Fair Waiting Protocol: Fairness and Isolation in Wireless Sensornets

Jung Il Choi[1], *Jung Woo Lee*[1], *Zhe Chen*[2], *and Philip Levis*[1]
[1]Computer Systems Laboratory, Stanford University
[2]Computer Engineering, Columbia University

1. Introduction

Isolation simplifies reasoning. For example, isolation between processes in an OS can make failures due to another program exceedingly rare. In a network architecture, isolation between network protocols can make failures due to another traffic pattern similarly rare.

The main goal of the Fair Waiting Protocol (FWP) is to isolate protocols on CSMA networks, such that protocols do not interfere with each other's operation. This problem is unique to wireless sensornets because they employ multiple layer 3 protocols such as collection, routing, dissemination, and synchronization. The key insight behind FWP is that layer 3 protocol isolation requires inter-protocol collision avoidance. For example, Deluge sends flurries of data packets using single-hop broadcasts, forming a region of intense interference. If other protocols suffer failures during the bursts, determining the cause of the failure can be very complicated because the failure is not caused by the protocol itself.

In Section 2, we will explain the mechanism of grant-to-send, a core algorithm of FWP, and its performance on collision avoidance. In Section 3, we show how the collision avoidance mechanism can be extended to provide inter-protocol collision avoidance to achieve protocol isolation.

2. Mechanism

FWP aims to utilize layer 3 information to avoid collisions across multiple hops of a data flow. Figure 1 shows a simple scenario of packet loss due to an intra-path collision. In order to avoid collision, nodes must wait until their grandparents forward the previous packet [3, 5].

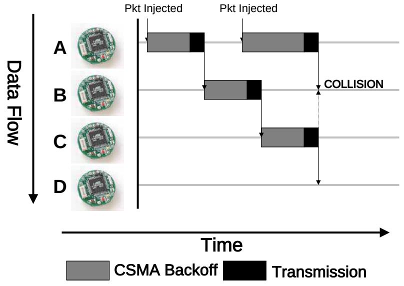


Figure 1: **An example scenario of intra-path collision. Solid lines are received packets, dashed lines are overheard packets. When node A and its grandparent send packets, collision occurs at node B.**

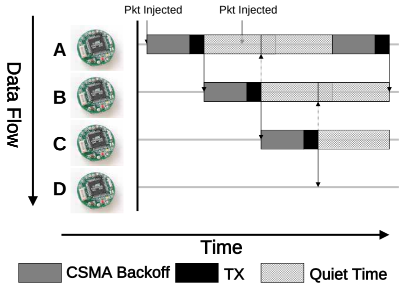


Figure 2: **Grant-to-send mechanism example. Solid lines are received packets, dashed lines are overheard packets. With grant-to-send mechanism, A sends a packet to B with a nonzero grant-to-send. A, the transmitter, must be quiet for the duration of the grant-to-send, but B, the receiver, is not suppressed. When B sends to C, both B and A are suppressed. A must wait until both grants have expired. Thus even if A has a packet to send, the transmission is suppressed while previous packets exit the collision range. With this mechanism, FWP aims to clear the channel for the receiver and the protocol which it selects to send.**

FWP prevents the packet collisions in Fig. 1 using a *grant-to-send* mechanism. A FWP transmission request includes a grant-to-send value along the packet to transmit. FWP puts this value in a packet as a one byte header. A grant-to-send is a quiet time during which only the recipient of the packet may transmit. During this quiet time, other nodes that overheard or transmitted the packet may not use the channel. Thus a transmission *grants* the channel around the transmitter for the recipient *to send*. Figure 2 shows an example of FWP operating across a route. Although packets are injected at the same time as in Fig. 1, B's grant to C forces A to wait, preventing interference along the path. When all quiet times expire, FWP submits a packet to the underlying CSMA.

In practice, FWP does not prevent all packet collisions. Since FWP makes transmission decision based on past transmissions, it only prevents collision on the tails of forwarding flows. RTS/CTS can be a perfect solution for collision avoidance. However, RTS/CTS does not easily support broadcasts, a common primitive in sensornet protocols such as Deluge. Furthermore, for the small datagrams typical of sensor netowrks, control overhead of RTS/CTS can be large. In contrast, FWP preserves all the flexibilities of CSMA with only a small overhead of one byte per packet.

Figure 3 shows TOSSIM simulation results for a TCP-like reliable transport protocol running on a simple 7 node

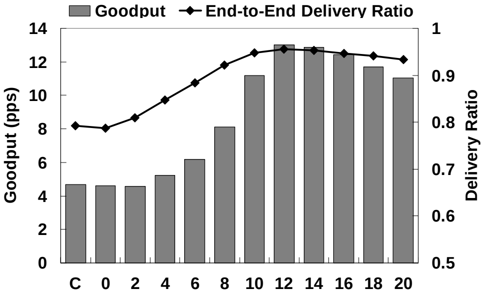


Figure 3: **The effects of FWP on TCP Performance on 7-node chain topology of TOSSIM. The x-axis indicates the length of grant-to-send values, with 'C' indicating bare CSMA. FWP achieves 280% gain on goodput due to its delivery reliability.**

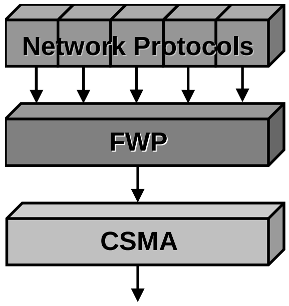


Figure 4: **FWP sits between network protocols and a CSMA MAC.**

line topology. In this simulation, the SNR on all links was high enough that only collisions caused packet losses. FWP achieves 2.8 times the goodput of CSMA when the quiet time is 12ms, which is approximately maximum packet time. Since FWP effectively prevents packet losses, TCP is able to maintain a higher send rate.

3. Protocol Isolation

The previous section has shown that the grant-to-send can be an effective way for collision avoidance. However, even if every protocol introduces waits between its packets to prevent self-intereference, the mechanism is no longer effective if one protocol cannot suppress other protocols. Correctly preventing interference across protocols requires a shared mechanism between them. FWP enables layer 3 protocols to share information by placing an additional collision avoidance layer between layers 2 and 3 as in Figure 4.

Figure 5 shows how FWP's isolation affects the performance of two collection protocols running on the 165-node motelab testbed [4]. The figure shows the cost of delivering packets with TinyOS 2.0's Collection Tree Protocol (CTP) [2] running over plain CSMA and over FWP. In the FWP experiment, CTP data packets have a quiet time of one packet time and CTP routing beacons have a quiet time of zero. While both handle a single instance of CTP well, two instances of CTP running over bare CSMA interfere with each other heavily. This is because CTP has built-in rate-limiting mechanisms that prevent self-interference, but these methods are ineffective when another protocol is simultaneously using the channel. FWP isolates the two instances of the protocol, resulting in lower packet delivery costs. Its suppression mechanism enforces rate-limiting across protocols, limiting the sending rate to what the network can handle. Because FWP drastically reduces the effects of protocols on one another, it simplifies debugging and makes identifying causes of failure easier.

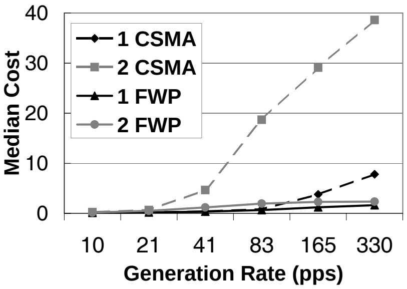


Figure 5: **Median packet delivery costs - retransmissions per successful transmission - for 1 and 2 instances of CTP running on bare CSMA and over FWP on 165-node network. FWP effectively isolates the two instances from each other to reduce packet retransmissions.**

Collision avoidance alone, however, does not provide isolation. If a protocol sends a burst of packets with maximum quiet time, other protocols would suffer suppression indefinitely. Preventing starvation requires fairness across protocols. FWP adopts fair queueing defined by Demers et al [1]. As the grant durations can be viewed as a channel occupation, FWP keeps track of channel usage of protocols by adding quiet times and air times. When multiple transmission requests are submitted to FWP, it selects the packet with the least channel usage. Therefore, FWP penalizes protocols with more packets or larger quiet times to give priority for protocols that have occupied the channel less.

Unlike programming abstractions such as SP, FWP is a protocol and is therefore OS, language, and platform-independent. While our current implementation is for the CC2420 radio under TinyOS 2.0, we do not foresee challenges porting it to other OSes or CSMA layers.

4. REFERENCES

[1] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *SIGCOMM '89: Symposium proceedings on Communications architectures & protocols*, pages 1–12, New York, NY, USA, 1989. ACM Press.

[2] R. Fonseca, O. Gnawali, K. Jamieson, S. Kim, P. Levis, and A. Woo. TEP 123: Collection Tree Protocol. http://www.tinyos.net/tinyos-2.x/doc/.

[3] T. Moscibroda, R. Wattenhofer, and Y. Weber. Protocol design beyond graph-based models. In *Proceedings of the 5th ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets)*, 2006.

[4] G. Werner-Allen, P. Swieskowski, and M. Welsh. Motelab: a wireless sensor network testbed. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 68, Piscataway, NJ, USA, 2005. IEEE Press.

[5] A. Woo and D. E. Culler. A transmission control scheme for media access in sensor networks. In *Proceedings of the seventh annual international conference on Mobile computing and networking*, Rome, Italy, July 2001.

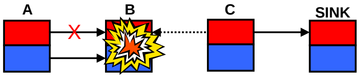
Fair Waiting Protocol: Fairness and Isolation in Wireless Sensornets

Jung Il Choi[1], Jung Woo Lee[1], Zhe Chen[2], and Philip Levis[1]

[1]Computer Systems Laboratory, Stanford University
[2]Computer Engineering , Columbia University

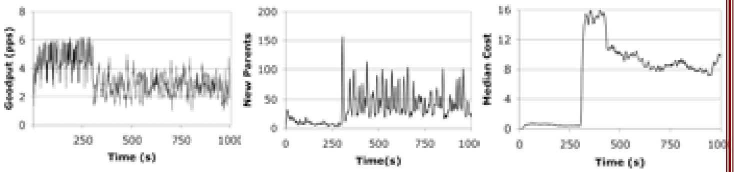
Problem Formulation

- ❖ Sensornets typically use multiple multihop network protocols – collection, dissemination, routing
- ❖ While reasonable network protocols avoids self-interference, they are still vulnerable to **inter-protocol interference**



Inter-protocol Interference

- ❖ On 165-node testbed, Deluge(dissemination) starts 5 minutes after two instances of CTP(collection) starts

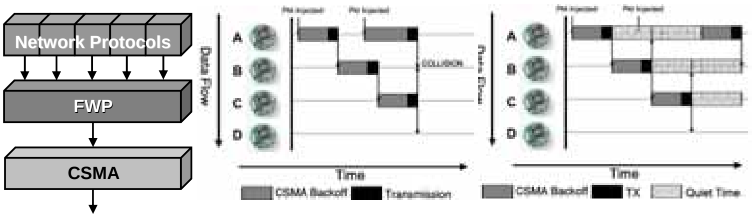


- ❖ No fault of protocols: inevitable with current net architecture
- ❖ Not only degrades performance, complex interactions between protocols makes system extremely complicated to understand

Isolation and Fairness

- ❖ The two key properties to avoid inter-protocol interference
- ❖ **Isolation**
  - ❖ Only one protocol should access the channel at a time
- ❖ **Fairness**
  - ❖ Isolation is insufficient – the simplest approach is to let only one protocol operate
  - ❖ Every protocol should receive its fair share

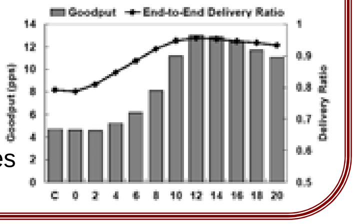
The Fair Waiting Protocol (FWP)



- ❖ Utilizes layer 3 information to estimate next transmission
- ❖ Sits between network protocols and CSMA to control when to transmit which packets.
- ❖ Isolation – ***Grant-to-Send*** Mechanism
  - ❖ Post-transmission period during which sender and overhearer should be quiet
  - ❖ Recipient is the sole user of the channel around the transmitter
- ❖ Fairness – Basic Fair Queueing Algorithm by Demers et al.
  - ❖ Uses grant durations and packet transmission times to estimate channel occupancy
  - ❖ When multiple packets in queue, FWP selects protocol with the least channel occupancy time

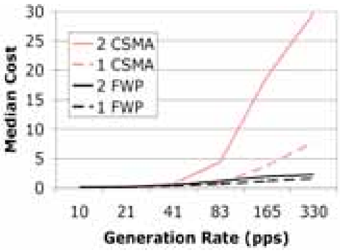
FWP Performance on TOSSIM

- ❖ TCP-like reliable transport protocol on 7-node chain topology
- ❖ SNR is high enough that only collisions caused packet losses
- ❖ FWP achieves 2.8 times the goodput of CSMA
- ❖ TCP is able to maintain higher send rate because FWP effectively prevents packet losses

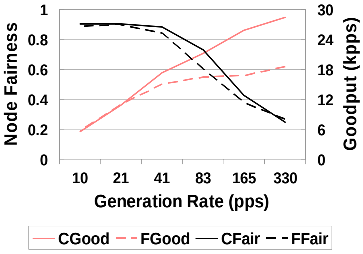


FWP Performance on Real World

- ❖ CTP (collection tree protocol) and Deluge (dissemination)
- ❖ Tested using all 165 operable node on the Motelab testbed
- ❖ Scenario 1 : Runs one/two instances of CTP on pure CSMA and FWP+CSMA

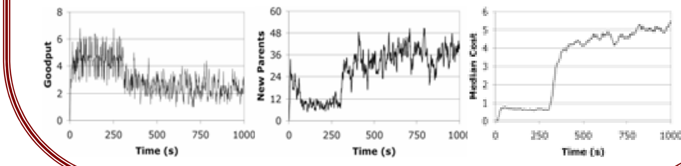


Median Cost = # of reTX per successful TX



- ❖ While FWP and CSMA behaves similarly under light traffic, FWP keeps CTP efficient under heavier traffic
- ❖ On CSMA, 2CTP case shows much higher delivery cost than 1CTP even though with the same generation rate
- ❖ This benefit comes from the sacrifice on goodput, because FWP limits traffic rates to prevent collisions.

- ❖ Scenario 2 : Deluge starts operation 5 minutes after two Instances of CTP start on FWP with generation rate of 10 pps
  - ❖ More efficient and robust operation on FWP



Conclusion and Discussion

- ❖ FWP helps avoiding inter-protocol interference by a small overhead of one byte per packet, while preserving the flexibilities of CSMA
- ❖ Inter-protocol interference avoidance cuts the interactions between protocols, which enhances manageability and visibility of the system
- ❖ FWP is a very simple protocol that it can be applied to any OSes, lanuages, and platforms.