

Performance Optimizations for Deploying VoIP Services in Mesh Networks

Samrat Ganguly, Vishnu Navda, *Student Member, IEEE*, Kyungtae Kim, Anand Kashyap, Dragos Niculescu, Rauf Izmailov, Sangjin Hong, *Senior Member, IEEE*, and Samir R. Das, *Member, IEEE*

Abstract—In the recent past, there has been a tremendous increase in the popularity of VoIP services as a result of huge growth in broadband access. The same voice-over-Internet protocol (VoIP) service poses new challenges when deployed over a wireless mesh network, while enabling users to make voice calls using WiFi phones. Packet losses and delay due to interference in a multiple-hop mesh network with limited capacity can significantly degrade the end-to-end VoIP call quality.

In this work, we discuss the basic requirements for efficient deployment of VoIP services over a mesh network. We present and evaluate practical optimizing techniques that can enhance the network capacity, maintain the VoIP quality and handle user mobility efficiently. Extensive experiments conducted on a real testbed and ns-2 provide insights into the performance issues and demonstrate the level of improvement that can be obtained by the proposed techniques. Specifically, we find that packet aggregation along with header compression can increase the number of supported VoIP calls in a multihop network by 2–3 times. The proposed fast path switching is highly effective in maintaining the VoIP quality. Our fast handoff scheme achieves almost negligible disruption during calls to roaming clients.

Index Terms—Handoff, mobility, quality-of-service (QoS), voice-over-IP, wireless mesh networks.

I. INTRODUCTION

VOICE-OVER-INTERNET PROTOCOL (VoIP) is steadily emerging as an important application over Internet with the tremendous popularity of Skype [1]. For example, Federal Communications Commission (FCC) data indicates that there are already 3–7 million residential customers using VoIP services from providers such as Vonage. In the corporate sector, the percentage of VoIP lines will become 44% by 2008. The cost savings and the easy deployment benefits achieved by VoIP using existing data infrastructures are the main factors driving the steady growth of VoIP.

Recently, due to the ubiquity of wireless local area network (WLAN) in homes and offices, wireless VoIP is also gaining significant popularity. Wireless VoIP can provide the caller the convenience of using portable phones with limited roaming. Already there exist softphones running on PDA, WiFi enabled VoIP handsets such as Zyxel and even dual cell phones with WiFi capabilities. Once VoIP over WLAN becomes widespread,

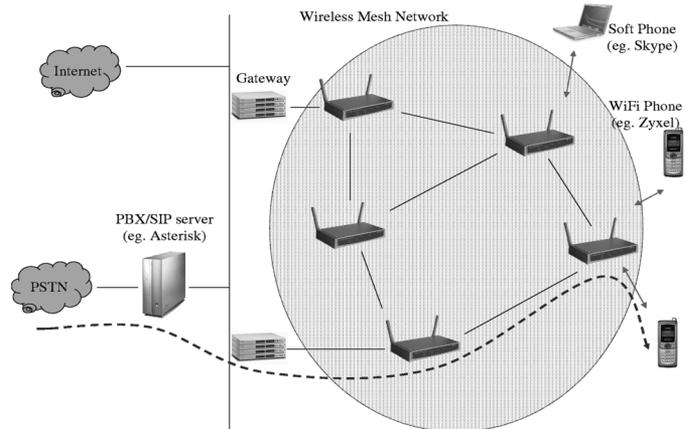


Fig. 1. Mesh network depicting two clients connected, and the paths maintained between them.

most cell phone or WiFi handset owners will migrate to using VoIP over WLAN inside the administrative boundaries of the enterprise buildings, campuses, public places such as airports or even in WLAN equipped homes.

Providing the VoIP users with true mobile phone services while having the freedom of roaming requires wide area wireless coverage. IEEE 802.11-based multihop wireless mesh networks have been considered as a practical and inexpensive solution for providing such wide area coverage. The benefits of mesh network compared with wired LAN connecting WiFi access points (APs) are: 1) ease of deployment and expansion; 2) better and wider coverage; 3) resilience to node failure; and 4) reduced cost of maintenance. Such a mesh network has the potential of creating an enterprise-scale or community-scale wireless backbone supporting multiple mobile VoIP users. A typical deployment scenario of VoIP service over mesh network is shown in Fig. 1. When such deployment scenarios become commonplace, it has the potential of driving the users from using fixed phones to wireless VoIP phones.

A. Contributions

In this work, we focus on designing a 802.11-based wireless mesh network that can efficiently support the VoIP calls. This work stems from our ongoing research on deploying VoIP service in an Enterprise Mesh Network. Toward a complete deployment of VoIP service, we target the following goals in this paper.

- *Increasing the VoIP capacity of mesh network:* As shown in Fig. 2, our experiment on a real mesh testbed with G.729 encoded VoIP calls indicates: 1) the number of supported medium quality calls decreases with the increase in hops

Manuscript received October 1, 2005; revised March 4, 2006 and May 1, 2006.

S. Ganguly, K. Kim, D. Niculescu, and R. Izmailov are with NEC Laboratories America, Princeton, NJ 08540 USA (e-mail: samrat@nec-labs.com; kyungtae@nec-labs.com; dragos@nec-labs.com; rauf@nec-labs.com).

V. Navda, A. Kashyap, S. Hong, and S. R. Das are with State University of New York at Stony Brook, Stony Brook, NY 11794 USA (e-mail: vnavda@cs.sunysb.edu; anand@cs.sunysb.edu; snjhong@ece.sunysb.edu; samir@cs.sunysb.edu).

Digital Object Identifier 10.1109/JSAC.2006.881594

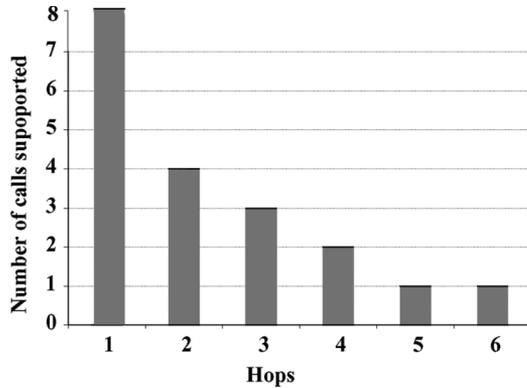


Fig. 2. Call capacity of a linear string of nodes degrade with the length of the string.

for a simple linear topology and 2) in a mesh network with 2 Mb/s link speed, the number of supported calls reduces from eight calls in single hop to one call after four hops. This significant reduction in the number of supported calls can be attributed to following factors: 1) decrease in the user datagram protocol (UDP) throughput because of self interference; 2) packet loss over multiple hops; and 3) high protocol overhead for VoIP packets and its small sizes (802.11/IP/UDP/RTP for each VoIP packet with 20 bytes payload).

Our objective is to increase the number of supported VoIP calls with the given quality measure over multihop mesh. We propose two approaches based on VoIP packet aggregation and header compression and demonstrate a 2–3 times increase in capacity using the proposed approaches.

- *Maintaining the VoIP quality:* Supporting delay sensitive real-time application such as VoIP over mesh network is a challenge. Due to the changing wireless conditions and channel interference, delay and loss characteristics can vary over time along a multihop path between a source and destination of a voice call. Such variations can severely impact the quality of a voice call. We exploit the knowledge of multiple paths between source and destination nodes to select a path with currently the best delay and loss characteristics. Using a zero overhead label switching scheme, voice traffic is routed along the best path among the available paths. Our experiments show that using the efficient path switching, we can meet the target quality 86% of the time compared with 46% of the time with no switching.
- *Supporting mobility to VoIP client:* Maintaining calls without disruptions for mobile VoIP clients that handoff to different APs during an ongoing voice call is crucial. We design a fast handoff scheme that uses link-layer feedback mechanism to detect client movement and update the routes within the mesh network accordingly. In addition, to prevent packet losses during a handoff, we buffer voice packets at the old AP and reroute these packets via the new AP to the client. Our experiments with Skype over multihop mesh demonstrate almost near to zero disruption in VoIP quality during handoff.

B. Roadmap

In the next section, we discuss the existing work related to VoIP and mesh network. In Section III, we present the VoIP quality measure and describe our mesh architecture for deploying VoIP services. Sections IV–VI present the proposed performance optimizing techniques to achieve our stated goals for efficient VoIP service deployment. We start with capacity improvement techniques in Section IV presenting experimental results alongside. In Section V, we discuss about VoIP call routing for maintaining call quality. Section VI describes the protocols and techniques for supporting VoIP client mobility. Finally, we draw important conclusions of this work in Section VII.

II. RELATED WORK

A significant research has been conducted in the area of 802.11-based wireless multihop mesh network. Study has been conducted to understand the capacity of multihop network, as presented in [2]. Research on improving the end-to-end performance of applications on multihop network by employing multiple radios has been considered in [3] and [4]. Further work on finding the best routing metric has been studied in [5]. Performance of real-time applications over 802.11 was studied initially in [6] and [7]. Reference [8] provides a decentralized service provisioning framework for mobile mesh networks. In this paper, we evaluate and optimize the performance of VoIP over a multihop wireless mesh network.

In the recent past, with growing importance of VoIP application, several research works have addressed the performance issues of supporting VoIP over Internet. The use of switching among multiple paths to reduce delay was proposed in [9] and recovering from packet loss was proposed in [10]. These strategies were used for delivering VoIP using overlay network. In supporting VoIP over a wired network like the Internet, the major factor affecting the performance is path delay, as for good quality, VoIP requires one-way delay to be less than 200 ms. In supporting VoIP over wireless network, the main factors affecting performance are the low capacity and packet loss.

Studies focusing on VoIP over 802.11 in [11] and [12] considered the delay and loss characteristics under point coordination function (PCF) and distributed coordination function (DCF) mode. Another recent work on VoIP over WLAN in [13] presents analytical studies on the number of calls that can be supported in a single-hop WLAN. We do a similar study over multiple hops. The study also reports that increasing the payload per frame increases the number of supported calls. Our work uses this observation for a multihop scenario to design the voice packet aggregation scheme.

Several performance optimization schemes have been proposed to improve the VoIP quality over a WLAN [14], [15]. Authors in [14] proposed the use of a dedicated queue to provide higher priority to VoIP traffic over data traffic, while in [15], packet aggregation is used to increase capacity.

Packet aggregation schemes for improving the end-to-end throughput for applications on various types of multihop networks have been addressed in [16]–[20]. In [16], the authors propose packet concatenation mechanism to reduce the 802.11 medium access control (MAC) overhead between a single

source-destination pair. Reference [17] describes the 802.11 protocol overhead and the importance of the packet aggregation in the ad hoc network. They propose two aggregation algorithms, forced and adaptive. While the forced algorithm adds additional delay at every intermediate node making it difficult to attain the delay bound for real-time application, the adaptive algorithm usually does not have enough packets to combine packets, which results in inefficient use of the bandwidth. Reference [18] proposes packet aggregation schemes for efficient transport of voice signals over a generic IP-based networks in order to provide VoIP services, but it does not take care of the characteristics of the wireless mesh network. Reference [20] models a packet aggregator for multiple VoIP streams in wireless networks and shows the relationship between the number of VoIP sources, output link rate, and path latency. However, the algorithm to aggregate VoIP flows creates the bigger packets at the prescribed rate, which is not adjustable for the network situation. This work extends on [19], which provides some preliminary results on path adaptation and packet aggregation techniques to improve quality and capacity of VoIP calls in mesh networks.

A VoIP application over wireless mesh network needs to support client-side mobility. The quality of a VoIP call should not go down as the client moves between the APs of a mesh network, causing handoffs. The WLAN research community has addressed the question of fast layer-2 handoffs in WLANs. The issue here is to save channel probing times to determine the best AP to handoff to. See, for example, [21] and [22]. Authors in [23] have independently considered efficient layer-3 handoffs for WLANs, but only in the context of standard-compliant mobile IP [24]. Our work employs a flat routing architecture similar to that proposed in [25] for reducing voice call disruption time during handoff and the layer-3 handoff latencies obtained here are much superior relative to the above reported studies.

III. VOIP SERVICE IN MESH NETWORK

A. VoIP Performance Metric

One of the popular voice encoders is G.729, which sends 50 packets/s of 20 bytes each. It is used by some available 802.11 VoIP phones (such as the Zyxel Prestige), and we emulate it in the testbed using constant-bit rate (CBR) UDP traffic. No silence period is considered, although a 30% utilization increase is generally expected when accounting for periods of silence when no packets are sent (the Zyxel phones and the Skype application also do not use silence suppression). To measure the quality of a call, we used a metric proposed in [26], which takes into account mouth to ear delay, loss rate, and the type of the encoder. Quality is defined by the R -factor, which should provide a value above 70, for medium quality

$$R = 94.2 - 0.024d - 0.11(d - 177.3)H(d - 177.3) - 11 - 40 \log(1 + 10e)$$

where

- $d = 25 + d_{\text{jitter_buffer}} + d_{\text{network}}$ is the total ear to mouth delay comprising 25 ms vocoder delay, delay in the dejitter buffer and network delay;

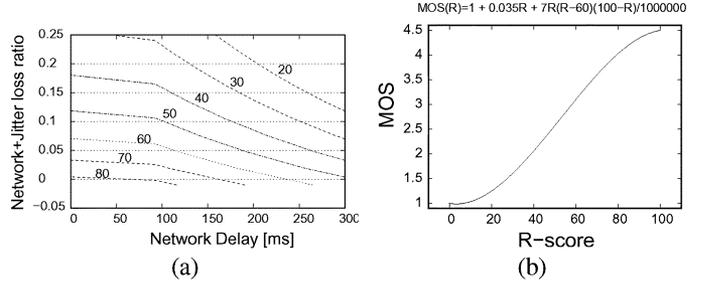


Fig. 3. (a) R -score for 60 ms jitter buffer. (b) Mean opinion score (MOS) as a function of R -score.

- $e = e_{\text{network}} + (1 - e_{\text{network}})e_{\text{jitter}}$ is the total loss including network and jitter losses;
- $H(x) = 1$ if $x > 0$; 0 otherwise is the heavyside function;
- the parameters used are specific to the G729a encoder with uniformly distributed loss.

The constants consider the delay introduced by the encoder for its lookahead buffer, and the delay introduced by the jitter buffer. We considered a jitter buffer of 60 ms, which has two contradictory effects: it increases end-to-end delay, therefore participating to the reduction of quality, but it reduces the jitter, which has an overall better effect. The R score is finally computed only from the loss and the delay in the network, which can be measured directly in our testbed. In order to emulate the behavior of the jitter buffer, we assume playout starts at the destination when packet 4 arrives (60 ms jitter buffer = 3 packets). Therefore, all the deadlines for the packets at the receiving side are established at this point. Loss in the jitter buffer is computed as the fraction of packets which do not meet their deadlines. In order to compute loss probabilities and average delay in the network, all packets from all flows in an experiment are considered together by macro-averaging.

Fig. 3(a) shows the values of the R -score with respect to network delay and total loss for 60 ms jitter buffer and 25 ms vocoder delay.¹ To obtain a score of 70, the network has to deliver all packets in less than 150 ms, or deliver 98% in less than 100 ms. From the figure, we can see that the quality is sensitive to even a couple of percents of loss, whereas the delay tolerates differences in tens of milliseconds. Loss also has a high variance in 802.11, as it depends on the quality of the channels and the cards, and on interference from external or internal sources. In a multihop setup, end-to-end loss is difficult to control and maintain under 2%, when individual nodes exhibit rapid large variations. Using the retry mechanism of 802.11, this loss can be reduced at the cost of increasing delay.

B. VoIP Service Deployment

In our current deployment, we consider the mesh as a multihop extension of the AP infrastructure. It is useful to use the concept of a layer-2 switch to see the entire mesh as a single element that switches packets between its ports. A port is in fact a mesh node which has at least two interfaces: one in ad hoc mode for the backhaul in the mesh, and one in infrastructure mode to connect to clients.

¹The relation between R -score and MOS is shown in Fig. 3(b), and explained in more detail in [26].

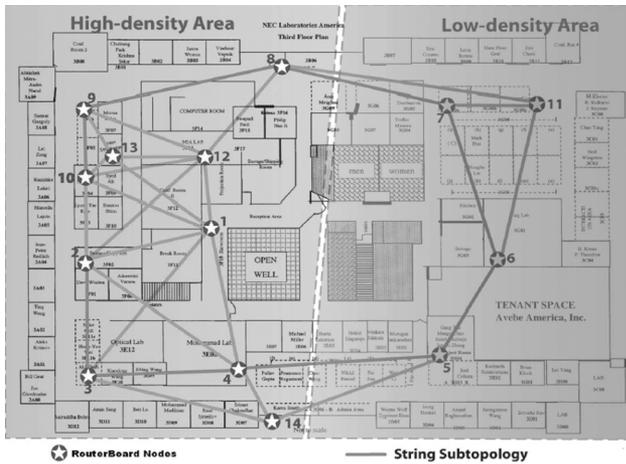


Fig. 4. 14 node testbed in a 70 m × 55 m building.

These clients can be VoIP wireless phones, softphones running on laptops, or handhelds. These clients see the mesh as a switch or hub in the sense that they are not concerned with the internal routing of the mesh. Implementation of the mesh is based on IP, even if it offers a layer-2 abstraction outside. The clients can have connections across the mesh to other wireless devices, as shown in the Fig. 1, to the institution Intranet, to other wired VoIP phones, out to the Internet with the help of a session initiation protocol (SIP) server, or to the public switched telephone network (PSTN). In the our specific deployment, we use Asterisk SIP server and private branch exchange (PBX) for supporting SIP phones and routing the VoIP calls to the PSTN.

Another fundamental design goal that we pursue is *client-side transparency*. The client mobile stations are unaware of the mesh networking backbone. They view the network as a conventional WLAN spread out over an extended geographic area. Thus, the clients still associate with an AP using a traditional association mechanism in WLANs. When the client moves and reassociates with a different AP, a layer-2 handoff event occurs that in turn triggers appropriate routing updates in the mesh network backbone. Thus, the handoff process involves both layer-2 and layer-3 procedures. We describe how the layer-2 and layer-3 handoffs work together efficiently, and present design choices for the layer-3 handoff process—one using a mobile IP [24] like solution called *transparent mobile IP* (TMIP) [27] and the other using a “flat” routing protocol based on link-state routing. We also present a detailed performance evaluation of the handoff latencies in both layers and their impact on VoIP application performance.

C. Testbed Hardware/Software Configuration

The testbed consists of 14 nodes based on the Routerboard 200 series processor boards with 256 MB of RAM, and 512 MB of compact flash. Each node is equipped with two 802.11b wireless interfaces (supports up to 4 mini-pci cards) and has an open slot for a third one (PCMCIA 16 bit). The testbed is spread over the third floor of NEC Research Laboratories, Princeton, NJ, and the layout is shown in Fig. 4. A high-density area on the left side of the building provides a proximity of nodes which allows the study of interference, and a lower density area to allow for

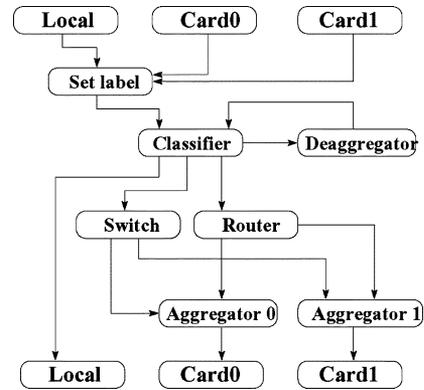


Fig. 5. Click node component for label switching, routing, and aggregation.

longer paths in the network. The wireless cards are running at the fixed rate of 2 Mb/s for providing more stable results for the indoor setting. Each node operates with two interfaces: one that is used to get client traffic from the VoIP 802.11 phones, and the other one for backhaul in the mesh. We generated traffic locally at the nodes for conducting the performance experiments.

Some of the experiments are done on the simulator. We used the ns-2 simulator with 11 Mb/s links. All the nodes are on a single channel. The topology and the traffic pattern for experiments are stated with their descriptions in following sections.

D. Mesh Node

In order to provide routing, forwarding, and other VoIP specific services, we used the click router [28] on each mesh node. The router architecture and configuration used for supporting VoIP is shown in Fig. 5. We implemented label-based forwarding and routing for voice packets. In this router configuration, when a packet is received from the cards, it may get labeled if it is a voice packet and if it needs to be routed over mesh network. For test purposes, we also allow labeling of traffic that is generated locally at the node. To support VoIP traffic, packet aggregation service (discussed in Section IV) is implemented which encapsulates multiple small VoIP packets into a single large packet and forwards it. For each interface, we have a corresponding aggregator, as shown in Fig. 5. Similarly, there is deaggregator to decapsulate the aggregated packet into the original VoIP packets. The most important component in the mesh node is the classifier that decides: 1) whether a packet is destined for the local machine (signaling, aggregated packets); 2) has to be routed (best effort, signaling); and 3) switched (voice). For aggregated packets, after the decapsulation, the resulting packets are fed back to the classifier.

E. Label-Based Forwarding

In order to support the proposed path switching that we describe in Section V, we implement label-based forwarding to achieve fast path switching. For labeling the IP packets, we use TOS field of each IP packet that provides 255 labels at each node. For packets with labels other than zero, label-based forwarding is used. On the other hand, packets with label zero follows underlying routing protocol [destination-sequenced distance vector (DSDV)] used in our mesh network. In order to perform label-based forwarding, each node maintains an addition

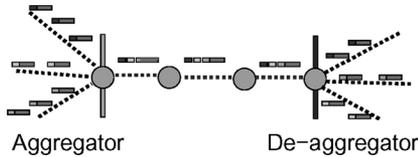


Fig. 6. Aggregation merges small voice packets from different calls into larger packets to improve channel utilization.

table. An entry in this table has the form: (*in_label*, *out_label*, *interface*, *gateway*).

Any packet that arrives on an interface and has a nonzero label (*in_label*) is stamped with the corresponding *out_label* and sent to the *interface* to be delivered to *gateway*, which is the next hop in the path. Once the outgoing label is set (for the switching), and the outgoing interface is determined (for both routing and switching, by their respective lookups), packets are pushed to “pull” queues associated with each interface. These queues perform the aggregation of packets that have the same next hop (only voice and probe packets). The meaning of “pull” in Click terminology is that these queues are queried by the cards when the transmission is possible, so the waiting time in these queues is used for the purpose of the aggregation.

IV. IMPROVING THE VOIP CAPACITY

A. VoIP Packet Aggregation

As most vocoders use samples of 10–100 ms, a mesh node is expected to get a large volume of small packet traffic. However, 802.11 networks incur a high overhead to transfer one packet, therefore small sizes of packets reduce the network utilization. The basic idea of aggregation (Fig. 6) is to combine together several small packets that arrive at the ingress node and forward them with one IP, MAC, and physical (PHY) header across the air.

The problem with small payloads is that 802.11 MAC spends most of the time sending headers and acknowledgments, waiting for separations distributed interframe space (DIFS) and short interframe space (SIFS), and contending for the medium. For example, in order to send a 20 byte VoIP payload, a 60 byte packet is assembled from 20 bytes IP header, 12 bytes RTP header, and 8 bytes UDP header. This takes 43.6 μ s to send at 11 Mb/s, but MAC header and physical headers, trailers, interframe periods, and acknowledgment (ACK) need a total of 444 μ s. That however does not consider the amount of contention which is on average of 310 μ s, and increases exponentially with contention. This way, to send a 20 bytes payload takes 800 μ s at 11 Mb/s, yielding approximately 1250 packets/s, which for a vocoder like G.729a corresponds to only 12 calls that can be supported. At 2 Mb/s, a similar computation leads to eight calls. Applying VoIP packet aggregation can significantly reduce the overhead and increase the number of supported calls. However, a common problem in packet aggregation is that it increases packet delay which can possibly reduce its suitability for delay sensitive VoIP services. Still, with an efficient aggregation technique, the resulting benefit from reducing MAC protocol overhead can easily outweigh the disadvantage.

In a mesh network scenario, one can consider two naive mechanisms for aggregation: *end-to-end* and *hop-by-hop*. In the end-to-end scheme, the aggregation is done only at the ingress node for all flows routed for a common destination. A forced delay is added at the ingress to perform packet aggregation. However, its applicability is limited to availability of enough packets for aggregation under the forced delay budget. In the hop-by-hop scheme, packets are aggregated and deaggregated at every hop by adding a forced delay at every hop. This scheme is oblivious to source or final destination of a given packet and can result in adding cumulative delay to all packets over multiple hops. The hop-by-hop scheme also suffers from extra complexity at each node for executing aggregation/deaggregation on every incoming packet.

For suitability in mesh network, we propose the *accretion aggregation algorithm* (Algorithm 1). This is a hybrid of the above two naive algorithms, where forced delay is introduced at ingress nodes. In addition, at intermediate node, medium access queueing delay is used for possible aggregation resulting in no extra delay on any packet. At the ingress, the forced delay depends on the budget allowed by the probing of available paths. At forwarding node, the aggregation becomes more effective in high load—the case where the need becomes natural for saving bandwidth.

Aggregation Logic for Ingress Nodes

```

P—packet being queued at a node;
A—aggregation packet being prepared;
minPackets—number of packets from the same flow
that have to be aggregated at the ingress
(corresponds
to the delay budget available for the flow;
MTU—maximum transmission unit, in voice packets;
find queue of P;
1: if size(queue) > minPackets
    add all packets from flow(P);
    if size(A) < MTU
        find a queue with the same dest
        go to 1;
    else
        send A directly to destination;
else
    if size(A) < MTU
        add pkts w same nexthop as p;
    else if aggregation timer is expired
        aggregate all packets from the queue
        of which timer is expired also;

```

B. Evaluation of VoIP Packet Aggregation

In *ns-2*, we simulated a string of six nodes with and without aggregation, and verified the results against a similar string in the testbed. In Fig. 7, we find that for the nonaggregated traffic, the simulation matches the testbed results in most points, but

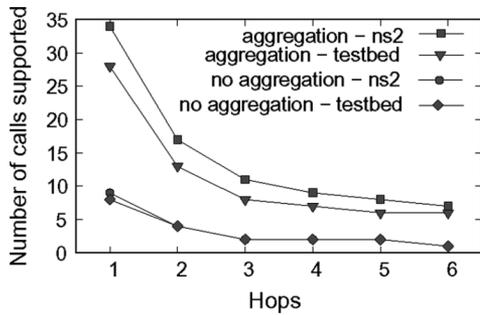


Fig. 7. Aggregation on a string: ns-2 versus testbed.

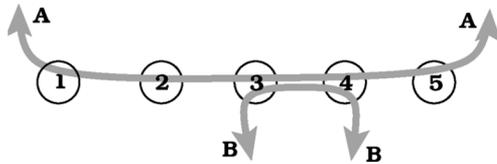


Fig. 8. Aggregation introduces only controlled delay at the source of flows. Intermediate nodes do not delay packets to improve aggregation, but use “natural” waiting required by MAC under load.

for aggregated traffic the testbed performs worse. The cause of this was identified in the fact that the capacity of some of the hops in 2 Mb/s mode was less than the optimal 1.7 Mb/s. The first hop, for example, was measured to provide a capacity of only 1.38 Mb/s, with 1500 bytes packets, accounting for the difference between the aggregated calls supported—29 in testbed versus 34 in *ns-2*. However, knowing that our simulation setup performs reasonably close to the testbed, we obtain most of the next results in simulation only in order to have better control and repeatability.

One of the main claims of our accretion aggregation method is that it does not introduce additional delay by using the wait for the MAC availability to club together packets destined to the same next hop. To verify this claim, we place five longer calls indicated by A in Fig. 8 and five short calls, indicated by B. Aggregation is performed for each group of flows independently at the source by introducing a controlled delay of 80 ms. This means that packets from A are not merged with packets from B during their common hop 3–4. The network time for flows A is about 61 ms in the absence of B and increases to 89 ms after B is added. This is normal, considering the increased interference for all the nodes and the longer queues at nodes 3 and 4. After we enable aggregation between A and B at hop 3–4, the network time for flows A decreases to 79 ms. Not only a delay is not added to the long flow, but the creation of larger packets reduces the contention for the hop 3–4, thus reducing the load on the network. This experiment reveals some interesting properties of our aggregation scheme: first, it only kicks in at higher load when waiting in the queues can be used to group packets with the same next hop. Second, it is completely distributed inside the mesh. The endpoints need to specify initial delays depending on their time budgets, but the intermediate nodes have the simpler task of looking for packets with the same next hop. Third, and the most important, **short flows do not delay long flows for the purpose of aggregation**. It is true that the mere existence of short flows increases load, and therefore delay for long flows,

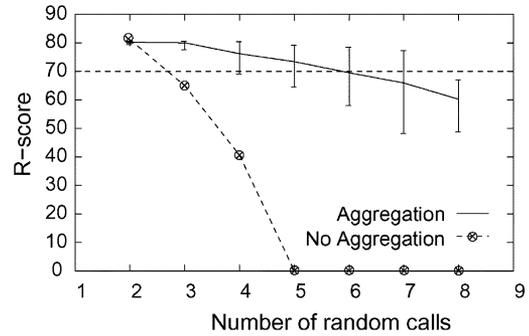


Fig. 9. Aggregation performance for random calls on a string topology.

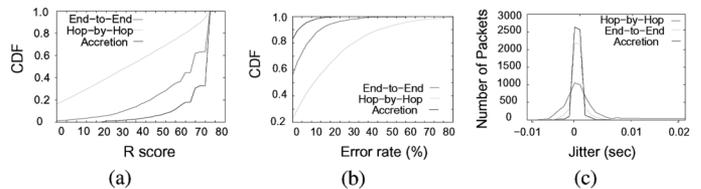


Fig. 10. Traffic statistics of three aggregation algorithms on three calls over four-hop leaf nodes, forced waiting time = 40 ms, G729 a, 11 M. (a) Comparison of R score. (b) Packet loss ratio. (c) Jitter distribution.

but that is inherent to the behavior of the shared medium. In the network, distributed aggregation is a method that reduces load without impacting the network time of flows.

In another experiment, we consider a more randomized scenario in terms of sources and destinations of calls. In an eight-hop string topology, calls between random sources and random destinations are placed. Considering ten random situations for each configuration of two to eight calls, we compute the minimum, maximum, and average value of the R -score for the offered load. When aggregation is not used, minimum and maximum only take values 0 or 82, and are not indicated in the figure. In Fig. 9, we see that if we consider an R -score of 70 as a threshold of call quality, then aggregation more than doubles the capacity, even with randomized traffic.

When the mesh network is used as an extension of the AP infrastructure, a popular pattern is to have voice calls forwarded to the wired infrastructure or to the PSTN. In this case, paths from the clients all lead to a common root which provides access to the wired leg of VoIP. We simulated a tree with 15 nodes with maximum four hops from root to leaf node. We find that on this tree, the maximum number of supportable calls is 16, 16, 24 calls in end-to-end, hop-by-hop and accretion algorithm, respectively.

Fig. 10 shows the call statistics for R -score, packet loss ratio, and jitter. The CDF on the R -score shows that proposed accretion mechanisms results in higher probability to meet the required R -score. The cumulative distribution function (CDF) in Fig. 10(b) on loss ratio shows that accretion mechanisms results in lower packet loss. Finally, Fig. 10(c) shows that the number of packets maintaining near to zero jitter (in seconds) is higher for the proposed algorithm. The above results demonstrates: 1) end-to-end algorithm makes poor use of bandwidth with fixed aggregation packet size and 2) hop-by-hop algorithm undergoes large packet jitter, which increases jitter drop ratio. Accretion algorithm provides low delay variation, larger aggregation packet.

C. Header Compression

Header compression is a complementary scheme related to aggregation. The usage for header compression is motivated by the fact that: 1) the VoIP payload is typically compressed at the application layer, which means another compression does not help reduce the payload size; 2) the headers occupies a large portion of the packet; and 3) the headers have significant redundancy. Packet headers with redundancy may be reduced through compression techniques as has been done with great success for cRTP or ROHC. The compressor and the decompressor should operate in synchronization, either implicitly (optimistic mode) or explicitly (acknowledged mode). However, the 802.11 wireless media shows high error rate, the links are fluctuating, and the paths may be changed frequently between two nodes. If the context between the compressor and the decompressor is out of sync, they need to exchange the information to recover their inconsistency. Such recovery requires time, complexity and bandwidth, and can lead to data loss.

For a VoIP flow RTP/UDP/IP headers take 40 bytes, but only 12 bytes of them changes when the packets get routed. Schemes such as cRTP or ROHC aims at compressing the 40 bytes into a 2 byte connection ID, but they are appropriate for one link only. In order to emulate a simpler scheme that only transmits the changing fields, we reduce the header from 40 to 14 bytes so that VoIP protocol overhead coming from large header size can be relieved.

In addition to the above compression logic, we propose *zero-length header compression* (ZHC) algorithm for redundant header elimination algorithm that leverages the VoIP packet aggregation mechanism we discussed earlier. It is to be noted that ZHC does not require context synchronization between two nodes.

The compressed headers from the compressor are needed to restore the original packet headers from them, as the index, at the decompressor. However, if the header compression method is used with the aggregation mechanism, there is no need of the headers from the second packet until the last packet, with the fresh header of the first packet from the same flow. The decompressor has enough information to restore all headers from the first fresh header, which does not depend on synchronization of the compressor/decompressor. Loss of one aggregation packet is isolated to packets lost in aggregation packet without requiring any extra information to restore the decompression.

D. Evaluation of Header Compression

We consider a linear topology and evaluate the benefits of header compression. Fig. 11 shows the relative benefits by incorporating aggregation and aggregation with ZHC. We observe that header compression results in a significant increase (up to two times) in the number of calls supported.

We next consider a tree topology with 16 nodes and maximum 4 hops. In this tree topology, we allow five VoIP calls from each leaf node to the gateway node. We consider the static header compression (SHC) which reduces the 40 bytes header to 12 bytes header and the proposed ZHC. It is to be noted that the ZHC only works in conjunction with aggregation. The number of supported VoIP calls for R -score greater than 75 is shown for tree and a 4×4 grid topology in Table I.

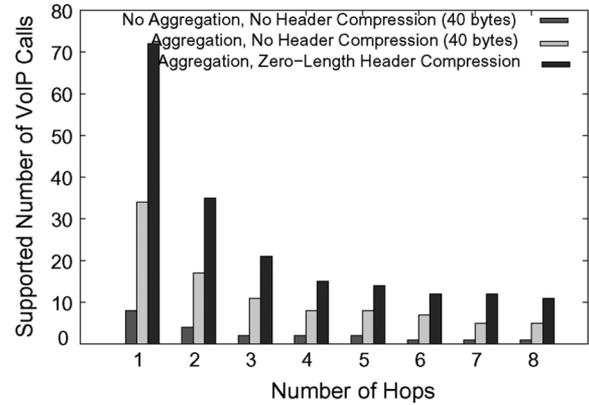


Fig. 11. VoIP performance with/without aggregation and header compression.

TABLE I
COMPARISON OF VOIP PERFORMANCE AGGREGATION/HEADER COMPRESSION STRATEGIES

Algorithms	Tree	Grid
No Aggregation + No HC	0	0
No Aggregation + SHC	0	6
Aggregation + SHC	5	15
Aggregation + ZHC	6	17
Aggregation + ZHC + SHC	7	18

V. MAINTAINING THE VOIP CALL QUALITY

In this section, we describe an adaptive path selection technique that we propose for efficient support of VoIP call quality over mesh network.

A. Routing

It is inherently a challenging problem to design good routing schemes for supporting real-time applications over wireless mesh. Existing work such as [3] and [4] focused mostly on the TCP throughput. The difficulty in routing arises from the number of factors on which a good route depends such as: 1) channel quality; 2) dynamic condition due to interference caused by traffic inside and outside the mesh network; and 3) traffic load on routes in the interference range. Jointly considering the routing and channel assignment problem [3] is hard, as well for wireless mesh network. The main limitations imposed by routing voice calls is that routing decision cannot be taken dynamically in reaction to changing conditions of the mesh network as that leads to delay in call admission, switching in handoff and packet loss and resulting interruptions in voice playback.

In our setup, we need to perform call admission within seconds of placing a voice call. This is achieved by using precomputed paths, even if the solution is suboptimal. Another important factor in this decision is that the call admission process should preferably be distributed. To achieve the above design goals, our voice call routing approach consists of two component: 1) route discovery and 2) adaptive selection of paths by the route discovery process. To choose paths, we opted for a solution based on probing in order to cumulate all factors (interference, load, channel quality), which are otherwise hard to account for.

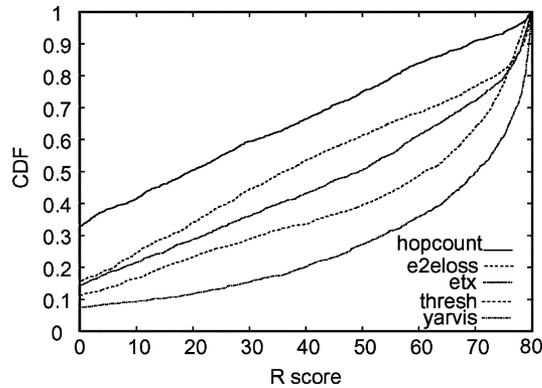


Fig. 12. DSDV has low performance for one call across the testbed using various metrics.

Route Discovery: For route discovery, DSR would have been a good choice, but we need to maintain multiple source destination paths, and the implementation available with Click performs poorly on our platform in terms of CPU usage and responsiveness. DSDV is the second option, but being distance vector based, it has the undesirable effect of frequently updating paths in the middle of the call. We experimented with several metrics: hop count, end-to-end loss, quantized end-to-end loss, a threshold-based loss metric, and ETX [29] (which is also based on loss). All these metrics provide unacceptable performance (Fig. 12) for voice in our testbed. This is mainly because of the default behavior of DSDV, which is aging routes and is always ready to accept new routes. The wireless environment in the building is also a factor that degrades the performance of routing. Even the fairly stable hop count metric exhibits a lot of route variance because occasionally packets may travel across the entire building long enough to make the routing algorithm believe a one-hop route is available. When this long link becomes unavailable, or is beyond a certain age, DSDV is looking for alternatives based on its current metric. It is in this case, during switching between routes when voice packets are lost and quality of voice degrades.

We opted instead for using DSDV to collect popular routes which are then pinned down and used with label-based routing. Each node is maintaining the top five most popular routes for each destination based on measured statistics. We ran DSDV with various loss-based metrics (including ETX, end-to-end loss, quantized end-to-end loss). We retained the five most popular paths chosen during last 24 hours. These top five paths were almost the same for the different metrics, although with differing popularities from one metric to another. This means that from the loss point of view, similar set of paths shows a consistently better quality over time.

Adaptive Path Selection: In order to use the paths for voice transport over our wireless mesh network, we pinned down the paths using label-based forwarding, as described in Section III-C. We chose a pair of nodes A and B at extremities of the building to maximize distance in hops and path diversity. To make use of the alternate paths and the possibility of fast switching without losing any packets, we implemented a simple strategy in which one of the nodes monitors all the paths with a low bandwidth ping (1 packet/s). When a call is placed, the

TABLE II
ADAPTIVE PATH SWITCHING VERSUS FIXED PATHS

Path	R_{avg}	$\text{cdf}(R > 70)$	Path Usage	R_{avg} of used path
Adaptive	71.2	0.86	-	-
4	40.4	0.48	47%	72.4
5	56.3	0.69	40%	73.2
6	17.1	0.19	11%	70.8
7	28.7	0.33	1%	5.1
8	6.5	0.07	1%	52.8

five paths are probed with a low bandwidth probe to evaluate the delay of each path, which is the most critical component. The probing traffic has the same characteristics and treatment as the voice traffic, namely, it can be aggregated or delayed based on localized conditions in the network. The size of the packet is chosen to be the same as the voice packet so that round-trip times are good estimates that can be extrapolated to voice packets. We compute an R -score estimate of the low rate probing traffic on each available path using its running averages of the delay and loss. When the exponentially averaged R -score of the probing traffic stays under 70 for more than a few seconds, the decision is taken to switch the actual voice traffic to another path based on the current estimated R -score of alternatives.

B. Evaluation of Adaptive Routing

In order to evaluate the routing strategy on our mesh testbed, a single voice call is run for 2500 s between nodes A and B, and the R -score is recorded every second. Each of the five available paths is measured independently. To create a repeatable pattern of disruption similar to office use of laptops, we selected a number of jammers outside the testbed that follow a predefined random, but fixed, sequence of traffic on the same channel as the testbed.

In Table II, we have the path labels in the first column, from 4 to 8. The first line corresponds to the adaptive strategy. The second column of the table shows the average R -score achieved, and the third column the fraction of time when $R > 70$. By using the available alternate paths, the simple adaptive strategy is able to route the voice traffic around the interference and congestion providing a good R -score 86% of the time, with an overall average of $R = 71.2$. The fourth column shows how paths are used by the adaptive strategy, and the fifth column the average R -score obtained by the respective paths on behalf of the adaptive strategy. Most service is provided by just three paths, which could help in reducing the amount of probing traffic to only proven quality paths. Probing of additional paths may be enabled only when the reduced path set does not provide the required quality.

Knowing that the R -score is a function of loss and delay, the question is which of the two factors is more important in our testbed. The network loss is less than 1/2% for most paths, except one, so delay must be the deciding factor. In Fig. 13(a) and (b), the delay distribution histogram is shown for one of the participating paths and for the adaptive case. Times over 200 ms are collapsed in the rightmost bin. These distributions confirm that the quality of our paths is dominated by delay. The source of this delay is cross traffic (from the

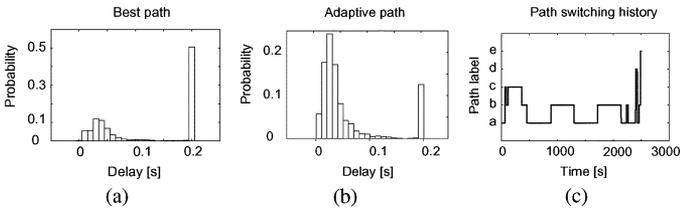


Fig. 13. Delay distribution adaptive versus fixed. Only 12% of the time the delay is greater than 200 ms when the path is adaptive. A fixed path provide delays greater than 200 ms 50% of the time.

jammers) and channel conditions (802.11 retry is used at the default of 7). Since an unloaded path experiences about 2–3 ms per hop, and our paths have four and five hops, the measured delay was confirmed to be in the range 8–15 ms, which is almost negligible compared with the delays experienced by the voice traffic during the experiment. Fig. 13(c) shows which paths were used during the experiment, corroborating the figures from Table II.

VI. MOBILITY SUPPORT FOR VOIP CLIENT

For providing complete client-side transparency, some of the mesh nodes have one of the interfaces operating in the *infrastructure* or Host-AP mode. While other interfaces on these nodes operate in ad hoc mode and provide connectivity to the mesh backhaul. This choice enables us to design the system without any specialized software on the mobiles. Use of ad hoc mode will need client device configuration with appropriate software. The client’s view of the network is still that of a WLAN, while the distribution system (DS) connecting the APs is now made of a wireless backhaul network. It is now sufficient to use the underlying “handoff” capability of the 802.11 client devices to handle mobility.

In this work, we particularly focus on the network-layer handoff issue. Link-layer handoff optimization has been considered in [21], [22], and [30] for minimizing latency in handoff between APs connected by a wired LAN.

In the mesh architecture the nodes form a multihop network, routable at the IP layer. This gives rise to a mobility management problem—how to deliver frames destined to a station when its point of attachment to the mesh network (i.e., the AP) has changed. Two broad approaches are possible that we both implement in our testbed and compare.

Mobile IP-Based Scheme: The first approach uses a technique similar to mobile IP [24], where each station has a unique “home” location or a home AP. The network implementing the DS keeps track of the mobile stations. Packets destined for the station are still delivered to the “home” AP for propagating to the mobile station. It is now the home AP’s responsibility to forward the packet to the AP the mobile station is currently “visiting.” This is achieved by a protocol called TMIP [27]. The significant difference from the standards-compliant Mobile IP is that the mobile station does not need to implement any specific protocol. This preserves the transparency we desire. There is a centralized server in the network called *mobile location register (MLR)*, which keeps the information about the “home” AP for every mobile station. When the mobile hands off to any “foreign” AP, the foreign AP sends a query to the MLR to find out

about its “home” AP. The foreign AP then notifies the home AP about the new endpoint of the mobile with a message handshake, and adds a new, one-hop route to the mobile. It also sends a gratis ARP response to the mobile so that the mobile updates its MAC address for its default gateway (which is still the home AP) and makes it the same as the foreign AP. Beyond this point, packets directed to the mobile are intercepted by the home AP and “tunneled” (using IP-in-IP encapsulation) to the foreign AP. The communication from mobile AP, on the other hand, can proceed in the normal fashion without involving the home AP. Note that TMIP makes it possible that the mobile station keeps the original IP address even in its “foreign” location.

While the TMIP approach is straightforward, the forwarding path for the mobile is clearly not optimal due to the so-called “triangular routing” scenario.

Flat Routing-Based Scheme: The approach that we promote in this paper is to use a full-fledged multihop routing infrastructure in the network of APs in the DS. The routing infrastructure is “flat”; the routing tables in the APs contain the IP addresses of all the mobile stations in the system. Optimizations are possible for very large-scale networks to limit the size of the routing tables, though we do not discuss these here. The basic idea of maintaining the routing infrastructure is to use any handoff as a trigger to generate and propagate necessary routing updates. Thus, the network-layer handoff consists of completion of notifications for the TMIP case, and convergence of routing updates for the flat routing case.

In the mesh testbed, we have chosen the DSDV routing protocol. The DSDV protocol runs on all backhaul interfaces at every node. The mesh node does not run DSDV on its client-side interface that operates in infrastructure mode (the interface the client associates to—typically `wlan0`) as the client is unaware of the routing. The link between the AP and mobile station is treated as an *external route* to the mesh network. The AP advertises such external routes within the mesh network.

Whenever a mobile station associates with an AP, the Host AP driver sends an association signal to the DSDV routing module, which deletes all preexisting routes to this station and adds a “direct” route to the client via its `wlan0` interface. This “external” route information is encoded as a triggered route update message and broadcasted in the network via the DSDV protocol. All APs, on receiving the route update message, delete all pre-existing routes to this station and add a new route via the AP to which it is currently associated.

Triggering Network-Layer Handoff: A few words are due here about address assignment and how it is related to triggering network-layer handoff. When a new mobile client joins the network, say by booting up and associating with one of the APs, the client acquires an IP address via DHCP from the address pool of this AP (each AP maintains an independent address pool). Let us denote this AP by AP1. The mobile uses AP1 as its default gateway. AP1 maintains a mapping of IP address to MAC address of mobile clients in an *IP-to-MAC address mapping table*. AP1 then adds a host-specific route to this mobile in its kernel routing table. At this stage, the mobile client has complete uplink connectivity. AP1 then advertises the new route to the mobile to all other nodes in the mesh network through a triggered route update via the DSDV protocol. The MAC address

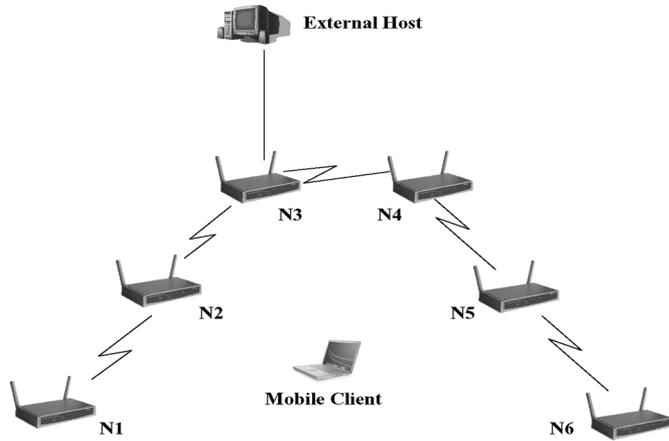


Fig. 14. Mesh topology used for handoff performance evaluation. Mesh nodes $N1$ to $N6$ have one interface operating in AP mode and another interface dedicated for backhaul.

of the mobile is included in the update message so that all APs in the mesh network can add the IP-to-MAC mapping in their own IP-to-MAC address mapping table. Downlink connectivity is available to the mobile at the end of the route update, as at this point all mesh nodes in the network have a host-specific route to the mobile client with AP1 as the last-hop node in the route.

The IP-to-MAC mapping is required to be distributed for a reason. When the mobile reassociates to another AP in future (say, AP2), AP2 must be able to determine its IP address so that it can send out routing updates appropriately. For this to happen without any involvement from the mobile (this is required for transparency), AP2 must know the mapping in advance.

A. Handoff Performance Evaluation

In order to conduct the experiments, we set up a linear topology, as shown in Fig. 14. In the following, we report the performance of the mesh architecture with the above setup running the flat routing protocol. For comparison, we have also ported TMIP [27] to our testbed.

In this subsection, we report measurements of handoff latency for a mobile station (STA) as it switches its association from one AP (old AP) to another AP (new AP). Handoff begins when the STA loses its association with oldAP (in our case, by receiving a deauthentication frame at old AP) and completes when STA associates with the new AP.

In our architecture with the flat routing protocol, layer-3 handoff starts with the advertisement of a new route to STA by the new AP. The DSDV protocol handles the broadcast of this message in the mesh network. Layer-3 handoff completes when the routes at all APs have been updated to reflect the new AP as the new point of attachment for STA. The handoff delay here depends on the number of route changes and the distance of these changes from the new AP. We have used a linear topology of APs (Fig. 14) to experiment with various distances (in number of hops) of these route changes, while keeping the size of the testbed reasonably small. We think that our experiments are quite comprehensive, as in a deployed mesh network, route changes are unlikely to be any more than a few hops, if we assume that there are no coverage holes. This

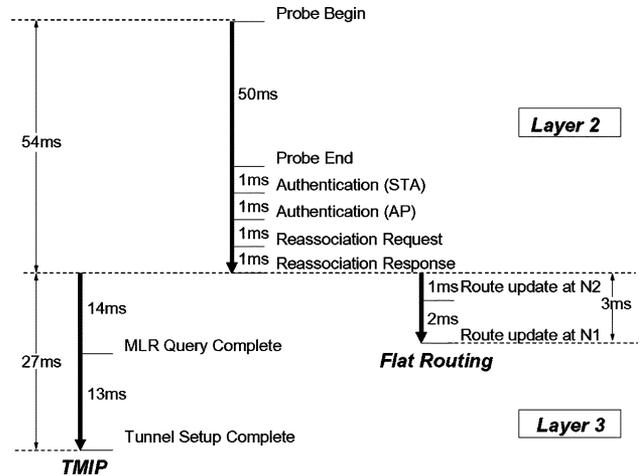


Fig. 15. Timeline describing a typical fast one-hop handoff for flat routing and TMIP schemes when mobile station switches association from $N1$ to $N2$.

is because typically the distance between the new AP and old AP will be less than twice the radio range. This makes the shortest path length in hops between these two APs very small. While route updates may be transmitted network-wide, actual route changes will be limited to within a small neighborhood of newAP, because of the stated closeness of newAP and oldAP.

In case of TMIP, the “foreign” AP (i.e., new AP) for the STA first determines the IP address of “home” AP (i.e., old AP) by querying the central server (which is the AP $N3$ in Fig. 14) that implements the MLR. It then notifies the home AP so that the home AP can tunnel packets to the foreign AP. Layer-3 handoff completes when this procedure is completed.

Fig. 15 depicts the handoff timeline with a representative set of timing measurements. Note that the layer-2 handoff delay is dominated by the probe delay, as explained before and examined critically in recent literature [21], [22], [30]. The timeline shows about 50 ms delay for probing. A single channel is probed. In case the methods used in [21] are applied in our testbed, only one channel will be probed. Recall that the entire network operates on the same channel in this set of experiments. If multiple channels need to be probed, the 50 ms probe time will be multiplied by the number of channels to be probed, and the handoff delay will increase accordingly.

The timeline also shows layer-3 handoff delay for the one-hop route change case. This happens, for example, when STA moves from $N1$ to $N2$. In this case, the handoff completes when the route is updated at $N1$, as the routes do not change in any other AP. This takes just 3 ms. On the other hand, for the case of TMIP, an order of magnitude longer time is taken to complete the MLR query and the notification process with the home AP so that packets can be tunneled.

The handoff latency measurements for handoffs for different hop lengths of route changes is shown in Fig. 16. For these experiments, $N1$ is always the old AP. The new AP is one of the five remaining nodes thus making up to five hop route changes. We evaluated the handoff latencies in presence of network load. We set up a 500 kb/s UDP flow between $N1$ and $N6$ resulting in a total background traffic load of 2.5 Mb/s. Looking at Fig. 16, layer-2 handoff delays are independent of the amount of route

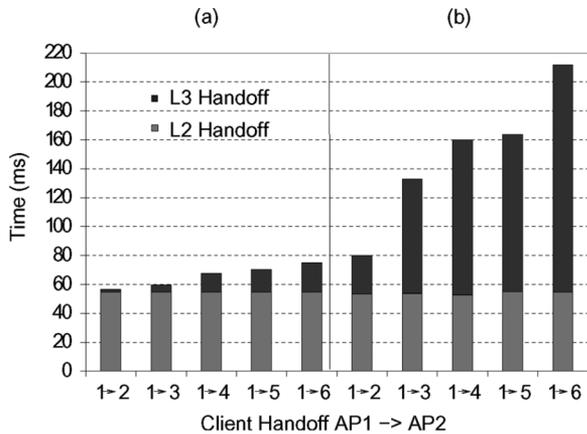


Fig. 16. Handoff latencies for flat routing and TMIP with background traffic of 2.5 Mb/s and client probing on a single channel.

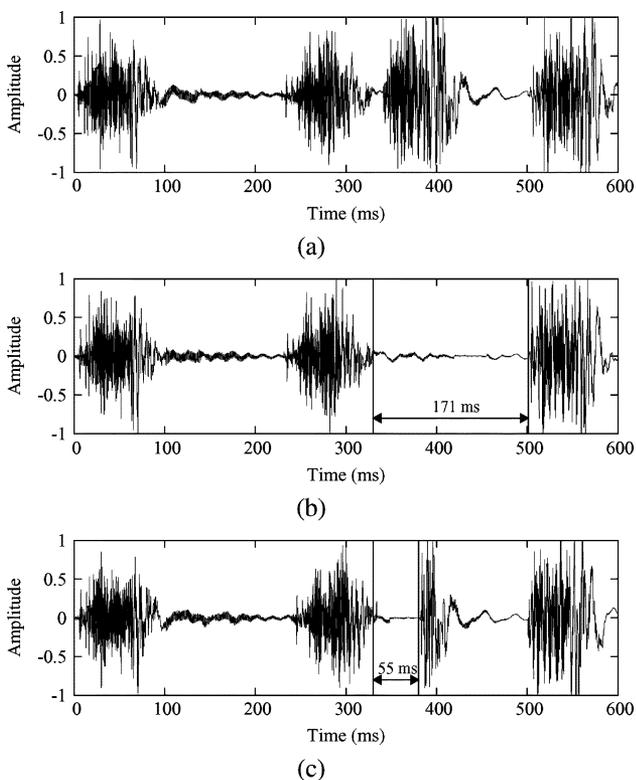


Fig. 17. Skype application audio signal plots recorded at the mobile client. A two-hop handoff is initiated at time 330 ms. (a) Received audio signal with no handoff. (b) Received audio signal during a handoff with TMIP. (c) Received audio signal during a handoff with flat routing.

changes as expected. Layer-3 delays, on the other hand, is proportional to the number of nodes along the path between new AP and old AP in the case of flat routing. In case of TMIP, latencies are much higher as the messages communicated between new AP and MLR and between new AP and old AP have to travel over longer paths.

Next, we analyzed the perceived call quality during a handoff using Skype VoIP application [1]. We ran Skype VoIP call between the external host and the mobile STA, while the STA is roaming. At the external host, the microphone input to the Skype application is fed through another application that plays from a wave audio file, while at the mobile STA, the audio output

is redirected to a file. Fig. 17 depicts the sound signal values recorded during three different scenarios: with no handoff, a two-hop handoff with flat routing scheme and a two-hop handoff with TMIP. Though both handoff schemes incur a call disruption during handoff, flat routing scheme performs significantly better with around 55 ms of silence period.

VII. CONCLUSION

In this work, we take the first step in addressing the practical issues related to deploying VoIP services over a mesh network. Our main contribution is in proposing and evaluating performance optimizing techniques that are crucial for supporting VoIP over mesh network. Based on investigating various design choices and experimentally evaluating the proposed optimization methods using real testbed and ns-2 simulations, we make the following important conclusions. For increasing capacity in terms of number of calls supported, voice packet aggregation and header compression together provide a factor of 3 benefit. In order to maintain the target R -score quality measure of an ongoing VoIP call, fast path switching using label-based forwarding can easily adapt to the fluctuating channel conditions. Low handoff latency can significantly reduce the quality disruption of an ongoing VoIP call during handoff. The combined layer-2 and layer-3 handoff latency using the proposed techniques yields less than 50–100 ms and thereby demonstrates the feasibility of supporting seamless mobility in a wireless mesh network even in the presence of frequent handoffs.

REFERENCES

- [1] Skype, [Online]. Available: <http://www.skype.net>.
- [2] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proc. 7th ACM Int. Conf. Mobile Comput. Netw.*, Rome, Italy, Jul. 2001, pp. 61–69.
- [3] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," in *ACM Mobile Comput. Commun. Rev.*, Apr. 2004, vol. 8, pp. 50–65.
- [4] R. Draves, J. Padhye, and B. Zill, "Routing in multi-hop multi-radio wireless mesh networks," in *Proc. ACM MobiCom*, Philadelphia, PA, Sep. 2004.
- [5] J. P. R. Draves and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *Proc. ACM SIGCOMM*, 2004, pp. 133–144.
- [6] J. L. Sobrinho and A. Krishnakumar, "Real-time traffic over the IEEE 802.11 MAC protocol," *Bell Labs Tech. J.*, pp. 172–187, 1996.
- [7] J.-Y. Yeh and C. Chen, "Support of multimedia services with the IEEE 802.11 MAC protocol," in *Proc. ICC*, 2002, pp. 600–604.
- [8] K. Farkas and B. Plattner, "Supporting real-time applications in mobile mesh networks," in *Proc. MeshNets Workshop*, Budapest, Hungary, Jul. 2005.
- [9] S. Tao, K. Xu, A. Estepa, T. Fei, L. Gao, R. Guerin, J. Kurose, and D. Towsley, "Improving VoIP quality through path switching," in *Proc. INFOCOM*, Mar. 2004, pp. 2268–2278.
- [10] Y. Amir, C. Danilov, S. Goose, D. Hedqvist, and A. Terzis, "1–800-overlays: Using overlay networks to improve VoIP quality," in *Proc. NOSSDAV*, 2005, pp. 51–56.
- [11] M. Veeraraghavan, N. Cocker, and T. Moors, "Support of voice services in IEEE 802.11 wireless LANs," in *Proc. INFOCOM*, 2001, pp. 488–497.
- [12] D. Hole and F. Tobagi, "Capacity of an IEEE 802.11b wireless LAN supporting VoIP," in *Proc. ICC*, 2004, pp. 196–201.
- [13] S. Garg and M. Kappes, "Can I add a VoIP call?," in *Proc. IEEE Int. Conf. Commun.*, Anchorage, AK, 2003, pp. 779–783.
- [14] J. Yu, S. Choi, and J. Lee, "Enhancement of VoIP over IEEE 802.11 WLAN via dual queue strategy," in *Proc. ICC*, 2004, pp. 3706–3711.
- [15] W. Wang, S. C. Liew, and V. O. K. Li, "Solutions to performance problems in VoIP over 802.11 wireless LAN," *IEEE Trans. Veh. Technol.*, pp. 366–384, 2005.

- [16] Y. Xiao, "Concatenation and piggyback mechanisms for the IEEE 802.11 MAC," in *Proc. IEEE WNCN*, 2004, pp. 1642–1647.
- [17] A. Jain, M. Gruteser, M. Neufeld, and D. Grunwald, "Benefits of packet aggregation in ad-hoc wireless network," Dept. Comput. Sci., Univ. Colorado, Boulder, CO, Tech. Rep. CU-CS-960-03, 2003.
- [18] H. YAMADA and N. HIGUCHI, "Voice quality evaluation of IP-based voice stream multiplexing schemes," in *Local Comput. Netw.*, 2001, pp. 356–364.
- [19] D. Niculescu, S. Ganguly, K. Kim, and R. Izmailov, "Performance of VoIP in a 802.11-based wireless mesh network," in *Proc. IEEE INFOCOM*, Barcelona, Apr. 2006, pp. 1–11.
- [20] R. Komolafe and O. Gardner, "Aggregation of VoIP streams in a 3G mobile network: A teletraffic perspective," in *Proc. Eur. Pers. Mobile Commun. Conf.*, 2003, pp. 545–549.
- [21] M. Shin, A. Mishra, and W. A. Arbaugh, "Improving the latency of 802.11 hand-offs using neighbor graph," in *Proc. ACM MOBISYS*, 2004, pp. 70–83.
- [22] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne, "Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs," in *Proc. ACM MOBIWAC*, 2004, pp. 19–26.
- [23] S. Sharma, N. Zhu, and T. cker Chiueh, "Low-latency mobile IP handoff for infrastructure-mode wireless LANs," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 4, pp. 643–652, May 2004.
- [24] C. Perkins, "IP mobility support," RFC 2002, Oct. 1996.
- [25] V. Navda, A. Kashyap, and S. R. Das, "Design and evaluation of iMesh: An infrastructure-mode wireless mesh network," in *Proc. IEEE Int. Symp. World of Wireless, Mobile, Multimedia Netw.*, Taormina, Italy, Jun. 2005, pp. 164–170.
- [26] R. Cole and J. Rosenbluth, "Voice over IP performance monitoring," *ACM Comput. Commun. Rev.*, vol. 31, pp. 9–24, Apr. 2001.
- [27] Transparent Mobile IP. [Online]. Available: http://www.slyware.com/projects_tmip_intro.shtml.
- [28] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Trans. Comput. Syst.*, vol. 18, pp. 263–297, Aug. 2000.
- [29] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. 9th ACM MobiCom*, San Diego, CA, Sep. 2003, pp. 419–434.
- [30] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 2, pp. 93–102, 2003.

Samrat Ganguly received the Bachelors degree in physics from the Indian Institute of Technology, KGP, Kharagpur, India, the M.E. degree from the Department of Computer Science and Automation, Indian Institute of Science, Bangalore, and the Ph.D. from the Department of Computer Science, Rutgers University, Piscataway.

He is a Research Staff Member at NEC Laboratories America, Inc., Princeton, NJ.



Vishnu Navda (S'05) received the B.Tech. degree in computer science from Bangalore University, Bangalore, India, in 2000. He is currently working towards the Ph.D. degree in computer science at the State University of New York at Stony Brook, Stony Brook, NY.

His research interests are in wireless mesh networks with focus on QoS, security, and mobility management.



Kyungtae Kim received the B.S. degree from the Department of Electronics Engineering, Hanyang University, Hanyang, Korea, and the M.S. degree in computer science from Columbia University, New York. He is currently working towards the Ph.D. degree at the Department of Electrical and Computer Engineering, State University of New York at Stony Brook, Stony Brook, NY.

He is currently working with NEC Laboratories America, Inc., Princeton, NJ, in the areas of multimedia communication over the wireless mesh network, security on VoIP, and mobility management.



Anand Kashyap is currently working towards the Ph.D. degree at the Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY.

His research is in the area of 802.11-based wireless mesh networks. His research includes creating empirical models for capacity, studying interference, and using these ideas for efficient routing, and real-time applications over a mesh network.



Dragos Niculescu received the B.S. degree from the University Politehnica of Bucharest, Bucharest, Romania, in 1994 and the Ph.D. degree from Rutgers University, Piscataway, NJ, in 2004, both in computer science.

Since 2004, he has been a Researcher with NEC Laboratories America, Inc., Princeton, NJ. His research interests are in communication networks with emphasis on wireless networks, sensor networks, VoIP, and related technologies.



Rauf Izmailov received the M.S. degree in mathematics from the Moscow State University, Moscow, Russia, in 1984, and the Ph.D. degree in control science from the Institute of Control Science, Moscow, in 1988.

From 1991 to 1993, he was with AT&T Bell Laboratories, Holmdel, NJ. Since 1994, he has been with NEC Laboratories America, Inc., Princeton, NJ, where he is currently a Department Head of the IP Networks and Distributed Systems Department. His research interests include control problems in communication networks, optimization algorithms, and performance evaluation.



Sangjin Hong (SM'05) received the B.S. and M.S. degrees in electrical engineering and computer science from the University of California at Berkeley, and the Ph.D. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor.

He is an Assistant Professor in the Department of Electrical and Computer Engineering, State University of New York at Stony Brook, Stony Brook, NY. His research interests are in the areas of low-power VLSI design of multimedia wireless communications and DSP systems.



Samir R. Das (M'90) received the Ph.D. degree in computer science from the Georgia Institute of Technology, Atlanta, in 1994.

He is an Associate Professor in the Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY. His research interests include wireless ad hoc and sensor networking, performance evaluation, and parallel discrete event simulation.