability, either autonomously or as part of a decision support system with a human in the loop.

In many cases, it is assumed that monitoring and control applications will require the use of devices that operate without persistent energy availability. Where no mains power is available, energy becomes a major constraint for applications expected to operate for increasingly long time periods. These periods are determined by the feasibility of maintenance of devices,

with respect to both practicality and cost. Therefore, a major recent theme of research has been to attempt to achieve *energy neutrality*. This has been approached from many different angles, including novel ultra-low power receiver circuits (wake-up radio, WuR) [1, 2], energy harvesting and hybrid-storage (battery-supercapacitor) systems [3], compressive and predictive sensing [4–6], and traditionally, energy-efficient communications protocols [7].

Achieving energy neutral operation requires a comprehensive understanding of the application at design time and is seen as a 'holy grail' for networked embedded computing devices. However, applications tend to be characterised by heterogeneous performance requirements [8], deployment environments and criticalities. Therefore, there are few, if any, 'one size fits all' solutions. If one assumes that sensors and actuators are low cost with respect to energy in terms of sampling and information processing (which is not always true in the case of industrial applications [9]), communication is widely accepted to be the primary consumer of energy [10]. This energy consumption occurs when the radio transceiver is in an active state to send, receive and/or route packets. Assuming a worst-case scenario whereby the radio transceiver is always active (a state which tends to require tens of milliwatts of power [11]), the obvious way to reduce energy consumption is to place the device in the lowest power mode available for as long as possible—a technique known as duty cycling [12], which can be applied to the radio transceiver (radio duty cycling, RDC) and the other components of the device. This is equally true for recent system on chip (SoC) solutions that integrate transceiver and microcontroller circuitry on the same chip.

However, the device must also be able to participate in a network, which necessitates the implementation of a communications protocol stack (Section 1.1.1). This is a long-standing research area in the wireless sensor networks (WSN) community, which in the past 10–15 years has developed numerous energy-efficient communications mechanisms that operate at and across various layers. The remainder of this chapter is dedicated to exploring the evolution of energy-efficient communications primitives, explaining the inherent trade-offs in the design space and providing a comprehensive description of the state of the art. The emphasis is placed on link and routing layers.

1.1. Wireless communication

There are several well-known wireless communications technologies in popular use. These range from cellular, now on the verge of the 5th Generation (5G) [13], to WiFi (IEEE 802.11) and Bluetooth Low Energy (BLE) [14], among the most widespread[1]. Recent WiFi and BLE chip-sets are significantly lower power than their predecessors and are increasingly competitive for certain classes of energy-efficient 'IoT'-type applications, particularly in the consumer electronics market.

This chapter focuses on wireless communications protocols suitable for use with RF transceivers and SoCs typically considered ideal for low-power WSN-type applications, such as the (now obsolete) TI CC2420 [16], CC2538 (used in the recent OpenMote devices) [17], and the NXP JN5168. These are compliant with the IEEE 802.15.4 standard for low-rate wireless

[1]Full coverage of wireless communication fundamentals is available in [15].

http://dx.doi.org/10.5772/65986

personal area networks (LR-WPAN) [18], which is responsible for underlying much of the research in this area and has found its way into numerous industrial standards and specifications including ZigBee [19], WirelessHART [20] and ISA100.11a [21].

1.1.1. Communications protocol suites

A communications protocol suite specifies how data should be formatted (i.e. in packets), transmitted and received (channel access), and routed in a network[2]. Layers of abstraction are used to describe the various networking functions involved and vary somewhat depending upon the model of abstraction. For example, the OSI model specifies seven layers, whereas the TCP/IP model (Internet Protocol suite) specifies four. This is a particularly relevant issue in WSN design, as it is well known that efficiencies are best achieved by co-designing the layers [22], but they are often designed independently (e.g. RPL) [23] to enable interoperability. The most recent IETF stack proposed by the 6TiSCH working group, which is particularly suited to IoT, is shown in **Figure 1** and loosely maintains the TCP/IP model. In this case, IEEE 802.15.4 time synchronised channel hopping (TSCH) is used at the 'link layer' (which includes medium access control), 6top is an interface layer to the 6LoWPAN adaptation and compression layer (used to reduce the size of the IPv6 header such that it comfortably fits in the 127-bytes available in an IEEE 802.15.4 physical layer packet), and RPL (Section 3.2.5) at the Internet/routing layer. UDP is used at the transport layer to mitigate the overheads associated with TCP (i.e. end-to-end handshaking), and the Constrained Application Protocol (CoAP) [24] handles application layer functionality.

While the higher layers of the stack benefit from having an understanding of, and in many instances real-time information from, the lower layers, the most critical from an energy efficiency

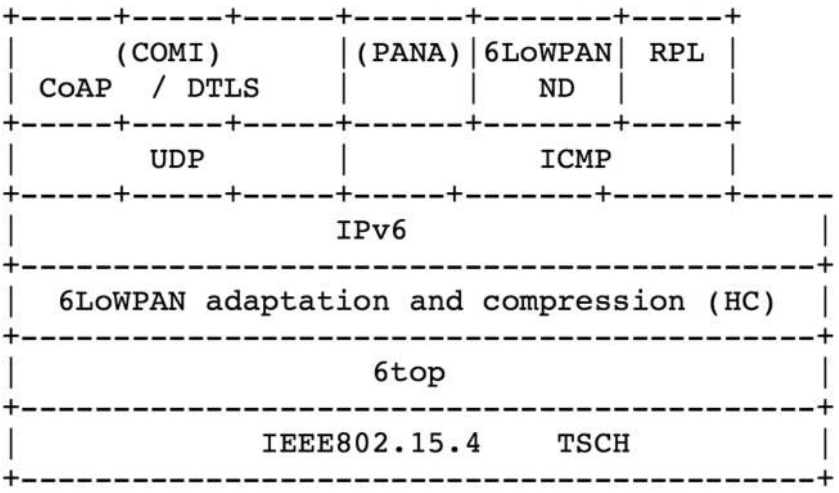

Figure 1. The 6TiSCH Protocol Stack [84].

[2]The practical implementation of a protocol suite is typically referred to as a 'stack'.

perspective are the physical layer (i.e. the radio transceiver itself) and the link layer, the latter of which is responsible for medium access control, and therefore, how long the transceiver remains in an active or sleep state. Energy-efficient medium access control (MAC) protocols are discussed in Section 2. However, as discussed in Section 2, the selection of a suitable MAC protocol is heavily dependent upon the application, specifically with regard to the statistical properties of the traffic generated in the network, the network topology and environmental factors. A well-designed application will take each of these factors into account at design time.

1.2. Hardware

The electrical characteristics of the hardware play a significant part in the overall energy efficiency of a networked device and the performance of a network. While a complete analysis of each component is beyond the scope of this chapter, it is worth highlighting where hardware mostly impacts the design of applications and implications for various layers in the protocol stack.

Firstly, selection of the radio itself is a key. From an application standpoint, basic quality of service requirements must be met, primarily bandwidth—otherwise the application is probably infeasible (without resorting to trickery in software, such as compressive or predictive sensing, and assuming this is satisfactory for an end-user). The worst-case scenario from an energy perspective is that the radio must be on 100% of the time. Therefore, once a wireless technology is selected (e.g. take IEEE 802.15.4), the designer sets about choosing a chip. The majority are reasonably similar in terms of their electrical characteristics, irrespective of the manufacturer. This makes life easier *but* it is worth considering the features of the chip selected. For example, on-board hardware acceleration of security functions can greatly reduce the overheads associated with implementation in software. Most contemporary RFICs compliant with IEEE 802.15.4 are in the tens of milliwatts range in active modes and drop to the microwatt range in low power modes, making them good candidates for long-term low-power applications. They require additional RF front-end design, including antenna and matching networks, plus an oscillator circuit to drive the clock. The latter requires the selection of inductors and capacitors, which are variable in their characteristics.

Important performance characteristics are often affected by environmental conditions such as temperature and humidity. In the case of a wireless node, these have effects internally (i.e. on each device) and externally (i.e. the effects on the wireless medium), which both impact on the performance of a network. In the case of the device, temperature effects and component selection have a significant effect on relative clock drift, which must be taken into account when tuning and learning protocol parameters like guard times and phase offsets, respectively (Section 2). Understanding clock drift is essential to tightly configure networking parameters, such as guard times to ensure accurate synchronisation, and has been studied in [25] where the authors investigate the effects of environmental temperature on clock drift and propose strategies to help designers choose optimal resynchronisation periods for given accuracies, and in [26] where the authors study the impact of oscillator drift on end-to-end latency over multiple hops using varying capacitor accuracies and show how to determine optimal parameters to minimise energy consumption in duty-cycled wireless sensor networks using

http://dx.doi.org/10.5772/65986

low power medium access control techniques. It is also worth noting that temperature influences battery performance, particularly as temperatures reduce, where capacity is degraded and voltage is known to reduce. This is a relatively understudied area in terms of IoT/WSN technologies, but is likely to be very important where devices are deployed outdoors in cold environments.

2. Energy-efficient medium access control

A large body of research exists concerning MAC protocols for WSNs. Notable examples of energy-efficient implementations include A-MAC [27], BoX-MAC [28], HuiMAC [29], SCP-MAC [10], ContikiMAC [30] and WiseMAC [31]. These MAC protocols are based on globally asynchronous, radio duty-cycled (RDC) approaches, where the objective is to minimise the active time of the RF transceiver. Typically, trade-offs are assumed to be inherent in the design of these protocols, and the Pareto-optimal solution is sought when considering energy efficiency and application level or performance requirements, such as throughput, reliability and latency. In [32], the authors consider low data-rate applications and attempt a tractable analytical approach to modelling latency and energy efficiency as functions of protocol parameters including duty cycle, slot duration and total slots, seeking to determine optimal settings for given workloads defined by application-level parameters. They find that WiseMAC best balances energy efficiency and latency based on the scenarios considered, and attribute minimising protocol overhead through local synchronisation (also referred to as *phase lock* optimisation in the literature, and exploited in several subsequent proposed MACs [30, 33]) and random channel access as key to achieving this balance.

These protocols, however, are just the tip of the iceberg (and are heavily oriented towards aggressively duty-cycled, energy-efficient scenarios for low data-rate applications). In [34], Bachir et al. present a comprehensive taxonomy of MAC protocols according to the various techniques being used, classifying them according to the challenge they address. They describe the importance of understanding the statistical properties of the network traffic when selecting and tuning a MAC protocol, which is argued to be a more useful approach for the application developer. The authors classify the traditional 'MAC families' as *Reservation-Based Protocols* and *Contention-Based Protocols* at the highest level, where the former is synonymous with scheduled approaches such as Time Division Multiple Access (TDMA), and the latter with simpler, popular approaches such as ALOHA and Carrier Sense Multiple Access (CSMA). They proceed to explore the functions relative to: high—*scheduled* protocols for high-rate applications such as multimedia; medium—protocols with *common active periods* for medium rate applications such as those found in many industrial applications; and low—*preamble sampling* for low-rate applications with rare event-reporting, such as long-term monitoring or metering, and finally consider *hybrid* protocols for time-varying application-level functionality. **Table 1** borrows the structure and updates the taxonomy of that presented in [34]. Specifically, A-MAC, Reins-MAC and IEEE 802.15.4e, which leverages TSMP and is closely linked with ongoing standardisation initiatives, are notable protocols published since Bachir et al. presented their study.

Functionalities	Protocols
Scheduled	TSMP, IEEE 802.15.4(e) [35], Arisha, PEDAMACS, BitMAC, G-MAC, SMACS, TRAMA, FLAMA, Î¼MAC, EMACs, PMAC, PACT, BMA, MMAC, FlexiMAC, PMAC, O-MAC, PicoRadio, Wavenis, f-MAC, Multichannel LMAC, MMSN, Y-MAC, Practical Multichannel MAC, LMAC, AI-LMAC, SS-TDMA, RMAC, Reins-MAC [36]
Common active period	SMAC, TMAC, E2MAC, SWMAC, Adaptive Listening, nanoMAC, DSMAC, FPA, DMAC, Q-MAC, MSMAC, GSA, RL-MAC, U-MAC, RMAC, E2RMAC
Preamble sampling	Preamble-Sampling ALOHA, Preamble-Sampling CSMA, Cycled Receiver, LPL, Channel Polling, BMAC, EA-ALPL, CSMA-MPS, TICER, WOR, X-MAC, MH-MAC, DPS-MAC, CMAC, GeRAF, 1-hopMAC, RICER, WiseMAC, RATE EST, SP, SyncWUF, STEM, MFP, 1-hopMAC, SpeckMAC-D, MX-MAC, IX-MAC [33], ContikiMAC [30], LPP [37], RI-MAC [38], A-MAC [27], Flip-MAC [39]
Hybrid	IEEE 802.15.4, ZMAC, Funneling MAC, MH-MAC, SCP, Crankshaft

Table 1. Summary of MAC protocols by functionality, updates [34].

Another interesting development relates to the exploitation of constructive interference, which is contrary to CSMA mechanisms and seeks to benefit from concurrent transmissions. Flip-MAC [39] and Glossy (not strictly a MAC protocol) [40], for example, make good use of this with regard to packet acknowledgements and efficient network flooding, respectively.

2.1. Standards and evolution

The IEEE 802.15.4 standard is one of the most important standards in WSN/IoT. First published in 2003, it specified the physical and medium access control mechanisms for low-rate wireless personal area networks and became part of the industrial standards and specifications listed in Section 1. The medium access control layer specified in the standard is effectively a hybrid construct that allows the use of slotted or un-slotted CSMA-CA with optional guaranteed timeslots and packet delivery. It was designed to accommodate a range of topologies, including star and peer-to-peer, specifying device classes that allowed for reduced protocol complexity for 'reduced' function devices (RFD) opposed to 'full' function device (FFD) capable of communication with any device in the network. While this allows for relatively low duty cycles to be achieved, there is an inherent trade-off between energy saving and latency and bandwidth. This was studied immediately, and simulations were used to illustrate the trade-offs related to using the various modes, for example, in [41].

De facto standards simultaneously emerged in the research community where a number of competing objectives made it difficult to build practical implementations for the hardware available at the time. Factors such as ultra-low power operation, collision avoidance, efficient channel utilisation, robustness to time-varying channel and networking conditions, scalability, and efficiency in terms of implementation and memory requirements needed to be addressed by design. B-MAC (short for Berkeley) was one of the earliest protocols built into an open source 'operating system' for WSNs, namely TinyOS [42]. B-MAC (2004) is a carrier sense protocol that improved upon the performance of S-MAC (one of the first to use RTS/CTS in sensor networking) under certain conditions, making use of clear channel

http://dx.doi.org/10.5772/65986

assessments, acknowledgements, back-offs and low power listening (LPL). This was later improved upon by merging features with X-MAC (a link-layer only approach, 2006), resulting in the BoX-MAC (2008) protocols which eventually shipped with TinyOS [28]. BoX-MAC demonstrates cross-layer design and how to achieve significant energy efficiencies and throughput improvements for reasonable workloads.

Within the Contiki community, a number of the same protocols were implemented and extended into standard link layers and radio duty cycling mechanisms in a cross layer fashion. Features from BoX-MAC and X-MAC, such as periodic wake-ups and strobes, and WiseMAC, specifically the phase lock optimisation, were integrated to form the ContikiMAC protocol, which has shipped as a *de facto* standard link layer since around 2011.

TSMP is one of the most notable TDMA MAC layers developed in the last decade [43]. It uses synchronisation between devices to communicate in scheduled timeslots (allowing low-energy radio management) and operates reliably in noisy environments by using channel hopping to avoid interference, with different time-slotted packets sent on different channels depending on the time of the transmission. Therefore, the approach is suitable for applications that require relatively low-power and high-reliability performance, characteristic of industrial automation scenarios, and has thus been included in the WirelessHART and ISA standards (Section 1.1). More recently, it has become fundamental to the development of the IEEE 802.15.4e amendment to the standard, which is actively being included as a core technology in the IETF standardisation effort (e.g. **Figure 1**).

2.2. Design trade-offs

It is important to understand the trade-offs inherent in the link layer design choice. From an energy efficiency perspective, selection of a low-power transceiver is critical, but selection of the appropriate link layer will depend on the application's performance requirements. All of the aforementioned protocols are *relatively* energy efficient. Therefore, depending on the reliability, throughput and latency required, certain protocols perform better than others. For critical applications, it is usual that energy efficiency is not prioritised as highly as reliability and low latency. Therefore, in a high-criticality application, TSMP would be preferable to an asynchronous protocol such as ContikiMAC.

Table 1 shows that the area of MAC design for wireless sensor network applications has been comprehensively researched. As a result, many recently proposed protocols integrate and build upon ideas previously presented. The parameters of importance are now reasonably well known. CSMA-CA protocols with arguably the best performance typically operate using request-to-send/clear-to-send (RTS/CTS) signalling, where networked devices periodically listen to the channel to determine whether any neighbours want to send packets, or alternatively begin to strobe RTS packets and wait for a CTS message from a listening neighbour if the device wants to transmit. The choice of what to send in the RTS packet is an interesting one, with many designers proposing to effectively send an entire data packet as the RTS strobe, such as in ContikiMAC, whereupon the receiving node sends an ACK having received the data payload. However, this can be suboptimal considering variable length data packets. It is argued in [33] that the use of a fixed-length RTS packet can bring efficiencies when

considering its relationship to strict timing parameters. Addressing information (to enable unicast, broadcast and/or multicast communication) is a key inclusion for such a packet.

Timing parameters for such protocols are bounded by performance characteristics of the RFIC (such as turnaround times between TX and RX modes) and the theoretical minimum times required to transmit, receive and act upon packets. Perhaps the most critical parameters are the *receive check* (RC) and *receive check intervals* (RCI). It is also worth noting that receive checks can be performed at both the physical (L1) and link layers (L2), which enforces a trade-off between absolute energy efficiency and reliability. Where L1 only methods are used, the RC operation is cheaper, but net energy efficiency may be worse if devices that are not being addressed stay in RX mode and parse packets not specifically intended for them.

The RC is where a node samples the wireless medium for energy to determine whether or not there is an incoming packet. An optimal duration for the RC is tightly coupled with the time taken to listen for a CTS strobe during a wake-up stream. Minimising the RC is a key to optimising energy efficiency, but this period is also closely related to reliability performance. Theoretical minimum values for length of the RC and the CTS listen period are calculated using the RXTX turnaround times for the transceiver, its bit rate and processing time.

The RCI is the interval between performing RCs, typically measured in Hz. It is possible to determine an optimal RCI with regard to energy efficiency, but this is detrimental to latency (where a latency is fundamentally lower bounded by the RCI and the number of hops over which a packet must travel to reach its destination). RCIs are often lengthened to reduce their impact on the quiescent power consumption of a device. Where large RCIs are used, for example, >0.5 Hz, it becomes more important to implement phase or offset learning (resulting from positive or negative relative clock drift) to ensure that energy consumption is minimised when beginning the next RTS/CTS wake-up stream.

Transmitting nodes may or may not, depending on the application, have an enforced periodic data reporting interval. This is often referred to as the inter-packet interval (IPI) in the literature and broadly reflects the frequency with which sensor data are transmitted from each device in the network. It is worth remembering that the IPI is not necessarily the same as the sampling rate of the sensor, where in many cases an aggregate or average values may be returned periodically (i.e. at the IPI rate of the node).

A factor less studied in the literature is the implementation of the CSMA algorithm itself. In [33], the authors demonstrate that a quantised approach to the implementation of back-offs performs better with regard to energy efficiency and reliability than randomised approaches.

So called receiver-initiated protocols have more recently been studied, whereby when a node wants to transmit a packet it waits for a neighbour (i.e. a potential receiver) to send a probe. Upon hearing the probe, the sending node transmits its packet. Examples of this approach are Low-Power Probing (LPP) as introduced in Koala [37], RI-MAC [38] and A-MAC [27].

The physical and link layers determine the lower bound for energy efficiency with respect to point-to-point communication. When a network extends beyond a one-hop (or star) topology, devices in the network must implement some routing functionality (usually referred to as

http://dx.doi.org/10.5772/65986

Layer 3 or the NWK layer), to ensure that packets arrive at the intended destination, which is explored in the next section.

3. Energy-efficient networking, data collection and dissemination

Routing algorithms are essential for every WSN as they define how packets flow within the network. Algorithms differ from each other based on their capabilities, effectiveness, amount of memory required to store the routing state, agility and energy awareness.

Routing protocols for WSNs can be categorised into three main groups depending on the functionality they provide: (*i*) protocols that route data only towards one or more predefined *base-station* or *sink* nodes, referred to as *data collection* protocols or *many-to-one* patterns of communication, (*ii*) protocols that allow *peer-to-peer* or *any-to-any* patterns of communication between nodes in the network and (*iii*) protocols allowing one node to *disseminate* a message to all or a subset of the nodes in the network, also referred to a *one-to-many* pattern of communication. Protocols which belong to the first group are usually based on building *routing trees* which are rooted at one (or more) sink node(s). Every node in the network forwards all data towards a sink via a selected parent. Protocols from the second group support peer-to-peer communication, that is, any node in the network can send a message to any other node. These protocols either exploit some kind of routing table, learned beforehand, to forward data towards the destination or they first need to find the destination node and establish a connection. Protocols from the last group are usually based on gossiping or flooding the whole network.

3.1. Data collection

Typically, WSNs have been deployed to collect data about certain phenomena in predefined geographic areas (sensing field). Nodes in such a network sample a sensor at a predefined rate and report the sensed data towards a sink node. The network may contain several base-stations, in which case a node typically forwards data towards the closest base-station only.

Two of the most common routing protocols are *Collection Tree Protocol* (CTP) [44] and *Routing Protocol for Low Power and Lossy Network* (RPL)[3] [23]. CTP is a standard library of the TinyOS [42] operating system, while the Contiki [46] operating system comes with two standard alternatives: (*i*) Collect, which is part of the basic Rime network stack [47] and is an alternative implementation of CTP for Contiki OS and (*ii*) RPL, which is part of the more complex IPv6 stack. There is also an RPL implementation for TinyOS called TinyRPL. These protocols are based on creating a directed acyclic graph rooted at the base-station. Each node in the network forwards all data towards the root.

A primary challenge in the design of these protocols is defining the metric by which a node chooses its parent, via which the node will forward all packets. Early adopters of this approach

[3]RPL is in fact designed to support any-to-any communication; however, very little work has been done in this regard, with early evaluations suggesting that it is not ready for actuation-type messages [45].

used hop count to the base-station as a metric [48]. Later, CTP used a new metric: Expected Transmission Count (ETX). ETX uses the number of transmissions required to deliver the packet to the destination without error. ETX depends on the quality of the link. Many protocols vary by how the quality of the link is measured and computed (e.g. based on RSSI, or a statistical measure of the number of packets lost, or a function of both). It has been shown that using this metric decreases the network traffic and leads to lower energy consumption.

Energy efficiency can be described as the ratio between the total number of packets received at the destination node (i.e. the base-station, in the case of collection protocols) and the total energy spent by the network to deliver these packets. Due to the overhead of IPv6 protocol, researchers were concerned about the energy efficiency of the RPL protocol. However, it was shown that packet delivery ratio of CTP and RPL is very similar, while the overall energy consumption is only 3% higher for the latter [49]. Similar results were observed in a study focused on interoperability of RPL implementation for Contiki and TinyOS operating systems [50].

A disadvantage of building rigid routing trees is that the nodes along the path towards the base-station have to transfer more data than other nodes, and hence their batteries deplete faster. This especially applies to the nodes closer to the base-station. To tackle this problem, Lindsey et al. proposed *Disjoint Paths* and *Braided Paths* algorithms [51]. Disjoint Paths algorithm constructs a small number of alternative paths from each sensor to the base-station. These paths are sensor-disjoint, that is, paths have no intermediate nodes in common. Braided Paths algorithm creates partially disjoint paths from the primary path, that is, for each node on the path an alternative path is created which does not contain given nodes. Therefore, in the case of a node failure, an alternative path can be used without the need to first find the path.

An alternative approach to creating rigid routing trees is the back-pressure protocol (BCP) [52] presented by Moeller et al. In networks with BCP, the routing decision depends on the size of the packet queue and the packet rate between two nodes. Each node maintains a queue of packets, where a base-station has a queue of zero length. A node forwards a packet to a neighbour only if the neighbour's queue is shorter than the queue of the sending node. The received packet is put on the top of the queue and in the next iteration forwarded to a node with a shorter queue. This can lead to more evenly spread network traffic while exploiting various routes towards the base-station.

An older approach which tries to eliminate rigid routing trees is *hierarchical routing*. This approach breaks the network into clusters, each of which has a *cluster-head*. Nodes send data to these cluster-heads which are then responsible for delivering data to the base-station.

Heinzelman et al. presented *Low-Energy Adaptive Clustering Hierarchy* (*LEACH*) [53], a popular clustering algorithm whose goal is to reduce energy consumption of nodes in a WSN. The operation of the algorithm is split into two phases: (*i*) the *setup phase* during which cluster-heads are elected and nodes choose which cluster they will be part of and (*ii*) the *steady phase* during which sensor nodes transfer data to the cluster-heads which aggregate received data and forward them to the base-station. Cluster-head election is distributed, and nodes do not require any global knowledge of the network. A disadvantage of this algorithm is that the communication between nodes and cluster-heads, as well as the communication

http://dx.doi.org/10.5772/65986

between the cluster-heads and the base-station is single-hop only, which limits the size of the network. Additionally, cluster-head selection does not take into account the residual energy of the node. Being a cluster-head is very energy demanding, as the cluster-head has to be 100% duty-cycled in order to receive messages from all nodes. If a node with small residual energy is elected, it may deplete its battery before it collects all data and forwards them to the base-station.

One of the extensions of LEACH algorithm presented by Lindsey and Raghavendra, *Power-Efficient Gathering in Sensor Information Systems (PEGASIS)* [54], creates chains of sensor nodes where each node aggregates data received from the previous node with its local data. In each iteration, a random node from the chain is chosen to forward aggregated data to the base-station. A disadvantage of PEGASIS is that it assumes that each node has global knowledge of the network layout, particularly the position of the nodes.

Younis and Fahmy presented another LEACH extension called *Hybrid, Energy-Efficient Distributed Clustering (HEED)* [55]. Unlike LEACH, it operates in multi-hop networks and for cluster selection it uses both the residual energy of the node and the node degree or density. Thus, the algorithm may better balance energy consumption amongst the nodes, hence prolonging the lifetime of the network.

Each of these hierarchical routing protocols scored three out of four points for energy efficiency in a large survey on routing protocols [56], meaning that these protocols achieved average packet delivery rates while choosing routes based on the residual energy, thereby prolonging the lifetime of the network.

3.2. Peer-to-peer routing

For networks deployed for the purpose of monitoring and continuous collection of data, peer-to-peer (P2P) communication is often not necessary. Each node only needs to know how to deliver data to one of the base-stations. However, as WSNs become more common and serve a wider range of purposes, communication among nodes in the network will become more important. WSNs are not only used to collect data but also to react to the environment and control it via *actuators*, that is, components of emerging cyberphysical systems. Nodes in the network need to send messages directly to other nodes, while lowering the overall traffic. With actuation networks, this requirement becomes even more important to send an actuation message directly to the actuator. For that purpose, routing protocols which allow P2P communication were developed.

Such P2P protocols may be categorised into five groups, depending on how they locate and forward messages to the communicate with a peer: (*i*) geographic routing, (*ii*) routing based on trees, (*iii*) hierarchical routing, (*iv*) ad hoc shortest path routing and (*v*) routing based on routing tables.

3.2.1. Geographic routing

In geographic routing, each node is not addressed by its ID or IP address but by its geographic location. The routing decision is then based on the position of the node making the

forwarding decision, the position of the destination node and the position of the neighbours of the forwarding node. The neighbour which is closest to the destination node is chosen as the next-hop. As geographic routing heavily relies on the exact geographic position of the nodes, specialised hardware is required (e.g. GPS) and/or a localisation algorithm must be used. However, specialised hardware increases the price of the node, increases the energy requirements and is sometimes not very precise. Similarly, using localisation algorithms tends to lead to additional network traffic and may also be imprecise [57–59].

Among others, Karp and Kung proposed *Greedy Perimeter Stateless Routing (GPSR)* [60], Kuhn et al. proposed *Geometric Ad hoc Routing* [61], and Yu et al. proposed *Geographic and Energy Aware Routing (GEAR)* [62], all of which implement greedy geographic routing. Even though this type of greedy routing works well under ideal conditions, it fails when a *routing void* is encountered. The routing void is a situation when there is no neighbour which is closer to the destination node. In practice, this situation is common when there is either an error in the localisation algorithm or a physical obstacle prevents radio communication between nearby nodes. Additionally, there is currently no practical way of porting geographic routing to the three-dimensional space [63], for example, in a network deployed in a building or a tower.

According to the survey on routing protocols in [56], GEAR outperforms GPSR in terms of the packet delivery; however, it scores only two points out of four for energy efficiency.

3.2.2. Routing trees

In networks where P2P communication is based on *routing trees,* the nodes are organised in one or several trees where each node stores its parent's ID only. The root of a tree is represented by a more powerful node storing connectivity information of the whole network. The packet is first routed to the root of a tree, where the central router computes the shortest path to the destination. The packet is then routed downwards towards the destination via this shortest path. The advantage of this approach is minimal memory requirements on the nodes and simplicity of the routing algorithm. The disadvantage is potentially high *routing stretch*, that is, the ratio between the length of the found path and the optimal one, and the requirement that the central router is aware of the whole network topology. Additionally, top-level nodes may become overloaded by the network traffic, especially in large networks.

To tackle some of the disadvantages mentioned above, several improvements to the routing trees have been introduced. The key improvements are based on storing meta-data on the nodes within the network. Dedicated nodes store meta-data about all nodes in a sub-tree rooted in given node. Then, a node can decide to route a packet down a tree without forwarding it to the root node. In RPL, the base-station holds a routing table for the whole network. However, any node in the network, provided it has enough memory, can store a routing table for a sub-tree rooted in given node. These nodes are referred to as *routing nodes*. If a routing node receives a packet, it first checks its local routing table to see whether it contains a record for the destination. If so, the packet is forwarded directly to the destination. Otherwise, it is forwarded towards the root.

Duquennoy et al. presented an opportunistic version of RPL called *Opportunistic RPL (ORPL)* [64]. Here, each node uses a Bloom filter [65] to store all node IDs from the sub-tree rooted

http://dx.doi.org/10.5772/65986

in a given node. Each node then uses the summary to decide whether the packet should be forwarded up towards the root of the tree or down the sub-tree rooted in a given node. When the packet travels up the tree, it does not necessary follow the spanning tree. Any node which is closer to the root can opportunistically forward the packet. These two improvements can significantly lower the routing stretch. However, the possibility of a false positive in Bloom filters is the main disadvantage of this approach. In this case, a special algorithm is required which can recover from the situation when a packet is routed down the tree based on a false positive. The packet has to be sent to the root of the tree which can then find the correct path to the destination. This leads to unnecessary traffic and large routing stretch.

Mihaylov et al. use a similar approach in their *Innet* algorithm [66], but, to reduce the routing stretch, they build up to three routing trees, each rooted in a different part of the network. Each node stores a summary for each sub-tree rooted in given node. The search for the destination node is performed in all routing trees in parallel. Furthermore, to avoid the problem of false positives in Bloom filters, the packet is always also forwarded up, until it reaches the root of a tree. The packet stores the path it takes until it reaches the destination node. The destination node replies to the source node by reversing this path. As the reply packet travels back to the source node, each node uses several techniques to find a shortcut between the communicating nodes. Innet was designed to support long-term communication, that is, the communicating peers exchange messages for a longer period of time. Therefore, the main goal of the algorithm is to minimise the routing stretch. The higher cost of the search phase is outbalanced by savings that could be achieved during the long-term communication amongst the nodes.

3.2.3. Hierarchical routing

In *hierarchical routing*, each node is a part of multi-level hierarchically organised clusters [63,67]. At the lowest level 0, each node is a member of its own singleton cluster. Then, a neighbourhood of level 0 clusters is organised into level 1 cluster, which in turn are grouped into level 2 cluster, until all nodes are member of one (or very few) big cluster(s). At each level, a node is a member of exactly one cluster.

At the centre of each cluster is a *cluster-head*. At each level i the cluster-head is advertised R_i hops away. The R depends exponentially on the level i. A node can be a member of a level i cluster if it is at most r_i hops away from the cluster-head, where $r_i \leq R_{i-1}$. In practice, usually $R_i = 2^i$ and $r_i = \lfloor R_i / 2 \rfloor$. Each node is addressable by concatenating the cluster-head ID at each level (e.g. $X.Y.Z$, where node's ID is X, Y is a level 1 cluster-head, and Z is a level 2 cluster-head).

The *routing table*, stored at each node, consists of entries for each cluster-head the node received an advertisement from. Remember that each cluster-head is at most R_i hops away at every level. Because the routing table is stored at every node, each node in a network acts as a router. When a node receives a packet, it tries to find the record in the routing table for the cluster-head from the lowest level. For example, if the packet's destination is $X.Y.Z$, the node first tries

to locate a record for X. If it is not found, it tries the same for Y, and Z, respectively. Because Z is the top-level cluster, every node in the network will know a route to it. The packet is routed towards the first found record. Because $r_i \leq R_{i-1}$ it is guaranteed that as the packet is routed towards the level i cluster-head, there will be a node on the path which knows the route towards the level $i-1$ cluster-head. Therefore, the packet will eventually reach the destination node. In practice, hierarchical routing achieves relatively small average routing stretch of 25% [63]; however, additional network traffic is required to keep the routing tables updated.

3.2.4. Ad hoc routing

Unlike other routing algorithms, *ad hoc routing* does not require any global preparation phase during which the network is prepared for P2P communication. However, when a node needs to communicate with a peer, a path between the nodes must be established first. This is usually done by flooding the network with a request [68–70]. The request contains the source node ID, the destination node ID and a path taken by the request so far. Each node, unless the node is the destination that receives the request, adds itself to the path and re-broadcasts the request. Once the destination node receives the request, it replies to the source node by reversing the path of relay nodes. The algorithm leads to discovery of the shortest path between two nodes.

The disadvantage of this approach is a very expensive path discovery. As the whole network is flooded with a request, this results in poor energy efficiency. Even though other approaches like routing via trees also rely on path discovery, the search in those networks is more directed and does not flood the entire network.

3.2.5. Routing based on routing tables

Approaches based on *routing tables* first execute a learning or a bootstrapping phase during which every node learns routes to nodes that are of interest. For every node, two other pieces of information are usually stored: $\langle distance, next\_hop \rangle$. As a distance, usually number of hops is used, but any other additive metric could be used. Once the bootstrapping phase is complete, each node can independently forward the packet using the locally stored routing table.

Routing algorithms like RPL [23] use a subset of nodes to store the routing table and only for nodes that are in the sub-tree rooted in a given node. Other approaches like *Energy Aware Routing* (EAR) [71] use a routing table to store several alternative routes to the base-station so a node can choose alternative routes in order to better distribute network load. Kolcun et al. proposed Dragon [72], where every node in the network stores a path to every other node in the network. The platform also includes algorithms for quick update of the routing table in the case of a node failure while keeping the network overhead low.

3.3. Dissemination protocols

The purpose of *dissemination* protocols is to deliver information to either all or a subset of the nodes in the network. This type of communication is often referred to as a *one-to-many*

http://dx.doi.org/10.5772/65986

communication pattern. Often the initiator of the communication is the base-station when it wants to, for example, update the reporting interval. Because most of the collection protocols do not support one-to-one communication, dissemination protocols are also often used if a node wants to deliver a message to one particular node only, even at the expense of flooding the whole network.

Two of the basic techniques used are flooding and gossiping [73]. While in the case of flooding each node just re-broadcasts every message it receives, in the case of gossiping, a node upon receiving a message randomly chooses a neighbour to which it forwards the message. The disadvantage of flooding is the implosion and duplication of messages, while the disadvantage of gossiping is a possible large delay in propagation of the message.

Heinzelman et al. proposed a family of *Sensor Protocols for Information via Negotiation (SPIN)* [74]. A node running SPIN, upon receiving a new message, broadcasts an *advertising* message containing meta-data of the received message. Nodes which have not previously received the message reply with a *request* message. The node then broadcasts the *data* message to all the nodes that requested the data. SPIN can significantly decrease the network traffic by eliminating redundant data. However, SPIN cannot guarantee delivery of the message to all the intended recipients due to message loss and unreliable communication.

Braginsky and Estrin proposed *Rumor Routing (RR)* [75] where nodes that wish to distribute information about a certain event generate special long-lived packets called *agents* which are then gossiped through the network. Every node which receives an agent packet creates a record in its *event table*. Subsequently, if a node is interested in a given event, the node generates a query and uses the gossip protocol to propagate it through the network. Any node that has a record in its event table can forward the query to the node that holds the information about the event. Rumor Routing can significantly decrease the network traffic in cases where the number of events is small and the number of queries is large.

Both SPIN and RR scored the lowest mark for energy efficiency—one out of four—in the survey on routing protocols [56], due to their low packet delivery rate and not being energy aware when disseminating the data.

Levis et al. introduced probably the most popular dissemination protocol called *Trickle* [76]. It uses a 'polite gossip' policy where a node periodically broadcasts the message it wants to propagate, but stays quiet if it hears a message from a neighbour which is identical to its own. Because the time interval after the same message is re-broadcast increases exponentially, each node hears only a small trickle of packets. The algorithm can achieve global dissemination of the message at a very low maintenance cost. Trickle facilitates the DRIP, DIP and DHV dissemination libraries available in TinyOS and is a key component of the CTP and RPL protocols.

Kolcun et al. introduced the *Static Attribute Propagation* algorithm as a part of their Dragon platform [72]. The algorithm eliminates unnecessary re-broadcasts of the message by overhearing the messages of its neighbours. It relies on the knowledge of a list of common neighbours at every node. A node upon receiving a message sets up a random timer. While waiting, the node overhears all its neighbours broadcasting the message. Upon expiration of the random

Functionalities	Protocols
Collection	CTP [44], RPL [23], BCP [52], LEACH [53], HEED [55], PEGASIS [54], LWB [77]
Dissemination	SPIN [74], RR [75], Static Attribute Propagation [72], RPL [23], LWB [77], Trickle [76]
Peer-to-peer	GEAR [62], GPSR [60], Geometric Ad hoc Routing [61], ORPL [64], Innet [66], AODV-BR [69], Dragon [72], LWB [77], Chaos [78]

Table 2. Summary of networking protocols by functionality.

timeout, the node checks whether the set of neighbours which broadcast the message cover all the nodes neighbours. If so, the message is discarded, and otherwise, the node broadcasts the message. This approach can significantly decrease the network traffic, especially in dense networks where the number of common neighbours is high.

3.3.1. Special case: low-power wireless bus

The low-power wireless bus (LWB) is a special case that considers multiple traffic patterns, including one-to-many, many-to-one and many-to-many traffic by exploiting the aforementioned Glossy mechanism to facilitate efficient and reliable floods [77]. It uses time synchronisation to manage access to the bus, where a global communication schedule is maintained (computed online based on immediate traffic) and flooded periodically to nodes (thus avoiding relative clock drift). The authors demonstrate that the LWB is comparable to or outperforms a number of state-of-the-art stacks with regard to many-to-one (i.e. collection) traffic, adapts well to varying traffic volumes, significantly outperforms contemporary approaches in terms of many-to-many, is robust to inference and intermittent node participation, and supports mobile nodes as source or sink network devices (**Table 2**).

4. Full stack implementations

4.1. IPv6 over LR-WPAN

With arrival of IPv6, researchers set about implementing it for sensor networks. This was met with several challenges. Because the main usage domain of IPv6 is Ethernet, to cope with increased Internet traffic, the maximum transmission unit (MTU) was increased from 576 to 1280 bytes, when compared to IPv4. IPv6 addresses are 128-bit long, and the standard IPv6 header size is 40 bytes. This is in strict contrast with the IEEE 802.15.4 standard whose throughput is limited to 250 kbps and the length of the frame to 127 bytes. The standard supports two addresses: short 16-bit and EUI-64 extended addresses. With link headers included, the effective size of the payload could be as small as 81 bytes, which make IPv6 headers seem too large.

In 2007, Mulligan and an Internet Engineering Task Force (IETF) working group published a proposal on how to transfer IPv6 packets in low-rate wireless personal area networks. A new protocol called 6LoWPAN [79] was introduced. The aim of the working group was to define a stateless header compression that would decrease the header size so it can be used with the IEEE 802.15.4 standard. The reduction was achieved by introducing four basic header types:

http://dx.doi.org/10.5772/65986

(*i*) Dispatch Header, (*ii*) Mesh Header, (*iii*) Fragmentation Header and (*iv*) HC1 Header (IPv6 Header Compression Header). 6LoWPAN implements variable-sized headers, where the header size varies from 4 to 13 bytes, depending on what kind of communication is required. The protocol is also prepared to support new types of headers in the future.

When an address needs to be included in the header, 6LoWPAN supports either 16-bit short addresses or full 64-bit addresses. These addresses are then translated to full 128-bit IPv6 addresses by a border router. The border router is a router that enables communication between a WSN and the Internet.

If a node needs to send a packet whose size is larger than the size of the payload of 802.15.4 frame (107 bytes), 6LoWPAN defines a fragmentation header which allows the node to split the original datagram into several packets. The header includes the size of the original datagram as well as the ordering number. Fragmentation is also necessary as the specification of IPv6 requires support of a minimum MTU of 1280 bytes.

6LoWPAN supports two types of routing: (*i*) mesh-under and (*ii*) route-over. In the mesh-under approach, routing is done by the link layer (layer two) using IEEE 802.15.4 frame or 6LoWPAN header. To send a packet to a destination node, EUI 64-bit or 16-bit short addresses are used to forward the packet to the next-hop neighbour, preferably closer to the destination. To complete a single IP hop multiple link layer hops may be required. If an IP packet is fragmented into several fragments, these fragments may travel over different paths. The packet is reassembled at the final destination only. The advantage of mesh-under is that the forwarding nodes do not need to reassemble the whole packet to make the routing decision which lowers the memory requirements. Additionally, because the packets may travel via different paths, mesh-under can increase throughput and lower the congestion of the network. On the other hand, if the destination node is missing, a fragment of the IP packet, the whole packet (i.e. all fragments) needs to be resent.

In the route-over approach, the routing decision is done on the network layer (layer 3) and each node acts as an IP router. Each link/hop is considered to be an IP hop too. If a packet is fragmented, then all fragments are first reassembled on the next hop neighbour and the packet is passed to the network layer. The network layer decides whether the packet should be processed on the node or forwarded to a neighbour. To make this decision, the node has to either store the routing table which maps the destination address to the next-hop address or the packet itself has to contain this information. In route-over approach, each node must have enough memory to reconstruct the packet, and all fragments are routed via one path only. On the other hand, if a fragment is lost during the transmission, the whole IP packet must be resent over one link layer hop only.

Recalling **Figure 1**, a full stack implementation requires some higher and lower layer primitives. At the application layer, the IETF has worked to standardise the so-called Constrained Application Protocol (CoAP), which is essentially a RESTful (Representational State Transfer) protocol that uses a small subset of HTTP commands, and more recently TSCH at the link layer, detailed earlier. To-date, there are no comprehensive evaluations of the energy performance of the entire stack. However, there are several which evaluate snapshots of the

stack, or subsets thereof (e.g. by layer), such as in the case of CoAP in [80], where the authors show that CoAP is efficient when implemented over RPL, 6LoWPAN and ContikiMAC (and reiterate that the key efficiencies are to be gained at the link layer), and 6TiSCH [81], where a realistic energy model presented for TSCH demonstrated that under certain conditions, sub-1% duty cycles are demonstrable for real and simulated networks under reasonable traffic loads.

4.2. Composable stacks

Many of the protocols described thus far have open source, modular implementations available in the libraries of the various operating systems. Therefore, they can easily be composed to suit an intended application scenario. We know that performance and appropriate selection depend on application level requirements and statistical properties of the network traffic generated by that application. Therefore, there are very few studies that explore in-depth full-stack implementations on per-application bases. However, there are a number well-documented implementations that comparatively evaluate performance such as in [77], where the authors compare the performance of LWB against Dozer (a highly efficient TDMA-based data collection protocol) [82], CTP+A-MAC and CTP+LPL, under a variety of conditions. CTP+A-MAC, LWB and more recently Chaos [78] are representative state-of-the-art stacks from the research community that rival the standards-based stacks which tend to adopt something very similar to the 6TiSCH approach (**Figure 1**), for example. While many of the protocols described so far have been implemented in the longer-standing operating systems developed in the research community, a number of more recent such operating systems have emerged, such as OpenWSN—https://openwsn.atlassian.net/wiki/, and RIOT—https://www.riot-os.org/#features, which provide implementations of the standardised stack for a variety of recent hardware development platforms.

4.3. Energy analysis

One of the most comprehensive evaluations of a relatively complete stack is presented in [44], where CTP is run on a number of heterogeneous test-beds, over a number of link layers, for a variety of inter-packet intervals (typically determined by the application scenario) and at a variety of radio frequencies on various channels. While the evaluation shows that the combination of beaconing and data path validation used in its design is robust over a variety of physical and link layers, the performance characteristics still do not quite meet those needed for ultra-long-lived, large (i.e. extremely dense) and highly reliable applications. The authors also leave open the question of whether these methods are suitable for distance vector algorithms synonymous with *ad hoc* networking.

Generally speaking, it is extremely difficult to validate or comparatively evaluate the energy performance of a protocol stack relative to another. The use of simulators and non-standard test facilities (e.g. community-known test-beds set-up with arbitrary configurations) contribute to this problem. This could be mitigated against by having a set of standard simulation and test-bed configurations against which to benchmark protocols at each layer, and in combination. This has been recently alluded to in the literature in [83], wherein the authors conclude that there is insufficient knowledge available for a majority of the community when

http://dx.doi.org/10.5772/65986

it comes to trialling experiments on real-world facilities such that they can be trustworthy, reproducible and thus independently verifiable.

5. Conclusion and future directions

Practical, energy-efficient wireless communications protocols have been comprehensively studied and documented in the literature in the preceding decades. We have presented a comprehensive summary review of the state-of-the-art concerning link and routing layer technologies developed during this period suitable for constrained wireless sensor network, IoT and CPS applications.

A great deal is known about their limitations and the trade-offs inherent in their selection and implementation. It is arguable that fundamental performance limits have been reached in the design of link layer technologies for contemporary radio transceivers. For this reason, and understanding the time-varying nature of the wireless medium, routing protocols are being developed by the standards bodies that take into account lower layer parameters in their design (e.g. IEEE 802.15.5). These may include link quality, residual energy and dynamic energy availability in the case of energy harvesting devices.

Devices are incrementally more efficient, and their inter-networking based on link and routing layer technologies is maturing to the point where protocols can confidently be selected where certain performance requirements must be satisfied. We tend to readily trade energy efficiency against reliability and determinism for industrial and high-criticality applications. This is problematic, however, because many potential application scenarios are dismissed as economically infeasible due to high network maintenance costs. Simultaneously, as devices and protocols maximise efficiency, the gap with feasible energy harvesting from devices' ambient environments is reducing. Coupled with other techniques and technologies, like compressive and predictive sensing, ultra-low power wake-up radio circuits and so on, there is an emergent design space—where applications can be holistically co-designed with regard to energy. It is almost certain that such approaches will be investigated in the short to medium term, which will result in the economic feasibility of a range of new connected monitoring and control applications.

Author details

David Boyle*, Roman Kolcun and Eric Yeatman

*Address all correspondence to: david.boyle@imperial.ac.uk

Department of Electrical and Electronic Engineering, Imperial College London, England

References

[1] L. Gu and J.A. Stankovic. Radio-triggered wake-up for wireless sensor networks. *Real-Time Systems*, 29(2–3):157–182, 2005.

[2] S.J. Marinkovic and E.M. Popovici. Nano-power wireless wake-up receiver with serial peripheral interface. *IEEE Journal on Selected Areas in Communications*, 29(8):1641–1647, 2011.

[3] M. Magno, D. Boyle, D. Brunelli, B. O'Flynn, E. Popovici, and L. Benini. Extended wireless monitoring through intelligent hybrid energy supply. *IEEE Transactions on Industrial Electronics*, 61(4):1871–1881, April 2014.

[4] R.G. Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, 24(4), 118–121 2007.

[5] R.G. Baraniuk, V. Cevher, M.F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010.

[6] U. Raza, A. Camerra, A.L. Murphy, T. Palpanas, and G.P. Picco. What does model-driven data acquisition really achieve in wireless sensor networks? In *2012 IEEE International Conference on Pervasive Computing and Communications (PerCom),* pp. 85–94. IEEE, 2012.

[7] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *2000 Proceedings of the 33rd annual Hawaii International Conference on System Sciences,* 10 pp. IEEE, 2000.

[8] L. Mottola, G.P. Picco, M. Ceriotti, S. Gunçž, and A.L. Murphy. Not all wireless sensor networks are created equal: A comparative study on tunnels. *ACM Transactions on Sensor Networks (TOSN)*, 7(2):15, 2010.

[9] D. Boyle, B. Srbinovski, E. Popovici, and B. O'Flynn. Energy analysis of industrial sensors in novel wireless shm systems. In *Sensors, 2012 IEEE*, pp. 1–4, Oct 2012.

[10] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1567–1576. IEEE, 2002.

[11] A. Moschitta and I. Neri. Power consumption assessment in wireless sensor networks. *ICT-Energy-Concepts Towards Zero-Power Information and Communication Technology*, 2014.

[12] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 95–107. ACM, 2004.

[13] T.S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G.N. Wong, J.K. Schulz, M. Samimi, and F. Gutierrez. Millimeter wave mobile communications for 5g cellular: It will work! *IEEE Access*, 1:335–349, 2013.

[14] SIG Bluetooth. Bluetooth core specification version 4.0. *Specification of the Bluetooth System*, 2010.

[15] T.S Rappaport et al. *Wireless communications: principles and practice*, vol. 2. Prentice Hall PTR ,New Jersey, 1996.

[16] Texas Instruments. Cc2420: 2.4 ghzieee 802.15. 4/zigbee-ready rf transceiver. *Available at* http://www.ti.com/lit/gpn/cc2420, 53, 2006.

[17] Texas Instruments. Cc2538 powerful wireless microcontroller system-on-chip for 2.4-ghz ieee 802.15. 4, 6lowpan, and zigbee applications. *CC2538 datasheet (April 2015)*, 2015.

[18] IEEE standard for information technology—local and metropolitan area networks—specific requirements—part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpans). *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pp. 1–320, Sept 2006.

[19] ZigBee Alliance. Zigbee specification, 2006.

[20] J. Song, S. Han, A.K. Mok, D. Chen, M. Lucas, and M. Nixon. Wirelesshart: Applying wireless technology in real-time industrial process control. In *Real-Time and Embedded Technology and Applications Symposium, 2008. RTAS'08. IEEE*, pp. 377–386. IEEE, 2008.

[21] S. Petersen and S. Carlsen. Wirelesshart versus isa100. 11a: the format war hits the factory floor. *Industrial Electronics Magazine, IEEE*, 5(4):23–34, 2011.

[22] R. Madan, S. Cui, S. Lall, and N. A. Goldsmith. Cross-layer design for lifetime maximization in interference-limited wireless sensor networks. *IEEE Transactions on Wireless Communications*, 5(11):3142–3152, Nov. 2006.

[23] T. Winter, P. Thubert, T. Clausen, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, and J. Vasseur. Rpl: Ipv6 routing protocol for low power and lossy networks, rfc 6550. *IETF ROLL WG, Tech. Rep*, 2012.

[24] C. Bormann, K. Hartke, and Z. Shelby. The Constrained Application Protocol (CoAP). RFC 7252, October 2015.

[25] T. Schmid, R. Shea, Z. Charbiwala, J. Friedman, M.B. Srivastava, and Y.H. Cho. On the interaction of clocks, power, and synchronization in duty-cycled embedded sensor nodes. *ACM Trans. Sen. Netw.*, 7(3):24:1–24:19, October 2010.

[26] E. O'Connell, B. O'Flynn, and D. Boyle. Clocks, latency and energy efficiency in duty cycled, multi-hop wireless sensor networks. In *Advances in Sensors and Interfaces (IWASI), 2013 5th IEEE International Workshop on*, pp. 199–204, June 2013.

[27] P. Dutta, S. Dawson-Haggerty, Y. Chen, C-J. Mike Liang, and A. Terzis. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pp. 1–14. ACM, 2010.

[28] D. Moss and P. Levis. Box-macs: Exploiting physical and link layer boundaries in low-power networking. *Computer Systems Laboratory Stanford University*, pp. 116–119, 2008.

[29] J.W. Hui and D.E. Culler. Ip is dead, long live ip for wireless sensor networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, pp. 15–28. ACM, 2008.

[30] A. Dunkels. The ContikiMAC radio duty cycling protocol. *Technical Report T2011:13*, Swedish Institute of Computer Science. December 2011.

[31] A. El-Hoiydi and J-D. Decotignie. Wisemac: An ultra low power mac protocol for multi-hop wireless sensor networks. In *Algorithmic Aspects of Wireless Sensor Networks*, pp. 18–31. Springer, 2004.

[32] K. Langendoen and A. Meier. Analyzing mac protocols for low data-rate applications. *ACM Transactions on Sensor Networks (TOSN)*, 7(2):19, 2010.

[33] E. O'Connell, B. O'Flynn, and D. Boyle. Energy & reliability optimal mac for wsns. In *2014 10th International Conference on the Design of Reliable Communication Networks (DRCN),* pp. 1–8, April 2014.

[34] A. Bachir, M. Dohler, T. Watteyne, and K.K. Leung. Mac essentials for wireless sensor networks. *Communications Surveys & Tutorials, IEEE*, 12(2):222–248, 2010.

[35] IEEE standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans) amendment 1: Mac sublayer. *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pp. 1–225, April 2012.

[36] M. Ceriotti and A.L. Murphy. A mac contest between lpl (the champion) and reins-mac (the challenger, an anarchic tdma scheduler providing qos). In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pp. 371–372. ACM, 2010.

[37] R. Musaloiu-E., C.J.M. Liang, and A. Terzis. Koala: Ultra-low power data retrieval in wireless sensor networks. In *2008 IPSN '08. International Conference on Information Processing in Sensor Networks,* pp. 421–432, April 2008.

[38] Y. Sun, O. Gurewitz, and D.B. Johnson. Ri-mac: A receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pp. 1–14, New York, NY, USA, 2008. ACM.

[39] D. Carlson and A. Terzis. Flip-mac: A density-adaptive contention-reduction protocol for efficient any-to-one communication. In *2011 International Conference on Distributed Computing in Sensor Systems and Workshops* (*DCOSS*), pp. 1–8. IEEE, 2011.

[40] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with glossy. In *2011 10th International Conference on Information Processing in Sensor Networks (IPSN),* pp. 73–84. IEEE, 2011.

[41] G. Lu, B. Krishnamachari, and C.S. Raghavendra. Performance evaluation of the IEEE 802.15.4 mac for low-rate low-power wireless networks. In *2004 IEEE International Conference on Performance, Computing, and Communications,* pp. 701–706, 2004.

[42] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. Tinyos: An operating system for sensor networks. In W. Weber, J.M. Rabaey, and E. Aarts, editors, *Ambient Intelligence*, pp. 115–148. Springer, Berlin Heidelberg, 2005.

http://dx.doi.org/10.5772/65986

[43] K. Pister and L. Doherty. Tsmp: Time synchronized mesh protocol. *IASTED Distributed Sensor Networks*, pp. 391–398, 2008.

[44] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pp. 1–14, New York, NY, USA, 2009. ACM.

[45] T. Istomin, C. Kiraly, and G.P. Picco. Is rpl ready for actuation? A comparative evaluation in a smart city scenario. In *Wireless Sensor Networks*, pp. 291–299. Springer, 2015.

[46] A. Dunkels, B. Gronvall, and T. Voigt. Contiki—a lightweight and flexible operating system for tiny networked sensors. In *2004 29th Annual IEEE International Conference on Local Computer Networks*, pp. 455–462, Nov 2004.

[47] A. Dunkels, F. Österlind, and Z. He. An adaptive communication architecture for wireless sensor networks. In *SenSys '07*, pp. 335–349, 2007.

[48] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, pp. 491–502, New York, NY, USA, 2003. ACM.

[49] J.G. Ko, S. Dawson-Haggerty, O. Gnawali, D. Culler, and A. Terzis. Evaluating the performance of rpl and 6lowpan in tinyos. In *Workshop on Extending the Internet to Low Power and Lossy Networks (IPSN)*, vol. 80, pp. 85–90, 2011.

[50] J.G. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, J-P. Vasseur, M. Durvy, A. Terzis, A. Dunkels, and D. Culler. Industry: Beyond interoperability: Pushing the performance of sensor network ip stacks. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, pp. 1–11, New York, NY, USA, 2011. ACM.

[51] S. Lindsey, C.S. Raghavendra, and K.M. Sivalingam. Data gathering in sensor networks using the energy delay metric. In *Proceedings of the 15th International Parallel & Distributed Processing Symposium*, IPDPS '01, pp. 188, Washington, DC, USA, 2001. IEEE Computer Society.

[52] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali. Routing without routes: The backpressure collection protocol. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '10, pp. 279–290, New York, NY, USA, 2010. ACM.

[53] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *2000 Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 10 pp. vol. 2, Jan 2000.

[54] S. Lindsey and C.S. Raghavendra. Pegasis: Power-efficient gathering in sensor information systems. In *Aerospace Conference Proceedings, 2002. IEEE*, vol. 3, pp. 3–1125–3–1130, 2002.

[55] O. Younis and S. Fahmy. Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, pp. 640, March 2004.

[56] A. Murtala Zungeru, L.-M. Ang, and K.P. Seng. Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison. *Journal of Network and Computer Applications*, 35(5):1508–1536, 2012. Service Delivery Management in Broadband Networks.

[57] P. Bahl and V.N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, pp. 775–784 vol. 2, 2000.

[58] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low-cost outdoor localization for very small devices. *Personal Communications, IEEE*, 7(5):28–34, Oct 2000.

[59] L. Doherty, K.S.J. Pister, and L. El Ghaoui. Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, pp. 1655–1663, vol. 3, 2001.

[60] B. Karp and H.T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom'00, pp. 243–254, New York, NY, USA, 2000. ACM.

[61] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: Of theory and practice. In *Proceedings of the Twenty-second Annual Symposium on Principles of Distributed Computing*, PODC '03, pp. 63–72, New York, NY, USA, 2003. ACM.

[62] Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical report, Technical report ucla/csd-tr-01-0023, UCLA Computer Science Department, 2001.

[63] K. Iwanicki and M. van Steen. On hierarchical routing in wireless sensor networks. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, IPSN '09, pp. 133–144, Washington, DC, USA, 2009. IEEE Computer Society.

[64] S. Duquennoy, O. Landsiedel, and T. Voigt. Let the tree bloom: Scalable opportunistic routing with orpl. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pp. 2:1–2:14, New York, NY, USA, 2013. ACM.

[65] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, July 1970.

[66] S.R. Mihaylov, M. Jacob, Z.G. Ives, and S. Guha. A substrate for in-network sensor data integration. In *Proceedings of the 5th Workshop on Data Management for Sensor Networks*, DMSN '08, pp. 35–41, New York, NY, USA, 2008. ACM.

[67] E. Cañete, M. Daz, L. Llopis, and B. Rubio. Hero: A hierarchical, efficient and reliable routing protocol for wireless sensor and actor networks. *Computer Communications*, 35(11):1392–1409, June 2012.

[68] D.B. Johnson and D.A. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H.F. Korth, editors, *Mobile Computing*, *The Kluwer International Series in Engineering and Computer Science*, vol. 353, pp. 153–181. Springer, USA, 1996.

[69] S-J. Lee and M. Gerla. Aodv-br: backup routing in ad hoc networks. In *Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE*, vol. 3, pp. 1311–1316, 2000.

[70] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *WMCSA '99. Second IEEE Workshop on Mobile Computing Systems and Applications, 1999. Proceedings*, WMCSA '99, pp. 90–100, Feb 1999.

[71] R.C. Shah and J.M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, vol. 1, pp. 350–355, Mar 2002.

[72] R. Kolcun, D.E. Boyle, and J.A. McCann. Efficient distributed query processing. *IEEE Transactions on Automation Science and Engineering*, pp. 1–17, July 2016.

[73] S.M. Hedetniemi, S.T. Hedetniemi, and A.L. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349, 1988.

[74] W.R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '99, pp. 174–185, New York, NY, USA, 1999. ACM.

[75] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, WSNA '02, pp. 22–31, New York, NY, USA, 2002. ACM.

[76] P. Levis, N. Patel, D. Culler, and S. Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation - Volume 1*, NSDI'04, pp. 2–2, Berkeley, CA, USA, 2004. USENIX Association.

[77] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele. Low-power wireless bus. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pp. 1–14, New York, NY, USA, 2012. ACM.

[78] O. Landsiedel, F. Ferrari, and M. Zimmerling. Chaos: Versatile and efficient all-to-all data sharing and in-network processing at scale. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pp. 1:1–1:14, New York, NY, USA, 2013. ACM.

[79] G. Mulligan. The 6lowpan architecture. In *Proceedings of the 4th Workshop on Embedded Networked Sensors*, EmNets '07, pp. 78–82, New York, NY, USA, 2007. ACM.

[80] M. Kovatsch, S. Duquennoy, and A. Dunkels. A low-power coap for contiki. In *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 855–860, Oct 2011.

[81] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K.S.J. Pister. A realistic energy consumption model for TSCH networks. *IEEE Sensors Journal*, 14(2):482–489, Feb 2014.

[82] N. Burri, P. von Rickenbach, and R. Wattenhofer. Dozer: Ultra-low power data gathering in sensor networks. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, IPSN '07, pp. 450–459, New York, NY, USA, 2007. ACM.

[83] G. Z. Papadopoulos, K. Kritsis, A. Gallais, P. Chatzimisios, and T. Noel. Performance evaluation methods in ad hoc and wireless sensor networks: a literature study. *IEEE Communications Magazine*, 54(1):122–128, January 2016.

[84] P. Thubert. An architecture for IPv6 over the TSCH mode of IEEE 802.15.4. Internet-Draft draft-ietf-6tisch-architecture-09, Internet Engineering Task Force, November 2015. Work in Progress.