Multilayered Video Multiple Trees Multicast Routing Algorithms for Heterogeneous Ad Hoc Wireless Networks

Osamah Badarneh[1], Michel Kadoch[1], Ahmed ElHakeem[2]
[1] Department de Génie Electrique, École de Technologie Supérieure
Université du Québec, Montreal, Canada
[2] Department of Electrical and Computer Engineering
Concordia University, Montreal, Canada
E-mail: osamah.badarneh.1@ens.etsmtl.ca, Michel.Kadoch@etsmtl.ca, ahmed@ece.concordia.ca

***Abstract*-In this paper, we address the issue of multilayered multicast routing in ad hoc wireless networks (AHWNs). Existing multilayered multicast protocols assume homogeneous ad hoc wireless networks; in which all nodes are of the same type (they have the same processing capability, data rate and communication capability and characteristics). A more realistic assumption is a heterogeneous network; in which nodes have different processing capability, communication capability and characteristic. In this paper, we assume heterogeneous network; in which nodes have different capacities. Three multilayered multicast routing schemes are proposed, namely, Multiple Trees Based on Shortest Path Tree (MSPT), Multiple Trees based on Pruning Broadcast Minimum Spanning Tree (MPBMST) and Multiple Trees based on Steiner Minimum Tree (MSMT). In our method, we assume that each destination has a preference number of video layers; which is equal to its capacity. The basic idea is to: (i) construct tree(s) that can meet the destinations' QoS requirements, i.e., the number of required video layers (ii) distribute a number of video/audio layers across the nodes. (i) and (ii) are done in a centralized manner, i.e., by the multicast source node. Simulations show that the proposed schemes greatly improve the QoS requirements (improve the user satisfaction ratio) for each destination. In addition, simulations show that multiple-trees schemes achieve substantially higher satisfaction ratio than the single-tree schemes.**

Index Terms: **Multiple-Trees, Multilayer multicast, Heterogeneous ad hoc networks, Hierarchical encoding.**

I. INTRODUCTION

Ad Hoc Wireless Networks (AHWNs) are comprised of fixed or mobile nodes connected by wireless channel without the support of any fixed infrastructure or central administration. The nodes are self-organized and can be deployed "on the fly" anywhere any time to support a certain purpose. Two nodes can communicate with each other if they are within the transmission range of each others; otherwise intermediate nodes can serve as relays (routers) if they are out of range (multi-hop routing). Multicasting plays a crucial role in ad hoc wireless network to support several applications (multimedia application and group meeting, etc).

In traditional multicast routing, all destinations nodes receive the same amount of multicast data. In contrast to traditional multicast, not all destinations nodes in multilayered multicast receive the same amount of data. Each destination node has a preference values for each layer of streams (QoS level) not only according to its available bandwidth but also according to its capacity (how many layers it can process).

A hierarchal encoding technique was proposed for efficient use of resources in heterogeneous networks [1]. This technique is a layered way of encoding information that can appear in different quality levels such as audio and video data. There are two types of hierarchical encoding techniques, namely, Layered coding (LC) and Multiple Description Code (MDC). In (LC) video or audio is encoded into a set of layers, one basic layer and some enhancement layers. The basic layer is enough for decoding the video or audio sequence but in the lowest quality, and the reception of enhancement layers is necessary to decoding higher quality. The l^{th} layer has the data which can further improve the quality of the video decoded from the 1st layer (the basic layer) and the 2nd, ... and $(l-1)^{th}$ layers (the lower extended layers) [2]. (MDC) has been proposed as an alternative to (LC) for streaming over unreliable channels [3-6]. In contrast to (LC), Multiple (MDC) is a coding technique which fragments a single media stream into n independent sub streams $n \geq 2$ referred to as descriptions. The packets of each description are routed over multiple, (partially) disjoint paths. In order to decode the media stream, any description can be used; however, the quality improves with the number of descriptions received in parallel. The idea of (MDC) is to provide error resilience to media streams. Since an arbitrary subset of descriptions can be used to decode the original stream, network congestion or packet loss, which is common in best-effort networks such as the Internet, will not interrupt the stream but only cause a (temporary) loss of quality. The quality of a stream can be expected to be roughly proportional to data rate sustained by the receiver [7, 8].

We investigated, a partial network topology with its link bandwidth (which takes into consideration the capacity of the nodes as we will explain later), the construction of multiple multicast trees to meet the destination capacity requirements with minimum cost.

The rest of the paper is organized as follows. Related work is presented in the next section. In section III, we presented the formulation of the problem and the assumptions. Also, we presented the proposed algorithms and we analyzed their complexity in section III. Our multicast routing protocol (QLMRP) is presented in section IV. Simulation results and analysis are discussed in section V. Finally, section VI concludes the paper.

II. Related Work

Many multicast routing protocols have been proposed for ad hoc wireless networks. Multiple tree protocol called Robust Demand-driven Video Multicast Routing (RDVMR) protocol have been proposed in [9]. RDVMR exploits the path diversity and error resilience properties of Multiple Description Coding (MDC). It constructs multiple trees 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 [10], two multiple tree multicast 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.

Multiple paths/trees in parallel are constructed to meet the QoS requirements [11]. 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.

Link bandwidth nor node capacity (destinations' heterogeneity) are not considered in [9, 10]. Both protocols construct multiple trees and exploit MDC in order to provide error resilience. If one path is broken, packets corresponding to the other description on the other path can still arrive to the destination. In [11] link bandwidth (number of free timeslots) is taken into consideration. This protocol does not support heterogeneous destinations, *i.e.,* it assumes that all destinations must have the minimum required bandwidth and they will receive the same multicast data.

Our method takes into consideration the link bandwidth as well as the nodes capacity. Moreover, not all the destination nodes receive the same multicast data transmitted by a multicast source. Each destination receives a preference number of video layers according to its capacity and link bandwidth.

III. Problem description and Assumptions

A. Network Model

We model the topology of the ad hoc network as a connected graph $G(N, L)$, where N represents a set of wireless nodes each with a random location, denoted by $N = \{1, 2, ..., n\}$ and L represents the set of wireless communication links between nodes. A link between node pair $\{u, v\}$ indicates that both nodes u and v are within each other's transmission range. We assume that all nodes have the same transmission range. In other word, if there is a link $l = \{u, v\}$, $l \in L$, it indicates v is within u's transmission range and u is within v's transmission range. Thus, the corresponding graph will be an undirected graph.

B. Assumptions

In this paper, we consider a session with single multicast source node. The layered video encoder (multicast source) can generate M layers, for simplicity we assume $M = 3$. Nodes in the network have different capacities[1]. There are several factors that can limit the capacity of a node, namely but not limited to, remaining power, number of session participating in and the type of the node (laptop, PDA, etc …).

We always assume that the multicast source has a capacity of three. The transmission of a specific number of layers between two nodes does not depend only on the available link bandwidth between the communicating nodes, but also it depends on the capacity of each node. We assume that an arbitrary link $l\{u, v\}$, $l \in L$, between node pair $\{u, v\}$ have always $BW(M)$, where $M = 3$ represents the number of video layers and $BW(M)$ is the total bandwidth required for the three video layers.

There are two parameters corresponding to each communication link:

I. $C(l)$ is the capacity of link $l, l \in L$, it can be defined as follows:

$$C(l) = \min\{C_n(u), C_n(v)\} \quad (1)$$

Where $C_n(u)$ and $C_n(v)$ represent the capacity of nodes u and v, respectively.

II. $c(l)$ is the cost of each link $l, l \in L$, is set to one. The cost of a path $c(P)$ between two nodes can be defined as:

$$c(P) = \sum_{i=1}^{k} c(l_i) \quad (2)$$

Where k is total number of links on the path P.

C. Topology (Graph) Construction

Our topology is constructed as follows:

(i) We first setup a random graph by creating N nodes whose coordinates are distributed uniformly in a square area 1000*m* X 1000*m* and setting the transmission range of nodes to be $R = 250m$.

(ii) Uniform random capacities are distributed over the nodes in the network, where the probability of generating a capacity of 1, 2 or 3 is equal to $1/3$.

(iii) A wireless communication link is existing between two nodes u and v if the distance between them $d\{u, v\} \leq R$. Thus the cost matrix can be defined as

[1] We define the capacity by how many video layers can be transmitted (encoded) by a node assuming that all nodes have the capability of receiving all the transmitted layers.

$$\cos t\_matrix\{u,v\} = \begin{cases} 1; & \text{if } d\{u,v\} \leq R \\ 0; & \text{otherwise} \end{cases} \quad (3)$$

(iv) A multicast source s is randomly selected and if $C_n(s) < 3$, we set $C_n(s) = 3$.

(v) A number of destinations (multicast group) are randomly picked up from the network graph such that any destination is at least 2-hops away from the multicast source s.

(vi) Construct a directed graph starting from the multicast source and spanning all the destination nodes (partial topology). The directed graph composes of the multicast source node, the multicast group and all the nodes (forwards nodes) that lead to each destination. In other word, the multicast source knows all paths to the destinations. Thus, the directed graph is identified by a virtual (logical) number of levels with a multicast source node as the first level, its neighboring node are in the second level and so on. In addition, the multicast source should know the capacity of every node in the partial topology.

The realization of step (vi) in ad hoc networks can be done as follows:

1. When a source node receives a request from the application layer to set up a multicast connection to a group of destination nodes, it initiates a RREQ (Route REQuest) packet by setting the value of hop-count to zero then broadcasts the RREQ packet to its neighbors.
2. A neighboring node of the multicast source that receives the RREQ packet for the first time, increases hop-count by one, appends its address and capacity to the RREQ packet and rebrocasts the RREQ packet.
3. This process continues until RREQ packets reach all destinations.
4. In order to increase the number of paths to each destination and at the same time to decrease broadcast overhead we use a novel approach for coping with duplicate RREQ packet. According to our approach (how nodes should cope with duplicate RREQ packets), nodes in the network are identified by logical levels and a node cannot be at different levels at the same time (it belongs just to one level). Our approach for level identification works as follows; when a node receives a duplicate RREQ packet it performs the following:

a) If the number of hops recorded in the duplicate RREQ packet is larger than the number of hops recorded in the previous RREQ packet, it drops the RREQ packet because it comes from a lower level or from the same level.

b) If the number of hops recorded in the duplicate RREQ packet is smaller than the number of hops recorded in the previous RREQ packet, it drops the RREQ packet because it comes from an upper level and therefore the node cannot be into two different levels at the same time.

c) If the number of hops recorded in the duplicate RREQ packet is equal to the number of hops recorded in the previous RREQ, it propagates the RREQ packet because it comes from another node which is in the upper levels.

When a destination node receives all the RREQ (or some of them according to a specific policy, for example after a timeout or after it receives a specific number of RREQ packets) packets from its neighboring nodes, it initiates a Rout REPly (RREP) packets, appends its address and its capacity. RREP packets are sent toward the source in the reverse path which is discovered during the propagation RREQ packets.

When the multicast source receives all RREP packets (or some of them according to a specific policy, for example after a timeout or after it receives a specific number of RREP packets) sent by each destination, it can construct a partial topology of the network (directed graph in step (vi)). The multicast source can identify the level of each node by checking the number of hop-count for each RREP packet. The number of hop-count is equal to destination's level number; the parent of the destination is one level before and so on until the multicast source finish all the nodes recorded in the RREP packet. According to the policy of how dealing with duplicate RREQ packet, all paths which lead to the same destination has the same number of hops (same length). Following section describes the construction of multiple multicast tree and video layers distribution.

D. Multiple Trees Multicast Routing Algorithms

In this section, we present three algorithms for constructing multiple trees for multicast video layers transmission and describe the distribution of video layers among different multiple trees. Our goal is to construct multiple trees to maximize the USR defined (5) such that NoCR defined in (6) is minimized. The three algorithms are, Multiple Trees Based on Shortest Path Tree (MSPT), Multiple Trees based on Pruning Broadcast Minimum Spanning Tree (MPBMST) and Multiple Trees based on Steiner Minimum Tree (MSMT).

D.1 Multiple Trees Based on Shortest Path Tree (MSPT)

Shortest path tree constructs a multicast tree with shortest path from a multicast source node to every destination node. Single shortest path tree that meet the requirement (the number of video layers) may not be existed, even though there is enough capacity in the network. Thus, MSPT can greatly increases the number of video layers delivered to each destination. The construction of the SPT is based on Dijkstra's shortest path algorithm.

As described in previous section (III.C topology construction), the multicast source has the full information about the partial topology (sub-graph) that contains all the destinations and forwarder nodes. A multicast source constructs a SPT for all destinations and assigned the first video layer to each node on this tree. After that it checks if there are any destinations of capacity of two or three if yes, it removes all nodes (including the destination nodes), which are on the first SPT, with capacity of one and delete their links because these nodes are already assigned the first video layer and they cannot transmit any more video layers. After that, the multicast source constructs the second SPT and assigned every node (on the second SPT) the second video layer. Finally, the multicast source node checks if there is any destination with capacity of three, if yes, it removes all nodes (including the destination nodes), which are on the second SPT, with capacity of one and delete their links for the reason mentioned before. In addition, it removes all nodes (including the destination nodes) with capacity of two that are participating in the first SPT because these nodes are already assigned the first and second video layer and they cannot transmit any more video layers. But nodes with capacity of two on second SPT which are not on the first SPT, they should remain connecting because they still have a capacity of one, thus these nodes can participate in the construction of the third SPT if it performed. After that, the multicast source checks if there still destination with capacity of three, if yes, it constructs the third SPT and terminates. Each times the multicast source check for capacity of two or three and the result is there is no more destination, it assigned the required number of video layers and terminates.

The previous procedure of constructing multilayered multicast tree and video layers assignment is the same for the next two algorithms. Fig. 1 below provides flow diagram for multiple trees construction and video layers assignment.

D.2 Multiple Trees based on Pruning Broadcast Minimum Spanning Tree (MPBMST)

In this algorithm, the construction of the multicast tree is as follows. First, a minimum broadcast tree is constructed as in the well-known Prim's algorithm [12] that spanning all the destinations and forwarder nodes. The broadcast tree is then pruned by eliminating all the paths that are not needed to reach the members of the multicast group. The construction of the second tree and third tree depends on the capacity of the multicast group members as discussed in the previous section.

D.3 Multiple Trees based on Steiner Minimum Tree (MSMT)

Steiner minimum tree algorithm constructs a tree (multicast tree) that spans all the multicast group members with minimum number of links. The construction of the MSMT is based on the Steiner tree algorithm described in [13]. Again, the construction of the second tree and third tree depends on the capacity of the multicast group members as discussed in the previous section.

D.4 Algorithms Complexity

We analyze the complexity of the three 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 (multicast group). As a result, the complexity of MSPT is given by $O(|V|^2\times|M|\times|C|)$.

For MPBMST, the minimum spanning tree algorithm (Prim's algorithm) is a complexity of $O(|V|^2)$, the pruning process is a complexity of $O(|V|)$, therefore the complexity is $O(|V|^2+|V|)=O(|V|^2)$ and finally, the algorithms iterates for $|C|$ times. As a result, the complexity of MPBMST is given by $O(|V|^2\times|C|)$.

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 $O(|S||V|^2\times|C|)$. Fig. 2(a) and Fig. 2(b) show the variation of normalized complexity with the variation of multicast group size and network size, respectively.

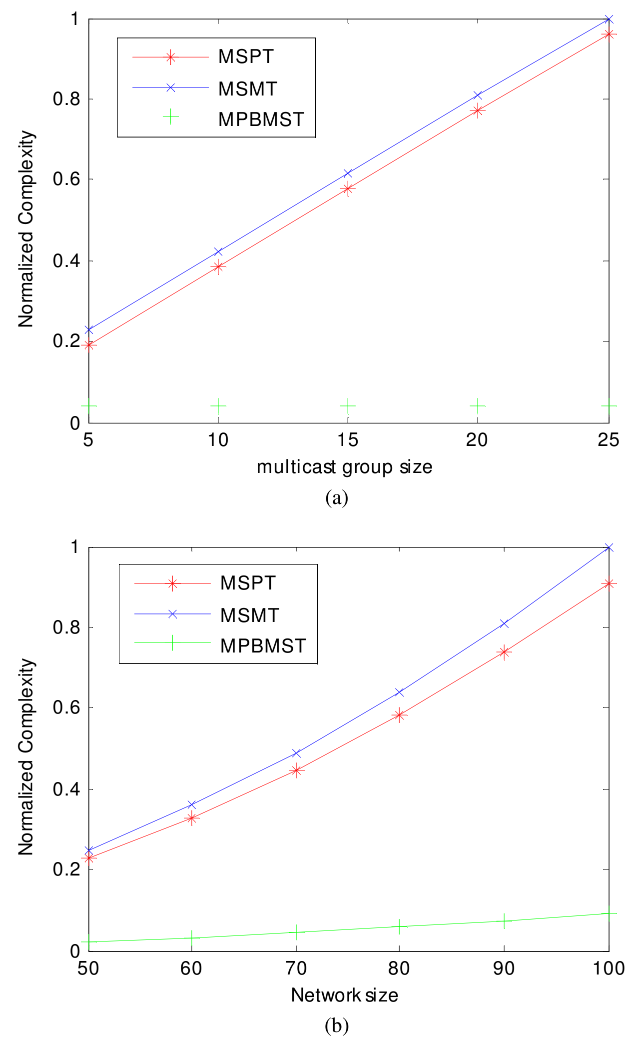


Fig. 2 Normalized complexity against (a) Multicast group size and (b) Network size.

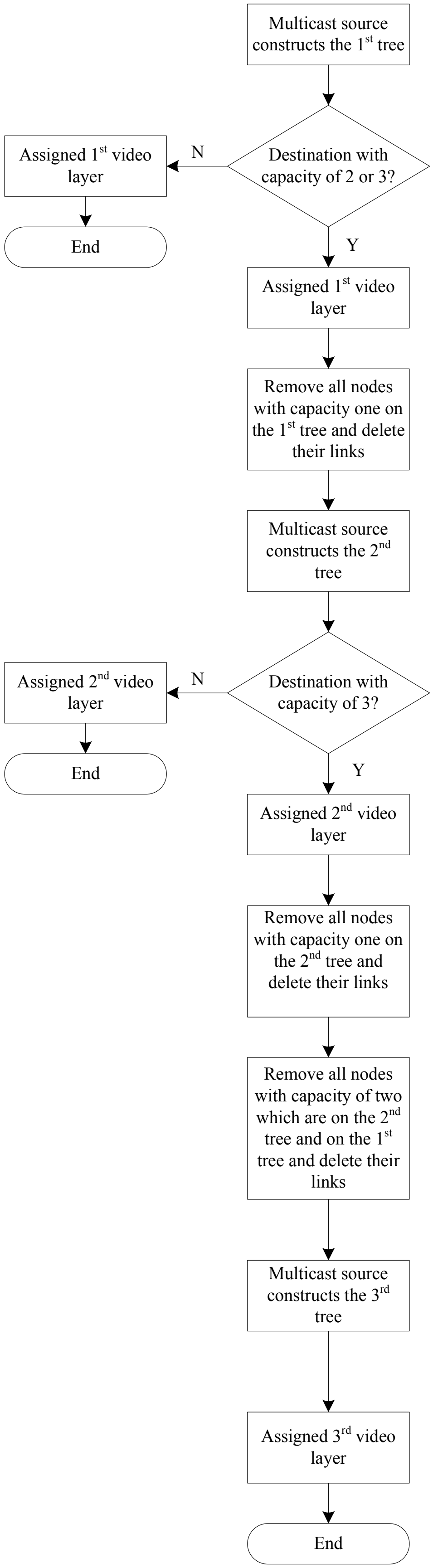


Fig. 1 Flow diagram for multiple trees construction and video layers assignment.

IV. QoS LAYERED MULTICAST ROUTING PROTOCOLS (QLMRP)

QLMRP is an on-demand multicast routing protocol. The multicast source node constructs tree(s) whenever it has a multicast data ready for transmission. The construction of tree(s) is performed using one of the aforementioned algorithms.

A. Path Discovery and Multiple Trees construction

In QLMRP, paths are discovered on-demand by propagating the Quality Route REQuest (QRREQ) packets and Quality Route REPply (QRREP) packets between the source node and all destinations. When a source node receives a request from the application layer, it determines the number of QoS levels (number of video layers) can be supported. After that it initiates and broadcasts a Quality Route REQuest packet (QRREQ), which includes the QoS levels and end-to-end delay in terms of number hops (TTL). When a neighbor node receives the QRREQ packet it checks if it received a QRREQ packet before 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 it has sufficient capacity. If yes, it updates the QRREQ packet by adding its address, capacity, and increases the hop-count by one and decreases the TTL by one. After that, it rebroadcasts the QRREQ to its neighboring nodes. If a duplicate QRREQ packet is received, a method described in sec. III is performed.

When any neighboring node receives the QRREQ packet, it follows the previous steps. This process continuous until all destinations receives the QRREQ packets. After each destination node receives all QRREQ packets from its neighboring nodes after a pre-specified timeout, it initiates a QRREP packet that contains all the routing information and sends it to the source node.

When a source node receives all the QRREP packets after a pre-specified timeout; it constructs a partial topology which contains the forwarder nodes and destinations with their capacities. After that, it checks the highest QoS level required by examine the capacity of each destination. Based on the highest QoS level, a multicast source determines the number of trees to be constructed in order to meet the QoS requirements. Multiple trees construction is performed as described in previous sections.

After a source node constructs the multiple trees, it sends all the information to the forwarder nodes and destinations. When a forwarder node receives the information sent by the source node, it records in its multicast routing table the source address, multicast group address, its parent node, its children and the assigned video layers (MDC1, MDC2 or MDC3). As mentioned before, MDC generates multiple independent layers with same priority; if any layer is lost the data can be decoded using other layer. In contrast, layered coding generates multiple independent layers but 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.

Finally, the multicast connection is established and the source can begin transmitting data over the chosen multiple trees.

B. Leaving a multicast Group

When a destination node (leaf node) wishes to leave the multicast group it initiates prune packet and sends it to it's upstream nodes and prune it self by deleting all information concerning the multicast group, *i.e.*, source address, multicast group address. If not a leaf node, it cannot leave the multicast group but it can mark itself as a forwarder node. When a node receives a prune packet, it checks in its routing table if it has a down stream node other than the node sending the prune packet. If it has, it cannot prune itself and therefore it should be connected on the tree and then it drops the prune packet. Otherwise; it prune 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. The source node should check the QoS level of the pruned path/paths, if it has the highest QoS level and there is no other receivers have the same level, then the source reduces its encoding to the next QoS level.

C. 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, then it can detect a broken link, for example, when there is no HELLO packet is received from its neighbors. On the other hand; if the downstream node is responsible for detecting and repairing, 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. The time needed for data transmission between nodes is much smaller than the period of HELLO packet, therefore; the downstream node will detect the broken link faster than the upstream node. Therefore, we propose a hard-state with downstream node approach.

When a downstream node detects a broken link, it floods a Quality Repair (QRER) packet searching its upstream node. The packet contains, node address, its upstream address. The upstream node may receive multiple QRER, therefore it selects the path(s) that can support the required number of video layers of its downstream node (the first node that sends the QRER packet) and then it assigns the video layers 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 QRER packet. Every node along the selected path(s) receiving a QREP packet updates its routing table by recording its upstream node, downstream nodes and the assigned video layers. Finally, the first node that initiates the QRER 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 [14].

D. Joining a Multicast Group

When a node wishes to join a multicast group, it transmits a Join REQuest (JREQ) packet including its address and multicast group address. Each node receiving a JREQ packet checks if it the available link bandwidth and its capacity are sufficient to handle one MDC at least. If there is no sufficient bandwidth or capacity; it simply drops the packet. Otherwise; it forwards the packet after it appends its address and capacity to the JREQ packet. When a node on the tree (forwarder node or destination node) receives a JREQ packet; it sends a Join REPly (JREP) packet to its downstream node (reverse path) and appends which video layers can be supported. When a destination node receives all JREP packets after a pre-specified timeout it constructs the proper path(s) and assigned the required video layers to each node on the selected path(s). After a destination node send Join ACTivation (JACT) packet along each path(s). Each node on the selected path(s) receiving a JACT packet records their downstream nodes and the video layers it currently save in its routing table. At the end of the joining process, a destination node becomes ready to receive the required video layers.

V. Simulation Results and Discussion

In order to evaluate the performance of the proposed algorithms, extensive simulations have been conducted and compared. In the following simulation experiments, at each simulation point the simulation runs 2000 times, *i.e.,* 2000 topology are constructed, and the value of each point in the various figures is the mean value of the total number of simulation runs. To fairly compare the proposed algorithms, for each generated random graph (topology), we apply all the algorithms proposed (multiple/single trees algorithms) and we calculate USR and NoCR in equations (5) and (6), respectively. In addition, at each simulation run, *i.e.,* 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 from a multicast source to every node in the network) or not. If the topology is not connected, simply discard it; otherwise continue the simulation steps in section III.C. Our metrics of interest are:

- User satisfaction ratio (USR): user satisfaction ration is measured by a fraction of the number of the requested video layers by users and the number of the received video layers. We define USR as follows:

$$failure\_rate = \frac{\sum_{i=1}^{D}\{C_n(R_i) - N_R(R_i)\}}{\sum_{i=1}^{D} C_n(R_i)} \quad (4)$$

Where, D represents the number of destination, $C_n(R_i)$ is the capacity (number of requested video layers) of the destination R_i, and $N_R(R_i)$ is the total number of received video layers of destination R_i, and $\sum_{i}^{D} C_n(R_i)$ is the total capacity of all destinations (*i.e.,* the total number of requested video layers); thus USR is given by:

$$USR = (1 - failure\_rate) \times 100\% \quad (5)$$

- Node Cost Ratio (NoCR): node cost ratio is measured by fraction of the nodes on the multicast trees (multicast source, forwarder nodes and destination nodes) by the network size. NoCR is given by:

$$NoCR = \left\{ \frac{\sum_{t=1}^{T} \sum_{i=1}^{K} n_{ti}}{network\_size} \right\} \times 100\% \quad (6)$$

Where, T is the total number of tress, K is the total number of nodes in the network (network size) including the multicast source and the destinations, n_{ti} is given by:

$$n_{ti} = \begin{cases} 1; & \text{if the node } n_i \text{ on the tree } t \\ 0; & \text{otherwise} \end{cases} \quad (7)$$

If a node participates on more than one tree, we count it once.

In the next sections, we perform two groups of simulation. In the first group, we vary the number of destination nodes (multicast group size) from 5 to 25 and we fix the network size (the number of nodes in the network) to 50 nodes. In the second group, we vary the network size from 50 to 100 and we also vary the multicast group size from 10 to 30. For both groups of simulation, the proposed algorithms for multiple/single trees were compared in terms of USR and NoCR

A. User Satisfaction Ratio and Node Cost Ratio versus Multiple Trees and Multicast Group Size

In this experiment, the multicast group size is simulated against two parameters; user satisfaction ratio (USR) and node cost ratio (NoCR). The parameter set for the simulation is listed in table I.

Fig. 3 plots the corresponding USR and NoCR, respectively, of the three algorithms. Fig. 3(a) shows the changes of USR with different multicast group sizes. MSMT achieves higher USR than MSPT and MPBMST. MSPT and MPBMST have the same USR. As the number of multicast group size increases, USR shows insignificant variation. The overall USR of the proposed algorithms is quit high (80%). Fig. 3(b) shows the linearity between NoCR and multicast group size. In terms of NoCR, MSMT performs better than MSPT and MPBMST. MSPT and MPBMST have the same NoCR.

TABLE I
SIMULATION PARAMETER

Parameter	Value
Area	1000*m* X 1000*m*
Transmission range (R)	250*m*
Network size	50 nodes
Multicast group size	[5, 10, 15, 20, 25] destinations

B. User Satisfaction Ratio and Node Cost Ratio versus Multiple Trees and Network Size

In this experiment, we compare USR and NoCR of the MSPT algorithm versus the number of nodes in the network (network size). The parameter set for the simulation is listed in table II.

TABLE II
SIMULATION PARAMETER

Parameter	Value
Area	1000*m* X 1000*m*
Transmission range (R)	250*m*
Network size	[50, 60, 70, 80, 90, 100] nodes
Multicast group size	[10, 20, 30] destinations

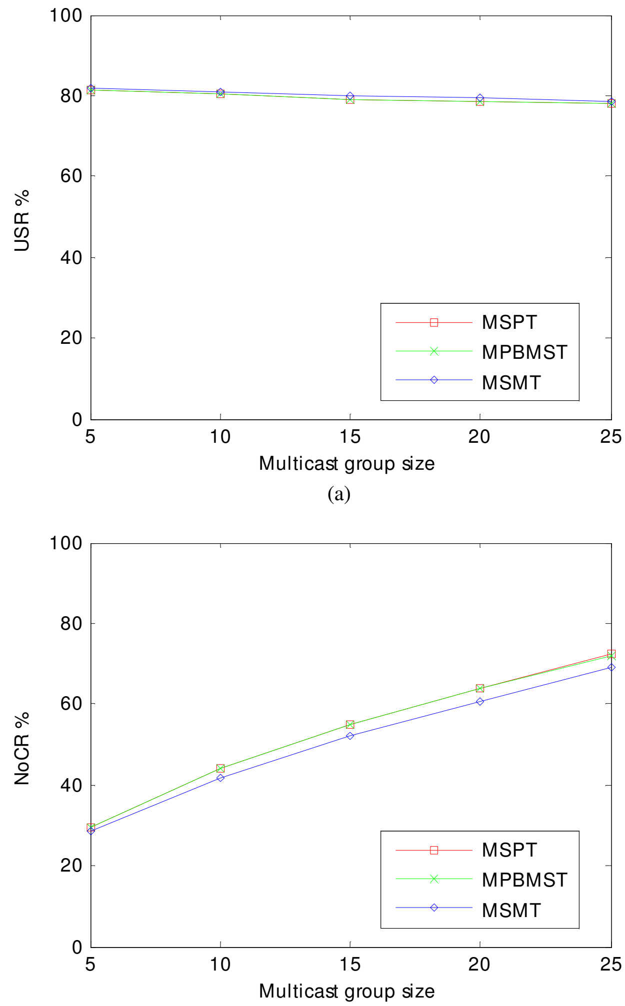
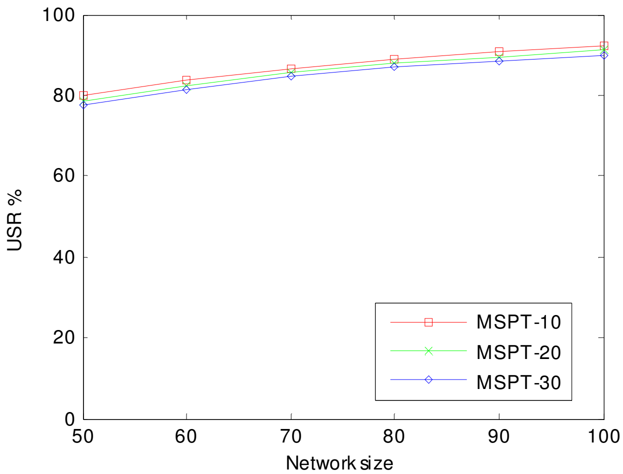


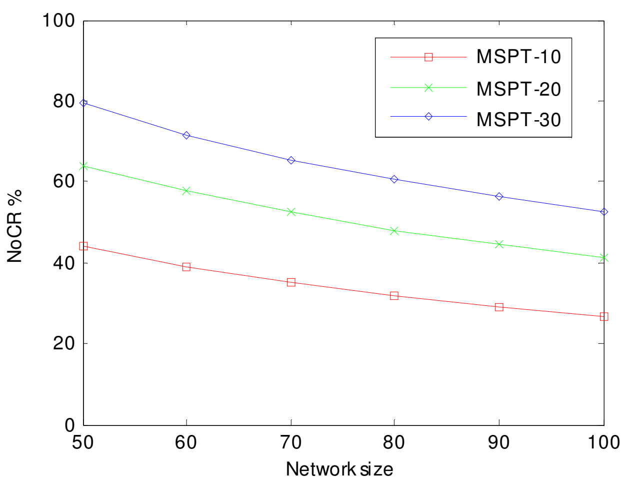
Fig. 3 User satisfaction ratio and node cost for MSPT, MPBMST and MSMT versus multicast group size. Network size is fixed and equal to 50.

Fig. 4(a) shows that for a fixed number of destinations (multicast group size), as the network size increases the USR increases. In addition, as the multicast group size increases from 15, 20 to 30, the USR decreases. Fig. 4(b) illustrates that the number of nodes, which is required to construct multiple shortest path trees in order to meet the number of video layers required by destinations, decreases as the network size increases. It is clear that as the multicast group size increases the number of nodes that are required to construct the multiple trees increases. For example, in order to achieve an USR of 95%, see Fig. 4(a) for 10 destinations in a network that contains 100 nodes, we need about 30% of the network size (30 nodes, *i.e.,* one multicast source, 10 destinations and 19 forwarder nodes) to build multiple trees. On the other hand, 55 nodes (55% of the network size), *i.e.,* one multicast source, 30 destinations and 24 forwarder nodes are required to construct multiple trees.

Fig. 5 and Fig. 6 show the simulation results of MPBMST and MSMT for the same parameters which are listed in table II. The same conclusions of Fig. 4 are also applicable for both figures (Fig. 5 and Fig. 6).



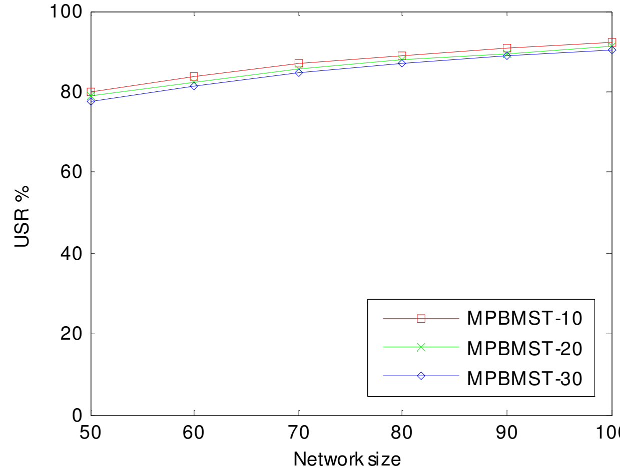
(a)



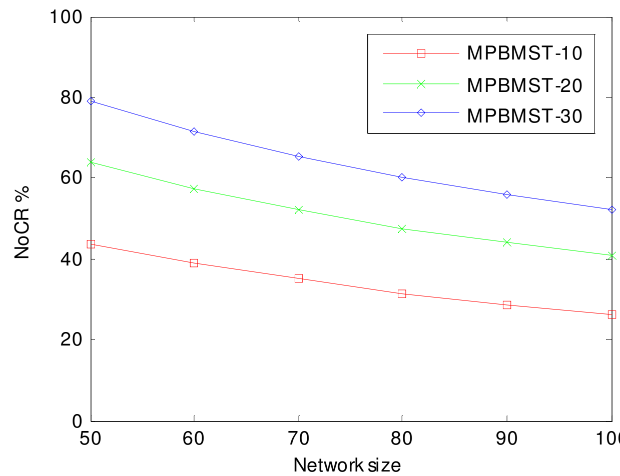
(b)

Fig. 4 User Satisfaction Ratio and node cost ratio for MSPT versus network size. Multicast group size is equal to 10, 20, and 30.

In figures 7, 8 and 9 we compare the USR and NoCR of the algorithms versus the network size. Fig. 7(a) shows that the MSMT achieves higher USR than the other algorithms. MSPT and MPBMST algorithms achieve the same USR. Fig. 7(b) shows that both algorithms MSPT and MPBMST have the same NoCR and they offer higher cost NoCR than MSMT algorithm.

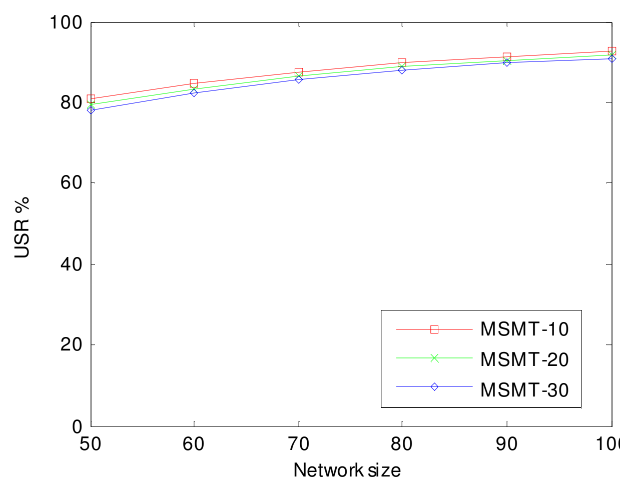


(a)

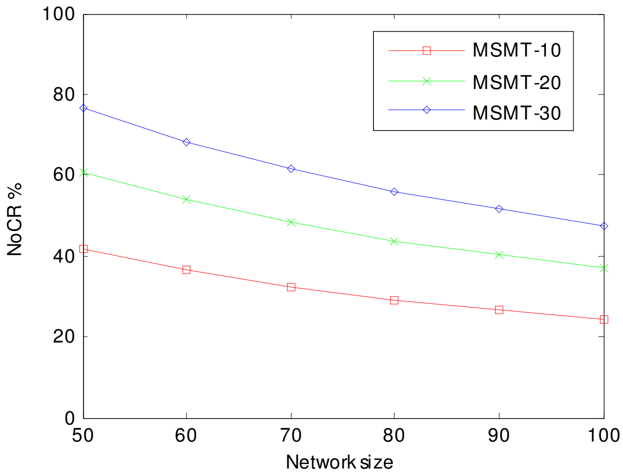


(b)

Fig. 5 User Satisfaction Ratio and node cost ratio for MPBMST versus network size. Multicast group size is equal to 10, 20, and 30.

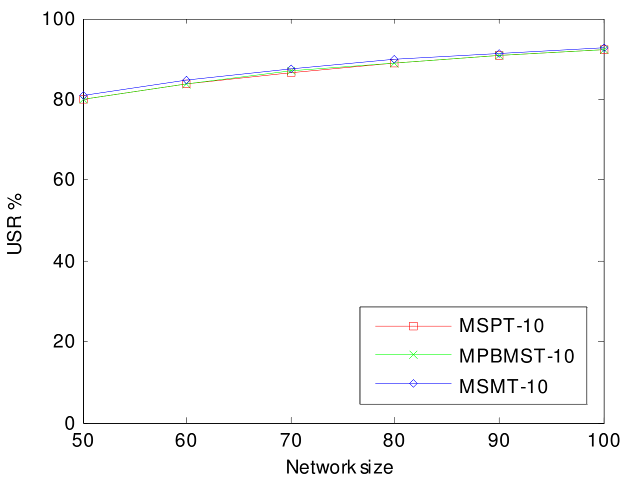


(a)

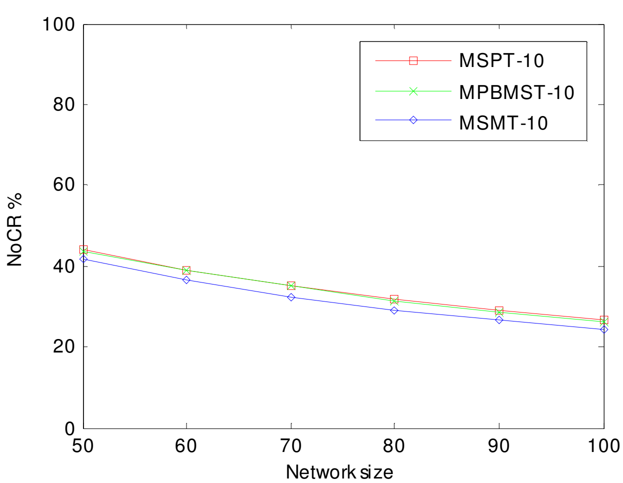


(b)

Fig. 6 User Satisfaction Ratio and node cost ratio for MSMT versus network size. Multicast group size is equal to 10, 20, and 30.

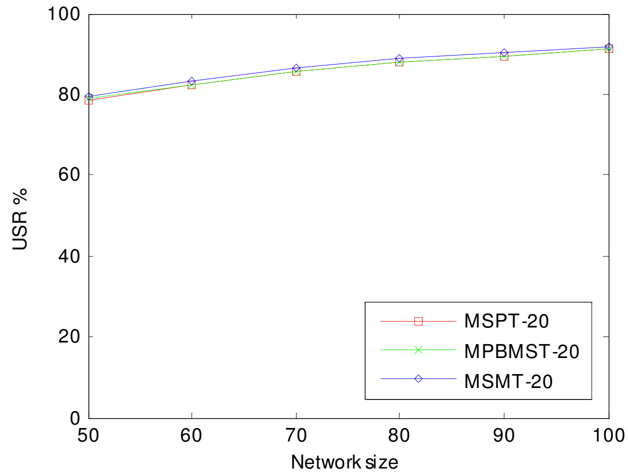


(a)

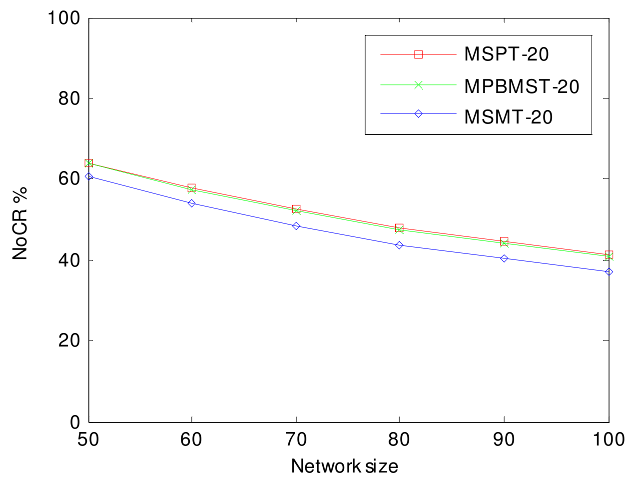


(b)

Fig. 7 User Satisfaction Ratio and node cost ratio for MSPT, MPBMST and MSMT versus network size. Multicast group size is fixed and equal to 10.

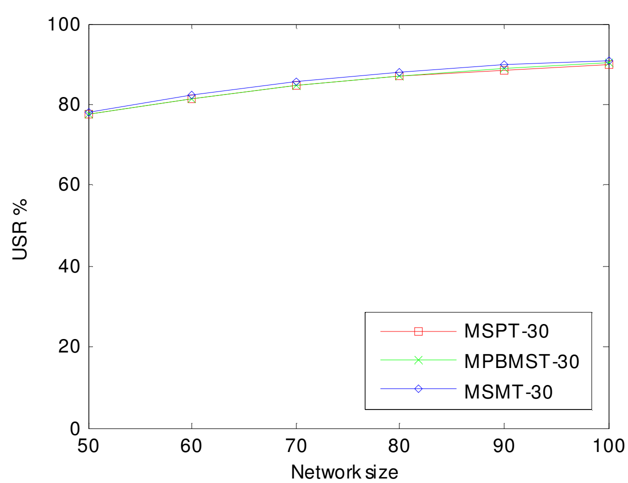


(a)

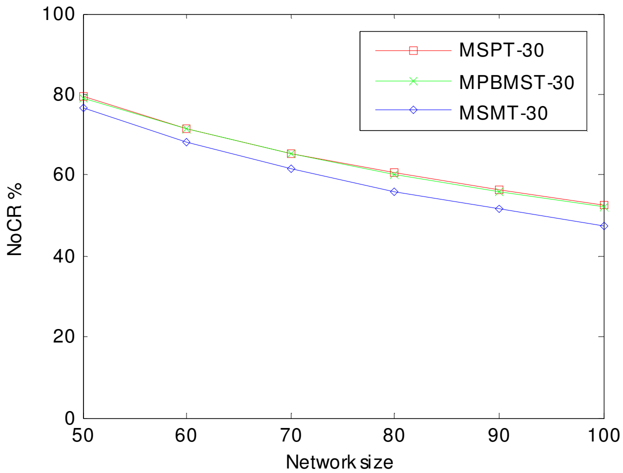


(b)

Fig. 8 User Satisfaction Ratio and node cost ratio for MSPT, MPBMST and MSMT versus network size. Multicast group size is fixed and equal to 20.

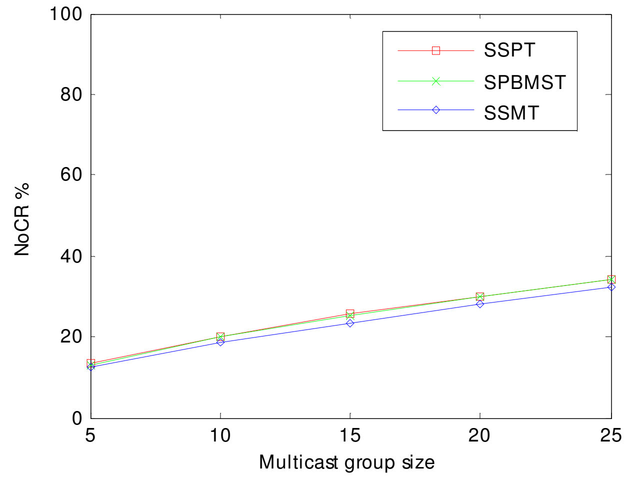


(a)



(b)

Fig. 9 User Satisfaction Ratio node cost ratio for MSPT, MPBMST and MSMT versus network size. Multicast group size is fixed and equal to 30.



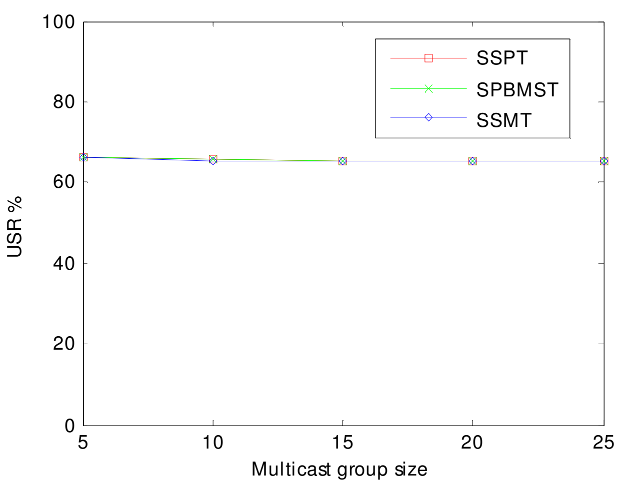
(b)

Fig. 10 User satisfaction ratio and node cost ratio for SSPT, SPBMST and SSMT versus multicast group size. Network size is fixed and equal to 50.

C. User Satisfaction Ratio and Node Cost Ratio versus Single Tree and Multicast Group Size

In this experiment, we compare USR and NoCR of the Single Shortest Path Tree (SSPT), Single Pruning Broadcast Minimum Spanning Tree (SPBMST) algorithm and Single Steiner Minimum Tree (SSMT) algorithm versus the multicast group size. We use the same simulation parameter in table I.

The three algorithms have the same USR (about 65%), but SSMT have less NoCR than SSPT and SPBMST. This is shown in Fig. 10.



(a)

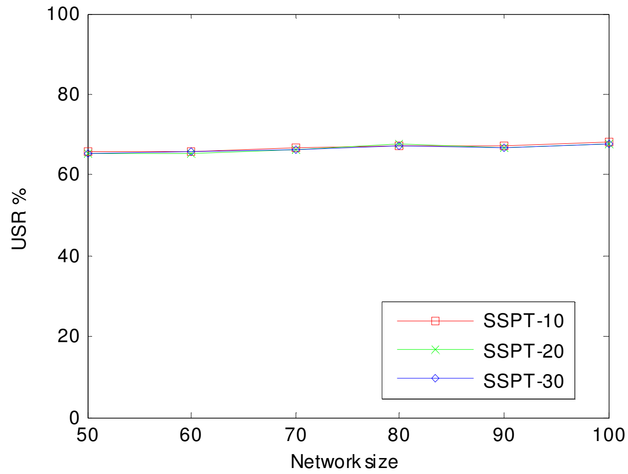
D. User Satisfaction Ratio and Node Cost Ratio versus Single Tree and Network Size

In this experiment, we compare USR and NoCR of the SSPT algorithm versus the number of nodes in the network (network size). The parameter set for the simulation is the same as in table II.

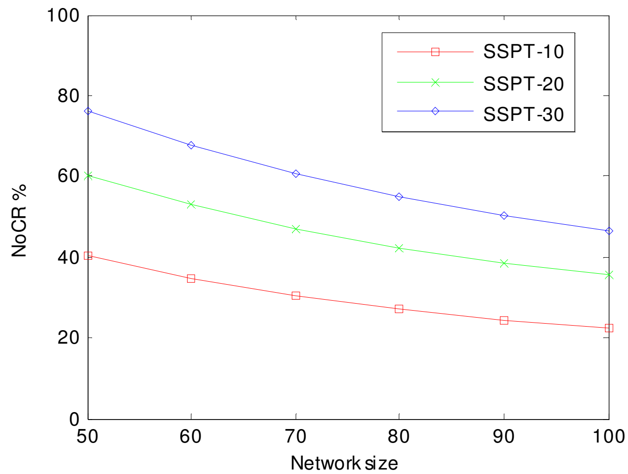
Fig. 11(a) shows that for a fixed number of destinations (multicast group size), as the network size increases the USR insignificantly increases. In addition, as the multicast group size increases from 15, 20 to 30, the USR stays unchanged. Fig. 11(b) illustrates that the number of nodes, which is required to construct single shortest path tree in order to meet the number of video layers required by destinations, decreases as the network size increases. It is clear that as the multicast group size increases the number of nodes that are required to construct a single tree increases. For example, in order to achieve an USR of 65% for 10 destinations in a network that contains 100 nodes, we need about 20% of the network size (20 nodes, *i.e.,* one multicast source, 10 destinations and 9 forwarder nodes) to build single tree. On the other hand, 45 nodes (45% of the network size), *i.e.,* one multicast source, 30 destinations and 14 forwarder nodes, are needed if there are 30 destinations.

Fig. 12 and Fig. 13 show the simulation results of SPBMST and SSMT for the same parameters in table II. The same conclusions of Fig. 11 are also applicable for both figures (Fig. 12 and Fig. 13).

In figures 14, 15 and 16 we compare the USR and NoCR of the algorithms versus the network size. Fig. 14(a) shows that the three algorithms have the same USR. Fig. 14(b) shows that both algorithms SSPT and SPBMST have the same NoCR and they offer higher NoCR than SSMT algorithm.

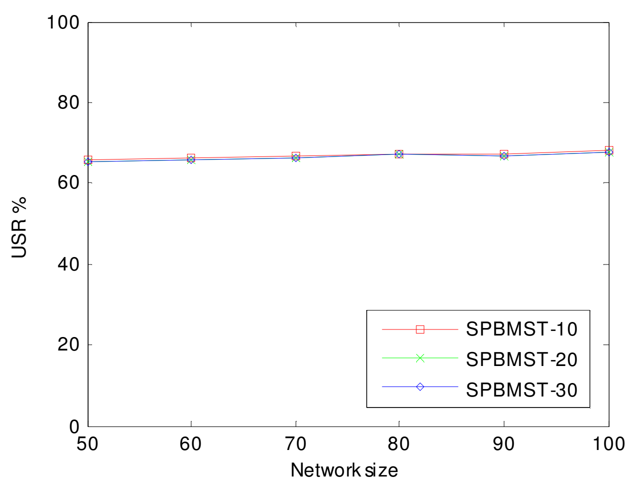


(a)

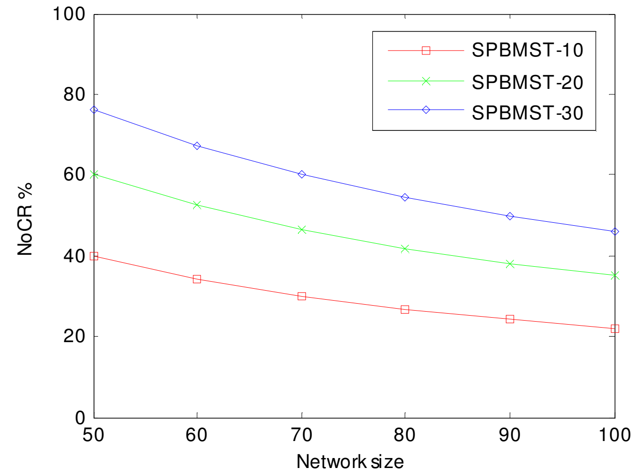


(b)

Fig. 11 User Satisfaction Ratio and node cost ratio for SSPT versus network size. Multicast group size is equal to 10, 20, and 30.

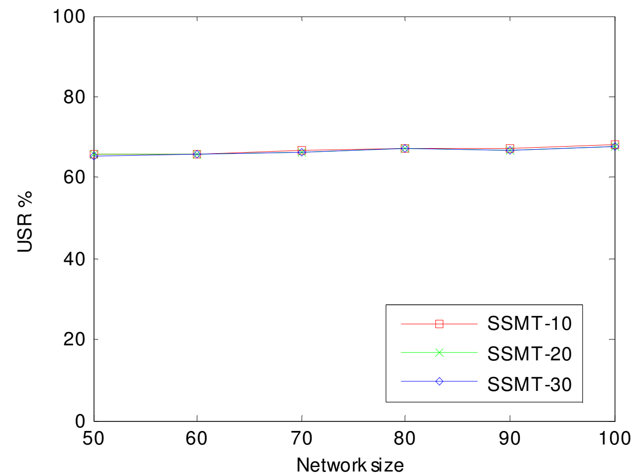


(a)

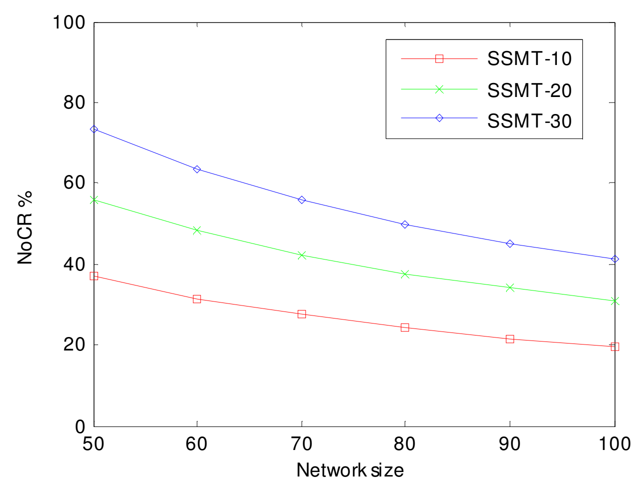


(b)

Fig. 12 User Satisfaction Ratio and node cost ratio for SPBMST versus network size. Multicast group size is equal to 10, 20, and 30.

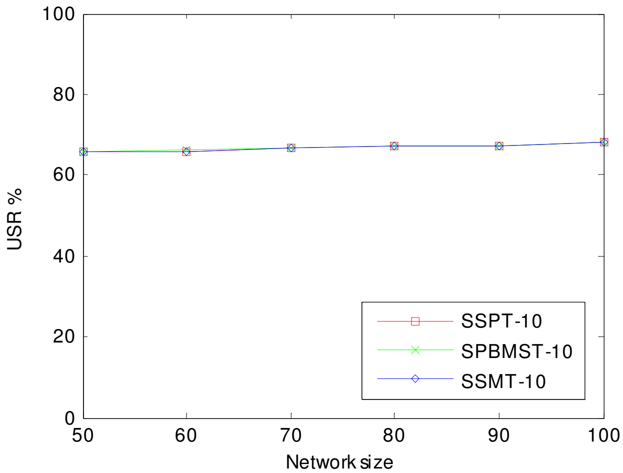


(a)

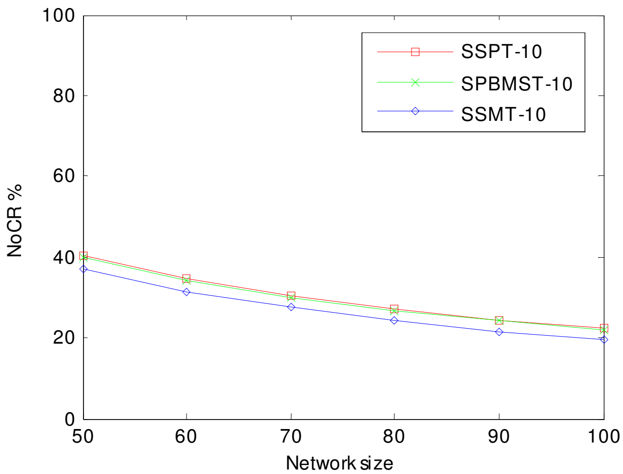


(b)

Fig. 13 User Satisfaction Ratio and node cost ratio for SSMT versus network size. Multicast group size is equal to 10, 20, and 30.

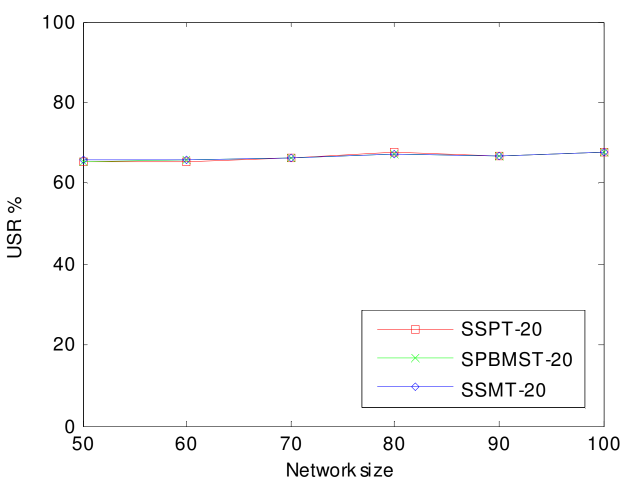


(a)

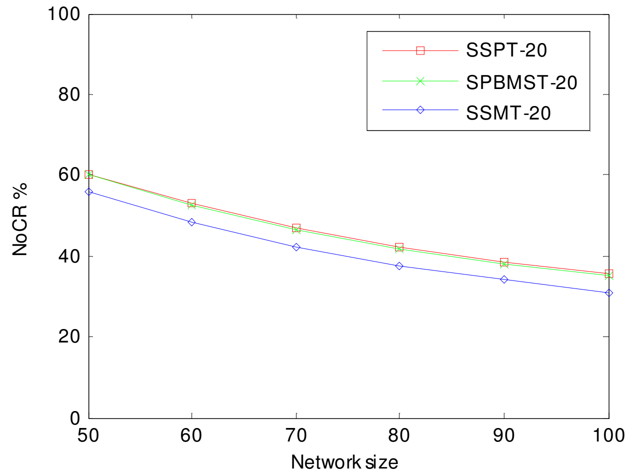


(b)

Fig. 14 User Satisfaction Ratio and node cost ratio for SSPT, SPBMST and SSMT versus network size. Multicast group size is fixed and equal to 10.

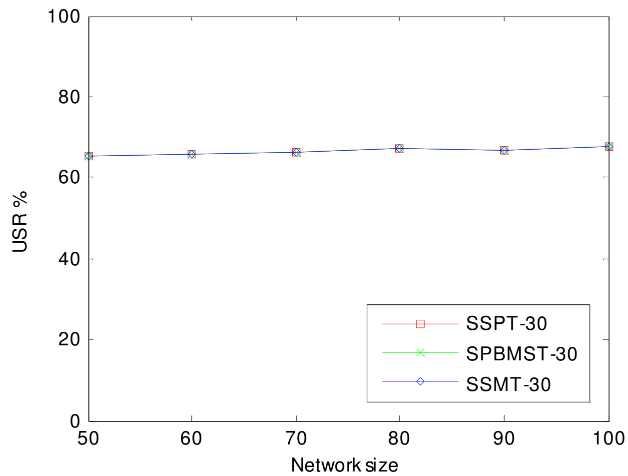


(a)

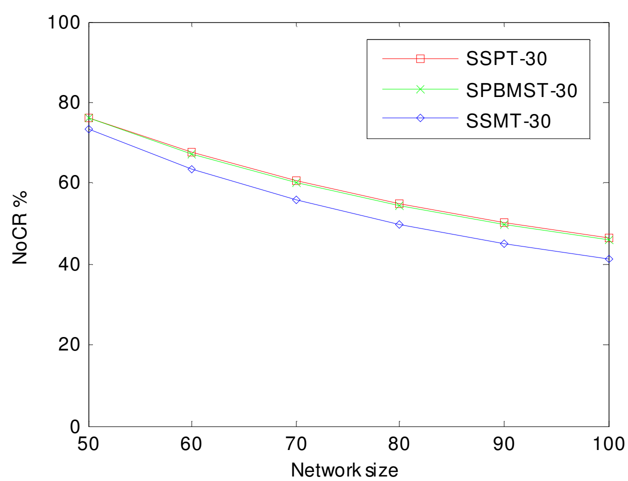


(b)

Fig. 15 User Satisfaction Ratio and node cost ratio for SSPT, SPBMST and SSMT versus network size. Multicast group size is fixed and equal to 20.



(a)

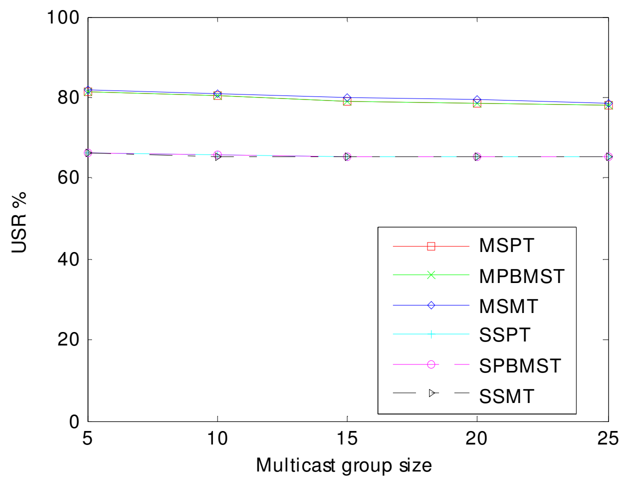


(b)

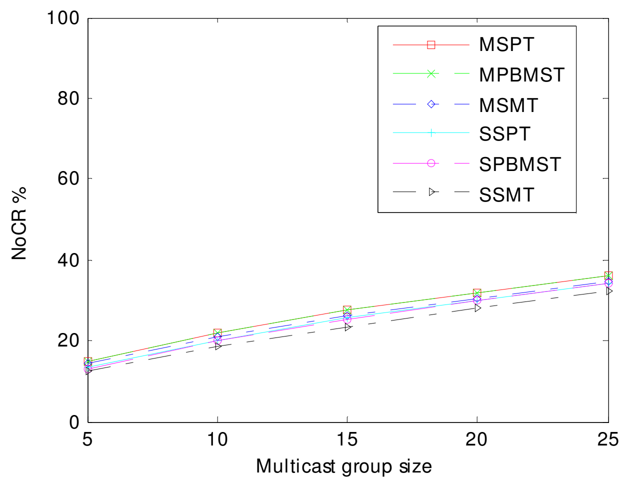
Fig. 16 User Satisfaction Ratio and node cost ratio for SSPT, SPBMST and SSMT versus network size. Multicast group size is fixed and equal to 30.

E. User Satisfaction Ratio and Node Cost Ratio versus Multiple Trees, Single Tree and Multicast Group Size

We compare in Fig. 17 the multiple trees algorithms and the single tree algorithms versus the multicast group size in terms of USR and NoCR, respectively. It is obvious that the multiple trees algorithms achieve higher USR than the single tree algorithms. On the other hand, single trees algorithms achieve lower cost (NoCR) than the multiple trees algorithms. The simulation parameters which are used in this experiment are the same parameters which are listed in table I.



(a)

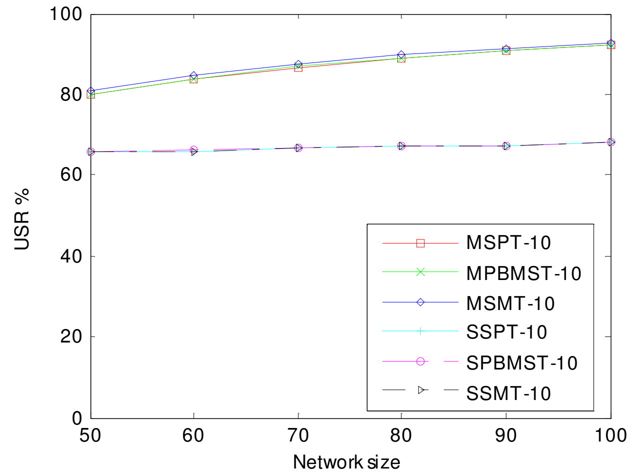


(b)

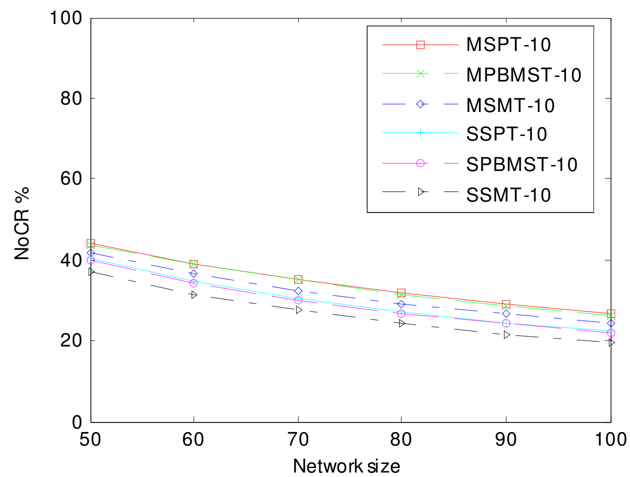
Fig. 17 User Satisfaction ratio and node cost ratio for multiple trees (MSPT, MPBMST and MSMT) and single trees (SSPT, SPBMST and SSMT) versus multicast group size. Network size is fixed and equal to 50.

F. User Satisfaction Ratio and Node Cost Ratio versus Multiple Trees, Single Tree and Network Size

In figures 18, 19 and 20 we compare the USR and NoCR of the multiple trees algorithms and single tree algorithms versus the network size. Fig. 18(a) shows that the multiple trees algorithms achieve higher USR than single tree algorithms. In addition, USR of multiple trees algorithms increases as the network size increases, on the other hand it (USR) stays unchanged for the single tree algorithms. Fig. 18(b) shows that the single tree algorithms achieve smaller NoCR than multiple trees algorithms.

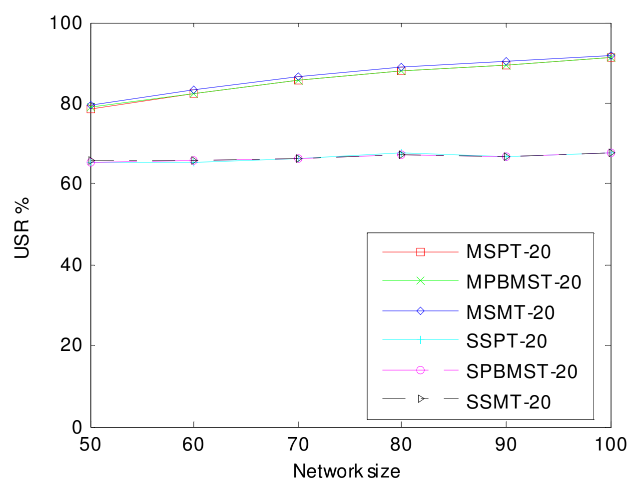


(a)

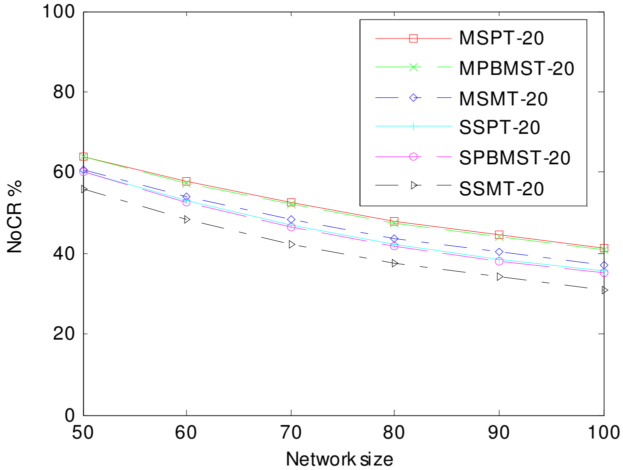


(b)

Fig. 18 User Satisfaction ratio and node cost ratio for multiple trees (MSPT, MPBMST and MSMT) and single trees (SSPT, SPBMST and SSMT) versus network size. Multicast group size is fixed and equal to 10.

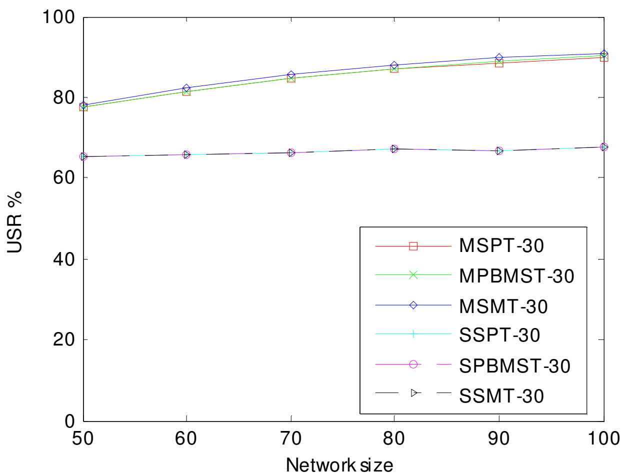


(a)

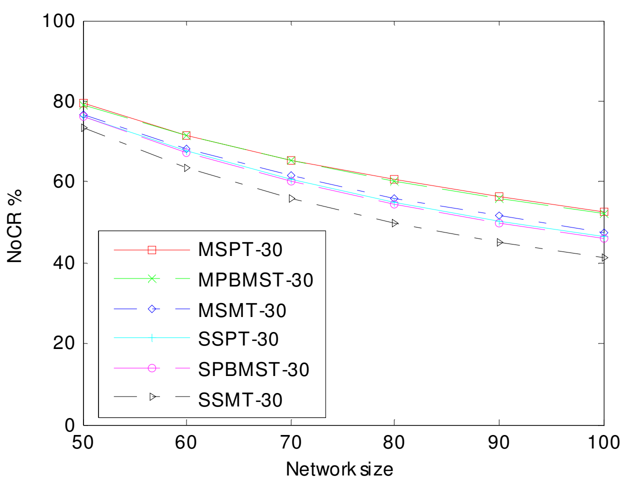


(b)

Fig. 19 User Satisfaction ratio and node cost ratio for multiple trees (MSPT, MPBMST and MSMT) and single trees (SSPT, SPBMST and SSMT) versus network size. Multicast group size is fixed and equal to 20.



(a)



(b)

Fig. 20 User Satisfaction ratio and node cost ratio for multiple trees (MSPT, MPBMST and MSMT) and single trees (SSPT, SPBMST and SSMT) versus network size. Multicast group size is fixed and equal to 30.

VI. Conclusion

We have presented the multilayered multicast routing in ad hoc wireless network. Three algorithms for constructing multiple trees to meet the requirements (number of video layers requested) of destination nodes were proposed and the complexity for them were analyzed. The three multilayered multicast algorithms are, MSPT, PBMST and MSMT.

Simulation results show that MSMT achieves better performance, in terms of USR and NoCR, than MSPT and MPBMST. In addition, simulation results show that the multiple trees algorithms achieve higher USR as compared with the single trees algorithms with some increase in cost (NoCR).

In this paper, we assumed that the bandwidth for each link between any two nodes have an available bandwidth that is sufficient to handle at least three video layers. If we want to consider the available link bandwidth, equation (1) can be modified as:

$$C(l) = \min\{C_n(u), C_n(v), BW(l)\} \qquad (8)$$

where $BW(l)$ is the available link bandwidth.

References

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

[2] M. V. G. Karlsson, "Packet video and its integration into the network architecture," *IEEE JSAC,* vol. 7, 1989.

[3] S. R. Comas D, Ortega A, "Rate-distortion optimization in a robust video transmission based on unbalanced multiple description coding," *In: Proc.workshop on multimedia signal processing, Cannes, France. IEEE Press,* pp. 581–586, 2001.

[4] A. Y. LeeY-C, "A collaborative multiple description transform coding and statistical error concealment method for error resilient video streaming over noisy channels," *In: Proc. international conference on acoustics, speech, and signal processing, Orlando, FL. IEEE Press,* vol. 2, pp. 2077–2080, 2002.

[5] J. H. Reibman A, Wang Y, Orchard M, Puri R, "Multiple description coding for video using motion compensated prediction," *In: Proc. international conference on image processing, Kobe, Japan. IEEE Press,* vol. 3, pp. 837–841, 1999.

[6] L. S. Wang Y, "Error resilient video coding using multiple description motion compensation," *In: Proc. workshop on multimedia signal processing, Cannes, France. IEEE Press,* pp. 441–447, 2001.

[7] V. K. Goyal, "Multiple Description Coding: Compression Meets the Network," *IEEE Signal Processing Magazine,* vol. 18, pp. 74-94, 2001.

[8] R. P. a. K. Ramchandran, "Multiple description source coding through forward error correction codes," *IEEE Proceedings Asilomar Conference on Signals, Systems, and Computers,* pp. 342-346, 1999.

[9] D. R. Agrawal, T.B. Murthy, C.S.R., " Robust Demand-Driven Video Multicast over Ad hoc Wireless Networks," *3rd International Conference on*

Broadband Communications, Networks and Systems, BROADNETS 2006. , pp. 1-10, 2006.

[10] 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.

[11] X. J. Huayi Wua, "QoS multicast routing by using multiple paths/trees in wireless ad hoc networks," *Ad Hoc Networks,* vol. 5, pp. 600-612, 2007.

[12] R. C. Prim, "Shortest connection networks and some generalisations," *Bell System Technical Journal,* vol. 36, pp. 1389–1401, 1957.

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

[14] a. C. S. R. M. B. S. M. Subir Kumar Das, "Weight Based Multicast Routing Protocol for Ad hoc Wireless Networks," *GLOBECOM,* vol. 1, pp. 117-121, 2002.