

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].

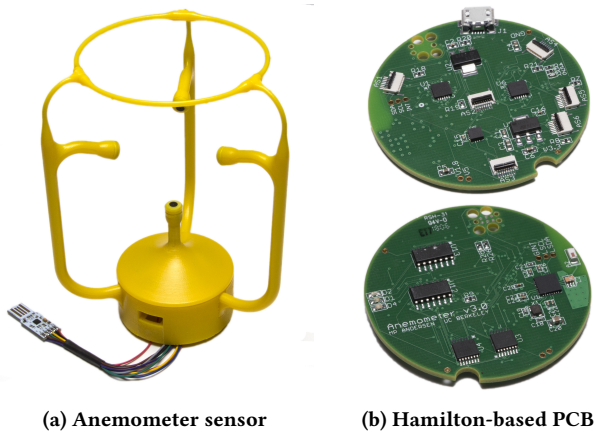
Since that time, WSN hardware has become more capable. With the TelosB [7], motes became powerful enough to comfortably run a full-fledged IP stack. Researchers observed that there is substantial benefit to using IP in wireless sensor networks [5]. Since then, IPv6 and 6LoWPAN have become commonplace 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



(a) Anemometer sensor

(b) Hamilton-based PCB

Figure 1: Hamilton-based ultrasonic anemometer

data, whereas CoAP [8], a WSN-specific alternative to TCP, uses a stop-and-wait policy. How consequential is this difference to these protocols' operation in WSNs?

- **To what extent is the prevailing view that TCP is ill-suited to WSNs actually valid?** Which WSN applications, if any, stand to benefit from using TCP? How does TCP compare to other WSN-specific protocols currently in use? The presence of a full-scale TCP implementation for WSNs allows us to perform experiments to understand which applications may benefit from TCP.

We believe that *TCP_{lp}* will be a useful vehicle for answering these questions and others.

3 APPLICATION SCENARIO

Our demonstration explores the diverse set of WSN applications that may benefit from full-scale TCP. One such application is the collection of data from anemometers, sensors that measure air velocity (Figure 1). Anemometers may be deployed in air ducts in a building to diagnose problems with the Heating, Ventilation, and Cooling System. Furthermore, they produce data at a high sample rate, and each individual sample is quite large (12 measurements). Given its high bandwidth requirements, we consider the anemometer a WSN node that would benefit from full-scale TCP.

Other applications that may benefit from TCP are condition-based maintenance (which could be done using, for example, accelerometer data from a Hamilton mote) or acoustic detection. Even “simple” WSN applications, such as low frequency environmental monitoring, may benefit from full-scale TCP.

4 DEMONSTRATION

Our demonstration consists of three parts:

- (1) One or more Hamilton devices transmit data to a server via a border router. A display shows collected data in real time.
- (2) One or more Hamilton devices run web servers, allowing a participant to use a web browser to see recent sensor readings.
- (3) One or more Hamilton devices run interactive shells, allowing a participant to connect over TCP and run commands for configuration or debugging.

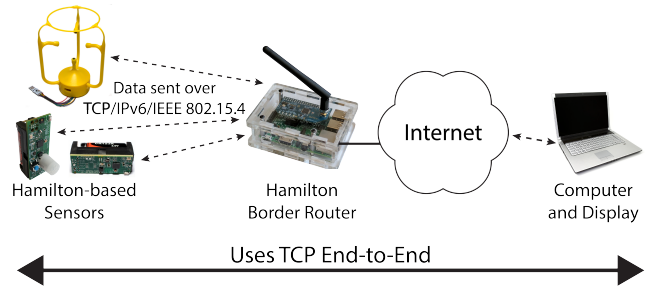


Figure 2: Demonstration setup

The devices form a network using the OpenThread implementation of the Thread routing protocol. Participants may affect sensor readings by interacting with the sensors and see the results in real time. Participants may also connect to a web server hosted on a Hamilton using a web browser, and connect to a shell running on a Hamilton using the `telnet` command line utility.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1752814. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. This research is also supported in part by the Department of Energy Grant No. DE-EE0007685, California Energy Commission, Intel Corporation, National Science Foundation Grant No. CPS-1239552, Fulbright Scholarship Program, and University of California, Berkeley.

REFERENCES

- [1] Michael P Andersen, Hyung-Sin Kim, and David E. Culler. 2017. Hamilton: A Cost-effective, Low Power Networked Sensor for Indoor Environment Monitoring. In *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys '17)*. ACM, Article 36, 2 pages. <https://doi.org/10.1145/3137133.3141453>
- [2] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. 1999. Next Century Challenges: Scalable Coordination in Sensor Networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*. ACM, 263–270. <https://doi.org/10.1145/313451.313556>
- [3] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. 2000. System Architecture Directions for Networked Sensors. In *Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS IX)*. ACM, 93–104. <https://doi.org/10.1145/378993.379006>
- [4] J. L. Hill and D. E. Culler. 2002. Mica: a wireless platform for deeply embedded networks. *IEEE Micro* 22, 6 (Nov 2002), 12–24. <https://doi.org/10.1109/MM.2002.1134340>
- [5] Jonathan W. Hui and David E. Culler. 2008. IP is Dead, Long Live IP for Wireless Sensor Networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08)*. ACM, 15–28. <https://doi.org/10.1145/1460412.1460415>
- [6] Hyung-Sin Kim, Michael P Andersen, Kaifei Chen, Sam Kumar, William J. Zhao, Kevin Ma, and David E. Culler. 2018. System Architecture Directions for Post-Soc/32-bit Networked Sensors. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. ACM.
- [7] J. Polastre, R. Szewczyk, and D. Culler. 2005. Telos: enabling ultra-low power wireless research. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005*. 364–369. <https://doi.org/10.1109/IPSNS.2005.1440950>
- [8] Z. Shelby, K. Hartke, and C. Bormann. 2014. *The Constrained Application Protocol (CoAP)*. RFC 7252. RFC Editor. 1–112 pages. <https://www.rfc-editor.org/rfc/rfc7252.txt>