EV-CAST: Interference and Energy-Aware Video Multicast Exploiting Collaborative Relays

Yeonchul Shin‡, Jaewon Hur†, Gyujin Lee†, Jonghoe Koo‡, Junyoung Choi†, Sung-Ju Lee*, and Sunghyun Choi†
‡Samsung Electronics, Suwon, Korea
†Department of ECE and INMC, Seoul National University, Seoul, Korea
*School of Computing, KAIST, Daejeon, Korea
yc1015.shin@samsung.com, {jwhur, gjlee}@mwnl.snu.ac.kr, jh89.koo@samsung.com
jychoi@mwnl.snu.ac.kr, profsj@kaist.ac.kr, schoi@snu.ac.kr

***Abstract*—Video multicast over wireless local area network (WLAN) has been gaining attraction for applications sharing a venue-specific common video with multiple users. However, wireless multicast is limited by a receiver that has the weakest communication link to the source. Collaborative relaying could overcome this challenge by enabling selected receiver nodes to relay the packets from the source to other receivers. We propose `EV-CAST`, an interference and energy-aware video multicast system using collaborative relays, which entails (i) online topology management based on interference-aware link characterization, (ii) joint selection of relay nodes and transmission parameters, and (iii) polling-based relay protocol. Our proposed algorithm, the core of `EV-CAST`, judiciously selects the relay nodes and transmission parameters in consideration of interference, battery status, and spatial reuse. Our prototype-based experiment results demonstrate that `EV-CAST` enhances video multicast delivery under various network scenarios. `EV-CAST` enables 2× more nodes to achieve a target video packet loss ratio with 0.59× shorter airtime than the state-of-the-art video multicast scheme.**

I. Introduction

Multicast has been considered an attractive solution when sharing common data with multiple receivers. It is especially useful in wireless networks as the inherent broadcast nature of wireless channels could be utilized. A major application of multicast is streaming a venue-specific common video to multiple receivers, e.g., sharing a screen with multiple students in a classroom and streaming live events to the audience in a stadium. Along with the increasing popularity of unicast-based video streaming, video streaming via multicast, so called *video multicast*, has attracted the interest from both research community and industry practitioners.

When WLANs are deployed in large classrooms or stadiums, a single sender, i.e., an Access Point (AP) in the infrastructure mode or a source node in the ad hoc mode, can hardly provide complete coverage. However, additional deployment of infrastructure to extend the coverage requires extra cost, which might be inefficient especially when video multicast service is needed only for temporary events like conferences or concerts. Therefore, collaborative relaying that enables chosen receivers to relay the packets from the source node has been considered to enhance the coverage without additional deployment of infrastructure.

Most collaborative relaying schemes employ intra-flow network coding (NC) [1–4]. With NC, the source node generates N encoded packets for each batch of K original packets by taking linear combination of the original packets with random coefficients. K and N are called *batch size* and *generation size*, respectively. The relay nodes then re-encode the packets and forward them. By exploiting the possibility of reception over all neighboring links thanks to the broadcast nature, the reliability and relay gain improve significantly.

One major challenge in collaborative relaying is selecting the relay nodes and their transmission parameters (TPs), i.e., physical (PHY) rate and generation size. In selecting the relay nodes, the current battery status of nodes should be considered, as energy efficiency is critical in wireless networks and mobile devices. While nodes with high battery level might be willing to relay, other nodes with low battery level might not. Additionally, the charging status, i.e., whether a node is being charged or not, should also be considered since mobile battery chargers are becoming popular. Another important factor is spatial reuse. If we choose spatially reusable nodes, i.e., nodes that can transmit simultaneously without interfering with each other, we can provide video multicast more efficiently.

In determining TPs of the relay nodes, interference should be considered carefully. In [5], impact of the interference on the selection of TPs is addressed in a single-hop network. For multi-hop networks, impact of the interference is more complex in that different interferers make more diverse impact on the network, thus making the problem more challenging.

In this paper, we propose `EV-CAST`, an interference- and energy-aware video multicast system that exploits collaborative relaying. `EV-CAST` entails an elaborate design fitted for video streaming over two-hop wireless networks that concerns (i) online topology management, (ii) joint selection of relay nodes and their TPs, and (iii) polling-based relay protocol.

As a tree-based multicast protocol, `EV-CAST` constructs a two-hop multicast tree based on our proposed interference-aware link characterization. As in [4], we focus on two-hop network where the destination nodes are one-hop or two-hop away from the source node, since video streaming across more than two hops is inappropriate due to large delay and

2155-6814/19/$31.00 ©2019 IEEE
DOI 10.1109/MASS.2019.00045

bandwidth limitation. All the links on the tree are characterized by desirable TPs selected based on the cause of packet losses, i.e., channel error or interference. A centralized algorithm then determines the relay nodes and their TPs by considering various factors including battery status, interference, and spatial reuse. Selected relay nodes are scheduled to transmit with consideration of spatial reusability.

In order to study the effectiveness and efficiency of our proposed scheme, we implement EV-CAST in Linux device driver and evaluate the performance of EV-CAST in imec w-ilab.t testbed [6]. Our experiment results demonstrate that EV-CAST outperforms the state-of-the-art schemes. To be specific, EV-CAST enables 2× more nodes to achieve a target video packet loss ratio with 0.59× shorter airtime and 1.05× longer lifetime than ViMOR [4].

Our major contributions are summarized as follows.

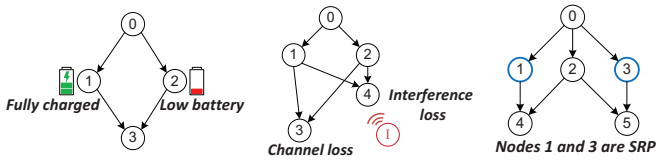
- We propose EV-CAST, an interference- and energy-aware video multicast system exploiting collaborative relaying. To our best knowledge, it is the first video multicast scheme that selects relays with consideration of battery status, interference, and spatial reuse.
- We propose an interference-aware link characterization for multi-hop networks, which enables the differentiated selection of TPs depending on the cause of packet losses.
- We propose an algorithm that jointly determines relay nodes and their TPs. Our proposed utility function integrates various factors including the number of neighbors, airtime, battery status, and spatial reuse. Based on that, our relay selection procedure enables judicious selection.
- We present a prototype implementation of EV-CAST and evaluate the performance in imec w-ilab.t testbed. Our experiment results demonstrate that EV-CAST significantly outperforms the state-of-the-art video multicast schemes.

The rest of this paper is organized as follows: In Section II, we discuss the factors to consider in relay selection. In Sections III and IV, we present the detailed design of EV-CAST and comparatively evaluate the performance, respectively. We discuss the related work in Section V, and finally conclude in Section VI.

II. New Factors for Relay Selection

In cooperative relaying, determining relay nodes and their TPs is the major problem. There have been various studies that address the factors to be considered for relay selection,[1] e.g., packet loss ratio (PLR),[2] expected transmission time (ETT), the number of neighbors, etc. In addition to them, we take three key factors into account: 1) battery status, 2) interference, and 3) spatial reusability.

Fig. 1 depicts the newly addressed factors. Each circle indicates a node while index zero is assigned to the source node. Fig. 1(a) shows an example where a source node has two



(a) (b) (c)

Fig. 1. New factors for relay selection: (a) battery status, (b) interference, and (c) spatial reusability.

neighbor nodes, i.e., relay candidates, having different battery status. Since the relaying functionality involves additional energy consumption, remaining battery is directly related to the duration of relaying. Besides the relay duration, battery status is related to the willingness to relay, i.e., nodes with higher battery level will be more willing to relay. We also consider the charging status, i.e., whether the battery is being charged or not, since portable battery chargers has become popular. It may be reasonable to select a node whose battery is being charged even though its current battery level is low.

Fig. 1(b) presents an example when there exists an interfering node (node I). Two two-hop nodes, i.e., nodes 3 and 4, experience packet losses due to different causes, i.e., low channel quality and interference. In [5], it is reported that the optimal TPs depend on the cause of packet losses, and diagnosing the cause of packet losses enables an interference-resilient video multicast. Therefore, we take interference into account for relay selection.

Fig. 1(c) presents an example when the spatial reuse, i.e, concurrent transmission of two or more nodes, is available. In order to serve nodes 4 and 5, if nodes 1 and 3 are selected as relays, which are not neighbor with each other and do not share any common neighbor nodes except for the source node, concurrent transmission is possible, thus reducing the airtime resource usage. In this work, we consider such a pair of nodes that can transmit simultaneously, which is called SRP (spatially reusable pair), for relay selection.

III. Design of EV-CAST

The design of EV-CAST follows the philosophy of tree-based schemes as in [4]. Unlike belt-based schemes such as MORE [2], where every node can operate as a relay, tree-based schemes allow only selected nodes to relay packets while receiver nodes can receive packets from any relay nodes thanks to the broadcast nature of wireless channel. In [4], it is shown that tree-based schemes are more suitable for real-time video streaming than belt-based schemes.

A. Network Model and Objective

We mainly consider an ad hoc mode WLAN that consists of a single source connected to a content server via wireline and multiple wireless destinations.[3] Although it might not achieve maximum coverage, we can also apply EV-CAST

[1]For brevity, relay selection refers to determining relay nodes and TPs.

[2]We distinguish MAC-layer PLR (MPLR), i.e., PLR before NC decoding, from application-layer PLR (APLR), i.e., PLR after NC decoding. For brevity, packet loss and PLR indicate MAC-layer packet loss and MPLR, respectively, unless specified otherwise.

[3]Although ad hoc network with smartphones is not widely utilized yet, smartphone ad hoc network (SPAN) has attracted interest from research community and industry practitioners, where geocommunity-based video multicast is one of the promising applications [7].

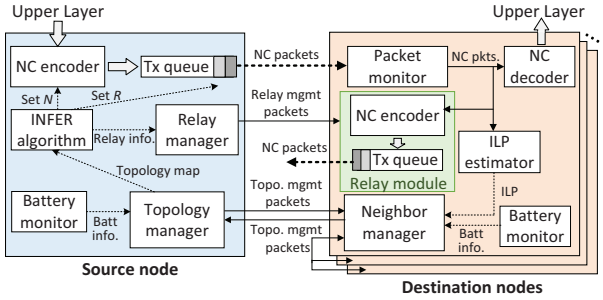


Fig. 2. Overall architecture of EV-CAST.

framework to the infrastructure mode as in [8, 9]. In addition, by connecting the content server to multiple source nodes in different channels, we can extend the coverage. We assume low mobility of the nodes, e.g., the majority of users are seated while watching a video. We consider real-time transport protocol/user datagram protocol (RTP/UDP)-based video streaming and MPEG-2 video coding scheme [10], which is a widely-employed protocol for video multicast.

Our primary objective is maximizing node satisfaction ratio (NSR), which is defined as the fraction of nodes watching videos with target APLR. We also aim to maximize video multicast service time while achieving NSR higher than a target value. Here, we set the target values for APLR and NSR to 1% [11] and 95% [5, 12], respectively.[4]

B. Overview

Fig. 2 depicts the overall architecture of EV-CAST. For every batch of K original packets, the source node generates N encoded packets by *NC encoder* and transmits them using PHY rate R determined by *interference and energy-aware relay selection (INFER) algorithm* (Section III-D). INFER algorithm also selects relay nodes and their TPs, and the results are input to *Relay manager*, which is in charge of sending relay management packets such as *RelaySetup* and *RelayPolling* packets to enable relay transmissions (Section III-E). INFER algorithm relies on a topology map updated by *Topology manager*, which exchanges topology management packets with the destination nodes (Section III-C). In addition, *Battery monitor* periodically updates the battery status.

Upon receiving NC packets, all destination nodes input them into *Packet monitor*, which extracts the input values for *interference-aware link state parameter (ILP) estimator* (Section III-C). The determined ILP, which represents the link quality, is recorded at *Neighbor manager*, which transmits topology management packets containing ILPs for all neighboring nodes to the source node or other destination nodes. Meanwhile, through *NC decoder*, each destination node recovers the original K packets upon receiving K *innovative packets*, i.e., packets with linearly independent encoding coefficients. Relay nodes selected by the source node re-encode the packets through *NC encoder*, and transmit them upon receiving the *RelayPolling* packet from the source node.[5]

The rest of this section provides the detailed description of three core operations in EV-CAST: (i) topology management, (ii) INFER algorithm, and (iii) polling-based relay protocol.

C. Topology Management

As in [4], we employ a centralized relay selection that a source node determines relay nodes. Since the relay nodes are determined based on a topology map, it generally selects more efficient relays than distributed relay selection schemes based on only local information. Although the centralized relay selection scheme has a drawback of large computational complexity, it is affordable for two-hop network.

1) Construction of global topology map: In order to construct a global topology map, the source node periodically sends short multicast packets, called *probing packets*. Nodes receiving the probing packets from the source node, called one-hop nodes, also send probing packets to discover two-hop nodes. We define two-hop nodes as nodes receiving the probing packets only from the one-hop node(s). In order to inform the local link state information to the source node, each destination node has a parent node. The source node is the parent node of all one-hop nodes, while each two-hop node sets the parent node to the one-hop node having the highest received signal strength indicator (RSSI). Each destination node sends feedback packets to its parent node by unicast, conveying its battery status and link state information for neighboring one-hop nodes or source node. Upon reception of feedback packets from two-hop nodes, one-hop node relays them to the source node.

2) Interference-aware link characterization: In most multicast schemes [3, 4, 9, 13], a link is characterized by PLR, but the PLR-based link characterization is unable to differentiate the cause of packet losses, i.e., whether the losses are due to channel error or interference. Instead of reporting just PLR, in InFRA [5], each receiver diagnoses the cause of losses among channel error, strong interference, and weak interference, and requests favorable TPs based on the differentiated loss statistics. Inspired by InFRA, we propose an interference-aware link characterization for multi-hop network.

Based on [5], each EV-CAST node determines two favorable pairs of PHY rate R and generation size N for neighboring nodes. PHY rate is normally determined by RSSI, but in some cases when there exist weak interference signals, using a lower PHY rate to induce 'capture effect' [14] is more desirable. To this end, each node determines a pair of channel quality-oriented PHY rate R_{ch} and generation size N_{ch} and a pair of capture-inducing PHY rate R_{cap} and generation size N_{cap}. In other words, a link from node i to node j is characterized by ILP $\mathbf{q}^{(i,j)} = (R_{ch}^{(i,j)}, N_{ch}^{(i,j)}, R_{cap}^{(i,j)}, N_{cap}^{(i,j)})$, which implies that node j can decode batches successfully if

[4]Based on our empirical results, we observe that 1% of APLR achieves peak signal-to-noise ratio (PSNR), a well-known video quality metric, over 37 dB, which typically indicates excellent video quality.

[5]For better practicality, *NC decoder* can be implemented in the application layer, which allows destination nodes without device driver modification to decode packets.

node i sends $N_{ch}^{(i,j)}$ encoded packets at $R_{ch}^{(i,j)}$ or sends $N_{cap}^{(i,j)}$ encoded packets at $R_{cap}^{(i,j)}$. Note that R_{cap} is lower than R_{ch} since a PHY rate more robust than R_{ch} is needed to receive packets even when there is interference. Accordingly, N_{cap} is smaller than N_{ch} because fewer parity packets are required when a more robust PHY rate is used.

In order to construct a topology map, each node determines ILPs for links from not only current senders, i.e., source node and relay nodes, but also possible senders, i.e., other neighboring one-hop nodes. Therefore, we propose separate link characterization methods for *sender nodes* and *non-sender nodes*, respectively. For a sender node, ILP is determined based on loss statistics measured from NC packets sent by the sender node. For a non-sender node, since loss statistics are not available, ILP is determined using RSSI of its probing packets and the loss statistics of neighboring sender nodes.

Algorithm 1 determines ILP for a neighboring sender node. For sender node i, node j keeps track of the current PHY rate ($R_{cur}^{(i)}$), generation size ($N_{cur}^{(i)}$), and the average RSSI per batch ($\overline{\gamma}^{(i)}$). As in the loss differentiation scheme in [5], for every batch, node j classifies total $l_t^{(i)}$ packet losses out of $N_{cur}^{(i)}$ packets into $l_c^{(i)}$ channel losses and $l_i^{(i)}$ interference losses, which consists of $l_s^{(i)}$ strong interference losses and $l_w^{(i)}$ weak interference losses by using cyclic redundancy check (CRC) error notification as well as RSSI. Then, node j updates loss parameters, i.e., ($l_{t,max}^{(i)}$, $l_{c,max}^{(i)}$, $l_{i,max}^{(i)}$, $l_{s,max}^{(i)}$), which are maximum numbers of total losses, channel losses, interference losses, and strong interference losses in a feedback period, respectively. Additionally, when decoding of a batch fails, the failed PHY rate ($R_{fail}^{(i)}$) is recorded in order to avoid the frequent trial of PHY rate increase. PHY rate increase to $R_{fail}^{(i)}$ is tried only when the success counter ($c_s^{(i)}$), which is raised by one whenever decoding is successful, reaches the success window ($w_s^{(i)}$), which is doubled whenever decoding fails. From now on, for brevity, we present the aforementioned parameters without superscripts representing nodes.

Network links are characterized by considering the cause of packet losses. We aim to find R_{ch} as the maximum PHY rate ensuring channel loss ratio ($l_{c,max}/N_{cur}$) to be smaller than a target value (ρ). When there is no channel loss, it is checked if PHY rate can be increased. If the condition of PHY rate increase is satisfied (line 3), R_{ch} is set to the next higher PHY rate than R_{cur} (R_{cur}^{+}). When the channel loss ratio is smaller than ρ, R_{ch} is set to R_{cur}. Otherwise, R_{ch} is determined by GETRATE function that finds the maximum R whose RSSI threshold ($\delta(R)$) is less than or equal to the input RSSI. We determine the RSSI thresholds as in InFRA [5]. N_{ch} is determined such that channel and interference losses can be recovered. If R_{ch} is equal to R_{cur}, we use the maximum total losses, $l_{t,max}$. Otherwise, we determine N_{ch} by assuming the maximum interference losses, $l_{i,max}$, and target channel losses of ρN_{cur}. We add ϵ (e.g., $\epsilon = 1$) to handle additional unexpected losses (lines 16−19).

Different from [5], where R_{cap} is determined by measuring interference signal strength, we simply set R_{cap} to the PHY rate Δ-step lower than R_{ch}. Since we consider multi-hop network where multiple senders exist inherently, it is difficult to separate interference from target senders, and hence, we employ a capture-inducing rate decrease step, Δ, which is empirically set to three in this work. Finally, N_{cap} is determined such that the strong interference losses can be recovered, assuming that R_{cap} is robust enough not to cause channel losses or weak interference losses (line 22).

Algorithm 1 Determination of ILP for a sender node

Input: $\overline{\gamma}, l_{t,max}, l_{c,max}, l_{i,max}, l_{s,max}, R_{cur}, N_{cur}, R_{fail}, c_s, w_s$
Output: $R_{ch}, N_{ch}, R_{cap}, N_{cap}$
1: **if** $l_{c,max} == 0$ **then** ▷ Determining R_{ch}
2: **if** GETRATE($\overline{\gamma}$) $> R_{cur}$ **then**
3: **if** $\left(R_{cur}^{+} < R_{fail}\right) \,||\, \left(R_{cur}^{+} == R_{fail} \&\& c_s == w_s\right)$ **then**
4: $R_{ch} \leftarrow R_{cur}^{+}$
5: **else**
6: $R_{ch} \leftarrow R_{cur}$
7: **end if**
8: **else**
9: $R_{ch} \leftarrow R_{cur}$
10: **end if**
11: **else if** $l_{c,max}/N_{cur} < \rho$ **then**
12: $R_{ch} \leftarrow R_{cur}$
13: **else**
14: $R_{ch} \leftarrow$ GETRATE($\overline{\gamma}$)
15: **end if**
16: **if** $R_{ch} \neq R_{cur}$ **then** ▷ Determining N_{ch}
17: $N_{ch} \leftarrow \left\lceil K \frac{N_{cur}}{(1-\rho)N_{cur} - l_{i,max}} \right\rceil + \epsilon$
18: **else**
19: $N_{ch} \leftarrow \left\lceil K \frac{N_{cur}}{N_{cur} - l_{t,max}} \right\rceil + \epsilon$
20: **end if**
21: $R_{cap} \leftarrow R_{ch} - \Delta$ ▷ Determining R_{cap}
22: $N_{cap} \leftarrow \left\lceil K \frac{N_{cur}}{N_{cur} - l_{s,max}} \right\rceil + \epsilon$ ▷ Determining N_{cap}

23: **function** GETRATE(γ)
24: Find out rate R s. t. $\delta(R) \leq \gamma < \delta(R^{+})$
25: **return** R
26: **end function**

Algorithm 2 Determination of ILP for a non-sender node

Input: $\overline{\gamma}, R_{cur}^{net}, \eta_{i,max}^{net}, \eta_{s,max}^{net}$
Output: $R_{ch}, N_{ch}, R_{cap}, N_{cap}$
1: **if** GETRATE($\overline{\gamma}$) $> R_{cur}^{net}$ **then** ▷ Determining R_{ch}
2: $R_{ch} \leftarrow R_{cur}^{net^{+}}$
3: **else**
4: $R_{ch} \leftarrow$ GETRATE($\overline{\gamma}$)
5: **end if**
6: $N_{ch} \leftarrow \left\lceil K \frac{1}{1-\rho-\eta_{i,max}^{net}} \right\rceil + \epsilon$ ▷ Determining N_{ch}
7: $R_{cap} \leftarrow R_{ch} - \Delta$ ▷ Determining R_{cap}
8: $N_{cap} \leftarrow \left\lceil K \frac{1}{1-\eta_{s,max}^{net}} \right\rceil + \epsilon$ ▷ Determining N_{cap}

Algorithm 2 presents the procedure of determining ILP for a neighboring non-sender node. Besides the average RSSI of the probing packets from the non-sender node ($\overline{\gamma}$), three parameters obtained from sender nodes are utilized. For each neighboring sender node, a node (node j) calculates interference loss ratio ($\eta_{i,max} = l_{i,max}/N_{cur}$) and strong interference loss ratio ($\eta_{s,max} = l_{s,max}/N_{cur}$), where $l_{i,max}$, $l_{s,max}$, and N_{cur} are recorded for Algorithm 1. After that, node j extracts R_{cur}^{net}, $\eta_{i,max}^{net}$, and $\eta_{s,max}^{net}$, where R_{cur}^{net} is the maximum value of R_{cur}'s of the current neighboring sender nodes, and $\eta_{i,max}^{net}$ and $\eta_{s,max}^{net}$ are maximum values of $\eta_{i,max}$'s and $\eta_{s,max}$'s, respectively. For non-sender nodes, we conservatively assume interference losses to be the maximum among those of neighboring sender nodes.

Then, R_{ch} is determined by GETRATE function. We limit R_{ch} to $R_{cur}^{net^+}$ in order to increase the PHY rate gradually. Other parameters such as N_{ch}, R_{cap}, and N_{cap} are determined similarly to as in Algorithm 1. We avoid unnecessarily frequent changes of relay nodes with such a conservative design.

3) Feedback timing: Each destination node regularly sends feedback packets in a long-term period. Furthermore, we adapt the feedback period of the regular feedback in order to reduce feedback overhead. The regular feedback period depends on node's priority: higher priority nodes continuously use the minimum feedback period, λ_{min},[6] while lower priority nodes double the feedback period whenever it sends the regular feedback two times until it reaches the maximum period, λ_{max}. Higher priority is given to relays and target nodes, which will be stated in the next section.

We additionally employ an event-driven feedback mechanism to handle the situation that fails to achieve the target APLR (1%). If a node fails to decode two batches before receiving 100 batches, it immediately sends a feedback. In this case, the node resets the regular feedback period to λ_{min}.

D. INFER Algorithm

1) Estimation of number of innovative packets: In relay selection, the source node chooses a relay node's TPs out of ILPs from the relay node to its neighboring nodes. To this end, the source node estimates how many innovative packets the neighboring nodes will receive. For example, if node i's TPs are chosen as $\left(R_{ch}^{(i,j)}, N_{ch}^{(i,j)}\right)$, node j can receive K innovative packets, and hence, can decode batches successfully. At the same time, other neighbor nodes of node i can also receive some innovative packets thanks to the broadcast nature, which should be considered in relay selection.

For this purpose, we now define an estimation function, $H(r, n, \mathbf{q})$, which estimates the number of innovative packets that a destination node will receive from a sender node with ILP $\mathbf{q}$ when the sender node sends n packets at PHY rate r. The estimation function is defined as

$$H(r,n,\mathbf{q}) = \begin{cases} \min\left(\left\lfloor n \cdot \frac{K}{N_{cap}} \right\rfloor, K\right), & \text{if } r \le R_{cap}, \\ \min\left(\left\lfloor n \cdot \frac{K}{N_{ch}} \right\rfloor, K\right), & \text{if } R_{cap} < r \le R_{ch}, \\ 0, & \text{if } r > R_{ch}. \end{cases} \tag{1}$$

The estimation function is interpreted as, for example, if r is lower than or equal to R_{cap}, only N_{cap} is required to provide K innovative packets, but if the number of transmitted packets is smaller than N_{cap}, then innovative packets will be received with the success ratio of K/N_{cap}.

2) Utility function: INFER algorithm iteratively finds the best set of sender nodes and their TPs that maximize a *utility function* through multiple rounds. We denote a triplet of sender node index, PHY rate, and generation size, by $Q = (I, R, N)$, which is called *IRN*, and a set π of IRNs, which is called *IRN assignment unit* (IU), is determined in every round. In order to serve a two-hop node, the source node needs to determine the best one-hop node to relay and its TPs, and also needs to determine its own TPs to serve the selected one-hop node. Therefore, source node's TPs are dependent on the IRN of the one-hop node. Moreover, if the one-hop node is spatially reusable with another one-hop node, it should be also considered.

For this reason, we choose at most three IRNs in a round, which is denoted by $\pi = \{Q_{src}, Q_{pr}, Q_{sr}\}$, where Q_{src}, Q_{pr}, and Q_{sr} are IRNs for the source node, the target one-hop node (also called primary relay), and the selected spatially reusable node (also called secondary relay), respectively. There are five types of IU: (i) IRN of the source node only ($Q_{pr} = Q_{sr} = \varnothing$), (ii) IRN of a primary relay only ($Q_{src} = Q_{sr} = \varnothing$), (iii) IRNs of the source and a primary relay ($Q_{sr} = \varnothing$), (iv) IRNs of a primary relay and a secondary relay ($Q_{src} = \varnothing$), (v) IRNs of the source, a primary relay, and a secondary relay.

For an IU π, a utility function $U(\pi)$ is defined as the ratio of benefit function $B(\pi)$ to cost function $C(\pi)$:

$$U(\pi) = B(\pi)/C(\pi), \tag{2}$$

where $B(\pi)$ is the number of nodes *newly served* by π and $C(\pi)$ is a weighted sum of the airtime required by π.

A *served* node is defined as a node whose expected number of the received innovative packets is equal to the batch size (K). Through multiple rounds, the estimated number of innovative packets increases, and when the number eventually becomes K, this node is referred to be *newly served*.

We define an energy factor to take the battery status into account for the cost function. The energy factor of node i, denoted by E_i, is defined as

$$E_i = \omega^{c_i} e_i, \tag{3}$$

where ω (≥ 1) is a constant weight associated with the charging status c, which is 1 if node i is being charged or 0 otherwise, and e_i is the remaining battery of node i in percentage. $T(r, n)$ is the estimated transmission time when transmitting n packets at PHY rate r:

$$T(r,n) = n \cdot \{(L_m + L_h)/r + \tau_O\} + \tau_{poll}, \tag{4}$$

where (i) L_m is the nominal multicast packet length (1,328 B with MPEG-2 TS format), (ii) L_h is the total length of headers including RTP/UDP/IP/LLC/MAC headers as well as the EV-CAST data packet header indicating K, N, sequence number, and the encoding coefficients, (iii) τ_O is the time duration for other overheads including preamble, PHY header, and backoff, and (iv) τ_{poll} is the time duration for relay polling, which will be addressed later.

Considering the airtime, energy factor, and spatial reusability, we develop the cost function as follows.

$$C(\pi) = F(Q_{src}) + \alpha(Q_{pr}, Q_{sr}) \{F(Q_{pr}) + F(Q_{sr})\}, \tag{5}$$

where

$$F(Q) = \begin{cases} T(R_Q, N_Q)/E_{I_Q}, & \text{if } Q \neq \varnothing, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

[6]We set λ_{min} to 100 batches in this work, since at least 100 batches are needed to make sure whether the target APLR (1%) is satisfied or not.

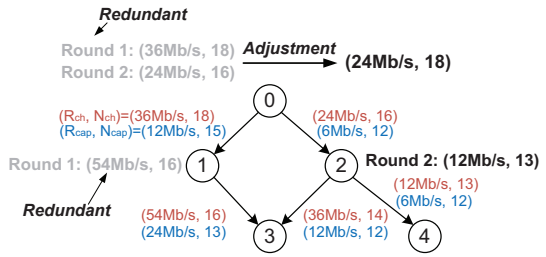


Fig. 3. An example of IRN adjustment.

and $\alpha(Q_{pr}, Q_{sr})$, a discount factor for the spatial reuse, is given by

$$\alpha(Q_{pr}, Q_{sr}) = \frac{\max\left(T(R_{Q_{pr}}, N_{Q_{pr}}), T(R_{Q_{sr}}, N_{Q_{sr}})\right)}{T(R_{Q_{pr}}, N_{Q_{pr}}) + T(R_{Q_{sr}}, N_{Q_{sr}})}. \tag{7}$$

Note that E_{I_Q}, R_Q, and N_Q denote the energy factor of node I_Q, PHY rate, and generation size associated with IRN Q, respectively. In order to reduce airtime, selecting type-4 or type-5 IUs that include SRP relay nodes is encouraged by adopting the discount factor. As the difference of the transmission time between SRP relay nodes decreases, relative time for simultaneous transmission to the total time increases, and hence, the discount factor is designed to be proportional to the transmission time difference.

The utility function captures the considerations in Section II. The utility function favors IUs that increase the number of served nodes with shorter airtime. Also, it favors nodes charging their batteries and having higher remaining battery level. Furthermore, it favors spatially reusable relays with higher spatial reuse gain.

By searching the best IU to serve each possible destination node, INFER algorithm finds the optimal IU in each round, where the destination node maximizing the utility function is called *target node*. As rounds are repeated, new IUs are selected, thus increasing both the number of served nodes and required airtime. Accordingly, the algorithm finishes when all nodes are served or no more airtime is available.

3) IRN adjustment: Such a greedy selection that determines new IRNs maximizing the number of newly served nodes per unit cost in every round might make pre-determined IRNs redundant, thus necessitating adjustment of IRNs.

Fig. 3 depicts an illustrative example of redundant IRNs and necessity of the IRN adjustment. We assume that all nodes' energy factors are 100. In round 1, type-3 IU, $\pi_\mathbf{1} = \{(0, 36\text{ Mb/s}, 18), (1, 54\text{ Mb/s}, 16)\}$ is selected to serve node 3 (target node). Note that $B(\pi_\mathbf{1})$ is 2 (nodes 1 and 3), and $C(\pi_\mathbf{1})$ is $F(0, 36\text{ Mb/s}, 18) + F(1, 54\text{ Mb/s}, 16) = 91 + 64 = 155$, thus resulting in $U(\pi_\mathbf{1})$ of 0.0129. After that, $\pi_\mathbf{2} = \{(0, 24\text{ Mb/s}, 16), (2, 12\text{ Mb/s}, 13)\}$ is selected to serve node 4, where $B(\pi_\mathbf{2}) = 2$ (nodes 2 and 4), $C(\pi_\mathbf{2}) = 253$, and $U(\pi_\mathbf{2}) = 0.0079$. Although $\pi_\mathbf{1}$ is selected due to its higher utility, $\pi_\mathbf{2}$ makes the IRNs assigned by $\pi_\mathbf{1}$ redundant. Therefore, an adjustment process is needed to remove the firstly assigned IRNs and increase $N_{Q_{src}}$ of $\pi_\mathbf{2}$ by 2 as shown in Fig. 3. In this case, the adjustment is conducted at the same node, so it is called *intra-node adjustment*. Whereas, Q_{pr} of $\pi_\mathbf{1}$ is replaced by IRN for node 2, and such adjustment between different nodes is called *inter-node adjustment*.

Motivated by this, we propose an IRN adjustment algorithm, which consists of two phases: source node's intra-node adjustment and relay nodes' inter-node adjustment. Firstly, we separate IRNs of the source node and find the set of nodes served by the source node, $\mathbf{S_0}$. For source's IRN adjustment, we choose the minimum PHY rate among the assigned PHY rates, and find N ensuring that the estimated number of innovative packets is K for all nodes in $\mathbf{S_0}$. For relays' IRN adjustment, we adjust IRNs in the increasing order of PHY rate, since packets with a lower PHY rate can be received by more nodes. After determining the set of nodes served by the selected relay node, we adjust N to ensure all nodes in the set can be served.

E. Assignment, Polling, and Re-selection of Relays

1) Relay assignment: By INFER algorithm, the source node determines a set of relay nodes along with their TPs and target nodes. Then, the source node sends a *RelaySetup* packet to each selected relay node, which includes the TPs and a list of the target nodes. Upon receiving a *RelaySetup*, a relay node sets up the TPs and sends *TargetNotification* packets to their target nodes. As in Section III-C, the relays and target nodes send regular feedback with the fixed minimum interval.

2) Relay polling: We employ a polling-based relay protocol in order to (i) avoid collisions between the source and relay nodes or between relay nodes and (ii) fully utilize the spatial reuse gain. Note that the spatial reuse gain via SRP can be fully utilized only when there are no other transmitting nodes.

The source node sends *RelayPolling* packets to allow a relay or SRP relay nodes to transmit. For SRP relay nodes, the source sends two *RelayPolling* packets subsequently to each relay node. A relay node prepares re-encoded video packets, and then transmits them as soon as it receives *RelayPolling* packet. After relaying, the relay node sends *RelayEnd* packet to the source node, and then the source node sends *RelayPolling* packets to another relay or SRP relay nodes.

3) Relay re-selection: The source node normally runs INFER algorithm every 100 batches. Additionally, there are two cases when the source node immediately re-selects relay nodes: (i) when it receives event-driven feedback from more than 5% of total receivers or (ii) when a relay node fails to decode a batch twice. For relay re-selection, the source node considers only the nodes from which receive the feedback during the re-selection period to reduce computational complexity. With the regular and event-driven re-selections, EV-CAST enables relay selection adaptive to topology variation.

IV. Performance Evaluation

We comparatively evaluate the performance of EV-CAST under various scenarios. We have implemented EV-CAST by modifying the latest ath9k device driver, backport 4.2.6-1 [15]. In our implementation, K is set to 10 as in [5], and the interval of one-hop and two-hop probing packets is set to 2 s. ρ and ω are set to 0.1 and 2, respectively.

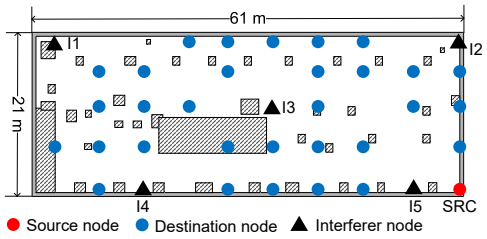


Fig. 4. Floor plan of w-iLab.t testbed.

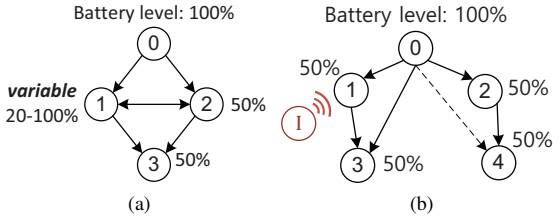


Fig. 5. Scenarios for micro-benchmark: (a) scenario 1: impact of battery status and (b) scenario 2: impact of interference and spatial reuse.

A. Measurement Setup

We evaluate the performance of EV-CAST in *w-iLab.t* testbed [6], which is a wireless testbed with programmable WLAN, Bluetooth, and LTE nodes. We use WLAN nodes equipped with a signal attenuator of 20 dB to make two-hop networks. Each node is a Ubuntu 14.04 machine equipped with Qualcomm Atheros AR9280 chipset. We configure an ad-hoc network on a channel at 5 GHz band, which consists of a source node, 30 destination nodes, and five interferer nodes as shown in Fig. 4. The source node generates constant bit rate (CBR) traffic of 1 Mb/s using Iperf 2.0.5 with fixed packet length of 1,328 bytes, unless specified otherwise. Since the nodes in *w-iLab.t* testbed are always AC-powered, we employ the energy consumption model in [16] for energy consumption evaluation. WLAN energy consumption parameters are referred to [17], and other energy models, such as the full charge capacity and the charging rate, are referred to [18].

We measure five performance metrics: (i) *APLR*, (ii) *NSR*, i.e., fraction of nodes satisfying the target APLR, (iii) *fractional airtime*, i.e., fraction of airtime occupied by WLAN traffic, (iv) *fractional transmit time*, i.e., ratio of transmitting time to the total time, and (v) *network lifetime*, which is defined as time duration from when video multicast service starts until the first node failure occurs due to energy depletion.

We compare EV-CAST with the following schemes: (i) *Legacy 802.11*, legacy multicast scheme using the lowest PHY rate without NC, (ii) *InFRA*, an ad-hoc version of InFRA [5], (iii) *ViMOR (Original)*, the original version of ViMOR [4] using $K = 10$ for a fair comparison, and (iv) *ViMOR (Multi-rate)*, a multi-rate extension of ViMOR. We modify ViMOR to utilize multiple PHY rates, where the source determines PHY rates of relay nodes based on the shortest path in terms of ETT, and assigns N minimizing the average APLR as in [4].

B. Micro-benchmark

Through the measurement under specific scenarios, we verify the featured operation of EV-CAST and impact of the factors addressed in Section II.

1) Impact of battery status: In order to observe the impact of the battery status, we set up the topology as shown in Fig. 5(a). We vary the battery level of node 1, while setting source node's battery level to 100% and the other nodes' battery levels to 50%.

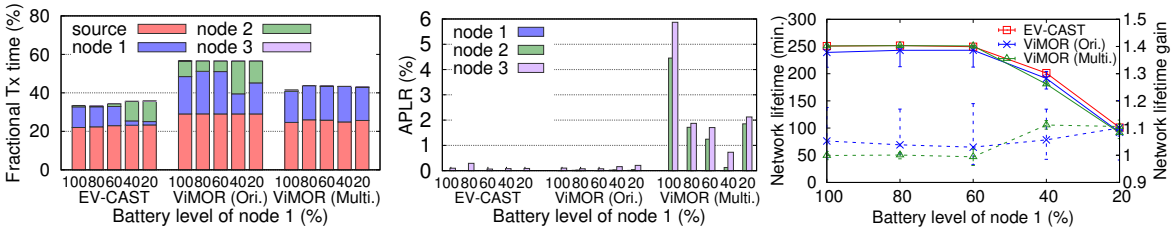
The measurement results are presented in Fig. 6. Fig. 6(a) presents the fractional transmit time of each node with respect to the battery level of node 1. We observe that EV-CAST changes the relay node from node 1 to node 2 when node 1's battery level is 40%, while ViMOR selects relay nodes regardless of node 1's battery level. The total height of the bars represents total airtime, and EV-CAST consumes lower airtime due to its judicious selection of TPs. Fig. 6(b) presents the APLR of each node. We observe that both EV-CAST and original ViMOR achieve the target APLR. For multi-rate ViMOR, incorrect channel quality estimation that relies only on the probing packets incurs higher APLR in some cases.

Fig. 6(c) presents the network lifetime, where each solid line represents the lifetime of each scheme, and each dashed line represents the lifetime gain of EV-CAST over each ViMOR scheme. When node 1's battery level is higher than 50%, the bottleneck node determining lifetime is node 2, and hence, the lifetime gain is marginal especially when the ViMOR schemes select node 1 as the relay. Note that lifetime gain due to airtime reduction is not significant because there are other energy consuming components such as CPU and display. On the other hand, when node 1's battery level is lower than 50%, the lifetime is determined by node 1, and hence, we observe a large gain of lifetime thanks to the energy-aware relay selection. EV-CAST achieves up to 10% and 11% longer lifetime than original and multi-rate ViMORs, respectively.

2) Impact of interference and spatial reuse: In order to observe the impact of interference and spatial reuse, we set up the topology as in Fig. 5(b), where nodes 1 and 2 are hidden to each other, i.e., they are an SRP. Distance between nodes 0 and 1 is shorter than that between nodes 0 and 3, and the channel quality from node 0 to node 4 is low such that PLR with 6 Mb/s is about 40%. We locate an interferer near node 1, and vary the source rate of the interferer.

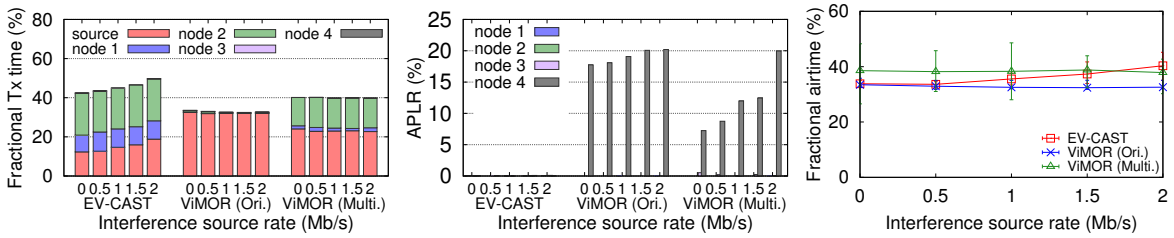
The measurement results are presented in Fig. 7. In Fig. 7(a), we observe that EV-CAST selects both nodes 1 and 2 as relays while multi-rate ViMOR selects only node 2 as a relay. With original ViMOR, only the source node transmits packets, since direct links from the source node have the shortest ETT for all nodes. We observe that the transmit time values of both nodes 0 and 1 increase as the interference source rate increases thanks to the interference-aware link characterization, while almost fixed transmit time distributions are observed in ViMOR schemes.

In Fig. 7(b), we observe that node 4 loses many packets with ViMOR. In original ViMOR, the packet losses are caused by the bad channel quality from the source node, while in multi-rate ViMOR, using higher PHY rate due to incorrect link quality estimation results in the packet losses. We observe



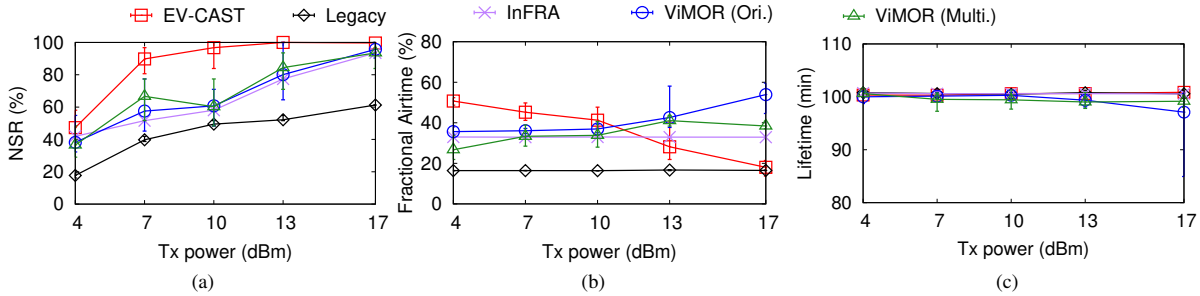
(a) (b) (c)

Fig. 6. Measurement results for scenario 1: (a) distribution of fractional tx time, (b) distribution of APLR, and (c) network lifetime.



(a) (b) (c)

Fig. 7. Measurement results for scenario 2: (a) distribution of fractional tx time, (b) distribution of APLR, and (c) fractional airtime.



(a) (b) (c)

Fig. 8. Measurement results with respect to the transmit power: (a) NSR, (b) fractional airtime, and (c) network lifetime.

that EV-CAST assigns longer transmit time to node 2, thus achieving near-zero APLR. Fig. 7(c) presents fractional airtime results. In contrast to Section IV-B1 where the total height is the same as the fractional airtime, node 1's transmission time does not increase the airtime thanks to spatial reuse. Accordingly, EV-CAST achieves shorter airtime than multi-rate ViMOR even though it transmits more packets, except when the inference source rate is 2 Mb/s. We observe that interference-aware TP selection of EV-CAST adaptively increases airtime depending on the interference source rate.

C. Macro-benchmark

We evaluate the performance with 30 destination nodes for various transmit power levels. The network shrinks as the transmit power decreases, and hence, we observe the impact of the network size by varying the transmit power. The battery level of each node is 50% or 100% with the equal probability of 0.5 and the charging probability is 0.5 as well.

Fig. 8(a) presents NSR results. We observe that NSR increases as the transmit power increases since more nodes can be served with higher transmit power. EV-CAST achieves the highest NSR for all the transmit power values. Some nodes become more than two hops away from the source node when the transmit power is 4 dBm, thus resulting in low NSR. We also observe that NSR of multi-rate ViMOR decreases due to collisions between relay nodes when the transmit power is 10 dBm. Fig. 8(b) presents the fractional airtime. Since longer airtime is required to serve nodes with lower transmit power, EV-CAST consumes more airtime as the transmit power decreases. On the other hand, the original ViMOR decreases the airtime as the transmit power decreases due to the following philosophy of ViMOR. ViMOR allocates the same N ($\geq K$) to all the relay nodes. However, if the required N is too large to be allocated to all the relay nodes under the available airtime, ViMOR does not allow any relay node to relay at all. Due to such an all-or-none mechanism, the source node does not select relay nodes when the transmit power is low. As a consequence, EV-CAST achieves higher NSR at the cost of the airtime when the transmit power is low, while achieving shorter airtime with the maximum NSR when the transmit power is high. Fig. 8(c) presents the network lifetime. We observe that EV-CAST achieves longer lifetime especially when the transmit power is 17 dBm thanks to its short airtime. Moreover, when the transmit power is low and the airtime is long, EV-CAST still achieves moderate lifetime due to its energy-aware relay selection.

With streaming a real video clip (1280×720 resolution, MPEG-4 codec, 1 Mb/s, 5 min), we measure the performance for five different interferers. We set the transmit power to

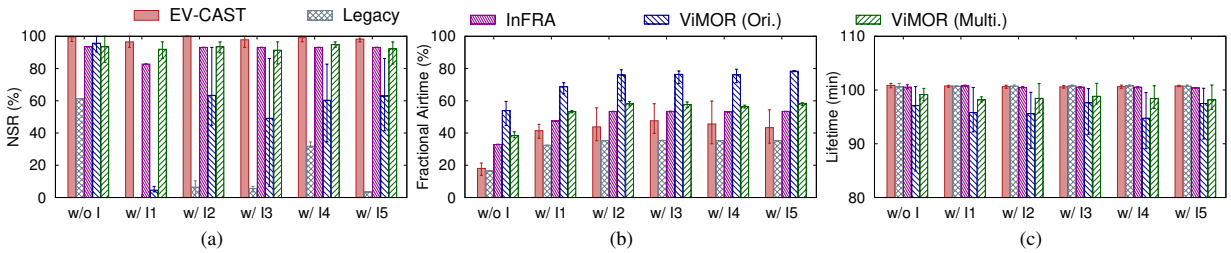


Fig. 9. Measurement results without and with five different interferer locations: (a) NSR, (b) fractional airtime, and (c) network lifetime.

17 dBm since all the schemes achieve the highest NSR with 17 dBm without interference. Each interferer broadcasts packets with source rate of 1 Mb/s using PHY rate of 6 Mb/s.

In Fig. 9(a), we observe that EV-CAST still achieves the target NSR while the other schemes fail to achieve it. Especially, legacy and original ViMOR schemes achieve very low NSR since they assign small N values, which are not enough to overcome the interference. In Figs. 9(b) and 9(c), we observe that EV-CAST achieves shorter airtime and longer lifetime than ViMOR schemes. With interference, on average, EV-CAST achieves 2× higher NSR, 0.59× shorter airtime, and 1.05× longer lifetime than original ViMOR. Compared with multi-rate ViMOR, EV-CAST achieves 0.78× shorter airtime and 1.02× longer liftime, while achieving the target NSR for all scenarios. With I5, EV-CAST achieves 1.11× higher NSR, 0.74× shorter airtime, 1.02× longer lifetime than multi-rate ViMOR.

V. Related Work

Collaborative relaying has been utilized in multicast routing schemes. MORE [2], one of the most well-known multicast routing schemes, firstly employs intra-flow NC to enhance reliability. Uflood [3] identifies major factors for relay selection, such as PLR and PHY rate, which are also addressed in EV-CAST. While aforementioned schemes are designed for applications requiring 100% reliability such as file transfer, ViMOR [4] firstly proposes a multicast routing scheme suited for real-time video streaming. Zhao *et al.* [19] suggest social-aware cooperative multicast scheme. On the other hand, there are relay-based video multicast schemes to enhance reliability of edge nodes [8, 9]. However, none of them considers interference, battery status, and spatial reuse in relay selection.

There are several efforts to enhance coverage by deploying multiple APs in academia as well as in industry. Choi *et al.* [13] propose multi-AP video multicast system that exploits a cooperative forward erasure correction approach. JurCast [20] proposes an algorithm determining user association, PHY rate, and video source rate. Cisco Connected Stadium deploys multiple APs to cover large sport stadiums [21]. However, they require additional AP deployment cost, and are less flexible than collaborative relaying schemes.

VI. Conclusion

We proposed EV-CAST, an interference and energy-aware video multicast system that exploits collaborative relays. On top of the proposed interference-aware link characterization, INFER algorithm determines relay nodes by taking into account various factors including battery status, interference, and spatial reusability. The polling-based transmission mechanism reduces collisions between relay nodes. Our prototype-based large-scale measurement results demonstrate that EV-CAST outperforms the state-of-the-art video multicast schemes. Specifically, with interference, EV-CAST achieves 2× higher NSR, 0.59× shorter airtime, and 1.05× longer lifetime than ViMOR. As future work, we plan to develop an incentive mechanism for selfish nodes.

Acknowledgements

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (No. 2018-0-00815, Development of wireless LAN platform with smart cloud based multi radio structure).

References

[1] T. Ho *et al.*, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
[2] S. Chachulski *et al.*, "Trading structure for randomness in wireless opportunistic routing," in *Proc. ACM SigComm*, Aug. 2007.
[3] J. Subramanian *et al.*, "Uflood: High-throughput flooding over wireless mesh networks," in *Proc. IEEE INFOCOM*, Mar. 2012.
[4] K. Choumas *et al.*, "Video-aware multicast opportunistic routing over 802.11 two-hop mesh networks," *IEEE Trans. Veh. Technol.*, vol. PP, no. 99, pp. 1–13, 2017.
[5] Y. Shin *et al.*, "Infra: Interference-aware PHY/FEC rate adaptation for video multicast over wlan," in *Proc. IEEE SECON*, Jun. 2017.
[6] imec w-iLab.t wireless testbed. [Online]. Available: http://doc.ilabt.imec.be/
[7] E. Soares *et al.*, "Experimentation with MANETs of smartphones," in *Proc. IEEE Wireless Days*, Mar. 2017.
[8] J. Xiong and R. R. Choudhury, "Peercast: Improving link layer multicast through cooperative relaying," in *Proc. IEEE INFOCOM*, Apr. 2011.
[9] O. Alay *et al.*, "Layered wireless video multicast using relays," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 8, pp. 1095–1109, 2010.
[10] *Generic coding of moving pictures and associated audio information: Systems*, ISO/IEC 13818-1 4rd edition, Jun. 2013.
[11] J. Prado *et al.*, "Application characteristics for HT usage scenarios," IEEE 802.11-03/364r0, 2003.
[12] V. Gupta *et al.*, "Experimental evaluation of large scale WiFi multicast rate control," in *Proc. IEEE INFOCOM*, Apr. 2016.
[13] M. Choi *et al.*, "Reliable video multicast over Wi-Fi networks with coordinated multiple APs," in *Proc. IEEE INFOCOM*, Apr. 2014.
[14] J. Lee *et al.*, "An experimental study on the capture effect in 802.11a networks," in *Proc. ACM WiNTECH*, Sep. 2007.
[15] Linux kernel backports website. [Online]. Available: http://drvbp1.linux-foundation.org/~mcgrof/rel-html/backports/
[16] W.-S. Lim and K. G. Shin, "EMS: Efficient multicast streaming scheme for multicasting within Wi-Fi hotspot," in *Proc. IEEE ICNP*, Nov. 2015.
[17] Data sheet for BCM4356. [Online]. Available: http://www.cypress.com/file/298796/download
[18] M. A. Hoque *et al.*, "Full charge capacity and charging diagnosis of smartphone batteries," *IEEE Trans. on Mobile Comput.*, vol. PP, no. 99, pp. 1–14, 2017.
[19] L. Zhao *et al.*, "Social-aware cooperative video distribution via svc streaming multicast," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.
[20] H. Wang *et al.*, "Jurcast: Joint user and rate allocation for video multicast over multiple APs," in *Proc. IEEE INFOCOM*, Apr. 2016.
[21] Cisco Connected Stadium. [Online]. Available: https://www.cisco.com/c/en/us/solutions/industries/sports-entertainment/connected-stadium.html