

Demo Abstract: Bringing Full-Scale TCP to Low-Power Networks

Sam Kumar

Computer Science Division
University of California, Berkeley
samkumar@cs.berkeley.edu

Hyung-Sin Kim

Computer Science Division
University of California, Berkeley
hs.kim@cs.berkeley.edu

Michael P Andersen

Computer Science Division
University of California, Berkeley
m.andersen@cs.berkeley.edu

David E. Culler

Computer Science Division
University of California, Berkeley
culler@cs.berkeley.edu

ABSTRACT

Although TCP has widespread adoption in the Internet, wireless sensor networks (WSNs) generally use simpler UDP-based protocols. The few existing TCP implementations for sensor network operating systems do not support all of the features of TCP. We present a full-scale TCP implementation for sensor networks, called *TCPlp*, based on the TCP protocol logic of the FreeBSD Operating System. Our implementation demonstrates that full-scale TCP can run within the resource constraints of a modern WSN platform, and serves as a vehicle to explore the benefits of using a full TCP stack in the WSN setting. We showcase *TCPlp* via three applications of TCP: (1) reliable data collection in the context of an application, (2) an interactive configuration/debug shell, and (3) a mote-based web server.

CCS CONCEPTS

• **Computer systems organization** → **Sensor networks**; • **Networks** → *Network experimentation*; Transport protocols;

KEYWORDS

Sensor Network, Low-Power Network, Transport Layer, Transmission Control Protocol, IEEE 802.15.4

ACM Reference Format:

Sam Kumar, Michael P Andersen, Hyung-Sin Kim, and David E. Culler. 2018. Demo Abstract: Bringing Full-Scale TCP to Low-Power Networks. In *The 16th ACM Conference on Embedded Networked Sensor Systems (SenSys '18)*, November 4–7, 2018, Shenzhen, China. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3274783.3275196>

1 INTRODUCTION

Wireless sensor networks (WSNs) have evolved substantially over the past two decades. Originally, researchers developed highly specialized networking solutions for WSNs, under the assumption that “sensor networks have different enough requirements to at least

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SenSys '18, November 4–7, 2018, Shenzhen, China

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5952-8/18/11.

<https://doi.org/10.1145/3274783.3275196>

warrant re-considering the overall structure of applications and services” [2]. These solutions were generally driven by the strict resource constraints of motes of that era, which had at most a few kilobytes of data memory (RAM) [3, 4].

Since that time, WSN hardware has become more capable. With the TelosB [7], motes became powerful enough to comfortably run a full-fledged IP stack. Researchers observed that there is substantial benefit to using IP in wireless sensor networks [5]. Since then, IPv6 and 6LoWPAN have become commonplace in WSNs. Operating systems for WSNs, including TinyOS, Contiki, and RIOT, provide implementations of these protocols. Recently, an industry consortium has formed around 6LoWPAN-based IPv6 interoperability (the Thread Group). The integration of IP into commercial offerings is bringing forth the Internet of Things.

2 FULL-SCALE TCP

Despite the rise of IP, TCP is not well accepted by the WSN research community. Many embedded network stacks (e.g., OpenThread) do not even provide a TCP implementation. Those that do (e.g., Contiki’s uIP) provide a standards-compliant but highly simplified implementation of TCP. This is not surprising, as the WSN research community perceives TCP as too heavyweight for embedded WSN hardware and a poor match to WSNs in general.

As part of our research, we refactor the fully-featured implementation of TCP in the FreeBSD Operating System to work within the constraints of Hamilton [1, 6], a modern WSN platform. Our implementation, which we call *TCPlp*, establishes that WSN hardware has crossed a critical resource threshold, and that low-cost, low-power embedded hardware is now capable of running complex network protocols such as full-scale TCP. The feasibility of full-scale TCP on WSN hardware motivates the following three research questions:

- **Does the “completeness” of *TCPlp*—or, stated differently, the presence of TCP features previously found only in traditional operating systems such as Windows, Linux, and FreeBSD—bring value in WSNs?** It is plausible that delayed acknowledgments help reduce power consumption and that selective acknowledgments help overcome high loss rates. The robustness of full-scale TCP—in other words, the fact that the FreeBSD implementation has received decades of testing and tuning—could also be an asset to WSNs.
- **How different are the underlying technical tradeoffs of TCP from those of WSN-specific protocols?** For example, a TCP sender maintains a sliding window of unacknowledged

