Peer-assisted Video-on-Demand in Multi-channel Switching WiFi-based Mobile Networks

Hayoung Yoon* and JongWon Kim†
* Gwangju Institute of Science and Technology (GIST), Gwangju, Korea
E-mail: hyyoon@nm.gist.ac.kr
† Gwangju Institute of Science and Technology (GIST), Gwangju, Korea
E-mail: jongwon@nm.gist.ac.kr

***Abstract*—Network convergence paradigm will substantially increase the pervasive use of WiFi-enabled smart mobile devices. Although various on-demand streaming services are already available over mobile WiFi-enabled devices, it remains a challenging problem due to WiFi's limited communication range, mobility and user population density issues. In this paper, we enhance our previous work on MOVi (Mobile Opportunistic Video-on-demand) by exploiting the use of multi-channel switching capability at mobile terminals. We reinvestigate the previous version of MOVi in this environment and propose an improved scheduling algorithm which incorporates collaborative congestion sensing and efficient channel allocation mechanisms. The scalability of the extended MOVi system is verified by extensive simulations. In terms of the number of supported user, an average of 25% improvement can be achieved. In addition, the proposed extension of MOVi provides more tolerance against increased volume of non-MOVi traffic.**

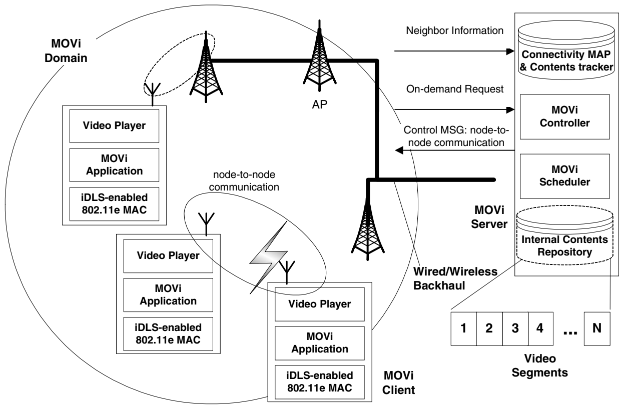


Fig. 1. Illustration of MOVi and its architectural components.

I. Introduction

Recent dramatic trends in network convergence lead mobile smart hand-held devices to be equipped with wireless communication interfaces for broadband access. It is especially noteworthy that WiFi (i.e., IEEE 802.11-based wireless local area network) is taking the leading role for the wireless communication due to its cheap integration cost and almost free-of-charge networking availability. This trend will be accelerated by upcoming industrial standard, WiFi Direct [1] (formerly known as WiFi peer-to-peer) that allows terminal-to-terminal WiFi connections while the infrastructure-based WiFi connections are sustained. Various WiFi-enabled applications are emerging by providing on-demand video streaming services to large numbers of mobile WiFi-devices. However, it is a challenging task to design a WiFi system providing such services due to its limited wireless communication range, user mobility, and variable user population density.

In our previous proposal [2], we questioned whether providing realtime streaming services is possible over Bittorrent-style content sharing protocols, where the centralized tracker helps mobile nodes to share network resources in content distribution. Specifically, we have proposed MOVi (Mobile Opportunistic Video-on-demand), a cooperative video-on-demand distribution and sharing system. MOVi considers the general problem of video-on-demand streaming service within the mobile peer-to-peer paradigm amongst WiFi-equipped devices.

In this paper, we reinvestigate our previous MOVi system in more realistic scenarios and propose its enhancement, namely MOVi2, using multi-channel switching capabilities on the WiFi device at mobile terminals. Specifically, we design and implement the extension of previous inter-BSS (basic subset) direct-link setup (iDLS) protocol to cater multi-channel switching capabilities. On top of miDLS, we build MOVi2 that incorporates collaborative congestion sensing and efficient channel allocation mechanisms. Hereinafter, we call MOVi1 as our previous proposal, MOVi2 as the proposed enhancement in this paper, and MOVi to point out common aspects for both MOVi1 and MOVi2. Through extensive computer simulations, we show that the improved MOVi scheduling algorithm produces up to 25% higher system capacity than the previous version.

This paper is organized as follows. In Section II, we review MOVi1. In Section III, we outline the challenges and design space of MOVi2. Also, we explain MOVi2 by especially focusing on scheduling enhancements. We evaluate our proposal using computer simulations in Section IV, where the scalability aspect is examined with up to 120 mobile nodes. We summarize the paper in Section V.

II. MOVi1 Overview

A. Architectural Components

MOVi [2] is designed with the principle of *"centralized state tracking and control with decentralized data distri-*

bution" in mind. Figure 1 shows the general architecture of MOVi. It comprises of two logical components: mobile client nodes, called as `MOVi Clients` (MCs), and a `MOVi Server` (MS). One MS and a set of associated MCs compose each `MOVi Domain`. In MOVi, we assume that all the contents are fragmented into multiple equal-sized segments (i.e., chunks in peer-to-peer networks) and initially available from content repositories such as local caches and external storages of MS. We note that each segment normally consists of several hundreds of packets. Upon MS receiving on-demand requests from MCs, a series of video segments, initially available from MS, is transferred to a number of MCs. All MCs cooperate to download all segments of an entire video content by relaying received segments to the neighbor MCs.

For video streaming, each MC downloads the request segments in order of playout sequences. Once the MC receives all segments that make up a temporal portion (e.g., frames) of video stream, these segments are handed to the media player for playout. If there are missing segments for *to-be-playout* video frames, the MC sends immediate on-demand requests to MS so that we can receive the missing parts in a timely manner.

B. Inter-BSS Direct Link Setup

Although a pair of two MCs (as a sender and a receiver) is being collocated within a direct communication range, in the infrastructure mode WLANs [4], all transmissions from nodes must be forwarded first to the associated AP and then transferred to the destination nodes. In this case, the cooperative video content segments relays from MCs to others may be inefficient. In fact, each relay of packets consumes wireless channel resources two times when compared with the downloads from MS. To avoid this, MOVi uses iDLS protocol [3] to set up direct peer-to-peer connectivity, which allows direct communications between MCs. By using iDLS, MCs can preserve the membership with an associated AP without the switching overhead (i.e., in the order of seconds due to switching from infrastructure to adhoc mode and vice versa [12]).

The iDLS protocol is motivated by the standard Direct Link Setup (DLS), defined in IEEE 802.11e Quality-of-Service (QoS) enhancements [8], to remove the dependency on the AP and facilitate direct communications across BSSs. By setting up both permanent and temporary Service Set IDentifiers (SSIDs), iDLS can construct direct link communications temporarily using a randomly generated unique SSID value and keeping the basic IEEE 802.11 protocol unchanged. Instead of defining Over-the-DS MAC control frame type and message exchanges for inter-AP communication, iDLS assumes that the management of direct communication is maintained inside the end applications. In our early works [2], we have designed two different types of end applications to manipulate iDLS sessions, which located at MC and MS, respectively. As depicted in Figure 1, the former is the `MOVi Application`, and the latter is the `MOVi Scheduler`.

C. Peer-assisted VoD Exploiting iDLS

The `MOVi Application` at each MC serves as a temporal cache to help content diffusion inside each `MOVi Domain`. It also acts as a channel state monitor by periodically observing link quality with its neighboring MCs and updating any noticeable changes in the link quality to MS. The periodic content request carries the noticeable changes in a form of piggybacking. In doing so, the control overhead can be minimized to maintain a connectivity map of link qualities between MCs (i.e., the relative location of MCs). From the periodic updates and content request messages, the `MOVi Controller` at MS can derive the connectivity map, evaluate the QoS measure, and track the status of content cache in all MCs within its domain. Therefore, MS has an overall view of its domain while each MC cares local information only.

The `MOVi Scheduler` at MS coordinates the segment-based scheduling by opportunistically triggering content segment transfers between pairs of MCs based on the connectivity map. The scheduled segment is delivered from the source MC to the destination MC following the iDLS setup (see more detailed setup procedures in our previous work [2]). We note that the source MC does not know which segment is transferred to the destination MC before it is being scheduled by the `MOVi Scheduler`. To complement MC-to-MC segment transfers, the `MOVi Scheduler` also executes the direct transfers of content segments from MS to isolated MCs mostly for non-available (from neighbors) content segments. In this case, MS works similar to a super-peer node [13].

D. Performance Gain of MOVi1 over Conventional VoD

In WiFi networks, the achievable throughput performance within an infrastructures dramatically decreases when the expected PHY-datarates between an AP and a set of nodes are decreasing [14]. In mobile scenarios, therefore, the conventional VoD (CVoD) that solely relies on the AP-to-mobile path in distributing contents exhibits the problem that produces inefficient wireless resource utilizations.

MOVi1 translates the contacts of mobile MCs sharing high link-qualities into the chance to deliver a segment of requested contents at high expected PHY-daterates. In doing so, MOVi1 increases the utilization of wireless resource and indeed raises the achievable throughput performance within in an infrastructure. In addition, MOVi1 minimizes an amount of playout discontinuity at each MC to increase QoS. MOVi1 estimates the remaining buffering time (RBT) that quantifies the time budget to play video smoothly without downloading of further segments. By scheduling segment downloads to MCs in increasing order of their RBT score, one having lower budget can download further segment faster. Our previous scheduling method is also tailored to trigger only one MC-to-MC segment transfer amongst a set of MCs within the communication range. This burst scheduling minimizes the duration of each segment transfer which makes each to be resilient against the link-failure due to mobilities.

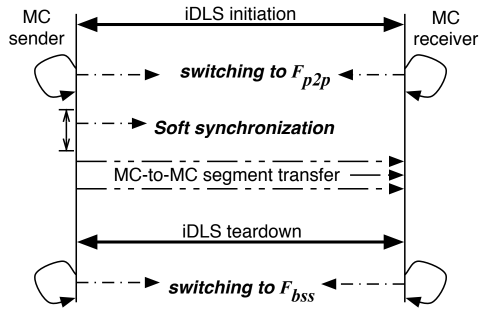


Fig. 2. Illustration of miDLS signaling, MC-to-MC segment transfer, and teardown sequences.

III. MOVi2: MOVi under Multi-Channel Environments

A. Multi-Channel inter-BSS Direct Link Setup

To facilitate multi-channel switching capability while a MC transfer a segment to another, we propose the revision of former iDLS [3], namely multi-channel inter-BSS Direct Link Setup (miDLS). The proposed miDLS temporally decoupling the traffic over direct communication links to another available channel which is orthogonal to the one set to the BSS. Figure 2 illustrates the sequence of a MC-to-MC segment transfer. At first, both MCs initialize the temporal direct link according to the iDLS specification. In this initiation stage, the target channel information F_{p2p} delivered from MS is shared to both MCs. Secondly, both MC changes operation channel channel to F_{p2p} upon the completion of the initiation. One of MCs transfers a segment wait for time T_{sync} before actual packet transmissions, where the marginal time budget T_{sync} is intended to make sure the completion of channel switching at another MC. After finishing the segment transfer, the direct link is teared down and two MCs return to the original channel F_{bss}. One can find more detailed procedure of initiation/teardown and synchronization mechanism to download packets buffered at AP while MC are detached from F_{bss} in [9].

WiFi chipset is known to provide very quick frequency switching speed about a few microseconds per switching at hardware level [15]. However, the application level overhead such as non-smart re/association, authentication and active spectrum scanning performed every time when user switch operative channel makes the switching latency increases up to several tens seconds. Throughput gain from frequency switching can not be obtained with that slow switching speed. To reduce frequency switching delay at the application level, we have implemented the *BSS cache* stores BSS registration information like ESSID and available data-rate sets. The *BSS cache* also stores not only the information of currently registered BSS but also the one of associable BSSs. When MC returns to the BSS channel, the information stored at the *BSS cache* is directly loaded to the device driver instead of having re-registration processes.

B. Challenges and Design Space

We have evaluated MOVi1 under the practical environments with various configurations [5], [10]. From these studies, it is revealed that MOVi1 does not fully exploit content distribution opportunities under the practical WiFi configurations, where multiple orthogonal channels are available. IEEE 802.11 [4] defines 3 and 6 orthogonal channels at 2GHz and 5GHz bands, respectively. By triggering the MC-to-MC segment transfers simultaneously over orthogonal channels, expected throughput performance of MOVi can be enhanced. In addition, non-MOVi traffic concurrently existing in the network is able to secure better end-to-end throughput performance.

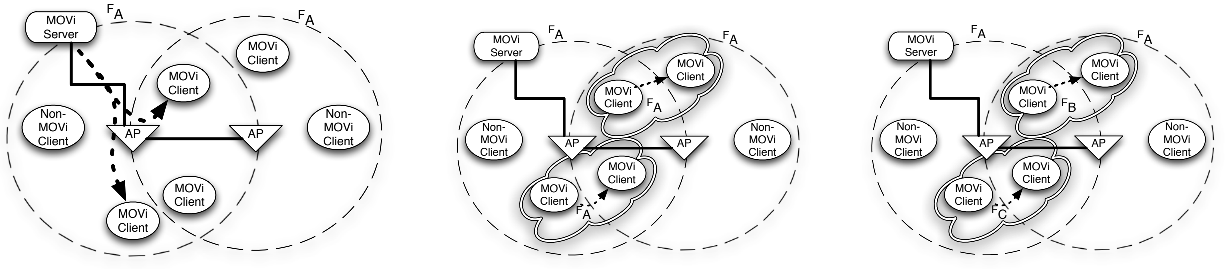
An example scenario is depicted in Figure 3(a), two segments are downloaded via AP-to-MC path and they then shared with neighboring MCs over MC-to-MC segment transfers. Without channel switching extension as depicted in Figure 3(b), two MC-to-MC segment transfers are scheduled and triggered one-by-one. Due to the contention, non-MOVi Clients' achievable bandwidth is also reduced. Figure 3(c) shows, where MOVi Server schedules two MC-to-MC segment transfer concurrently over orthogonal channels F_B and F_C. By respectively scheduling two at F_B and F_C, MOVi can achieve better throughput performance while it keeps preserving the bandwidth at F_A for non-MOVi Clients. Upon finishing MC-to-MC segment transfers, all MCs are returned to the BSS channel F_A (means physical channel switching).

Nevertheless the benefit from exploiting multi-channel switching capabilities, this extension of MOVi is not readily available. There are various technical challenges to extent MOVi1 under multi-channel environment.

1) Application-driven channel switching is hard and produce large latency. For off-the-shelf WiFi devices, changing operational channel is identical to associate with new BSS. Therefore, the latency involves active scanning, authentication, association processes [16]. Too much latency to switch channel reduce the gain in use of concurrent MC-to-MC segment transfers over multiple channels.
2) Amount of busyness on each channel must be measured to effectively guide channel allocations. Not all off-the-shelf WiFi devices are limited to measure the channel busy time [17] and direct measurement of busyness of all available channels wastes wireless resource utilization.
3) Efficient allocation of different channels on each possible MC-to-MC segment transfer should be fulfilled topology-aware, and it is identical to vertex coloring problem of arbitrary random graph. Some of existing algorithms to maximize the end-to-end throughput performance [18] can not be directly applicable due to the channel-switching capabilities at AP is out-of-control under the design space of MOVi.

C. Collaborative Congestion Sensing and P2P Channel Allocation

Each receiver MC r of the MC-to-MC segment transfer performed at channel c calculates the congestion rate $Cngs_r^c$



(a) (b) (c)

Fig. 3. MOVi media contents distribution scenario in (a) downloading, (b) MC-to-MC segment transfers in the previous MOVi, and (c) MC-to-MC segment transfers with multi-channel switching via miDLS.

upon it finishes to receive the segment. $Cngs_r^c$ is the ratio of actual time taken to receive the segment from sender MC s to the ideal air-time of all received packets and is computed as

$$Cngs_r^c = \frac{T_{last} - T_{first}}{\sum_{i=first}^{last} L/R_i^{s,r}}, \quad (1)$$

where T_{last}, T_{first}, $R_i^{s,r}$ and L are time to receive the last packet, time to receive the first packet, PHY-datarate of the i^{th} successfully received packet of the segment from s, and length of each packet, respectively. The computed $Cngs_r^c > 1$ is increased when c is getting congested. To notify the level of congestion experienced at each r, $Cngs_r^c$ is also piggybacked to the on-demand requests message and delivered to MS. Thus, `MOVi Scheduler` is able to construct the expected level of congestion for MC-to-MC segment transfer over all channels at each MC by differently allocating c.

The expected throughput performance of segment transfer for an arbitrary *contact* (s, r, c), $ETP_{(s,r,c)}$, is estimated as

$$ETP_{(s,r,c)} = \frac{L/(1/\overline{R^{s,r}} + T_{oh}^{(s,r,c)}/PPS)}{\frac{1}{|N_r|} \cdot \sum_{i=1}^{|N_r|} \overline{Cngs_i^c}}, \quad (2)$$

where $\overline{R^{s,r}}$, $T_{oh}^{(s,r,c)}$, PPS, N_r, and $\overline{Cngs_i^c}$ are the PHY-datarate expected by link quality between s to r [7], per-segment overhead to transfer segment, number of packets per segment, a set of neighbors of r, and expected congestion rate of MC i at channel c, respectively. We note that $\overline{Cngs_i^c}$ is computed by exponentially weighted moving average (EWMA) that highly weights the latest sample of $Cngs_i^c$ reports. $T_{oh}^{(s,r,c)}$ is induced to set up the direct-link and to switch operating channel of device and is computed as

$$T_{oh}^{(s,r,c)} = I(c \neq F_{bbs}^{s,r}) \cdot 2T_{swt} + I(s \notin A) \cdot T_{setup}, \quad (3)$$

where $I(x)$, T_{swt}, A, and T_{setup} are an indicator function returns 1 if the condition x is true otherwise 0, latency to change the channel, a set of APs within MOVi Domain, and expected time to initiate and teardown the direct-link [3]. That is, $T_{oh}^{(s,r,c)} = 0$ if one receive a segment from AP-to-MC path. $T_{oh}^{(s,r,c)}$ does not include timing overhead to switch channels while it consumes time to setup the direct-link if a MC-to-MC segment transfer is executed at F_{bss}. We set a threshold th_{p2p} to avoid the misallocation of MC-to-MC segment transfer at the congested channel. Therefore, we set $ETP_{(s,r,c)}$ is $unknown$ if it is less than th_{p2p}. The channel having $unknown$ $ETP_{(s,r,c)}$ are rarely used.

`MOVi Scheduler` should decide carefully F_{p2p} if there is more than one candidates for a pair of MCs going to execute a MC-to-MC segment transfer via miDLS. To fairly increase the video playout smoothness at each MC, `MOVi Scheduler` gives higher priority in downloading further segments to the one having smaller tolerance range of RBT [2]. To avoid such worst case channel allocation, `MOVi Scheduler` prefers the channel c that is most rarely available first so that the one widely available can be utilized for other MC-to-MC segment transfers. To this end, `MOVi Scheduler` examines the channel availability map on all neighbors of r and s when it decides F_{p2p}. If the availability of such channel is unknown and there is no available channel between a pair of MC going to execute a MC-to-MC segment transfers, `MOVi Scheduler` assigns the one randomly picked non-overlapped channel among unknowns. `MOVi Scheduler` can progressively updates $Cngs_r^c \forall r, c$.

D. Priority-based Scheduling

The enhancement of scheduling algorithm includes two idea explained before. Suppose we have a set of K MCs in the MOVi Domain. Each MC k has a set of neighbors N_k and is associated with one of AP $a_k \in A$. Each node periodically sends MS an on-demand request to download next segment to playout. The link quality between a_k and k is measured locally at k. The one between k and each $n_k \in N_k$ is also measured either passively or actively [2]. Each k has a set of available channels C to switch, and locally computation of $Cngs_k^c$ is done at k upon k has downloaded a segment from c right before. Each k keeps a set of downloaded segments, P_k ($P_k = \{p^1, p^2, ..., p^{|P_k|}\}$), of the targeted content. The progress of P_k is updated at MS by examining the received on-demand request messages, periodically transmitted from k. For each on-demand request from k to download $p^{|P_k|+1}$, the link quality measure $q(k, j) \forall j, where j \in \{a_k\} \cup N_k$ and $Cngs_k^c$

are piggybacked to the request [1].

Upon receiving on-demand request from each k, MS progressively updates the node connectivity graph G_t based on the piggybacked information $q(k,j)\forall j$. Then it updates the link-adjacent graph G_l that is mapping of G_t regarding *contact* between neighbors. At this step, edges between k and n_k is removed if either there is no reciprocal segments available between two or $q(k,a_k) > q(k,n_k)$. By expanding each vertex of G_l having multi-channel availability, MS is able to map G_l to the multi-channel link adjacent graph G_m. At this step, an edge established between two vertexes if they shares same channel and located within a same interference range. We assume that all two-hop vertexes of arbitrary k are within the interference range of k.

The vertexes of constructed G_m indicates available segment transfers within MOVi Domain. The effective scheduling of such transfers has to 1) minimize the playout distortion (i.e., palyout pause time) on each k as well as 2) maximize number of concurrent MCs meet the target QoS. In fact, the optimal scheduling is identical to the one of NP-hard problems that is *minimum sum coloring* of weighted (by ETP) graph G_m [19], [20].

To this end, we proposed a heuristic algorithm to solve the problem. Our approach is to build a priority-base scheduling that finds a series of schedule to maximize achievable throughput performance. The scheduling priority is assigned to reflect the urgency for the segment transfer which is based on the value of RBT for each k. The RBT quantifies the QoS measure of each k by describing the available time budget that k can continue video playout without receiving additional segments. Therefore, MCs with the lower value of RBT have the higher priority in downloading $to\text{-}be\text{-}playout$ content segments. The RBT of each k can be computed at MS as

$$RBT_k = \frac{|P_k| \times PPS \times L}{P^{Rate}} + T_i^{1^{st}REQ} - T^{NOW}, \quad (4)$$

where P^{Rate}, $T_i^{1^{st}REQ}$, and T^{NOW} refer to the video playout rate (bit-per-sec), the first on-demand request packet arrival time of k, and current time at MS, respectively.

IV. NS2-based Performance Evaluation

A. Simulation Implementation and Setup

We evaluate the performance of MOVi using the NS-2 simulator [11]. Taking the vanilla version of NS-2.29 as a starting point, we made extensive modifications to include miDLS and MOVi2 functionalities described in previous sections. Other major NS-2 modifications include:

1) **Cumulative interference-aware IEEE 802.11a PHY**. We implemented a new channel and PHY model that accumulates interference concurrently and therefore gives a more realistic SINR (signal to interference plus noise ratio) measurement. This enables us to accurately simulate the neighboring peer discovery of MOVi. We also implemented and applied the Ricean channel model [21] for our simulation. By using Auto Rate Fallback (ARF) [22] for PHY data-rate adaptation, we adapt appropriate PHY data-rate from 8 different PHY data-rates specified in IEEE 802.11a standard [7].
2) **Multi-channel Environments**. The PHY modification of NS2 simulator described above is extended to the multi-channel case where the TX/RX can be executed at the same time if they are not performed at the same channel. Also, there is no mutual interferences between two TX/RX performed in different channels.
3) **IEEE 802.11e MAC and AP selection scheme**. As the MAC protocol for MOVi, prioritized IEEE 802.11e EDCA [23] is implemented and used. We map the highest priority level of IEEE 802.11e [8] to the control packets such as beacon, address resolution protocol (ARP), miDLS control, and MOVi control messages. We also implemented a simple threshold-based AP selection scheme. For example, if the RSSI value cached for a particular AP is 2.0 dB (approximately 1.7 times) larger than the current associated AP, a node would associate itself with the new AP. It is assumed that no layer-3 handover will occur in the simulation and therefore no handover mechanism is implemented. To get rid of the active scanning and association steps to roam across APs configured different channel, we allow each mobile node can receive beacon messages from all APs in offline if the one within the transmission ranges. In fact, there is no additional bandwidth usages to transmit and receive beacon messages in our configurations.

Mobility and Topology. We use mobility traces extracted from the CRAWDAD data [6]. The estimated movement direction is derived using the triangular centroid method [24]. We are able to extract approximately 500 movement traces. Each of them lasts for approximately 20 minutes and fits with the human walking speed criteria. The locations of the APs are also based on the CRAWDAD trace. We select those popular APs that fit into an imaginary 1000m $\times$ 1000m boundary. We assume that the combined AP coverage can blanket the entire area. We then select a list of filtered traces which depict movements inside this area for the entire simulation duration. The simulation topology consists of 8 APs which are connected by 100Mbps wired links to MS. The link delays between APs and MS are set uniformly over [1ms:10ms]. To approximate a realistic setting, background traffic is generated by associating each AP with a stationary wireless nodes located at [+/-50m:+/-200m] from each AP. The traffic is set to 100kbps (in light loaded cases) and to 600kbps (in heavily loaded case) and is based on the Pareto distribution with a shape parameter of 1.5 and 500ms for the idle and the burst length, respectively.

Simulation Parameters. We set the content size to 14 Mbytes. The content is divided into 50 segments. Each segment consists of 200 packets. A packet is 1400 bytes (approximately the size of an Ethernet frame). The downloading request interval of each MC is based on the Pareto

[1]We only sends changes in practice to reduce the overhead [2].

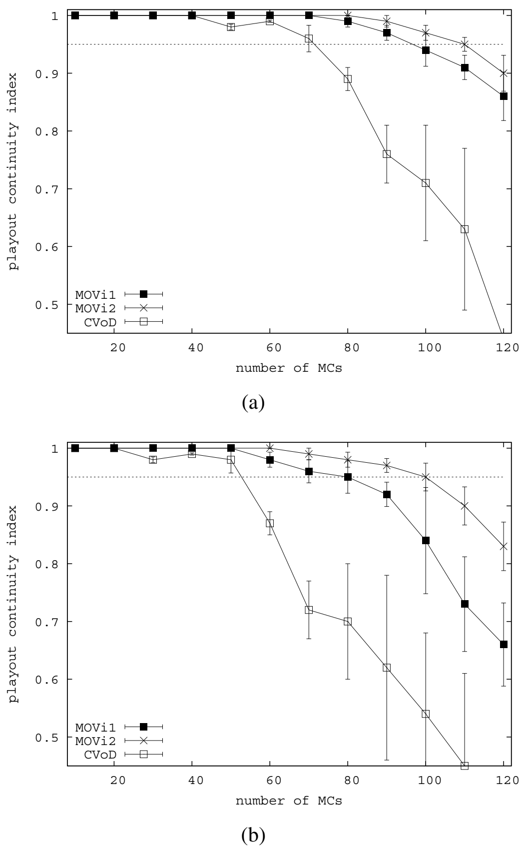


Fig. 4. PCI performance comparison under (a) light-loaded and (b) heavy-loaded.

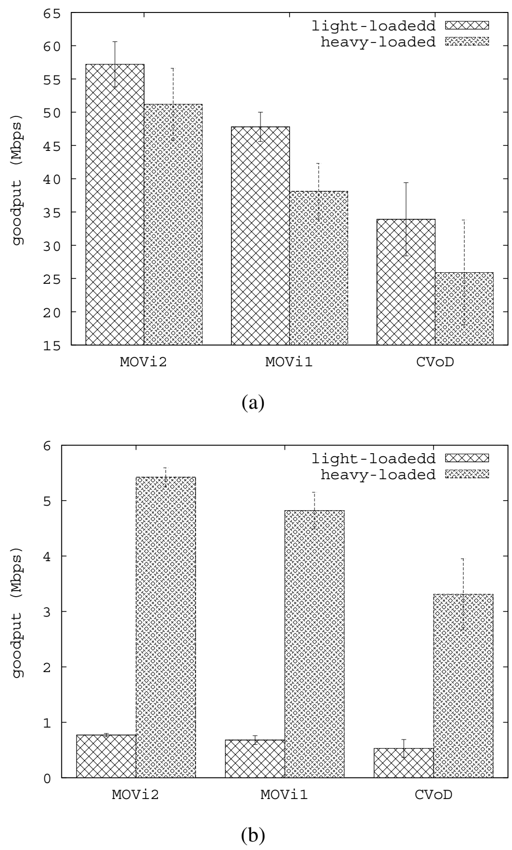


Fig. 5. Goodput performance comparison of 1) video traffic and 2) background traffic under both light-loaded and heavy-loaded.

distribution with a mean of 10 and shape parameter of 1.2. This distribution is obtained from an estimation of ISP (Internet service provider) streaming measurement [25] and represents the worst case scenario in the context of on-demand peer-to-peer streaming. Under this scenario, a large number of requests arrive in a short period of time. This leads to a saturated network condition and is unfavorable for peer-to-peer systems since only a few nodes are able to act as seeding peer nodes. The video source rate is set at 500kbps. The initial video buffering is set at 10% of the content size. UDP (user datagram protocol) is used as the transport protocol. The scheduling interval (T_{sch_int}) is set to 20ms. The MAC-layer RTS/CTS-protection disabled as usually done in practical WiFi deployment scenarios. Additionally, we assume to use four orthogonal channels available. We have set th_{p2p} to 10 Mbps. It is intended to select less busy channel to use for MC-to-MC segment transfers. It is noted that the channel allocation to each AP in the topology is intended to minimize the inter-cell interference. For the background traffic generations, each background generator randomly selects the channel before transmit a series of packets [2]. We have set the latency of channel switching to 5msec that is an average latency measured in our testbed and 5msec to T_{sync}.

[2]we use on/off traffic generation based on Pareto distribution.

We compare three schemes that runs the common video-on-demand scenario. The first one is CVoD. The second is the previous MOVi1. The third is the proposed MOVi2. We evaluate the performance of three by investigating the measure of PCI and achieved goodput performance under the both light and heavy background load scenarios.

B. Comparative Results

Playout Continuity Index (PCI) and Performance Comparison. PCI is a measure of the continuity of video playback in an end device. It is defined as

$$PCI = 1 - \frac{Pause\ Time}{Playout\ Time}, \tag{5}$$

where pause time is the total pause (freeze) duration experienced by the viewer and playout time is the total duration of successful video playback. The range for this index is 0<PCI≤1. A higher PCI value indicates less pause time and hence better video quality. Usually, a PCI value of 0.95 is regarded as an acceptable target performance for VoD services [26]. Figure 4(a) depicts the PCI performance under light-loaded background traffic cases. As it shows, CVoD experiences large amount of PCI loss compared to MOVi cases. We can also observe that MOVi2 outperforms MOVi1 at every population density. Figure 4(b) shows the PCI performance under heavy-loaded background cases. The proposed extension

of the `MOVi Scheduler` with miDLS increases the PCI performance more than 25% compared to MOVi1.

From the measurement output of the experiments, we have confirmed that the MOVi2 produces more concurrent MC-to-MC segment transfers. Although 15msec latency additionally applied to each MC-to-MC segment transfer, its higher concurrency can compensate the latency. It is also noted that the MOVi2 finding the less busy channel for initiation of MC-to-MC segment transfers. Thus it allows MOVi2 to be more tolerant to the increment of background traffic volume.

Goodput Comparison. One of the key contributors to the superior PCI performance of MOVi2 over both MOVi1 and CVoD is its ability to utilize the wireless medium more efficiently by using direct peer-to-peer communication and channel switching capabilities. This leads to an increase of achievable throughput and therefore increases the number of supported MCs given a specific PCI level. Figure 5(a) depicts the goodput of traffic that MOVi generates when the number of MCs is 100 under both the light-loaded and the heavy-loaded background traffic cases. For both cases, the measured goodput performance of MOVi2 is grater than both MOVi1 and CVoD. Especially for the heavy-loaded case, MOVi2 outperforms more than 35% and more than 100% when compared to MOVi1 and CVoD, respectively. Figure 5(b) shows the goodput performance of generated background traffic. In this scenario, the measured goodput performance of MOVi2 is grater than both MOVi1 and CVoD. This result shows that MOVi2 is also beneficial to non-MOVi users existing in the network.

V. Conclusion

In this paper, we propose enhancement of the previous version of MOVi, namely MOVi2, under the multi-channel environment, where each MOVi Client can switch WiFi operating channel quickly. Specifically we propose a scheduling algorithm to leverage the switching capability in order to increase the efficiency in distributing on-demand media contents. We further pointed out that the feasibility and the performance gain of MOVi2 via prototype-based implementations and experiments performed in OMF-based testbed configurations. Moreover, we confirm that MOVi2 can achieve up to 25% better capacity (in terms of concurrent users) compared with previous version of MOVi. In addition, the proposed extension of MOVi shows that the tolerance against increased volume of non-MOVi traffic increases.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012R1A2A2A01014687).

References

[1] WiFi Direct FAQ. (2010 May). [online]. Http://www.wi-fi.org/files/20091019_WiFi_Direct_FAQ.pdf

[2] H. Yoon et al., "On-demand video streaming in mobile opportunistic networks," in *Proc. IEEE PerCom '08*, 2008, pp. 80–89.

[3] H. Yoon et al., "iDLS: Inter-BSS direct link setup in IEEE 802.11 WLANs," in *Proc. ISCIT '07*, 2007, pp. 1015–1020.

[4] IEEE 802.11 WG, Part 11: Wireless LAN MAC and PHY specification, Aug. 1999.

[5] H. Yoon et al., "Mobility emulator for DTN and MANET applications," in *Proc. ACM WINTECH '09*, 2009, pp. 51–58.

[6] CrawDAD. (2010 May). [online]. Http://crawdad.cs.dartmouth.edu

[7] IEEE 802.11a WG, Part 11: Wireless LAN MAC and PHY specifications amendment 1: High-speed PHY in the 5GHz band.

[8] IEEE 802.11e WG, Part 11: Wireless LAN MAC and PHY specifications amendment 8: MAC QoS Enhancements.

[9] H. Yoon and J. Kim, "miDLS: Multi-channel inter-BSS direct link setup in IEEE 802.11 WLANs," *GIST NML Technical Report*, 2009.

[10] OMF Case-study: MOVi. (2010 May). [online]. Http://mytestbed.net/wiki/omf/CaseStudy2

[11] UCB/LBNL/VINT, network simulator - ns (version 2). (2010 May). [online]. Http://www.isi.edu/nsnam/ns/

[12] R. Chandra et al., "Multinet: Connecting to multiple IEEE 802.11 networks using a single wireless card," in *Proc. IEEE INFOCOM '04*, 2004, pp. 882–893.

[13] B. Yang and H. Garcia-Molina, "Designing a super-peer network," in *Proc. IEEE ICDE '03*, 2003.

[14] M. Heusse et al., "Performance anomaly of 802.11b," in *Proc. IEEE INFOCOM '03*, 2003, pp. 836–843.

[15] A. Sharma and E. Belding, "FreeMAC: Framework for multi-channel MAC development on 802.11," in *Proc. ACM PRESTO '08*, 2008.

[16] J. Eriksson et al., "Cabernet: Vehicular content delivery using WiFi," in *Proc. ACM MobiCom '08*, 2008, pp. 199–210.

[17] P. Acharya et al., " Congestion-aware rate adaptation in wireless networks: A measurement-driven approach," in *Proc. IEEE SECON '08*, 2008, pp. 1–9.

[18] D. Chafekar et al., "Cross-layer latency minimization in wireless networks with sink constraints," in *Proc. ACM MobiHoc '07*, 2007.

[19] E. Kubicka and A. J. Schwenk, "An introduction to chromatic sums," in *Proc. ACM Computer Science Conference '89*, New York, NY, 1989, pp. 39–45.

[20] D. Marx and D. A. Marx, "Graph coloring problems and their applications in scheduling," in *Proc. John von Neumann PhD Students Conference*, 2004, pp. 1–2.

[21] R. J. Punnoose et al., "Efficient simulation of Ricean fading within a packet simulator," in *Proc. IEEE VTC-Fall*, 2000.

[22] A. Kamerman and L. Monteban, "WaveLAN-II: A high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, 1997.

[23] S. Wiethoelter and C. Hoene, "Design and verification of an IEEE 802.11e EDCF simulation model in ns-2.26," *Technical Report of Telecommunication Networks Group, TU-Berlin*, 2003.

[24] M. Kim et al., "Extracting a mobility model from real user traces," in *Proc. IEEE INFOCOM '06*, 2006, pp. 1–13.

[25] C. P. Costa et al., "Analyzing client interactivity in streaming media," in *Proc. ACM WWW '04*, 2004, pp. 534–543.

[26] B. Li et al., "An empirical study of the CoolStreaming+ system," *IEEE Journal of Selected Areas in Communications*, vol. 25, pp. 1627–1639, 2007.