Reinforcement Learning-Based Opportunistic Routing for Live Video Streaming over Multi-Hop Wireless Networks

Kexin Tang*, Chenglin Li†, Hongkai Xiong*, Junni Zou* and Pascal Frossard†
*MIN, Shanghai Jiao Tong University, China
Email: {tkx1994-china, xionghongkai}@sjtu.edu.cn
zou-jn@cs.sjtu.edu.cn
†LTS4, EPFL, Switzerland
Email: {chenglin.li, pascal.frossard}@epfl.ch

***Abstract*—Real-time video services are usually delay sensitive and have strict constraints on the transmission reliability, which poses challenges to live video streaming over multi-hop wireless networks, since the unpredictable packet losses and network congestions caused by time-varying wireless channels greatly degrade the received video quality. To address this, in this paper, we propose a reinforcement learning (RL)-based opportunistic routing (OR) scheme for wireless video streaming with high-reliability and low-delay requirements. It can exploit the broadcast nature of the wireless shared medium and path diversity through OR to improve the transmission reliability, and find the low-delay paths between the source-destination pair dynamically for video packets through the RL module embedded in each relay node. Specifically, we design for the OR a new path-cost metric called the expected anypath delay (EAD), to estimate the end-to-end delay of a packet between the current relay node and the destination. The EAD is dynamically measured and updated over time, thereby reflecting the changes of link quality and the congestion level at the relay node. Moreover, we utilize the ACK message to piggyback the EAD of each relay node to its previous-hop node. Based on the local communication of the EADs from the neighbors, each node in the network can iteratively and independently run the RL module to update its own EAD value. Then, the next-hop forwarder node on a low delay route can be determined by assigning higher relay priority to the candidate forwarder nodes with lower EADs in OR. Simulation results show that the proposed RLOR algorithm can achieve a proper tradeoff between the transmission reliability and latency, so as to support the low-delay transmission of wireless video streams with high received video quality.**

I. Introduction

In recent years, real-time video streaming services, such as video conference, video surveillance, and live broadcast, have increasingly become important applications. Meanwhile, multi-hop wireless networks, e.g., wireless mesh networks (WMNs) [1], wireless sensor networks (WSNs) [2], and mobile ad hoc networks (MANETs), have attracted much attention for future mobile communication systems due to their ease of deployment, low infrastructure cost, and high flexibility of multi-hop and multi-path topologies. However, there are still significant challenges for live video streams over multi-hop wireless networks. On the one hand, live video streaming has stringent requirements on the transmission. For example, video contains massive data and requires a large bandwidth consumption to guarantee an acceptable viewing quality. Besides, unlike other media objects (audio, image, etc.), the transmission of video is more prone to the packet loss due to the encoding/decoding dependency among consecutive video packets. In addition, there also exists a low end-to-end delay constraint, since video packets that arrive at the decoder too late to be decoded before the scheduled display time are useless and considered lost. On the other hand, the inherent temporal variation and error prone properties of wireless channels usually incur high packet loss and high latency to the video transmission, which greatly degrades the received video quality. Therefore, it is interesting yet challenging to design an appropriate routing scheme for live video streaming over multi-hop wireless networks that enables reliable and low-delay transmission.

Traditional wireless routing protocols, such as ad hoc on-demand distance vector (AODV) [3] and optimized link state routing (OLSR) [4], usually pre-select an optimized route before transmission starts. These protocols actually inherit some path computation methods that are initially conceived for wired networks and adapted to meet the specific requirements of wireless networks. Nevertheless, the unreliable and time-varying nature of wireless medium significantly impairs their performance, resulting in substantial retransmission overhead. Opportunistic routing (OR), on the contrary, takes advantage of the broadcast nature of the wireless medium, regarding the wireless shared channel as an opportunity rather than a limitation. Instead of selecting a fixed next-hop relay node in advance, in OR, a node broadcasts a data packet to multiple neighbors and determines the next-hop forwarder on-the-fly among the nodes that successfully receive this packet based on their relay priority.

Biswas *et al.* [5] design and implement ExOR, the original

The work has been partially supported by NSFC under Grants 61501293, 61529101, 61425011, 61622112 and 61472234, the Program of Shanghai Academic Research Leader under Grant 17XD1401900, the China Postdoctoral Science Foundation under Grants 2016T90372 and 2015M570365, and the China Scholarship Council.

978-1-5090-3649-3/17/$31.00

OR protocol, which deals with the wireless packet losses and improves the throughput by a factor of two to four in multi-hop wireless networks compared to single-path routing. The metric used to select and prioritize relay nodes is extremely critical in opportunistic routing, and greatly affects the routing performance. As revealed by [6], commonly used metrics are based on the geographical distance or link quality, including the expected one-hop throughput (EOT) [7], the expected transmission count (ETX) [8] and the expected anypath transmission count (EAX) [9]. By adopting these metrics, most of the existing OR schemes are designed either to increase the overall network throughput, or to decrease the number of retransmission attempts (i.e., to increase the reliability).

Besides throughput/reliability, the end-to-end delay of a video packet is another important measure affecting the overall performance of the live video streaming services. However, to the best of our knowledge, studies to date have seldom taken this factor into account to minimize the total end-to-end delay in OR. In [10], a video-aware multicast opportunistic routing protocol is proposed by extending the network coding-based OR method, MORE [11]. However, it does not fit a more complicated scenario where multiple coexisting flows are competing for the shared network resource and the load balancing becomes an imperative problem. When multiple video flows are disseminated over wireless networks, the unavoidable network congestion will incur a long queuing delay and potential buffer overflow, which degrades the video quality sharply. In practice, the network congestion conditions vary with traffic pattern generated randomly by users, which requires each node to actively learn the dynamic change of the network based only on the local communication. To this end, Boyan *et al.* [12] propose the Q-routing algorithm, the first work to apply reinforcement learning (RL) in the routing protocol design for finding the minimum delay path in wired networks. Based on Q-routing, a QoE-aware dual RL routing strategy is designed in [13] to dynamically adjust the routes for different multimedia service flows. However, the dilemma between exploration and exploitation becomes more prominent when applying the RL algorithm to the wireless routing protocol, since the frequent exploration phases required to capture the dynamic network changes will result in a large amount of probe packet overhead that is not affordable by the limited wireless resource.

To address the above issues, we propose a reinforcement learning-based opportunistic routing (RLOR) scheme for live video streaming over multi-hop wireless networks, by utilizing the broadcast nature of OR and integrating the periodical probe packets in the exploration phase of RL into the hop-by-hop ACK messages. The proposed RLOR scheme can simultaneously capture the wireless link variation over time and dynamically detect the network congestion to achieve low delay transmission for wireless video streaming. Specifically, we design a new path-cost metric called the expected anypath delay (EAD), which is defined as the estimated end-to-end delay of a packet between the current relay node and the destination node. Then, each node utilizes the acknowledgment (ACK) message that piggybacks the EADs of its neighboring nodes to independently update its own EAD based on RL algorithm. As a result, the low delay routes can be learned by dynamically updating the EAD of each node, and by choosing the appropriate candidate forwarder set and the relay priority of each candidate forwarder according to the learned EADs. Moreover, the learning is continual and online using only local information. The proposed scheme is therefore able to adapt well in time-varying wireless networks. We conduct extensive experiments on the discrete event simulator NS-3 [14]. The simulation results show that the proposed RLOR algorithm can achieve a lower end-to-end delay for video packet transmission over multi-hop wireless networks while the video viewing quality is still higher than existing schemes, which is suitable for live video streaming applications.

The remainder of the paper is organized as follows. In Section II, we introduce the system model and the basic module of OR. In Section III, we design a new path-cost metric called EAD and develop a RLOR algorithm. Section IV presents our experimental configuration and simulation results. Finally, concluding remarks are given in Section V.

II. System Model

A. Network Model and Notations

As illustrated in Fig. 1, we model the multi-hop wireless network as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ is the set of wireless links. In accordance with the packet transmission, we assume that the time is slotted and indexed by $t \in \{0, 1, 2, \cdots\}$. Within each time slot t, a complete packet broadcasting and feedback procedure are accomplished between the sender node $i \in \mathcal{V}$ and the ordered candidate forwarder set (CFS) of i, denoted by $F_i(t)$. The CFS $F_i(t)$ comprises the neighboring nodes of i that could further forward the packet to destination and is ranked in a descending order of their relay priority. The set of wireless links between node i and its CFS $F_i(t)$ then constitutes a hyperlink $(i, F_i(t)) = \{(i, n_j) \in \mathcal{E} | \forall n_j \in F_i(t)\}$. Since wireless channels experience random packet losses due to fading, shadowing and interference, a packet might need multiple transmission attempts until the successful delivery over a wireless link. Accordingly, we denote by p_{i,n_j} the delivery probability of wireless link $(i, n_j) \in \mathcal{E}$. Thus, $1/p_{i,n_j}$ represents the number of expected transmissions for a successful packet delivery from i to n_j, i.e., the ETX over the link (i, n_j). In practice, each node i can use the ACK messages received from node n_j to estimate p_{i,n_j}. For the hyperlink $(i, F_i(t))$, we denote by $p_{i,F_i(t)}$ the hyperlink delivery probability, which is the probability that a packet transmitted from node i is successfully received by at least one of the nodes in its CFS $F_i(t)$. With the assumption of independent deliveries, $p_{i,F_i(t)}$ is formulated as

$$p_{i,F_i(t)} = 1 - \prod_{n_j \in F_i(t)} (1 - p_{i,n_j}). \tag{1}$$

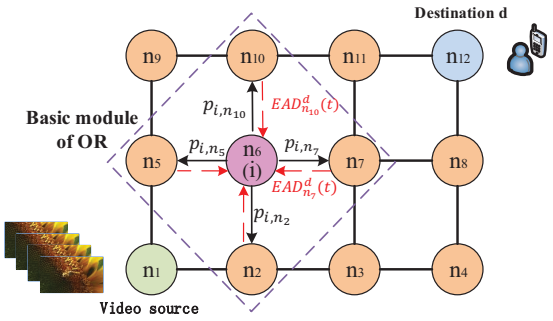

Fig. 1. Illustration of live video streaming over a multi-hop wireless network.

Likewise, $1/p_{i,F_i(t)}$ refers to the expected number of transmissions for a successful packet delivery from the node i to the CFS $F_i(t)$.

B. Basic Module of Opportunistic Routing

In traditional wireless routing, each node pre-computes the minimum cost path to the destination, and then forwards to a fixed next-hop node all packets that are targeted to the destination. Therefore, the path diversity is ignored and under-utilized. If a transmission fails, the sender node needs to retransmit the same packet to the fixed relay node even though other nodes may have overheard it, thereby failing to exploit the broadcast nature. In contrast, OR determines the next-hop relay node for a packet on the fly, with a packet forwarding process comprising the following three steps, as illustrated in Fig. 1.

1) Candidate forwarder set (CFS) selection: Sender node i compares at each time slot t its transmission cost (e.g., EAD, ETX, EAX) with its neighbors, and then selects the adjacent nodes with lower cost to constitute its CFS $F_i(t)$. The nodes in $F_i(t)$ are further ranked in a descending order of their relay priority, with a lower cost indicating a higher relay priority. For example, here we have the CFS $F_i(t) = \{n_7, n_{10}\}$ if we assume that $EAD^d_{n_7}(t) < EAD^d_{n_{10}}(t) < EAD^d_i(t) < EAD^d_{n_2}(t) < EAD^d_{n_5}(t)$. **2) Data broadcast:** The sender node appends the ranked list of the CFS $F_i(t)$ into the packet and then broadcasts this packet. **3) Forwarder coordination:** If multiple nodes receive this packet, the nodes out of $F_i(t)$ discard this packet directly. The nodes that are within $F_i(t)$ and receive this packet successfully will then relay the packet based on the order of relay priority. To avoid duplicated transmission, we specify that a node in $F_i(t)$ only forwards a packet when it successfully receives the packet and all the nodes with higher relay priority fail to do so.

In this paper, the forwarder coordination is practically implemented by the acknowledgement (ACK) messages and relay node (RN) messages, as follows. At each time slot t, sender node i broadcasts a data packet appended with the CFS. Then, all nodes in the CFS that have successfully received this packet send an ACK message back to the node i. After receiving the ACK messages, the node i broadcasts a RN message to announce which candidate node actually forward this packet. If there is no node in the CFS receiving this packet, the node i will retransmit this packet until at least one of its neighboring nodes within the CFS receives it or the retransmission count exceeds the upper limit. This procedure is repeated on each relay node traversed by this packet until it reaches the destination. Besides, we assume that the ACK and RN messages are transmitted without error, and that due to the packet size difference, their transmission time is negligible compared to the data packet transmission delay.

III. Reinforcement Learning Based Opportunistic Routing

A. Path-cost Metric

The metric of path-cost significantly influences the choice of the relay nodes in the CFS, and therefore impacts the ultimate network performance. Considering that delay is the critical factor for live video transmission, we propose a novel path-cost metric called the expected anypath delay (EAD) to estimate the total delivery time of a packet sent from the current node to a given destination. In general, EAD consists of three delay components, namely, the queue delay at the current node, the expected one-hop transmission delay, and the expected delivery delay on the remaining path. The EAD of each node reflects the congestion level and delivery ability of the current node, as well as the corresponding measure of its relay node iteratively.

We denote by $q_i(t)$ the instant queuing delay for node i at time slot t, and adopt the moving average method to estimate the current waiting time $Q_i(t)$ of a packet in the queue of node i at MAC layer, based on the previous queueing delay:

$$Q_i(t) = \frac{\sum_{k=0}^{M-1} q_i(t-k)}{M}, \tag{2}$$

where M is the size of the sliding window.

Furthermore, we define the expected one-hop transmission delay $T_{i,F_i(t)}(t)$ to estimate the time spent for a packet to be successfully transmitted from sender node i to at least one node in its CFS $F_i(t)$:

$$T_{i,F_i(t)}(t) = \frac{1}{p_{i,F_i(t)}} \times \frac{S}{R}, \tag{3}$$

where $p_{i,F_i(t)}$ is the delivery probability of the hyperlink $(i, F_i(t))$, S and R are the size of the packet in bits and the data transmission rate in bps, respectively. The ratio S/R then indicates the time of a single transmission over the hyperlink $(i, F_i(t))$. Here, the definition of $T_{i,F_i(t)}(t)$ is a generalization of the expected transmission time (ETT) [15] in the traditional wireless routing, by substituting a single next-hop node with a CFS $F_i(t)$.

The expected delivery delay on the remaining path can be interpreted as the EAD of the CFS $F_i(t)$ to the destination. It can be calculated by a weighted sum of the EAD of the nodes in the CFS $F_i(t) = \{f_1(t), f_2(t), \cdots, f_r(t)\}$, where $f_j(t)$ represents the node with the j-th highest relay priority, and r denotes the total number of candidate relay nodes in the

CFS $F_i(t)$. Thus, we can define the estimated delay from the CFS $F_i(t)$ to the destination node d as

$$EAD^d_{F_i(t)}(t) = \sum_{f_j(t) \in F_i(t)} \omega_j \cdot EAD^d_{f_j(t)}(t), \quad (4)$$

where the weight ω_j represents the probability of the node $f_j(t)$ being selected as the actual relaying node. This only happens when node $f_j(t)$ successfully receives the packet and none of nodes with a higher priority than $f_j(t)$ succeeds to do so. Therefore, this probability can be formulated as

$$\omega_j = \frac{p_{i,f_j(t)} \cdot \prod_{k=1}^{j-1} (1 - p_{i,f_k(t)})}{p_{i,F_i(t)}}, \quad (5)$$

with the denominator being the normalization constant. By combining Eqs. (2)-(4), we can estimate the expected anypath delay of node i to a given destination d at time slot t as

$$EAD^d_i(t) = Q_i(t) + T_{i,F_i(t)}(t) + EAD^d_{F_i(t)}(t). \quad (6)$$

B. RLOR Algorithm Design

In OR, the selection of the CFS for a sender node based on the EADs of its neighbors plays a key role in determining the actual next-hop forwarder of a packet and its path to the destination. Therefore, the routing performance is greatly affected by the estimation accuracy of the EAD of each node. However, due to the time-varying characteristics of wireless networks, these EADs change dynamically over time. In addition, there is no "training signals" for evaluating these EAD values and improving the routing policy until a packet finally reaches the destination. To address the above issues, reinforcement learning (RL) provides a framework for each node to quickly learn a good estimation of its EAD by using only local information through the interaction with its neighbors. This is achieved within the basic data packet broadcast and ACK feedback process of OR. Specifically, after transmitting a packet, the sender node will use the received ACK messages containing the EADs of the nodes in its CFS to update the estimate of its own EAD based on Eq. (6), which is denoted as $\widehat{EAD^d_i}(t)$. It then runs the RL update procedure to revise its EAD value based on this estimate:

$$EAD^d_i(t) = EAD^d_i(t-1) + \mu(\widehat{EAD^d_i}(t) - EAD^d_i(t-1)), \quad (7)$$

where parameter μ is the learning rate. The reinforcement learning-based opportunistic routing (RLOR) algorithm is proposed in Algorithm 1, and explained in the following.

At time slot $t = 0$, we assume that no data traffic exists in the network. Therefore, in the initialization step, we set the queuing delay in all nodes to zero. Then, each node broadcasts some probe packets to measure the link delivery probability. After that, we employ the anypath Bellman-Ford (ABF) algorithm in [16] with the EAD metric to initialize the network. Since any path in the network cannot exceed $|\mathcal{V}| - 1$ hops, the ABF algorithm consists of at most $|\mathcal{V}| - 1$ rounds. At each round, each node $i \in \mathcal{V}$ gets its neighbors using the GetNeighbors(i) function, stores them to the set $\mathcal{C}$,

Algorithm 1 RLOR scheme.

Initialization step: (at $t = 0$, for destination d)
(1) $EAD^d_d(0) \leftarrow 0$
(2) **for each node** $i \in \mathcal{V} \setminus \{d\}$ **do**
  $EAD^d_i(0) \leftarrow \infty$, $F_i(0) \leftarrow \varnothing$, $Q_i(0) \leftarrow 0$
(3) **for** $m \leftarrow 1$ **to** $|\mathcal{V}| - 1$
  for each node $i \in \mathcal{V}$ **do**
    $J \leftarrow \varnothing$, $\mathcal{C} \leftarrow$ GetNeighbors(i)
    while $\mathcal{C} \neq \varnothing$ **do**
      $j \leftarrow$ ExtractMin($\mathcal{C}$), $J \leftarrow J \cup \{j\}$
      if $EAD^d_j(0) < EAD^d_i(0)$ **then**
        $F_i(0) \leftarrow J$
        Compute $EAD^d_i(0)$ based on Eq. (6)
Iteration step: (at $t = 1, 2, \ldots,$ for destination d)
for each node $i \in \mathcal{V} \setminus \{d\}$ **do**
  (1) Node i broadcasts a packet appended with $F_i(t-1)$.
  (2) $\mathcal{H} \leftarrow$ GetAckNodes(i)
  (3) $J \leftarrow \varnothing$
  for each neighbor $n_j \in F_i(t-1)$ **do**
    if $n_j \notin \mathcal{H}$ **then**
      $EAD^d_{n_j}(t) \leftarrow EAD^d_{n_j}(t-1)$
    if $EAD^d_{n_j}(t) < EAD^d_i(t-1)$ **then**
      $J \leftarrow J \cup \{n_j\}$
    $F_i(t) \leftarrow$ Descend(J)
  (4) **Compute** the estimated $\widehat{EAD^d_i}(t)$ based on Eq. (6)
  (5) **Update** $EAD^d_i(t)$ base on the RL update Eq. (7)

and sets J as the temporary CFS. The ExtractMin(i) function extracts the neighboring node of i with the minimum EAD to destination d from $\mathcal{C}$ at each loop, which is denoted as j and added into J. Next, we check whether $EAD^d_j(0)$ is smaller than $EAD^d_i(0)$. If yes, we update $F_i(0)$ (the formal CFS) by J, and then compute $EAD^d_i(0)$ based on Eq. (6). After the update of EADs of all nodes, each node announces its new EAD to the neighbors, and then goes to the next round.

At time slot $t = 1, 2, \ldots,$ multiple flows are generated between multiple source-destination pairs. The algorithm then goes to the iteration step. Here, we assume that the node i receives a data packet to be delivered to the destination node d at time slot t. First, node i broadcasts the packet appended with the CFS $F_i(t-1)$, which is selected and prioritized based on the EADs of its neighbors to the destination d at last time slot $t-1$. Nodes whose EADs are less than $EAD^d_i(t-1)$ can be listed in the CFS with a descending order of the relay priority. Next, nodes that are within $F_i(t-1)$ and successfully receive the packet send an ACK message to node i, and piggyback their EADs. Then, the node i announces which node is responsible for actually forwarding the packet by a RN message. We apply the GetAckNodes(i) function to record the nodes within the CFS $F_i(t-1)$ that send back ACK messages containing their EADs to i and place them in the set $\mathcal{H}$. We can then update the CFS $F_i(t)$ as follows. Likewise, J still denotes the temporary CFS. For nodes that do not send the ACK messages to node i at time slot t, we still use their EADs at last slot time $t-1$ to represent the current EAD values. Here, we add $n_j \in F_i(t-1)$ into J only if $EAD^d_{n_j}(t)$ is smaller than $EAD^d_i(t-1)$. Then, we can get the formal CFS $F_i(t)$ by using the Descend(J) function to sort J in a descending order

of the relay priority. Correspondingly, the node i can obtain an EAD estimate $\widehat{EAD_i^d(t)}$ according to Eq. (6), and then use this estimate to update its EAD based on the RL update procedure in Eq. (7).

IV. Experiments

A. Settings

The discrete event simulator NS-3 [14] is used to simulate the wireless video streaming scenario with other multiple flows co-existing in the network. We compare the routing performance of the proposed RLOR algorithm with two existing OR algorithms using EAX (EAX-OR [9]) and ETX (ETX-OR [5]), and the traditional RL-based single-path routing method (RL-TR [13]), respectively. Nodes in the wireless network are located in an approximate 3×4 grid where the distance between each two nodes is randomly distributed in the range of [180, 185] m, as shown in Fig. 1. Each node has a random movement within a 6 m-by-6 m square area around its original location, resulting in a time-varying delivery probability of each wireless link. We take the IEEE 802.11b standard in ad-hoc mode as the communication specification. The data transmission rate of each wireless link is set to 11 Mbps without automatic rate control. Besides, we devise an appropriate channel assignment and the receiving gain to avoid the interference among adjacent nodes. Thus, our implementation is capable of sending and receiving data packets simultaneously for each node. In addition, we employ the log-distance propagation loss model to simulate the fading of signal on the wireless channel. The path loss is defined as $L = L_0 + 10n\log(\frac{dist}{dist_0})$, where $dist$ is the distance between the transmitter and receiver, $n = 3$ is the path loss exponent, $dist_0 = 1$ m and $L_0 = 40.046$ dB are the reference distance and the path loss at that reference distance, respectively. The maximum transmission power level is set to 20 dBm (100 mW). To estimate the packet loss, we utilize the NistErrorRateModel in the NS-3. We adopt the UDP format with 1040-byte payload in every data packet and limit the buffer size of each node to 300 packets.

Without loss of generality, we assume that there is one video flow transmitting from nodes n_1 to n_{12}. Two test video sequences (*Tractor* and *Sunflower*) with 1080p resolution (1920×1080), available at [17], are selected as the candidate video flows for transmission. These two videos are encoded at a frame rate of 60 fps, with an IPPPP Group of Picture (GOP) coding structure that includes one I-frame and 29 P-frames in one GOP. In addition, we select three source-destination pairs, $n_6 \rightarrow n_{12}, n_5 \rightarrow n_{11}, n_2 \rightarrow n_8$, respectively, to generate background data traffic at a fixed source rate. To transmit a video flow, we sequentially send the corresponding data packets at the source node, by letting the sending rate of data packets equal to the source rate of that video flow. The detailed simulation parameters are shown in Table I.

TABLE I
SIMULATION SETUP

Parameter	Value
Simulator	NS-3.26 [14]
Topology	3×4 grid
Distance between nodes	[180, 185] m
Nodes communication	802.11b
Propagation model	LogDistancePropagationLossModel
Error rate model	NistErrorRateModel
Remote station manager	ConstantRateWifiManager
WiFi data rate	11 Mbps
Source rate of the video flow	5 Mbps (unless stated otherwise)
Source rate of the other flows	2 Mbps
Transmission power	20 dBm (unless stated otherwise)
Packet size	1040 bytes
Buffer size	300 packets
Learning rate μ	0.5
Test video sequence	*Tractor* (or *Sunflower*)
Frame rate	60 fps
Resolution	1080p (1920×1080)
GOP coding structure	IPPPP, 30 frames

B. Results

Fig. 2(a) shows the average end-to-end delay of each packet versus the average throughput of the destination node n_{12} by varying the source rate of the video flow from 1 Mbps to 5 Mbps, when the transmission power of each node is set to 20 dBm. It can be seen that to achieve the same average throughput, the average end-to-end delay achieved by the other three competitor algorithms is much larger than the proposed RLOR algorithm. And such delay gap becomes more significant if the user in node n_{12} requests a higher video source rate. This is because the other two OR approaches prefer to select the most reliable transmission path and neglect the possible congestion at some popular nodes. In comparison, the RLOR algorithm is able to learn the network congestion in real time by dynamically updating the EAD of each node, and then to optimize the relay priority of the candidate forwarder nodes. Sometimes, for the low-delay transmission of video streaming, it is more appropriate to select the path with less congestion even at a cost of inferior link delivery ability. In other words, there exists a tradeoff between reducing the retransmission overhead and avoiding the queuing delay, and our algorithm is capable of balancing these two factors. We illustrate in Fig. 2(b) the case when the transmission power level is reduced to 19.5 dBm, where the performance of the RL-TR algorithm becomes the worst. This is because the RL-TR is a single path routing algorithm. When the link delivery probability drops with the decrease of the transmission power, there will be more packet losses and retransmission overheads on the fixed single path, which thereby results in a very long transmission delay.

In Fig. 3, we measure the instant delivery delay of packets and the throughput between the node pair $n_1 \rightarrow n_{12}$ with a time window of 0.2 s, when the video source rate is set to 5 Mbps. It can be seen that the delivery delay of packets in the other three methods fluctuate greatly. Especially, the ETX-OR algorithm presents a large delay at t=3 s and 6 s. The reason is that ETX is essentially a single-path metric and ignores to exploit the path diversity. In contrast, a stable and low end-to-end delay can be guaranteed for each video packet by the proposed RLOR algorithm, since it can balance the data traffic

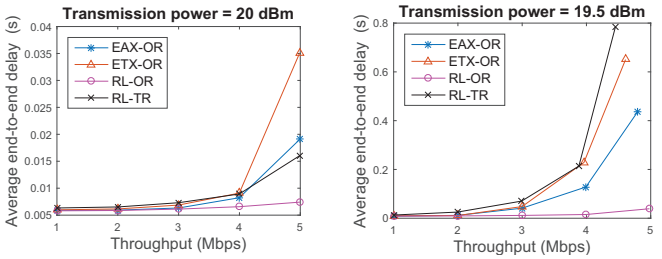

Fig. 2. Average end-to-end delay vs. throughput when the transmission power is set to (a) 20 dBm and (b) 19.5 dBm.

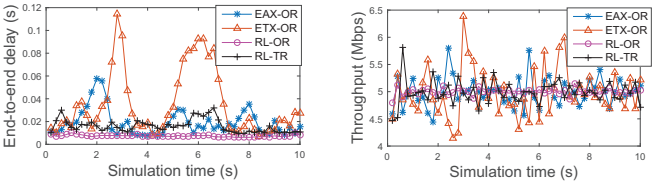

Fig. 3. End-to-end delay and throughput over simulation time.

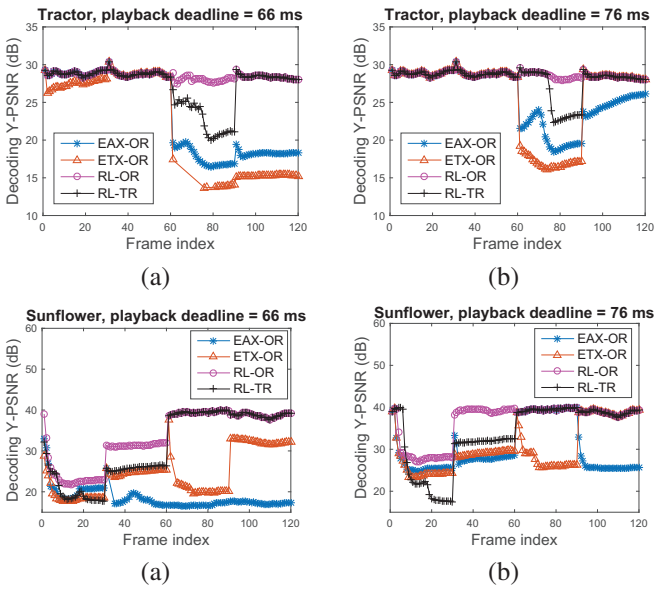

Fig. 4. Frame-wise decoding Y-PSNR when the playback deadline is set to (a) 66 ms and (b) 76 ms.

over all nodes within the network.

Fig. 4 shows the decoded video quality in Y-PSNR of the first 120 frames of *Tractor* and *Sunflower* sequences achieved by different algorithms under two playback deadline settings of 66 ms and 76 ms. Here, a video packet is considered lost if it arrives later than the given playback deadline. As the playback deadline increases, the delay constraint on each packet is relaxed, thereby leading to a better overall video quality for each algorithm. Overall, the proposed RLOR algorithm outperforms the other three algorithms when transmitting different videos, achieving both higher average Y-PSNR and smaller video quality variation over time.

V. Conclusions

In this paper, we proposed a reinforcement learning-based opportunistic routing (RLOR) scheme for live video streaming over multi-hop wireless networks. Specially, we designed a new path-cost metric, the expected anypath delay (EAD), to estimate the end-to-end delay of a packet between the current sending node and the destination. We then applied the proposed RLOR algorithm to efficiently learn the EAD of each node dynamically through reinforcement learning update procedure and to opportunistically forward the packets among wireless nodes based on their EADs. The simulation results have shown that the proposed RLOR algorithm is capable of balancing the network traffic, such that the low-delay requirement of the live video streaming over multi-hop wireless networks is guaranteed with better average video viewing quality and smaller temporal quality variation.

References

[1] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Communications Magazine*, vol. 43, no. 9, pp. S23–S30, Sep. 2005.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[3] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," IETF RFC 3561, Jul. 2003.

[4] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," IETF RFC 3626, Oct. 2003.

[5] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," in *Proc. ACM SIGCOMM*, 2005, pp. 133–144.

[6] N. Chakchouk, "A survey on opportunistic routing in wireless communication networks," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2214–2241, Fourthquarter 2015.

[7] K. Zeng, W. Lou, J. Yang, and D. R. B. Iii, "On throughput efficiency of geographic opportunistic routing in multihop wireless networks," *Mobile Networks and Applications*, vol. 12, no. 5, pp. 347–357, Dec. 2007.

[8] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. ACM MobiCom*, 2003, pp. 134–146.

[9] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Valuable detours: Least-cost anypath routing," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 333–346, Apr. 2011.

[10] K. Choumas, I. Syrigos, T. Korakis, and L. Tassiulas, "Video aware multicast opportunistic routing over 802.11 two-hop mesh networks," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2017.

[11] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proc. ACM SIGCOMM*, 2007, pp. 169–180.

[12] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: a reinforcement learning approach," *Advances in Neural Information Processing Systems*, vol. 6, pp. 671–678, 1994.

[13] R. Matos, N. Coutinho, C. Marques, S. Sargento, J. Chakareski, and A. Kassler, "Quality of experience-based routing in multi-service wireless mesh networks," in *Proc. IEEE ICC*, 2012, pp. 7060–7065.

[14] "Ns3 project (release ns-3.26). [online]. Available: http-s://www.nsnam.org/," 2016.

[15] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. ACM MobiCom*, 2004, pp. 114–128.

[16] R. Laufer, H. Dubois-Ferriere, and L. Kleinrock, "Polynomial-time algorithms for multirate anypath routing in wireless multihop networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 742–755, Jun. 2012.

[17] "Xiph.org video test media. [online]. Available: http://media.xiph.org/video/derf/," .