This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2691706, IEEE Transactions on Mobile Computing

Multi-source Video Multicast in Internet-connected Wireless Mesh Networks

Wanqing Tu
School of CS
Robert Gordon University
UK
Email: w.tu@rgu.ac.uk

Cormac Sreenan
Department of CS
University College Cork
Ireland
Email: cjs@cs.ucc.ie

Sanjay Jha
School of CS
University of New South Wales
Australia
Email: sanjay@cse.unsw.edu.au

Qian Zhang
Department of CS
The Hong Kong University of Science and Technology
China
Email: qianzh@cse.ust.hk

***Abstract*—Wireless mesh networks (WMNs) connect to the Internet via access gateways. This paper studies multi-source video multicast in Internet-connected WMNs. The focus is on the design of a shareable integrated multicast that allows the multicasts of video sources to employ common Internet shortcuts or WMN paths to avoid potentially high WMN overheads and excessive Internet usage. Several algorithms are described that together form a video multicast framework running a controlled number of shareable multicasts under the constraint of Internet availability. These algorithms are the resource-efficient source group algorithm, the efficient integrated architecture algorithm, and the interference-controlled multicasting tree algorithm. These algorithms represent different approaches to overcoming various costs arising from multi-source video multicast, enabling multiple video sources to distribute delay and throughput-guaranteed videos to receivers across large-scale areas. Simulation results are presented that quantify the performance gains that can be achieved.**

***Keywords*—Multi-source multicast, wireless mesh networks, large-scale routing, video streaming, shareable multicast.**

I. Introduction

Emerging public-oriented video applications (e.g., HD video conferencing, streaming sensor feeds, 3D virtual worlds) often require multiple communication nodes to economically deliver performance-guaranteed content to a group of receivers distributed over large areas and in many instances over wireless links. Wireless Mesh Networks (WMNs) offer a low capital expenditure (CAPEX) solution for such services by providing a multi-hop relay backbone composed of low-cost mesh gateways or routers with fixed power supplies. Although a multi-hop WMN backbone can be physically extended to cover wide areas, theoretical studies [17] have found that transmissions via multiple wireless hops suffer observed performance loss. This loss is more severe when carrying higher traffic loads, as reported in the literature (e.g., [19]). Hence, for an application with multiple video sources generating large data volumes, scalable data distribution presents a difficult challenge.

Studies on multimedia communications via WMNs have developed both straightforward broadcast to all users as well as single-source multicast to selected subscribed users only. Major strategies adopted in these studies focus on efficient wireless resource utilization, including the use of channel diversity (e.g., [1,15]) to provide non-interfering transmission capacity, the scheduling of transmission rates (e.g., [3]) or video multicasts (e.g., [2]) to make the most of wireless connection capacity, the collaboration of WMN peer nodes to provide alternative paths to avoid bottlenecks (e.g., [4]), and the exploration of cognitive radio networks to seek extra wireless bandwidth (e.g., [5]). These strategies intelligently create more transmission opportunities. However, the gains in capacity or transmission times are subject to the availability or reliability of wireless resources which may not be sufficient for video communications over long durations or distances. As such, the exploitation of Internet resources for WMN communications via access gateways (also called as mesh gateways) has become a topic of interest in quite a few research projects [7-11].

Via the Internet, instead of relying on pure wireless paths inside a WMN (called intramesh routes), alternative routing paths (called integrated routes) can be established to bridge distant WMN nodes using wired links which avoids long-distance transmissions inside a WMN. In general, WMN nodes use Internet shortcuts either naively through their closest gateways (e.g., [9]) or by a planned structure that is designed with consideration for gateway conditions and Internet accessibility (e.g., [10-11]). The structure approach performs better with respect to communication scalability because the employment of intramesh routes or integrated routes takes network conditions and WMN node locations into account. However, with the structured integration between WMNs and the Internet, when an application has multiple sources to send video data to a group of receivers, as we will discuss in Section III, video transmissions from different sources may yield different choices of integrated structures. This not only increases complexity in establishing or maintaining communication topologies but also produces great overheads such as extra control traffic, complicated interference, additional computation tasks, etc., negatively affecting video performance. Thus, in order to realize simple yet efficient large-scale communications for a multi-source WMN video application, new research efforts are needed.

This paper studies a new multicast framework in order to efficiently address a set of challenges regarding improving the scalability of performance-guaranteed multi-source video communications by using structured integration. Without loss of generality, we use the term *integrated multicast* to refer to a multicast combining available Internet resources and WMN bandwidth, and the term *session* to mean the multicast of a single source in a multi-source application. We first investigate and account for the fundamental causes of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2691706, IEEE Transactions on Mobile Computing

challenges when running multi-source video multicast with state-of-art integrated multicast. On this basis, we develop the following interlinked novel algorithms to form our *multi-source supporting integrated video multicast* (MSIM) scheme.

- The *resource-efficient source group* (RESG) algorithm separates video sources into *performance groups* so that sessions issued by the video sources belonging to the same performance groups are able to share integrated routing paths (built by the following two algorithms) and hence control overhead cost in constructing individual routes. Moreover, in order to reduce the overheads of forming MSIM as well as reasonably utilize Internet resources, the algorithm sets up a controlled number of performance groups under the constraint of available Internet capacity.
- The *efficient integrated architecture* (EIA) algorithm develops integrated routing paths that can be shared by video sessions belonging to the same performance groups. The algorithm improves communication scalability by constructing best-effort access areas (interconnected via a wired network) that 1) can make full use of wireless capacity heterogeneous between different paths before resorting to Internet capacity, and 2) have greatly reduced overlaps so as to cover the same group of video receivers with controlled overheads.
- The *interference-controlled multicasting tree* (IMT) algorithm forms multicast routing strategies inside the WMN of our MSIM. In order to limit the influence of various sources of wireless interference on the performance of multi-session video multicasting, different strategies, including the construction of multicast trees, the efficient utilization of channel diversity, and the performance-guaranteed scheduling of video sessions, are explored to constitute the IMT algorithm.

The rest of the paper is organized as follows. Section II discusses related work. Section III formulates the problems studied in the paper. Section IV presents the algorithms for constructing MSIM. Section V describes the discrete event simulation setup and evaluation of various scenarios. Finally, Section VI concludes the paper.

II. Related Work

Studies on WMN multimedia communications have investigated the use of modern wireless techniques to improve transmission performance inside a WMN. Zeng *et al.* [1] used multiple channels with no overlap or low overlap to increase WMN multicast capacity. Moreover, Breadth First Search was employed to control the number of relay nodes in order to reduce interference that could decrease WMN capacity. However, due to the limited available channels, the proposed multicast scheme may not achieve sufficient capacity to carry multi-source multicast over multiple WMN hops. Hence, Tu [2] explored how to efficiently utilize channel capacity. By designing a new channel aggregation scheme and a new flow scheduling scheme in multi-flow multicasting scenarios, extra multicasting opportunities are achieved when WMN channels are considered to be "saturated" in conventional studies. Apart from multiple channels, the advantages of multiple transmission rates have also been studied in the literature. Qadir *et al.* [3] proposed a WMN broadcast scheme which enables a mesh node to schedule multiple channels to work at different rates in order to provide short delays to broadcast receivers. With this scheme, a channel transmits at a rate that enables data to reach neighbours located outside the coverage of any transmission rates larger than the employed rate. Tu [19] introduced the parallel low-rate transmission (PLT) scheme in which multiple channels transmit at the same low rate to balance the tradeoff between transmission throughput and transmission coverage. A multicast algorithm (LC-MRMC) is then designed to deliver a higher aggregate throughput to widely distributed multicast receivers under limited channel availability. Chang *et al.* [20] studied rate adaptation to avoid interference caused by concurrent transmissions. A metric known as standard deviation of average remaining broadcast time is proposed to determine the priority between transmission rates and the number of concurrent transmissions.

In [4], Xiong *et al.* proposed PeerCast to engage mesh users in cooperative relaying, allowing access points to adaptively adjust transmission rates to avoid bottleneck nodes. Hou *et al.* [5] explored cognitive radio technology to make use of spare licensed radio spectrum to gain extra transmission capacity for wireless multimedia communications. A framework that employs cooperative transmissions and network or superposition coding to multicast layered videos in multi-channel cognitive radio networks is proposed. Bhattacharya *et al.* [6] split a multicast stream to fit it into licensed spectrum fragments with different sizes. Although the above studies intelligently achieve additional transmission opportunities inside WMNs, as analyzed in [17], the per-node multicast throughput of a random multihop network with n nodes, n_s multicast sources, and n_d destinations is bounded by $O(min(1, \frac{\sqrt{n}}{n_s\sqrt{n_d}\log^n}))$ with a high probability. This implies that multicast throughput is a decreasing function of the network size, showing that the additional transmission opportunities yet achieved are not reliable across long distances to cover more users.

Integrated wireless transmissions were then studied to address the challenge by exploiting the potential advantages provided by mesh gateways. Lakshmanan *et al.* [7] presented a multi-gateway association model in which a mesh user may adaptively choose different mesh gateways to transmit different packets to a single destination. Although the model helps to balance traffic load in a WMN, it unfortunately generates complicated topologies if being used to send multimedia packets to a group of receivers. Liu *et al.* [8] investigated the integration of a WMN with the Internet via mesh gateways. This work focuses on unicast traffic, and shows that the scalability of network capacity would be significantly increased if Internet shortcuts were employed. Ruiz *et al.* [9] proposed a routing mechanism where mesh nodes form an "island" with prefix continuity. Mesh nodes connect to the Internet through a shared "closest" gateway. The selection of "closest" gateways purely depends on topology, failing to consider the tradeoff between the selection of a closer but more congested gateway *vs.* the use of a farther, less utilized gateway. Therefore, in [10-11], the resource-aware video multicast framework (RAM) makes judicious use of gateway resources to efficiently combine available Internet resources and intra-WMN bandwidth. RAM

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2691706, IEEE Transactions on Mobile Computing

provides an efficient solution for large-scale single-source video multicast. This paper studies how to efficiently multicast a multi-source video application to widely distributed video receivers.

III. Problem Formulation

In this section, we use RAM [11] as an example to briefly illustrate the terminology of integrated multicast. We then explain how the state-of-art is inappropriate for multi-source multicast. On this basis we form our study objectives in the paper.

Let $\mathbf{N}$ denote the natural numbers. Suppose a group of $M (M \in \mathbf{N}, M > 1)$ nodes participates in a video application V that has $I (I \in \mathbf{N}, I \leq M)$ sending sources. Table I lists major symbols appearing in this paper.

TABLE I: Symbol List

I	Number of video sources that the application V has.
M	Number of sources and receivers that V has.
N	Number of nodes that construct or implement the multicast of V.
n_j	The jth ($j \in [0, N-1]$) node in the multicast.
G	Number of performance groups (or shareable sub-architectures) formed for V.
g_i	Number of video sources in the ith ($i \in [0, G-1]$) performance group.
BS_i	The benchmark source of the ith ($i \in [0, G-1]$) performance group.
U_j	Uploading performance of the jth ($j \in [0, I-1]$) video source.
ΔU	Threshold of uploading performance differences used to form performance groups.
C_a	Maximum Internet capacity additionally allocated to any session of V when the session shares an integrated multicast.
σ_j	Burstiness of the jth ($j \in [0, I-1]$) video session.
ρ_j	Average transmission rate of the jth ($j \in [0, I-1]$) video session.
$\bar{\sigma}$	Average burstiness of I video sessions.
$\bar{\rho}$	Average transmission rate of I video sessions.

A. Introduction of Integrated WMN Multicast

We use RAM [11] as an example to introduce the current integrated WMN multicasting technology as well as the terminology of integrated WMN multicast that will be used in the paper. In Fig. 1 (a), suppose node 1 multicasts video data to nodes $2 \sim 5$. As shown by the blue dashed lines, RAM assigns multicasting nodes to different access areas, based on the hop distances from the sender to these multicasting nodes. Such hop distances should not exceed a threshold K[1] in order to achieve acceptable end-to-end throughput, where $K \in \mathbf{N}$ is the maximum number of WMN hops that video data can travel without violating the minimum required throughput. In this example, $K = 3$. The first access area (referred to as the source access area) is formed by the sender (i.e., node 1) by selecting nodes within a K-hop wireless distance. Access area 1 in Fig. 1 (a) is the source access area. This access area may cover more than one mesh gateway among which the sender selects one as the uploading gateway (UG) by the WGU algorithm [11]. For example, G_1 is selected as the UG of node 1 in Fig. 1 (a). The WGU algorithm ensures that, through the UG, the sender is able to upload video data to the Internet quickly and reliably.

Thereafter, the UG selects AGs to form more access areas (called non-source access areas) in order to connect all multicasting receivers to the integrated multicast. An AG is a mesh gateway that has the least value of $\frac{d}{o}$ among all gateways that have not joined any access areas, where d is the gateway's delay distance to the UG and o is the available (residual) wireless transmission capacity of the gateway. In Fig. 1 (a), G_3 is the AG of access area 2. AGs select nodes within a $(K-k)$-hop wireless distance to form non-source access areas, allowing those receivers who have not joined an access area to be linked to the Internet, where $k \in \mathbf{N}$ and $k \leq K$ is the hop distance between the sender and its UG. In this example, $k = 1$.

Within each access area, as shown by the red arrows, a link-controlled routing tree (LCRT) is constructed to multicast video data to receivers wirelessly. Such a tree may have multiple roots that are the AG and corresponding gateways (CGs) in the access area. As one of the tree roots, a CG is a mesh gateway that jointly delivers video data coming from wired links to receivers in its access area. In Fig. 1 (a), G_4 is a CG of access area 2.

Access areas are connected via Internet links through the UG, AGs, and CGs, allowing the UG of the video source to deliver video data to AGs and CGs.

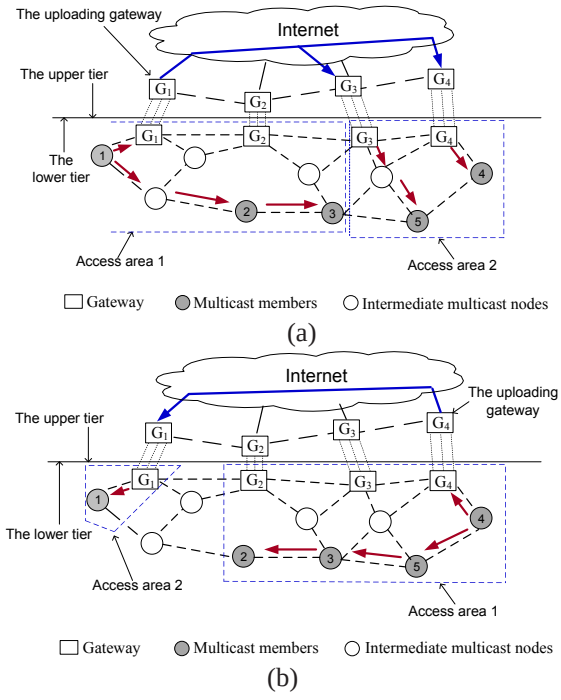


Fig. 1: An example of RAM integrated multicast [11]: (a) node 1 is the video source; (b) node 4 is the video source.

[1]According to [11], the threshold of wireless hops K should meet

$$K \leq log_{\frac{1}{\ell - l}} \frac{\min\{r_V, \frac{C}{\lambda \pi (\kappa d)^2}\}}{(r_V)_{basic}},$$

where r_V is the sending rate of V, $(r_V)_{basic}$ is the rate of the lowest-acceptable video quality, C is the capacity of a wireless link in the system, λ is the average density of multicasting nodes in our WMN system, d is the average distance of one wireless hop, $\kappa > 1$ is a factor introduced to express that an interference range is usually larger than a transmission range, $\ell = \max\{\ell_i, i \in [1, K]\}$, and ℓ_i is the loss rate of the link at the ith hop. Thus, the threshold takes the throughput performance into account.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2691706, IEEE Transactions on Mobile Computing

B. Problem Formulation

Now assume node 4 is another video source in Fig. 1. The RAM architecture shown in Fig. 1 (a) does not allow node 2 to receive throughput-guaranteed video data from node 4, as the integrated route $<$ *node 4*$\rightarrow G_4 \xrightarrow[InternetLinks]{via} G_2\rightarrow$*node 6*$\rightarrow$*node 3*$\rightarrow$*node 2* $>$ has 4 wireless hops which is longer than the threshold 3. In order to deliver the packets of node 4, a new RAM architecture (shown in Fig. 1 (b)) will be established which however is not suited to delivering node 1's session, having too many WMN hops between nodes 1 and 2. The fundamental reason for the two video sessions to employ different RAM integrated multicasts is because the two sessions are uploaded to the Internet via different paths. Different paths have different hop distances. Accordingly, the remaining hop distances (before reaching K) after the two sessions arrive at the Internet are different. Since RAM constructs its non-source access areas based on the remaining hop distances, different integrated architectures are constructed for the two sessions.

The above observation exists when other metrics (rather than hop distances) are employed to form access areas. This paper takes throughput and delays into account when evaluating communication performance as they are two major performance metrics of video applications. We define the uploading performance of a video session as the performance when the video session arrives at the Internet. More specifically, the uploading performance of the ith ($i \in [0, I-1]$) video source is defined as

$$U_i = (T_u)_i \times \frac{1}{(d_u)_i}, \tag{1}$$

where $(T_u)_i$ and $(d_u)_i$ are the uploading throughput and uploading delay of the ith source respectively, i.e., the throughput and delay when the ith session arrives at the Internet. Denote the end-to-end performance bound as $\bar{P} = \frac{\bar{T}}{\bar{D}}$, where $\bar{T}$ and $\bar{D}$ are the end-to-end throughput bound and the end-to-end delay bound of V respectively. For two video sources (say the ith or the jth sources ($j \in [0, i) \bigcup (i, I-1]$)), if the uploading performance of the jth source is worse than the uploading performance of the ith source, we have $U_j \leq U_i$. When the sessions generated by the jth source and the ith source arrive at the Internet, the performance leeways, i.e., the remaining performance that can be consumed before degrading to the bound $\bar{P}$, of these two sessions are $U_j - \bar{P}$ and $U_i - \bar{P}$ respectively. Since $U_j \leq U_i$, we obtain $U_j - \bar{P} \leq U_i - \bar{P}$. This means that, after transiting the Internet and then being distributed to receivers wirelessly from the same gateways, the jth session can travel a wireless distance which should not be longer than the wireless distance that the ith session can travel. Therefore, if a WMN path is established to ensure the distribution of the jth video session with guaranteed performance, the same path can be used by the ith video session to distribute its data with guaranteed performance.

We use the example in Fig. 2 to illustrate the above insight. In the figure, the blue dotted lines delimit the source access areas of s_1 and s_2 respectively. Suppose $U_2 \leq U_1$, i.e., the uploading performance of s_2 is worse than the uploading performance of s_1. After passing across the Internet, the performance leeway of the video session of s_2 allows the session to travel a short wireless distance, i.e., from G_3 to its two directly connected black nodes or from G_6 to its two directly connected black nodes, as shown by the red lines. For the video session issued by s_1, after arriving at G_3 via Internet connections, its performance leeway (which is greater than that of s_2) allows the session to use the paths illustrated by red dotted lines (i.e., the paths constructed for the session of s_2) to transmit with guaranteed performance. Therefore, for two video sources (say the ith or the jth sources), if $U_j \leq U_i$, the video session issued by the ith source is able to share non-source access areas constructed for the video session issued by the jth source, but the reverse is not true. This insight motivates us to study shareable integrated multicasts which can effectively control overheads caused by constructing different architectures for different video sessions.

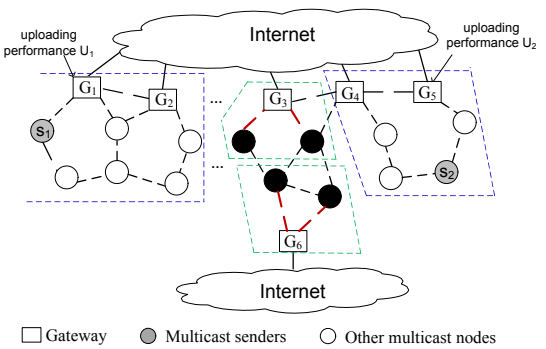


Fig. 2: An example of sharing integrated multicast created by the source with the worst uploading performance.

Ideally, all I video sessions share common non-source access areas which is possible if these non-source access areas are built by the video source with the worst uploading performance cost. However, such sharing may not efficiently utilize both wired and wireless resources as the sessions of other video sources have to stop WMN multicasting even when they could be spread to more distant areas wirelessly. For example, in Fig. 2, suppose the performance leeway of video session of s_1 allows the session to be spread to all black nodes via G_3. However, if sharing the architecture built up for s_2, as shown in the figure, the session of s_1 may have to be delivered to black nodes via both G_3 and G_6 through two small access areas (delimited by the green dotted lines). This indicates that sharing an integrated multicast leads to a greater consumption of Internet capacity while leaving WMN capacity under-utilized, as access areas are connected via Internet links in an integrated multicast. When several video sources share the non-source access areas of the video session with much worse uploading performance, the claiming of extra Internet capacity may quickly result in excessively using Internet links (while WMN capacity may be under-utilized) which degrades performance. We define C_a as the maximum Internet capacity that any single session of the application V is allowed to additionally use for the purpose of sharing an integrated architecture. Our first objective is to balance the tradeoff between sharing integrated multicasts (for greatly controlled operational overheads) and keeping the extra Internet usage caused by a session to share integrated multicasting below C_a. Namely, our design will efficiently use

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2691706, IEEE Transactions on Mobile Computing

both WMN and Internet capacities.

Another challenge relating to multi-source video multicast is the complicated interference topology inside a WMN. This is because the wireless transmission medium is shared between transmissions. For nodes belonging to different architectures, if they are adjacent to each other, multicasting via these architectures causes mutual interference. Also, for nodes on the same architecture, if they are close to each other (i.e., within each other's interference range), multicasting interference happens on the same architecture. The sharing of integrated multicasts between video sessions reduces the chance of interference as common routing paths are employed. However, once interference happens, it could be intensive as each shared multicast carries several video sessions generating a high accumulated transmission rate. Hence, another of our major objectives is to take further and effective control of interference in order to multicast a multi-source video application in WMNs with guaranteed performance.

IV. Multi-source Supporting Integrated Video Multicast (MSIM)

The overall architecture of our *multi-source supporting integrated video multicast* (MSIM) consists of multiple integrated sub-architectures. In order to construct an integrated sub-architecture, as shown in Fig. 3, the following three algorithms are implemented.

- **Resource-efficient source group** (RESG). RESG groups I video sources into a controlled number of performance groups under the limitation of Internet capacity so as to reduce overheads when running EIA to construct sub-architectures. To achieve this, two major processes are proposed 1) selecting benchmark sources (BSs) and 2) deciding a threshold of uploading performance differences between benchmark sources and the video sources that can be assigned into the performance groups of these BSs.
- **Efficient integrated architecture** (EIA). EIA constructs the overall architecture of MSIM by forming an integrated sub-architecture for each performance group. A sub-architecture is a set of best-effort access areas (BEAAs) that are interconnected by a multicast wired network. The formation of BEAAs considers the heterogeneity of WMN links as well as reducing overlaps between BEAAs. This enables WMN capacity to be efficiently utilized to extend video wireless transmission range and hence reduce traffic load introduced to the Internet.
- **Interference-controlled Multicasting Tree** (IMT). IMT is the video multicasting strategy inside the WMN of the overall MSIM architecture. It efficiently controls multicast interference on single sub-architectures by building interference-controlled multicast routes and applying channel diversity, and multicast interference between sub-architectures by studying a new scheduling transmission scheme.

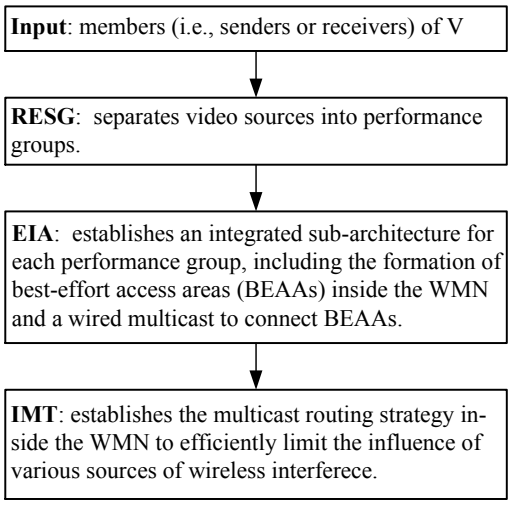


Fig. 3: The implementation of MSIM algorithms.

A. Resource-efficient Source Group

RESG groups the I video sources of V into G performance groups based on the uploading performance costs of these sources. Each performance group has a benchmark source that should have the worst uploading performance (i.e., the least value of U) among all video sources belonging to the performance group, as analyzed in Section III. Suppose the ith video source is a benchmark source. A video source (say the jth source, $j \in [0, i) \cup (i, I-1]$) can join the performance group of the ith source if $U_j - U_i \leq \Delta U$, where ΔU is the threshold of uploading performance difference employed to form performance groups. We call the jth video source as a non-benchmark source. Then, among the remaining video sources that have not joined any performance groups, the source with the worst uploading performance becomes a new benchmark source which chooses its performance group members in the same fashion as the ith video source. The procedure continues until all video sources join a performance group. In the rest of the paper, we denote the benchmark source of the ith ($i \in [0, G-1]$) performance group as BS_i and the number of video sources in the ith performance group as g_i.

1) ΔU: The value of ΔU decides the number of performance groups that will be formed for the application V. If the I video sources are assigned to a smaller number of performance groups, less overhead will be generated when constructing the MSIM architecture as each performance group requires a sub-architecture. However, a smaller number of performance groups means that each performance group may include more video sources which potentially have a larger range of uploading performance. This will increase the usage of Internet links while leaving WMN capacity underused. In order to save Internet capacity, the value of ΔU should balance the above tradeoff so as to reduce the number of performance groups while not exceeding the Internet capacity C_a additionally allocated to any session of the application V for the purpose of sharing integrated architectures.

Our analysis employs the theoretical results in [12] to model multimedia flows. Given $\sigma > 0$ and $\rho > 0$, for a multimedia flow, if its transmission rate at time t is given by the function $R(t)$, the following inequality exists if and only

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2691706, IEEE Transactions on Mobile Computing

if $y \geq x$ for all x and y

$$\int_x^y R(t)dt \leq \sigma + \rho(y-x),$$

where σ and ρ are the burstiness and the average transmission rate of the multimedia flow. More specifically, if the multimedia flow is fed to a node that works at rate ρ, the size of backlog will never be larger than σ. Namely, σ is in spirit somewhat related to the peakedness characterization of the multimedia flow.

Theorem 1 *For a video application V with I sources implementing an MSIM integrated multicast, we denote the performance bound as $\bar{P} = \frac{\bar{T}}{\bar{D}}$, where $\bar{T}$ and $\bar{D}$ are the end-to-end throughput and delay bounds of V respectively. In order to guarantee the extra Internet capacity required by any session of V not to exceed the allocated amount of capacity C_a when the session shares an integrated architecture, the uploading performance difference threshold expressed as*

$$\Delta U = \bar{P}(\frac{C_a}{\lambda} - 1)$$

helps to form I sources into a controlled number of performance groups, where $\lambda = \sigma_{max} + \rho_{max}$, and σ_{max} and ρ_{max} are the burstiness and the average transmission rate of the session that has the largest transmission rate among all video sessions of V.

Proof. As we analyzed, after passing across the Internet, a session issued by a non-benchmark video source in performance group i ($i \in [0, G-1]$) should have more performance remaining (before reaching $\bar{P}$) than the video session of BS_i. As a result, the remaining performance of the non-benchmark video session enables this session to be wirelessly spread to more receivers. In other words, if the non-benchmark video session shares the integrated multicasting paths of BS_i, it occupies extra Internet capacity as compared to using integrated multicasting paths built on its own. More specifically, for a non-benchmark video session (say the jth video session ($j \in [0, I-1]$)) in performance group i, it will require $(\frac{U_j - \bar{P}}{U_i - \bar{P}} - 1)$ times more Internet capacity than if the same session were to use the integrated multicast built on its own basis. Since $U_j \in [U_i, U_i + \Delta U]$, it is inferred that $\frac{U_j - \bar{P}}{U_i - \bar{P}} - 1 \leq \frac{U_j - U_i + \bar{P}}{U_i}$[2].

Denote the transmission rate of the jth session at time t as $R_j(t)$. The Internet capacity additionally occupied by the session during a period τ is $\frac{U_j - \bar{P}}{U_i - \bar{P}} \frac{\int_t^{t+\tau} R_j(t)dt}{\tau}$.Based on [12], we have

$$\int_t^{t+\tau} R_j(t)dt \leq \sigma_j + \rho_j \tau,$$

where σ_j and ρ_j are the burstiness and the average transmission rate of the jth non-benchmark video session. This means that the additional Internet capacity requested by this session has an upper bound of

$$\frac{U_j - \bar{P}}{U_i - \bar{P}}(\frac{\sigma_j}{\tau} + \rho_j).$$

Among all $(I - G)$ non-benchmark sources in the G performance groups, the greatest request for extra Internet capacity by a single session of V (in order to share integrated multicasting paths) arises when the session is issued by a non-benchmark source that has the largest uploading performance difference from that of its benchmark source among all non-benchmark sessions and the session has the highest transmission rate among all video sessions of V. This additional Internet capacity is given by

$$\frac{U_j - U_i + \bar{P}}{U_i}(\frac{\sigma_{max}}{\tau} + \rho_{max}),$$

where σ_{max} and ρ_{max} are the burstiness and the average transmission rate of the session with the highest transmission rate. Furthermore, as this session has the largest uploading performance difference from that of its benchmark source among all non-benchmark sessions, and $U_i \geq \bar{P}$, we have

$$\frac{U_j - U_i + \bar{P}}{U_i}(\frac{\sigma_{max}}{\tau} + \rho_{max}) = \frac{\Delta U + \bar{P}}{U_i}(\frac{\sigma_{max}}{\tau} + \rho_{max})$$

$$\leq \frac{\Delta U + \bar{P}}{\bar{P}}(\frac{\sigma_{max}}{\tau} + \rho_{max}) = (1 + \frac{\Delta U}{\bar{P}})(\frac{\sigma_{max}}{\tau} + \rho_{max}).$$

We now consider the wired links involved in the video application V. Assume a maximum of C_a Internet capacity is additionally allocated to any single session of the application to implement shared integrated multicast. In order not to exceed this allocated Internet capacity, we have

$$(1+\frac{\Delta U}{\bar{P}})(\frac{\sigma_{max}}{\tau}+\rho_{max}) \leq C_a \Rightarrow \Delta U \leq \bar{P}(\frac{C_a}{\frac{\sigma_{max}}{\tau} + \rho_{max}} - 1). \quad (2)$$

For the value of τ, as a video application, we assume that it is greater than one second. Hence, expression (2) holds if $\Delta U \leq \bar{P}(\frac{C_a}{\sigma_{max}+\rho_{max}} - 1)$ holds. Let $\lambda = \sigma_{max} + \rho_{max}$. In order to control the number of performance groups, we use

$$\Delta U = \bar{P}(\frac{C_a}{\lambda} - 1).$$

Q.E.D

2) The RESG Algorithm: We assume the existence of a group manager (GM)[3] that generates an ID (denoted as *g_id*) for the multicast application V and maintains a list of members (i.e., senders and receivers) of V. In order to form performance groups based on Theorem 1, the following algorithm is implemented. Note that, in Algorithm 1, the throughput field of an ACK is initially set by the gateway issuing the ACK with the value $c \times l$, where c and l are the available capacity and the loss rate of the link that the gateway transmits the ACK out. This field is updated by any intermediate node if it sends the ACK with a lower throughput.

Algorithm 1 Resource-efficient Source Group
Input: I video sources of V, ΔU achieved by Theorem 1
Output: G performance groups

1. Each source broadcasts a REGISTRATION message (*shown by the black arrows in Fig. 4*) that includes the fields of *g_id* and *s_time*; // *s_time* records the time that the source broadcasts this REGISTRATION;
2. On receiving a REGISTRATION packet, a node (either

[2] $\frac{U_j - \bar{P}}{U_i - \bar{P}} - 1 = \frac{U_j - \bar{P} - U_i + \bar{P}}{U_i - \bar{P}} = \frac{U_j - U_i}{U_i - \bar{P}} \leq \frac{U_j - U_i + \bar{P}}{U_i}$

[3] There are quite a few studies (e.g., RRAS multicast group manager) proposed methods to set up a GM.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2691706, IEEE Transactions on Mobile Computing

a mesh gateway or a WMN node) subtracts the value in *s_time* from the receiving time of the packet; // The node only calculates the delay of the first REGISTRATION packet received from a source;

3. If the calculated delay $\leq$ the one-way delay bound $\bar{D}$, the node continues broadcasting this received REGISTRATION via WMN links; // The purpose of such broadcast is to search potential gateways that a source can reach with guaranteed delays;

4. **If** the node is also a mesh gateway, it replies an ACK packet (*shown by the blue arrows in Fig. 4*) via the path on which it receives the REGISTRATION. The ACK carries the calculated one-way delay as well as collects the throughput of the path;

5. Among all gateways that reply a source with ACKs, the source selects the one with the largest $\frac{T_u}{d_u}$ value as its UG; (*In Fig. 4, source* s_0 *and* s_1 *select* G_0 *and* G_3 *as their UGs respectively.*)

6. The GM designates the first video source of V that contacts the GM (*e.g.,* G_0 *in Fig. 4*) as the coordinator of RESG;

7. The UG of each non-coordinator source unicasts a GATEWAY message to the coordinator (*shown by the red arrows in Fig. 4*), including the uploading performance cost and its IP address[4];

8. The coordinator sorts I sources in decreasing order of uploading performance costs to their UGs, sets $i = 0$, $G = 0$;

9. While $i \leq (I-1)$

10. The ith source becomes BS_G; $g_G = g_G + 1$; $j = 1$;

11. **If** $(U_{(j+i)} - U_i \leq \Delta U)$, the $(i+j)$th source joins performance group G, $g_G = g_G + 1$, $j = j + 1$;

12. **Otherwise**, the Gth performance group $\{s_i, s_{(i+1)}, ..., s_{(i+j-1)}\}$ is formed;

13. The coordinator unicasts a GROUP message to all members in the Gth performance group (*shown by the green arrows in Fig. 4*), including the performance group id G and the IP addresses of these members;

14. BS_G starts constructing an integrated sub-architecture for video sessions belonging to the Gth performance group by the algorithms in the next sections; $i = i + j$; $G = G + 1$;

Algorithm 1 selects a UG and calculates the uploading performance for each video source based on which the I sources are compared with each other to form performance groups. These steps have a complexity of $O(I^2)$ for Algorithm 1, where I is the number of video sources that the application V has. With dynamic network conditions, a performance group (say group i $(i \in [0, G-1])$) may not remain the best for a video source as the uploading performance of this source may change. In this case, the UG of this source informs the RESG coordinator of the new uploading performance U'. By comparing $(U' - U_{BS_i})$ and ΔU, the RESG coordinator assigns the source a new performance group, where U_{BS_i} is the uploading performance of the ith benchmark source.

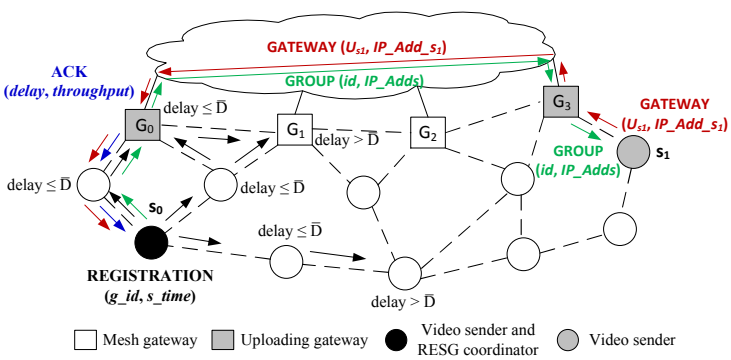


Fig. 4: An example of the RESG algorithm.

- If this source is a benchmark source, the coordinator selects a backup benchmark source at random from all sources belonging to the ith performance group. The UG of this backup benchmark source will be informed of all AGs on its sub-architecture by the UG of the original benchmark source. (The formation of a sub-architecture will be introduced in the next section.)
- Otherwise, the UG of the video source is informed of the AGs on the sub-architecture, established for the performance group that the video source joins, by the UG of either the benchmark source or the backup benchmark source.

A video source may cache the AG information for each sub-architecture that it has joined, helping it to change between performance groups with greatly reduced control overhead. However, if the video source cannot find an existing performance group to join (because of $(U' - U_{BS_i}) > \Delta U$), it invokes the procedure of establishing a new sub-architecture as described in the next section.

B. Efficient Integrated Architecture

1) Best-effort Access Area Construction (BE-AAC): While video sources discover their UGs (steps 1-5 in Algorithm 1) and help to form performance groups (steps 6-14 in Algorithm 1), video receivers also issue their REGISTRATION messages to search for plausible gateways. A plausible gateway[5] is a gateway that receives at least one receiver's REGISTRATION message within the one-way delay bound $\bar{D}$. Plausible gateways inform the GM of their existence and the receivers registered with them. Once the ith $(i \in [0, G-1])$ performance group is formed, by Algorithm 2, BS_i constructs the first best-effort access area (BEAA) for performance group i, also called the "benchmark source BEAA" of performance group i.

Algorithm 2 Best-effort Access Area Construction

Input: BS_i

Output: Performance group i's benchmark source BEAA

1. BS_i broadcasts a HELLO message (*as illustrated by the black arrows in Fig. 5*) that records the performance group ID (i.e., i), the BEAA ID [6], and the sending time of this message;

2. Once a WMN node receives a HELLO message, it

[4]The unicast first reaches the coordinator's default gateway via wired links and then the coordinator via WMN links.

[5]Only plausible gateways are eligible to be elected as an area gateway or a corresponding gateway in the following algorithms.

[6]The BEAA ID is 0 as this is the first BEAA of performance group i. This ID is increased by 1 every time a new BEAA is constructed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2691706, IEEE Transactions on Mobile Computing

calculates the delay of the message;

3. If the delay $> \bar{D}$, this WMN node stops forwarding the HELLO message;

4. Otherwise, this node becomes a potential BEAA member of BS_i and continues broadcasting the HELLO message to discover more potential members;

5. **If** the potential member is a gateway or receiver (*e.g.*, G_0, G_1, r_0 *in Fig. 5*), it replies an ACK message to BS_i. The message reserves entries for its forwarders' IP addresses and available throughput;

6. Once the HELLO message is issued, BS_i waits a period $2\bar{D}$[7] for receiving sufficient returned ACKs;

7. If the throughput carried by an ACK message $\geq \bar{T}$, BS_i unicasts a MEMBER packet along the path (discovered by the ACK message, shown by red arrows in Fig. 5) to instruct nodes on the path to join the BEAA.

Algorithm 2 enables a source or an AG to communicate with nodes to select eligible ones (i.e., those nodes which can carry out performance-guaranteed communications with the source or the AG) into its BEAA. Hence, the complexity of Algorithm 2 is $O(m')$, where m' $(\leq N)$ is the number of WMN nodes that exchange control packets with BS_i. In Algorithm 2, for those potential members who are not selected to be BS_i's BEAA members, they become the adjacent nodes of this BEAA. Adjacent nodes play an important role in selecting area gateways (in the next section) and avoiding multicast interference (in Section IV. C. 1). Unlike existing access area construction algorithms (e.g., [10]) that form access areas based on a threshold of wireless hop distances, the BE-AAC algorithm takes the performance of throughput and delays on individual WMN paths into account which ameliorates several drawbacks of previous studies. The threshold of wireless hop distances is normally derived based on the worst WMN link conditions in the multicast system which ignores the fact that most other WMN links are capable of carrying more video data than the assigned load. The BE-AAC algorithm makes full use of heterogeneous capacities on individual WMN links to form wireless transmission paths. In addition, it is costly (in terms of overheads) to use a threshold derived from the worst WMN conditions globally, especially in a large WMN system, while the BE-AAC algorithm only requires the collection of local throughput and delays.

2) MSIM Sub-architectures And Overall Architecture: We refer to the MSIM sub-architecture constructed for video sessions in the ith $(i \in [0, G-1])$ performance group as sub-architecture i. Once the benchmark source BEAA on sub-architecture i is formed, BS_i informs the GM of those receivers that have been included in its BEAA. In order to establish more BEAAs (referred to as "non-source BEAA") to cover receivers located outside the benchmark source BEAA, MSIM selects area gateways (AGs). In the literature, an AG is the closest plausible gateway to the UG or the AG of the latest established access areas (e.g., G_2 in Fig. 5 (a) once the benchmark source BEAA is formed). As delimited by black lines, the access area constructed by G_2 may have a large overlap with the established benchmark source BEAA - an undesirable feature overlooked in previous studies. The consequence is not only overheads caused by forming more access areas to cover all receivers but also the complicated interference topology between access areas. MSIM improves AG selection by the following procedures.

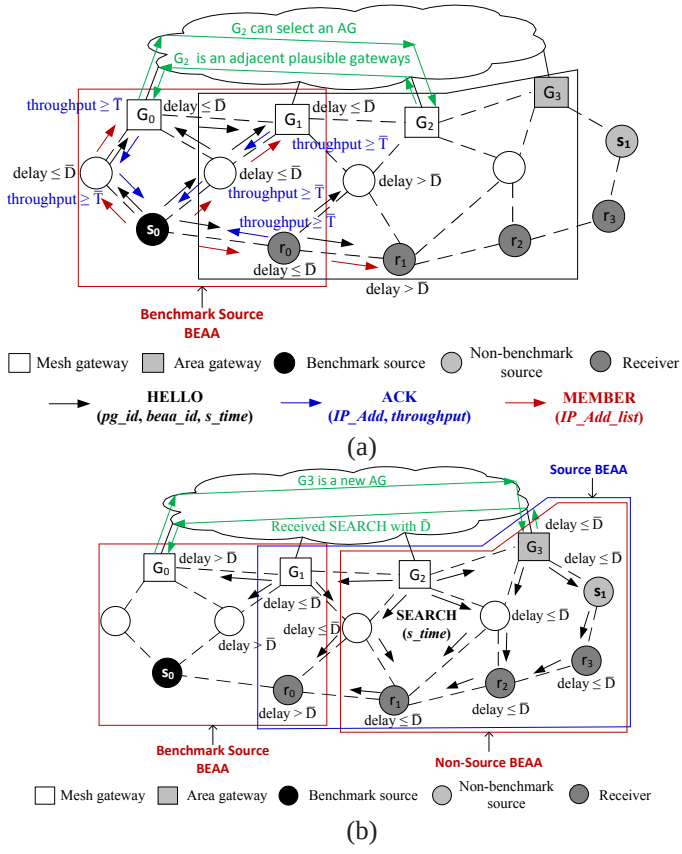


Fig. 5: An example of the EIA algorithm.

During the benchmark source BEAA construction, the adjacent nodes of this BEAA are detected. If an adjacent node is a plausible gateway (e.g., G_2 in Fig. 5), it informs UG_i (i.e., the UG of BS_i) of its existence via wired links. Among all adjacent plausible gateways, UG_i randomly chooses one (e.g., G_2 in Fig. 5) to seek an AG. In detail, the selected adjacent plausible gateway broadcasts a light-weight SEARCH message (via WMN links) which records the sending time of this message, as shown by the black arrows in Fig. 5 (b). Any plausible gateways that receive the message within $\bar{D}$ report to UG_i. Among all reported gateways, the one registered by the greatest number of unallocated receivers will be selected as a new AG. In Fig. 5 (b), shown by green arrows, G_3 is selected in a new AG. The new AG forms a new non-source BEAA by Algorithm 2 but with itself as the input, $(\bar{D} - d_u^i)$ and $(\frac{\bar{T}}{T_u^i})$ as the one-way delay bound and the throughput bound respectively in order to take the uploading delay d_u^i and the uploading throughput T_u^i into account. As such, the BEAA constructed by a farther gateway has much less overlap with existing BEAAs - the non-source BEAA in Fig. 5 (b) does not overlap with the benchmark source BEAA. While this non-source BEAA is constructing, the following two procedures are

[7]This time period is used in order to collect routing paths with guaranteed delays.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2691706, IEEE Transactions on Mobile Computing

in progress in parallel in order to reduce the time consumed in constructing an MSIM sub-architecture.

1) Similarly, UG_i selects more adjacent plausible gateways to seek new AGs to construct more non-source BEAAs for sub-architecture i. These selected adjacent plausible gateways should not have received any SEARCH messages within $\bar{D}$ from previously selected adjacent plausible gateways. When each receiver has been allocated to a benchmark source BEAA or a non-source BEAA, UG_i stops seeking new AGs.
2) For non-benchmark video sessions, the $(g_i - 1)$ non-benchmark video sources form their own BEAAs (referred to as "source BEAAs") in a similar fashion as BS_i. This enables non-benchmark sessions to be delivered via WMN-only paths (to save Internet capacity) to as many receivers as WMN conditions and performance bounds allow. The blue lines in Fig. 5 (b) delimit the source BEAA of s_1.

Within a BEAA, the video source or the AG of this BEAA builds up an interference-controlled multicasting tree (IMT) (see the next section) to multicast videos to receivers located in the BEAA. Between BEAAs, a shared receiver-driven distribution tree (say PIM-SM) is built up through wired links to connect UGs, AGs, and corresponding gateways (CGs)[8] in different BEAAs. At this stage, sub-architecture i is fully constructed. The overall architecture of MSIM includes G sub-architectures which are constructed in parallel by the procedures similar to those employed to construct sub-architecture i. When the construction of G sub-architectures concludes, the MSIM overall architecture is completed.

C. Interference-controlled Multicasting Tree (IMT)

The IMT algorithm, run by the video source in a (benchmark) source BEAA or by the AG in a non-source BEAA, is proposed to multicast video data so as to minimize the impact of wireless interference. When multicasting the I-source video application via the MSIM overall architecture, there may be various sources of interference. Within a BEAA, if multicasting paths are in each other's interference ranges at at least one hop, video transmissions via these paths suffer from interference. Such interference can be efficiently controlled by the WMN multicasting algorithms in the literature. For example, the LCRT algorithm [11] controls such interference by employing the minimum number of nodes that can contribute high transmission capacity to forward multicasting data. In detail, the selection of LCRT forwarders is based on the metric η to choose a node n_j $(j \in [0, N-1])$

$$\eta = D \times \frac{1}{N} \times \frac{C}{\sum_{f=0}^{F-1} r_f}, \tag{3}$$

where D is the number of direct child nodes of n_j, N is the number of nodes within n_j's interference range, C is n_j's transmission capacity, F is the number of existing data flows at n_j, and r_f is the transmission rate of the fth flow. Owing to LCRT's simple procedure yet efficient multicast performance, IMT employs the LCRT algorithm to construct a multicast tree within each BEAA.

Between BEAAs, interference occurs when a multicasting forwarder or receiver hears signals from nodes in other BEAAs. If these signals are unwanted which mostly happens between adjacent BEAAs[9] on the same sub-architectures, IMT employs channel diversity to control such interference; otherwise, if these signals are useful video data which mostly pertains to BEAAs on different sub-architectures, a new session scheduling policy is studied to enhance the multicasting performance of multiple video sessions.

1) Channel Diversity: Suppose the multicasting system has a set of k orthogonal channels $\{c_0, c_1, ..., c_{k-1}\}$. IMT assigns orthogonal channels to (benchmark) source BEAAs with priority. This is because, as compared to a non-source BEAA that forwards videos to a subset of receivers, the transmission performance that a (benchmark) source BEAA can provide affects all multicast receivers. Recall that during BEAA construction (benchmark) source BEAAs can detect that they are adjacent to others, i.e., that they have members that are the adjacent nodes of other (benchmark) source BEAAs. With the information of BEAA adjacency, IMT employs steps 9∼10 in Algorithm 3 (refer to Section IV. C. 3) to assign channels to (benchmark) source BEAAs on the MSIM overall architecture. For non-source BEAAs on the MSIM overall architecture, as with the channel assignment for (benchmark) source BEAAs, they choose channels to multicast if the channels are orthogonal to those being used in their adjacent BEAAs. When the number of orthogonal channels is limited as compared to the number of BEAAs on the overall architecture, IMT employs channels with low overlaps to adjacent BEAAs as the study in [1] has demonstrated the effectiveness of low overlapping channels in controlling interference.

2) The Scheduling Policy: On the overall architecture of MSIM, a forwarding or receiving node receives I sessions via different sub-architectures. Due to limited channel availability, this node often connects to some sub-architectures by the same channel. Suppose the node receives the I sessions of V via k $(1 \leq k \leq I)$ channels. Via the jth $(j \in [0, k-1])$ channel, it receives I' $(I' \leq I)$ sessions from F $(F \leq I')$ forwarders. Meanwhile, the node receives non-V traffic with the burstiness $\hat{\sigma}$ and the average transmission rate $\hat{\rho}$ from the jth channel. We study the following scheduling policy to control interference caused by these simultaneous transmissions on the jth channel.

Theorem 2 *In order to control interference on the jth input channel without sacrificing transmission performance, the node's forwarders on the jth channel should occupy the channel capacity C in turn in such a fashion:*

- *the ith $(i \in [0, F-1])$ video forwarder transmits a period $\tau_i = (\frac{\sigma_i + \rho_i T}{\sigma + \rho T})T$ at time $(nT + \sum_{l=0}^{i-1} \tau_l)$ $(n \in 0 \cup \mathbf{N})$, and*
- *the forwarder(s) of non-V traffic transmits a period $\tau_F = (\frac{\hat{\sigma} + \hat{\rho} T}{\sigma + \rho T})T$ at time $(nT + \sum_{l=0}^{F-1} \tau_l)$,*

where σ_i and ρ_i are the burstiness and the average rate of video session(s) forwarded by the ith video forwarder on the

[8] A CG, selected by the IMT algorithm, is a gateway that collaborates with the AG and other CGs in its BEAA to deliver videos coming from wired links to receivers in the BEAA.

[9] Adjacent BEAAs are those BEAAs that have adjacent multicasting forwarders or receivers.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2691706, IEEE Transactions on Mobile Computing

channel, $\sigma = \hat{\sigma} + \sum_{i=0}^{F-1} \sigma_i$, $\rho = \hat{\rho} + \sum_{i=0}^{F-1} \rho_i$, *and* $\sum_{i=0}^{F} \tau_i = T = \frac{\sigma}{C-\rho}$ *is the schedule period.*

Proof. Denote the accumulated transmission rate of the video sessions received by the multicasting node from the ith forwarder (on the jth channel) as $R_i(t)$ and the transmission rate of the non-V traffic is $\hat{R(t)}$. Then, during a period T, the total amount of input data at this node is $\int_t^{t+T} \hat{R(t)}dt + \sum_{i=0}^{F-1} \int_t^{t+T} R_i(t)dt$. In order to guarantee the timely receiving of all application data at the node, we have

$$\int_t^{t+T} \hat{R(t)}dt + \sum_{i=0}^{F-1} \int_t^{t+T} R_i(t)dt \leq CT, \tag{4}$$

where C is the available capacity of the jth ($j \in [0, k-1]$) input channel of the node.

Let σ_i and ρ_i be the accumulated burstiness and the average transmission rate of video session(s) multicasted by the ith video forwarder via the jth input channel. Based on [12], we have

$$\int_t^{t+T} \hat{R(t)}dt + \sum_{i=0}^{F-1} \int_t^{t+T} R_i(t)dt \leq$$

$$(\hat{\sigma} + \hat{\rho}T) + \sum_{i=0}^{F-1} (\sigma_i + \rho_i T).$$

Hence, expression (4) holds if

$$(\hat{\sigma} + \hat{\rho}T) + \sum_{i=0}^{F-1} (\sigma_i + \rho_i T) \leq CT \tag{5}$$

holds. Expression (5) infers $T \geq \frac{(\sum_{i=0}^{F-1} \sigma_i) + \hat{\sigma}}{C - (\sum_{i=0}^{F-1} \rho_i) - \hat{\rho}}$. By scheduling, the node's forwarders on the jth channel transmit in turn which generates backlog data at these forwarders. In order to transmit backlog data in a timely fashion, we use

$$T = \frac{(\sum_{i=0}^{F-1} \sigma_i) + \hat{\sigma}}{C - (\sum_{i=0}^{F-1} \rho_i) - \hat{\rho}}.$$

The schedule period T is split into $(F+1)$ time slots with the first F slots assigned to the F video forwarders and the last slot reserved for non-V traffic. As for the length of a time slot, it refers to the traffic load required to transmit in the time slot in order to guarantee the throughput performance of all traffic on the jth input channel. Namely, within each T, the time slot assigned to the ith ($i \in [0, F-1]$) video forwarder is

$$\tau_i = \frac{\sigma_i + \rho_i T}{\sigma + \rho T} T, \tag{6}$$

where $\sigma = (\sum_{i=0}^{F-1} \sigma_i) + \hat{\sigma}$ and $\rho = (\sum_{i=0}^{F-1} \rho_i) + \hat{\rho}$. For the length of the last time slot, i.e., the time slot reserved for non-V traffic, it should be

$$\tau_F = \frac{\hat{\sigma} + \hat{\rho}T}{\sigma + \rho T} T.$$

In such a the round robin fashion, the ith video forwarder should transmit at the time $nT + \sum_{l=0}^{i-1} \tau_l$ ($n \in 0 \cup \mathbf{N}$) for a period of τ_i and the non-V traffic can occupy the jth input channel at the time $nT + \sum_{l=0}^{F-1} \tau_l$ for a period of τ_F. **Q.E.D**

By Theorem 2, the scheduling policy for the other $(k-1)$ input channels at the node can be derived. The practical implementation of the schedule policy should take the clock skew between different nodes into account. A number of studies have proposed useful schemes (e.g., the Network Time Protocol) to distributively address the clock skew which helps the IMT algorithm to avoid asynchronous clocks at different nodes. Theorem 2 focuses on single-hop transmissions. In our multi-hop multicast system, forwarders at different hops experience different communication conditions (e.g., different traffic loads) which causes schedules at different hops to fall out of synchronization. A simple but effective way to address such desynchronization is to enable nodes to be consistent in the schedule plan that requires the longest schedule period (denoted as T_{max}) in the system. This is because, if a node employs a schedule plan that has a shorter schedule period than its own schedule period, it cannot output all received data in real time.

3) The IMT Algorithm: The IMT algorithm systematically combines the above proposed approaches to build up an interference-controlled multicasting strategy inside the WMN.

Algorithm 3 Interference-controlled Multicasting Algorithm

Input: BEAAs and their members

Output: The IMT multicasting strategy on the MSIM overall architecture

1. For $i = 0$ to $(G-1)$
2. **For** $j = 0$ to $(B_i - 1)$ // B_i is the number of BEAAs on sub-architecture i
3. A video source (if $BEAA_{i,j}$ is a source BEAA) or an AG (if $BEAA_{i,j}$ is a non-source BEAA) assigns a node level (which equals to the shortest hop distance) to each node in $BEAA_{i,j}$ and sets $l = L - 1$; // $BEAA_{i,j}$ represents the jth BEAA on sub-architecture i, and L is the highest node level in $BEAA_{i,j}$
4. **While** $l > 0$
5. **If** there are receivers or forwarders at level $(l+1)$ that haven't found upstream forwarders
6. The source or the AG selects a l-level node n' that has the maximum η value (see (3)) among all l-level non-forwarding nodes as a l-level forwarder, and updates that n''s direct child nodes have found their forwarder;
7. $l = l - 1$;
8. The source or the AG broadcasts a list of forwarders within $BEAA_{i,j}$ to inform forwarders of their roles;
9. For $i = 0$ to $(I-1)$
10. The ith source selects channels from the set $\{c_0, c_1, ..., c_{k-1}\}$ for forwarders in its BEAA to multicast video sessions. Channels that are orthogonal to those being used in the adjacent (benchmark) source BEAAs are selected with priority;
11. For $i = 0$ to $(G-1)$
12. The AG of a non-source BEAA selects channels from the set for forwarders in its BEAA to multicast video sessions. Channels that are orthogonal to those being used in the adjacent BEAAs are selected with priority;
13. $i = 0$; $j = 0$;
14. While $i \leq (G-1)$
15. **While** $j \leq B_i$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2691706, IEEE Transactions on Mobile Computing

16. Each forwarder or receiver in $BEAA_{i,j}$ encapsulates its schedule plan (calculated by Theorem 2) into a SCHEDULE packet and sends this packet to the node that forms $BEAA_{i,j}$ (i.e., a source or an AG);
17. The source or the AG encapsulates the scheduling plan with the longest schedule period into a BEAA_SCHEDULE packet which is unicasted to UG_i via Internet links;
18 Among the B_i received schedule plans, UG_i encapsulates the one with the longest schedule period into an ARCHITECTURE_SCHEDULE packet. This packet is unicasted to the GM via Internet links;
19. The GM encapsulates the schedule plan with the longest schedule period into an APPLICATION_SCHEDULE packet and multicast the packet to the I video sources and all AGs via the shared receiver-driven distribution trees in the Internet[10].

Algorithm 3 constructs an LCRT tree and assigns channels for each LCRT forwarder within a BEAA. Forwarders on the LCRT tree inform the source or the AG that constructs this LCRT tree of their schedule plans in order to carry multiple video sessions. This means that the complexity of the algorithm follows $O(m'')$, where m'' $(\leq N)$ is the number of nodes in a BEAA. We use an example in Fig. 6 to illustrate the IMT algorithm. Suppose the BEAAs delimited by the red and blue lines belong to the same sub-architecture. As the benchmark source BEAA and the source BEAA overlap, based on steps 2∼3 in Algorithm 3, two different channels are used by the LCRT trees in the two BEAAs (shown by the red arrows and the green arrows respectively). For the non-source BEAA that is adjacent to the benchmark source BEAA and overlaps with the source BEAA, by steps 4∼6 in Algorithm 3, shown by the blue arrows, the employed channel is orthogonal to the channels used in the other two BEAAs. Now, suppose the black lines delimit a BEAA belonging to another sub-architecture. Node 3 and r_0 require to receive videos from both sub-architectures - node 3 via G_2 and G_3, and r_0 via node 2 and s_0 respectively. In order to avoid interference, by step 10 in Algorithm 3, node 3 and r_0 calculate their schedule plans and send their SCHEDULE packets to G_2 (i.e., the AG of their BEAA). If r_0 requires a longer schedule period than node 3, after steps 11∼13 in Algorithm 3, r_0's schedule plan will be eventually adopted by node 3. As shown by the red switching arrows, node 3 and r_0 then receive videos from both sub-architectures in turn without interference.

V. Performance Evaluation

By conducting experimental studies with the discrete event network simulator NS2.33 [13], we compare the following five multicasting schemes in this section.

- RAM: the two-tier integrated multicast in [11];
- EM [9]: the multicast that uses Internet resources via the closest gateway (in terms of IP address prefix) of multicasting nodes;
- IR: a broadcast routing inside a WMN that uses the shortest paths to reach receivers;

[10]Recall that, in Section IV. B. 2, a shared receiver-driven distribution tree is built up for each sub-architecture in order to connect different BEAAs through wired links.

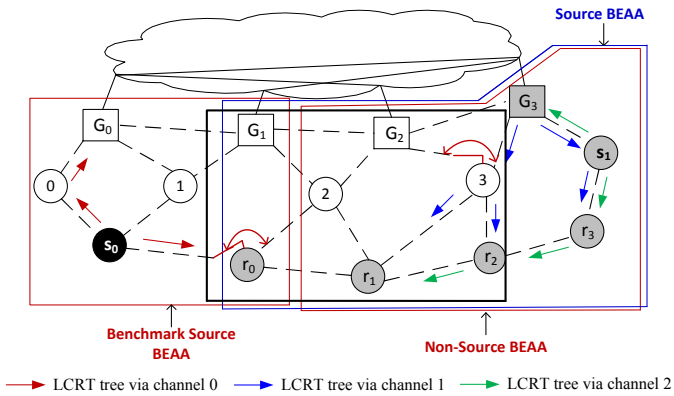


Fig. 6: An example of the IMT algorithm.

- REBS: the multicast that employs only the RESG algorithm in the paper;
- MSIM: the multicast which combining our RE-BSS, BE-AAC and SLCRT algorithms.

Our simulations focus on 1) average multicast delays (AMDs) that evaluate the real time multicast for an I-source video application. An AMD is expressed by $\frac{\sum_{i=0}^{I-1}\sum_{j=0}^{M-1} d_{i,j}}{IM}$, where $d_{i,j}$ is video session i's multicast delay at the jth multicasting node and M represents the number of nodes in the multicast; 2) average multicast throughput (AMT) that evaluates the video representation quality (e.g., resolution) for an I-source multicast. An AMT is expressed by $\frac{\sum_{i=0}^{I-1}\sum_{j=0}^{M-1} T_{i,j}}{IM}$, where $T_{i,j}$ is video session i's multicast throughput at the jth multicasting node. We list the major simulation parameters in Table II. Performance curves in this section are plotted based on the average values of 20 simulation runs.

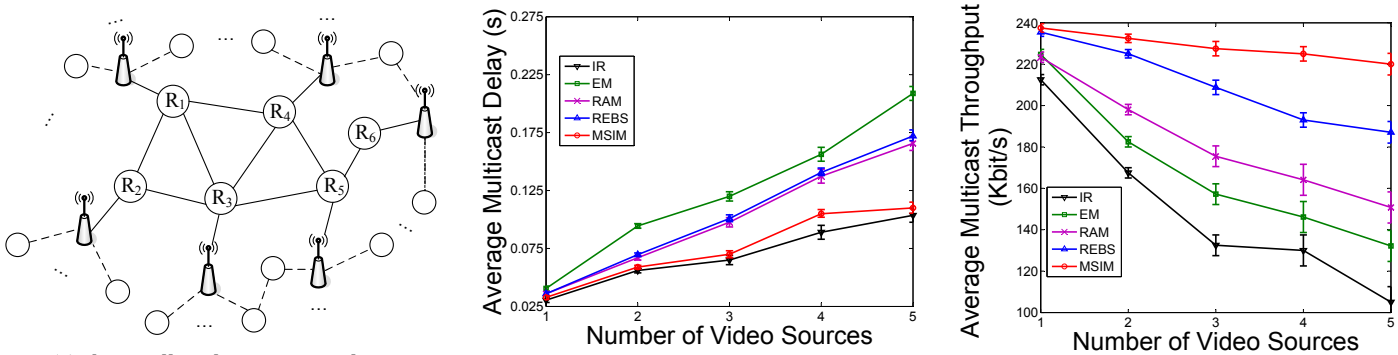
TABLE II: Simulation Parameters

Radio propagation model	Nakagami
MAC protocol	802.11 with 11Mbps data rate
Bandwidth of wired links at routers or gateways	1000Mbps
Transmission range	100m
Simulation time	500s
Interference factor (κ)	$\sqrt{2}$
Group size	Around 30% of WMN sizes
Node distribution density	3 per transmission range
Link loss rate	$[0, 0.2]$

A. Evaluation using a small-scale WMN

1) Impact of the number of video sources on performance: Fig. 7 (a) shows the topology of the simulated small-scale WMN. There are 6 mesh gateways connecting to 6 wired routers to provide Internet access for 50 mesh nodes. The number of video sources varies from 1 to 5 in order to study how this change impacts multicast performance. The video transmission rate is 256Kbit/s. Based on [16], Skype video calls require an upload speed $\geq$ 128Kbit/s. Hence, 256Kbit/s is realistic video rate in the Internet. To create interference or contention, disturbance transmissions are generated in the areas close to the nearest gateways of several video sources. Each

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2691706, IEEE Transactions on Mobile Computing



(a) the small-scale WMN topology

(b) the AMD performance

(c) the AMT performance

Fig. 7: The topology and performance of small-scale WMN simulations.

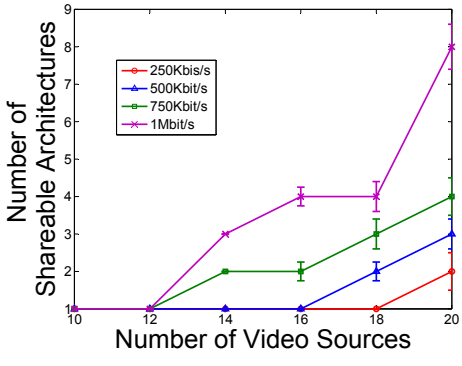


Fig. 8: Number of shareable architectures under different video rates.

of these disturbing transmissions has a constant rate chosen uniformly in the range $[32Kbps, 256Kbps]$.

Fig. 7 (b) presents a comparison of AMDs. EM experiences the longest AMDs as EM users connect to their physically closest gateways to transmit data - some receivers that receive videos from their nearby WMN nodes in RAM, REBS, and MSIM have to go all the way back to gateways to receive data. REBS has slightly longer AMDs than RAM does mainly because REBS architectures are constructed by video sources with the worst uploading performance. MSIM achieves shorter AMDs than RAM does. A major reason for this result is that the formulation of BEAAs enables more WMN nodes to receive videos directly from closer WMN forwarders instead of Internet routers. IR broadcasts videos without forming any architectures which allows it to achieve the shortest AMDs. However, broadcast transmissions in IR cause significant packet loss rates as evidenced by it AMT curve in Fig. 7 (c). This figure also shows that EM achieves a lower AMT than those of RAM, REBS, and MSIM. This is not only because EM does not control interference between video sessions but also because EM does not consider gateway conditions. In the simulations, several gateways are busy with disturbance traffic. REBS runs fewer multicast architectures than RAM which reduces the interference probability between architectures in REBS. MSIM overtakes REBS in AMTs which proves the effectiveness of IMT in controlling interference.

2) Evaluation of MSIM architecture: We then investigate the complexity in constructing an MSIM architecture. We first observe the number of shareable architectures required by MSIM to carry a multi-source video application in the topology of Fig. 7 (a). The bottleneck Internet capacity is set as 20Mbps. The number of video sources increases from 10 to 20 and the video transmission rates vary from 250Kbps to 1Mbps. Fig. 8 shows that, in the simulation with 250Kbps multicast rate, only one shareable architecture is constructed when the number of sources is under 20. For a multicast with a video rate larger than 250Kbps, it only needs a single architecture when the number of sources is under 12 but requires additional shareable architectures when video sources grow. MSIM requires 8 shareable architectures to support 20 video sessions with the transmission rate of 1Mbps. Furthermore, Theorem 1 says that the improvement of Internet capacity helps to control the number of shareable architectures. Our simulation observations meet this theoretical conclusion, e.g., multicasts with 1Mbps rate require 6 shareable architectures when the bottleneck Internet capacity increases to 25Mbps - 2.5% of the total bandwidth of a simulated wired link. Table III gives the number of BEAAs on a shareable architecture when the multicast group size grows from 60 to 150. The number of receivers is $30\% \sim 40\%$ of the group size.

TABLE III: Number of BEAAs on a shareable architecture.

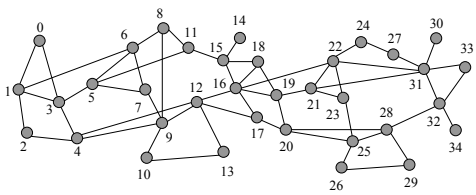
No. of wireless nodes	No. of BEAAs	No. of wireless nodes	No. of BEAAs	No. of wireless nodes	No. of BEAAs
60	3	90	4	120	5
70	4	100	5	130	6
80	4	110	5	140	7

B. Evaluation using a large scale WMN

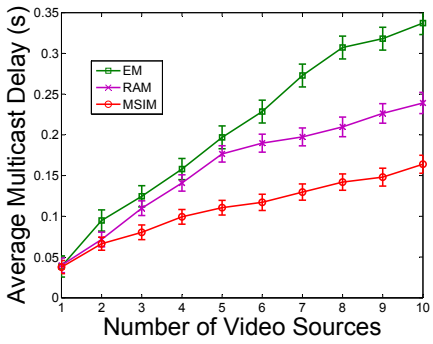
In this group of simulations, we evaluate EM, RAM, and MSIM that multicast the video StarWarsIV.dat in a large-scale WMN[11]. The topology of wired connections (shown in Fig. 9 (a)) has 35 domains (represented by 35 routers). The WMN part consists of mesh nodes distributed across 15 domains via

[11] IR is not evaluated as it is not practical to use IR in a large-scale WMN.

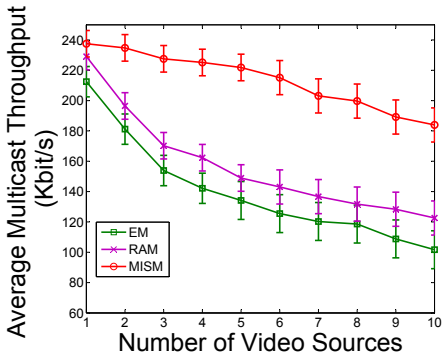
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2691706, IEEE Transactions on Mobile Computing



(a) the topology of wired links in the simulated network



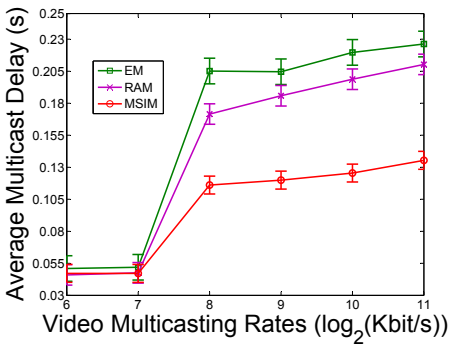
(b) the AMD performance



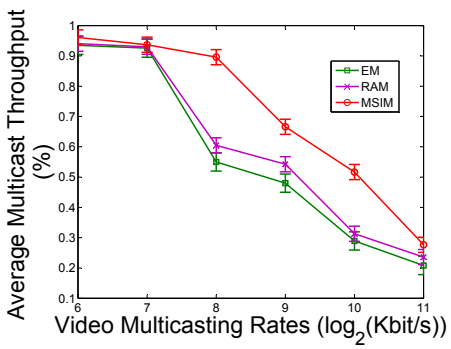
(c) the AMT performance

Fig. 9: The topology and performance of large-scale WMN simulations.

25 gateways. The video trace file StarWarsIV.dat is composed of I, B, P frames that have different sizes. We observe the three schemes when the number of video sources increases, the video transmission rates increase, or the WMN size grows.



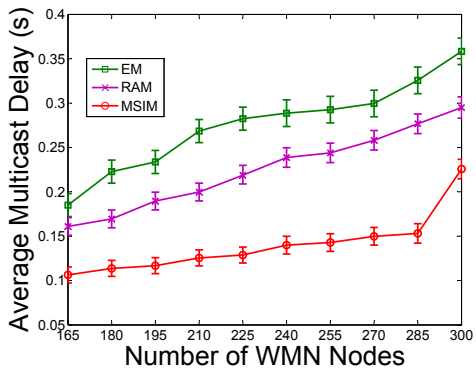
(a)



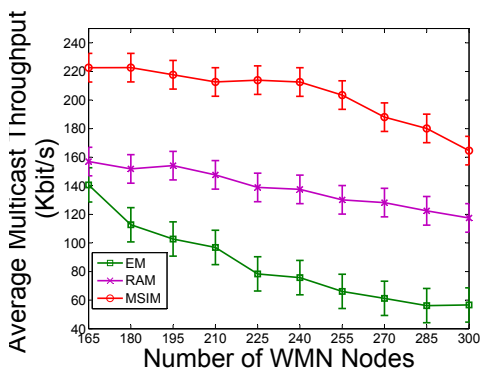
(b)

Fig. 10: Comparison of AMDs (a), and AMTs (b) when video transmission rates increase (the x axis is logarithmic).

1) Impact of the number of video sources on performance: There are 150 mesh nodes in the simulation. Video data are transmitted at 30 frames per second. We vary the number of video sources (from 1 to 10) to observe video performance. According to Fig. 9 (b), as with the results from the small-scale WMN simulations, EM has the longest AMDs as some EM receivers inefficiently receive videos via long-haul back paths.



(a)



(b)

Fig. 11: Comparison of AMDs (a), and AMTs (b) when WMN sizes increase.

MSIM overtakes RAM in AMDs mainly because RESG enables more receivers to connect to nearby wireless forwarders and the use of shareable multicasts reduces overheads and the likelihood of interference. Fig. 9 (c) compares the AMT performance. MSIM achieves an obvious AMT improvement, owing to the greatly controlled interference by IMT and sharing multicast architectures.

2) Impact of video transmission rates on performance: We then vary video frame rates from 8 frames per second

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2691706, IEEE Transactions on Mobile Computing

to 250 frames per second to evaluate the performance of the three schemes. Among the 150 mesh nodes, there are 5 video sources. Based on Fig. 10 (a), the AMDs of the three schemes increase with the increasing video rates. Due to the similar reasons for the AMD comparison in previous simulations, MSIM achieves shorter AMDs than RAM and EM in this simulation. Fig. 10 (b) shows that MSIM cannot have good AMT performance when video rates are larger than 512 Kbps, although its throughput performance is better than EM and RMG. For EM and RGM, they do not scale well when video rates are larger than 128Kbps.

3) Impact of WMN size on performance: At last, we evaluate the three schemes (with the frame rate of 30 frames per second) when WMN nodes grows from 165 to 300. There are 5 video sources. The collected AMDs are plotted in Fig. 11 (a). MSIM guarantees acceptable AMDs when the number of WMN nodes is less than 300, while RAM has unacceptable AMDs when the number of WMN nodes is more than 240 and EM has unacceptable AMDs when the number of WMN nodes is more than 180. The achieved AMT performance is presented in Fig. 11 (b) which demonstrates the obvious throughput improvement of MSIM. This is because of the similar reasons for the results from previous simulations.

VI. Conclusion

This paper designed a novel multicasting framework *multi-source supporting integrated video multicast* (MSIM) for performance-guaranteed multi-source video multicast in large-scale WMN areas. The formation of an MSIM involves using a set of algorithms which together schedule video sessions between shareable integrated architectures. The RESG algorithm selects a greatly controlled number of BSs to construct a controlled number of integrated multicasts (with each serving more than one video session) without violating the constraints of available Internet resources. The EIA algorithm clusters multicasting nodes into BEAAs which extends performance-guaranteed WMN coverage by making full use of the capacity of heterogeneous WMN connections and avoids great overlaps between BEAAs. The IMT algorithm efficiently combines our theoretically-analyzed schedule policy with LCRT trees and channel diversity to control interference on individual sub-architectures or between sub-architectures of MSIM.

Our design principles have been validated by extensive simulation results. Under various important scalability criteria in terms of the number of video sources, video multicasting rates and WMN sizes, the MSIM achieves great performance improvements as compared to the evaluated existing schemes. The MSIM may also support multiple multicasting applications by taking all members in these applications into account to form integrated multicasting paths connecting receivers in different applications to their sources. In this case, in order to avoid delivering packets to receivers belonging to a different application, multicasting nodes such as mesh gateways can be developed to have the ability to filter transmissions by checking packets. For the infrastructure of MSIM, in general, its complexity (e.g., the number of shareable architectures) is proportional to the video traffic load but inversely proportional to the Internet availability, as reported by our simulations. However, such complexity can be easily ameliorated through increasing the Internet availability. With modern Internet technology (e.g., optical fibre Internet [14]), the cheap availability of network bandwidth is promising to allow MSIM to run a few shareable architectures in parallel even when there are a great many video sources in future applications.

References

[1] G. Zeng, B. Wang, Y. Ding, L. Xiao, M. Mutka. Efficient Multicast Algorithms for Multichannel Wireless Mesh Networks. IEEE Trasactions on Parallel and Distributed Systems, vol. 21, no. 1, 2010.

[2] W. Tu. Efficient Resource Utilization for Multi-Flow Wireless Multicasting Transmissions. IEEE Journal on Selected Areas in Communications, volume 30, issue 7, pages 1246-1258, 2012.

[3] J. Qadir, C. Chou, A. Misra, J. Lim. Minimum Latency Broadcasting in Multiradio, Multichannel, Multirate Wireless Meshes. IEEE Transactions on Mobile Computing, vol. 8, no. 11, pages 1510-1523, 2009.

[4] J. Xiong, R. Choudhury. PeerCast: Improving Link Layer Multicast Through Cooperative Relaying. In Proc. of IEEE INFOCOM, pages 2939-2947, Shanghai, China, 2011.

[5] F. Hou, Z. Chen, J. Huang, Z. Li, A. Katsaggelos. Multimedia Multicast Service Provisioning in Cognitive Radio Networks. In Proc. of IWCMC, pages 1175-1180, Cagliari, Italy, 2013.

[6] A. Bhattacharya, R. Ghosh, K. Sinha, B. Sinha. Multimedia Communication in Cognitive Radio Networks Based on Sample Division Multiplexing. In Proc. of COMSNETS, pages 1-8, India, 2011.

[7] S. Lakshmanan, R. Sivakumar, K. Sundaresan. On Multi-gateway Association in Wireless Mesh Networks. (ELSEVIER) Ad Hoc Networks, vol. 7, issue 3, pages 622-637, 2009.

[8] B. Liu, P. Thiran, D. Towsley. Capacity Of A Wireless Ad Hoc Network With Infrastructure. In Proc. of ACM MobiHoc, pages 239-246, Montreal, Canada, 2007.

[9] P. M. Ruiz, F. Galera, C. Jelger, T. Noel. Efficient Multicast Routing in Wireless Mesh Networks Connected to Internet. In Proc. of InterSense, pages 1-10, Nice, France, 2006.

[10] W. Tu, C. Sreenan, C. Chou, A. Misra, S. Jha. Resource-Aware Video Multicasting via Access gateways in Wireless Mesh Networks. In Proc. of IEEE ICNP, pages 43-52, Florida, USA, 2008.

[11] W. Tu, C. Sreenan, C. Chou, A. Misra, S. Jha. Resource-Aware Video Multicasting via Access Gateways in Wireless Mesh Networks. IEEE Transactions on Mobile Computing, vol. 11, issue 6, pages 881-895, 2012.

[12] R. Cruz. A Calculus For Network Delay, Part I: Network Elements in Isolation. IEEE Transactoins on Information Theory, vol. 37, no. 1, pages 114-131, 1991.

[13] UC Berkeley, LBL, USC/ISI, and Xerox PARC. Ns Notes and Documentation, 1999.

[14] N. Bozinovic, Y. Yue, Y. Ren, M. Tur, P. Kristensen, H Huang, A. Willner, S. Ramachandran. Terabit-Scale Orbital Angular Momentum Mode Division Multiplexing in Fibers. Science, vol. 340, no. 6140, pages 1545-1548, 2013.

[15] S. Lim, Y. Ko, C. Kim, N. Vaidya. Design and Implementation of Multicasting for Multi-Channel Multi-Interface Wireless Mesh Networks. (Springer) Wireless Networks, vol. 17, issue 4, pages 955-972, 2011.

[16] Skype. How much bandwidth does Skype need? https://support.skype.com/en/faq/FA1417/how-much-bandwidth-does-skype-need.

[17] S. Shakkottai, X. Liu, and R. Srikant. The Multicast Capacity of Large Multihop Wireless Networks. In Proc. of ACM MobiHoc, pages 247-255, 2007.

[18] N. Hu, L. Li, Z. Mao, P. Steenkiste and J. Wang. Locating Internet Bottlenecks: Algorithms, Measurements, and Implications. In Proc. of ACM Sigcomm, Portland, USA, 2014.

[19] W. Tu. A Multi-rate Multi-channel Multicast Algorithm in Wireless Mesh Networks. In Proc. of IEEE LCN, pages 55-63, Edminton, Canada, 2014.

[20] Y. Chang, Q. Liu, and X. Jia. A Data Rate and Concurrency Balanced Approach for Broadcast in Wireless Mesh Networks. IEEE Transactoins on Wireless Communications, vol. 13, issue 7, pages 3556-3566, 2014.