Poster Abstract: Exploiting the Capture Effect for Low-Latency Flooding in Wireless Sensor Networks

Jiakang Lu and Kamin Whitehouse
Department of Computer Science, University of Virginia
Charlottesville, VA, USA
{jklu,whitehouse}@cs.virginia.edu

ABSTRACT
In this paper, we present the Flash flooding protocol that exploits the capture effect to produce low-latency network floods. The capture effect is the ability of some radios to correctly receive one of several concurrently transmitted messages, even if the received strengths of the two messages are almost the same. We exploit this phenomenon in a network flooding scenario by allowing nodes to propagate the flooding message concurrently, thus reducing delays due to neighborhood contention. Our experimental results indicate that Flash can reduce latency by 75-80 percent without sacrificing flooding reliability or coverage.

Categories and Subject Descriptors: C.2.2 [Computer-Communication Networks]: Network Protocols

General Terms: Design, Experimentation, Performance

Keywords: Capture Effect, Flooding, Latency, Wireless Sensor Networks

1. INTRODUCTION
Network floods are common and important operations that are at the heart of most wireless sensor network algorithms and applications. They are the starting point for code and data dissemination, the creation of routing trees, clock synchronization, node localization, and group formation. However, network floods are costly in terms of latency. In most cases, whenever a vertex cover cannot be identified in advance, the flood must be propagated by all nodes in order to ensure complete and reliable coverage of the network. Thus, a main cause of flooding latency is neighborhood contention: nodes cannot propagate the flood immediately because they must wait for neighboring nodes that are also trying to propagate the flood. This problem is exacerbated in low-duty cycle networks where protocols like LPL [1] and X-MAC [2] send extremely long packets.

In this paper, we explore several ways to improve flooding latency by exploiting the capture effect. The capture effect is the ability of some radios to correctly receive one of several concurrently transmitted messages, even if the received strengths of the two messages are almost the same [3]. We exploit this phenomenon in a network flooding scenario by allowing nodes to propagate a flooding message concurrently which reduces delays due to neighborhood contention. To address the issue that too many concurrent transmissions result in packet loss despite the capture effect [4], we develop techniques to automatically control the number of concurrent transmissions and to restart floods that have halted due to collisions and packet loss. We call the combination of these techniques the Flash flooding protocol.

2. Flash flooding protocol
The Flash flooding protocol exploits the capture effect to reduce flooding latency by eliminating neighborhood contention without sacrificing coverage or reliability. We explore Flash in two fundamental network scenarios in wireless sensor networks: high-duty cycle networks where all of the nodes are always on and low-duty cycle networks where nodes sleep most of the time and wake up only periodically to see if a message is being sent. Flash provides flexible choices, based on different application criteria, to achieve low-latency flooding.

2.1 Flash-I: Complete Concurrency
Carrier sense is a mechanism common to almost all modern wireless communication stacks for avoiding packet collisions. Before sending a message, the sender performs a clear channel assessment (CCA) and if the channel is clear, the sender transmits immediately; otherwise, it waits for a random period of time before trying to send again. However, carrier sense may be too conservative in the presence of capture effect. In a network flood when all nodes want to send the same packet, this can result in unnecessary neighborhood contention which slows the flood. We present a new algorithm called Flash-I that completely removes the MAC delay and CCA in order to decrease the delay due to neighborhood contention.

Our testbed experiment results indicate that Flash-I, without any neighborhood contention, will complete much more quickly than a flood using traditional CSMA. However, by removing MAC delay and CCA, Flash-I will also likely experience more packet collisions and the flood may not reach all nodes. For a thorough comparison of these two protocols, we run the Flash-I algorithm in a high-duty cycle, large-scale 4×4 scaled testbed network with 768 nodes. Flash-I usually covers over 95 percent of the network, but full coverage is almost never achieved. These results show that Flash-I is useful in applications where latency is the primary concern and full coverage is not critical.

2.2 Flash-II: Restarting the Flood
In some applications, it is not acceptable to flood to only 95 percent of the nodes when the network is fully connected.
In order to improve coverage, we present a new algorithm Flash-II that is a simple variant of Flash-I: we follow the first packet with another packet with MAC delay and CCA. This is equivalent to all nodes rebroadcasting the flooding message within one-hop neighborhood after the first flash of Flash-I. The main goal of the second phase after Flash-I is to restart a flood that has stopped due to collisions and ensure reception by any nodes that missed the first round.

Following the same experimental setup as the experiments in Section 2.1, the results show that Flash-II can maintain similar latencies as Flash-I for most nodes and also achieve coverage at least as good as CSMA floods. To more thoroughly evaluate the performance of Flash-II, we evaluate it at larger scales and higher densities in simulation. The results show that Flash-II is nearly 75 percent faster than CSMA flooding in high-duty cycle networks. These results are consistent across multiple network sizes and densities.

2.3 Flash-III: Sensing Concurrency

The capture effect embraces more simultaneous transmissions, but a fine balance must be achieved: being too aggressive with concurrency will result in collisions and packet loss while being too conservative will result in latency due to neighborhood contention. This is particularly important for a low-duty cycle network, in which MAC protocols such as X-MAC transmit for a long enough period so that the receiver is guaranteed to wake up at least once and notice the transmission. However, the long length of the packet train will multiply the delays due to neighborhood contention and increase packet collisions caused by more concurrent transmissions.

In order to address this problem, Flash-III applies a new technique to sense the amount of transmission concurrency in a network. First, we introduce a small interpacket spacing (IPS) between packets in the packet train. Second, we introduce a very small CCA before the packet train is sent. In particular, the CCA time in Flash-III is much smaller than the interpacket spacing. The key idea behind Flash-III is that a node is unlikely to detect packet train activity in periods of low concurrency and the opposite will occur in periods of high concurrency. Therefore, we can control the maximum degree of concurrency through a parameter $\alpha$ defined to be the ratio of the CCA and the IPS: $\alpha = \frac{CCA}{IPS}$. When $\alpha$ is small, the nodes are very likely to transmit and a high degree of concurrency will be achieved. Thus, Flash-III is essentially the same as X-MAC with a value $\alpha = 1$ and the same as low-duty cycle Flash-I when $\alpha = 0$.

3. IMPLEMENTATION AND EVALUATION

To explore the capture effect in a realistic environment, we deployed the VineLab wireless sensor network testbed with 48 Tmote-sky motes deployed in Ohlson Hall, the Computer Science building at our University. We collect a Capture Map, the empirical data that indicates, for each node in the testbed, which nodes will receive a message when any combination of its radio neighbors is transmitting simultaneously. With the transmission power -10dBm, our VineLab testbed has an average node degree of 7.3 and there exists 19,956 transmitter set cases on VineLab in total. By employing empirical traces and statistics using the capture map, we have developed a new capture-aware simulation framework that represents the capture dynamics in our testbed. The fidelity of our simulator is verified by comparing the results on the real testbed with experiments ran on the simulator. For fairness, the same parameters are used in the simulation and the empirical experiments. The simulator allows us to evaluate different solutions as scales and densities that are not feasible on the physical testbed. Figure 1 is an example of simulation over multiple densities. The results indicate that Flash-III can wake up low-duty cycle networks more than 80 percent faster than X-MAC flooding as the network density increases.

![Figure 1: Simulation result of Latency vs. Network Density in low-duty cycle networks where sleep interval is 1000 milliseconds.](image)

4. CONCLUSIONS

We presented a flexible flooding protocol that exploits the capture effect to produce low-latency floods in wireless sensor networks. The Flash protocol effectively provides flexible choices in design space based on different system criteria and tradeoffs. Capture effect has proved to be prevalent in wireless communication and we believe the core idea of Flash can be applied across many applications and bring significant performance improvement. Our empirical study of network-wide capture dynamics and novel simulation framework based on capture map will also inspire new studies on capture and open a new door for capture-aware algorithms and applications in the future.

5. REFERENCES


