Towards Precision Control in Constrained Wireless Cyber-Physical Systems

David Boyle(✉), Roman Kolcun, and Eric Yeatman

Department of Electrical and Electronic Engineering,
Imperial College London, London, UK
{david.boyle,roman.kolcun,e.yeatman}@imperial.ac.uk

Abstract. This paper introduces the problem of high precision control in constrained wireless cyber-physical systems. We argue that balancing conflicting performance objectives, namely energy efficiency, high reliability and low latency, whilst concurrently enabling data collection and targeted message dissemination, are critical to the success of future applications of constrained wireless cyber-physical systems. We describe the contemporary art in practical collection and dissemination techniques, and select the most appropriate for evaluation. A comprehensive simulation study is presented and experimentally validated, the results of which show that the current art falls significantly short of desirable performance when inter-packet intervals decrease to those required for precision control. It follows that there is a significant need for further study and new solutions to solve this emerging problem.

Keywords: Wireless sensor networks · Cyber-physical systems · Control · Communications protocols · Data collection · Dissemination · Routing · Reconfiguration · Structural monitoring · Fundamental limits · Performance

1 Introduction

Wireless sensor and actuator networks will be a key enabling technology in the next generation of cyber-physical systems (CPS). The CPS paradigm introduces new functional and associated design requirements that are not typically considered in the development of wireless sensor network (WSN) technologies. A key differentiator between WSN and CPS can be stated in terms of the evolution from sensor networks designed exclusively to collect sensor data to those with the ability to augment sensor data collection with physical control over the environment using one or more actuators. Thus, the notion of *control* becomes a significant factor, which raises a number of important questions. Furthermore, the potential to interact with and exert control over the physical environment presents an entirely new challenge to the research community, particularly in attempting to bring cheap and effective networked embedded systems to the fore in industrial monitoring, control and automation applications. Depending

B. Mandler et al. (Eds.): IoT 360° 2015, Part II, LNICST 170, pp. 292–306, 2016.
DOI: 10.1007/978-3-319-47075-7_33

on the application, numerous potential control strategies may be applicable. This work considers the case where a central entity, man or machine, requires the ability to remotely control devices in a network. Notionally, this requirement may be to send a control message to a device equipped with an actuator to alter some physical state in the environment (e.g. adjusting a valve to achieve a desired rate of flow). This is not dissimilar to disseminating a message in a traditional WSN to change, for example, the rate at which a node samples a connected sensor, or its reporting interval (often referred to as *inter-packet interval* (IPI), i.e. the rate at which the application generates data packets to be transmitted towards a sink node) [7]. However, with the exception of RPL (Sect. 3), all contemporary approaches disseminate messages using network *flooding* mechanisms, irrespective of the intended recipient. We make the following assumptions:

- Applications of cyber-physical systems may include heterogeneous devices, i.e. numerous connected (or integrated) sensors and/or actuators [2], which participate in the same network
- Applications will to evolve from traditional networks collaboratively detecting distributed phenomena to targeted sensing of discrete, localised phenomena contributing to the monitoring and control of macro systems
- Applications will require the ability to exert fine-grained control over individual devices, which may include commands that can be generated autonomously or manually by *users*, i.e. human-in-the-loop

An important question is therefore raised: can (and if not, how can) we ensure effective, i.e. reliable and timely, transmission of *control* or *actuation* messages[1] to *specific* devices in a network? Furthermore, can this be done whilst respecting and adhering to traditional WSN design objectives, including energy efficiency, robustness and reliability? Similar questions have attracted increased attention in the recent literature, e.g. in [9,10][2] and [19]. Perhaps equally important are the questions: can both *collect* and *control* traffic types effectively coexist in a full network stack, what is the relationship between them, and what are the inherent trade-offs? There is a significant gap in the literature concerning how data collection, a standard function of most sensing systems, and fine-grained control, a *feature* that is increasingly required, can coexist in a fully integrated application stack in the context of CPS. In attempting to answer these questions, we present the following contributions in this paper:

- A rigorous evaluation of the performance of coexisting state-of-the-art *collect* and *control* protocols in a full network stack (Sect. 4), including quantified assessment of the trade-offs with regard to latency, energy efficiency and reliability performance under various operational loads. Our evaluation is predicated on a real application where WSN technology is used to monitor and

[1] The terms are hereinafter used interchangeably, and may apply to sending an actuation command or a reconfiguration command.

[2] *Downward routing* is a term also used to describe the traffic pattern for such messages, particularly in the standards community, e.g. [21].

control critical infrastructure, described in Sect. 2. We use selected state-of-the-art protocols described in Sect. 3.
– Evidence and subsequent examination, in Sects. 5 and 6, respectively, showing that the existing art does not sufficiently enable the level of high-fidelity control required for CPS without significant degradation in one or more performance metrics.

2 Background and Motivation

This work originates from efforts to apply wireless cyber-physical systems to the monitoring and control of critical infrastructures. Specifically, it stems from efforts to demonstrate the feasibility of WSN technology applied to structural health monitoring in an operational environment [1], i.e. that of a cable stayed bridge[3], during and post construction. Some interesting technical challenges were involved in the first instance, such as using heterogeneous sensors. Industrial, commercially available sensors were specified by collaborating geomonitoring specialists[4]. These included displacement, strain, inclination, acceleration, and pressure transducers, soil moisture probes, anemometers and precision temperature sensors. Each sensor type had varying electrical characteristics and communications interfaces [2]. They were connected to mote-class devices (below) in varying numbers and configurations, depending on the specified location on the structure and measurement of interest. This is representative of a new challenge in applied WSN research, whereby heterogeneous sensors, and their physical configuration, are used in combination to monitor a macro system whilst using a common wireless communication infrastructure. Traditional WSN implementations tend to use homogeneous devices across the network, collaboratively monitoring distributed phenomena, such as light and temperature. These features, coupled with highly localised sensing (e.g. using a displacement transducer, with a limited sensing range (i.e. centimetres), for crack detection), contributes to new challenges in terms of remote device interaction and control. The incorporation of hybrid energy harvesting and storage to effectively provide perpetual energy to the devices in the field was also required and demonstrated [1,17]. A simple system architecture was developed, where sensor data is periodically communicated over multiple hops towards a sink node, which in turn connects to a gateway and transfers data to networked servers, thus allowing analytics to be performed by domain experts.

Hardware. Bespoke sensor nodes, *mote*-class devices, were developed to satisfy monitoring and energy requirements, which include MSP430F5437[5] microcontrollers and TI CC2520[6] RFICs as key computational and communications components. We refer the interested reader to [1] and [17] for more on the hardware.

[3] http://www.fr.ch/poya/fr/pub/index.cfm.
[4] http://www.solexperts.com.
[5] http://www.ti.com/lit/ds/symlink/msp430f5419a.pdf.
[6] http://www.ti.com/lit/ds/symlink/cc2520.pdf.

Embedded Software and Communications. Original firmware was built using TinyOS [14], and used the available communications stack, i.e. CTP for data collection over BoX-MAC [18]. Remote management of the network was done using DRIP [20], with Deluge used for reprogramming [8] - both packaged with the TinyOS operating system. These are described in more detail in Sect. 3.

Energy Efficiency and Reliability. Energy efficiency is key to prolonged field operation. The hardware was designed to be as energy efficient as possible, achieving a quiescent current of <10 μA in the lowest power mode with sensors attached. Assuming inter-packet intervals of *hours* (suitable for long term monitoring), the system could be duty cycled aggressively (∼1 %). CTP has been shown to deliver >90 % reliability under most conditions. For low-rate, i.e. large IPI, and low-density deployments, this approaches 100 % depending on the link layer. For long term monitoring tolerant of minor loss, it was sufficient to achieve >95 % packet delivery, typically achievable using CTP over BoX-MAC [7].

2.1 Cyber-Physical Systems: Features and Design Objectives

The ability to remotely interact with individual devices in the network is a desirable feature that was not sufficiently implemented in our initial system. The end user requires control of individual devices during anomalous periods, e.g. to investigate a potentially dangerous situation detected by a device. This is achievable by changing the sampling rate or reporting interval of a particular sensor, connected to a particular port of a particular device. Theoretically, this also shifts focus towards high-fidelity control and actuation in CPS, where additional features and associated performance requirements are certain to emerge as applications become more complex. This work focuses on achieving high-precision control over devices, a required *feature* of both our original system and emerging CPS applications.

High-Precision Control. Where an actuator exists in the network, it is necessary to communicate with this device directly, with a high degree of reliability and low latency. Therefore, the ability to remotely exert fine grained control over each of these devices is of primary interest. There are some existing mechanisms to perform this task, assessed in more detail in the next section, where we later show and argue that they do not necessarily meet the following goals of an effective precision control mechanism for CPS. The main goals of a precise control scheme for wireless CPS are as follows:

- *Addressability:* each device should be individually addressable
- *Reliability:* control and actuation messages should achieve maximum delivery, i.e. approach 100 %
- *Efficiency:* the system should retain the principles of minimising the resources required to deliver messages to target devices, i.e. minimum amount of transmissions, minimal state, energy, etc.
- *Low Latency:* control messages should exhibit low latency, approaching fundamental lower bounds

– *Robustness:* the system should continue to operate irrespective of dynamic communications conditions, topologies, loads, etc.

There are additional requirements, such as hardware independence and security, of equally significant importance. We retain the principle of ensuring hardware independence, but regard security as being beyond the scope of this contribution.

3 Data Collection and Dissemination Protocols

Most implementations of WSNs use a single point of data collection. This typically requires the creation of a tree routing structure, rooted in a sink node (or relatively fewer sink nodes to sensing nodes). This constitutes converge-cast network traffic (many-to-one), and has been the focus of the majority of research and development efforts to-date. Data dissemination requires transmitting messages in the opposite direction, either sending network-wide messages (one-to-many, or flooding), or one-to-one (sometimes referred to as point-to-point or any-to-any). The latter requires more complex routing information, which we consider in more detail in the following sections.

Data Collection Protocols. There are numerous popular solutions for data collection in WSNs, such as Collection Tree Protocol (CTP) [7] and RPL [21], both of which are based on converge-cast communication towards one or more sink nodes. In the standards community, these are often described as *destination oriented directed acyclic graphs*. We disregard early protocols, such as MulithopLQI[7] and MintRoute [22], which are obsoleted and improved upon by CTP. There have been recent efforts to enable any-to-any, e.g. [6,11], and multi-mode downward routing, e.g. [10], with similar objectives in mind, discussed later.

Dissemination Protocols. If we consider a multi-hop network - with *any* underlying communications infrastructure - controlling individual devices primarily requires finding a *route* to the device in question, and transmitting the necessary command to that node. There are solutions for disseminating data in WSNs, including DIP [16], DRIP [20] and RPL [21], many of which are closely related to and use the Trickle algorithm [15]. These differ from protocols like Deluge [8] and MNP [12], in that they are designed to deliver small values, whereas the latter are designed to deliver larger files such as binary code updates. The majority of traditional WSN research considered networks of homogeneous devices, where updates and configuration commands were disseminated to every device in the network, leveraging local neighbourhoods to advertise and agree upon *versions*. More complex dissemination is possible using the existing art, e.g. as described in TEP 118[8], performing predicate-based changes, implemented by layering interfaces above the existing low-level networking primitives. There are other proposed solutions that attempt to solve both simultaneously, such as Chaos [13]. As with many advances in WSNs, each solution is often well suited to different

7 http://www.tinyos.net/tinyos-2.x/tos/lib/net/lqi/.
8 http://www.tinyos.net/tinyos-2.1.0/doc/html/tep118.html.

scenarios, having been developed with different functionalities and performance metrics in mind. The function of a collection protocol is intuitive, however, the thinking behind dissemination requires more careful consideration, particularly as high-fidelity precision control is required over individual devices. For the purposes of our evaluation, we select CTP and Trickle as the most applicable protocols to evaluate as part of a full application. These are mature and well studied protocols that exemplify the current art in efficient and reliable wireless networking. Furthermore, their implementation does not require the accommodation of additional overheads associated with the standards under development. Finally, commercial protocols, including ZigBee and WirelessHART, are not considered. This is due to their implementation of the IEEE 802.15.4 MAC, which is founded on global time synchronisation[9]. This is significantly less energy efficient than asynchronous (or semi-synchronous) radio duty cycled MAC protocols, such as ContikiMAC and BoX-MAC.

4 Experimental Evaluation

To evaluate the performance of the selected coexisting collection and dissemination protocols, we implemented a full application using the Contiki operating system [3][10]. The application uses CTP for data collection, implemented as the Contiki collect protocol, and Trickle for the dissemination of messages using the Rime network stack [5], with ContikiMAC to manage the radio [4] using the default settings. ContikiMAC tends to provide better reliability and energy efficiency than BoX-MAC, and is therefore a natural design choice. We include a randomised destination address (of a node in our network) for each dissemination message in the payload of the packet (which in our case is less than the minimum Contiki packet size), and post-process the stored log files to compute results, thus allowing Trickle to be used without further modification.

4.1 Simulation

Cooja, the simulator/emulator packaged with the Contiki operating system, was used for this study. The Multi-path Ray-tracer Medium (MRM) radio model provided by Cooja was used to simulate a realistic radio environment. The noise floor was set to −90 dBm with a standard deviation of 2 dBm. These values were shown to be realistic for low noise environments in [9].

Topology. The topology used for simulation consists of a network of 45 nodes deployed in 3-dimensional space, guided by the bridge (see footnote 3) monitoring scenario described in Sect. 2. This corresponds to nodes placed 50 m apart, lining either side of the bridge (10 m separation) spanning 850 m. Nodes are

[9] The amended IEEE802.15.4e (TSCH) is insufficiently mature for consideration.

[10] We use the latest stable Contiki release, Contiki 2.7, available: http://www.contiki-os.org/download.html.

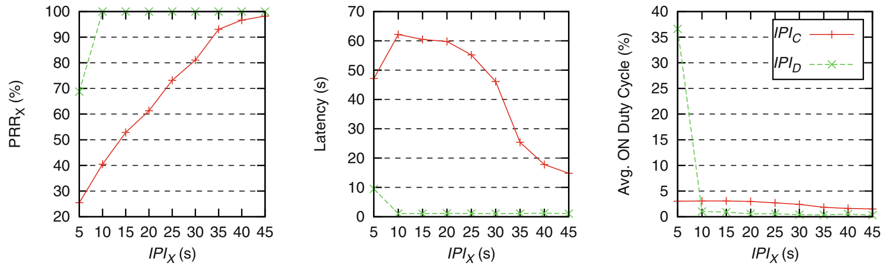
placed vertically perpendicular to the x- and y-axes on two pylons (325 m and 625 m from the sink, respectively) to a height of +150 m in the z-axis, using 3 nodes spaced 50 m apart. The sink node is placed 50 m from the node closest to the end of the bridge.

Parameter Selection and Evaluation Metrics. Inter-packet intervals (IPI) are swept from 5 to 45 s with 5 s interleaved, denoted IPI_D for dissemination and IPI_C for collect. IPI_C tends to vary depending on the application and the granularity of the sensor data required. It may be tuned towards saturation (i.e. $IPI_X \rightarrow 0$) during periods when additional data points are required to more thoroughly assess a situation[11]. The duration of the simulation was chosen so both protocols sent at least 250 messages, i.e. $t = 250 \times max(IPI_D, IPI_C)$, where *every* node must transmit the minimum number of collect messages. Therefore, the total number of simulated collect messages transmitted ranges from ∼11,250 to ∼101,250, and the number of control messages varies from a minimum of 250 to a maximum of ∼2250. This has a significant impact on the duration of each simulation run. At the beginning of the simulation, the network is afforded 120 s for the collect protocol to settle and obtain information about nodes' neighbourhoods. Thereafter, both protocols begin transmitting packets at the predefined rates. For the collect protocol, upon the expiration of the predefined interval, a secondary random timer is started, within a maximum length of 20 % of the main timer, to reduce congestion and increase the delivery rate. When the main timer expires on the base-station, a message is generated and sent to a node in the network, chosen at random. Because Trickle is initiated by one node only, there is no need for a secondary timer, as it would have no impact on the delivery rate. The evaluation focuses on following metrics: (i) reliability, (ii) latency, and (iii) energy efficiency. Reliability is calculated as percentage of messages that are successfully delivered to their destination. Latency is calculated as the time difference between the moment when the message was generated by the source node and the time the message is delivered to its destination. Timestamps are collected from the log file generated by Cooja. Energy efficiency is evaluated with the PowerTracker plugin, which tracks, for each node, how long the radio is turned on, transmitting or receiving.

5 Results and Evaluation

For each of the 81 $\{IPI_D, IPI_C\}$ pairs, we ran the experiment three times (i.e. 243 runs in total). For each run, average reliability, latency, and energy efficiency were computed. We discuss these metrics individually in the following subsections, where the results (i.e. each data point) is the computed average from three simulation runs. In addition to running the collect and dissemination protocols

[11] The literature suggests typical IPI_C values $\simeq$ 15 s. We include this interval, in addition to approaching saturation and selecting numerous divergent values. The same is done for IPI_D, where typical values for this frequency are relatively unknown.



(a) Packet Reception Rate (b) Latency (c) Radio Duty Cycle

Fig. 1. Baseline figures for reliability (a), latency (b) and energy (c) performance for collect (red) and disseminate (green) protocols running independently (Color figure online)

in parallel, we also ran them separately to obtain baseline data. The results are illustrated in Fig. 1, which shows performance to be in line with expectation.

5.1 Reliability

Figure 2 shows that the collect protocol delivers less than 10 % of the messages when $IPI_D \leq 10$ s, regardless of the collect interval. For larger IPI_D, the reliability of the collect protocol increases, and stabilises at a certain value. This value differs, on average, by ∼1.7 % from the baseline collect reliability, i.e. without the dissemination protocol running in parallel. This shows that significant performance degradation occurs in terms of data collection when dissemination intervals are below 25 s. Dissemination reliability is shown in Fig. 3. It is clear that for $IPI_D \leq 25$ s, the packet reception rate for disseminated messages (PRR_D) degrades significantly. There are two further interesting observations to be made. The first is that, for $IPI_D \geq 25$ s, PRR_D is consistently approaches 100 %. This demonstrates how successful dissemination is when compared to collection. This

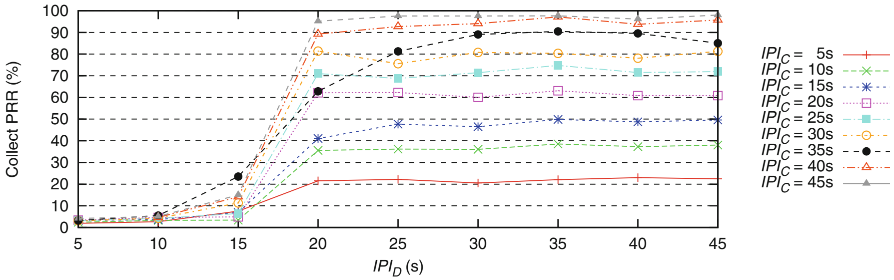


Fig. 2. Packet Reception Rate (PRR) for collect messages (PRR_C) for each inter-packet interval pair $\{IPI_D, IPI_C\}$.

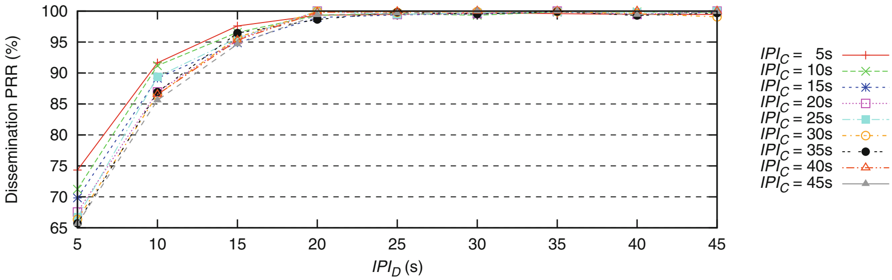


Fig. 3. Packet Reception Rate (PRR) for disseminate messages (PRR_D) for each inter-packet interval pair $\{IPI_D, IPI_C\}$.

is intuitive, as data collection via a tree rooted in a sink (many to one) is inherently more difficult than network-wide message dissemination (one to many). Secondly, there are some counter-intuitive data points in the graph. Looking at $IPI_C = 5\,\text{s}$, we can see that the average PRR_D is greater than that for larger IPI_C intervals. Owing to the random component used in the simulation and use of computed averages, the exact reason is difficult to pinpoint. High reliability is characteristic of the Trickle algorithm's dynamic tuning, and ability to rapidly propagate messages [15]. Practical evaluation of this phenomenon is inconclusive (Fig. 7(a)), and is worthy of further study. This does not detract from the overall result showing that for shorter IPI_D & IPI_C reliability performance is significantly degraded.

5.2 Latency

Figure 4 shows the time required for packets to be collected from the network. It follows the same trend as for reliability. It is worth remembering that the depth of the network is significant, where the number of hops can be as many as 15 to and from the sink[12]. This is a linear function of the delay experienced in end-to-end communication over multiple hops, and depends heavily on the *receive check interval* of the underlying MAC protocol in duty cycled asynchronous[13] CSMA-CA approaches. For $IPI_D \leq 10\,\text{s}$, it takes on average more than one minute for a message to be delivered to the sink, regardless the value of IPI_C. As IPI_D increases, the collect time stabilises at a certain value. This value does not differ by more than $\sim$10 % from the average collection time of the collect protocol baseline for that IPI_C. Figure 5 shows the time taken for messages to be disseminated to predetermined destination nodes in the network, on average. In this case it is clear that performance degrades for $IPI_D \leq 20\,\text{s}$ for all IPI_C. In all cases, this is approximately an order of magnitude more efficient than data collection. This is similarly attributable to the fundamental difference between

[12] The average number of hops across all experiments is $\simeq 4.3$.

[13] We consider asynchronous MACs to be those without global or centrally coordinated time synchronisation.

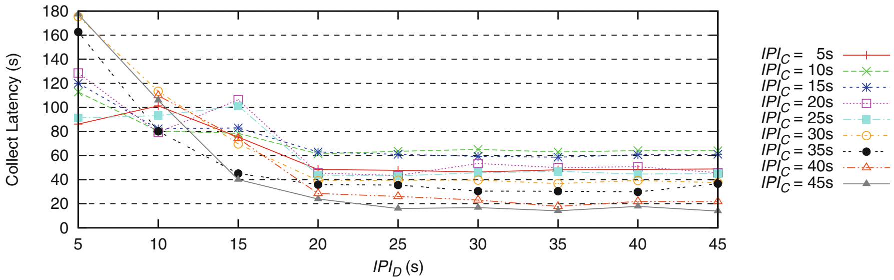


Fig. 4. End-to-end latency for collect messages for each inter-packet interval pair $\{IPI_D, IPI_C\}$.

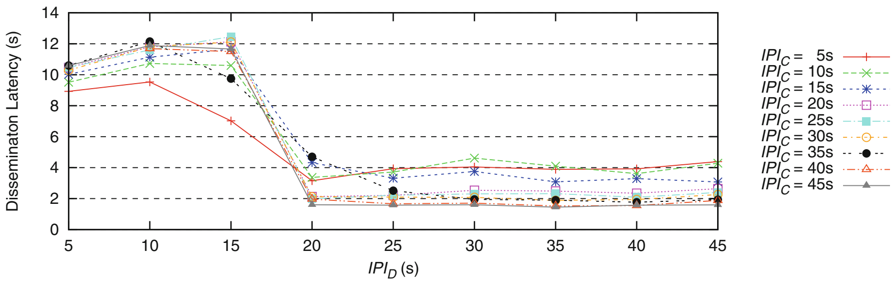


Fig. 5. End-to-end latency for disseminate messages for each inter-packet interval pair $\{IPI_D, IPI_C\}$.

collection towards a single point and dissemination throughout the network, and Trickle's ability to rapidly propagate messages.

5.3 Energy Efficiency

To estimate energy efficiency, we use the radio duty cycle (RDC) as a proxy. Although it is not a true approximation of the energy cost for our target system (where sampling the sensors is often significantly more energy intensive than radio communication [2]), we are interested in the impact of the protocols' coexistence on the system's energy efficiency, and thus it is sufficient for the purposes of this evaluation. Figure 6 shows the total time for which the radio is active for each $\{IPI_D, IPI_C\}$ pair. It shows that for $IPI_D < 20$ s the RDC ranges from 31–36%. The radio *ON* time in Fig. 6 is the sum of the TX and RX on times[14]. This is a significant energy cost, where total RDC for these types of systems would ideally be kept as close to $\sim 1\,\%$ as possible. The ratio of time the radio spends in each of RX (i.e. *listen*) and TX (*transmit*) modes is approximately

[14] We disregard the total ON time. For *all* $IPI_D < 20$ s, ON is in the region 60–80%. We recalculate this as the sum of RX and TX active times, as they are reflective of the higher energy modes of the RFIC when listening and transmitting, respectively.

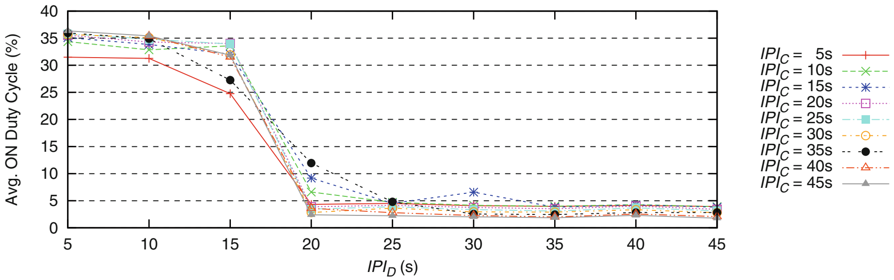
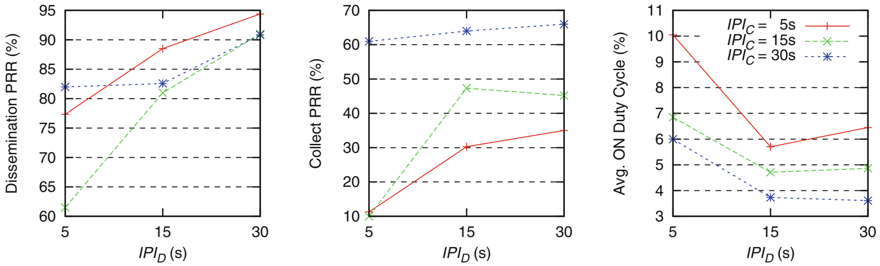


Fig. 6. Energy efficiency overhead estimation by proxy of radio duty cycle for each inter-packet interval pair $\{IPI_D, IPI_C\}$.



(a) Disseminate PRR (b) Collect PRR (c) Average RDC

Fig. 7. Practical results for a subset of $\{IPI_D, IPI_C\}$ pairs showing (a) disseminate Packet Reception Rate (PRR), (b) collect PRR, and (c) radio duty cycle

10 : 1, respectively. It is again noticeable that for $IPI_C = 5\,\text{s}$, there is a counter-intuitive result. In this case, looking at Fig. 6, the radio consumes less energy. However, this does not detract from the overall result that there is a significant cost increase for all $IPI_D < 20\,\text{s}$, and corroborates the preceding reliability and latency results. Compared with the baseline energy performance results for each protocol shown in Fig. 7(c), it can be seen that there is a significant overhead when the protocols coexist, most noticeably for $IPI_D \leq 20\,\text{s}$.

6 Discussion

The results in Sect. 5 demonstrate that there are significant shortcomings in the performance of the best, i.e. most efficient, reliable and robust, communications protocols for data dissemination and collection when concurrently deployed in the context of emerging applications of CPS that may use constrained wireless networks. Our evaluation shows that energy requirements are significantly increased for small inter-packet intervals, and the reliability performance of the collection protocol is significantly affected by the presence of regular control messages in the network. PRRs $< 30\,\%$ for $IPI_D < 20\,\text{s}$ will hardly be acceptable for CPS. This is hugely problematic when considering the nature of control applications, where timely feedback, i.e. sensor data, is required to make control decisions. Furthermore, we show that latency with regard to sensor data collection is largely doubled for small IPI_D.

6.1 Caveats and Limitations

Our initial evaluation is based on simulation only, which is largely considered to be insufficient. There are justifiable reasons to adopt this approach. An evaluation *in the wild* for our use case would require uninterrupted access to approximately 1 km of operational civil infrastructure to conduct 81 individual experiments (243 to obtain similar averages). Gaining such access is almost impossible for the purposes of experimentation. Therefore, simulation is essential to conduct necessary pre-deployment validation and test. Accordingly, the simulation experiments in Sect. 4 considered a very specific application scenario, which tightly governs the network topology. Nevertheless, we conducted a smaller scale practical experiment on a subset of the $\{IPI_D, IPI_C\}$ pairs to assess and demonstrate the effects presented in Sect. 5, as follows.

Practical Evaluation. We built a reduced linear network, where 13 TelosB (clones) were distributed over three floors of the Electrical and Electronic Engineering building at Imperial College London. Experiments were conducted to generate data points for 9 $\{IPI_D, IPI_C\}$ pairs, governed in duration by IPI_D for 250 disseminated messages in each case. The average number of hops per message transmitted was $\simeq 3.8$; reasonably close to the 4.3 in the simulated networks. IEEE 802.15.4 channels 25 and 26 were used for each pair, in a typically noisy environment, and averages computed.

Figure 7 shows the results for PRR of dissemination and collect messages, and energy efficiency (by proxy of RDC, calculated by counting the clock ticks during active (i.e. TX and RX) modes of operation). Irrespective of the reduced scale of the experiment, the results show that performance degradation is in relative agreement with the simulation, thus validating the existence of the problem. It is likely that for different network topologies, depths and dynamic RF environments, the results will vary.

Interestingly, there is a marked improvement in the PRRs for $\{IPI_D, IPI_C\} = \{5\,\text{s}, 30\,\text{s}\}$. This is a fairer reflection, and attributable to both significantly fewer collect messages congesting the network and fewer overall nodes in the network (with respect to the simulation study). Nonetheless, the performance is still less than ideal for prospective wireless CPS applications, and the network is shown remain essentially non-functional for shorter IPI pairs. The energy performance of the nodes in the network relatively improved, but in larger networks, more messages in both directions will increase the active periods of the RFIC.

Finally, we neglect to propose a new solution that improves the *status quo*. This is because it is essential to completely understand the performance limitations of the state-of-the-art before proposing any new alternative. This constitutes essential knowledge for potential adopters of technologies using constrained wireless components. Methods to improve the situation are under development, but are left to future work. New methods to effectively store and manage routing information will be essential to overcome these performance limitations, and will exploit increasing on-chip memory, processing capability and energy efficiency.

7 Related Work

Stolikj *et al.* proposed improvements to the performance of Trickle by enhancing cross-layer interactions, showing how MAC layer implementations can lead to violations of Trickle's delicate timing requirements [19]. Isomin *et al.* show that current, and improved, versions of contemporary dissemination techniques fall short of the requirements for actuation in wireless CPS, and question of whether *P2P* traffic support is even necessary for CPS. They illustrate that Trickle outperforms an improved variant of RPL in terms of efficiency, a result that we leverage in the selection of Trickle in this work [9]. Furthermore, based on our evaluation, it is evident that there is cause to implement more complex downward routing information to improve overall network performance. Ko *et al.* propose DualMOP to solve the problem of heterogeneous modes of operation (i.e. storing and non-storing modes) of the RPL standard in overlapping networks [10]. Dunkels *et al.* proposed the Announcement Layer to coordinate beacons in the network to reduce overheads associated with concurrent collection and dissemination strategies. They extend Contiki Shell application through `netcmd` to demonstrate how a command may be run on *all* nodes in the network. They do not consider the requirement for direct addressability, or finding efficient methods that do not flood the entire network. They show how beacon coordination can contribute to improvements, e.g. reducing the overall number of beacons required by 9 % by exploiting coordinated announcements [4]. Dunkels *et al.*'s approach, if extended, would be similar to the explanation in TEP 118 on how to handle more complex dissemination, discussed in Sect. 3. However, none of the existing approaches directly address the problem of high precision control in wireless sensor and actuator networks where there exists the need for concurrent, highly reliable, timely and energy efficient data collection and (addressable) dissemination, which will be required for wireless CPS applications.

8 Conclusion

In this paper we describe the emerging problem of high precision control over cyber-physical systems with constrained wireless devices in the loop. We articulate the need for concurrent collection and dissemination strategies that satisfy conflicting objectives; namely energy efficiency, high reliability and low latency. We select two of the most established, studied and optimised protocols for each, CTP and Trickle, and evaluate them across a comprehensive range of inter-packet intervals in simulation. We demonstrate how these protocols, when implemented concurrently, do not achieve the performance requirements necessary for applications of cyber-physical systems. It follows that there is a significant need for additional study and new solutions to solve the problem of high precision control in constrained wireless cyber-physical systems.

References

1. Boyle, D., Magno, M., O'Flynn, B., Brunelli, D., Popovici, E., Benini, L.: Towards persistent structural health monitoring through sustainable wireless sensor networks. In: ISSNIP 2011, pp. 323–328 (2011)
2. Boyle, D., Srbinovski, B., Popovici, E., O'Flynn, B.: Energy analysis of industrial sensors in novel wireless SHM systems. In: 2012 IEEE Sensors, pp. 1–4, October 2012
3. Dunkels, A., Gronvall, B., Voigt, T.: Contiki - a lightweight and flexible operating system for tiny networked sensors. In: 29th Local Computer Networks, November 2004
4. Dunkels, A., Mottola, L., Tsiftes, N., Österlind, F., Eriksson, J., Finne, N.: The announcement layer: beacon coordination for the sensornet stack. In: Marrón, P.J., Whitehouse, K. (eds.) EWSN 2011. LNCS, vol. 6567, pp. 211–226. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19186-2_14
5. Dunkels, A., Österlind, F., He, Z.: An adaptive communication architecture for wireless sensor networks. In: SenSys 2007, pp. 335–349 (2007)
6. Duquennoy, S., Landsiedel, O., Voigt, T.: Let the tree bloom: scalable opportunistic routing with ORPL. In: SenSys 2013
7. Gnawali, O., Fonseca, R., Jamieson, K., Moss, D., Levis, P.: Collection tree protocol. In: SenSys 2009
8. Hui, J.W., Culler, D.: The dynamic behavior of a data dissemination protocol for network programming at scale. In: SenSys 2004
9. Istomin, T., Kiraly, C., Picco, G.P.: Is RPL ready for actuation? A comparative evaluation in a smart city scenario. In: Abdelzaher, T., Pereira, N., Tovar, E. (eds.) EWSN 2015. LNCS, vol. 8965, pp. 291–299. Springer, Heidelberg (2015). doi:10.1007/978-3-319-15582-1_22
10. Ko, J., Jeong, J., Park, J., Jun, J.A., Gnawali, O., Paek, J.: DualMOP-RPL: supporting multiple modes of downward routing in a single rpl network. ACM Trans. Sen. Netw. **11**(2), 39:1–39:20 (2015)
11. Kolcun, R., McCann, J.: Dragon: data discovery and collection architecture for distributed IoT. In: Internet of Things (IOT) 2014, pp. 91–96, October 2014
12. Kulkarni, S., Wang, L.: MNP: multihop network reprogramming service for sensor networks. In: Proceedings of the Distributed Computing Systems, ICDCS 2005 (2005)
13. Landsiedel, O., Ferrari, F., Zimmerling, M.: Chaos: versatile and efficient all-to-all data sharing and in-network processing at scale. In: SenSys 2013, pp. 1:1–1:14. ACM
14. Levis, P., et al.: Tinyos: an operating system for sensor networks. In: Weber, W., Rabaey, J., Aarts, E. (eds.) Ambient Intelligence, pp. 115–148. Springer, Berlin (2005)
15. Levis, P., Patel, N., Culler, D., Shenker, S.: Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In: NSDI 2004
16. Lin, K., Levis, P.: Data discovery and dissemination with DIP. In: IPSN 2008
17. Magno, M., Boyle, D., Brunelli, D., O'Flynn, B., Popovici, E., Benini, L.: Extended wireless monitoring through intelligent hybrid energy supply. IEEE Trans. Ind. Electron. **61**(4), 1871–1881 (2014)

18. Moss, D., Levis, P.: BoX-MACs: exploiting physical and link layer boundaries in low-power networking. Computer Systems Laboratory Stanford University (2008)
19. Stolikj, M., Meyfroyt, T.M.M., Cuijpers, P.J.L., Lukkien, J.J.: Improving the performance of trickle-based data dissemination in low-power networks. In: Abdelzaher, T., Pereira, N., Tovar, E. (eds.) EWSN 2015. LNCS, vol. 8965, pp. 186–201. Springer, Heidelberg (2015). doi:10.1007/978-3-319-15582-1_12
20. Tolle, G., Culler, D.: Design of an application-cooperative management system for wireless sensor networks. In: EWSN 2005
21. Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J., Alexander, R.: RPL: IPv6 routing protocol for low-power and lossy networks. RFC 6550 (Proposed Standard), March 2012
22. Woo, A., Tong, T., Culler, D.: Taming the underlying challenges of reliable multihop routing in sensor networks. In: SenSys 2003