Dynamic Channel Switching for High-Definition Peer-to-Peer 802.11-based Video Streaming

Erin Sullivan, Taylor Johnson, and Ben Lee
School of Electrical Engineering and Computer Science
Oregon State University
Corvallis, OR 97331
Email: {sullivae, johnstay, benl}@eecs.orst.edu

***Abstract*—Peer-to-peer streaming of HD video over ad-hoc 802.11 wireless networks is a popular application, but the received video quality, and therefore the user's Quality of Experience (QoE), depends heavily on the condition of the wireless channel used. This paper presents *Active Scanning-based Dynamic Channel Switching* (ASDCS), which ensures wireless video streaming takes place over the channel whose condition is most likely to provide good received video quality. ASDCS selects an initial channel before video streaming begins, assesses the performance of the current channel during streaming, and dynamically searches for a better channel if the current one is insufficient. Our simulation results show that ASDCS outperforms existing static queue-threshold and SINR-based methods.**

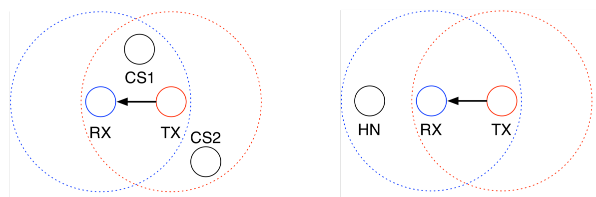
I. Introduction

Peer-to-peer HD video streaming over WLANs has become pervasive in our society. However, as the demand for this service increases and as new high-bitrate technologies such as 4K Ultra-HD and N-screens [1] emerge, network congestion among streaming node pairs and interference from neighboring WLANs pose a significant challenge to providing good quality streamed video.

One simple approach to avoiding network congestion is to choose the wireless channel with the least interference before beginning a video stream. There are many existing 802.11-based commercial solutions for video streaming [2], some of which may take this basic approach, but they are often proprietary or limited to a small subset of consumer devices. Also, channel conditions and available bandwidth vary over time due to node mobility and activity, thus a channel selection method that also dynamically assesses the current channel is much more desirable.

For these reasons, this paper presents the *Active Scanning-based Dynamic Channel Switching* (ASDCS) method, which addresses channel selection and evaluation for 802.11 wireless networks with the specific objective of improving user Quality of Experience (QoE) for peer-to-peer HD video streaming. ASDCS takes a unique approach to channel selection by considering the specific impact that channel interference and congestion have on received video quality. ASDCS also takes a novel approach to channel evaluation by combining the average transmission rate of the current wireless channel with the characteristics of encoded video to predict whether important video packets will arrive before their playout deadline.

Our simulation study shows that ASDCS is able to predict received video quality more accurately than existing methods, and identify which channel is best for video streaming, all while preserving user QoE by switching channels with minimal delay.



(a) Carrier sensing. (b) Hidden node.

Fig. 1: Carrier sensing and hidden node scenarios.

II. Background

This section presents the background information necessary to understand the proposed channel switching method.

The 802.11 Distributed Coordination Function (DCF) uses *carrier sensing* to perform Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [3]. When a station or node needs to transmit on the WLAN, it first senses the medium to determine if a nearby node is transmitting on the same channel. If so, it will defer its transmission and try again after a random backoff period. A *carrier sensing* (CS) node is any node within the carrier sensing range of the transmitter that could potentially contend for access to the wireless medium. Fig. 1a depicts a transmitter (TX) and receiver (RX) node pair with two CS nodes, where CS1 is visible to both RX and TX while CS2 is only visible to TX. Sharing a channel with CS nodes can cause packet delays, possibly resulting in dropped packets for nodes performing video streaming with strict playout deadlines.

A more severe problem is the presence of *hidden nodes*. Fig. 1b depicts a typical hidden node scenario, where a hidden node (HN) is outside of TX's carrier sensing range, but is visible to RX. TX and HN cannot sense each other during their usual contention for the wireless medium, and may unknowingly transmit packets at the same time causing packet collisions at RX. The presence of even a single hidden node can have a drastic impact on the quality of a wireless channel.

A simple method of dealing with hidden nodes is to use the optional RTS/CTS Exchange, which supplements CSMA/CA by specifying two additional control packets: Ready-to-Send (RTS) and Clear-to-Send (CTS). While RTS/CTS is somewhat effective at solving the hidden node problem, the additional overhead incurred by sending RTS and CTS frames will lead to a significant drop in overall throughput. Furthermore, the fact that other nodes are required to wait leads to increased packet delay, which again is especially a problem for real-time applications.

Implementing a video-centric dynamic channel switching method on 802.11 has a few notable challenges. A decision to switch channels during streaming must be made when the current channel is not providing good received video quality. This determination is difficult because most quantitative metrics are obtained after the video streaming has completed. A dynamic channel switching method must instead predict received video quality on-the-fly.

If the current channel's condition is poor, a dynamic channel switching method must pause the video stream and then take steps to characterize and rank prospective new channels. Even after a new channel is selected, there is additional delay since changing the channel on a station or an Access Point (AP) means the device's antenna must be tuned to a new frequency. Therefore, any video-centric channel switching algorithm must be able to quickly characterize and rank new channels.

III. Related Work

There are a number of related works that deal with hidden nodes, dynamic channel switching methods, and HD video streaming on mobile ad-hoc networks (MANETs). The following subsections discuss the state-of-the-art.

A. Hidden Node Detection

Beyond the basic RTS/CTS solution provided by the 802.11 standard, there have been several efforts to address hidden nodes. Li *et al.* proposed a passive detection method where a listening node enters the *promiscuous mode* and checks if a neighboring node sends a data packet without the corresponding ACK packet, or vice versa [4]. Hidden nodes are detected using this method only if they are directly communicating with the neighbor. Thus, the authors also proposed an active variant that uses probe packets, which is more effective in detecting hidden nodes. However, active probing requires modifications to all the nodes in the network to respond to probe packets. In addition, active probing detects hidden nodes whose signal strengths are low enough to be considered negligible interference. In contrast, the proposed ASDCS only requires modifications to the node pair performing video streaming, and uses a SINR threshold to disregard any hidden nodes that are unlikely to cause collisions at the receiver.

Hidden nodes can also be indirectly detected by looking for packet loss or delay levels that are disproportionate compared to the measured SINR, or combining signal strength or MAC-layer statistics with knowledge of the approximate loss due to carrier sense nodes [5], [6]. However, these methods require spending a significant amount of time observing each channel, which is impractical for use with real-time video streaming.

B. Channel Selection and Evaluation

Existing channel selection methods [7]–[9], often referred to as multi-channel methods, encourage *cooperation* between nodes by providing a *control channel*. Before transmission, each node will use the control channel to determine which data channel can best support its traffic without adversely affecting other nodes. The control channel may be a separate physical channel [7], which requires a second antenna, or the control channel is only accessed during synchronized time windows [8], [9], which reduces the overall available bandwidth. These methods are not well-suited for video-aware channel selection because adding another antenna is often impractical for consumer devices and reducing the available bandwidth is detrimental to video stream performance.

Other existing methods are *distributed*, where nodes make independent decisions about channel selection, often using either receiver SINR [10] or packet delay [11], [12] as a channel quality metric. Although packet delay can be a strong indicator of a channel's ability to stream video successfully, determining the packet delay of all prospective new channels requires sending many test packets, which is too time consuming for a video streaming application.

In contrast with existing methods, ASDCS does not require any control channel mechanism, or impose any sort of bandwidth sharing or restriction. In addition, ASDCS goes a step further than existing selection and evaluation methods by having a video-centric perspective that informs its predictions and decisions.

C. Video streaming over MANETs

Some existing techniques do specifically address video quality when transmitting over a wireless network, such as responding to a drop in video quality by lowering the PHY rate to decrease the overall error rate [13], [14]. This strategy was tested on low-resolution videos (QCIF and CIF size), and while it was shown to improve quantitative video quality, reducing the data rate of HD video will also cause a drop in QoE.

Qin and Zimmermann used a multi-layer encoding where videos have a base layer and multiple enhancement layers to be added or dropped depending on the available bandwidth [15]. This technique is effective at preventing service interruptions and dealing with brief congestion on the network, but it may not be enough to provide an acceptable QoE under prolonged poor channel conditions.

A method that has been used with some success is a priority-based scheme that gives an advantage to real-time packets. Oh and Chen used 802.11e Enhanced Distributed Channel Access (EDCA) [16], and Fiandrotti *et al.* used the same functionality to give priority to packets containing I-frame data [17]. These techniques are able to provide better video stream performance under heavy congestion, but they do not detect or protect against hidden node collisions and channel background noise.

IV. The Proposed ASDCS Method

The proposed ASDCS method periodically performs *Channel Evaluation*, and if the channel is deemed unacceptable for video streaming, then ASDCS performs *New Channel Selection*. The following subsections discuss these two processes.

A. Channel Evaluation

There are several ways to assess the ongoing quality of the received video stream using only the transmitter's perspective. Intuitively, there is a connection between outgoing IP queue size and channel quality, and therefore received video quality, which has been confirmed by a number of previous studies [11], [12]. In addition, the various channel conditions that have an impact on the IP queue size are due to carrier sense and hidden nodes. The presence of carrier sense nodes forces the streaming node pair on the same channel to wait its turn, limiting the rate at which packets can leave the IP queue. On the other hand, the presence of hidden nodes can cause frame collisions and thus MAC layer retransmissions, which again limits the rate of packets leaving the queue. Based on this relationship, the proposed ASDCS derives the estimated packet delay and uses this information, in conjunction with the knowledge of which H.264 NAL units have the most impact on received video quality, to predict the upcoming received video quality. The end result is that the channel bandwidth can be estimated based on the rate of change in IP queue size, which takes into account the cumulative effect of these various channel conditions.

To explain how ASDCS estimates the channel bandwidth, consider the following: Suppose a frame of size f_0 bytes is enqueued at time t_0 resulting in a total of Q_0 bytes in the IP queue. Packets will proceed to leave the queue as they are transmitted. Then, at time t_1 = t_0 + 1/*fps*, the transmitter enqueues f_1 bytes of the next frame, after which there are Q_1 bytes in the IP queue. This means that $Q_0 - (Q_1 - f_1)$ bytes successfully left the queue during an interval of 1/*fps*. If the IP queue did not empty before time t_1 (i.e., $Q_1 - f_1 > 0$), then the *estimated transmitter bandwidth* during the 1/*fps* period (BW_{est}^{Tx}), which represents the rate at which bytes were sent from the queue, can be calculated using the following equation:

$$BW_{est}^{Tx} = \frac{Q_0 - (Q_1 - f_1)}{1/fps}. \quad (1)$$

In practice, BW_{est}^{Tx} will fluctuate from frame to frame, and a brief downward spike in estimated available bandwidth does not necessarily indicate that network conditions have deteriorated significantly. Therefore, a simple moving average $avgBW_{est}^{Tx}$ of several recent estimated bandwidth samples is used to avoid false positives.

When all the packets of a frame are enqueued together, each packet's delay can be estimated by using $avgBW_{est}^{Tx}$. The *estimated delay* of the i^{th} packet ($Delay_{est}^{pkt_i}$) is the time required to transmit the packet itself *plus* the time to transmit all the packets ahead of it in the IP queue, and is modeled using the following equation:

$$Delay_{est}^{pkt_i} = \frac{Q + p_i}{avgBW_{est}^{Tx}}, \quad (2)$$

where Q is the number of bytes currently ahead of packet i in the queue and p_i is the size of packet i. Any packet with a $Delay_{est}^{pkt}$ value greater than the jitter parameter (typically 150 ms) is predicted to be lost.

The proposed ASDCS performs *Video Quality Prediction* by looking at which packets are predicted to be lost, and estimating whether the Peak Signal-to-Noise Ratio (PSNR) of the received video's frames will be above or below 40 dB (a well-known acceptable PSNR threshold [18]). Not all packets of H.264 video have an equal impact on received quality, so the estimate is based on the relative importance of the H.264 NAL units in the predicted lost packets. If any packets containing a Sequence Parameter Set (SPS) or Picture Parameter Set (PPS) are predicted to be lost, the received video quality is considered unacceptable. In addition, video quality is considered unacceptable if the predicted packet loss for I-frames, P-frames, or B-frames exceeds certain thresholds ($I_{lost} = 0.2$ frames, $P_{lost} =$ 2 frames, B_{lost} = 2 frames, respectively). These thresholds were determined experimentally to represent a conservative estimate of the packet loss necessary to result in PSNR dropping below 40 dB.

B. Channel Selection

To select a new channel for video streaming, ASDCS performs a two-step process of active scanning to detect AP nodes and passive listening to detect non-AP nodes, called *Active Link Detection*, to ultimately generate a list of nearby nodes called a *Visible Node List* that also contains their signal strengths.

When a transmitter and receiver pair each perform this process individually, their visible node lists may differ depending on the relative position of nearby nodes. Therefore, the streaming nodes compare their Visible Node Lists to identify which nodes are carrier sense nodes (i.e., visible to both) and which are hidden nodes (i.e., only visible to one half of the pair). This supplemental information is added to create a *Node Neighborhood List* that is useful for tracking which type of interference will come from which nodes and is used to evaluate the relative qualities of each channel.

Some of the hidden nodes in the Node Neighborhood List may have signal strengths that are too low to have significant impact on the SINR at the receiver. Similarly, some of the carrier sense nodes in the list may have signal strengths that are too low to require negotiation for access to the medium. The hidden nodes and carrier sense nodes that meet these criteria are excluded from the Node Neighborhood List. Finally, the remaining pared-down list is used by the transmitter to decide which new channel to select.

ASDCS uses a *node score* to evaluate prospective channels, with the lowest node score indicating the channel most likely to provide acceptable video streaming quality. Each hidden node counts for 2 points, while each carrier sense node counts for 1 point. In the event of a node score tie, the channel with the fewest carrier sense nodes will be selected. If a tie still exists, the channel with the lowest average background interference will be selected. The 2:1 scoring scheme and tie-breaking methods were chosen by observing, across multiple scenarios, the relative impact that carrier sense nodes and hidden nodes (of the same bitrate) have on received PSNR.

1) Channel Negotiation: While ASDCS's Channel Evaluation is based on transmitter-only observations, the Active Link Detection process used for Channel Selection requires some information-sharing between the transmitter and receiver. For

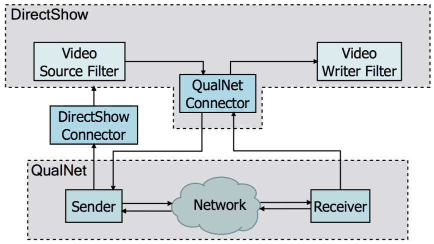


Fig. 2: The general structure of OEFMON.

this reason, ASDCS implements a TCP-based message exchange procedure to send the receiver's Visible Node List to the transmitter, and also to notify the receiver of the selected new channel.

In brief, this procedure consists of a collection of timeouts, acknowledgement packets, control states, and a prioritized IP queue for control packets. Our simulation results have shown that including this message exchange process in ASDCS helps ensure that the new channel is selected based on information from the perspectives of *both* devices without significantly increasing the total delay incurred by the channel switch.

V. Experimental Setup

A. Simulation Environment

The proposed ASDCS was implemented and evaluated using *Open Evaluation Framework for Multimedia Over Networks* (OEFMON) [19], which integrates Microsoft multimedia framework DirectShow [20] and network simulator QualNet 7.3 [21]. Together, they provide visualization of the underlying network details and on-the-fly display of sent and received videos.

A simplified diagram of OEFMON is shown in Fig. 2, which requires the following inputs: A video file in raw YUV or encoded H.264 format, a QualNet scenario file, a QoS mapping parameter file, and a DirectShow graph. Three outputs generated during simulation are a sender log, a receiver log, and a received video file. The logs are used after simulation to compute network performance metrics such as throughput, delay, and packet loss ratio. The received video file is used (along with the input video file) to compute the PSNR of each received frame.

For our simulation study, OEFMON was provided with H.264 encoded videos as input, which were sent via Real-time Transport Protocol (RTP) over UDP. The two test videos used were a 13-second clip of the *African Cats* trailer (1080p, 6 Mbps VBR, 30 fps) and a 45-second clip of the *Life of Pi* trailer (1080p, 6 Mbps VBR, 24 fps). All simulations were performed over 802.11g at its 54 Mbps maximum transmission rate. Background traffic was generated at a constant bitrate (CBR), and the amount of traffic on "bad" channels was calibrated so that each video streamed would experience no packet loss in low bitrate sections of the VBR video and medium to high packet loss (due to missed playout deadlines) in high bitrate sections. On average, the cumulative CBR and average VBR totaled 20 Mbps.

B. Channel Scenarios

Fig. 3 shows three of the five different network scenarios simulated to assess the performance of ASDCS. Scenario 1 shown in Fig. 3a consists of one hidden node pair and two carrier sense node pairs. This single channel scenario was used along with Scenarios 2 and 3 (not shown), which had carrier sense nodes only and hidden nodes only, respectively, to test the Video Quality Prediction portion of ASDCS (see Sec. IV-A). Scenarios 4 and 5 represent multi-channel networks that were used to assess the Active Link Detection portion of ASDCS (see Sec. IV-B). Scenario 4 shown in Fig. 3b contains a video streaming node pair initially on channel 1, a hidden node pair on channel 1, a hidden node pair on channel 6, and no transmitting nodes on channel 11. Scenario 5 shown in Fig. 3c is the same as Scenario 4, but one carrier sensing node pair is added to channel 11. Note that nodes 6 and 7 in the multi-channel scenarios are APs and all other nodes are non-AP.

To limit the frequency of channel changes, a channel switch backoff time of 10 seconds was used. Also, only the non-overlapping channels 1, 6, and 11 were scanned when searching for a new channel, as QualNet does not currently include channel overlap interference in its physical layer model.

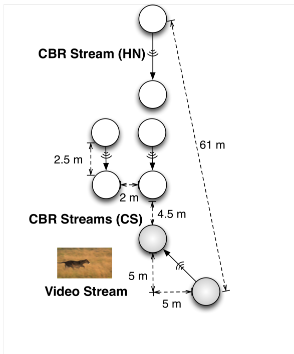
VI. Results

A. Channel Evaluation Results

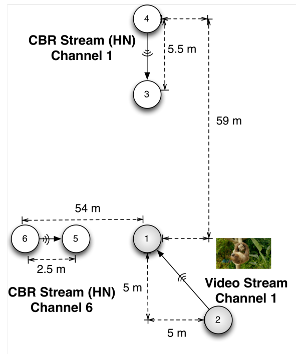
The first set of simulations was performed to demonstrate the accuracy of the Video Quality Prediction of ASDCS. A sample of these results (using the *African Cats* video) is shown in Fig. 4, where for each scenario the top plot represents the actual received video PSNR, the middle plot represents the queue-threshold prediction method presented in [12], and the bottom plot represents ASDCS's prediction of video quality. Note that the PSNR of a perfectly-received frame is represented as 111 dB.

The queue-threshold prediction method is based on the relationship between queue length and channel quality, where if the number of bytes in the queue exceeds a fixed threshold, the channel's quality is poor enough that received video quality will be unacceptable. For our comparison, a reasonable threshold of 50% was identified experimentally. A threshold level much lower than 50% was found to be too sensitive to network conditions and increases the number of "false positive" predictions of unacceptable PSNR, while a threshold level much higher than 50% was found to react too slowly to poor channel conditions, or even miss them completely.

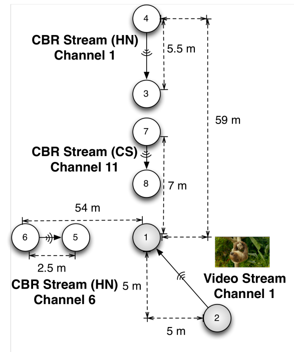
Fig. 4a shows the results for Scenario 1. Both methods miss a drop in quality that occurs shortly after the 4^{th} second, but later ASDCS detects the next drop just after the 6^{th} second sooner and more accurately than the queue-threshold prediction method. ASDCS missed the first drop in quality because the number of I-frames predicted to be lost was below the I_{lost} threshold, i.e., ASDCS underestimated the PSNR degradation caused by the I-frame packets that were predicted lost. The rapid changes in queue-threshold predictions for this scenario indicate an interesting event where the queue briefly surpasses the threshold when the next frame is enqueued, then drops



(a) Scenario 1: Single channel with one hidden and two carrier sense node pairs.

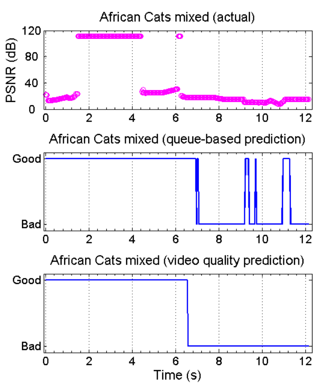


(b) Scenario 4: Multi-channel with two hidden node pairs.

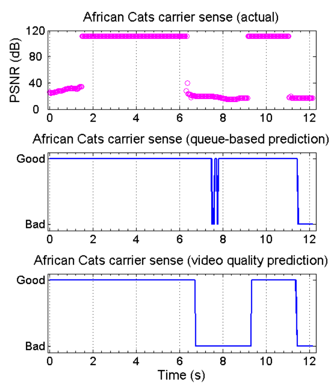


(c) Scenario 5: Multi-channel with two hidden and one carrier sense node pairs.

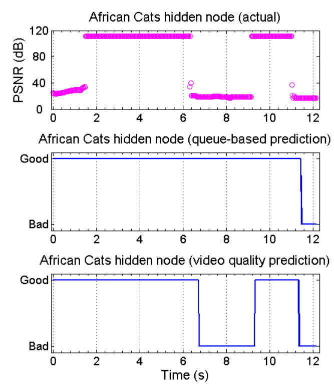
Fig. 3: Simulation scenarios.



(a) Scenario 1



(b) Scenario 2



(c) Scenario 3

Fig. 4: Actual vs. predicted video quality for the *African Cats* video in the three interference scenarios.

below the threshold as packets are sent, and then surpasses the threshold again when the next frame is enqueued.

Fig. 4b shows the results for Scenario 2. Both ASDCS and the queue-threshold prediction method detect the drop in quality that occurs shortly after the 6^{th} second, but ASDCS identifies the quality degradation sooner and also more accurately detects its duration. The next drop in quality that occurs at the 11^{th} second of streaming is predicted equally well by both methods.

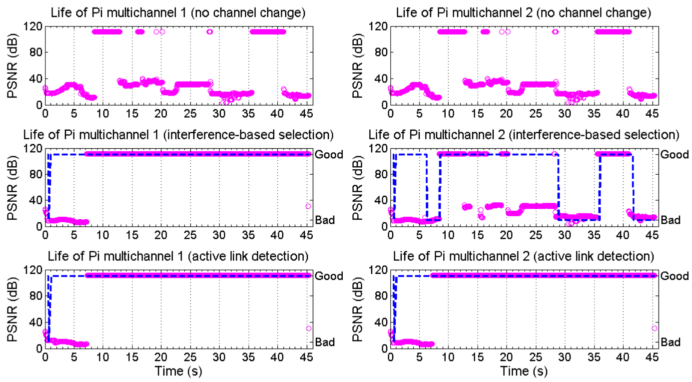
For the Scenario 3 results shown in Fig. 4c, ASDCS reliably predicts the received video quality, while the queue-threshold prediction method completely misses the drop in quality that begins right after the 6^{th} second. This indicates that there were queueing delays significant enough to degrade the received video quality, but the queue was empty enough when the delays began that the queue-threshold was never exceeded.

Our additional simulation results, which use the *Life of Pi* video with Scenarios 1$\sim$3, confirm that ASDCS's Video Quality Prediction is consistently better, and almost always no worse than the static queue-threshold prediction method. Intuitively, a static threshold will only outperform Video Quality Prediction if the selected threshold just happens to be at the ideal level for a particular combination of test video and network scenario.

B. Channel Selection Results

The second set of simulations was performed to demonstrate the effectiveness of Active Link Detection in two multi-channel scenarios. Note that initial Channel Selection is disabled for these scenarios in order to highlight the performance of the mid-stream channel switch. Simulation results using the *Life of Pi* video in both scenarios are shown in Fig. 5. Results from the Active Link Detection method are shown alongside the original PSNR when the channel was not changed, and the PSNR using a channel selection method that selects the channel with the lowest measured interference (interference-based selection), which is a straightforward and simple method as suggested by [10]. Predicted Video Quality is overlaid as a dotted line.

Fig. 5a shows the results for Scenario 4 in Fig. 3b, which shows that both Active Link Detection and the interference-based method were able to select a new channel that provided acceptable video quality (i.e., PSNR $>$ 40 dB). In fact, both methods selected channel 11 that had no other nodes. In this scenario, ASDCS was able to perform Channel Selection while pausing the video stream for only 380 ms. This channel switch pause time represents the cumulative amount of time needed



(a) Scenario 4. (b) Scenario 5.

Fig. 5: Actual vs. predicted video quality for the *Life of Pi* video in the two multi-channel scenarios.

to: perform active probing and passive listening, compare TX and RX Visible Node Lists and generate a Node Neighborhood List, compute each channel's node score, and then physically tune to the new channel. Most often, the channel switch pause time for ASDCS was in the range of 200∼400 ms, and it never took more than 770 ms to switch channels. When both ASDCS and the simple interference-based method chose the same new channel, the interference-based method actually outperformed ASDCS (in the sense that it completed the switch sooner) by virtue of its simplicity.

Fig. 5b shows the results for the more complicated scenario in Fig. 3c. This second scenario was an interesting 'non-ideal' test case because no channels were completely free of interference. In this case, the interference-based method incorrectly switched to a channel with a hidden node (i.e., channel 6), and continued to experience reduced video quality throughout the rest of the simulation. In contrast, ASDCS correctly predicted that the hidden node on channel 6 would be worse for the video quality than the single nearby carrier sense node on channel 11, and as a result, the received video quality while streaming on channel 11 stayed acceptable for the rest of the scenario.

VII. Conclusion

This paper presented ASDCS, which performs dynamic channel switching based on the specific characteristics of real-time streaming video. ASDCS's Video Quality Prediction exploits the connection between queue removal rate and delay to accurately predict when important video stream packets will miss their deadline. ASDCS's Active Link Detection characterizes all available channels and ranks them according to their expected impact on video quality.

Our simulation study results show that (1) ASDCS is better at predicting the received video quality than a static queue-threshold method, (2) ASDCS is better than receiver SINR-based methods at identifying which new channel is most likely to allow for good received video quality, and (3) ASDCS promotes good user QoE by switching to a channel within a reasonable time frame of 200∼770 ms.

Future work will include testing more advanced scenarios, incorporating node mobility, upgrading to 802.11n or 802.11ac, and implementing ASDCS in real hardware.

References

[1] C. Yoon, T. Um, and H. Lee, "Classification of N-Screen Services and its standardization," in *14th International Conference on Advanced Communication Technology (ICACT)*, 2012, pp. 597–602.

[2] Wireless Display Standards Explained: AirPlay, Miracast, WiDi, Chromecast, and DLNA. [Online]. Available: http://www.howtogeek.com/177145/

[3] "IEEE Std 802.11n^{TM}-2009, IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks - Specific Requirements," 2009.

[4] F. Y. Li, A. Kristensen, and P. Engelstad, "Passive and active hidden terminal detection in 802.11-based ad hoc networks," in *IEEE INFOCOM*, 2006.

[5] R. M. Kortebi, Y. Gourhant, and N. Agoulmine, "On the use of SINR for interference-aware routing in wireless multi-hop networks," in *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, 2007, pp. 395–399.

[6] M. Kim and C.-H. Choi, "Hidden-Node Detection in IEEE 802.11n Wireless LANs," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2724–2734, 2013.

[7] M. Sarkar, S. Nagaraj, and I. H. Balsania, "A SINR based MAC layer protocol for multi-channel ad-hoc networks," in *7th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2011, pp. 1889–1893.

[8] S. Wiwatthanasaranrom and A. Phonphoem, "Multichannel MAC protocol for ad-hoc wireless networks," in *Proceedings of the National Computer Science and Engineering Conference*, 2003, pp. 115–120.

[9] J. Mo, H.-S. So, and J. Walrand, "Comparison of Multichannel MAC Protocols," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 50–65, 2008.

[10] V. Kanodia, A. Sabharwal, and E. Knightly, "MOAR: A multi-channel opportunistic auto-rate media access protocol for ad hoc networks," in *First International Conference on Broadband Networks*, 2004, pp. 600–610.

[11] G. Athanasiou, I. Broustis, T. Korakis, and L. Tassiulas, "LAC: Load-aware channel selection in 802.11 WLANs," in *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2008, pp. 1–6.

[12] M. Ihmig and P. Steenkiste, "Distributed dynamic channel selection in chaotic wireless networks," in *13th European Wireless Conference*, 2007.

[13] M. H. Manshaei, T. Turletti, and T. Guionnet, "An evaluation of media-oriented rate selection algorithm for multimedia transmission in MANETs," *EURASIP Journal on Wireless Communications and Networking*, vol. 2005, no. 5, pp. 757–773, 2005.

[14] G. C. Lee and H. Song, "An effective cross layer-based video streaming algorithm over mobile ad hoc network," in *6th Consumer Communications and Networking Conference (CCNC)*, 2009, pp. 1–5.

[15] M. Qin and R. Zimmermann, "An adaptive strategy for mobile ad hoc media streaming," *IEEE Transactions on Multimedia*, vol. 12, no. 4, pp. 317–329, 2010.

[16] B. J. Oh and C. W. Chen, "Performance evaluation of H. 264 video over ad hoc networks based on dual mode IEEE 802.11 B/G and EDCA MAC architecture," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2008, pp. 3510–3513.

[17] A. Fiandrotti, D. Gallucci, E. Masala, and E. Magli, "Traffic Prioritization of H.264/SVC Video over 802.11e Ad Hoc Wireless Networks," in *Proceedings of 17th International Conference on Computer Communications and Networks (ICCCN)*, 2008, pp. 1–5.

[18] V. V. Lukin, M. Zriakhov, N. Ponomarenko, S. Krivenko, and M. Zhenjiang, "Lossy compression of images without visible distortions and its application," in *10th IEEE International Conference on Signal Processing (ICSP)*, 2010, pp. 698–701.

[19] C. Lee, M. Kim, S. J. Hyun, S. Lee, B. Lee, and K. Lee, "OEFMON: An open evaluation framework for multimedia over networks," *IEEE Communications Magazine*, vol. 49, no. 9, pp. 153–161, 2011.

[20] DirectShow. [Online]. Available: http://msdn.microsoft.com/en-us/library/ms783323.aspx

[21] QualNet — SCALABLE Network Technologies. [Online]. Available: http://web.scalable-networks.com/content/qualnet