Resource-Aware Video Multicasting via Access Gateways in Wireless Mesh Networks

Wanqing Tu, *Member*, *IEEE*, Cormac J. Sreenan, *Member*, *IEEE*, Chun Tung Chou, *Member*, *IEEE*, Archan Misra, *Member*, *IEEE*, and Sanjay K. Jha, *Senior Member*, *IEEE*

Abstract—This paper studies video multicasting in large-scale areas using wireless mesh networks. The focus is on the use of Internet access gateways that allow a choice of alternative routes to avoid potentially lengthy and low-capacity multihop wireless paths. A set of heuristic-based algorithms is described that together aim to maximize reliable network capacity: the *two-tier integrated architecture* algorithm, the *weighted gateway uploading* algorithm, the *link-controlled routing tree* algorithm, and the *dynamic group management* algorithm. These algorithms use different approaches to arrange nodes involved in video multicasting into a clustered and two-tier *integrated* architecture in which network protocols can make use of multiple gateways to improve system throughput. Simulation results are presented, showing that our multicasting algorithms can achieve up to 40 percent more throughput than other related published approaches.

Index Terms—Multicasting, resource awareness, wireless mesh networks, large-scale integrated routing, video streaming.

1 Introduction

Video streaming represents one of the fastest growing segments of traffic in the Internet today. Multicasting of video over wireless networks is a challenging problem, due to the combination of high data rates (relative to wireless capacity) and low-latency constraints and the need to support multiple receivers with time-varying link quality. Wireless mesh networks (WMNs) offer an attractive solution for low-cost connectivity over large urban areas. A WMN consists of a set of mesh nodes offering connectivity to end user devices; the mesh nodes form a relatively static, multihop wireless backbone. Supporting video applications (e.g., wireless video television distribution, large-scale video conferencing, urban traffic management, and multimedia distance learning) in such mesh environments is particularly challenging, given the observed loss in network throughput that occurs when packets traverse multiple successive wireless hops. Accordingly, to support video multicasting over WMNs, we need to maximize the use of available network resources.

Various papers, e.g., [11], [12], [13], have examined the use of multiple wireless transmission rates, multiple radio interfaces, multiple paths, and wireless broadcast advantage to improve wireless network capacity. These approaches are *intramesh* schemes, in that they aim to support video transmission in WMNs by using wireless resources more effectively. An alternative approach is to enhance WMNs with access gateways that can provide alternative routing paths via the Internet. This allows a reduction in wireless transmission distance (the number of wireless hops traversed) and hence an improvement in residual capacity. We call this gateway-based approach the *integrated* architecture and illustrate its potential in Fig. 1. The *intramesh* communication from S to R experiences at least six hops, while the *integrated path* (shown by the arrow lines) traverse only two wireless hops (to or from the gateways) by taking advantage of an *Internet shortcut* among the gateways. Besides reducing the hop count, an additional set of advantages accrue from the higher bandwidth and lower loss rates that the Internet (wired) paths offer, compared to the WMN wireless links.

Gateway-based approaches [8], [9] for routing in WMNs typically aim to improve forwarding performance over large physical distances (possibly across multiple network domains), by using wired paths preferentially over wireless links whenever possible, without regard to the quality and congestion experienced by different links. However, in many instances, the use of an Internet-based path may actually prove counterproductive, especially if the vicinity of the access gateways is congested. In Fig. 1, packets from S prefer the *intramesh* routing to reach R' because the *intramesh* routing has the same number of wireless hops as the *integrated path* but needs no Internet access. Hence, the choice between an *intramesh* route versus an *integrated path* is an involved one, that must clearly take into account

- *W. Tu is with the School of Computing and Communication Technology, PP14, Plas Coch Campus, Glyndŵr University, Mold Road, Wrexham LL11 2AW, United Kingdom. E-mail: w.tu@glyndwr.ac.uk.*
- *C.J. Sreenan is with The Mobile & Internet Systems Laboratory, Department of Computer Science, University College Cork, IT Building 1-075, Western Road, Cork City, Ireland. E-mail: cjs@cs.ucc.ie.*
- *C.T. Chou and S. Jha are with the School of Computer Science and Engineering, The University of New South Wales, Sydney, NSW 2052, Australia. E-mail: {ctchou, sanjay}@cse.unsw.edu.au.*
- *A. Misra is with the School of Information Systems, Singapore Management University, 80 Stamford Road, Singapore 178902. E-mail: archanm@smu.edu.sg.*

Manuscript received 30 June 2010; revised 26 Mar. 2011; accepted 8 Apr. 2011; published online 29 Apr. 2011.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2010-06-0322.
Digital Object Identifier no. 10.1109/TMC.2011.103.

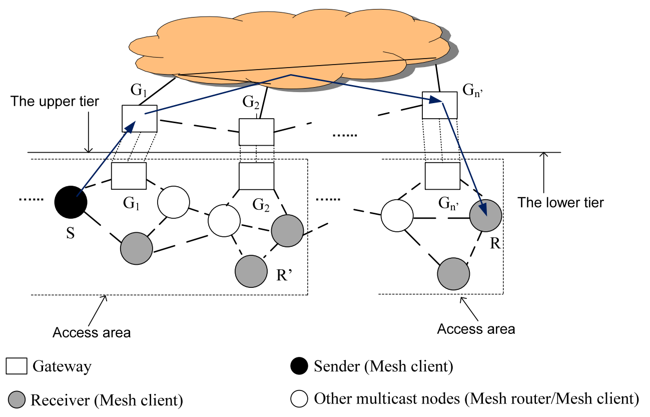


Fig. 1. An example of the *two-tier integrated architecture*.

relative position of the sending and receiving nodes, and more importantly, the traffic congestion and link quality on both the *intramesh* and *integrated* paths.

Based on these observations, this paper develops a video multicasting framework for large-scale environments that exploits the *combination* of available Internet resources and intra-WMN wireless bandwidth. To start, we account for the degradation in video transmission that arises from its transfer over multiple successive wireless links within the WMN. This observation forms the basis for our *resource-aware multigateway WMN video multicasting* scheme, which uses a set of interlinked novel algorithms to construct *integrated* multicast routes that maximize the network's capacity for video traffic that is sensitive to quality of service (QoS) requirements such as end-to-end delays and throughput. Our multicasting framework improves our previous work [26] on resource-efficient and reliable video multicasting by reducing the associated signaling overheads and by better utilizing the dynamically changing available capacity on different wireless paths. To construct this framework, the following algorithms are presented.

- *The two-tier integrated architecture* (TIA) algorithm establishes a hybrid wired-wireless routing hierarchy. To avoid the use of an excessively large number of wireless hops, TIA (unlike the schemes in [8], [9]) employs a threshold on the number of wireless hops to initially cluster WMN gateways and mesh nodes into different lower-tier *access areas*. To connect different *access areas* for video delivery to all group receivers, TIA (in contrast to [26], which employs broadcasting to find gateways and thus is limited to operation within a single domain) uses a receiver-driven multicast protocol to establish a distribution tree connecting gateways in different Internet domains.
- *The weighted gateway uploading* (WGU) algorithm is used by a sender in its *access area* to select one among multiple candidate gateways for directing video via the Internet, so as to ensure efficient and high throughput video routing. Instead of using the static hop distance as the only metric for gateway selection [26], the enhanced WGU uses a metric that balances the traffic load of a gateway, the path reliability and the delay distance from the gateway to the sender.
- *The link-controlled routing tree* (LCRT) algorithm builds a multicast tree, inside each *access area*, that decreases the interference-induced delays within the WMN while guaranteeing the highest possible transmission throughput. In addition to exploiting wireless broadcast advantage by constructing a forwarding tree that minimizes the number of mutually interfering forwarding nodes [26], the enhanced LCRT algorithm presented here also seeks to use better quality links for reliable video multicasting.
- *The dynamic group management* (DGM) algorithm maintains the multicasting framework with controlled overheads when dynamic changes take place. New members are admitted to the group through a short and reliable path. Multicast interruption is recovered by using interference-limited routing paths.

The rest of the paper is organized as follows: Section 2 assesses related previous work. Section 3 analyzes the problem of degradation of video quality in large-scale wireless mesh networks. Section 4 then presents our *resource-aware multigateway WMN video multicasting* solution. Computer simulations and evaluation are detailed in Section 5. Finally, Section 6 concludes the paper.

2 Related Work

Research in the area of video multicasting using wireless mesh networks can be classified as being either *intramesh*, where the focus is on the optimization of wireless links and interfaces, or *integrated* where the use of Internet access gateways is assumed.

Intramesh video multicasting utilizes modern wireless techniques such as multiple rate transmission, multiple channels, wireless broadcast advantage, etc. Liu et al. [11] proposed the *Rate and Contention Aware Multicast* (RCAM) scheme that exploits link-rate diversity to construct a multicast forwarding tree, based on the link transmission rates and the associated congestion load expressed via a *cumulative transmission time fraction (CTTF) metric*. For efficient wireless broadcasting, Chou et al. [12] suggested the use of a metric that optimizes the product of the link rate and the coverage area. Subsequently, Wang et al. [13] proposed a broadcast tree construction algorithm that reduces start-up delays, while exploiting the relationship between transmission rates and their coverage range. Apart from these papers, it is analytically well known [2] that the per-node multicast throughput of a random multihop network with n nodes, n_s multicast sources, and n_d destinations is fundamentally bounded by

$$O\left(\min\left(1, \frac{\sqrt{n}}{n_s\sqrt{n_d}\log n}\right)\right)$$

with a high probability, implying that the multicast throughput is a decreasing function of the size of the network. This motivates our design goal of limiting the depth of a wireless multihop path.

Integrated wireless transmission has been studied by Liu et al. in [24]. This paper analyzes unicast flows and shows that using Internet shortcuts would significantly increase network capacity. For *integrated* multicasting in WMNs, Ruiz et al. [8] proposed a routing mechanism where mesh nodes connect to their "closest" gateway by the procedures described in [25]. Mesh nodes that form an "island" with prefix continuity connect to the Internet through a shared "closest" gateway. The selection of gateways is based purely on topology and fails to consider the tradeoff between the selection of a closer (i.e., less hops) but more congested gateway versus the use of a farther, less utilized gateway, especially when the end-to-end path consists of additional intra-WMN hops. Amir et al. [9] presented a hybrid routing protocol for multihomed wireless mesh networks that provides uninterrupted connectivity and fast handoffs, rather than load-based multicast dissemination. In general, these papers do not consider load-balanced access via the Internet and use wired resources irrespective of the relative merits of intramesh versus gateway-based paths. In this paper, we present a multistep *integrated* procedure that makes a judicious use of Internet resources while cooperatively sharing intra-WMN wireless bandwidth for single-source video multicasting applications.

3 Problem Formulation

For a wireless mesh network, suppose that a set of n nodes participates in multicasting a given video flow. The set of n nodes, including G gateways (g_i, $i \in [0, G-1]$) and M mesh nodes ($m_j, j \in [0, M-1]$), can be denoted as $U = \{g_0, g_1, \ldots, g_{(G-1)}, m_0, m_1, \ldots, m_{(M-1)}\}$. That is, $n = G + M$. More specifically, the G gateways play different roles in our multicasting architecture.

- Corresponding gateway: a gateway that joins in a video multicasting session V.
- Area gateway: a gateway that creates a new *access area*, besides multicasting a video session V.
- Uploading gateway: a gateway, located in the *access area* of a multicast source, that forwards a video session V to remote group members through the Internet. The *uploading gateway* also selects *area gateways* for different *access areas*.

The M mesh nodes are composed of mesh routers and mesh clients, and play different roles below in our multicast architecture.

- Multicast group members are mesh nodes that either send a video flow V and/or are receivers of a multicast video flow V.
- Intermediate nodes are mesh nodes that are involved in constructing the hierarchical multicast architecture, i.e., they include the mesh nodes within an *access area*, irrespective of whether or not they become active forwarders on multicast traffic distribution trees.

For brevity, we will use the term "nodes" to refer to any elements in U. Nodes that are selected to relay a video session V to receivers through multicast trees are called multicast forwarders in this paper. Table 1 lists the major symbols used in this paper.

To allow video multicasting over a large-scale WMN without suffering unacceptably high degradation, we first capture the degradation of wireless video signals during transmission (called wireless video communication cost), in terms of either throughput cost or delay cost.

TABLE 1
Symbol List

U	Set of nodes participating in multicasting a video session V
s	Sending source of a video session V
n	The number of nodes (including gateways and mesh nodes) in U
G	Number of gateways in U
M	Number of mesh nodes (group members and intermediate mesh node) in U
g_i	The ith ($i \in [0, G-1]$) gateway in U
m_j	The jth ($j \in [0, M-1]$) mesh node in U
K	Threshold of the number of wireless hops that guarantees the quality of wireless transmission
k	Number of wireless hops from s to its *uploading gateway*
$r_V(bit/s)$	Transmission rate of a video session V
$(r_V)_{basic}$ (bit/s)	Transmission rate of the lowest-acceptable video quality of V
L	Number of wireless links on path
P	Number of packets transmitted
$C_l(bit/s)$	Capacity of link l
ρ	Average distribution density of wireless nodes in the multicasting system of V
d	Average distance of one wireless hop in the multicasting system of V
κ	Interference factor of wireless channels in the system
ℓ	The highest link loss rate in the system
w_i	Weight of the ith ($i \in [0, G']$) gateway in the source *access area* with G' gateways (WGU algorithm)
η_j	Weight (defined in (3)) of the jth ($j \in [0, n'-1]$) node in an *access area* with the size of n' (LCRT algorithm)

- Throughput cost is defined as the difference in the transmission data rate of a source and the reception data rate by a multicast receiver node. This difference may occur as a result of either buffer overflow at multicast forwarders or transmission loss (due to interference or link quality degradation) on forwarding links.
- Delay cost is defined as the additional end-to-end latency that results from buffering by nodes in order to resolve link contention prior to subsequent transmission of a packet.

Fundamentally, the improvement of throughput cost and delay cost depends on factors such as the number of wireless links (L) in the path, the minimum capacity of the L links, and the capacity lost by interference and errors at each link. For throughput cost, reducing the wireless hop count increases video multicasting capacity because the decreased L reduces the interference-induced packet loss. Previous proposals [19], [20], [21] extend wireless communication range by minimizing L (choosing shortest paths) within wireless ranges, but these proposals can only extend the range by a few more wireless hops, which is of limited benefit. Hence, our first objective is to use Internet shortcuts to limit the total number of wireless hops traversed by the video traffic.

Besides simply reducing the total number of wireless hops, we must also minimize the capacity loss and delays on each link caused by transmission interference/contention and link unreliability. In multihop wireless networks, link loss is very common and contention can arise because multicast transmissions require more than one node within an interference area to forward multicasting traffic. Fig. 2 illustrates a possible situation where the multicast transmissions of sibling nodes can interfere with each other's reception. The simultaneous transmissions $A \rightarrow C$ and $B \rightarrow D$ generate redundant mutual transmissions

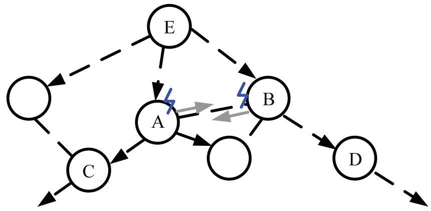


Fig. 2. An example of interference when multicasting video streams.

between A and B as shown by the gray arrow lines between them, which interfere with the reception of A and B from their upstream node E, as illustrated by the sibling interference flashes at A and B. Hence, our second objective is to decrease the capacity loss by employing reliable wireless links while minimizing the number of potentially interfering packet transmissions.

Finally, to reduce the overall throughput and delay costs, it is also necessary to ensure that the "bottleneck link" (the one with the minimum capacity on the forwarding path) possesses a large and reliable residual bandwidth. This is especially important in *integrated* wireless communications, where gateways connecting to the Internet are likely to become "bottlenecks" because they carry both ingress and egress traffic between the mesh nodes and the Internet. Hence, our third objective is to avoid the use of "busy" gateways. In the next section, we shall present our multitier algorithmic approach that addresses each of these three objectives.

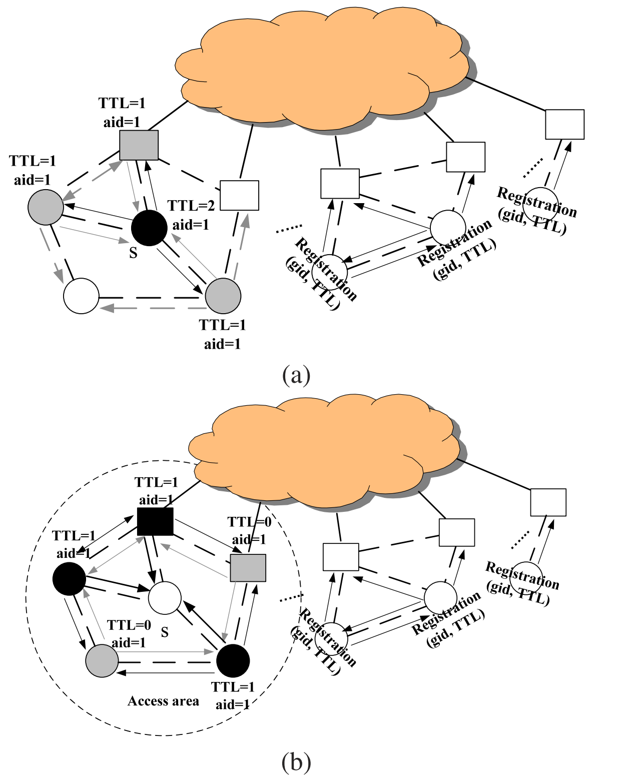
4 Resource-Aware Multigateway WMN Video Multicasting

In this section, we present the four key elements of the *resource-aware multigateway WMN video multicasting* scheme:

- **Two-tier integrated architecture.** The TIA constructs *access areas* interconnected by a wired network. Nodes are assigned to an *access area*, so as to bound the number of wireless hops to a gateway, in a way that assures acceptable QoS.
- **Weighted gateway uploading.** The WGU algorithm is used by a multicast source node to select a gateway node within its access area based on load levels, path reliability, and hop count.
- **Link-controlled routing tree.** The LCRT algorithm routes traffic within each *access area* so as to reduce interference and channel contention while achieving a consistently higher throughput.
- **Dynamic group management.** The DGM algorithm deals with group membership and recovery from transmission interruptions.

4.1 Two-Tier Integrated Architecture

As previously discussed, distant users are likely to benefit from employing an Internet shortcut, while proximal users would likely prefer intramesh routing. The TIA is designed to facilitate the selection of paths, keeping in mind the differences in objectives among different receivers. As illustrated in Fig. 1, all of the nodes join in the lower tier where they are separated into *access areas*. From this point on, we call an *access area* in which the video source resides as a source *access area*; an *access area* in which no video source resides is referred to as a nonsource *access area*. In a given source *access area*, there is one *uploading gateway* that is selected by the source s using the *weighted gateway uploading* algorithm; in a given nonsource *access area*, there is one *area gateway* that is selected based on a defined metric among gateways that have registered with group receivers.



(a)

(b)

Fig. 3. An example of the *access area construction*. Square nodes represent gateways. Circle nodes represent mesh routers/mesh clients. Black nodes and gray nodes are the senders and the receivers of AREA_CONSTRUCTION, respectively.

4.1.1 Access Area Construction

The *access area construction* algorithm clusters nodes into individual *access areas*, such that the hop distance between nodes within an *access area* does not exceed a maximum permitted threshold:

- $2K$ for a source *access area*,
- $2(K - k)$ for a nonsource *access area*,

where k is the hop distance between the source s and its *uploading gateway*. The value of K for practical applications will be analyzed later in this paper.

We now explain the *access area construction* by introducing the formation of the source *access area* which requires two key messages: AREA_CONSTRUCTION and JOIN_REPORT, illustrating with the example in Fig. 3. To start the construction, the source s of the video flow V broadcasts an AREA_CONSTRUCTION request packet, with $TTL = K$ (in the example, $K = 2$), that includes three fields:

1. *area id* of this constructing *access area*,
2. *address* of the packet sender/forwarder, and
3. packet's TTL which is initially set to K, the maximum permitted number of wireless hops.

Nodes that receive AREA_CONSTRUCTION (i.e., the gray nodes in Fig. 3a) use the *area id* to set their own ID, and then rebroadcast the message (after decrementing the TTL field by 1) if it was arrived with a higher TTL than any previously received message copy (as illustrated by the black arrow lines in Fig. 3b). In this way the AREA_CONSTRUCTION is flooded over the K hop neighborhood to construct the source *access area*. The source *access area* is the dotted circle area in Fig. 3b.

Nodes also send JOIN_REPORT messages to s, as illustrated by the gray arrow lines in the source *access area* of Fig. 3. This message provides their *node types* (i.e., gateways or mesh nodes), *addresses* and *hop distances* to s, allowing it to discover nodes that are within a K-hop intra-WMN neighborhood. A JOIN_REPORT is broadcast to s and each intermediate node forwards the same JOIN_REPORT only once. The primary motivation for the use of broadcast is to search for *corresponding gateways* in a nonsource *access area* (Section 4.3), but it also provides the reliable delivery of JOIN_REPORT to source s.

For JOIN_REPORT broadcasting in a nonsource *access area*, a gateway receiving the message contacts its *area gateway* directly through the Internet instead of using wireless broadcasting. This significantly improves the overhead performance of our previous algorithm [26].

4.1.2 Other Gateway-Initiated Access Areas

When the source *access area* is constructed, s uses the *weighted gateway uploading* algorithm to select an *uploading gateway* from all of the *corresponding gateways* in the source *access area*. This procedure is illustrated in Fig. 3b in which the black square represents s's *uploading gateway*. The *uploading gateway* is responsible for selecting *area gateways* among plausible gateways for nonsource *access areas* before it distributes V to group receivers in nonsource *access areas* through wired connections. Plausible gateways are the gateways that can forward the video flow V to the receivers without violating the "the total number of wireless links $< K$" constraint, i.e., gateways that lie within K hops of at least one multicast receiver. As indicated by the arrows outside the source *access area* in Fig. 3a, plausible gateways are searched by multicast receivers through sending registration packets (potentially asynchronously) with the *group id*.

The procedure to select *area gateways*, as illustrated by the arrow lines in Fig. 4a:

- Plausible gateways request s for an *uploading gateway*[1] first. Requests are routed via the Internet to s's default gateway, which passes these requests to s.
- On receiving a request, s replies with the *uploading gateway*'s IP address and the value of $(K - k)$ (which is 1 in the example of Fig. 4) by issuing a reply back along the reverse path.
- Hereafter, plausible gateways run a receiver-driven multicast protocol (say PIM-SM) to establish a distribution tree rooted at the *uploading gateway*.

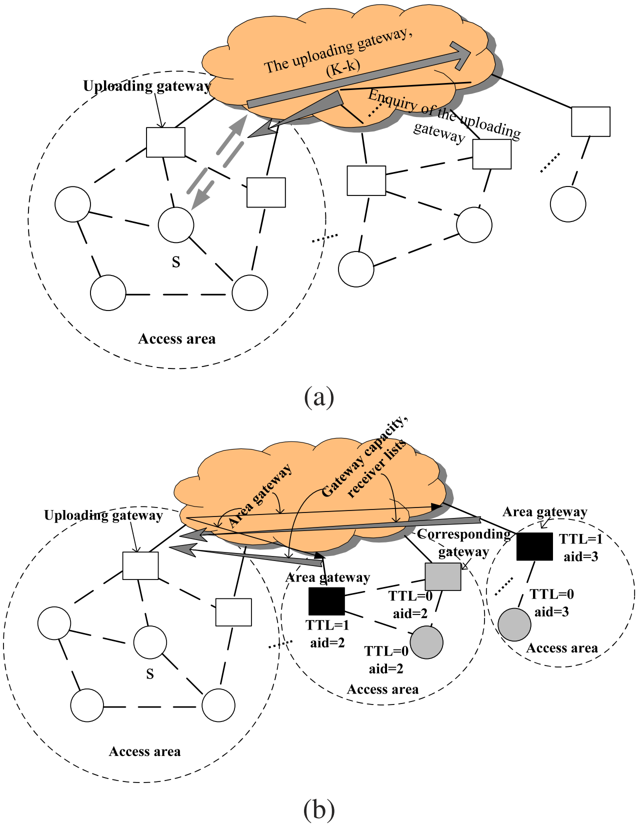


Fig. 4. An example of constructing a *two-tier integrated architecture*. Square nodes represent gateways. Circle nodes represent mesh routers/mesh clients. Black nodes and gray nodes are the senders and the receivers of AREA_CONSTRUCTION, respectively.

This latter procedure is illustrated by the big gray arrow lines in Fig. 4b. When building the receiver-driven tree, the gateways also inform the *uploading gateway* of their:

- *available wireless transmission capacity*,
- *lists of registered receivers*, and
- *hop distances* to the registered receivers.

The *uploading gateway* then selects *area gateways* based on the $\frac{d}{o}$ values of gateways, where d is the gateway's dynamic distance (e.g., delay distance) to the *uploading gateway* and o is the available (residual) wireless transmission capacity of the gateway.

The first *area gateway* to be selected is the one that has the minimum $\frac{d}{o}$ value. Using the Internet, the *uploading gateway* informs this selected *area gateway* of its role and uses it as a base node to choose additional gateways for this *access area* that have registered multicast receivers that are within the $(K - k)$-hop coverage of the *area gateway*. Subsequently, the second selected *area gateway* is that with the minimum $\frac{d}{o}$ value among the gateways that are not in a nonsource *access area*. The procedure continues until *area gateways* can cover all receivers. Selected *area gateways* then implement a similar process to that of s, except for using the diameters of $2(K - k)$, to construct nonsource *access areas*. In Fig. 4b, *area gateways* (labeled by black squares) send the AREA_CONSTRUCTION packets with $TTL = 1$ because $k = 1$ in this example. This proposed construction algorithm allows *area gateways* to form *access areas* in parallel, and is thus faster than the sequential algorithm originally presented in [26].

[1] The IP address of the video sender is published with the *group id* by the sender s itself.

When the construction of nonsource *access areas* concludes, the two-tier architecture is completed. Video multicasting in the upper tier is implemented by the *uploading gateway* through the receiver-driven distribution tree. *Area gateways* and *corresponding gateways* (selected by the LCRT algorithm in Section 4.4) multicast V within their *access areas* through *link-controlled routing trees* once receiving V from the *uploading gateway*.

4.1.3 Discussion

We use *the threshold of the number of wireless hops* (K) to provide an initial static approach to avoiding paths with an excessive number of wireless links. This section studies how to decide the threshold K initially.

Denote the average density of transmitting multicast nodes in our WMN system as ρ and the average distance of one wireless hop as d. Then, the average interference range of each sender/forwarder is κd, where $\kappa > 1$ is a factor induced to express that an interference range is usually larger than a transmission range. Hence, the average number of nodes in an interference range is $\rho\pi(\kappa d)^2$.

To calculate the transmission throughput that a K-hop path can provide, the bottleneck capacity on this path should be considered. Since a wireless channel can only be used by one node in an interference range in order to avoid interference, a bottleneck link on the K-hop path will be a link such that all nodes in the same interference range are transmitting data. Under these circumstances and for simplicity, assuming an ideal MAC layer protocol that provides channel access equally to interfering nodes, the bottleneck capacity is then $\frac{C}{\rho\pi(\kappa d)^2}$, where C is the capacity of individual wireless links in the system. Note that nodes in the same interference areas may not always need to send data simultaneously and therefore there may be a higher bottleneck capacity. However, to make sure that the threshold can guarantee acceptable performance, we use a bottleneck capacity of $\frac{C}{\rho\pi(\kappa d)^2}$ to calculate K. Denote the sending rate of V as r_V. The receiving rate at the receiver on this K-hop path should be, without considering link loss, $\min\{r_V, \frac{C}{\rho\pi(\kappa d)^2}\}$.

We now consider how link loss affects K. Denote the loss rate of the ith link as ℓ_i. The receiving rate at the receiver then becomes $r_V^K = \min\{r_V, \frac{C}{\rho\pi(\kappa d)^2}\}\prod_{i=1}^{K}(1-\ell_i)$. Suppose $\ell = \max\{\ell_i, i \in [1, K]\}$. Then, we have

$$r_V^K \geq \min\{r_V, \frac{C}{\rho\pi(\kappa d)^2}\}(1-\ell)^K.$$

Since r_V^K should be at least higher than rate $(r_V)_{basic}$, to ensure the lowest-acceptable video quality, the expression $\min\{r_V, \frac{C}{\rho\pi(\kappa d)^2}\}(1-\ell)^K \geq (r_V)_{basic}$ should be satisfied. Accordingly, it follows that the maximum number of acceptable wireless hops K must meet

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

Expression (1) shows, in practical systems, K is determined by the choice of radio interfaces/channels, the node distribution, the video data rate, and the QoS desired by an application. A channel with a larger transmission neighborhood (i.e., a larger d) implies a larger number of contending transmissions. A dense distribution of wireless transmission nodes (i.e., a larger ρ) means intensive interference. An application requiring a higher $(r_V)_{basic}$ can tolerate a smaller value of K. All of these three situations require smaller *access areas* in order to reliably achieve acceptable video performance. Wireless throughput could be more complicated in practical WMN systems. The above calculation can always be employed to decide the initial value of K. K may be changed later on if some nodes consistently experience unacceptable video performance. These unsatisfied nodes send LEAVE packets including their IDs (e.g., IP addresses) to their forwarders,[2] and then join other *access areas* (existing or newly created by a new gateway) according to the algorithm in Section 4.4, along with any downstream forwarders/receivers that they may have.

4.2 Weighted Gateway Uploading

The *WGU* selects an *uploading gateway* through which a source can send its video data to group receivers in nonsource *access areas*. In our previous work [26], a static metric—the gateways' distances to s was used to choose such an *uploading gateway*. This new algorithm takes the reliability of the connection between the *uploading gateway* and the sender into account because group receivers in nonsource *access areas* depend on this connection to receive video traffic. More specifically, the WGU algorithm assigns each gateway in the source *access area* a weight which is a function of the gateways' available capacity and their reliability to connect to the sender. The gateway with the largest weight is selected as the *uploading gateway*. The detailed algorithm procedure is described below.

During the construction of the source *access area*, each gateway piggybacks a gateway report into its JOIN_REPORT when replying to s's AREA_CONSTRUCTION. The gateway report includes three fields:

- *IP address* of the gateway,
- *available capacity* of the gateway's wireless link which is assessed by the gateway based on the transmission rate that it currently provides, and
- *reliability* of the path employed to send this report.

The *reliability* $r_{i,j}$ of a path j from the ith gateway to s is the product of the *reliability* of all L links on the path. Namely,

$$r_{i,j} = \prod_{j'=0}^{L-1}(r_{i,j'}), i \in [0, G'-1],$$

where $r_{i,j'} = (1-\ell_{i,j})$ is the *reliability* of the j'th wireless link on the path j, $\ell_{i,j}$ is the loss rate of the j'th wireless link, and G' is the number of gateways in the source *access area*. s might receive multiple JOIN_REPORT messages from a gateway but via different paths because of wireless broadcast transmission. It selects the path with the highest R value as the best path to reach this gateway. The R value of the ith gateway is

[2] An on-tree forwarder stops multicasting (to reduce traffic load and interference) when it receives LEAVE packets from all of its direct child nodes.

$$R_i = max\left\{\frac{r_{i,j}}{d_{i,j}}, i \in [0, G' - 1], j \in [0, J_i - 1]\right\},$$

where J_i is the number of JOIN_REPORT received by s from the ith gateway, and $d_{i,j}$ is the time delay between s sending AREA_CONSTRUCTION and s receiving the jth JOIN_REPORT. This equation generally guarantees that each gateway reliably connects to s through a short delay path.

To avoid a long delay to decide the *uploading gateway*, after a period of $2 \times T^3$ since s receives the first JOIN_REPORT, s starts to select the *uploading gateway* based on the gateways' weights. The weight of the ith gateway w_i is

$$w_i = C_i \times R_i, i \in [0, G'' - 1], \tag{2}$$

where $G'' \leq G'$ is the number of gateways from which s received their replies, and C_i is the ith gateway's available capacity. Equation (2) indicates that the selection of *uploading gateway* uses a load-reliability balanced metric to find a "nonbusy" and close gateway that can receive stable wireless transmission from s, thereby avoiding "bottleneck" gateway nodes.

Although the selected *uploading gateway* may not remain the best in terms of its weight value calculated by (2) because of dynamic network conditions, the *uploading gateway* is not changed dynamically due to the cascading effect of this change on the TIA algorithm: resulting in high signaling load and reconfiguration delays. Over the long term and across the network as a whole, the utilization of gateways to upload data is balanced because gateways in better conditions will be always chosen as *uploading gateways* by newly coming data sources. However, if the *uploading gateway* does become unavailable (detected by s when it fails to receive three consecutive HELLO messages periodically from the *uploading gateway*), s selects a new *uploading gateway* among the remaining gateways in its *access area* and then invokes the procedure such that *area gateways* and *corresponding gateways* in nonsource *access areas* run a receiver-driven multicast protocol to establish a distribution tree rooted at the new *uploading gateway*.

4.3 Link-Controlled Routing Tree

The *LCRT* algorithm is run by the source in the source *access area* or by the *area gateway* in a nonsource *access area* to construct a routing tree that multicasts packets so as to minimize the impact of wireless interference during intraaccess area forwarding.

4.3.1 LCRT Metrics

It was analyzed in Section 3 that a smaller L improves the performance of V. The link-controlled routing tree in a nonsource *access area* is a multiroot distribution tree, on which multiple gateways, both *area gateways* and *corresponding gateways*, serve as multiple virtual senders for the nonsource *access area*. Each group member will receive V by connecting (directly or through intermediate nodes) to its closest gateway, allowing a shorter path (i.e., a smaller L) than if it has to connect to a single input gateway. The availability of multiple roots is guaranteed by multicasting between gateways in the upper tier.

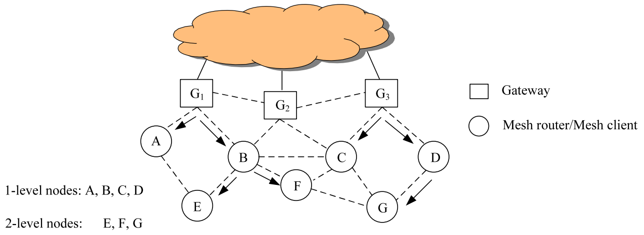


Fig. 5. An example of the *link-controlled routing tree*.

To reduce wireless interference analyzed in Section 3 (Fig. 2), in [26], the least number of nodes that can cover all multicast group members in an *access area* are selected as on-tree forwarders. This paper improves that algorithm by further combining the metric of *availability* with the motivation of achieving reliable and interference-controlled multicasting performance. The *availability* of the jth node in an *access area* is

$$U_j = \frac{C_j}{\sum_{f=0}^{F_j-1} r_f}, j \in n',$$

where n' is the set of nodes in the nonsource *access area*, C_j is the jth node's output channel capacity, F_j is the number of data flows that the jth node is transmitting, and r_f is the transmission rate of the fth flow. The LCRT algorithm essentially employs a weighted dominating set cover heuristic, where nodes are associated with a metric η defined as

$$\eta_j = D_j \times \frac{1}{N_j} \times U_j, j \in n', \tag{3}$$

where D_j is the number of direct downstream nodes of the jth node, and N_j is the number of nodes that are transmitting V or other data flows in the jth node's interference range. Usually, $D_j \leq N_j$. Nodes with larger values of η have the priority to be on-tree forwarders. Intuitively, a node with a larger η implies that it is currently serving fewer other data flows and also has a smaller number of interfering neighbors.

4.3.2 LCRT Algorithm

Before explaining the procedure of the LCRT algorithm, we define the following terms:

- *Node level* is defined as the least number of wireless hops from the multicast node to its closest gateway. A l-level node has at least l wireless hops to its closest gateways. As an example, in Fig. 5, A, B, C, and D are 1-level nodes because they only need one hop to reach their closest gateways.
- A node's *uncovered out degree* refers to the number of its direct child nodes (including group receivers and nonmember forwarders) that are not covered by any current multicast forwarders. In Fig. 5, the *uncovered out degree* of G_2 is 0.

We now see how to search the multiple roots of a LCRT. During the construction of a nonsource *access area*, each node responding to the *area gateway*'s AREA_CONSTRUCTION includes a TTL field into its JOIN_REPORT message. The TTL in JOIN_REPORT is initially set as 0 and increases by 1

[3] T is the difference between the time that s sends AREA_CONSTRUCTION and the time that s receives the first JOIN_REPORT.

after passing each hop. If the JOIN_REPORT message passes a corresponding gateway (in the same *access area*) on the way back to the *area gateway*, this corresponding gateway records the TTL and *IP address* from JOIN_REPORT and reports to the *area gateway* via the Internet that it has a shorter distance to the node than the *area gateway* does. Finally, the *area gateway* decides a *node level* based on the information received from both wireless and wired links. The TTL of the node to its closest gateway among both corresponding gateways and *area gateway* in the *access area* is set as the *level* of this node.

With the knowledge of nodes' levels, s or *area gateways* run the tree construction (Algorithm 1) in their own *access areas* which starts at the nodes with the largest levels.

Algorithm 1. Link-Controlled Routing Tree

```
Input:     An access area, the source $s$ in the source access
           area, or the area gateway $AG$ in a nonsource access area.
Output:    The link-controlled routing tree in the access area.
 1. $s/AG$ obtains each area node's address information,
    availability, and its hop distance to $s/AG/CG$
    (corresponding gateway) during constructing access
    areas;
 2. $s/AG$ assigns a node level to each area node according
    to its hop distance to $s/AG/CG$;
 3. $s/AG$ sets the multicast receivers with the levels of $L$ as
    leaf nodes, and $l = L - 1$ to start the tree construction;
    // $L$ is the highest level number
 4. While $l > 0$
 5.     While $um_l \neq 0$ // $um_l$ is the number of uncovered
        nodes (including group receivers and forwarders) in
        the $(l+1)$th level
 6.         $s/AG$ selects a $l$-level node that has the
            maximum $\eta$ value among all nonforwarding $l$-level
            nodes to be a forwarder;
 7.         $s/AG$ removes the $m$ $(l+1)$-level nodes that
            have been covered by the selected forwarder from the
            uncovered member set, and updates $um_l = um_l - m$;
 8.     $l = l - 1$;
 9. $AG$ sends on-tree forwarder lists via the Internet to $CG$s
    at which the forwarders are rooted.
10. $s/AG/CG$ sends a list of forwarders rooted at it
    wirelessly within its access area.
11. Nodes that received the forwarder information check
    whether they are on the list. If so, they remove their
    entries from the lists and pass on the updated
    forwarder lists.
12. The tree is constructed when all the forwarder lists
    in an access area are empty.
```

The procedures of Algorithm 1 is illustrated by the example in Fig. 5. Suppose B has the largest available output capacity. B is firstly selected to be a forwarder by the *area gateway* (suppose G_1) because B has the largest *uncovered out degree* as well. Then, among the remaining 1-level nodes, D is selected to be a forwarder because D covers the uncovered nodes in the second level and is not adjacent to B as well. Since B and D cover all receivers (i.e., E, F, and G), G_1 stops selecting forwarders and sends the list $\{D\}$ to G_3. Hereafter, two forwarder lists: $\{B\}$ and $\{D\}$ are sent wirelessly by G_1 and G_3, respectively. B and D become forwarders after receiving the lists.

Although this LCRT construction principle is based on the assumption that each wireless channel has one transmission rate, the algorithm can be easily extended to optimize other performance metrics in multirate channel environments, e.g., using previously developed algorithms for minimizing worst-case packet latency [12]. Moreover, the video streams can be made yet more robust by the recovery scheme (in the next section) which compensates for transmission outages with low overhead.

4.4 Dynamic Group Management

4.4.1 Admitting New Members

When a new member wants to join the video session V, it broadcasts JOIN_GROUP with $TTL = K$ and *group id* to detect existing *access areas*.[4] Eligible nodes respond to this new request with a JOIN_AVAILABLE message, which includes:

- responder's *area id;*
- responder's *confidence* (which is the reciprocal of the number of transmission outages that have happened at this node during the transmission of V); and
- responders' hop distances to both s and this new member.

A node in U is eligible when its wireless distance to s (over the hierarchical architecture) added to its wireless distance to the new member is less than K.

To reduce control overhead, when an eligible node receives JOIN_AVAILABLE from another eligible node, it does not forward this message if it offers a shorter hop distance from s to the new member. However, the eligible node sends its own JOIN_AVAILABLE, thereby providing the new member more choices in the selection of an upstream forwarding node. After receiving JOIN_AVAILABLE, the new member selects the responder through which the new member can connect to s via the least number of wireless hops with the most reliable wireless links, as its forwarders. Hence, this new member becomes a leaf node of the multicasting tree and starts receiving V from its upstream node.

4.4.2 Recovering from Transmission Interruptions and Outages

In order to reduce control overheads, link broken in our system is detected through application-layer monitoring of transmission outages instead of using conventional periodic link-advertisements. More specifically, if the transmission of V is not completed[5] but a node (say m_0) fails to receive γ ($\gamma = 3$ for our current implementation) consecutive packets of V, m_0 suspects that its upstream link is broken. m_0 broadcasts a BROKEN message to its upstream node via the control channel (Step 1 of Algorithm 2). Apart from a *message type* field, a BROKEN message includes an *interruption_id* field, incremented by 1 for each new link failure detection, which helps to distinguish different BROKEN messages issued by the same node. The upstream node (say m_1) checks whether it is sending and receiving V or not. If not, m_1

[4] Please note that the current value of K and the channel information are advertised, along with the *multicast group id* by the sender s.

[5] The end of V is illustrated by a *completion* packet in our multicasting.

implements Steps 3 and 4 in Algorithm 2. Apart from sending RESPONSE message which includes the *message type* and the *interruption_id* (the same as the one in the received BROKEN message), m_1 reports its interruption to its upstream node (say m_2) in the same way that its downstream node m_0 does. Otherwise, if m_1 is currently sending and receiving V, the BROKEN message from its downstream node triggers the recovery procedure in Steps 5-15 of Algorithm 2.

Algorithm 2. Link-Broken Interruption Recovery

```
Input:    A node (say $m_0$) detects the loss of three packets before the completion packet
Output:   The interruption is recovered
1.  $m_0$ sends BROKEN to its upstream node (say $m_1$);
2.  If $m_1$ is experiencing interruption
3.      $m_1$ sends RESPONSE to $m_0$ to stop it sending BROKEN with the same interruption_id;
4.      $m_1$ implements Algorithm 2 to deal with interruption as $m_0$ does;
5.  Else if $m_1$ receives $V$ well
6.      $m_1$ transmits $V$ via the backup channel;
7.      If $m_1$ receives another BROKEN from $m_0$ with an increased interruption_id
8.          $m_1$ informs its direct upstream node $m_2$ to select a new forwarder by LCRT to replace itself;
9.          $V$ is transmitted by the new forwarder via the back-up channel and by the original forwarder via the original channel as well;
10.         If $m_0$ cannot receive $V$ after issuing RECOVER to stop the new forwarder's transmission
11.             Go to step 8;
12.         Else, the broken-link interruption is recovered and the transmission from $m_1$ to $m_0$ will use the original channel.
13.     Else if $m_1$ does not receive new BROKEN from $m_0$
14.         $m_1$ multicasts $V$ through the backup and the original channels; $m_0$ receives from the backup channel and listens to the original channel;
15.         $m_1$ stops sending through the backup channel when $m_0$ detects a good transmission from the original channel.
```

A simple way with low overhead to reconnect m_0 and m_1 is to use a backup channel[6] at the upstream node m_1 to deliver V. m_0 will listen to both the data channel and the backup channel. If m_1 receives another BROKEN message (with an increased *interruption_id*) from m_0 after using the backup channel for transmission, m_1 reports the interruption to its upstream node, m_2 which then checks whether another node in the same level can reach m_0. Recall that nodes know about their downstream nodes during the group receivers' registration procedure. If more than one such node exists, one of them (say $m_1^{'}$) will be selected based on the LCRT algorithm to start multicasting V to m_0 via the backup channel (in order not to interfere the current on-tree multicasting). Meanwhile, m_1 continues the transmission of V to m_0 via the original channel. When m_0 receives packets from m_1, it broadcasts a RECOVER message via the control channel. After receiving RECOVER, $m_1^{'}$ stops its multicasting. If, however, the delivery of V to m_0 via m_1 resumes soon, m_0 sends a CONF_RECOVER message to m_1, which then continues delivering V via the original channel. Otherwise, m_1 asks its upstream node m_2 to reopen the connection between $m_1^{'}$ and m_0 via the backup channel.

The recovery scheme can be easily extended to fix interruptions caused by the unexpected departure of forwarding nodes (normally mesh clients). If m_0 does not receive either a RESPONSE message from the control channel or video data from the backup channel within a time period of transmitting γ consecutive packets after issuing a BROKEN message, m_0 broadcasts a FAILURE message with the information of the departing node (m_1) and a *failure_id* field (used to distinguish different FAILURE messages issued by the same node) back to the *area gateway*. A FAILURE message is only forwarded once by nodes with the levels not higher than the nodes that forwarded the FAILURE to them. On receiving the FAILURE report, the *area gateway* runs the LCRT algorithm to select a new forwarder for m_0. To avoid throughput loss of m_0, m_1's upstream forwarder may pick up temporary forwarder from the same level of m_1 for m_0 if it receives a FAILURE report about m_1. Temporary forwarders multicast V to m_1's child nodes via the backup channel.

5 Performance Evaluation

This section presents the results of our extensive simulation-based evaluation of our proposed algorithms, conducted using the discrete event network simulator NS2.33 [17]. For the purpose of comparative evaluation, we selected the following five multicast schemes:

1. EM: The integrated multicasting algorithm proposed in [8] (please refer to Section 2 for a brief overview of EM).
2. IR: An *Intramesh* shortest-path routing that broadcasts wireless multimedia packets to group members.
3. S-RMG: The video multicasting framework that includes TIA, WGU, and LCRT proposed in [26].
4. IW: A reduced version of our *integrated* multicasting algorithm which includes only TIA and WGU.
5. RMG: Our proposed *integrated* multicasting framework which includes all the algorithms in Section 4.

We evaluate the performance of the above multicasting schemes using the following metrics:

- Average multicast delay (AMD). AMD is used to evaluate the real time of video multicasting streams. In our simulations, it is calculated by
$$AMD = \frac{\sum_{i=0}^{n-1} AD_i}{n},$$
where AD_i is the average packet delay at the ith group member, and n is the group size.
- Average multicast throughput (AMT). AMT is used to evaluate the quality of video multicasting such as resolution. In our simulations, it is calculated by
$$AMT = \frac{\sum_{i=0}^{n-1} AT_i}{n},$$

[6] In our system, the backup channel is a channel different from the data channel and the control channel. The forwarding nodes on the tree are informed of this channel during the procedure of receiver registration.

TABLE 2
Simulation Parameters

NS-2 version	2.33
Radio propagation model	Nakagami
MAC protocol	802.11 with 11Mbps data rate
Bandwidth of wired links between routers	1000Mbps
Bandwidth of wired links between routers and WMN gateways	1000Mbps
Transmission range	100m
Video transmission rate	500Kbit/s
Simulation time	500s
Number of simulation runs	20
Network dimension	2 dimensions
Interference factor (κ)	$\sqrt{2}$

where AT_i is the average packet throughput at the ith group member.

- Average multicast delay jitter (AMDJ). AMDJ is observed to evaluate the continuous transmission of video multicasting. A smaller AMDJ is expected. In the simulations, AMDJ is calculated by

$$AMDJ = \frac{\sum_{i=0}^{n-1} ADJ_i}{n},$$

where ADJ_i is the average video delay jitter at the ith group member.

- Average multicast peak signal-to-noise ratio (AMPSNR). PSNR is a metric that captures the performance error between the original and the reconstructed video frames. Average multicast PSNR helps to access the application-level QoS of video multicasting transmissions. It is calculated by

$$AMPSNR = \frac{\sum_{i=0}^{n-1} APSNR_i}{n},$$

where $APSNR_i$ is the average PSNR of video at the ith group member. In our simulations, PSNR data are collected by using the EvalVid tool-set [27].

Table 2 lists the common parameters that we use to implement the simulations. In the simulations, we employ a probabilistic *Nakagami* propagation model which represents channel fading characteristics of a wide-range urban settings. Group members are randomly selected, to constitute around $20\% \sim 25\%$ of the whole WMN size. Each simulation result is the average of 20 simulation runs, which last for 500 seconds each. The node density guarantees that there are on average three nodes covered by one transmission range. Based on these parameters and our analysis in (1), we use $K = 3$ in our simulations.

5.1 Evaluation Using a Small-Scale WMN

This section studies the performance of the five multicast schemes using the small-scale WMN shown in Fig. 6. We vary the WMN size (including gateways, group members, and intermediate nodes) from 18 to 44 nodes and study the impact of this on the network performance in terms of average multicast delay, average multicast throughput, average multicast delay jitter, and average multicast PSNR. In the topology, the wired and wireless links are shown with solid lines and dashed lines, respectively. The integrated network consists of two routers (R_1 and R_2) that connect to two WMN gateways. To show the performance difference between EM and our algorithms, we place the video sender in the middle area between two gateways which makes EM separate the sender and part of its adjacent neighbors into two different "prefix islands." We also add disturbance traffic into the area close to the sender's closest gateway to generate transmission interference/contention. Disturbance traffic is injected at a constant rate, chosen uniformly over the range of [32 Kbps, 256 Kbps].

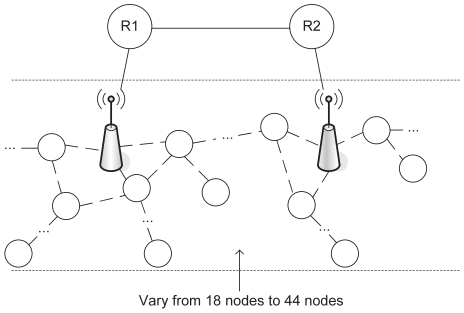


Fig. 6. The topology of small-size WMN.

5.1.1 Impact of Multicast Group Size on Performance

Fig. 7 shows average multicast delays when the WMN size varies. All other schemes generate longer average multicast delays than IR because of their overhead to construct a multicast tree. The reduced delays of RMG, in comparison with IW, show the positive effect of the LCRT algorithm in efficiently sharing wireless resources. The reduced delays of RMG, in comparison with S-RMG, show the positive effect of using reliable links in wireless networks. Note that EM generates the longest delays because it creates "islands" based on the locations of nodes and gateways. Our simulation settings force EM to separate the video source and part of its adjacent neighbors into two different "islands." Hence, although wireless transmissions to these separated adjacent neighbors can achieve better performance, video multicasting to these neighbors pass through the Internet first and then from the Internet back to WMN. In our algorithms (IW and RMG), the construction of *access areas* allows direct wireless transmissions from the sender to its adjacent neighbors. Also, the disturbance traffic in the area near to the sender's gateway degrades EM's performance. These

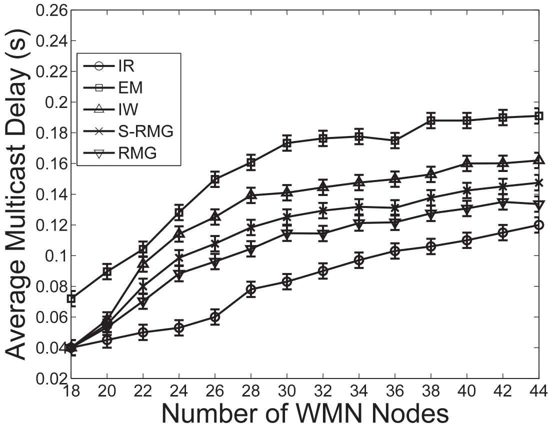


Fig. 7. The average delay performance when the WMN size in Fig. 6 increases from 18 to 44.

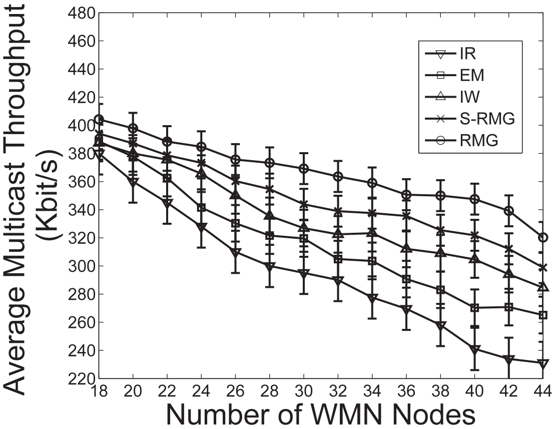


Fig. 8. The average throughput performance when the WMN size in Fig. 6 increases from 18 to 44.

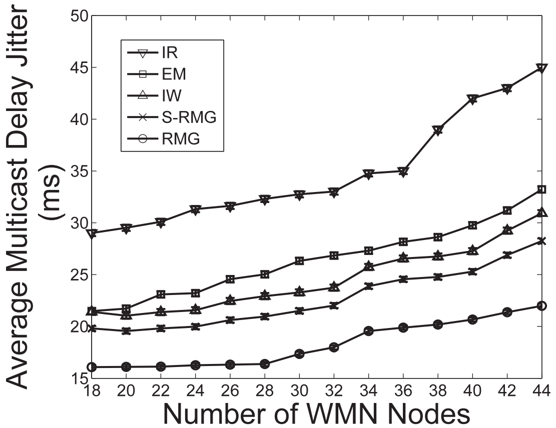


Fig. 9. The average delay jitter performance when the WMN size in Fig. 6 increases from 18 to 44.

results prove that the backhaul to wired resources should be used judiciously in order to improve wireless communication performance. Some nodes generate equal delays in RMG and EM when they have the same wireless hop distances via both *intramesh* routing and *integrated* routing. Some nodes generate longer delays in RMG than in EM when their wireless hop distances to the sender is within K but larger than the wireless hop distances in *integrated* routing. However, in either situation, the multicast delays of RMG are within the real-time delay bound.

Fig. 8 shows the average multicast throughput of the five multicast schemes. The integrated schemes (EM, IW, S-RMG, and RMG) achieve a higher throughput than IR because they utilize the stable and high-capacity wired connection. Note that, by constructing LCRT, RMG achieves higher throughput than IW, and by employing reliable links and decreasing control overheads, RMG achieves higher throughput than S-RMG. Note also that both RMG and IW perform better than EM because of the use of the *weighted gateway uploading* algorithm. Similar to the multicast delay performance, the throughput of nodes in RMG is lower than for the EM case where wireless hop distances to the sender are within K but larger than the wireless hop distances for *integrated* routing. However, the throughput achieved by these nodes is still acceptable.

Fig. 9 shows the average delay jitter for the five schemes. IR has the worst jitter performance while RMG achieves the best jitter performance. The large performance gap between RMG and IW can be attributed to the use of LCRT. LCRT greatly reduces packet delay variation caused by interference between sibling nodes. The delay jitter improvement of RMG, as compared to S-RMG, arises because the selection of reliable links aids uninterrupted transmissions. We also found that the RMG delay jitter of a few nodes closer to the sender's nonclosest gateway cannot overtake their EM delay jitter. It is mainly because EM uses internet shortcuts to decrease the wireless distance of these nodes to the sender.

Fig. 10 shows the average multicast PSNR performance of the five schemes. IR generates the worst average multicast PSNR performance because it uses only wireless links to transmit packets. The wireless broadcast property of wireless medium causes severe signal interference/contention and fading in IR. RMG achieves the best average multicast PSNR performance. RMG's superior performance for average multicast PSNR can be explained by its formation of *access areas* (limiting the number of wireless transmission hops and therefore reducing the problem of signal fading as compared to EM), its employment of reliable links to upload V to the Internet (greatly reducing signal loss as compared to EM and S-RMG), and its construction of LCRT trees (controlling the noise due to interference/contention as compared to EM, IW, and S-RMG). RMG's average multicast PSNR performance under different simulated network conditions is acceptable.

5.2 Evaluation Using a Large-Scale WMN

This section studies the impact of the WMN size on the performance of the four multicast schemes EM, IW, S-RMG, and RMG. We did not evaluate IR in the simulation because it is not practical to use IR in a large-scale network. As shown in Fig. 11, the wired part of the integrated WMN topology is a combination of two MCI-ISP backbone networks. Each router in the topology represents a domain and therefore there are 35 domains in the wired network. The wireless part, a WMN consisting of gateways and mesh nodes, increases its size from 115 nodes to 250 nodes. These nodes spread across 15 domains in Fig. 11 through 25 gateways. We set up four background flows to disturb the video multicasting in the wireless part of the simulated network: one disturbance flow is introduced

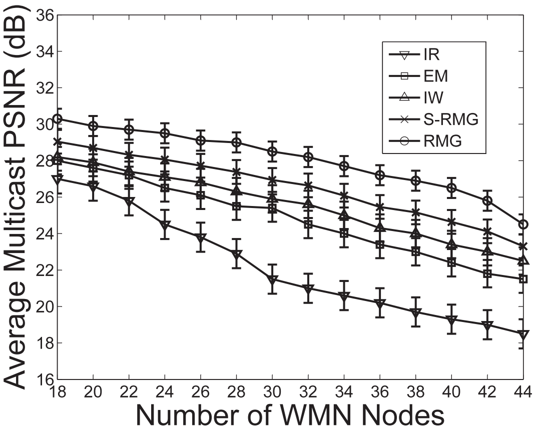


Fig. 10. The average PSNR performance when the WMN size in Fig. 6 increases from 18 to 44.

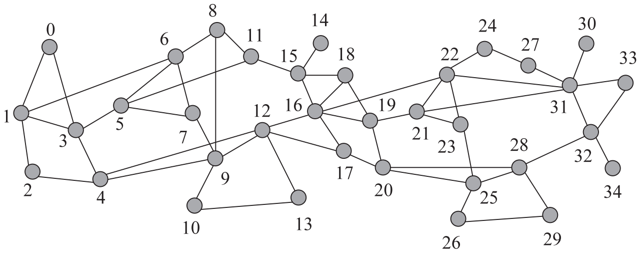


Fig. 11. The wired backbone in the large-scale simulation.

near to the video sender's closest gateway; the other three flows are introduced into nonsource *access areas* by nodes near to an *AG* or a *corresponding gateway* that is a root of the LCRT tree in its *access area*. Transmission rates of disturbance flows are randomly decided by the programs in the range of [32 Kbps, 256 Kbps].

5.2.1 Impact of Multicast Group Size on Performance

The average multicast delay of the four multicast schemes are shown in Fig. 12. Note that IW achieves shorter delay performance than EM does. This is because the sender in EM needs to upload the flow to its closest gateway that is interfered by disturbance traffic in the simulation. Hence, the flow in EM requires more time to reach the receivers in another "island." RMG achieves shorter delays than IW. It shows the effectiveness of controlling sibling transmissions (introduced in Fig. 2) achieved by LCRT in a large size WMN. Also, it is observed that the use of an unreliable link in the source *access area* adversely affects the multicast reception by group members more significantly in a large area WMN, than in a small area WMN. Therefore, the improvement of delay performance that RMG overtakes S-RMG is better in the large WMN simulation than in the small WMN simulation. For the same reason of the small-scale WMN simulation, a few nodes closer to the nonuploading gateways in the source areas achieve shorter delay performance in EM. Nodes in nonsource access areas of RMG achieve better delay performance with 100 percent confidence than those of EM. The above observations suggest that RMG suits large-scale multicasting.

Fig. 13 shows the average multicast throughput for EM, IW, S-RMG, and RMG when the WMN size increases from 115 to 250 nodes. The throughput of EM is lower as compared to those of IW and RMG. A major reason for this phenomenon is because of the disturbance traffic near to the video sender's closest gateway. The sender of EM passes video traffic to its closest gateway, but IW and RMG choose a gateway with a good condition (i.e., a good balance between high available capacity and good reliability) to connect for *integrated* video multicasting. These results prove that the backhaul to wired resources should consider dynamic network conditions to avoid bottleneck gateways. Furthermore, with the use of reliable LCRT, RMG performs best.

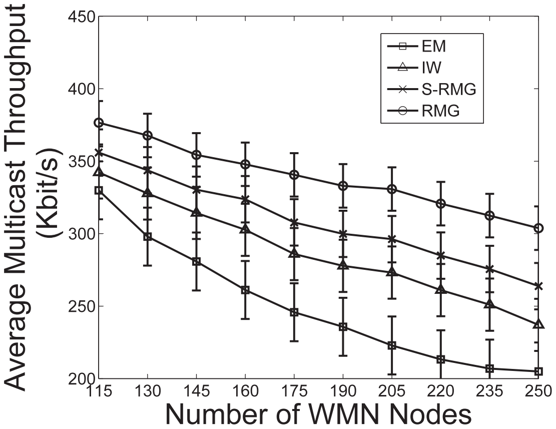


Fig. 13. The average throughput performance when the WMN size increases from 115 to 250.

Fig. 14 compares the average multicast delay jitter of the four multicast schemes. When increasing the WMN size, the level of jitter of IW is better than that of EM because of the positive effectiveness of the WGU algorithm. The jitter of all four schemes increases with increasing WMN size. However, the jitter of RMG is controlled within the delay jitter bound of wireless communications (30 ms). These results prove the advantages of using LCRT with reliable links in RMG in reducing the interference between sibling transmissions.

Fig. 15 shows the average multicast PSNR performance for the four schemes. As observed, RMG achieves the best average multicast PSNR performance, while, expectedly, the PSNR drops with an increase in the size of the WMN. The plotted results in this figure can be explained by the similar reasons for the results in Fig. 10.

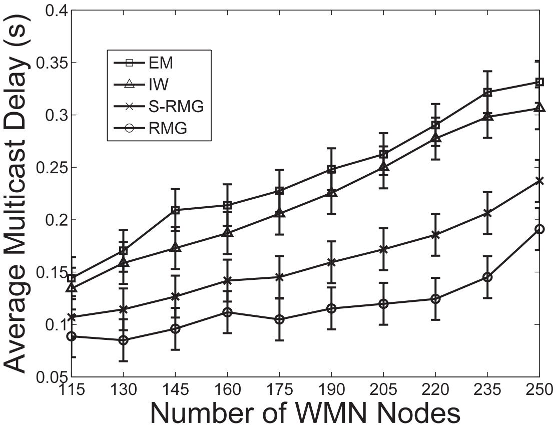


Fig. 12. The average delays when the WMN size increases from 115 to 250.

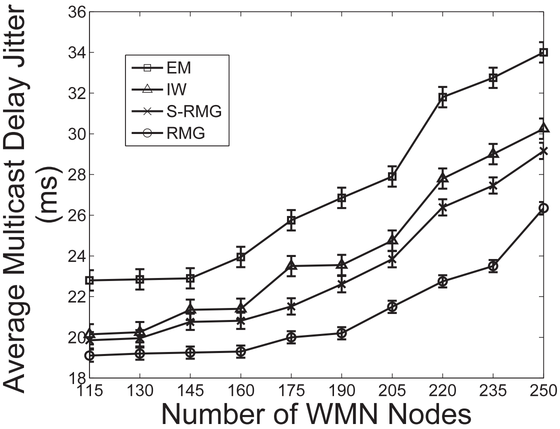


Fig. 14. The average delay jitter performance when the WMN size increases from 115 to 250.

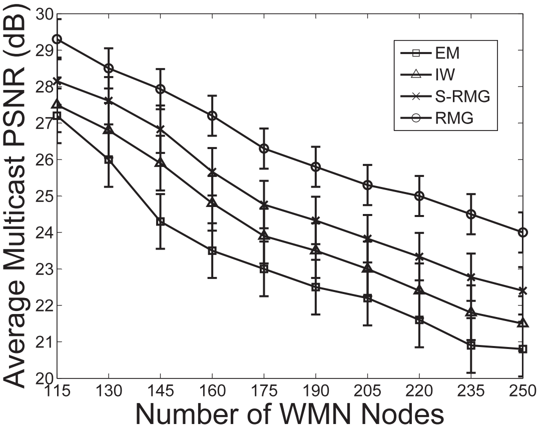


Fig. 15. The average PSNR performance when the WMN size increases from 115 to 250.

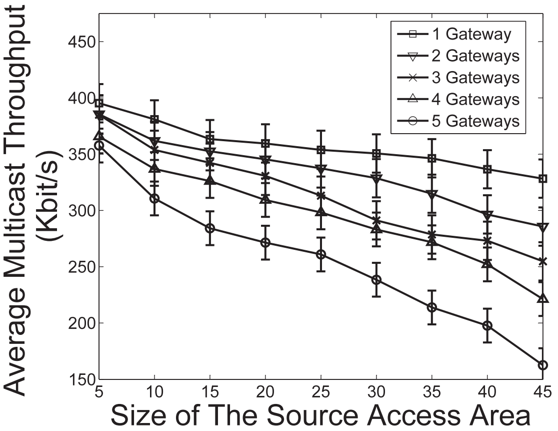


Fig. 17. The average throughput performance varies with the source *access area* size and the number of gateways.

5.2.2 Impact of Number of Gateways and Number of Nodes in an Access Area

This section studies the impact of the number of gateways and the number of nodes in the source *access area* on the performance of RMG. We vary the number of gateways per source *access area* from 1 to 5 and the number of mesh nodes per source *access area* from 5 to 45, respectively—this is done by explicitly placing gateway nodes (for a given topology) to ensure that a source or receiver is within K hops of the specified number of gateway nodes. Mesh nodes are uniformly distributed in the source *access area* with the density of three nodes per transmission range. The total number of nodes in other *access areas* are 160.

Fig. 16 shows the average multicast throughput of RMG when the number of gateways in each source *access area* varies from 1 to 5 and the number of nodes in each source *access area* is 10, 25, and 45. The figure shows that multicast throughput increases with an increase in the number of gateways in the source *access area*, illustrating that the use of multiple gateways can significantly improve the throughput of an *integrated* network.

Fig. 17 shows how the number of nodes in a source *access area* affects the average throughput of RMG. It shows that increasing the size of a source *access area* has a negative effect on the throughput. However, this can be corrected by increasing the number of gateways per source *access area*.

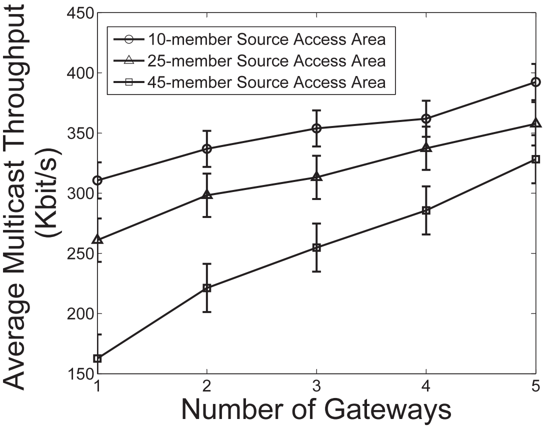


Fig. 16. The average multicast throughput performance varies with the number of gateways in the source *access area*.

5.2.3 Impact of Access Area Threshold K

We now observe K's impact on video performance. The simulation employs the backbone in Fig. 11 to connect WMN gateways for the video communications in a group of 260 WMN nodes. Table 3 lists the video performance (average throughput) achieved and the system resources (WMN gateways) required by the *resource-aware multigateway WMN multicasting* scheme when the *access area* varies. Increasing K degrades video performance, as the video traffic experiences more interference when traversing a larger number of wireless links. WMN video communications tend to prefer *intramesh* routing when K becomes large enough. This is verified by our results which show that the number of required WMN gateways decreases with increasing K. Moreover, the results in the table show that varying K presents a tradeoff between high video performance and low system resources, validating our observation of a tradeoff between $(r_V)_{basic}$ and ρ in (1). In practical applications, as we have analyzed in (1), K can be determined according to the desired performance and the practical system conditions, such as mesh node average distribution density and one-hop distance. In our simulations, if we put the following simulation parameters into (1): $C = 11$ Mbps, $r_V = 500$ Kbps, $(r_V)_{basic} = 250$ Kbps, $\kappa = \sqrt{2}$, the link loss rates in the range of $[0, 0.2]$, and the node density distribution which guarantees that there are on average three nodes covered by one transmission range, we have $K \approx 3.11$ which closely meets the simulation observation that $K = 3$ is the best choice when considering both video performance and consumed network resources.

TABLE 3
Variation of Average Throughput and Number of WMN Gateways with Different Ks

K	2	3	4	5	6	7	8
Average throughput (Kbit/s)	312.5	287.8	245.6	200.5	159.3	130.2	117.3
Number of WMN gateways	48	25	18	16	16	16	15

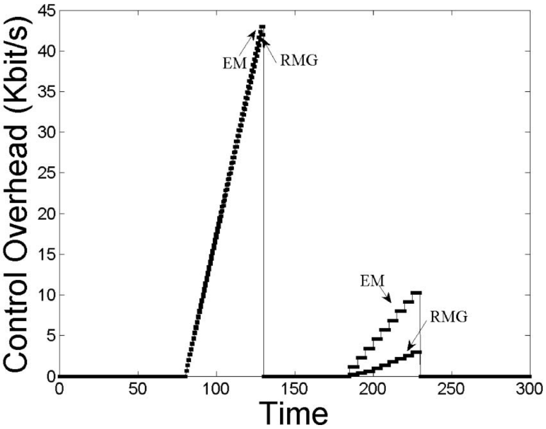


Fig. 18. Control overhead created in maintaining dynamic group membership.

5.3 Evaluation of Dynamic Group Performance

This section presents simulation results to evaluate the performance of EM and RMG when group members dynamically change. There are 350 multicast nodes. The simulation period is 300 seconds and includes two different phases: a join phase and a broken link phase. In the join phase, 64 new hosts join the group uniformly at random between the simulated time 80 and 130 seconds. In the broken link phase, 10 wireless links break at a time uniformly distributed between 180 and 230 seconds. The results in Fig. 18 show that EM and RMG generate similar amounts of overheads when admitting new group members. However, the join procedure of EM nodes is based on the periodic advertisements which introduce more control packets to the WMN. These overheads are not plotted in the figure in order to clearly compare the performance of two protocols to deal with dynamic changes. As for the control overheads generated by recovering broken links, the simulation results show that RMG generates less overheads than EM. In EM, when a node detects interruption, it depends on the advertisements of other nodes to rebuild the connection to the multicast group. In RMG, some broken links are recovered by the backup channel without requiring a new upstream node. Although EM can also explore the backup channel recovery, the periodic advertisements may still introduce more overheads (not shown in the figure) into the WMN. Overall, our simulations establish that RMG can achieves lower multicast delay and jitter, and higher video throughput, than EM, without incurring any additional overhead.

6 Conclusions

The paper described and studied a *resource-aware multi-gateway video multicasting* framework for WMNs that reduces the negative impacts of multiple wireless hops by judiciously employing high-capacity wired Internet shortcuts. This framework allows high-bandwidth multicasting to be performed over a wide geographic area. The *two-tier integrated architecture* algorithm chooses communication paths—*intramesh* paths or *integrated* paths—between nodes by organizing them into a clustered and layered architecture; the *weighted gateway uploading* algorithm avoids "busy" gateways by uploading video in a manner that balances load and reliability; the *link-controlled routing tree* algorithm decreases interference from parallel multicasting by constructing a multicast tree with the least number of forwarders in each *access area*; and the *dynamic group management* algorithm provides low-overhead maintenance of the multicast forwarding trees when dynamic changes take place. Our design principles are validated by our extensive simulation results, showing that the multicasting algorithms can achieve up to 40 percent more throughput than other related published approaches.

Our hierarchy architecture is flexible enough to incorporate progressively sophisticated enhancements to each individual algorithmic phase. For example, while the *access area* construction uses a hop count threshold to avoid the usage of excessive wireless links, this static metric can be easily replaced by a dynamic metric (e.g., delay distance threshold). Moreover, the architecture dynamically changes with the variation of video performance and group membership. In general, our scheme not only achieves high-performance video multicasting of a single flow but also enables a WMN to admit more video streams because of the balance in using wired and wireless resources and the controlled overheads. These improvements require the use of multiple gateways within the distance of K wireless hops for a source *access area* or $(K - k)$ wireless hops for a nonsource *access area*.

In future work, we plan to enhance this *integrated* approach to better support the concurrent transmission of multiple, time-varying multicast flows, as well as investigate the use of multiple distinct integrated paths for the transmission of layered video within a single multicast flow.

Acknowledgments

The first author thanks Glyndŵr University for providing conditions needed to conduct this research. This research is partially supported by the IRCSET Embark Initiative Postdoctoral Research Fellowship and the Australian Research Council Discovery grants DP0664791 & DP1096353.

References

[1] S. Weber, X. Yang, J. Andrews, and G. Veciana, "Transmission Capacity of Wireless Ad Hoc Networks with Outage Constraints," *IEEE Trans. Information Theory,* vol. 51, no. 12, pp. 4091-4102, Dec. 2005.

[2] S. Shakkottai, X. Liu, and R. Srikant, "The Multicast Capacity of Large Multihop Wireless Networks," *Proc. ACM MobiHoc,* pp. 247-255, 2007.

[3] J. Jun and M.L. Sichitiu, "The Nominal Capacity of Wireless Mesh Networks," *IEEE Wireless Comm.,* vol. 10, no. 5, pp. 8-14, Oct. 2003.

[4] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable Application Layer Multicast," *Proc. ACM SIGCOMM,* pp. 205-217, Aug. 2002.

[5] W. Tu and W. Jia, "A Scalable and Efficient End Host Multicast for Peer-to-Peer Systems," *Proc. IEEE Global Telecomm. Conf. (GlobeCom '04),* pp. 967-971, Nov./Dec. 2004.

[6] B. Zhang, S. Jamin, and L. Zhang, "Host Multicast: A Framework for Delivering Multicast to End Users," *Proc. IEEE INFOCOM,* pp. 1366-1375, June 2002.

[7] W. Tu, C. Sreenan, and W. Jia, "Worst-Case Delay Control in Multi-Group Overlay Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 18, no. 10, pp. 1407-1419, Oct. 2007.

[8] P.M. Ruiz, F.J. Galera, C. Jelger, and T. Noel, "Efficient Multicast Routing in Wireless Mesh Networks Connected to Internet," *Proc. First Int'l Conf. Integrated Internet Ad Hoc and Sensor Networks,* 2006.

[9] Y. Amir, C. Danilov, R.M. Elefteri, and N. Rivera, "An Inter-Domain Routing Protocol for Multi-Homed Wireless Mesh Networks," *Proc. IEEE Int'l Symp. World of Wireless, Mobile and Multimedia Networks (WoWMoM '07)*, 2007.

[10] P.M. Ruiz, A.F. Gomez-Skarmeta, and I. Groves, "The MMARP Protocol for Efficient Support of Standard IP Multicast in Mobile Ad Hoc Access Networks," *Proc. First Mobile and Wireless Comm.*, pp. 478-482, June 2003.

[11] B. Liu, C. Chou, A. Misra, and S. Jha, "Resource-Aware Routing of Broadcast and Multicast in Multi-Rate Wireless Mesh Networks," *Mobile Networks and Applications*, vol. 13, nos. 1/2, pp. 38-53, 2008.

[12] C. Chou, A. Misra, and J. Qadir, "Low Latency Broadcast in Multi-Rate Wireless Mesh Networks," *IEEE J. Selected Areas Comm.*, Special Issue on Multi-Hop Wireless Mesh Networks, vol. 24, no. 11, pp. 2081-2091, Nov. 2006.

[13] T. Wang, X. Du, W. Cheng, Z. Yang, and W. Liu, "A Fast Broadcast Tree Construction in Multi-Rate Wireless Mesh Networks," *Proc. IEEE Int'l Conf. Comm. (ICC '07)*, June 2007.

[14] W. Tu and C. Sreenan, "Adaptive Split Transmission for Video Streams in Wireless Mesh Networks," *Proc. IEEE Wireless Comm. and Networks Conf. (WCNC '08)*, 2008.

[15] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable Application Layer Multicast," *Proc. ACM SIGCOMM*, pp. 205-217, Aug. 2002.

[16] W. Tu and W. Jia, "A Scalable and Efficient End Host Multicast for Peer-to-Peer Systems—DSCT," *Proc. IEEE 47th Ann. Global Telecomm. Conf. (GlobeCom '04)*, pp. 967-971, Nov./Dec. 2004.

[17] UC Berkeley, LBL, USC/ISI, and Xerox PARC, "ns Notes and Documentation," Oct. 1999.

[18] J. Edwards, "Integrated Wired and Wireless: The Best of Both Worlds," *IQ Magazine*, vol. VII, no. 2, 2006.

[19] D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, vol. 353, pp. 153-181, 1996.

[20] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destinaation-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *ACM SIGCOMM Computer Comm. Rev.*, vol. 24, no. 4, pp. 234-244, 1994.

[21] C.E. Perkins and E. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *Proc. IEEE Second Workshop Mobile Computing Systems and Applications*, pp. 90-100, 2002.

[22] S. Kwon and N. Shroff, "Paradox of Shortest Path Routing for Large Multi-Hop Wireless Networks," *Proc. IEEE INFOCOM*, pp. 1001-1009, May 2007.

[23] S. Kwon and N. Shroff, "Analysis of Shortest Path Routing for Large Multi-Hop Wireless Networks," *IEEE/ACM Trans. Networking*, vol. 17, no. 3, pp. 857-869, June 2009.

[24] B. Liu, P. Thiran, and D. Towsley, "Capacity of a Wireless Ad Hoc Network with Infrastructure," *Proc. ACM MobiHoc*, Sept. 2007.

[25] C. Jelger and T. Noel, "Proactive Address Autoconfiguration and Prefix Continuity in IPv6 Hybrid Ad Hoc Networks," *Proc. IEEE Second Ann. Sensor and Ad Hoc Comm. and Networks (SECON '05)*, Sept. 2005.

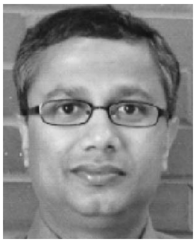
[26] W. Tu, C. Sreenan, C. Chou, A. Misra, and S. Jha, "Resource-Aware Video Multicasting via Access Gateways in Wireless Mesh Networks," *Proc. IEEE Int'l Conf. Network Protocols (ICNP '08)*, Oct. 2008.

[27] J. Klaue, B. Rathke, and A. Wolisz, "EvalVid—A Framework for Video Transmission and Quality Evaluation," *Proc. 13th Int'l Conf. Modeling Techniques and Tools for Computer Performance Evaluation*, pp. 255-272, Sept. 2003.

Wanqing Tu received the PhD degree from the Department of Computer Science at the City University of Hong Kong in 2006. She is a Lecturer in the School of Computing at Glyndŵr University, United Kingdom. Her research interests include QoS, wireless communications, multimedia communications, overlay networks, and distributed computing. She received an IRCSET Embark Initiative Postdoctoral Research Fellowship. She received the Best Paper Award in 2005. She is a member of the IEEE and the IEEE Computer Society.

Cormac J. Sreenan received the PhD degree in computer science from Cambridge University. He is a professor of computer science at University College Cork (UCC) in Ireland. Prior to joining UCC in 1999, he was on the Research Staff at AT&T Labs—Research, Florham Park, New Jersey, and at Bell Labs, Murray Hill, New Jersey. He is currently on the editorial boards of the *IEEE Transactions on Mobile Computing*, the *ACM Transactions on Sensor Networks*, and the *ACM/Springer Multimedia Systems Journal*. He is a fellow of the British Computer Society and a member of the IEEE and the IEEE Computer Society.

Chun Tung Chou received the BA degree in engineering science from the University of Oxford and the PhD degree in Systems and Control Engineering from the University of Cambridge. He is a associate professor at the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia. His current research interests include wireless mesh networks, wireless sensor networks, multimedia networking, and network optimisation. He has published more than 90 journal and conference articles in these areas. He is a member of the IEEE and the IEEE Computer Society.



Archan Misra received the PhD degree in electrical and computer engineering from the University of Maryland at College Park in May, 2000, and the BTech degree in electronics and communication engineering from IIT Kharagpur, India, in July 1993. He is currently an associate professor of information systems at Singapore Management University (SMU) and a deputy director of the Living Analytics Research Center (LARC) at SMU. His current research interests include pervasive computing & mobile systems, with specific focus on energy-efficient stream processing, data mining for semantic activity recognition, and hybrid cloud+P2P middleware for context-based mobile applications. Over the past 10 years, as part of his previous employment with IBM Research and Telcordia Technologies, he has published extensively in the areas of wireless networks and pervasive computing and is a coauthor on papers that received the Best Paper awards from EUC 2008, ACM WOWMOM 2002, and IEEE MILCOM 2001. He is presently an area editor of the *Journal of Pervasive and Mobile Computing* and he chaired the IEEE Computer Society's Technical Committee on Computer Communications (TCCC) from 2005-2007. He is a member of the IEEE and the IEEE Computer Society.

Sanjay K. Jha received the PhD degree from the University of Technology, Sydney, Australia. He is a professor and head of the Network Group at the School of Computer Science and Engineering at the University of New South Wales. He is an associate editor of the *IEEE Transactions on Mobile Computing*. He was a Member-at-Large, Technical Committee on Computer Communications (TCCC), IEEE Computer Society for a number of years. He has served on program committees of several conferences. He was cochair and general chair of the Emnets-1 and Emnets-II workshops, respectively. He was also the general chair of the ACM Sensys 2007 symposium. His research activities include a wide range of topics in networking including wireless sensor networks, ad hoc/community wireless networks, resilience/quality of service (QoS) in IP networks, and active/programmable networks. He has published more than 100 articles in high quality journals and conferences. He is the principal author of the book *Engineering Internet QoS* and a coeditor of the book *Wireless Sensor Networks: A Systems Perspective*. He is a senior member of the IEEE and the IEEE Computer Society.