## Poster Abstract:

# Shipping Data from Heterogeneous Protocols on Packet Train

Chieh-Jan Mike Liang
Microsoft Research Asia
5 Dan Ling Street
Beijing, 100080 China
liang.mike@microsoft.com

Kaifei Chen
Microsoft Research Asia
5 Dan Ling Street
Beijing, 100080 China
v-kaiche@microsoft.com

Jie Liu
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
liuj@microsoft.com

Nissanka Bodhi Priyantha
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
bodhip@microsoft.com

Feng Zhao
Microsoft Research Asia
5 Dan Ling Street
Beijing, 100080 China
zhao@microsoft.com

## ABSTRACT

The maturity and availability of network protocols have enabled wireless sensor networks (WSN) designers to build heterogeneous applications by composing different protocols. A common heterogeneous application combines data collection and dissemination for environmental monitoring with node retasking. While these co-located protocols on the same node have different goals, many of them share requirements and characteristics. Examples of commonalities include the use of bi-directional traffic for reliable transmissions and tree for packet routing. This work explores how the MAC layer can reduce the network transmission overhead of heterogeneous applications by taking advantage of protocol commonalities to aggregate outgoing packets. In other words, this aggregation creates a train of packets destined to the same receiver. Finally, we discuss a strawman implementation of packet train and how our data center monitoring deployment leverages it.

## Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols

## General Terms

Design, Performance, Experimentation

## Keywords

Link protocols, MAC protocols, Wireless sensor networks

## 1. INTRODUCTION

After a decade of active research, the wireless sensor networks (WSN) community has developed and published many open-source implementations of networking protocols. These protocols fall under different categories according to their objectives (e.g., data collection vs. dissemination), delivery latency requirement (e.g., real-time vs. delay-tolerant), energy requirement (e.g., duty-cycled vs. always-on) and so on. One observation from these work is the difficulty in architecting a single protocol that can address all application requirements. Therefore, one common practice in developing WSN applications is composing multiple network protocols to form the application logic. In other words, these protocols co-locate on the same node and result in a heterogeneous application. For example, one could combine data collection and dissemination protocols for an environmental monitoring application with code retasking [1, 3, 8].

Previous work have looked at various aspects of building such heterogeneous WSN applications: reducing the complexity in development [6], minimizing the interference among multiple co-located protocols [5], optimizing the node energy costs [2] and so on.

This work aims to improve the link latency and bandwidth usage by aggregating and scheduling outgoing packets with common parameters to create a packet train. Our contributions are identifying the advantages and challenges in integrating packet train, and presenting a real-world use of packet train in a data center monitoring WSN deployment.


*IPSN'12,* April 16–20, 2012, Beijing, China.
ACM 978-1-4503-1227-1/12/04.

## 2. PACKET TRAIN

Section 1 uses the trend of heterogeneous applications in WSN deployments to motivate our interests on packet train. This section first describes the idea of packet train, and argues that packet train can improve the network performance by reducing various overheads. Then, we present design considerations for implementing packet train.

Packet train builds on the idea of piggy-backing small control packets onto data packets. TCP is an example that piggybacks packet acknowledgment onto data packets [7]. In addition to merging packets, packet train can also treat packets with common parameters as a group and send these packet back-to-back. Effectively, this group of packets can be considered as one "mega" packet, and the transmission costs would be amortized over these sub-packets. While common parameters can include the destination address, radio channel and so on, this paper focuses on the destination address. As one requirement of packet train is the global view on outgoing packets, we propose that packet train should be a MAC-layer feature.

**Advantages.** Packet train can reduce the transmission over-

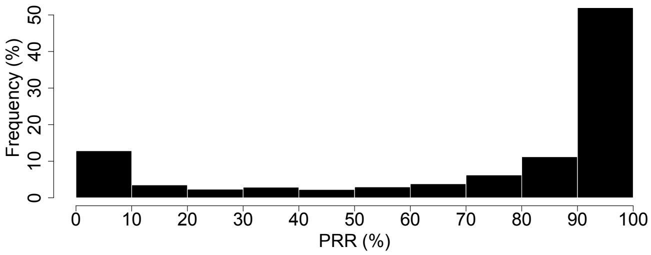


**Figure 1: Packet reception ratios (PRR) across all the links from a 52-node data center site survey. Many links have a PRR above 70%**

head in both latency and bandwidth. First, most WSN deployments rely on CSMA to resolve contentions among concurrent senders. However, CSMA incurs overhead due to the backoff and wait times. By sending packets destined to the same receiver back-to-back, the sender can simply perform CSMA once at the beginning of the packet train. In fact, the sender also takes advantage of the short-term link stability and transmits packets in burst. Second, a typical WSN packet consists of preambles, header and footer, in addition to the data payload. The preambles, header and foot are control information that do not carry application data, and thus lower the goodput. For example, in TinyOS, 802.15.4 packet preambles, header and footer occupy about 15% of the space with respect to the maximum-sized packet. Considering the case of multiple small data payloads, packet train can merge them in one packet to share one set of preambles, header, and footer. Third, packet train can amortize the energy expenditure across co-located protocols. Dunkels et al. presented the case where aggregating broadcast beacons of co-located duty-cycling protocols can significantly reduce energy costs incurred on each beacon [2].

**Design Considerations.** Packet train breaks some of the radio and protocol assumptions on the packet structure and delivery. We discuss the implications and design considerations in implementing packet train. First, packet train breaks most radio PHYs' assumption that one incoming radio frame represents a single packet. Therefore, certain radio features may not always work. For example, if a packet contains two payloads where one requires acknowledgment, the radio PHY might not be able to look at all embedded headers to trigger the hardware acknowledgment properly. Second concern is the protocol fairness. As the MAC-layer shuffles the outgoing queue to aggregate outgoing packets with common parameters, the order of packet transmission is not the same as the order of submission from protocols. This reordering can delay pending packets that do not have common parameters with previous pending packets in the queue. Finally, merging payloads increases the packet size. The networking community has shown that the larger the packet size the higher the chance of packet corruption. Therefore, packet train may not be suitable under all conditions.

# 3. IMPLEMENTATION

We have integrated packet train with our data center monitoring application, MeshNet (based on our previous work, RACNet [4]). MeshNet is a heterogeneous deployment of upstream QoS-aware data collection and downstream selective dissemination protocols. Figure 1 shows that a larger percentage of links in our network have a fairly good packet reception ratios (PRR). This represents opportunities for nodes to take advantage of a larger network MTU, and short-term link stability. MeshNet uses packet train in the following two ways.

All co-located protocols rely on acknowledgments for packet delivery reliability. Since acknowledgment packets are small, they are suitable candidate for merging with other data packets or acknowledgment packets. To improve the chance of merging, we delay low-priority data packets in the outgoing queue for 100 ms. And, before transmitting an acknowledgment packet, the MAC layer scans through the MAC layer buffer for packets with the same destination address. As upstream data traffic dominate in MeshNet, we expect that the acknowledgment packets for different upstream packets can often be merged. Another opportunity for packet train is to merge acknowledgment packets with downstream protocol traffic, similar to TCP.

MeshNet also actively groups data packets destined to the same receiver. Experiment results suggest that the different requirements in end-to-end delivery latency can result in outgoing packets being queued, which in turn increases the opportunities for packet train.

# 4. REFERENCES


[1] G. W. Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and Yield in a Volcano Monitoring Sensor Network. In *OSDI*, 2006.

[2] A. Dunkels, L. Mottola, N. Tsiftes, F. Osterlind, J. Eriksson, and N. Finne. The Announcement Layer: Beacon Coordination for the Sensornet Stack. In *EWSN*, 2011.

[3] K. Langendoen, A. Baggio, and O. Visser. Murphy Loves Potatoes: Experiences from a Pilot Sensor Network Deployment in Precision Agriculture. In *WPDRTS*, 2006.

[4] C.-J. M. Liang, J. Liu, L. Luo, A. Terzis, and F. Zhao. RACNet: A High-Fidelity Data Center Sensing Network. In *SenSys*, 2009.

[5] C.-J. M. Liang and A. Terzis. Rethinking Multi-Channel Protocols in Wireless Sensor Networks. In *HotEmNets*, 2010.

[6] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica. A Unifying Link Abstraction for Wireless Sensor Networks. In *SenSys*, 2005.

[7] A. Tanenbaum. *Computer Networks*. Prentice Hall Professional Technical Reference, fourth edition, 2002.

[8] G. Tolle and D. Culler. Design of an Application-Cooperative Management System for Wireless Sensor Networks. In *EWSN*, 2005.