Computer Communications 74 (2016) 63–76

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

# TinyIPFIX: An efficient application protocol for data exchange in cyber physical systems ☆

Corinna Schmitt [a,*], Thomas Kothmayr [b], Benjamin Ertl [c], Wen Hu [d], Lothar Braun [b], Georg Carle [b]

[a] Department for Informatics, Communication Systems Group, University of Zurich, Switzerland
[b] Department of Computer Science, Technische Universität München, Germany
[c] Institute for Telecommunication Systems, Chair for Next Generation Networks, Technische Universität Berlin, Germany
[d] CSIRO, Brisbane, Australia

ARTICLE INFO

*Article history:*
Available online 12 June 2014

*Keywords:*
Wireless sensor networks
TinyIPFIX
Aggregation
Efficient data format
Cyber-physical system

ABSTRACT

Wireless sensor networks (WSNs) as a central part of cyber-physical systems are gaining commercial momentum in many areas, including building monitoring and intelligent home automation. Users wish to successively deploy hardware from different vendors. Interoperability is taken for granted by the customers who want to avoid the need for exhaustive configuration and set-up. Therefore, the need for an interoperable and efficient application layer protocol for machine-to-machine communication in and across the boundaries of WSNs arises. We address these issues with our implementation of TinyIPFIX, an adaption of the IP Flow Information Export (IPFIX) protocol. Throughout the paper we show how to leverage TinyIPFIX in the context of an office scenario and we discuss how the protocol may be applied to other significant WSN deployments presented in literature over the past few years. This article additionally shows how to improve the functionality of TinyIPFIX by adding both syntactic and semantic aggregation functionality to the established system. Finally, we evaluate the performance of TinyIPFIX in a large test bed with over 40 motes running TinyOS and analyze TinyIPFIX's system performance in comparison with previous approaches.



## 1. Introduction

The Internet of Things (IoT) is seeing rapid adaption across many industries. For example, Cisco Systems is predicting a growth from 369 million machine-to-machine (M2M) modules in 2012 to 1.7 billion M2M modules globally in 2017 [3] – and these figures are only for mobile M2M devices that connect via cellular networks. Arguably, the number of devices connecting via local wireless networks is even higher. Because the IoT has a plethora of different usage scenarios [4], it also covers a wide range of device classes from powerful smartphones on the high end to devices that are highly constrained in memory, energy supply and computing capacity.

The focus of this paper lies in delivering an efficient application protocol for machine-to-machine communication in a cyber-physical system. Our target device classes are the constrained devices (motes) found in wireless sensor networks (WSNs), which often form a key component of a CPS. One common application area can be found in the field of building automation, meaning the autonomic monitoring and control of environmental conditions in residential and commercial buildings for improved comfort, as well as a reduced energy usage and carbon footprint. Wireless sensors are deployed to monitor key values, such as room temperature or brightness, in different locations. They transmit the data to a control and management system, which analyses the measurements and reacts on the results, e.g. turning on/off heating or lights. Not only the devices themselves are constrained in this scenario. The low power wireless network over which they communicate also imposes severe limits on throughput and message size. Any application protocol used in this scenario must be efficient in its use of

☆ Part of this work was published in Proceedings of the 7th European Conference on Wireless Sensor Networks (EWSN) 1 and was mostly done when Corinna Schmitt and Benjamin Ertl were with Technische Universität München 2. The extensions to the EWSN article include: First, an in-depth analysis on how to apply TinyIPFIX for a wide range of sensor network deployments. Secondly we demonstrate in-network aggregation support under TinyIPFIX, which includes data and message aggregation, as well as individual and direct configuration of the aggregation functionality on aggregator nodes. Third, an extensive analysis and system level evaluation of TinyIPFIX's transmission efficiency and resource consumption is presented along with a comprehensive comparison with other approaches. Compared to the original paper, there are significant modifications in Sections 3.4, 4 and 5.

* Corresponding author. Tel.: +41 44 635 7585.
*E-mail addresses:* schmitt@ifi.uzh.ch (C. Schmitt), kothmayr@in.tum.de (T. Kothmayr), benjaminertl@tu-berlin.de (B. Ertl), wen.hu@csiro.au (W. Hu), braun@net.in.tum.de (L. Braun), carle@net.in.tum.de (G. Carle).

http://dx.doi.org/10.1016/j.comcom.2014.05.012

network, computational and energy resources. However, it should still be comfortable to use for the system developer as well as complete and generic enough to allow easy deployment by the end user.

Web services are a well-known approach to M2M communication, but directly adapting the techniques from traditional computing are not feasible in constrained networks because they rely on verbose XML formats to exchange messages. For example, it takes up to 442 bytes to get a temperature value encapsulated with SOAP 1.2 [5]. This is often addressed through compression of the XML data [6] or by directly encoding the XML message in a binary format [7]. HTTP itself is also considered too resource intensive for constrained networks and alternatives like the Constrained Application Protocol (CoAP) [8] have been developed. The related work is discussed in Section 2 in more detail.

While suited to the domain of constrained networks, CoAP also introduces additional implementation complexity that might not be needed in all usage scenarios of WSNs. For example, a wireless thermometer that periodically reports to an automation server with a direct user interface is duplicating functionality that is available by accessing the data on the data sink it is reporting to. Our approach focuses on the core functionality of stationary Wireless Sensor Networks: Periodic reporting of sensor data to a data sink with low network, memory and computational overhead while still enabling easy integration of diverse sensor hardware on motes from different vendors with minimal configuration and maintenance overhead. The approach is centered around TinyIPFIX, a lightweight adaption of the IP Flow Information Export (IPFIX) protocol [9] for WSNs. Section 3 presents a brief characterization of the TinyIPFIX protocol and discusses its characteristics focusing on the constrains of wireless sensor nodes. In general, the design space for an application protocol to achieve tight integration of a WSN into a CPS consists of four areas:

### 1.1. Metrology

Sensor devices measure data, which has a specific format and must be represented accordingly. This representation should be general and universal, meaning that a protocol should be able to uniquely designate each measurement type across all WSN deployments. A measurement type is defined here as a reading from a specific model of a sensor, which carries information about the data type and its conversion to scientific units, rather than an abstract quantity such as "temperature". In the case of TinyIPFIX the sensor measurement data is identified by an individual Type ID and Enterprise Number (EID), which are registered with the Internet Assigned Number Authority (IANA).[1] This ensures adaptability to other platforms or new measurement types. Since an IPFIX Template only carries syntactical meta data for the measurements sent in an IPFIX Data packet the semantics for that data still need to be supplied. If the Enterprise and Type IDs have been allocated globally unique, a public repository for this semantic data, presented as XML markup, becomes feasible. We will give an example for such a markup in Section 5.4.

### 1.2. Resource efficiency

The resources of sensor nodes are limited in terms of power, memory space and computational capacities. We evaluate TinyIPFIX with regard to its memory requirements and energy consumption. Additionally, we implemented the TinyIPFIX-Aggregation protocol, which offers in-network aggregation mechanisms for data pre-processing. By leveraging in-network aggregation additional energy savings can be achieved through transmission reduction.

### 1.3. Syndication

The benefits of using IPv6 in sensor networks were detailed in previous work [10]. We choose to send TinyIPFIX packets via the BLIP [11] implementation of IPv6 and UDP, because it offers seamless integration into an existing IP-based network infrastructure.

### 1.4. Scalability

In Section 5.1 we discuss the flexibility of TinyIPFIX by showing how it could have been leveraged in other significant deployments presented at IPSN or SenSys over the past few years. We present the results of numerous real world test runs of TinyIPFIX assuming an office scenario (see Figs. 1 and 10) and in a large WSN deployment on the Harvard Sensor Network Testbed (Motelab) testbed [12].

Section 4 describes the integration of a TinyIPFIX based wireless sensor network into a cyber-physical system used for building automation. We evaluate the performance of the TinyIPFIX protocol concerning its hardware requirements and demonstrate the functionality of the whole system in Section 5 before concluding the paper in Section 6.

## 2. Related work

Widespread adaption of traditional web services in constrained networks is stymied by HTTP's verbosity. With the Constrained Application Protocol (CoAP) Shelby et al. introduced a lightweight, yet interoperable, alternative to HTTP that allows the adaption of the web service principle to constrained networks [8]. Compared to HTTP, CoAP's main benefits are a reduced header size and no requirement for reliable message transport (i.e., CoAP only requires UDP and not TCP). Motes that have data to expose implement a CoAP server and expose their data offerings to the data consumers via a discovery service. Similar to HTTP, CoAP does not specify the actual format in which data is transported but supports different content encodings. Our approach, which is centered on IPFIX, is more comparable to a content encoding format in the context of CoAP. CoAP and IPFIX for sensor networks therefore have different concerns: While CoAP's goal is to bring the full suite of features that is required for a web-like experience to constrained networks, we aim to offer a simple M2M application protocol with minimal implementation and network overhead that can be used where the full set of features offered by CoAP is not necessary or the implementation complexity cannot be afforded.

A more direct comparison can be drawn between IPFIX and different content encoding formats used with CoAP, HTTP or standalone: XML is arguably the most well known format for transferring structured data in a human readable way. However, the clear text format of XML results in very large message sizes and slow processing times – even in the field of traditional computing. JSON is a more compact format to transfer structured, human readable data but it still cannot achieve the same level of message compactness as binary formats. A large amount of effort has been undertaken to reduce the size of XML documents while simultaneously improving their processing speed. Two representative approaches are Fast Infoset [13] and Efficient XML Interchange (EXI) [7]. Compared to Fast Infoset, EXI achieves a higher rate of compression because it is able to take the structure information provided by an XML schema into account. However, both the encoding and decoding party needs to process the matching schema to leverage the increased rate of compression. A schema-less mode is available in EXI as well.

[1] http://www.iana.org.