Crucial Video Scattering with QoS above Wireless Network by Selective Repeat Algorithm

E. Madhusudhana Reddy¹, M. Dinesh²

Professor Dept. of CSE¹, DRK College of Engineering & Technology, Hyderabad¹

Abstract—Now a day’s the broadcast casting of video place’s a major role in wireless network. If we want to broadcast a video, it takes a time to transmit from source to destination, due to large size of video. In wireless network the packet must be lightweight, due to that it takes a large amount of time to transmit large file. We propose a technique to easily transmit or broadcast the data/video efficiently with short duration. If a node wants to do video broadcast it can select the video and transmit in the transmission our (data) video is divided (splitting) small packets these packets are less weight, due to that packets are easily move in the wireless network with less time when compare to previous one. Now a day’s the broad casting of video place’s a major role in wireless network. If we want to broadcast a video, it takes a time to transmit from source to destination, due to large size of video. In wireless network the packet must be lightweight, due to that it takes a large amount of time to transmit large file. We propose a technique to easily transmit or broadcast the data/video efficiently with short duration. If a node wants to do video broadcast it can select the video and transmit in the transmission our (data) video is divided (splitting) small packets these packets are less weight, due to that packets are easily move in the wireless network with less time when compare to previous one. After reaching all packets to the destination we are going to merge the packets according to the sequence number which is already allocated while doing the splitting with help of Selective repeat algorithm we can achieve it. Furthermore, we use low complexity algorithm’s (approximation algorithm) for selecting the most energy efficient distribution for the entire set of directed acyclic graph (DAG). The flow selection and resource allocation process is adapted for each video frame. The flow selection and resource process is adapted in each video layer.

Index Terms—Wireless Network; MANET; Video Stream; Video Broadcast.

I. INTRODUCTION

A mobile ad hoc network (MANET) is a collection of wireless mobile nodes dynamically forming a temporary network without the use of any existing network infrastructure or centralized administration. Due to the inadequate broadcast range of wireless network interfaces, to communicate with nodes outside its transmission range, a node needs multiple hops to forward packets to the destination across the network. Since there is no stationary infrastructure such as base stations, each node operates not only as a host but also as a router. Hence, routing protocol for MANETs runs on every node and is affected by the resources at each mobile node. Considering typical characteristics of a MANET, such as a lack of infrastructure, dynamic topologies, constrained band-width, constrained energy and so on, a good routing protocol should minimize the limited resources and meanwhile maximize the network efficiency.

In recent years, a variety of routing protocols have been proposed for MANETs. Such protocols can be classified as proactive or reactive, depending on whether they keep routes continuously updated, or whether they react on demand. They can also be classified as unicast routing, broadcast routing and multicast routing, according to the type of applications. Unicast routing supports communications between one source and one destination. Dynamic Source Routing (DSR) [1], Ad Hoc On-demand Distance Vector (AODV) [2], Destination Sequenced Distance Vector (DSDV) [3], Optimized Link State Routing (OLSR) [4] protocol and so on are the typical unicast routing protocols proposed for MANETs. Multicasting is the transmission of data packets to more than one node sharing one multicasting address. The receivers form the multicast group. Actually, there could be more than one sender sending to a multicast group. Typical multicast protocols include On-Demand Multicast Routing Protocol (ODMRT) [5], Multicast Ad-Hoc on-Demand Distance Vector (MAODV) [6] and Ad hoc Multicast Routing (AM Route) [7] and so on. Broadcasting is a special case of multicasting, which supports sending messages to all nodes in the network.

II. MULTICAST/BROADCAST PROTOCOLS

The multicast/broadcast services are critical in applications characterize by the close cooperation of teams with requirements for audio and video conferencing and sharing of text and images. Moreover, most routing protocols in MANETs rely on the broadcast function to trade important routing packets between mobile nodes and need the multicast function to make more capable use of network bandwidth for some particular multimedia applications. Hence, broadcast and multicast are essential operations for mobile nodes to create a routing path in MANETs.

A. Multicast Protocols

Multicasting is the broadcast of data packets to more over one node sharing one multicasting address. It is proposed for group-oriented computing. Several multicast routing protocols have been anticipated for MANETs, which can be classified as unicast-based, tree-based, mesh-based, or hybrid (fusion) protocols, according to how allocation paths among group members are constructed.
B. Protocols Classification

- Unicast-based multicast protocols
  Tree-based multicast routing protocols can be supplementary divided into source-tree-based and shared-tree based schemes. In a source-tree-based multicast protocol, a multicast tree is conventional and maintained for each source node of a multicast group, and shared-tree-based multicast protocols use a single shared tree for all multicast source nodes. In the source-tree-based multicast protocol, every multicast packet is forwarded along the most accomplished path, i.e. the shortest path, from the source node to each multicast group member, but this method incurs a lot of manage overhead to keep many trees. For the shared-tree-based multicast protocol, it has minor control overhead since it retain only a single tree for a multicast group and thus it is more scalable. Adaptive Demand-driven Multicast Routing (ADMR) is source-tree-based and Multicast Ad Hoc On-Demand Distance Vector (MAODV) [6] is a shared-tree-based multicast protocol developed for MANETS.
- Tree-based multicast protocols
  Tree-based multicast routing protocols can be supplementary divided into source-tree-based and shared-tree based schemes. In a source-tree-based multicast protocol, a multicast tree is conventional and maintained for each source node of a multicast group, and shared-tree-based multicast protocols use a single shared tree for all multicast source nodes. In the source-tree-based multicast protocol, every multicast packet is forwarded along the most accomplished path, i.e. the shortest path, from the source node to each multicast group member, but this method incurs a lot of manage overhead to keep many trees. For the shared-tree-based multicast protocol, it has minor control overhead since it retain only a single tree for a multicast group and thus it is more scalable. Adaptive Demand-driven Multicast Routing (ADMR) is source-tree-based and Multicast Ad Hoc On-Demand Distance Vector (MAODV) [6] is a shared-tree-based multicast protocol developed for MANETS.

C. Simplified Multicast Forwarding (SMF)

In MANETs, unicast routing protocols can provide effective and efficient mechanisms to flood routing control messages in the wireless routing area. One such solution is the Simplified Multicast Forwarding (SMF) specification designed within the Internet Engineering Task Force (IETF) [8].

D. Broadcast Methods

Broadcasting is the process in which a source node sends a message to all other nodes in the network, and it is also a special case of multicasting. Since even unicast and multicast routing protocols often have a relay component, broadcasting is important in MANETs. For instance, protocols such as DSR [1], AODV [2], Zone Routing Protocol (ZRP) [5] and Location Aided Routing (LAR) [6] use broadcasting to establish routes. Broadcasting methods have been categorized into four families utilizing the IEEE 802.11 MAC specifications [7].

E. Classical Flooding

In Classical Flooding, a source node broadcasts a message to all its neighbors, and each of these neighbors will check if they have seen this message before. If yes, the message will be dropped; otherwise the message will be rebroadcast at once to all their neighbors. The process goes on until all nodes received the same message. Although this method is very reliable for MANETs with low density nodes and high mobility, it is very harmful and unproductive as it may cause severe network congestion and quickly exhaust the nodes energy. Typical Flooding is the simplest case of SMF multicast forwarding we will state later.

F. Probability-Based Methods

The probability-based method tries to solve the problem of the Classical Flooding method. Each node i∈N is given a predetermined probability pi for rebroadcasting. Thus, the network congestion and collisions can be minimized if some nodes do not rebroadcast. In this approach, there is a danger that some nodes will not receive the broadcast message. Probabilistic Scheme and Counter-Based Scheme are both probability-based methods which were proposed by [8].

- Probabilistic Scheme
  The Probabilistic Scheme is parallel to flooding, apart from that nodes only rebroadcast with a prearranged possibility. In transparent networks, multiple nodes share related transmission exposure. Thus, randomly having some nodes that will not rebroadcast can accumulate network resources. In sparse networks, there is not as much of shared coverage, thus, nodes cannot collect all the transmit packets with the Probabilistic system unless the possibility parameter is high. When the probability is 100%, this scheme is matching to Classical Flooding.
- Counter-Based Scheme
  In Counter-Based Scheme, when node tries to rebroadcast a packet, the packet may be barren (blocked) by the eventful medium, retreat procedure and other queued packets. During this stage, a node may collect the same packet from other nodes before the queued packet is send out. A counteract c is used to record the number of times the transmit packet is received. A counter threshold C is select. When c>C, stop the rebroadcasting, or else, the packet should be rebroadcasted.

G. Area-Based Methods

Area-based methods assume nodes have common broadcast distances. A node will rebroadcast only if the rebroadcast will reach a enough added coverage area. Distance-Based Scheme and the Location-Based Scheme planned also both area-based methods.

H. Distance-Based Scheme

In the Counter-Based Scheme, a counter is used to decide whether to retransmit or not. In the Distance-Based Scheme, the distance between nodes is used to make the decision.

I. Location-Based Scheme

In the Location-Based Scheme, each node needs to create its own location in order to approximate the additional coverage more accurately. If the additional coverage area to rebroadcast is less than a given threshold, the message is dropped, otherwise the message will be rebroadcast.

J. Neighbor Knowledge Methods

In a Neighbor Knowledge process, neighborhood information is maintained, and is used to decide whether to
retransmit frames that have timed out. It may be used as a stop
and-wait protocol for the delivery of ACK as used with go-back-n. The receiver accepts out-of-order frames and buffers them. The sender independently retransmits frames that have timed out. It may be used as a protocol for the delivery and acknowledgement of message units, or it may be used as a procedure for the delivery of subdivided message subunits. When used as the procedure for the release of messages, the sending process continues to send a quantity of frames specified by a window size even after a frame loss. Unlike Go-Back-N ARQ, the receiving process will continue accept and acknowledge frames sent after an preliminary error; this is the general case of the sliding window protocol with both broadcast and receive window sizes greater than 1.

The receiver method keeps track of the sequence number of the initial frame it has not received, and ends that quantity with every acknowledgement (ACK) it sends. If a frame from the sender does not arrive at the receiver, the sender continues to send ensuing packet until it has empty its window. The receiver continues to fill its receiving window with the consequent frames, replying each time with an ACK containing the sequence number of the earliest missing frame. Once the sender has sent all the frames in its window, it re-
sends the frame numeral given by the ACKs, and then continues where it left off.

The size of the sending and receiving windows must be identical, and half the maximum series number (assuming that sequence numbers are numbered from 0 to (n−1) to avoid miscommunication in all cases of packets being dropped. To identify this, consider the case when all ACKs are destroyed. If the receiving window is bigger than half the maximum series number, some, possibly even all, of the packages that are resent after timeouts are duplicates that are not predictable as such. The sender moves its window for every packet that is acknowledged.

III. PROPOSED SYSTEM

In this paper, we use routing information for the following reasons:

A. Shortest Path Routing

Kruskal’s is most efficient shortest path routing algorithm for transmission of multimedia file in wireless networks. This approach has better performance when edges are high in a network. And also It considers only the edges in (DAG) for transmission of file. The algorithm maintains a forest of trees an edge is accepted if it connects vertices of distinct Trees.

We need a data structure that maintains a partition, i.e., a group of disjoint sets with the following operations:
- find(u): return the set storing u
- union(u,v): replace the sets storing u and v with their union.

Pseudo Algorithm Kruskal (G):
Input: A weighted graph G.
Output: A minimum spanning tree T for G.

let P be a priority queue storing the edges of G and their weights
T ← ∅
While Q ≠ ∅ do
(u,v) ← Q.removeMinElement()
if P.find(u) ≠ P.find(u) then add edge (u,v) to T
P.union (u,v)
return T

Running time: O ((n+m) log n)

Analysis
Compare Prim and Kruskal
• Both have the same output • MST
• Kruskal’s starts with forest and merge into a tree
• Prim’s always stays as a tree
• If you don’t know all the weight on edges • use Prim’s algorithm
• If you only need incomplete solution on the graph • use Prim’s algorithm

B. Representation of a Partition
• Each set is stored in a sequence
• Each element has a reference back to the set operation find(u) takes O(1) time in operation union(u,v), we shift the elements of the smaller set to the sequence of the larger set and update their references.
• The time for operation union(u,v) is min(nu, nv), where nu and nv are the sizes of the sets storing u and v. Whenever elements are processed, it goes into a set of size at least double therefore each element is processed at most log n times.

Analysis
Compare Prim and Kruskal
• Both have the same output: MST
• Kruskal’s starts with forest and merge into a tree
• Prim’s always stays as a tree
• If you don’t know all the weight on edges: use Prim’s algorithm
• If you only need incomplete solution on the graph: use Prim’s algorithm.

Complexity
Kruskal: O (NlogN) Comparison sort for edges Prim: O (NlogN)
Search the least weight edge for every vertices.

Analysis of Kruskal’s Algorithm
Running Time = O(m log n) (m = edges, n = nodes)
Testing if an edge creates a cycle can be slow unless a complicated data structure called a “union-find” structure is used. It usually only has to check a small fraction of the edges, but in some cases (like if there was a vertex connected to the graph by only one edge and it was the longest edge) it would have to check all the edges. This algorithm works most excellent, of course, if the number of edges is kept to a minimum.

Analysis of Prim’s Algorithm
Running Time = O(m + n log n) (m = edges, n = nodes)
If a heap is not used, the run time will be O(n²) instead of O(m + n log n).

However, using a heap complicates the code since you’re complicating the data structure. A Fibonacci heap is the best type of heap to use, but again, it complicates the code. Unlike Kruskal’s, it does not need to see the whole graph at once. It can deal with one piece at a time. It also does not need to be troubled if adding an edge will create a cycle since this algorithm deals primarily with the nodes, and not the edges. For this algorithm, the number of nodes wants to be kept to a bare minimum in addition to the number of edges. For tiny graphs, the edges matter more, while for big graphs the number of nodes matters more.

Kruskal’s has improved running times if the number of edges is high, while Prim’s has a improved running time if both the number of edges and the number of nodes are low.

Go-Back-N: ARQ is a specific instance of the automatic repeat request (ARQ) procedure, in which the sending process continues to send a number of frames specified by a window size even without receiving an acknowledgement (ACK) packet from the receiver. It is a special case of the universal sliding window protocol with the transmit window size of N and receive window size of 1.

The receiver process keeps track of the sequence number of the next frame it expects to receive, and sends that number with each ACK it sends. The receiver will leave any frame that does not have the correct sequence number it expects – whether that frame is a “past” duplicate of a frame it has already ACK’ed or whether that frame is a “future” frame past the last packet it is waiting for. one time the sender has sent all of the frames in its window, it will identify that all of the frames since the first last frame are excellent, and will go back to sequence number of the last ACK it received from the receiver process and fill its window beginning with that frame and continue the process over again.

Go-Back-N: ARQ is a more well-organized use of a connection than Stop-and-wait ARQ, since not like waiting for an acknowledgement for every packet; the connection is still being utilized as packets be sent. In other words, during the

\[
N = \text{window size} \quad R_n = \text{request number} \quad S_n = \text{sequence number} \quad S_b = \text{sequence base} \\
S_m = \text{sequence max} \\
\text{Receiver: } R_n = 0 \\
\text{Do the following forever:} \\
\text{If the packet received} = R_n \& \& \text{the packet is error free} \\
\text{Accept the packet and send it to an upper layer} \\
R_n = R_n + 1 \\
\text{Send a Request for } R_n
\]
Else
   Refuse packet
Send a Request for Rn
   Sender: Sb = 0
   Sm = N - 1
Repeat the following steps forever:
1. If you receive a request number where Rn > Sb
   Sm = Sm + (Rn - Sb)
   Sb = Rn
2. If no packet is in transmission, Transmit a packet
   where Sb <= Sn <= Sm.
   Packets can transmit in order.

The goal of this project is to explore energy-efficient protocols in broadcasting scenarios and compare a suitable protocol with three other broadcast protocols in Wireless network. We adopted the multipoint relay selection strategy based on residual energy in the EOLSR protocol and use it in the broadcasting scenarios. This EMPR selection strategy takes into account the energy dissipated in transmission and reception up to 1-hop from the transmitter and was verified to prolong the network lifetime and increase the packet delivery rate when combined with the proposed unicast routing strategy. The select repeat sequence approach helps in faster distributing and merging the media file packets at both source and destination ends. The future work in the design of energy-efficient broadcast routing protocols in wireless should try to reduce the transmission redundancy and overall network overhead, and thus achieve the minimum energy consumption and the maximum network lifetime.

IV. CONCLUSION

The goal of this project is to explore energy-efficient protocols in broadcasting scenarios and compare a suitable protocol with three other broadcast protocols in Wireless network. We adopted the multipoint relay selection strategy based on residual energy in the EOLSR protocol and use it in the broadcasting scenarios. This EMPR selection strategy takes into account the energy dissipated in transmission and reception up to 1-hop from the transmitter and was verified to prolong the network lifetime and increase the packet delivery rate when combined with the proposed unicast routing strategy. The select repeat sequence approach helps in faster distributing and merging the media file packets at both source and destination ends. The future work in the design of energy-efficient broadcast routing protocols in wireless should try to reduce the transmission redundancy and overall network overhead, and thus achieve the minimum energy consumption and the maximum network lifetime.

REFERENCES