Adaptive Voice Stream Multicast over Low-power Wireless Networks

Liqun Li[1,2,3], Guoliang Xing[3], Qi Han[4], Limin Sun[1]
[1] Institute of Software, Chinese Academy of Sciences, China
[2] Graduate University of Chinese Academy of Sciences, China
[3] Department of Computer Science and Engineering, Michigan State University, USA
[4] Department of Math and Computer Sciences, Colorado School of Mines, USA

Abstract

Low-power Wireless Networks (LWNs) have become increasingly available for mission-critical applications such as security surveillance and disaster response. In particular, emerging low-power wireless audio platforms provide an economical solution for ad hoc voice communication in emergency scenarios. In this paper, we develop a system called Adaptive Stream Multicast (ASM) for voice communication over multi-hop LWNs. ASM is composed of several novel components specially designed to deliver robust voice quality for multiple sinks in dynamic environments: 1) an empirical model to automatically evaluate the voice quality perceived at sinks based on current network condition; 2) a feedback-based Forward Error Correction scheme where the source can adapt its coding redundancy ratio dynamically in response to the voice quality variation at sinks; 3) a Tree-based Opportunistic Routing (TOR) protocol that fully exploits the broadcast opportunities on a tree based on novel forwarder selection and coordination rules; and 4) a distributed admission control algorithm that ensures the voice quality guarantees when admitting new voice streams. ASM has been implemented on a low-power hardware platform and extensively evaluated through experiments on a testbed of 18 nodes.

1 Introduction

Recently, using Low-power Wireless Networks (LWNs) for emergency response has gained increasing popularity. In particular, the emerging low-power wireless audio platforms (e.g. Crossbow IMB 400 [28], SenEar [2], and Csiro [19]) have enabled the construction of critical ad hoc voice communication networks in many emergency scenarios (*e.g.,* rescuing miners trapped underground [1, 18]). These platforms use small batteries as power sources and are typically equipped with low-power CPUs, low-bandwidth radios, and microphones/speakers. Because of the portability and low cost in comparison with current 802.11 platforms, they can be massively installed in buildings or mines and provide multi-hop voice communication capabilities in emergency scenarios. The feasibility of such systems has been demonstrated by several pilot system prototypes [16, 18]. However, a key shortcoming shown by recent studies is that the number of *concurrent* voice streams that can be supported by multi-hop LWNs is extremely small due to the limited radio bandwidth (typically 50$\sim$500 Kbps). For instance, only one voice stream can be active at any time in networks based on CC2420 radio [18] and typically two or three concurrent streams are supported in networks based on CC1100 radio [16]. Moreover, audio streaming over multi-hop LWNs often results in unsatisfactory voice quality because of significant loss of low-power wireless links. These limitations hinders the application of LWNs in the scenarios where robust voice communication must be maintained among multiple users.

In this paper, we study the problem of multicasting voice streams from a source to *multiple* sinks over multi-hop LWNs. Voice multicast is an essential service in many emergency scenarios (*e.g.,* communication among victims and between victims and rescue workers [1]). We envision that a number of low-power devices are deployed in emergency areas to provide the voice communication service. By taking the advantage of broadcast nature of wireless medium, multicast can enable more concurrent voice streams. However, a key challenge is to simultaneously maintain the quality of multiple voice streams over lossy wireless links. Our testbed experiments show that the end-to-end loss rate of a multi-hop path must be lower than $\sim$13% in order to achieve acceptable voice quality. Moreover, various dynamics such as environmental noise and interference can easily render the voice quality unstable. On the other hand, the existing quality of service (QoS) assurance mechanisms designed for wired network are largely inapplicable due to the severely limited network bandwidth of multi-hop LWNs.

In this work, we develop a system called Adaptive Stream Multicast (ASM) for voice communication over multi-hop LWNs. ASM is a *multi-layer* solution that are composed of several novel components specially designed to deliver robust voice quality for multiple sinks with minimum bandwidth usage: 1) ASM employs an empirical voice quality model to automatically evaluate the application-level voice quality perceived at sinks based on current network condition. 2) To mitigate the impact of lossy links on voice quality, ASM employs a transport-layer feedback-based Forward Error Correction (FEC) adaptation scheme to allow the source to dynamically adapt its redundancy ratio in response to the voice quality variation at sinks. ASM also uses several novel mechanisms to reduce the signaling overhead of FEC adaptation, filter the coding redundancy on a multicast tree, and account for the impact of bursty packet loss. 3) We develop a lightweight Tree-based Opportunistic Routing (TOR) that fully exploits the broadcast opportunities on a multicast tree based on novel forwarder selection and coordination rules. 4) Finally, ASM includes a distributed cross-layer admission control algorithm that prevents a new stream from violating the voice quality guarantees for existing streams. The algorithm accounts for both link-level capacity induced by interference as well as transport-layer rate constraints from multiple voice streams. Moreover, it can be implemented in a fully distributed fashion. To the best of our knowledge, ASM is the first work that systematically addresses the challenges of supporting quality-assured voice communication over multi-hop LWNs. We have implemented ASM on SenEar, a low-power hardware platform we developed for wireless audio communication. Our extensive experiments on a testbed of 18 SenEar nodes show that ASM can achieve satisfactory multicast voice quality in dynamic environments while incurring low communication overhead.

The rest of this paper is organized as follows. Section 2 reviews related work. In Section 3, we describe the system architecture of ASM. We discuss the design and implementation of ASM in Section 4. We offer experimental results in Section 5 and conclude the paper in Section 6.

2 Related Work

Quality of Service (QoS) for multimedia applications has been extensively studied in the Internet community [24]. However, the existing QoS mechanisms for wired networks are inapplicable to our problem due to the challenges in multi-hop LWNs such as severely limited network capacity caused by wireless interference. We now review three bodies of work related to this paper.

Recent work [16, 20] shows that packet loss has a more significant impact on the quality of voice streams than delay. Several studies [8, 14] integrated network coding with opportunistic routing to achieve reliable multicast in mesh networks. However, these solutions require packets to carry long code vectors and hence are ill-suited for the LWNs which have limited frame sizes. For instance, a code vector of 32 bytes adopted in [8, 14] takes 51.6% of the maximum frame size of the CC1100 radio [29].

FEC is an effective loss mitigation scheme for multimedia streaming in both wired [32] and wireless networks [10, 13]. Although adopting FEC for packet recovery is not new, we develop a novel feedback-based adaptation mechanism that dynamically adjusts coding redundancy ratio in response to the voice quality variation. Moreover, the existing FEC schemes do not explicitly exploit multicast to save network bandwidth usage. We present several mechanisms to reduce the signaling overhead of FEC adaptation and the coding redundancy on a multicast tree.

Several systems have been developed for voice streaming in LWNs. In [7], the authors evaluated the impact of several factors including delay, jitter, throughput, and packet loss on an 802.15.4 Zigbee testbed. In [18], a system was developed for voice unicast in sensor networks that achieves a global TDMA schedule via hardware-based time synchronization. In our previous work [16], we developed a system called QVS for voice streaming. This paper distinguishes from [16] and [18] in several key aspects. First, the systems in [16, 18] are designed for a convergecast-based scenario where a small number of sources send voice streams to the base station while we focus on voice stream multicast from a source to a large number of sinks. Second, the main mechanisms described in this paper (*e.g.,* adaptive FEC, opportunistic routing, and admission control) are specially designed to reduce network bandwidth usage for multicast, which are not studied in [16, 18]. To our best knowledge, this paper is the first work that addresses the issue of voice stream multicast in multi-hop LWNs.

3 System Overview

ASM has two main design objectives in mind: (1) the system should assure the voice quality of multiple sinks in face of dynamic link quality; and (2) the network bandwidth utilization should be minimized in order to support a large number of sinks.

The design of ASM is based on the assumption that a multicast routing tree has been built in the network (*e.g.* , using several existing techniques such as the shortest-path or Steiner tree [17, 30]). We also assume that the routing metric used is the Expected Transmission Count (ETX), which has been widely adopted in wireless networks [30]. We assume that a buffer is maintained at each sink for mitigating the impact of inter-packet jitter and packet disorder. Several existing schemes such as [27] can be employed for buffer management. Figure 1 shows the architecture of ASM. ASM features the following components that reside on multiple layers of network stack to ensure voice quality while minimizing the network bandwidth usage.

(1) **Empirical Voice Quality Modeling**: Every sink estimates application-level voice quality based on the data loss rate and codec-specific characteristics. The estimation is sent back to the source and used by other components of ASM to achieve automatic online quality assurance.

(2) **Adaptive FEC for Multicast**: At the transport layer, the source applies an adaptive FEC scheme that can dynamically adjust coding redundancy ratio according to feedback. At the same time, relay nodes only forward the feedback information from the sink with the highest path loss rate. Moreover, relay nodes use a redundancy filter that drops redundant voice packets received from upstream nodes before forwarding to sinks with low path loss rates.

(3) **Tree-based Opportunistic Routing**: To improve throughput and reduce bandwidth consumption, we develop a tree-base opportunistic routing protocol (TOR). TOR can fully exploit the broadcast opportunities on a multicast tree based on novel forwarder selection and coordination rules.

(4) **Distributed Admission Control**: A cross-layer admission control algorithm is designed to prevent new streams from violating the voice quality guarantees for existing streams. The algorithm can accurately estimate the available link-level network capacity in the presence of interference and also account for the transport-layer rate constraints of multiple voice streams.

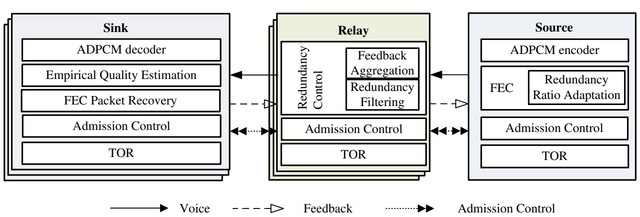

Figure 1. The system architecture of adaptive voice stream multicast system ASM.

We have implemented ASM on *SenEar*, a low-cost hardware platform designed for voice sampling and wireless communication in our earlier work [2]. The details of the SenEar platform are described in Section 5. ASM does not include a power management component as it is mainly designed for short-term emergency situations. For instance, a coal mine monitoring network only needs to operate for about a week when it is fully activated [18]. The system lifetime can be significantly extended by duty-cycling nodes. However, integrating ASM with duty-cycling protocols is left for future work.

4 Design and Implementation of ASM

In this section, we present the details of each component in the ASM system and discuss how our design objectives are achieved via the interactions between the source, sinks, and relay nodes.

4.1 Voice Quality Modeling

Voice streaming in LWNs often suffers frequent quality variation due to network dynamics caused by unpredictable noise and interference. To maintain satisfactory voice quality, ASM employs a voice quality model to automatically estimate the current streaming quality and adjust system parameters accordingly. Mean Opinion Score (MOS) is a widely adopted subjective model that qualifies speech quality with a score between 1 and 5 given by a group of audiences. A voice stream with MOS of 4.0 or higher is considered satisfactory, while 2.6 or below is considered unacceptable. Although MOS has been shown effective in evaluating voice quality, it is subjective and hence cannot be used for automatic voice evaluation and control.

We adopt the E-model [11] for objective voice quality assessment in this work. In the E-model, a quantity called R-value is derived from delays and equipment impairment factors of a voice stream. The impact of delays can be effectively mitigated by using playback buffers at sinks. For instance, our experimental results show that a delay of 177.3 *ms* only decrease the voice quality by 4.26%. Therefore, the quality degradation due to delay can be safely approximated as a constant as long as the delay does not exceed a certain threshold. In particular, ASM is designed to satisfy moderate voice quality requirement and hence can tolerate an end-to-end communication delay up to several hundred milliseconds that can be easily achieved on multi-hop wireless networks. Therefore, in the following, we focus on modeling the impact of data loss on voice quality.

To reduce bandwidth usage, the SenEar platform compresses the raw voice data captured by the microphone at 8KHz by an Adaptive Differential Pulse Code Modulation (ADPCM) [12] encoder which is suitable for low-cost platforms. The microphone outputs 8-bit values that are compressed to 5-bit, 4-bit, 3-bit or 2-bit ADPCM codes. The corresponding data rates are 40 Kbps, 32 Kbps, 24 Kbps, and 16 Kbps, respectively. For each data rate, we conducted extensive experiments on the SenEar platform to evaluate the quality of voice streams with different loss rates. The results are omitted here due to space limitation and can be found in [15]. Based on the results, ASM adopts the 24 Kbps ADPCM as the codec as it achieves a desirable tradeoff between bandwidth usage and voice quality.

As shown in [16], a MOS score of 2.6 is a reasonable lower bound on acceptable voice quality. Therefore, the *voice quality constraint* in our system is that $MOS \geq 2.6$. MOS values can be mapping to R-values according to the model in [11]. Therefore, a MOS score of 2.6 corresponds to a data loss rate of 12.8% (referred to as L_{th} hereafter) for the 24 Kbps ADPCM codec. Therefore, the quality of a voice stream can be ensured if the end-to-end data loss rate is lower than L_{th}.

4.2 Adaptive FEC for Multicast

In this section, we first introduce the background on FEC, and discuss the design of feedback-based adaptive FEC for voice streaming. We then refine this FEC scheme to address special challenges raised by multicast, followed by an extension to the FEC scheme to handle bursty packet loss.

4.2.1 Coding Scheme Selection

A desirable feature of FEC is that only the source and sink nodes are active participants in the coding/decoding process while no processing is needed at the intermediate nodes. Typical FEC schemes mitigate packet loss by sending raw data packets together with additional *parity packets*. In particular, ASM employs the (n, k) systematic FEC code where a block of k data packets (output from the ADPCM codec) are encoded at the sender to generate a block of n $(n \geq k)$ encoded packets, including $n - k$ parity packets. If the receiver receives any k packets in the block, it can recover the original k data packets. If less than k encoded packets are received, a fraction of the data packets can still be recovered. An important performance metric of FEC is the *redundancy ratio* α defined as n/k.

ASM adopts the Reed-Solomon (R-S) FEC code for its high capability of packet recovery. The details on how parity packets are generated may be found in [21]. For the convenience of discussion, we use *packet loss rate* (denoted by p) to represent the raw loss rate of a network path while *data loss rate* (denoted by P_l) to represent the actual data loss after recovering from coded packets received by the sinks. Assuming that the packet loss follows a Bernoulli process, the actual data loss rate after applying FEC can be described by the following function of n, k, and p [23].

$$P_l = L(n,k,p) = p\left(1 - \sum_{j=0}^{n-k} \binom{n}{j} p^j (1-p)^{n-j}\right) \qquad (1)$$

Note that the above equation is derived under the assumption that packet loss follows a Bernoulli process. However, recent empirical results [26] show that packet losses on low-power wireless links often yield considerable burstiness. We have developed an empirical method to account for the impact of burst loss in the derivation of P_l. However, it is omitted here because of space limitation and can be found in [15] . Using Eq. (1), the source node can calculate a proper n to meet the voice quality constraint. Formally, the minimum n is calculated by:

$$n_{min} = \arg\min L(n,k,p) \leq L_{th} \qquad (2)$$

Although the optimal coding parameter n can be determined based on the current path quality, the following issues remain to be addressed for voice stream multicast. First, wireless links suffer inherent quality variation, which makes the optimal coding parameter n highly variable. Second, in order to ensure the voice quality of multiple sinks, the source must select the redundancy ratio based on the sink with the *highest* path loss rate. As a result, other sinks on the multicast tree with a lower path loss rate can receive excessive redundant coded packets leading to a waste of bandwidth. ASM employs several novel mechanisms to address the above challenges. First, we develop a light-weight feedback scheme for adaptive stream multicast based on implicit piggyback. The redundancy ratio at the source is adaptively adjusted based on real-time voice quality perceived by the sinks. Second, the relay nodes on each branch of multicast tree minimize the bandwidth consumption by filtering out redundant coded packets being forwarded on this branch while satisfying the quality requirement of the voice stream.

4.2.2 Feedback and Redundancy Filtering

In order to achieve automatic voice quality assurance, sinks monitor the quality of the received voice stream and send feedback to the source. We design an adaptive scheme where each sink maintains a sliding window containing the packet loss rates during the last s seconds. The sink calculates the mean packet loss rate during the last s seconds and then drives the required coding redundancy ratio α according to Eq. (2). If the new ratio is different from the last value, the sink will send feedback to inform the source.

To reduce the network bandwidth usage, ASM further optimizes the feedback-based FEC using two mechanisms. First, it sends feedback to the source using an implicit piggybacking scheme. Specially, sinks first send the feedback to their parents on the multicast tree. Then, upon receiving the feedback, the nodes piggyback it in the header of the data packets to be forwarded. Upper stream nodes obtain the feedback by overhearing the data packets from their child nodes. Moreover, as the sink with the *highest* data loss rate determines the coding redundancy ratio of source, the feedback from multiple sinks is aggregated at each tree junction node. In other words, only the feedback from the sink with the highest path loss rate is passed along the tree

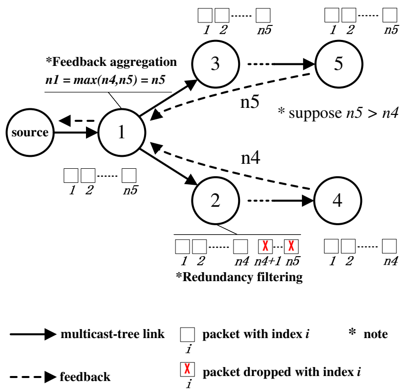

Figure 2. Feedback aggregation and redundancy filtering. The coding parameters computed by $node_4$ and $node_5$ are n_4 and n_5, respectively.

toward the source, while other smaller values are dropped during aggregation at intermediate nodes.

After receiving the feedback, the source determines the redundancy ratio α, which is decided by two parameters k and n. A larger k is typically more robust to burst loss, but leads to higher computation and storage overhead. In our implementation, we preset k to 8. On SenEar nodes, encoding and decoding one parity packet takes an average time of 0.215ms and 0.3ms, respectively. The time increases with several factors such as the payload size and the parameter k. For any given packet loss rate p, n can be derived offline using Eq. (2). In our implementation, the source initially sends the voice data with a default redundancy ratio 1.0, *i.e.,* $n = k$, and then adjusts it according to the feedback from sinks. Note that a large n may cause the total data rate to exceed the available network capacity. Therefore, we restrict n by an upper bound (12 in our implementation, corresponding to a redundancy ratio of 1.5).

Since the redundancy ratio is adjusted according to the sink with the highest path loss rate, it may result in considerable bandwidth waste if relay nodes forward all received packets directly. We introduce a redundancy filter to address this issue. Each junction node on the multicast tree maintains the highest redundancy ratio required by the sinks in the subtree rooted at it. Since k is fixed in our implementation, the nodes only need to maintain coding parameter n. Suppose the coding parameter maintained at a relay node i and the source are n_i and n_{src}, respectively. According to our discussion, n_{src} should be no smaller than n_i. Each packet in the same block is sent with an index from 1 to n_{src}. To filter out redundant packets, node i will drop all packets with the index higher than n_i.

Fig. 2 illustrates the feedback aggregation and redundancy filtering using a five-node topology. Suppose the coding parameters computed by $node_4$ and $node_5$ are n_4 and n_5, respectively. We also assume that $n_5 > n_4$ and n_5 arrives at node 1 first. As a result, when n_4 is received by node 1, it will be dropped as it has no impact on the coding redundancy ratio of the source that is determined by a larger value n_5. When node 2 receives the data packets, it will drop all the packets with indexes in $(n_4, n_5]$ as n_4 packets are sufficient for achieving satisfactory voice quality at node 4. In contrast, n_3 cannot drop any packet as the voice stream is generated by the source according to the requirement of a downstream node ($node_5$).

4.3 Tree-based Opportunistic Routing

Opportunistic Routing (OR) has been shown to be a promising technique to improve network throughput by exploiting the broadcast nature of wireless networks. In typical OR protocols, the sender broadcasts its data among the nodes that can overhear the transmission. Then, the node closest to the destination is selected to forward the data.

There exist a number of opportunistic routing schemes such as ExOR[6] and SOAR [22]. However, to our best knowledge, none of them has addressed the following issues specific to stream multicast over LWNs. (1) *Candidate forwarder selection*: Unlike unicast, there are multiple sinks on a multicast tree. Therefore, selecting candidate forwarders by only considering the ETX from a relay node to the single sink will not work. A simple adjustment is to have the sender specify several candidate lists in the packet header for different sinks. However, this is not scalable for low-power platforms due to the frame size limitation. (2) *Coordination between forwarders*: When multiple forwarders overhear a transmission, only a subset of them should forward the packet to avoid duplicate transmissions. However, it is nontrivial to select the right subset and also coordinate the transmissions among them because this subset can have nodes on the paths to multiple sinks in the case of multicast.

The unique challenges described previously motivate the design of a new OR protocol called tree-based OR (TOR) for stream multicast. TOR runs on top of a routing tree connecting the source to the sinks. We assume that each link chosen by the multicast tree or TOR is bidirectional although the ETXs in two directions may be different. We now discuss the design of TOR in detail.

Candidate forwarder selection: Denote e_i as the total ETX from the source to node i. The following constraints must be satisfied for node j to be eligible to be a candidate forwarder of node i in TOR: (C1) Node j is on the multicast tree; and (C2) The ETX from the source to node i and node j should satisfy $e_i - e_j \leq \theta$ ($\theta \geq 0$).

C1 ensures that only the nodes on the tree can be potential candidate forwarders. The intuition behind C2 is that a node farther away from the source is more preferable to be the candidate forwarder. C2 allows node j to be the candidate in two cases. (1) If i and j are on the same path from the source to a sink, j can be a candidate forwarder of i only if i is the ancestor or parent of j on the multicast tree. (2) If i and j are on different paths, j can be a candidate forwarder of i if j has a larger or comparable ETX from the source than i (as θ is a small positive constant). Allowing to overhear from siblings (or uncles) improves reliability.

Coordination between forwarders: Eligible candidates that satisfy both C1 and C2 need to be coordinated such that the most efficient candidate will actually forward the packet. An ideal coordination rule should ensure that, *when candidates on a path from the source to a sink overhear the same transmission, the one closest to the sink should start the forwarding before any other candidate on the same path*. A key difference between ORs for unicast and multicast is that this rule must be ensured for *each* path in multicast despite the fact that a node may lie on multiple paths on the tree. In other words, multicast OR must account for the ordering of forwarding candidates on each path from the source to a sink. With this in mind, we now describe how we enforce this rule in a distributed fashion. The implementation is based on adjusting backoff timer at each candidate and the pseudo-code is illustrated in Algorithm 1. When node i sends a packet m, it includes the ETX value e_i in the packet header. When node j overhears this transmission, the treatment to m can be divided into two cases.

In the first case (Lines 1 to 6), j is not the parent of i. j first determines whether it is a candidate forwarder locally by checking the two constraints (C1) and (C2). If j is not a candidate forwarder (i.e., $e_i - e_j > \theta$) or it has received the packet m before, j will drop m. Otherwise, j sets a timer $T_{m,j}$ according to the difference between e_i and e_j (Line 6) and buffer packet m. When the timer fires, j will start the transmission of m.

In the second case (Lines 8 to 14), j is the parent of i. In order to avoid duplicate transmissions, j checks the timer $T_{m,j}$. If $T_{m,j}$ is on, it implies that m is pending. If j has only one child, j cancels the timer to avoid duplicate transmissions; if j has multiple children, it keeps waiting until it overhears the transmissions of m from all children. When it does, $T_{m,j}$ is canceled to avoid duplicate transmissions. Otherwise, the timer is reset to its initial length to wait for transmissions from remaining children as shown in Line 12. If the timer fires, which implies that some children have not received m, j will forward m.

T_{max} and ΔT in Algorithm 1 are two parameters set before run time. T_{max} should be large enough in order to ensure positive timer duration. ΔT should account for the time for transmitting a packet, and the variable delay due to queueing and contention. We set $\Delta T = 4ms$, larger than roughly twice the time for sending a packet on SenEar, and $T_{max} = 6ms$. These settings have been tested in various settings in our experiments.

Algorithm 1: Forwarding algorithm of node j.

Input: Packet m sent by i and the ETX from the source to the sender i, *i.e.* , e_i

1 **if** j is not the parent of i **then**
2   **if** $e_i - e_j > \theta$ **or** j has received m before **then**
3   drop m;
4   **else**
5   setup a timer with timeout
6         $T_{m,j} \leftarrow T_{max} - |e_j - e_i + \theta| \cdot \Delta T$;
7 **else**
8   **if** $T_{m,j}$ is on **then**
9   **if** j has overheard the transmission of m from **all** its children **then**
10   cancel the timer $T_{m,j}$;
11   **else**
12   reset $T_{m,j}$ according to Line 6;
13   **else**
14   drop m;
15 **Event: Timer** $T_{m,j}$ **fires**
16   send m.

We now take an example shown in Fig. 3 (left) to explain the algorithm and the parameter settings. To simplify the discussion, we assume the ETX of all links on the tree is e, so the ETX from $node_1$ to $node_2$, $node_3$, and $node_4$ are e, $2e$, and $2e$, respectively. Suppose a packet m is sent by $node_1$, then $node_2$, $node_3$, and $node_4$ overhear m at the same time. Afterwards, they hold the forwarding of m and set a timer according to Line 6 in Algorithm 1. It can be verified that the timer settings satisfy:

$$\begin{aligned} T_{m,2} &= T_{max} - |e - 0 + \theta| \cdot \Delta T \\ T_{m,3} &= T_{max} - |2e - 0 + \theta| \cdot \Delta T \\ T_{m,4} &= T_{max} - |2e - 0 + \theta| \cdot \Delta T \end{aligned} \quad (3)$$

According to Eq. (3), $node_3$ and $node_4$ will start the forwarding of m before $node_2$, which is consistent with our design objective. If $node_2$ overhears the transmission of m from both $node_3$ and $node_4$, $node_2$ will cancel the forwarding of m. Otherwise, when $T_{m,2}$ fires, $node_2$ will send m. We should note that Algorithm 1 also accounts for overhearing packets from siblings (or uncles). For example, when both $node_4$ and $node_6$ receive a packet m from $node_3$, $T_{m,4} > T_{m,6}$, so $node_6$ forwards m.

4.4 Distributed Admission Control

Admission control determines whether a sink can be accommodated in the network while achieving satisfactory voice quality for all admitted sinks. Existing admission control algorithms designed for Ad-hoc networks [9, 31] do not

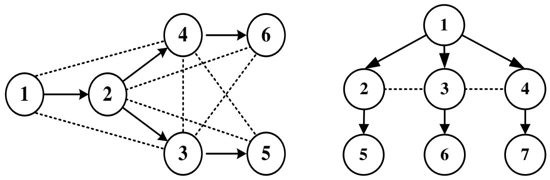

Figure 3. Left: A topology with 6 nodes. Right: A network consisting of 7 nodes. The solid lines are the links on the multicast tree while the dashed lines are interference links.

support stream multicast with voice quality guarantee and hence are not suitable for ASM.

We assume that sinks request to join the multicast tree one at a time. This assumption is reasonable because the admission control process in ASM only takes several hundred of milliseconds. The network capacity may be exceeded in two cases: First, when a new sink joins the network, the traffic load of a node may increase if it lies on the path from the source to the new sink or it is interfered by any node on the path. Second, the traffic load may increase if the source increases the redundancy ratio as requested by a sink. For the first case, ASM predicts the traffic load of each node. If the new load of any node exceeds the available bandwidth capacity, the admission of the new sink is denied. For the second case, each sink compares the voice quality before and after redundancy ratio adjustment. If the voice quality decreases with an increased redundancy ratio at the source, it implies that the network capacity may be exceeded leading to a high packet loss rate along the path. As a result, the sink will first leave the multicast tree and then request to join as a new sink. Our following discussion only focuses on the case that a new sink joins the network.

An advantage of ASM is that the admission control is performed efficiently in a distributed fashion. A key step of the admission control algorithm is that each node checks whether the new total traffic load would exceed the available bandwidth if a new sink were admitted. We define the maximum bandwidth available at a node as *local capacity*. We first introduce two concepts used to estimate the local capacity, *contention domain* and *saturation rate* [5].

Contention domain of node i refers to the set of nodes who share the bandwidth with i (including i itself), *i.e.,* whose transmissions interfere with i. The saturation rate is the maximum throughput observed at a receiver when all senders are within the contention domain of each other. Recent experiments on CC2420 CSMA MAC show that saturation rate could be an reasonable approximation of the local capacity [25]. The saturation rate is solely dependent on the number of nodes in a contention domain and hence it can be accurately measured.

Let $C(i)$ represent the contention domain of node i and B_i represent the local capacity of node i. λ_i denotes the data rate of node i and $\lambda_{C(i)}$ is the aggregated data rate of all nodes in $C(i)$. The *local capacity constraint* is then given as follows:

$$\lambda_{C(i)} = \sum_{j \in C(i)} \lambda_j \leq B_i \tag{4}$$

We now illustrate this using an example shown in Fig. 3 (right). As the contention domain of $node_3$ includes all the nodes except $node_5$ and $node_7$, the local capacity constraint of $node_3$ can be expressed as follows:

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_6 \leq B_3 \tag{5}$$

The objective of admission control is to ensure that the local capacity constraint for each node in the network should be met if the new sink is admitted. In ASM, a new sink joins the multicast by sending a *request* with a desirable data rate λ_{req} to the source. This λ_{req} is derived by accounting for several factors including raw data rate, path loss rate, and redundancy ratio, as discussed in Section 4.1 and 4.2. Then, the request is forwarded along the path and each node interfered by the new stream checks whether its local capacity constraint is violated. The new stream is accommodated if and only if the local capacity constraints of all nodes interfered by the new stream are respected. Due to space limitation, the detailed description of the admission control process is omitted and can be found in [15].

5 Performance Evaluation

ASM has been implemented on the SenEar platform [2] that we developed for low-power voice communication. SenEar uses a 32-bit Atmel AT91SAM7S256 microcontroller [4] with 256 KB of flash memory and 64 KB of RAM. The transceiver is Chipcon CC1100 [29] which has a 64-byte FIFO buffer and a maximum data rate of 500 Kbps. The frame size is limited to 64 bytes including 2 status bytes for CRC and RSSI. The Blaze radio stack [3] including the CC1100 driver and a CSMA-based MAC protocol from TinyOS-2.x has been ported to SenEar. Our implementation of ASM has a code size of 56.4KB and uses 1.43KB of RAM. The power consumption of SenEar at full workload is measured as 77.67 mA. SenEar is designed to connect with replaceable external sockets for various batteries such as AA, C, and D batteries. Depending on the capacity of battery, the expected life time of a SenEar node ranges from 38 to 154 hours with two batteries. SenEar is a suitable platform for us as ASM is mainly designed for short-term emergency situations. In practice, the system life can be significantly extended by periodically switching nodes off when there is no voice communication.

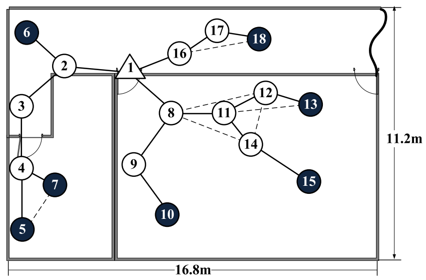

Figure 4. A testbed with 18 SenEar nodes. Solid lines are the tree-edge links, while dashed lines are interference links.

We evaluate ASM using an 18-node deployment in our lab illustrated in Fig. 4. The CC1100 radio on each node uses a transmit power of -30 dBm. We set the MOS value of 2.6 as the threshold for an acceptable voice stream. In our deployment, $node_1$ is the source and the black nodes are sinks. The multicast tree is built by running a tree-based routing protocol where minimum ETX is used as the cost metric. The source generates a voice stream, which is sent to all sinks on the multicast tree. Nodes may be turned off in order to evaluate ASM on different topologies.

5.1 Evaluation of Adaptive FEC Scheme

We first use a single voice stream ($sink_5$) to evaluate the feedback scheme used in our adaptive FEC scheme. We refer to the one described in Section 4.2.2 as *Sliding*. Sliding maintains a window of measured voice qualities in history, one per second, then average across the window to determine wether to send a feedback to the source. For comparison, we implement a baseline in which each sink periodically calculates immediate voice quality and sends the required redundancy ratio as feedback (referred to as *Periodic*).

Figure 5 shows the CDFs of the average voice quality for the two schemes under various settings. The outage time (when $MOS < 2.6$) of the voice stream in Periodic with the period of 5s, 10s, and 15s are 20.5%, 25.5%, and 36.7%, respectively. The outage time for Sliding under the window size of 10s and 30s are 14.7% and 22.7%, respectively. Fig. 6 shows the number of feedback packets sent by the sinks. Periodic with period of 5s performs the worst among all schemes. Sliding with window size of 10s incurs higher overhead than Periodic with period of 10s because the feedback is triggered more often than every 10s due to the high path loss rate. Interestingly, when the window size is 30s, Sliding sends less feedback packets than Periodic with period of 15s while the quality outage time is considerably shorter (Fig. 5). This is because a long sliding window smooths out the average path loss rate and hence sporadic high loss rates can not trigger the feedback. Nevertheless, it is able to capture the voice quality variation more accurately than Periodic with a shorter period. In our following evaluation we adopt Sliding as the feedback scheme.

The main design objective of adaptive FEC is that the source should adjust its coding redundancy ratio according to the feedback from all sinks. We use $sink_5$ and $sink_{10}$ to evaluate its performance on multiple sinks. We compute the required redundancy ratios at the two sinks based on their feedback information, and then record the actual redundancy ratio used by the source. Figure 7(a) shows the three redundancy ratios. We can see that the source always adjusts its redundancy ratio according to the sink requiring higher redundancy (*i.e.,* with higher packet loss rate), which is consistent with what we described in Section 4.2.2. Figure 7(b) shows the corresponding voice quality at $sink_5$ and $sink_{10}$, respectively. We can see the voice quality constraints for both streams are satisfied in most of the time.

5.2 Evaluation of TOR

We mainly focus on evaluating TOR's effectiveness in bandwidth reduction. Two sinks $sink_{13}$ and $sink_{15}$ on our testbed are used. We compared TOR against a tree-based multicast protocol (refereed to as *Tree*). Tree is built using the ETX-based shortest path algorithm. Note that TOR uses the same tree topology as Tree while the difference between them is whether the broadcast opportunities on the topology is explored. During the experiments, we recorded the packet reception rate at the two sinks.

The key observation is shown in Fig. 8. Fig. 8 shows the packet reception rates in the cases of one and two sinks, which is about 4.9% improvement over Tree. There are two reasons for the reduction of bandwidth consumption using TOR. First, the improvement in packet reception rate implies the reduction of redundancy ratio. In the case that only $sink_{15}$ joins the network, the data loss rate under TOR and Tree are 85.3% and 90.6%, respectively, corresponding to the redundancy ratio of 1.125 and 1.0. Second, the total number of transmissions is slightly reduced by leveraging broadcast opportunities. For instance, on the path from $node_1$ to $sink_{13}$ in Fig. 4, a fraction of packets (more than 30% in our experiments) broadcasted by $node_8$ is directly forwarded by $node_{12}$.

5.3 Evaluation of Admission Control

In the first set of experiments, Tree is used and the sinks join the network in the order of $sink_5$, $sink_6$, $sink_7$, $sink_{10}$, $sink_{18}$, and $sink_{13}$. Figure 9 shows the MOS values perceived at the sinks $sink_{18}$, $sink_{10}$, and $sink_{13}$. For clarity, the curves of voice quality perceived at other sinks

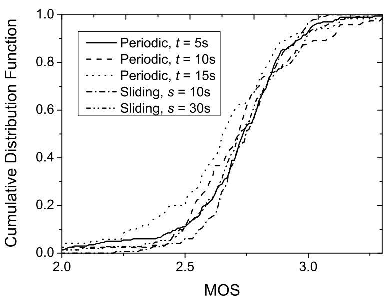

Figure 5. The cumulative distribution function of voice quality (MOS value) for different feedback schemes.

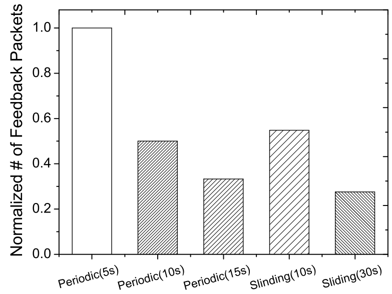

Figure 6. The normalized number of feedback packets sent for different feedback schemes.

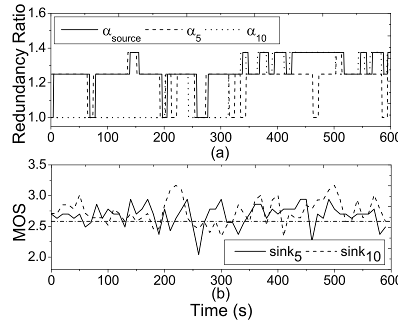

Figure 7. Adaptation of redundancy ratio and its impact on voice quality. The dash dotted line in (b) represents the voice quality threshold.

are not shown. The average voice quality at the other three sinks $sink_5$, $sink_6$, and $sink_7$ are 2.78, 3.35, and 3.01, respectively.

In Fig. 9, the MOS values of all sinks stay above the threshold before the joining of $sink_{13}$. When $sink_{13}$ joined the network, the local capacity at $node_8$ was exceeded and hence it was rejected by admission control. However, in order to test the accuracy of estimation of the local capacity, we intentionally programmed all nodes to only record the admission control decisions without notifying the source. As a result, the joining of $sink_{13}$ is accommodated. In Fig. 9, we can see both the MOS values perceived at $sink_{10}$ and $sink_{13}$ were below the threshold, on average 1.67 and 1.74, respectively. The MOS values at other sinks were not evidently affected and still above the threshold. The results show that our admission control algorithm can accurately estimate the local network capacity and admit the maximum number of flows.

In the second set of experiments, TOR is used and the sinks join the network in the same order as above. The results are shown in Fig. 10. Compared with Fig. 9, $sink_{13}$ joined the network without violating the local capacity constraint of $node_8$. This is due to the reduction of bandwidth usage by TOR. To study the impact of TOR on admission control, we also recorded the aggregated data rate in the contention domain of $node_8$, which consists of $node_1$, $node_8$, $node_9$, $node_{11}$, and $node_{12}$. The saturation throughput is exceeded when Tree is used, leading to high packet loss rate and unacceptable voice quality at both $sink_{10}$ and $sink_{13}$. In contrast, when TOR is used, the bandwidth consumption is reduced below the local capacity. As a result, TOR accommodates an additional stream under our experiment situation.

6 Conclusion

In this paper, we develop the Adaptive Stream Multicast (ASM) system for voice communication over multi-hop LWNs. ASM is composed of several novel components specially designed to deliver robust voice quality in dynamic environments. Our extensive experiments on a testbed of 18 nodes show several advantages of ASM. The adaptive feedback-based FEC scheme allows the source to adapt its coding redundancy ratio dynamically based on the voice quality variation at sinks. The Tree-based Opportunistic Routing (TOR) protocol can effectively explore the broadcast opportunities on a tree and save network bandwidth usage, which leads to more concurrent sinks that can supported. Finally, the distributed admission control algorithm can accurately estimate the local network capacity and admit the maximum number of streams in our experiments.

7 Acknowledgement

This work was supported by the State Key Program of National Natural Science of China (Grant No. 60933011), the State Key Development Program for Basic Research of China (Grant No. 2011CB302902), and the US National Science Foundation (Grant No. CNS-0916576).

References

[1] Mining communications and tracking workshop. Sponsored by National Institute of Occupational Safety and Health, Lakewood, Colorado, May 13-14, 2009.

[2] SenEar wireless audio platform. `http://www.cse.msu.edu/~glxing/SenEar/`.

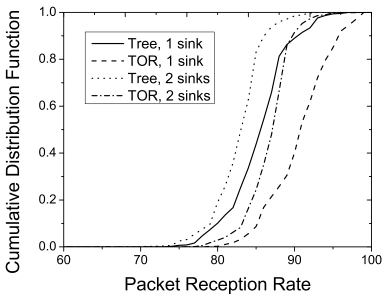

Figure 8. Tree vs. TOR in packet reception rate in the cases of one and two sinks.

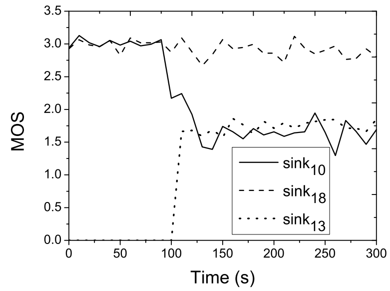

Figure 9. Admission control with the Tree routing strategy.

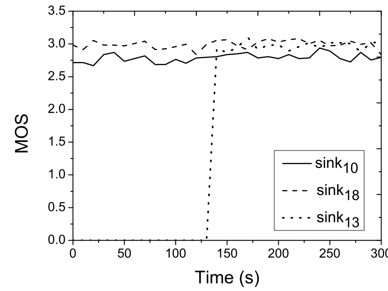

Figure 10. Admission control with the TOR routing strategy.

[3] The Blaze protocol stack. `http://tinyos.cvs.sourceforge.net/tinyos/tinyos-2.x-contrib/blaze/`.

[4] Atmel. AT91SAM7S256 datasheets. 2003.

[5] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE JSAC*, 2000.

[6] S. Biswas and R. Morris. ExOR: opportunistic multi-hop routing for wireless networks. In *SIGCOMM*, 2005.

[7] D. Brunelli, M. Maggiorotti, L. Benini, and F. L. Bellifemine. Analysis of audio streaming capability of zigbee networks. In *EWSN*, volume 4913, pages 189–204, 2008.

[8] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *SIGCOMM*, pages 169–180, 2007.

[9] I. D. Chakeres and E. M. Belding-Royer. PAC: Perceptive admission control for mobile wireless networks. In *QSHINE*, pages 18–26, 2004.

[10] V. R. Gandikota, B. R. Tamma, and C. S. R. Murthy. Adaptive fec-based packet loss resilience scheme for supporting voice communication over ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, 2008.

[11] International Telecommunication Union. The E-model, a computational model for use in transmission planning, G.107. Technical report, 2005.

[12] ITU-T Recommendation G.726. 40,32,24,16 kbit/s adaptive differential pulse code modulation(ADPCM), 1990.

[13] D. Koutsonikolas and Y. Charlie Hu. Exploring the design space of reliable multicast protocols for wireless mesh networks. *Ad Hoc Netw.*, 7(5):932–954, 2009.

[14] D. Koutsonikolas, Y. C. Hu, and C.-C. Wang. Pacifier: High-throughput, reliable multicast without "crying babies" in wireless mesh networks. In *Proceedings of IEEE INFOCOM*, April 19-25 2009.

[15] L. Li, G. Xing, Q. Han, and L. Sun. Adaptive voice stream multicast over low-power wireless networks. Technical Report MSU-CSE-10-16, Computer Science and Engineering, Michigan State University, East Lansing, Michigan, 2010.

[16] L. Li, G. Xing, L. Sun, and Y. Liu. QVS: quality-aware voice streaming for wireless sensor networks. In *ICDCS*, 2009.

[17] H. Luo, G. Xing, M. Li, and X. Jia. Dynamic multiresolution data dissemination in storage-centric wireless sensor networks. In *MSWiM*, 2007.

[18] R. Mangharam, A. Rowe, R. Rajkumar, and R. Suzuki. Voice over sensor networks. In *RTSS*, 2006.

[19] C. S. Networks. Audio and visual boards. `http://www.sensornets.csiro.au/daughterboard\_av.htm`.

[20] D. Niculescu, S. Ganguly, K. Kim, and R. Izmailov. Performance of VoIP in a 802.11 wireless mesh network. In *In Proc. IEEE INFOCOM*, 2006.

[21] L. Rizzo. Effective erasure codes for reliable computer communication protocols. *SIGCOMM Comput. Commun. Rev.*, 27(2):24–36, 1997.

[22] E. Rozner, J. Seshadri, Y. Mehta, and L. Qiu. SOAR: Simple opportunistic adaptive routing protocol for wireless mesh networks. In *IEEE Transactions on Mobile Computing.*, 2009.

[23] D. Rubenstein, J. Kurose, and D. Towsley. Real-time reliable multicast using proactive forward error correction. In *In Proceedings of NOSSDAV '98*, 1998.

[24] B. Sat and B. Wah. Analysis and evaluation of the Skype and Google-Talk Voip systems. In *Multimedia and Expo, 2006 IEEE International Conference on*, July 2006.

[25] A. Sridharan and B. Krishnamachari. Explicit and precise rate control for wireless sensor networks. In *ACM Sensys*, 2009.

[26] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis. The β-factor: measuring wireless link burstiness. In *SenSys*, pages 29–42, 2008.

[27] L. Sun and E. Ifeachor. New models for perceived voice quality prediction and their applications in playout buffer optimization for voip networks. *ICC*, 3:1478–1483, 2004.

[28] C. Technology. Imb multimedia board. 2009.

[29] TI. Chipcon CC1100 datasheet. 2006.

[30] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *SenSys*, 2003.

[31] Y. Yang and R. Kravets. Contention-aware admission control for ad hoc networks. *Mobile Computing, IEEE Transactions on*, 4(4):363–377, July-Aug. 2005.

[32] S.-W. Yuk, M.-G. Kang, B.-C. Shin, and D.-H. Cho. An adaptive redundancy control method for erasure-code-based real-time data transmission over the internet. *Multimedia, IEEE Transactions on*, 2001.