High-Throughput, Reliable Multicast without "Crying Babies" in Wireless Mesh Networks

Dimitrios Koutsonikolas, Y. Charlie Hu and Chih-Chun Wang
Purdue University, West Lafayette, IN, USA

ABSTRACT

There are two primary challenges to supporting high-throughput, reliable multicast in wireless mesh networks (WMNs). The first is no different from unicast: wireless links are inherently lossy due to varying channel conditions and interference. The second, known as the "crying baby" problem, is unique to multicast: the multicast source may have varying throughput to different multicast receivers, and hence trying to satisfy the reliability requirement for poorly connected receivers can potentially result in performance degradation for the rest of the receivers.

In this paper, we propose *Pacifier*, a new high-throughput reliable multicast protocol. *Pacifier* seamlessly integrates four building blocks, namely, *tree-based opportunistic routing, intra-flow network coding, source rate limiting, and round-robin batching,* to support high-throughput, reliable multicast routing in WMNs, while at the same time effectively addresses the "crying baby" problem.

1. MOTIVATION

In contrast to unicast routing, high-throughput, reliable multicast routing has received relatively little attention. Reliable multicast routing has many important applications in WMNs, such as software updates and video/audio file downloads. These applications have a strict requirement of **100% Packet Delivery Ratio (PDR)**. This requirement makes many of the reliable multicast protocols proposed in the past inappropriate, since they cannot guarantee 100% PDR. In addition, reliability for this class of applications cannot come at the cost of significantly reduced throughput, since Internet users always desire fast downloads.

The fundamental challenge in achieving reliable multicast in WMNs is no different from that of reliable unicast – that wireless links are lossy. To overcome this, researchers have applied classic techniques such as Automatic Repeat reQuest (ARQ), Forward Error Correction (FEC), or combinations of the two. More recently, researchers have applied network coding (NC), a technique originally developed for the wireline Internet, to overcome the above challenge. NC can be viewed as a technique equivalent to performing hop-by-hop FEC, without the delay penalty incurred by the decoding operations at each hop, that would be required by hop-by-hop FEC. Practical work that exploits the idea of utilizing NC for reliable multicast is still at a preliminary stage. MORE [1] is the *only practical* NC-based protocol that supports high-throughput, reliable multicast.

A second fundamental challenge in reliable multicast, which is unique to multicast, is the "crying baby" problem as first pointed out in [2] in the context of multicast in the Internet. If one receiver has a particularly poor connection, then trying to satisfy the reliability requirement for that receiver may result in performance degradation for the rest of the receivers.

In this paper, we propose *Pacifier*, the *first high-throughput, reliable multicast protocol that simultaneously addresses the above two challenges*. *Pacifier* seamlessly integrates four building blocks, namely, *tree-based opportunistic routing, intra-flow NC, source rate limiting, and round-robin batching,* to support high-throughput, reliable multicast routing and at the same time solve the "crying baby" problem.

2. DESIGN

Tree-based Opportunistic Routing The use of opportunistic routing in the form used in MORE is an overkill for multicast for two reasons. First, even for a single destination, having too many nodes acting as FNs can cause congestion. Second, the benefit of overhearing of broadcast transmissions, which is explored by opportunistic routing in MORE, is naturally explored in a fixed multicast tree, where the use of broadcast allows nodes to receive packets not only from their parent in the multicast tree, but also from ancestors or siblings, essentially transforming the tree into a mesh. Hence, *Pacifier* constructs a shortest-ETX tree to connect the source to the destinations and leverages the broadcast nature of the wireless medium to provide path redundancy.

Batching and Coded Forwarding The source and the intermediate FNs in *Pacifier* use *intra-flow* random linear

ACM CoNEXT 2008 Student Workshop, December 9, 2008, Madrid, SPAIN

NC. The hop-by-hop nature of NC requires the source to break a file into small batches of packets so that the packet header overhead, encoding/decoding time, and memory requirements at the intermediate FNs remain low. For each batch, the source sends random linear combinations of the packets belonging to that batch. Intermediate FNs store all the innovative packets of the batch and also send random linear combinations of them.

How many packets does an FN send? Despite the use of a multicast tree for data forwarding, the use of 802.11 broadcast effectively enables opportunistic routing, i.e., a node can opportunistically receive packets from nodes other than its parent in the multicast tree. We derive the number of transmissions each FN needs to make for every packet it receives and define it as the TX_credit for that FN. Each FN node j keeps a credit counter. When it receives a packet from an *upstream* node (defined below), it increments the counter by its TX_credit. When the 802.11 MAC allows the node to transmit, the node broadcasts a coded packet only if the counter is positive, and decrements the counter.

The credit calculation in *Pacifier* is based on the simple principle that in disseminating a packet from the root, each FN in the multicast tree should ensure that each of its child nodes receives the packet *at least once*. Let $A(j)$ denote the set of j's upstream nodes, i.e., nodes closer in ETX distance to the source than node j, and $C(j)$ denote the set of j's child nodes in the multicast tree. The TX_credit of FN j is given by:

$$\text{TX\_credit}_j = \frac{z_j}{\sum_{i \in A(j)} z_i (1 - \epsilon_{ij})} \quad (1)$$

In (1), z_j is the expected number of transmissions that FN j must make in disseminating one packet (from the root) down the multicast tree. z_j is calculated as:

$$z_j = max_{k \in C(j)} \frac{L_{jk}}{1 - \epsilon_{jk}} \quad (2)$$

where L_{jk} is the amount of packets node j actually needs to forward for child k, given by:

$$L_{jk} = min(\sum_{i \in A(j)} z_i (1 - \epsilon_{ij}), 1) - \sum_{i \in A(j)} z_i (1 - \epsilon_{ik}) \quad (3)$$

Source Rate Limiting *Pacifier* applies a simple yet efficient form of backpressure-based rate limiting to the source.[1] The basic idea is to have the source wait until it overhears its child nodes forward the previous packet it sent before it transmits the next packet. Since the number of transmissions by the source z_s has already factored in packet losses to its child nodes, the source does not need to worry about losses of individual transmissions. Instead, the source waits until it overhears a transmission from *any* of its child nodes or until a timeout.

Solving the "Crying Baby" Problem In MORE, the source keeps transmitting packets from the same batch until all the

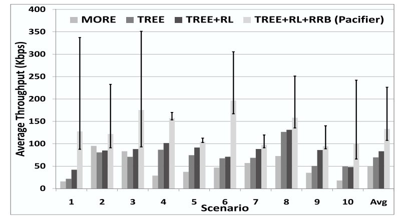

Figure 1: Average, max, and min throughput with MORE, TREE, TREE+RL, and TREE+RL+RRB (*Pacifier*) for 10 different scenarios.

receivers acknowledge that batch. This policy makes the protocol susceptible to the "crying baby" problem, since if the connection to one receiver is poor, it can slow down the rest of the receivers. *Pacifier* applies a practical solution to the problem, which requires no more memory than MORE, i.e., FNs still maintain only one batch at a time in their memory.

The source iteratively sends the batches of a file in a *round-robin* fashion, for as many rounds as required, until it has received acknowledgments of receiving all batches from all the receiver. The source maintains a counter C_{s_i} for each batch i which is equal to the number of remaining packets the source has to transmit for that batch. Each intermediate FN only buffers packets belonging to the current batch. The source sends packets from batch i until either C_{s_i} reaches 0 or it receives acknowledgment for that batch from *one* receiver; it then moves to the next batch for which there are still receivers that have not acknowledged.

3. EVALUATION

We provide now a preliminary evaluation of *Pacifier* in Glomosim simulator, using 50 nodes randomly deployed in a $1000m \times 1000m$ area, 1 multicast group with 10 members, and a realistic signal propagation model (*TwoRay* model with Rayleigh fading). Figure 1 summarizes the average, maximum and minimum throughput of MORE, and different versions of *Pacifier* – TREE (tree-based opportunistic routing), TREE+RL (rate limiting added to the previous version), and TREE+RL+RRB (complete version). We observe that on average, *Pacifier* outperforms TREE+RL, TREE, and MORE by 60%, 90%, and 171%, respectively. In addition, *Pacifier solves the "crying baby" problem*, allowing well-connected receivers to achieve much higher throughput, up to 20x higher than with MORE. Interestingly and importantly, *Pacifier* also *improves throughput of the worst receiver* in all 10 scenarios by up to 4.5x, compared to the other 3 protocols.

Acknowledgment

This work was supported in part by NSF grant CNS-0626703.

4. REFERENCES

[1] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *ACM SIGCOMM*, 2007.

[2] H. W. Holbrook, S. K. Singhal, and D. R. Cheriton. Log-based receiver-reliable multicast for distributed interactive simulation. In *Proc. of ACM SIGCOMM*, 1995.

[1] The use of TX_credits implements a form of rate control for the intermediate FNs.