QoS Multilayered Multicast Routing Protocol for Video Transmission in Heterogeneous Wireless Ad Hoc Networks

OSAMAH BADARNEH[1], MICHEL KADOCH[1], AHMED ELHAKEEM[2]
[1]Department de Génie Electrique, École de Technologie Supérieure, Université du Québec
[2]Department of Electrical and Computer Engineering, Concordia University
MONTREAL, CANADA
{osamah.badarneh.1@ens.etsmtl.ca, Michel.Kadoch@etsmtl.ca, ahmed@ece.concordia.ca}

Abstract: - In wireless ad hoc networks, nodes are expected to be heterogeneous with a set of multicast destinations greatly differing in their end devices and QoS requirements. This paper proposes two algorithms for multilayered video multicast over heterogeneous wireless ad hoc networks. The two algorithms are, Multiple Shortest Path Tree (MSPT) and Multiple Steiner Minimum Tree (MSMT). In this paper, we assume that each destination has a preference number of video layers; which is equal to its capacity. Moreover, we do not consider only the capacities of nodes in the network but also the bandwidth of each link. In order to increase user satisfaction for a group of heterogeneous destinations, we exploit different types of multiple multicast trees policy. Simulations show that the proposed schemes greatly improve the QoS requirements (increase user satisfaction) for a set of destinations. In addition, simulations show that multiple Hybrid-II multicast trees offer higher user satisfaction than multiple Hybrid-I multicast trees and multiple node-disjoint trees. The cost of that is the robustness against link failure. Therefore, it is a trade off between providing robustness against path breaks and increasing user satisfaction.

Key-Words: - Multilayered multicast; MDC; LC, Heterogeneous wireless ad hoc networks.

1 Introduction

Wireless ad hoc network is compromised of wireless nodes that can dynamically self-organize into an arbitrary and temporary topology to form a network without the support of any fixed infrastructure or central administration. These features make wireless ad hoc network well suited for disaster recovery, emergency operations and military activities.

Multicasting plays a crucial role in many applications of wireless ad hoc networks. Because of the limitations of the transmission bandwidth and strict battery power in wireless ad hoc networks, multicasting can significantly improve the performance of this type of network. Over the past couple of years, several multicast routing protocols have been proposed. These routing protocols can be categorized into different categories [1, 2].

In traditional multicast routing, all destination nodes receive the same amount of multicast data. In contrast to traditional multicast, not all destination nodes in multilayered multicast receive the same amount of data. Each destination node has a preference values for each layer of streams (QoS level) according to its available bandwidth.

Hierarchical encoding technique [3] was proposed for the efficient use of resources in heterogeneous networks. There are two types of hierarchical encoding techniques, namely, layered coding (LC) and multiple description code (MDC). LC produces multiple *dependent* sub-streams, base layer and some enhancement layers. The base layer contains the most important information, without it the video cannot decoded properly, and can provide a low (basic) level of quality, but remaining enhancement layers serve to refine the base layer quality. On the other hand, MDC generates multiple *independent* sub-streams (each called description), *i.e.,* no specific description is needed in order to render the remaining descriptions useful. The reception of one description can guarantee a basic level of quality and additional descriptions can further improve the quality.

In this paper, we assume heterogeneous wireless ad hoc networks in which nodes have different capacities (capabilities). The capacity of a node means the number of video layers that can be received and transmitted. For example, nodes have a capacity of one means that they can receive only one video layer and then transmit it if they are forwarding nodes or display the video if they are destination nodes. There are several factors that can limit the capacity of a node, namely but not limited to, remaining power, number of sessions participating in, buffer size and the type of the node (laptop, PDA, …). However, the number of video layers that can be handled by an arbitrary node does not depend only on its capacity but also on its available bandwidth.

The rest of the paper is organized as follows. Related work is presented in the next section. In the section three, we present the formulation of the problem. Our multiple multicast trees routing protocol is presented in section four. In section five, we present the proposed algorithms for the construction of multiple multicast trees. Simulation results are discussed in section six. Finally, section seven concludes the paper.

2 Related Work

Little attention has been paid to multicast video over wireless ad hoc networks. Multiple tree protocol called Robust Demand-driven Video Multicast Routing (RDVMR) protocol has been proposed in [4]. RDVMR exploits the path diversity and error resilience properties of Multiple Description Coding (MDC). RDVMR constructs multiple trees (*K*-tree) in parallel with a reduced number of shared nodes among them to provide robustness against path breaks. A novel path based Steiner tree heuristic have been proposed to reduce the number of forwarding nodes and as a result reducing the total data overhead.

In [5], multiple multicast trees (2-trees) routing protocols have been proposed. The first scheme constructs two disjoint multicast trees in a serial (serial multiple disjoint trees multicast routing protocol (serial MDTMR)), but distributed fashion. In order to overcome routing overhead and construction delay, parallel multiple nearly-disjoint trees multicast routing protocol (parallel MNTMR) is proposed. Both protocols exploit MDC to provide robustness for video multicast applications.

A multicast scheme for MD video over ad hoc networks was proposed in [6]. In this scheme a number of multicast trees are used, with each multicast tree supporting one video description. Further, each description is coded into a base layer and a number of enhancement layers. Packets belonging to the same description from both the base layer and enhancement layers are transmitted on the same tree. Authors show that this MD video multicast approach can effectively deal with frequent link failures and diverse link qualities in wireless ad hoc networks.

Multiple Trees Multicast Ad Hoc On-demand Distance Vector (MT-MAODV) routing protocol, which is an extended version of MAODV, is proposed for video multicast in ad hoc wireless network [7]. MT-MAODV constructs two optimally disjoint multicast trees in a single routine. Multiple description coding (MDC) scheme is used to split the video into several independent and equally important video descriptions. Each description is transmitted to different trees.

Multiple paths/trees in parallel are constructed to meet the QoS requirements of a multicast call are proposed in [8]. Three multicast routing schemes are proposed, namely; shortest path tree based multiple paths (SPTM), least cost tree based multiple paths (LCTM) and multiple least cost trees (MLCT). Each of the three schemes has a different objective, such as minimizing the delay of the call or minimizing the overall network cost. In this work; nodes' capacities and destinations' heterogeneity are not taken into account.

Previous protocols [4, 5, 7] did not consider the available link bandwidth or the heterogeneity of the destination nodes. On the other hand, work in [6] only take into consideration the available link bandwidth. The main objective of the previous works is to improve the quality of the received video by exploiting the path-diversity and error resilience properties of MDC.

Our work is different from previous works in the following aspects:

- It takes into consideration the heterogeneity of the destination nodes.
- It takes into consideration, both the available links bandwidth and nodes' capacity.

3 Problem Formulation

3.1 Code Division Multiple Access (CDMA) over TDMA

In this paper, the MAC sub-layer adopts the CDMA over TDMA channel model [9]. Bandwidth is measured in the unit of free timeslots. Assigning free timeslots to a link to maximize the available bandwidth of the path is NP-hard [9]. However, path bandwidth depends not only on the free timeslots over the links, but also on the timeslot assignment scheme. In this paper, we adopt the timeslots assignment algorithm proposed in [9]. Each node, in CDMA, uses a pre-assigned code for communication with neighboring nodes in a conflict free fashion [10].

In general, a wireless ad hoc network having V nodes (vertices) and E links (edges) can be modeled as a random graph $G(V,E)$, where V is the set of vertices; representing wireless nodes, and E is the set of edges; representing wireless links. A wireless link, $e=\{x,y\}$, $e\in E$ between two nodes,

x and y, indicates that both nodes are within the transmission's range of each other.

Definition 1. The available bandwidth of a link $e_i, 1 \le i \le k$, is the number of common free timeslots over the link.

$$Bw(e_i) = free\_timeslot(e_i) \quad (1)$$

Definition 2. A path P is compromised of a set of links $e_i, 1 \le i \le k$. The bandwidth of a path P is the minimum of the residual bandwidth of all links on the path (bottleneck bandwidth).

$$Bw(P) = min\, BW(e_i), \quad e_i \in P, 1 \le i \le k \quad (2)$$

Definition 3. The capacity of path P is the minimum capacity of the nodes on the path.

$$Cp(P) = min\, Cp(x_i), \quad x_i \in V, \ 1 \le i \le m \quad (3)$$

Definition 4. The number of video layers that can be transmitted over a path P is the minimum of the minimum bandwidth of the path and the minimum capacity of the nodes on the path.

$$N_{VD}(P) = min\{Bw(P), Cp(P)\} \quad (4)$$

Definition 5. The bandwidth of a tree T_l is the bandwidth of the bottleneck link.

$$Bw(T_l) = \min_{e \in T_l}\{Bw(e)\} \quad (5)$$

Definition 6. The capacity of a tree T_l is the minimum capacity of all nodes on the tree.

$$Cp(T_l) = \min_{x \in T_l}\{Cp(x)\} \quad (6)$$

Definition 7. The number of video layers that can be transmitted over a tree T_l is the minimum of the minimum bandwidth of the tree and the minimum capacity of the tree.

$$N_{VD}(T_l) = min\{Bw(T_l), Cp(T_l)\} \quad (7)$$

In this paper, we consider a session with single multicast source. The multicast source can generate M description (layers), without loss of generality, we assume $M = 3$. Therefore, our problem can be formulated as follows. Given a number of wireless nodes with random capacities and available links bandwidth between them, find a multicast tree rooted at the source node and spanning a set of destinations with different QoS-level (each destination has a preference number of video layers) such that user satisfaction defined in (8) is maximized.

4 QoS Multilayered Multicast Routing Protocol (QMMRP)

QMMRP is an on-demand multicast routing protocol. The multicast source node constructs multiple multicast trees whenever it has a multicast data video ready for transmission. The main goal of QMMRP is to build multiple multicast trees such that user satisfaction defined in (8) is maximized. The construction of multiple multicast trees is performed using one of the algorithms in section 5.

In contrast to traditional multicast routing protocol in wireless ad hoc networks, not all destination nodes in QMMRP receive the same amount of multicast data; each destination has a preference values for each layer of stream (QoS level) according to its capacity.

As mentioned before, MDC generates multiple *independent* descriptions with same priority; if any layer is lost the data can be decoded using other layer. In contrast, layered coding generates multiple *dependent* layers with different priority, the basic layer is the most important one and without it the data cannot be decoded. Therefore, we propose to exploit MDC as a video encoding technique. We refer to layers, descriptions interchangeably.

4.1 Path Discovery and Reply Phases

In QMMRP, paths are discovered on-demand by propagating the Quality Route REQuest (QRREQ) packets and Quality Route REPply (QRREP) packets between the source and the destination nodes. When a multicast source receives a request from the application layer, it initiates and broadcasts a QRREQ packet to its neighbors. Each packet contains the following fields: (*source, destinations, request_id, type, route, free-timeslots, QoS_level, TTL, num_hops*), where (*source, request_id*) is used to uniquely identify a packet. A group of destinations is recorded in the *destinations* field. The *type* refers to packet type, it mat be QRREQ. The *route* records the path from the source to the current traversed node. The *free-timeslots* is a list of free timeslots among the current traversed node and the last node in the route. The *TTL* is used to limit hop-count of the path (end-to-end delay). The *num_hops* field records the number of hops from the source to the current traversed node. The number of video description that can be delivered by the source is recorded in the *QoS_level* field.

When a neighboring node receives a QRREQ packet it checks if it has already received it by checking its routing table for source address, multicast group address and sequence number. If it received for the first time, then it checks if the “TTL” field is not equal to zero and if there is at least one common free timeslots between itself and the node that sent the QRREQ packet. If yes, it updates the QRREQ packet by adding its address, capacity, common free timeslots, and increases the hop-count by one and decreases the TTL by one. At this point, there is no bandwidth admission made at each node. After that, it rebroadcasts the QRREQ packet to its neighboring nodes. When a neighboring node receives the QRREQ packet, it performs the previous steps. This process will continue until it reaches a destination node.

Since this is a multicast routing, a destination node can be a forwarding node in the multicast tree. A destination node will continue to flood the QRREQ packet if the value of TTL is not zero. A destination node will wait either for a pre-specified timeout or the reception of a certain number of QRREQ packets. After that, the destination initiates and sends back a QRREP packet including all the information about multiple paths going through it. The packet contains the following fields: (*source, destination, request-id, type, route, cand_timeslots, req-QoS_level, node_cap*), where *type* may be QRREP. The *route* contains the forwarding nodes from the source to the destination. The *node_cap* field contains the capacities of the nodes which are recorded in the *route* filed.

At this point, there is no timeslots reservation. The reservation of the timeslots is done after the multicast source constructs the multiple multicast trees. It is not necessarily that all timeslots which are recorded in the *cand_timeslots* field should be reserved. For this purpose, we called them candidate timeslots. The *req-QoS_level* records the required number of video layers for a destination node which is recorded in the *destination* field. When a node receives a QRREP packet, it checks if one of the next nodes address in the QRREP packet matches its own address. If it does, the node realizes that it is a candidate node on the path to the destination. It then sends the QRREP packet to its upstream node. This process continues until the QRREP packet reaches the multicast source node.

In order to increase the number of discovered paths for each destination and at the same time decreases the broadcast overhead, we adopt the following policy for coping with duplicate QRREQ packet. If a node receives a duplicate QRREQ packet, it checks for its own address if it stored in the QRREQ packet. If it exists, this means that this node has already rebroadcast the QRREQ packet. Therefore, it drops the QRREQ packet. Otherwise; it computes the number of hops from the source node to itself and compares it to the number of hops for the first (previous) QRREQ packet which is stored in its routing table. If the number of hops is less than or equal to the number of hops for the previous packet, the node will update the QRREQ as before and rebroadcast it to its neighboring nodes.

When a source node receives the QRREP packets after a pre-specified timeout; it constructs a partial topology which contains the forwarding nodes, destinations with their capacities and links with their available bandwidth in terms of timeslots. As we see, the multicast source node learns this partial topology through the path discovery and reply phases. Then it constructs multiple multicast trees as it will be described in the next section. After a source node constructs the multiple multicast trees, it sends all the information to the forwarding nodes and destinations. When a forwarding/destination node receives the information sent by the source node, it records in its multicast routing table the source address, multicast group address, its parent (upstream) node, its children nodes (downstream), reserved timeslots and the assigned video layers (MDC_1, MDC_2 and MDC_3). Finally, the multicast connection is established and the source can begin transmitting the video over the multiple multicast trees.

4.2 Joining a Multicast Group

When a node wishes to join a multicast group, it transmits a Join REQuest (JREQ) packet including its address and free timeslots. Each neighboring node receiving a JREQ packet checks if its capacity is sufficient to handle at least one video layer and there is at least one common free timeslot between itself and the node that forwards the JREQ packet. If there is no sufficient capacity or there are no common free timeslots; it simply drops the packet. Otherwise; it forwards the packet after it appends its address, common free timeslots and capacity to the JREQ packet.

When a node on the tree (source, forwarding or destination node) receives a JREQ packet; it computes the number of hop counts from the new destination to the source node and deletes those routes beyond the delay bound (number of hops). After that, this node sends a Join REPly (JREP) packet to its downstream node (reverse path) and appends its address, timeslots and which video layers it can support. When a destination node receives all JREP packets after a pre-specified timeout it constructs the proper path(s) and assigns the required video layers and timeslots to each node on the selected path(s). After that a destination node sends a Join ACTivation (JACT) packet along the selected path(s). Each node on the selected path(s) receiving a JACT packet records their downstream nodes, reserved timeslots and the assigned video layers in its routing table. At the end of the joining process, a destination node becomes ready to receive the required video layers.

4.3 Leaving a Multicast Group

When a leaf destination wishes to leave the multicast group it initiates prune packets and sends it to its upstream nodes and prune it self by deleting all information concerning the multicast group, *i.e.*, source address, multicast group address and releases the reserved timeslots and marks them as free. If the destination is not a leaf node, it cannot leave the multicast group but it can mark itself as a forwarding node. When a node receives a prune packet, it checks in its routing table if it has a downstream node other than the node sending the prune packet. If it has, it cannot prune itself and therefore it should be connected to the tree and drops the prune packet. Otherwise; it prunes itself and sends prune packet to its upstream nodes.

This process continuous until a prune packet arrives at the source node. If a source node receives a prune packet from its downstream node it deletes it from its routing table. After that the source checks if the common timeslots between itself and the deleted downstream node are not reserved with its other downstream nodes. If yes, it releases them and marks them as free. Otherwise; they should be reserved. The process of releasing the reserved timeslots gives the opportunity for other traffics to use these free timeslots. The source node should check the QoS level of the pruned path/paths (QoS level of the pruned destination), if it has the highest QoS level and there are no other destinations have the same QoS level, then the source reduces the number of video layers to the next QoS level.

In order to increase the available bandwidth in the network and decrease the number of transmitted video layers, we propose the following new mechanism. Assume both nodes X and Y, in Fig. 1, carry two descriptions, D_1 receives one description and D_2 receives two descriptions. Let us assume the following two scenarios. In the first scenario, only the destination D_1 sends a prune packet to its upstream node X and prunes itself. When node X receives the prune packet it cannot prune itself and then the prune packet is dropped. But node X should release its common timeslots with destination D_1 because there are no more needed. But if they are common with destination D_2, node X cannot release them.

In the second scenario, only the destination D_2 sends a prune packet to its upstream node X and prunes itself. When node X receives the prune packet it cannot prune itself and the prune packet is dropped. It then releases the common timeslots with D_2, but if the timeslots are common with the destination D_1, node X cannot release them. Note that node X still receives two descriptions and, because it now needs only one description it sends Clear packet (CLR) to node Y. Therefore node X and Y release their timeslots which is reserved for this unneeded description. This method increases the available bandwidth in the network and in the same time minimizes the bandwidth on the tree.

It is important that each node on the multicast tree should maintain a multicast routing table in which it records its downstream nodes with common timeslots and which video layers (MCD_1, MDC_2 and MDC_3) it currently supports.

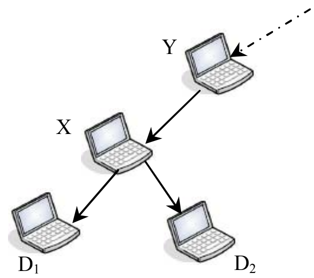


Fig. 1. Example of CLR packet policy.

4.4 Route Repair

We adopt the hard-state approach for maintaining and repairing broken links. In the hard-state approach, the upstream node (the node nearest to the multicast source node) or the downstream node (the node farthest to the multicast source node) is responsible for detecting and repairing broken links.

If the upstream node is responsible for detecting and repairing the broken link, then it can detect it when there is no HELLO packet (HELLO packet is used to exchange information between neighboring nodes, for example, free timeslots) received from its neighbors. On the other hand; if the downstream node is responsible for detecting and repairing the broken, then it can detect a broken link when there is no HELLO packet is received from its neighbors and also if there is no data received from its upstream nodes. In this paper, we adopt the hard-state with downstream node approach for detecting and repairing broken links.

When a downstream node detects a broken link, it floods a Quality REpair (QRE) packet searching its upstream node. The packet contains node address, and its upstream address. The upstream node may receive multiple QRE packets, therefore it selects the path(s) that can support the required number of video layers of its downstream node (the first node that initiates the QRE packet) and then it assigns the video layers and timeslots to each node along the selected path(s). After that it sends back a Quality REPly (QREP) packet including the path information to the node that initiates the QRE packet. Every node along the selected path(s) receiving a QREP packet updates its routing table by recording its upstream node, downstream nodes reserved timeslots and the assigned video layers. Finally, the first node that initiates the QRE packet will receive the QREP packet and rejoin the desired multicast group.

In order to handle broken links in a seamless way and increase the reliability (decreasing the packet loss); a downstream node may deploy a localized prediction technique in which the maintenance process is executed before the link is expected to break [11].

5 Multiple Multicast Trees Construction Algorithms

When the path discovery and reply phases are completed, the next step is to construct multiple multicast trees according to one of our two algorithms. The two algorithms are, Multiple Trees Based on Shortest Path Tree (MSPT) and Multiple Trees based on Steiner Minimum Tree (MSMT).

5.1 Multiple Shortest Path Trees (MSPT) Algorithm

Shortest path tree (SPT) constructs a multicast tree with shortest path from a multicast source node to every destination node. SPT ensures that the end-to-end delay from the multicast source to each destination is the minimum. Single SPT that meet the destinations' requirement (the number of video layers required) may not exist, even though there are enough resources in the network. Thus, MSPT can greatly increases the number of video layers delivered to each destination.

Based on the partial topology, a multicast source constructs MSPT and assigns the video layers to all nodes on the trees. The construction of MSPT, Hybrid-II, and video layers assignment is done as follows. The multicast source node constructs the first multicast tree and assigns all the nodes on the tree the first description (MDC_1). Then it removes all nodes and links with capacity and bandwidth of one, respectively, after that it constructs the second multicast tree for destinations with capacity of two and three and assigns all the nodes on the tree the second video description (MDC_2). Before the third multicast tree is constructed for destinations with capacity of three, all nodes and links (on the first tree) with capacity and bandwidth of one are removed, all nodes and links (on the second tree) with capacity and bandwidth of one are removed, all nodes and links with capacity and bandwidth of two which are on the first and second tree are removed. After that the third tree is constructed and nodes are assigned the third description (MDC_3). See Fig. 9. The construction of node-disjoint and Hybrid-I multiple multicast trees are shown in Fig. 3 and Fig. 6, respectively.

5.2 Multiple Steiner Minimum Trees (MSMT) Algorithm

Steiner minimum tree (SMT) algorithm constructs a multicast tree that spans all the multicast group members with minimum number of links. SMT guarantees certain bound on the end-to-end delay of the multicast tree.

The construction of the MSMT is based on the Steiner tree algorithm described in [12]. MSMT are constructed based on SMT. Video descriptions assignment and MSMT construction are done in the same way of MSPT. Figs. 3, 6, and 9 provide flow diagram for the different types of MSMT construction and video descriptions assignment.

5.3 Types of Multiple Paths

There are two types of multiple paths, disjoint paths and non-disjoint paths. However, there are two types of disjoint paths: node-disjoint and link-disjoint. Node-disjoint paths, also known as totally disjoint paths, do not have any common nodes, except the source and destination. In contrast, link-disjoint paths do not have any common links, but may have common nodes. On the other hand, non-disjoint paths can have nodes and links in common. Refer to Fig. 2 for examples of the different kinds of multiple paths.

Node-disjoint paths offer some advantages over non-disjoint and link-disjoint paths. In non-disjoint and link-disjoint paths, a single node failure can cause multiple paths that share that node to fail. In node-disjoint paths, a single node failure will only cause a single path to fail. Thus, node-disjoint paths offer the highest degree of fault-tolerance.

The main advantage of non-disjoint paths is that they can be more easily discovered. Because there are no limitations that require the paths to be link or node disjoint, more non-disjoint paths exist in a given network than link or node disjoint paths. Because node-disjointedness is a stricter requirement than link-disjointedness, node-disjoint routes are the least abundant and are the hardest to find. In moderately dense networks, there may only exist a small number of node disjoint routes between any two arbitrary nodes, especially as the distance between the nodes increases [13]. This is because there may be sparse areas between the two nodes that act as bottlenecks.

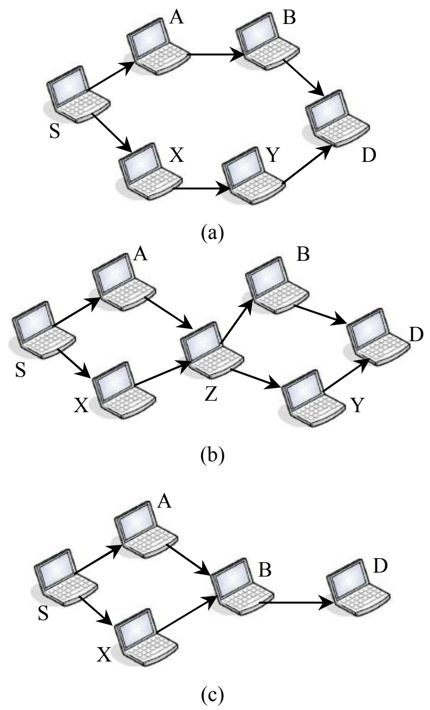


Fig. 2. Different types of multiple paths (a) node-disjoint: paths SABD and SXYD have no nodes or links in common. (b) link-disjoint: paths SAZYD and SXZBD have node Z in common. (c) non-disjoint: paths SABD and SXBD have node B and link BD in common.

5.4 Complexity Analysis of the Algorithms

We analyze the complexity of the proposed algorithms as follows. For MSPT, the shortest path algorithm (Dijkstra's algorithm) is of complexity $O(|V|\log|V|+|E|)\leq O(|V|^2)$ where $|V|$ and $|E|$ is the number of nodes and number of wireless communication links in the partial topology, respectively. Since it iterates $|M|$ times, where $|M|$ is the number of destinations; therefore the complexity is $O(|V|^2\times|M|)$ and finally the algorithm iterates $|C|$ times, where $|C|$ is the value of maximum capacity of the destination set. As a result, the complexity of MSPT is given by $O(|V|^2\times|M|\times|C|)$.

For MSMT, the complexity of the Steiner tree algorithm is $O(|S||V|^2)$ where $|S|$ is the set of multicast group members (source and destination nodes only). Since MSMT iterates $|C|$ times, as a result, the complexity of MSMT is given by $O(|S||V|^2\times|C|)$.

6 Simulations

In this section, we present simulation results obtained from MATLAB[1]. In the following simulation experiments, 2000 topologies are constructed and the value of each point in the various figures is the mean value of the total number of topologies. To fairly compare the proposed algorithms, for each generated random graph, all the proposed algorithms are applied and compared in terms of user satisfaction defined in (8) and the number of forwarding nodes. In addition, at each generated random topology, Breadth-First Search (BFS) is performed to examine if the generated topology is connected (at least there is one path between any two nodes) or not. If the topology is not connected, simply discard it; otherwise continue the simulation. Our metrics of interest are:

- *User satisfaction:* user satisfaction is measured as the total number of the received video layers by all destinations divided by total capacities of all destinations. We define user satisfaction as follows:

$$user\_satisfaction=\sum_{i=1}^{D}N_R(R_i)\Big/\sum_{i=1}^{D}C_n(R_i) \qquad (8)$$

Where, D represents the number of destinations, $N_R(R_i)$ is the total number of received video layers of destination R_i, and $C_n(R_i)$ is the capacity of destination R_i.

- *Number of forwarding nodes:* number of forwarding nodes defined as the number of nodes on the multicast trees except the multicast source and the leaf destinations.

[1]MATLAB is a registered trademark of The MathWorks Inc.

6.1 Simulation Setup

The simulation is performed using the following parameters. **N** represents the number of nodes in the network, **Ω** represents the number of destinations in the network (multicast group size), and r represents the radius of node's *transmission* area ($r = 250m$).

Random graphs are generated in a 1000×1000 square units of a 2-D simulation area, by randomly distributing a certain number of nodes. Once the nodes are placed in the square area and their transmission ranges are decided (each node has the same transmission radius), thus a network graph is generated where two nodes within each other's transmission radius (the distance between them is less than radius r) will have a link. Each node in the network has a capacity of one, two, or three units, where the probability of generating one, two, or three is equal to $1/3$. The multicast source and destinations are randomly chosen such that a multicast source has a capacity of three and the destination nodes are located at least two-hops away from the multicast source. The source node generates three descriptions and the bandwidth required for each description is set to be one timeslot. The number of timeslots at each node is set to be 16. To load the network, random traffics are generated and injected in the network.

After the multicast source nodes receives all the RREP packets, as discussed in the previous section, it constructs a partial topology and then it constructs multiple multicast trees according to one of our algorithms.

In the next sections, we perform two groups of simulations. In the first group, we vary the number of destination nodes (**Ω**) from 5 to 25 in step of 5 and we fix the network size (the number of nodes in the network, **N**) to 50 and 100 nodes. In the second group, we vary the network size from 50 to 100 in step of 10 and we set the multicast group size to 10 and 30. For both groups of simulation, the proposed algorithms were compared in terms of user satisfaction and number of forwarding nodes.

6.1.1 Multiple Node-Disjoint Trees

In multiple node-disjoint tress or totally disjoint trees, there are no common nodes between the multicast trees except the source and destinations. In other words, all paths to each destination are totally node-disjoint. In this simulation, multiple node-disjoint trees are constructed. In multiple node-disjoint trees, forwarding nodes and links are allowed to handle only one description. The first multicast tree is composed of all nodes that have MDC_1. Nodes that have MDC_2 form the second tree. Finally, nodes with MDC_3 form the third multicast tree. Fig. 3 shows the flow diagram of the construction of multiple node-disjoint trees.

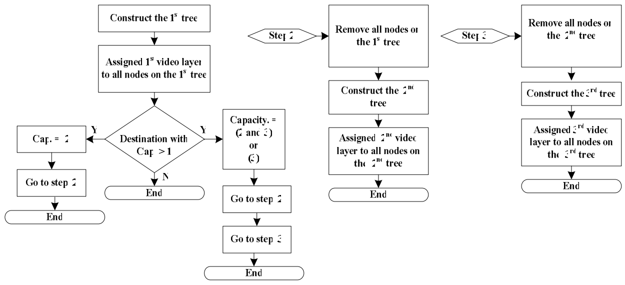
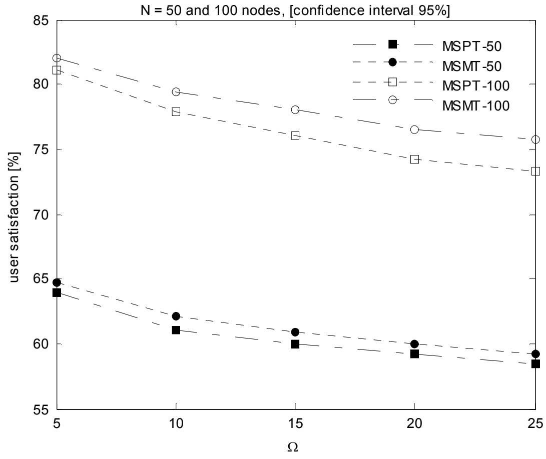


Fig. 3. Construction of multiple node-disjoint multicast trees.

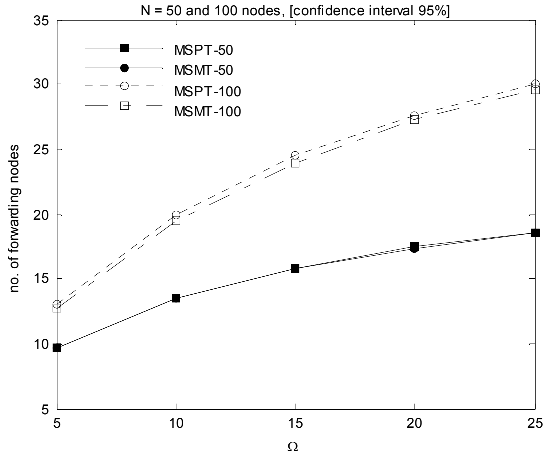
In Fig. 4 we plot user satisfaction and number of forwarding nodes for both algorithms MSPT and MSMT versus the multicast group size (number of destinations). The number of nodes in the network (network size) is set to 50 and 100 nodes. It is clear that user satisfaction drops steadily as the multicast group size increases. This is because the available network resources become less and user satisfaction therefore becomes lower no matter which algorithm is used. As the network size increase from 50 to 100 nodes; user satisfaction increases. This is because the available resources in the network are increased. Statistically speaking, both algorithms have the same user satisfaction.

Fig. 4 (b) shows that as the number of destinations increases; the number of forwarding nodes increases. In addition, as the network size increase from 50 to 100 nodes; the number of forwarding nodes increases. Both algorithms have the same number of forwarding nodes.

Fig. 5 (a) plots user satisfaction versus the number of nodes in the network. Number of destinations is set to 10 and 30 nodes. For both algorithms, as the number of nodes in the network increases user satisfaction ratio increases. This is because the available network resources are increased. When the number of destinations increases from 10 to 30 nodes; user satisfaction decreases. Fig. 5 (b) shows that both algorithms have the same number of forwarding nodes. As the number of destinations increases from 10 to 30 nodes; the number of forwarding nodes increases.

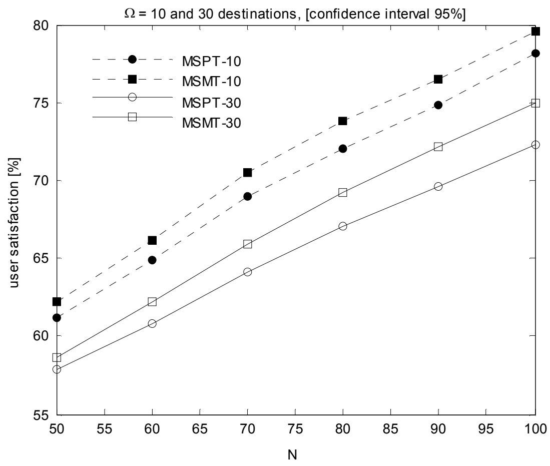


(a)

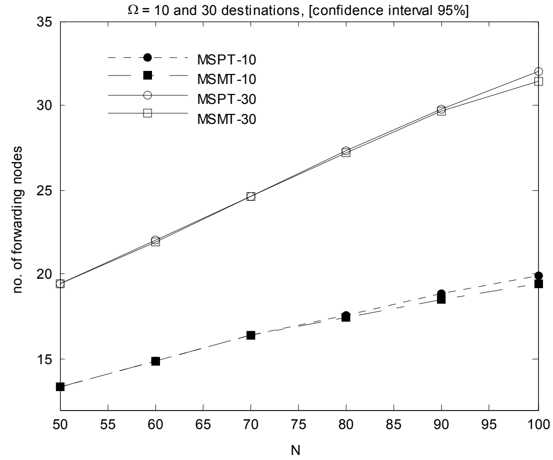


(b)

Fig. 4. (a) User satisfaction and (b) number of forwarding nodes versus multicast group size. The network size is 50 and 100 nodes.



(a)



(b)

Fig. 5. (a) user satisfaction and (b) number of forwarding nodes versus the network size. The number of destination is 10 and 30 nodes.

6.1.2 Multiple Hybrid-I Multicast Trees

In this simulation, we allow nodes to handle more than one description according to its capacities, but links are allowed to handle only one description. Therefore, link and node disjoint paths may be appeared. This will depend on the aggregate resources of paths for each destination. On the other hand, non-disjoint paths will not be existed. Because our multiple multicast trees will have nodes in common and there are no links in common, we called it Hybrid-I. Fig. 6 shows the flow diagram of the construction of multiple Hybrid-I multicast trees.

Fig. 7 (a) shows that user satisfaction increases as the number of network increases from 50 to 100 nodes. On the other hand, for a fixed number of nodes the network, it decreases as the number of destinations increases from 5 to 25 destinations. As the number of destinations increases, the number of forwarding nodes increases. When the number of nodes in the network decreases from 100 to 50 nodes; the number of forwarding nodes decreases. This is shown in Fig. 7 (b).

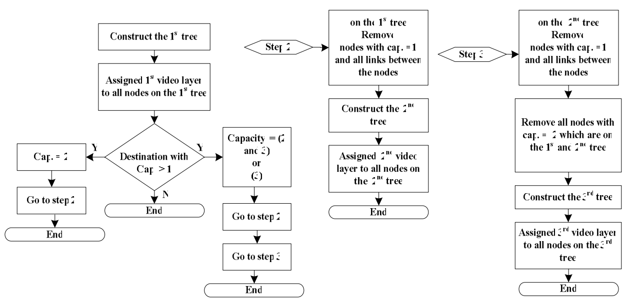


Fig. 6. Construction of multiple Hybrid-I multicast trees.

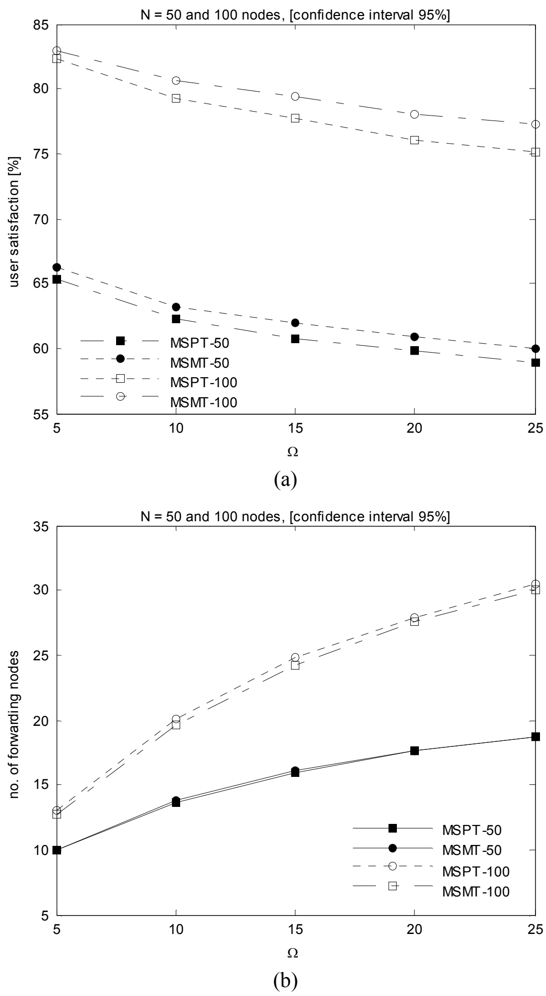


Fig. 7. (a) User satisfaction and (b) number of forwarding nodes versus multicast group size. The network size is 50 and 100 nodes.

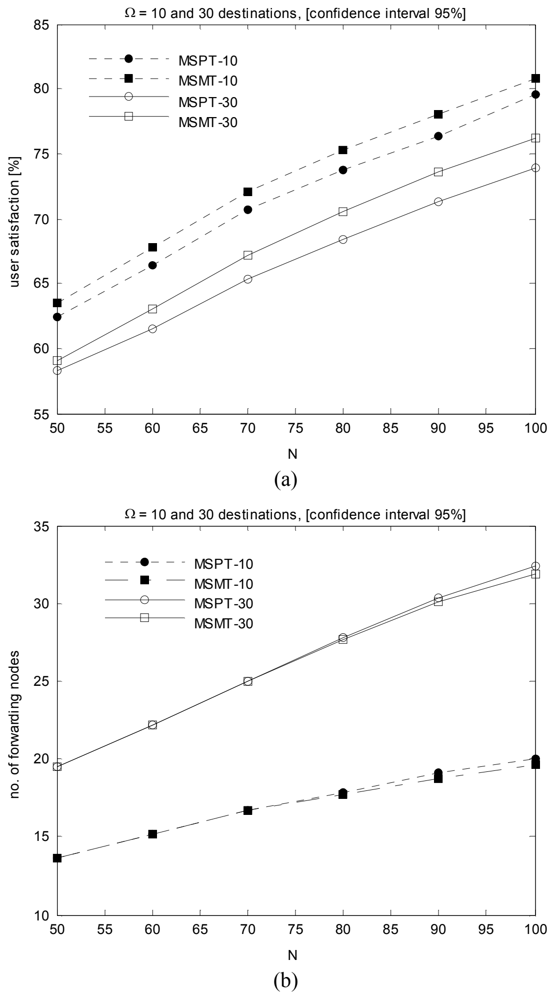


Fig. 8. (a) user satisfaction and (b) number of forwarding nodes versus the network size. The number of destination is 10 and 30 nodes.

Fig. 8 shows the change of user satisfaction and number of forwarding nodes for two set of destinations, 10 and 30, as the number of nodes in the network changes from 50 to 100 nodes.

It is clear, that user satisfaction increases as the number of nodes in the network increases. When the number of destination decreases from 30 to 10 destinations, user satisfaction increases, as depicted in Fig. 8 (a). Fig. 8 (b) shows as the number of nodes in the network increases from 50 to 100 nodes the number of forwarding nodes increases for set of destinations, 10 and 30 destinations. This because the destinations becomes more sparse from the source, therefore more forwarding nodes are needed to connect them to the multicast tree. As the number of destinations decreases from 30 to 10 destinations, the number of forwarding nodes decreases.

6.1.3 Multiple Hybrid-II Multicast Trees

In this simulation, we allow nodes and links to handle more than one description according to their capacities and available bandwidth, respectively. Therefore, non-disjoint paths may be existed. In addition, node and link disjoint paths may be existed. This will depend on the aggregate resources of paths for each destination. Thus, the resulting multiple multicast trees may have nodes and links in common. We called this type, Hybrid-II. Fig. 9 shows the flow diagram of the construction of multiple Hybrid-II multicast trees.

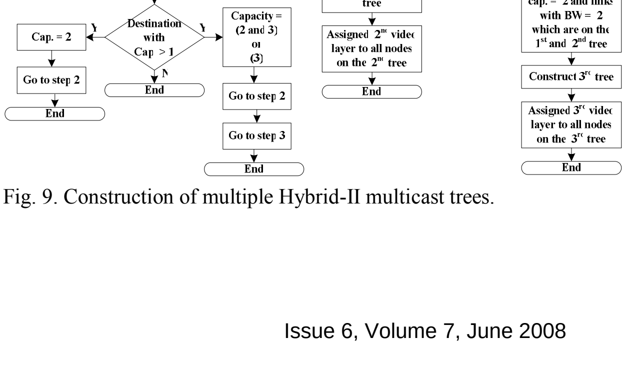
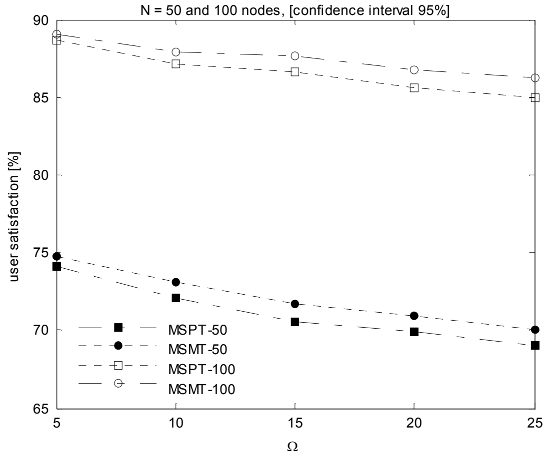
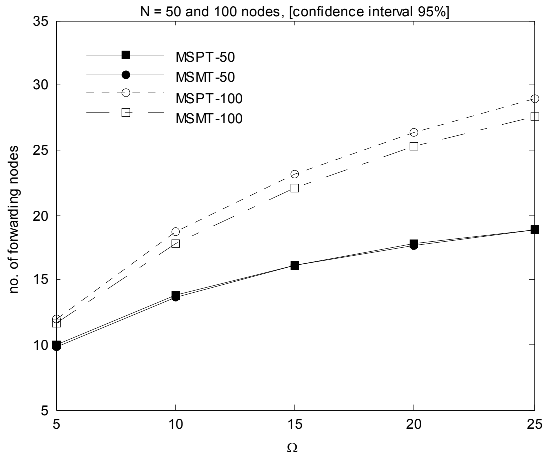


Fig. 9. Construction of multiple Hybrid-II multicast trees.

In Fig. 10 we plot user satisfaction and number of forwarding nodes versus the number of destinations. The number of nodes in the network is set to be 50 and 100 nodes. As in multiple node-disjoint trees and link- disjoint trees, user satisfaction decreases as the number of destinations decreases from 5 to 25 destinations. As the number of nodes in the network increases from 50 to 100 nodes, user satisfaction increases, as shown in Fig. 10 (a). Fig. 10 (b) shows that as the number of destinations increases, the number of forwarding nodes increases. On the other hand, as the number of nodes decreases from 100 to 50 nodes, the number of forwarding nodes decreases.



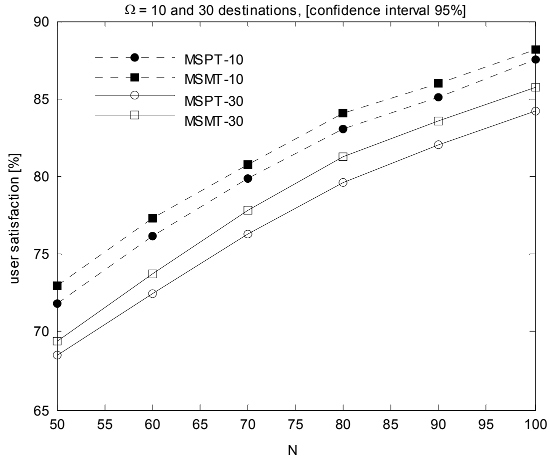
(a)



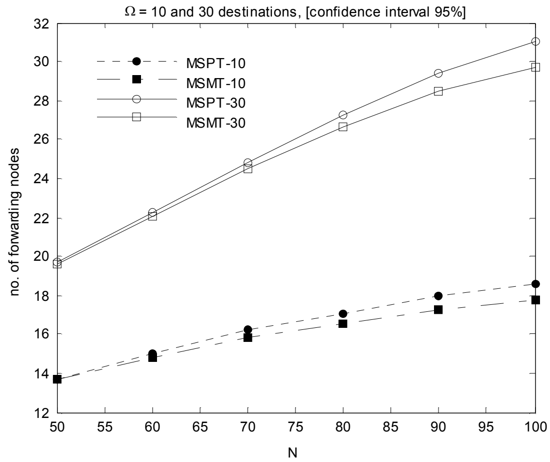
(b)

Fig. 10. (a) User satisfaction and (b) number of forwarding nodes versus multicast group size. The network size is 50 and 100 nodes.

Fig. 11 (a) shows that as the number of nodes in the network increases, user satisfaction increases. This is because the available network resources are increased. When the number of destinations increases from 10 to 30 destinations, user satisfaction decreases, as depicted in Fig. 11 (b).



(a)



(b)

Fig. 11. (a) user satisfaction and (b) number of forwarding nodes versus the network size. The number of destination is 10 and 30 nodes.

6.1.4 Multiple Node-Disjoint vs. Hybrid-I vs. Hybrid-II Multicast Trees

Simulation results demonstrate that, for both algorithms MSPT and MSMT, multiple Hybrid-II multicast trees offers higher user satisfaction than multiple node-disjoint and Hybrid-I multicast trees. This is because nodes and links are allowed to handle more than one description depending on their capacities and available bandwidth, respectively. On the other hand, in multiple node-disjoint multicast trees links and node are allowed to handle only one description. In multiple Hybrid-I multicast trees, nodes are allowed to handle more than one description according to their capacities but links can handle only one description. Clearly, multiple node-disjoint multicast trees have the lowest satisfaction because each node is allowed to handle only one video description and the number of discovered disjoint paths for each destination is small. See Figs. 13-16. The cost for that is the robustness against links failure. In multiple Hybrid-II multicast trees, a single node failure may cause multiple paths to fail and therefore multiple destinations could be affected.

Fig. 12 gives an illustrative example. It shows the discovered paths for the destination D_1. The number of video layers required by D1, according to its capacity, is three. The integer number on each link represents the bandwidth of the link and the capacity of each node is shown inside the circle. It is clear that there is only one node-disjoint path, S→A→ B→ M→ D_1, which can handle only one layer. In case of Hybrid-I, there are two paths, S→A→ B→ N→ D_1 and S→ Z→ W→ B→ M→ D_1, each of them can handle only one layer and therefore D_1 will receive two layers. For Hybrid-II, there are three paths, S→A→ B→ N→ D_1, S→ Z→ W→ B→ M→ D_1 and S→ X→ Y→ B→ N→ D_1 and there for D_1 will receive three layers.

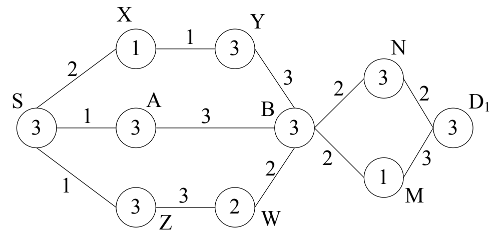
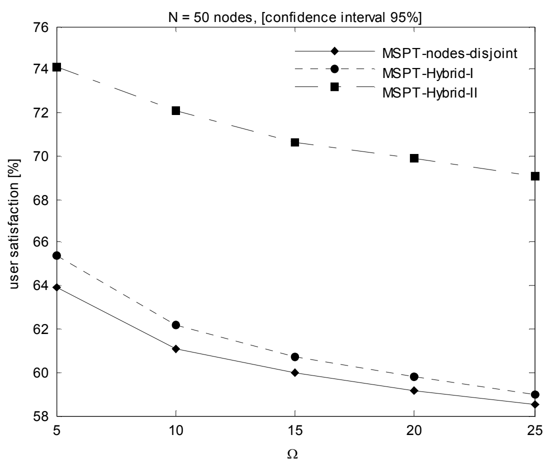


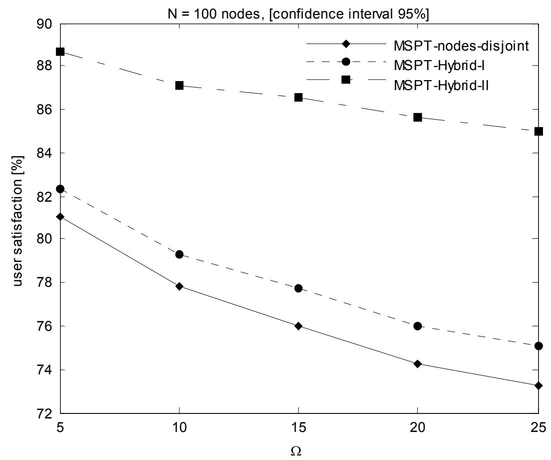
Fig. 12. An illustrative example.

Fig. 13 shows that user satisfaction versus number of destinations for MSPT. The network size is set to be 50 and 100 nodes. Hybrid-II has higher user satisfaction than node-disjoint and Hybrid-I. This is because we allow nodes and links to be on different trees at the same time, therefore the number of available resources in the network is higher than that of link disjoint trees and node disjoint trees. Fig. 12 gives some explanation.

Fig. 14 shows user satisfaction versus the network size for MSPT. User satisfaction increases as the number of nodes increases. Hybrid-II offers higher user satisfaction.

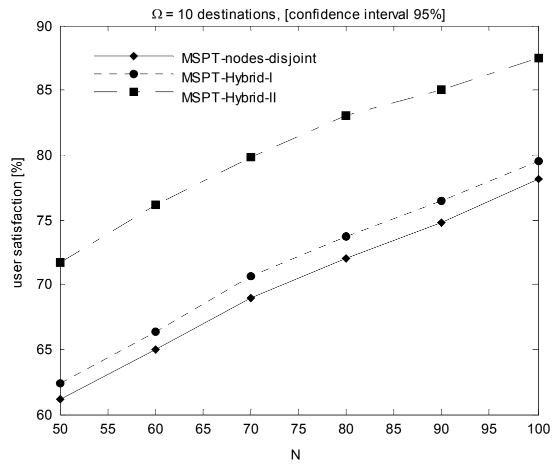


(a)

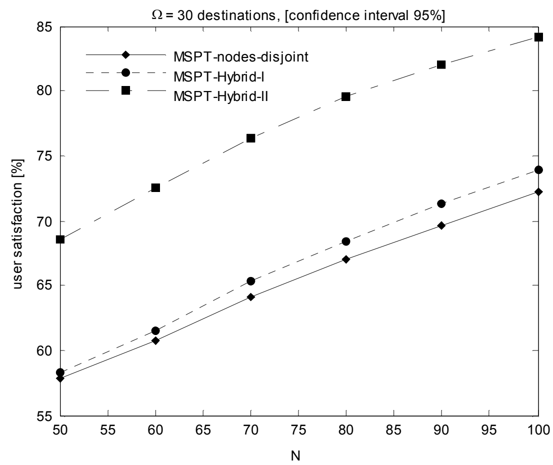


(b)

Fig. 13. User satisfaction for MSPT versus number of destinations (a) N = 50 nodes (b) N = 100 nodes.



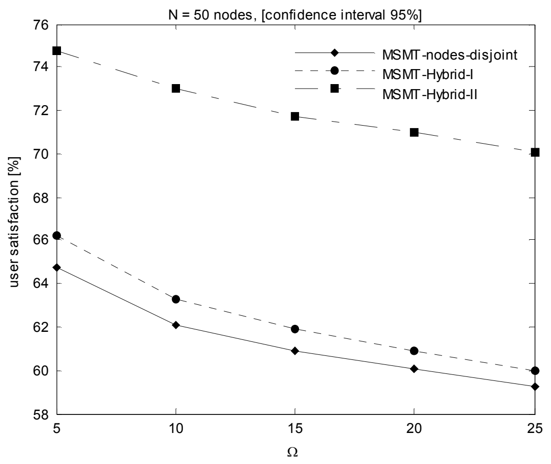
(a)



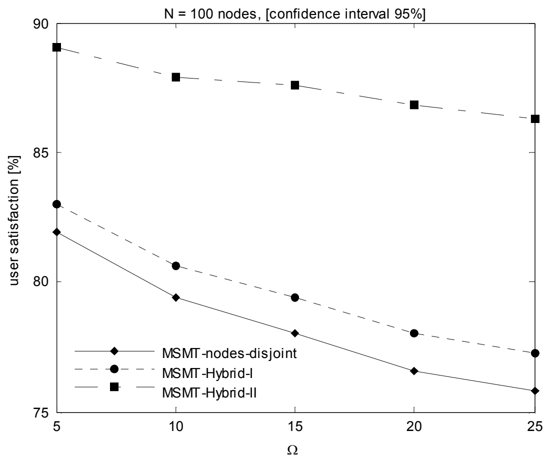
(b)

Fig. 14. User satisfaction for MSPT versus network size (a) $\Omega = 10$ destinations (b) $\Omega = 30$ destinations.

Fig. 15 and Fig. 16 plot user satisfaction, for MSMT algorithm, versus the number of destination and network size, respectively. To avoid repetition, the same conclusions of Fig. 13 and Fig. 14 are applicable for MSMT in Fig. 15 and Fig. 16.

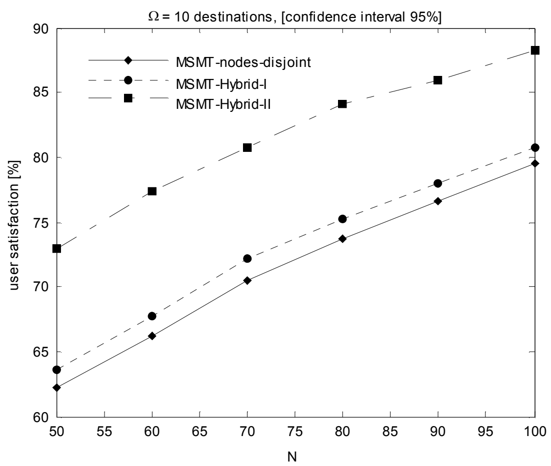


(a)

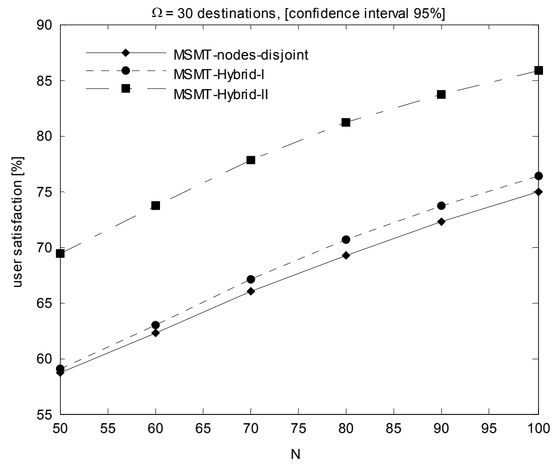


(b)

Fig. 15. User satisfaction for MSMT versus number of destinations (a) N = 50 nodes (b) N = 100 nodes.



(a)



(b)

Fig. 16. User satisfaction for MSMT versus network size (a) $\Omega = 10$ destinations (b) $\Omega = 30$ destinations.

7 Conclusions and Discussions

We have presented the multilayered multicast routing in heterogeneous wireless ad hoc network. Two algorithms for constructing multiple trees to meet the requirements (number of video layers requested) of destination nodes were proposed. The two multilayered multicast algorithms are, MSPT and MSMT. Our proposed algorithms increase user satisfaction by means of finding multiple paths for each destination. Simulation results have demonstrated the effectiveness of our method. In addition, simple video layers assignment was proposed.

In addition, simulation results have demonstrated that multiple Hybrid-II multicast trees offer higher user satisfaction than multiple Hybrid-I and node-disjoint multicast trees. The cost of that is the robustness against link failure. Thus, it is a trade off between providing robustness against path breaks and increasing user satisfaction.

References:

[1] O. B. Michel Kadoch, "Multicast Routing Protocols in Ad Hoc Wireless Networks: A Comparison Survey and Taxonomy," *submitted to ad hoc networks journal,* 2007.

[2] C. S. R. M. B.S. Manoj, "Ad Hoc Wireless Networks: Architectures and Protocols," *Prentice Hall,* 2004.

[3] N. Shacham, "Multipoint communication by hierarchically encoded data,," *INFOCOM,* vol. 3, pp. 2107 - 2114, 1992.

[4] T. B. R. D. Agrawal, and C. Siva Ram Murthy, "Robust Demand-Driven Video Multicast over Ad hoc Wireless Networks," *3rd International Conference on Broadband Communications, Networks and Systems, BROADNETS,* pp. 1-10, , 2006.

[5] W. Z. Wei, A. , "Multiple Tree Video Multicast Over Wireless Ad Hoc Networks," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 17, pp. 2-15, 2007.

[6] S. M. a. X. C. a. Y. T. H. a. H. D. Sherali, "Multiple description video multicast in wireless ad hoc networks," *Mobile Networks and Applications,* vol. 11, pp. 63-73, 2006.

[7] H. I. Chee-Onn CHOW, "Multiple Tree Multicast Ad Hoc On-Demand Distance Vector (MT-MAODV) Routing Protocol for Video Multicast over Mobile Ad Hoc Networks," *IEICE TRANS. COMMUN.,,* vol. E91–B, pp. 428 - 436, 2008.

[8] H. W. X. Jia, "QoS multicast routing by using multiple paths/trees in wireless ad hoc networks," *Ad Hoc Networks* vol. 5, pp. 600-612, 2007.

[9] C. R. L. a. J. S. Liu, "QoS routing in ad hoc wireless networks," *IEEE Sel. Areas Comm.,* vol. 17, pp. 1426-1438, 1999.

[10] L. Hu, "Distributed code assignment for CDMA packet radio networks," *IEEE/ACM Transactions on Networking* vol. 1, pp. 668-677, 1993.

[11] B. S. M. Subir Kumar Das, and C. Siva Ram Murthy, "Weight Based Multicast Routing Protocol for Ad hoc Wireless Networks," *IEEE GLOBECOM,* vol. 1, pp. 117-121, 2002.

[12] G. M. L. Kou, Leonard Berman, "A Fast Algorithm for Steiner Trees," *Acta Informatica,* vol. 15, pp. 141-145, 1981.

[13] Z. Ye, Krishnamurthy, S.V., Tripathi, S.K., "A Framework for Reliable Routing in Mobile Ad Hoc Networks," *IEEE INFOCOM,* vol. 1, pp. 270-280, 2003.

[14] B. M. Wushao Wen, S.-H. Gary Chan, Dipak Ghosal, "LVMSR: an efficient algorithm to multicast layered video," *Computer Networks,* vol. 41, pp. 363-383, 2003.