An Experimental Study on Wi-Fi Ad-Hoc Mode for Mobile Device-to-Device Video Delivery

Bo Xing[1], Karim Seada[2], and Nalini Venkatasubramanian [1]

[1]School of Information and Computer Sciences, University of California, Irvine. {bxing, nalini}@ics.uci.edu
[2]Nokia Research Center, Palo Alto. karim.seada@nokia.com

Abstract— **The demand for video content is continuously increasing as video sharing on the Internet is becoming enormously popular recently. This demand, with its high bandwidth requirements, has a considerable impact on the load of the network infrastructure. As more users access videos from their mobile devices, the load on the current wireless infrastructure (which has limited capacity) will be even more significant. Based on observations from many local video sharing scenarios, in this paper, we study the tradeoffs of using Wi-Fi ad-hoc mode versus infrastructure mode for video streaming between adjacent devices. We thus show the potential of direct device-to-device communication as a way to reduce the load on the wireless infrastructure and to improve user experiences. Setting up experiments for Wi-Fi devices connected in ad-hoc mode, we collect measurements for various video streaming scenarios and compare them to the case where the devices are connected through access points. The results show the improvements in latency, jitter and loss rate. More importantly, the results show that the performance in direct device-to-device streaming is much more stable in contrast to the access point case, where different factors affect the performance causing widely unpredictable qualities.**

I. Introduction

The recent few years have witnessed a tremendous growth in video content generation and distribution. This can be noticed in the large number of sites that make it easy for users to upload and share videos. Some of these sites like YouTube [1] are among the most trafficked sites on the web, in addition to social networking sites such as FaceBook [2] and MySpace [3] which allow users to share videos online with their friends.

Some of the technological factors that lead to such increase are the wide spread of wireline broadband Internet access, such as cable and DSL, with which it becomes much faster to download/upload high resolution videos. In addition, the ubiquitous availability of portable devices with cameras, such as smart phones, PDAs and Internet Tablets, makes it now very convenient for regular users to create videos at a reasonable quality instantaneously. This has a strong impact on video content, as many of the videos we see now are not created by professional media studios, but could actually be created by anyone. Although mobile devices play an important role in creating these personal videos, till now they have not played a major role in uploading, downloading, or generally sharing these videos, due to the limited capacity and speed of current wireless connectivity. Nevertheless, there is increasing interest in playing online video on mobile devices, and many of the high-end devices come with Flash players [4], Windows Media Player Mobile [5], or even customized YouTube players reflecting a huge demand from users to have this functionality.

However, mobile video delivery is a challenging application for any wireless technology as it needs high bandwidth and has tight latency requirements. As the demand on downloading and uploading videos using mobile devices increases, this could cause an overload on the limited capacities of the current cellular data networks and even Wi-Fi hotspots, negatively affecting their performance and the experience of users using these networks. Our observation is that many of the video sharing scenarios potentially can be carried out locally without increasing the load on the network infrastructure. For example, sharing of the personal videos created between families and friends at certain events happens directly on the spot; a group of people simultaneously watch videos, whether they are friends downloading and watching the same video or at a public place where news and announcements are being broadcasted.

The above scenarios and others share the characteristic that a few devices in the proximity are downloading or exchanging the same videos. Although this could be achieved all through using the cellular data network or Wi-Fi hotspots, the quality of service might vary from case to case as users normally do not have controls of the conditions and parameters of the infrastructure. Hence, it would also make sense, whenever possible, to share videos directly without going through the infrastructure (thus sharing can happen even where infrastructure access is unavailable or intermittent). However, although current mobile devices come with increasing networking capabilities, their direct communication capability has not been fully exploited. Moreover, it is still unclear which connectivity mode (direct connectivity or infrastructure connectivity) would offer better performance for video delivery, and in what situations.

Hence, in this paper we study video delivery between devices that are in the proximity of each other. We choose to look at 802.11 Wi-Fi, due to the increasing availability of Wi-Fi connectivity in mobile devices as well as its large transmission range and high capacity compared to Bluetooth. Our goal is to identify the tradeoffs between using Wi-Fi ad-hoc and infrastructure modes for device-to-device video delivery. In the meantime, we seek to shed light on the benefits of using ad-hoc mode as an alternative for mobile video streaming, and thus stimulate further studies and deployments of it.

We empirically examine video streaming performance by running extensive experiments on real mobile devices. We measure the latency, jitter and loss rate of streaming videos at different Wi-Fi operating modes and settings. We investigate the variations of the metrics under varying traffic load, contentions/collisions and inter-device distances. We note that the focus of our study is on the comparisons of direct device-to-device communications against communications using infrastructures; we do not address detailed video specifics such as encoding and framing, nor do we address the optimizations of Wi-Fi protocols for supporting multimedia traffic.

Bo Xing was an intern at Nokia Research Center Palo Alto during this work.

In the following section we start by reviewing prior work in related areas. We then discuss the factors that may have impact on streaming performance in Wi-Fi networks. In Section IV, we describe our experiment testbed and methodology. We present our experiment results and observations in Section V. We finally conclude the paper in Section VI.

II. Related Work

Although there has been a large amount of research work in the area of wireless ad-hoc networks [6] which inherently rely on device-to-device connectivity, the majority of the work focuses on routing issues and is based on simulation studies [7]. We believe that the performance of actual device-to-device communication has not been adequately studied and evaluated in real-world settings. There have been related experimental studies in the areas of mesh [8] and sensor networks [9] [10] as they are gaining more deployments, but less in the context of networks formed by mobile personal devices which fall behind in terms of serious real-world deployments. The work by Anastasi et al. [11] pioneered experimental studies of 802.11 ad-hoc network performance on real testbed, and pinpointed discrepancies between the results from simulations and realities. Their work however focuses on the performance of ad-hoc mode only, and does not consider infrastructure mode.

Multimedia over wireless networks has drawn increasing attentions in recent years. The authors of [12] study video streaming in 802.11b infrastructure networks, focusing on the effect of encoding parameters and frame packetization on delay. Ad-hoc mode device-to-device communication was not studied there. In [8], a mesh network testbed is used to study the performance of multimedia traffic under different network conditions and wireless card configurations. The results are interesting in showing the capacity of the mesh network and the performance of real-time traffic on paths of multi-hop mesh routers. MobiUS [13] presents a collaborative video application, where a high resolution video downloaded by a mobile device is displayed across the screens of two adjacent mobile devices. The focus of that paper is on the challenging half-frame decoding and rendering requirements and its impact on the processing power and battery life, rather than the underlying communication mechanism. From our perspective, it would be interesting to see, for such an emerging application, what is the best way of transferring the video between the devices so that an enjoyable "together-viewing" user experience can be achieved.

An interesting perspective on Video-on-Demand (VoD) is recently presented in [14], where the profitability of Internet VoD in general is examined. It is argued that VoD is a costly service to provide and this cost will increase with growing demand. Accordingly, the authors present the benefits of a peer-assisted approach, where peers assist the server by redistributing the videos that they are currently watching. Although this work is not related to wireless and considers peers in the general term, it maps well with our objective, where peers in our case are the devices in the proximity which can communicate directly. Similar idea has also been adopted by MOVi [15], a recently proposed application specifically designed for mobile opportunistic networks. In MOVi, VoD on mobile devices is carried out by exploiting both the downlinks from the servers and the direct peer-to-peer links between mobile devices. It is shown that this approach sheds the load on the infrastructure and improves overall system throughput.

III. Preliminaries

Before getting into the details of our experimental study, we briefly review the preliminaries of 802.11 Wi-Fi and the factors that may have impact on its performance.

A. Wi-Fi Operating Modes

802.11 Wi-Fi has two modes of operation: infrastructure mode and ad-hoc mode. In infrastructure mode, devices need to connect to an access point and all communications happen through the access point. The access point and the devices associated with it form a basic service set (BSS), identified by an SSID. Currently, most wireless networks we use at home, in enterprise, or at hotspots work in infrastructure mode. In ad-hoc mode, devices within one another's communication range communicate directly without going through access points [16]. The devices thus form an independent basic service set (IBSS), which is identified by an SSID as well. As compared to infrastructure mode, ad-hoc mode is not frequently used, and its potentials in real deployments remain to be explored.

B. Wi-Fi Medium Access Control

The basic medium access method of 802.11 Wi-Fi, called Distributed Coordination Function (DCF), is basically a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. Both infrastructure mode and ad-hoc mode use CSMA/CA; in infrastructure mode, an access point is treated equally as a mobile device (collectively called stations) in contending for the medium. CSMA/CA works as follows. A station willing to transmit senses the medium, if the medium is busy then it defers. If the medium is free for a specific period of time, then the station transmits. If the transmission is a unicast, the receiving station sends an acknowledgement (ACK). If the sender does not receive the ACK, it will retransmit until it gets acknowledged or discarded after a given number of retries.

In unicast transmissions, optionally an RTS/CTS mechanism is employed to reduce the probability of two stations colliding after they both sense the medium to be idle. A station willing to transmit will first transmit a short control packet called RTS (Request To Send), which contains the source, the destination, and the duration of the following transaction (the packet and the respective ACK). The destination station will respond (if the medium is free) with a CTS (Clear To Send) packet which includes the same duration information. Only after the sender receives the CTS will it start transmitting the actual data. All stations hearing the RTS and/or the CTS will refrain from transmitting during the given duration of time.

C. Factors Affecting Streaming Performance

Video streaming performance in Wi-Fi networks is affected by a number of factors. Other than the operating modes, the network settings and network conditions play important roles as well. In the following, we discuss some of the key factors.

1) Traffic Load: The bandwidth of 802.11 Wi-Fi is limited compared to wireline networks, and the actual bandwidth available in reality is lower depending on many factors. For instance, 802.11b theoretically supports up to 11Mbps; when two devices communicate with each other through an 802.11b access point, each station obtains only 13% of the theoretical maximum (about 1.6Mbps) [17]. If the traffic load placed on the network is heavy, the medium can be very busy. When the transmission channel is close to being fully utilized, network congestion occurs [18], which leads to packet losses, long and unstable latency, and can cause significant performance degradation to applications. Hence, in both infrastructure mode and ad-hoc mode, streaming videos with high bit rates may congest the network. In comparison, ad-hoc mode should be less affected, because there are fewer stations sharing the network bandwidth and thus the congestion threshold is higher.

2) Contentions and Collisions: The wireless medium is inherently a broadcast medium. Stations that wish to transmit have to contend for accessing the medium, as we have described earlier. With more stations concurrently attempting to transmit, on average each of them needs to wait for a longer time before successful transmissions. In video streaming, this translates into longer latency with larger variations. Although the RTS/CTS handshake mechanism reduces the probability of stations colliding with each other, collisions still will occur, especially when stations all have continuous data traffic to send at the same time. These collisions lead to either packet losses, or retransmission attempts (which incur longer latency or cause further contentions and collisions). Intuitively, infrastructure mode is affected by contentions and collisions to a greater extent, because more stations are involved.

3) Interferences: Interferences happen to a Wi-Fi network when nearby networks or RF equipments operating at the same frequency band are transmitting signals. Because of the way CSMA/CA works, interferences may cause stations to delay transmissions or perform unnecessary retransmissions. As compared to mobile devices communicating directly, adding an access point to the network enlarges the sensing range of the network, and in effect introduces an extra interference-susceptible point. Furthermore, in an area where Wi-Fi infrastructure is not well planned, e.g., multiple access points tuned to the same channel are placed with overlapping coverage, interferences could be more severe. In contrast, ad-hoc mode is usually used for impromptu communications; an IBSS when initiated can automatically scan the channels and select the one with minimum interferences. Hence, infrastructure mode potentially suffers more from interferences than ad-hoc mode.

4) Beaconing: 802.11 Wi-Fi networks use beacon frames to enable stations to establish and maintain communications in an orderly fashion. The beacon frames are periodically transmitted management frames, independent from any data frames from higher layers. They are sent using CSMA/CA and hence need to compete for the medium with other frames. The beacons provide the "heartbeats" of a Wi-Fi network; they advertise the network configuration parameters in addition to the SSID. In infrastructure mode, the access point is responsible for periodically broadcasting beacons. Whereas in ad-hoc mode, peer devices rotate in sending beacons. As video streaming is a data-intensive and bandwidth-consuming application, beaconing can have an impact on its performance. While infrastructure mode places the beaconing responsibility solely on a single node, it could be affected more severely.

IV. Experiment Methodology

A. Experiment Platform

To empirically study the tradeoffs of Wi-Fi infrastructure and ad-hoc modes, we set up a testbed in our office building, using off-the-shelf handheld devices that come with Wi-Fi capabilities and can perform video streaming. The devices we use are the Nokia N800 Internet Tablets [19]. The N800 operating system, Maemo [20], is a modified version of Debian GNU/Linux. It allows for setting up Wi-Fi ad-hoc connectivity with configurable parameters. We tune the devices' Wi-Fi transmission power uniformly to 10mW (transmission range being approximately 35 meters). Moreover, we turn off power saving mode on them to minimize the impact of the interfaces switching on and off. When working at ad-hoc mode, the devices are configured to connect to the same network SSID, and the frequency channel is automatically selected. In examining infrastructure mode, the access points we use are the LinkSys Wireless-G Broadband Routers WRT54GL [21]. We set all their parameters to factory default values (e.g., the beacon interval is 100ms).

For video recording and streaming on the devices, we employ the Gstreamer library [22]. To retain consistency across all runs, we experiment with a pre-recorded AVI video file (1-minute long) instead of live video. The signature video is captured by the N800's embedded camera at the frame rate of 15 frames per second. The resolution is 320*240. It is encoded using an H.263 codec (provided by the "hantro4200enc" element in Gstreamer) at the bit rate of 96kbps. The server of a streaming session runs a Gstreamer pipeline which encapsulates the H.263 video data in RTP packets and sends them to the client through UDP. The client also runs a Gstreamer pipeline, which, upon receiving the RTP packets, decodes the payload and displays the video. When needed, we increase the traffic load between a server/client pair by simultaneously running multiple streaming sessions of the signature video on them. This is in effect similar to streaming a video whose bit rate equals the signature video's bit rate multiplying the number of sessions.

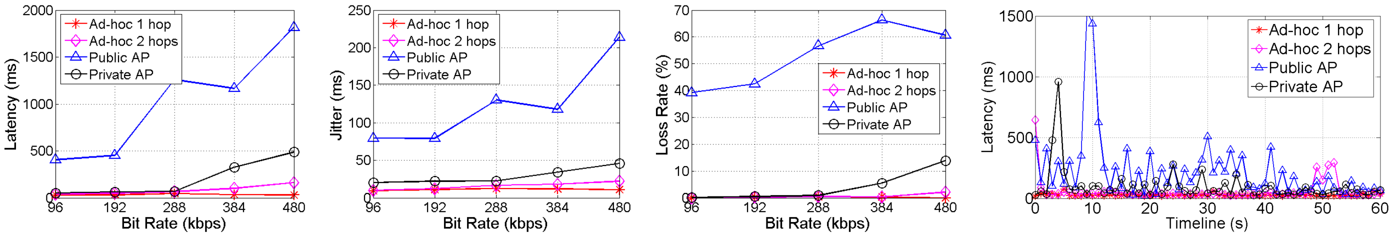
B. Performance Measuring

Measuring the absolute values of streaming latency requires synchronization between servers and clients, which however is extremely hard on mobile handheld devices. Since our measurements are for comparison purposes only, obtaining absolute values is not necessary. We hence bypass the synchronization problem by measuring the performance metrics in a "round-trip" fashion as described in the following.

At the server side, every RTP packet being sent out to the client is in the meantime directed to a local UDP port, where the packet's sequence number, payload size, timestamp as well as its departure time (based on the server's clock) are logged. As soon as the packet arrives at the client side, while being delivered to the application for presentation, it is bounced back to the server through UDP (note that this immediate bouncing back happens at the UDP layer, not at the MAC layer). The server then logs the returning packet's arrival time (again, based on the server's clock). When the streaming session finishes, based on these logs, the server calculates the performance metrics (as will be described shortly). We believe that this "round-trip" method is fair to all compared scenarios, and the results are roughly equivalent to averaging two opposite unidirectional streams running in parallel. Each average result we report in this section is averaged over 10 experiment runs.

C. Performance Metrics

We capture the performance of video streaming on mobile devices using metrics defined as follows.



(a) Latency Vs Bit Rate

(b) Jitter Vs Bit Rate

(c) Loss Rate Vs Bit Rate

(d) Average Latency at Each Second along the Timeline in Typical Runs at 480bps Bit Rate

Fig. 1. Experiment Results under Varying Traffic Load

1) Latency (Round-Trip Time): The latency of a particular RTP packet, is the time elapsed from when the packet is sent out by the server till when the packet is bounced back and arrives at the server. The latency of a streaming video, is the average of the latencies of all its RTP packets.

2) Jitter: The jitter of a particular RTP packet, is the difference between the latency of this packet and the latency of its preceding packet (the packet whose sequence number is smaller by 1). The jitter of a streaming video, is the average of the jitters of all its RTP packets.

3) Loss Rate: The loss rate of a streaming video, is the ratio of the number of packets that are not bounced back to the total number of RTP packets the video contains.

D. Experiment Settings

To acquire deeper understanding of the tradeoffs as well as for fairer comparisons, rather than simply examining ad-hoc mode versus infrastructure mode, we measure streaming performance under the following four settings.

1) Ad-Hoc 1 Hop: the server device and the client device communicate directly.

2) Ad-Hoc 2 hops: The server device and the client device communicate through a relaying device. The routing tables on the devices are configured such that all the traffic between the server and the client devices will go through the relaying device. This setting is intended to have a same topology as infrastructure mode and thus reveal more insights on the tradeoffs between ad-hoc and infrastructure modes.

3) Public AP: The server and the client devices communicate through an access point, which is accessible to all people in the building (and thus may have unknown background traffic).

4) Private AP: The server device and the client device communicate through an access point, which is only accessible to our experiments (and thus has no other traffic).

E. Experiment Design

Our experiments are designed to compare the network settings and examine video streaming performance under various network conditions, taking into account the following factors.

1) Varying Traffic Load: In our first set of experiments, we generate varying traffic load by increasing the number of concurrent streaming sessions between the same server/client pair from 1, 2, .., to 5 (the overall bit rate of the traffic thus varies from 96kbps, 192kbps, ..., to 480kbps). With the increase in bit rate, the network is becoming congested. This is to mimic the scenario where two users are sharing a video with increasingly high quality.

2) Varying Level of Contentions and Collisions: In our second set of experiments, we introduce two more mobile devices and thus make two server/client pairs. We run two streaming sessions (each at 96bps bit rate) separately over the two server/client pairs in parallel. As the devices are placed closely, increased amount of collisions and contentions will occur. This is to simulate the scenario where users at the same site simultaneously share videos with one another. Note that in the ad-hoc 2 hops setting, the two streaming sessions go through the same intermediate device; similarly, in the infrastructure setting, they go through the same access point.

3) Varying Distance: In the above experiment sets, to minimize the impact of fading and attenuation, we place all involved handheld devices and access points closely. In the third experiment set, we look into the impact of distance. The experiments are conducted in an open area with minimum obstacles. We vary the distance between a server device and a client device from 0m, 10m, ..., to 40m, and run 5 concurrent streaming sessions (480kbps bit rate) between them. The relaying device or the access point, if present, is placed in the exact middle of the server device and the client device. Note that when the distance is 40m, the server/client devices are actually out of each other's range. Hence, the ad-hoc 1 hop setting has no measurements for the distance of 40m.

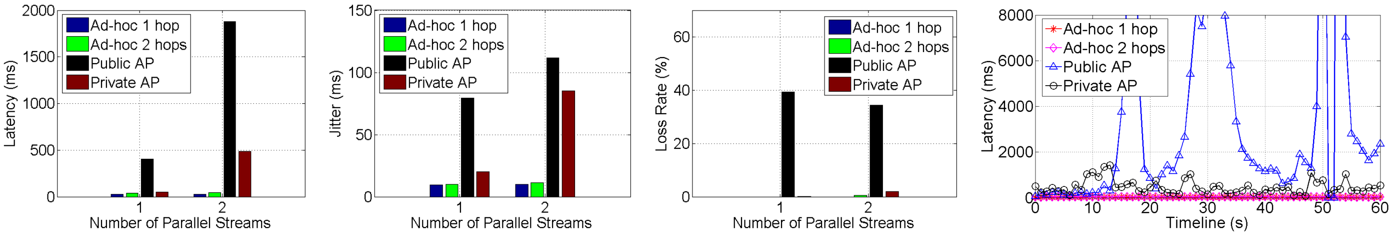
V. Experiment Results and Findings

Having experimented with signature videos with low motion, medium motion and high motion, respectively, we have observed that the motion characteristic of a signature video has a negligible impact on the comparisons between the network settings. Hence, in this section, we present only the results from our experiments with a signature video with medium motion.

A. Varying Traffic Load

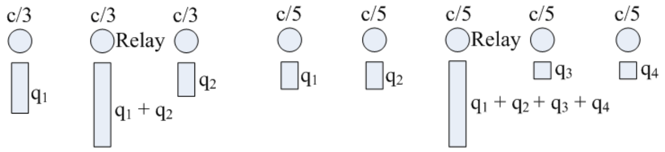
The average results from our first set of experiments are potted in Fig. 1(a), Fig. 1(b) and Fig. 1(c). To further show the variations of the time-related metrics, we pick typical runs under 480bps bit rate, calculate the average latency for each second along the timeline, and plot them in Fig. 1(d).

From these graphs, the first and the most obvious observation is that, the performance of public AP is significantly worse than the other three settings. In general, video streaming through public AP experiences high latency, jitter and loss rate, as well as poor playback quality at the client. The root cause for that is the unknown background traffic from other users of the access point, which adds traffic load and incurs more contentions and collisions. This is evident if we compare the public AP setting to the private AP setting, in which no background traffic is present. Private AP offers much better and relatively stable



(a) Latency Vs Streams (b) Jitter Vs Streams (c) Loss Rate Vs Streams (d) Average Latency at Each Second along the Timeline in Typical Runs with Parallel Streams

Fig. 2. Experiment Results under Varying Level of Contentions and Collisions



(a) 3 Stations (b) 5 Stations

Fig. 3. The relaying node which has the highest queue occupancy but shared medium bandwidth becomes a bottleneck

performance in all aspects. Further, in our experiments with the public AP setting, we observed wide performance variations, both in successive runs and at different time of a day, which makes video quality very unpredictable.

With the increase in video bit rate, streaming performance tends to degrade – latency, jitter and loss rate all go up. The impact of traffic load, as show in Fig. 1, however is relatively small on the ad-hoc 1 hop setting. It is slightly larger on the ad-hoc 2 hops setting, and further larger on the private AP setting. The main reason why the ad-hoc 1 hop setting outperforms others is that it does not use a relaying node – as fewer stations share the bandwidth of the network medium, more traffic can be accommodated without causing network congestion.

The comparison between the ad-hoc 2 hops setting and the private AP setting is interesting and insightful. Although the two settings have the same topology (using a relay), ad-hoc 2 hops performs slightly better. We believe that two major factors make this happen: interferences and beaconing. First, an access point typically has larger sensing range than a mobile device, and hence is more susceptible to interferences, leading to longer waiting time before transmissions and more retransmissions.

Second, in the ad-hoc 2 hops setting beaconing is distributed on all devices rather than the relaying device alone, whereas in the private AP setting beacons are all sent by the access point causing extra overhead on the bottleneck node. This impact is more significant when the network bandwidth is close to being fully utilized. We explain this in more detail in the following. In 802.11, the medium is equally shared among all active nodes irrespective of their loads. This is intended to provide fair capacity sharing, but has the drawback of the relaying node becoming a bottleneck. This problem was first identified in [23], which models the bottleneck for evaluating overall delay and buffer characteristics. As illustrated in Fig. 3(a), the relaying node has the highest queue occupancy ($q_1 + q_2$) but only one third of the shared medium capacity ($c/3$). Adding b beaconing traffic to the bottleneck node in infrastructure mode compared to adding $b/3$ in ad-hoc mode degrades overall performance more. In comparison, ad-hoc mode saves network resources by distributing beaconing traffic and thus reducing the load on the bottleneck node. Similar effect has also been observed in other areas such as network coding [24], where access points intelligently merging packet transmissions has lead to larger gains at the MAC layer than what was expected as just sending fewer packets. To uncover more on this phenomenon, we manually change the configuration of the private AP and vary its beacon interval from 20ms to 1000ms. We observe that, especially when the traffic load is heavy, the performance metrics improve with the increase in beacon interval.

B. Varying Level of Contentions and Collisions

We present the results from our second set of experiments in a similar fashion in Fig. 2. We pick typical runs that have relatively few packet losses from the parallel-stream scenario and plot the average latency along the timeline in Fig. 2(d).

When multiple parallel streams are running over separate server/client pairs, there are more occurrences of contentions and collisions. In this case, all the performance metrics tend to degrade, as shown in Fig. 2. In comparison, the ad-hoc settings are less affected, as in ad-hoc mode fewer stations are involved. In particular, the ad-hoc 1 hop setting performs consistently well in all scenarios, achieving zero loss rate (Fig. 2(c)) and stably low latency (Fig. 2(a) and Fig. 2(b)).

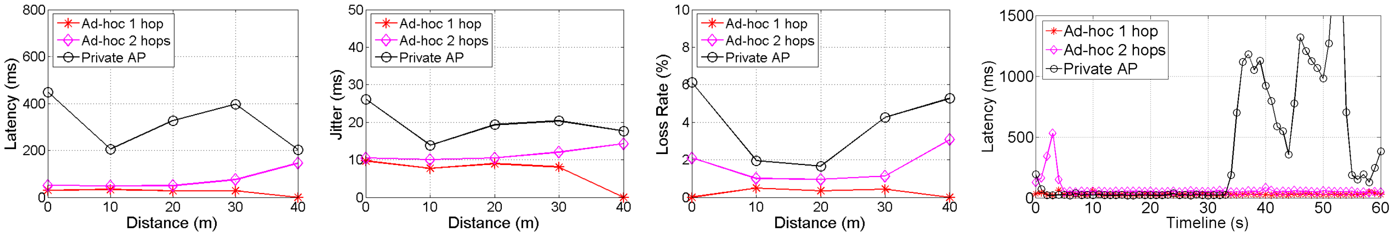
In an environment with increasing contentions and collisions, the public AP setting again has the worst performance, due to similar reasons we have discussed earlier. In some cases, half of the RTP packets are lost. Even in certain runs where its loss rate is relatively low, its latency and jitter are sometimes outrageous, as shown in Fig. 2(d).

With parallel streams, the performance gap between the private AP setting and the ad-hoc 2 hops setting becomes more significant. This can be explained using Fig. 3(b). Now that there are five stations in the topology, the relaying node has even higher queue occupancy (the sum of all others') but further lower capacity share ($c/5$) – it is now a more severe bottleneck. Because in the private AP setting the beaconing responsibility is taken by the bottleneck node (the access point) alone, its performance is impaired to a larger extent.

C. Varying Distance

The results from our third experiment sets are presented in Fig. 4. In particular, Fig. 4(d) plots the average latency along the timeline in typical runs (which have relatively few packet losses) in the 30m-distance scenario. In these graphs, we exclude the public AP setting considering its notoriously poor performance. The metrics for the ad-hoc 1 hop setting at the distance of 40m are all plotted as zero due to no measurements.

From the experiments we observe that, in general, the distance between the server and the client devices does not have a deterministic impact on streaming performance. When the distance changes, the average values of the metrics fluctuate



(a) Latency Vs Distance (b) Jitter Vs Distance (c) Loss Rate Vs Distance (d) Average Latency at Each Second along the Timeline in Typical Runs at 30m Distance

Fig. 4. Experiment Results under Varying Distance

to some extent, but the fluctuations follow no patterns. While intuitively an increasing distance leads to longer latency and more packet losses, our results show that distance is a less influential factor compared to other factors (e.g., interferences).

In comparing the network settings across varying distance, the observations are consistent with those from our previous experiment sets: ad-hoc 1 hop performs better than ad-hoc 2 hops, which in turn outperforms private AP; the ad-hoc settings show better performance consistency than the AP setting both across multiple runs and within individual runs. It is worth mentioning that even when the server and the client devices are out of each other's range (e.g., distance being 40m), using ad-hoc mode is still advantageous – it improves streaming performance by having beaconing decentralized. This is not to say that ad-hoc mode can replace infrastructure mode. Indubitably infrastructure mode is the major connectivity, especially in situations where the communicating devices are far apart and no intermediate relaying devices are present. However, when certain devices are sufficiently close, *ad-hoc mode can serve as an alternative to infrastructure mode if the load on the wireless infrastructure is heavy or undetermined*.

D. Experiment Summary

In summary, our experiments show that, *for delivering video between mobile devices in close proximity, Wi-Fi ad-hoc mode performs comparatively to, and in many cases outperforms infrastructure mode*. The performance gain of ad-hoc mode is especially significant when the streamed video is of high bit rate, and/or multiple co-located devices are streaming videos simultaneously. By enabling direct communications, ad-hoc mode increases the bandwidth share at the devices and reduces possible contentions/collisions. Moreover, it evenly distributes the beaconing task on all devices, rather than concentrating it on an access point which in that case becomes a performance bottleneck. On the other hand, the service quality offered by infrastructure mode can vary tremendously and is unpredictable, depending on various factors that users might not have control of (such as infrastructure planning, background traffic, etc.). Hence, *Wi-Fi ad-hoc mode could provide a beneficial complement to infrastructure mode*. For example, in local sharing scenarios where a group of devices at the same site are streaming the same video, it would be helpful to have some devices download it through access points while having others retrieve it directly from nearby devices (as advocated in [15]).

VI. Concluding Remarks

In this paper, we present our experimental study on mobile device-to-device video delivery. We empirically examine the tradeoffs between using Wi-Fi ad-hoc and infrastructure modes, and investigate the potential of direct device-to-device communication in real deployments. Our experiments show that mobile devices are offered better and more stable video streaming quality when communicating directly instead of through access points. Infrastructure mode could perform comparably as ad-hoc mode, but the parameters of and the activities on access points have to be optimized to offer best support for specific applications. The findings from our experimental study indicate that ad-hoc mode can be exploited as a very good complement to infrastructure mode in sharing workload and enhancing streaming performance; this well backs up the feasibility of recently proposed applications such as MobiUS [13] and MOVi [15]. We believe that they will also provide helpful guidelines to the design of new mobile video applications and protocols.

References

[1] http://www.youtube.com.
[2] http://www.facebook.com.
[3] http://www.myspace.com.
[4] http://www.adobe.com.
[5] http://www.microsoft.com/windowsmobile.
[6] C. Perkins, *Ad Hoc Networking*. Addison-Wesley Professional, 2001.
[7] C. Sweet, V. Devarapalli, and D. Sidhu, "Ieee 802.11 performance in an ad-hoc environment," 2004, whitepaper.
[8] Y. Sun, I. Sheriff, E. M. Belding-Royer, and K. C. Almeroth, "An experimental study of multimedia traffic performance in mesh networks," in *WitMeMo*, 2005.
[9] O. Younis and S. Fahmy, "An experimental study of routing and data aggregation in sensor networks," in *MASS*, 2005.
[10] G. Anastasi, M. Conti, E. Gregori, C. Spagoni, and G. Valente, "Motes sensor networks in dynamic scenarios: An experimental study for pervasive applications in urban environments," *Journal of Ubiquitous Computing and Intelligence*, vol. 1, no. 1, April 2007.
[11] G. Anastasi, E. Borgia, M. Conti, and E. Gregori, "Ieee 802.11 ad hoc networks: Performance measurements," in *ICDCSW*, 2003.
[12] N. Cranley and M. Davis, "Study of the behaviour of video streaming over 802.11b wlan networks," in *WiMob*, 2006.
[13] G. Shen, Y. Li, and Y. Zhang, "Mobius: Enable together-viewing video experience across two mobile devices," in *MobiSys*, 2007.
[14] C. Huang, J. Li, and K. W. Ross, "Can internet video-on-demand be profitable?" in *SIGCOMM*, 2007.
[15] H. Yoon, J. Kim, F. Tan, and R. Hsieh, "On-demand video streaming in mobile opportunistic networks," in *PerCom*, 2008.
[16] "Ieee standard: Wireless lan medium access control (mac) and pysical layer (phy) specifications," 1999.
[17] PassMark, "802.11 wireless bandwidth tests," http://www.passmark.com/support /wireless_bandwidth_testing.html.
[18] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-royer, "Understanding congestion in ieee 802.11b wireless networks," in *IMC*, 2005.
[19] http://tableteer.nokia.com.
[20] http://maemo.org.
[21] http://www.linksys.com.
[22] http://gstreamer.freedesktop.org.
[23] H. van den Berg, M. Mandjes, and F. Roijers, "Performance modeling of a bottleneck node in an ieee 802.11 ad-hoc network," in *Ad Hoc Now*, 2006.
[24] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," in *SIGCOMM*, 2006.