Seamless Video Streaming: A Light Weight Session Handoff Scheme For Dynamic Stream Merging Based Wireless Mesh Networks

G. Sai Thrilok[1]

[1]Computer Science & Engineering Department

[1]Saveetha School Of Engineering, Saveetha University

Abstract--- The flexibility of Wireless Mesh Network (WMNarchitecture has made it easier for the delivery of videostreams over such networks. However, delivering video streams over WMNs increases workload on the mesh nodes, increasesbandwidth usage due to dedicated streams and limits scalability. These challenges are overcome by using WMNs with Dynamic Stream Merging Capability (DSM) [1]. DSM is acommunication sharing technique that can merge redundant streams at upstream nodes before flowing to downstream nodesin the multicast streaming tree. In spite of these benefits,there is a growing need to use a distributed session handoff scheme to support seamless video streaming for fast moving clients, which not many research works are concentratingon. Thus in this paper, we propose a light weight, fast anddistributed scheme that uses a novel session rate prediction technique and a novel Dynamic Relationship Tree (DRT). The proposed session handoff scheme promotes workload sharingand supports mobile clients moving at speeds as high as 250 miles per hour. Based on extensive simulations performed in NS2 we found that the proposed scheme can provide robust and seamless inter and intra-domain session handoff producing less latency and overhead compared to existing schemes.

I. INTRODUCTION

Wireless Mesh Networks are dynamically self-organized and self-configured, with the nodes in the network automatically establishing an ad hoc network and maintaining the mesh connectivity [2]. WMNs have the ability to provide internet connectivity to users within a long range using just one gateway. The issues related to service throughput,

robustness of the network and efficient data-sharing during video streaming over WMNs are efficiently solved by a

distributed method called DSM [1].

Contributions of this paper: This paper aims at designing a completely distributed session handoff scheme which provides seamless video delivery for fast moving clients. The proposed scheme consists of two parts, namely, a novel Dynamic Relationship Tree (DRT) which helps the mobile client to handover from home node to foreign node (reassociating mesh node after handoff from home node) and a novel session rate prediction algorithm which accurately predicts the most probable session rate of the video after handoff. Although, we apply the proposed scheme on mesh nodes with DSM capability, the scheme is generic enough to use in any video streaming application over WMNs. The motivation of this paper: In general all handoff schemes are designed to ensure connectivity a specific layers. Most of the works done so far are on Layer 2 handoff. They focus on reducing handoff delays by using a backhaul channel [3], by choosing the optimum handoff trigger time [4] or by using clique based theoritcal approach

[5]. All these aforementioned schemes including Mobile IP [6] and a novel Xcasting cross layer scheme [7], will not

acheive seamless video streaming since their schemes are affecting only the lower layers and will not effectively help

in maintaining connectivity at higher layers. Even though SMESH [8] follows a distributed framework, it will not overcome the challenges dealt with in this paper because the architecture of SMESH is not meant for providing video continuity for fast moving clients. Though there are few works done in session handoff like [9] which depends on multi-path video encoding, [10] which depends on access point and [11] which depends on proxy servers, these solutions impose dependency, thus are not be suitable for distributed framework. To illustrate the need for a distributed session handoff, we present the sequence trace (Figure 1) of the video stream received by a client moving at 100 miles per hour. This simulation result shows the video continuity maintained by the proposed scheme and the sudden break in connectivity (red line) when the proposed scheme is not used. Thus, this result shows the importance of designing a specific handoff technique for maintaining continuity in video streams, which is not studied in any of the research work. This fact motivated us in proposing a novel distributed session handoff scheme. We review the DSM technique in Section 2. In Section 3 we present the proposed session handoff scheme in detail. In Section 4 we discuss about our simulation study, and the paper is concluded in Section 5.

II. DYNAMIC STREAM MERGING

Dynamic Stream Merging (DSM) was proposed in [1]. DSM is a communication sharing technique which performs three functions: stream merging, intelligent buffering and merge termination. When a mesh node has DSM capability.

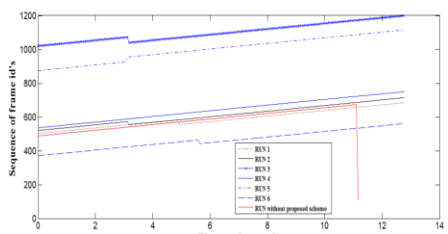

Fig. 2: Sample Cloud Architecture

- It will be able to construct a multicast streaming tree incrementally whenever there is a new request for a video stream.
- It will be able to detect redundant streams and merge using intelligent buffering, before sending them downstream.

- It will be able to avoid flow of dedicated streams.
- Thus DSM capability enhances the ability of a WMN to save more bandwidth and relax the stress at the gateway.

Consider a two-stream merge scenario to explain the working of DSM: A client is already watching a video (say *V*1) and has received the 100th packet (*P*100). During that instant a new client requests the same video *V*2. DSM-nodes will be able to detect that the videos are redundant and will create a new buffer. The buffer will store the packets

sent to the first client. Now, *V*1 and *V*2 becomes the merger and mergee streams respectively. The buffer starts to store

*V*1 from packet id 100 and when the new client receives the 100th packet of the stream *V*2, the node automatically sends a halt message to the gateway and requests to stop transmitting *V*2. The new client will play the video of the buffered stream *V*1 from *P*100. If the stream *V*1 stops abruptly, *V*2 is resumed using a resume message to gateway. In DSM every mesh node maintains a Session Table that contains a detailed entry about each stream the particular node is serving. For detailed explanations about DSM the reader is directed to these references [1] [12] .core competency.

III. Proposed Session Handoff Scheme

A. *Dynamic Relationship Tree (DRT)*

Recall that during video streaming, the streams travel hop by hop through mesh nodes to reach the client. This forms a multicast streaming tree. New mesh nodes join and leave the tree regularly depending on the client requests. In the proposed scheme a novel approach called Dynamic Relationship Tree (DRT) is introduced. This tree introduces a parentchild relationship between upstream and downstream nodes.

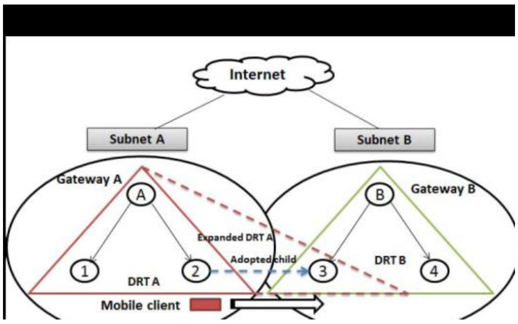

fig .3 :

Algorithm 1: DRT maintenance

1. if *send.RREQ and peertable.empty()* then
2. Parent = this→node;
3. if *recv.RREQ* then
4. Parent(this→node) = sender→RREQ ;
5. send.RREP(sender→RREQ) ;
6. if *recv.RREP* then
7. child(this→node) = sender→RREP ;
8. if *parent doesn't exist* then
9. push.peertable(child) ;
10. else if *parent exist* then
11. send.UpdateParent(child) ;
12. push.peertable(child) ;
13. if *recv.UpdateParent* then
14. push.peertable(child from UpdateParent) ;
15. return;

This process aims to promote workload sharing among mesh nodes and relax the stress on the gateway further. Maintanence of DRT is essential due to the dynamic nature of mesh nodes. The parents and their children are designated dynamically during the network discovery phase of a routing protocol. For instance, in AODV routing protocol [13], when there is a new request for data the source sends a Route

Request (RREQ) to find the destination, and the destination responds to the source using a Route Reply (RREP). Algorithm 2: The handoff scheme

1. //....Mobile client detects signal loss from home node..// send.HandoffRequest(client→home node) 2 if gateway.id(this.node) == gateway.id(this.DRT) then
2. //.......Intra-domain session handoff........//
3. if recv.HandoffRequest then
4. switch this.node←sender do
5. case home node←client OR parent←home node
6. if set of children ∈ peertable == foreign
7. Nodethenforward.HandoffRequest(this.node→foreign
8. node) ;
9. else
10. forward.HandoffRequest(this.node→parent(
11. home node)) ;
12. case foreign node←home node
13. FLAG = 0 ;
14. send.HandoffRequestACK(foreign
15. node→home node) ;
16. if recv.HandoffRequestACK then
17. switch this.node←sender do
18. case home node←foreign node
19. home node.calculate (FIDtran) ;
20. FLAG = 1 ;
21. FID = FIDtran ;
22. send.HandoffRequestACK(homenode→foreign node) ;
23. case foreign node←home node Premature buffering for client ;
24. FLAG = 2 ;
25. send.HandoffRequestACK(foreign node→home node) ;
26. //.....support client....//
27. return

such process, the sender of the RREQ (Route Request) becomes the parent and the sender of the RREP (Route Reply) for that request becomes its child in the DRT (shown in Algorithm 1 lines 1-6). In any distributed framework communication between mesh nodes are vital to update the state. For that purpose the proposed scheme uses a light weight “UpdateParent” packet. This packet is used for the downstream nodes (children) to notify the upstream nodes (parents) if there is a change in their state (lines 7-14).

This packet is sent to the parent if a new mesh node joins as its child in the DRT. The peer table mentioned in the algorithm will help the mesh nodes to know about its parent and children. Once the DRT is formed and regularly maintained the mesh nodes form a backbone to support distributed session handoff. The proposed session handoff scheme supports both inter-domain and intra-domain handoff. Inter-domain and intra-domain handoff occurs when the mobile client moves from one DRT to another or within the same DRT respectively. Intra-domain handoff:Conventionally, clients will be served by an arbitrary mesh node chosen by the routing protocol. However, in the proposed distributed framework every mobile client chooses the best mesh node with respect to signal strength. When a mobile client detects a better mesh node (foreign node) than the current one (home node) it sends a HandoffRequest packet to the home node requesting a handoff with the foreign node (refer to Algorithm 2 line

1). The client records its current video frame sequence in the request and sends it. When the home node receives the request, it will search for the foreign node in its peer table (indirectly checks if the foreign node is its child). If it finds it will forward the request to the foreign node. If it fails it will forward the request to its parent. This process continues until the foreign node receives the request (lines 2 to 11). The foreign node responds to the home node by sending a HandoffRequestACK (lines 12-14). Before sending a final response to the foreign node, the home node will calculate the most probable video frame sequence to deliver after handoff using the session rate prediction algorithm (refer to section 3.2). Finally the foreign node starts premature buffering from the calculated sequence and supports the client after association (lines 15-26). Inter-domain handoff: In inter-domain handoff the crucial issue to solve is the communication between two DRTs(refer to Figure 2). DRT merging is inculcated in the proposed scheme for solving the issue. Inter-domain handoff is triggered when the gateway id of the requested foreign node is not same as the home node (unlike intra-domain handoff, refer to line 2 of Algorithm 2). The inter domain handoff works similar to intra domain handoff, but after receiving the request the home node broadcasts it instead of forwarding. To avoid unwanted broadcasts,the parent of the home node after receiving the broadcast will wait for an update about foreign node from its child for a specific time. The parent re-broadcasts only when its wait timer expires. Now, the foreign node in another DRT will be able to detect the request. The foreign node responds in the same way as in intra-domain handoff. However, in this case the home node will be an adopted parent of the foreign node. This update in the relationship will merge the DRTs thereby ensuring seamless session handoff. Rest of the handoff process are similar to intra-domain handoff. *B. Session Rate Prediction algorithm* Recall that the home node calculates the most probable sequence of video streams to deliver for the mobile client after handoff. This section formulates the algorithm for calculating the session rate. Consider *FIDrate* as the rate at which the client receives the video streams, *RTTopt* is the optimum Round Trip Time value chosen by the algorithm (depends on parameter μ) and let *FIDtran* is the sequence number of the video stream segment from which the foreign node has to deliver the mobile client to ensure seamless

video streaming.

FIDrate = (Timenow − Timereq)

(Newfid − Reqfid)

RTTopt = μ × RTTcalc

FIDtran = (RTTopt)

(FIDrate) + Newfid

where *RTTcalc* is the sum of all the RTTs calculated from each path along the route of request from the home node to the foreign node, *Timenow* is the current time, *Timereq* is the time when client initiates handoff, *Newfid* and *Reqfid* is the current video sequence number at *Timenow* and *Timereq* respectively. It can be seen from the above equations that the *FIDtran* is dependant on *RTTopt*. Choosing the right *RTTopt* value depends on the parameter μ. The prediction technique considers $\mu \approx Ts/RTTcalc$, where *Ts* is the total time, the home node supports the client until re association. The time *Ts* inturn depends on transmission range and speed of the client. For instance, the total time supported by a mesh node is going to increase if the client moves slowly or when the mesh node could support the client longer (happens when transmission range is large). Thus, this customizable μ value changes with tranmission range of the node, number of nodes in the network and the speed of the client. We recommend low μ value if the network framework consists of few mesh nodes. Several simulation tests were donen to find the most optimum μ value (refer Figure 3) and was found to be $\mu = 2$ for a WMN with forty mesh nodes with transmission ranges of 50 meters. The prediction level is defined as the difference between the predicted sequence to transmit and the actual sequence that should have been transmitted to the client. This result also proves that the proposed scheme results in low prediction levels producing optimal session handoff without any jitter or delay. According to [14], the time at which a client reassociates with the foreign node and the time at which the client leaves the home node is completely stochastic and depends on the speed of the client. This randomness affects the session rate prediction. Thus we recommend to choose

the optimum value for μ to allow tolerance. IV. SIMULATION STUDY NS-2.34 is used for extensive simulations. We chose toapply the proposed session handoff technique over mesh nodes with DSM capability, as DSM-nodes can form a multicast streaming tree and merge streams to improve the network performance. The simulation results show that the proposed session handoff scheme is flexible, simple, and distributed with less overhead and latency. Figure 4 shows the topology used for the simulation results. There are nodes at every intersection point in the 1000 × 500 sqm. grid, but we wanted to focus only on the multicast streaming trees. DRT A denotes the multicast streaming tree rooted

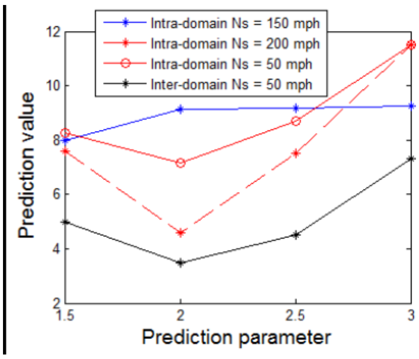

Fig. 3: Sensitivity testing at different speeds.

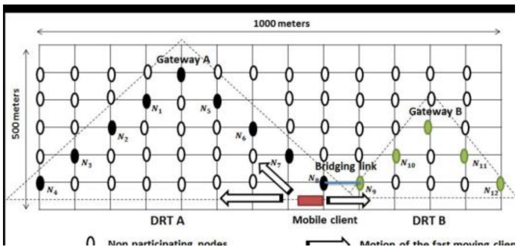

Fig. 4:Topology for simulation.

at gateway A and DRT B denotes the multicast streaming tree rooted at gateway B. The same topology is used to test both intra and inter-domain handoff. A modified Evalvid [15] framework is used for obtaining better trace files to detect the discontinuity in the video stream while the client is moving. The simulation had 40 mesh nodes with 50 meters transmission range following Two Ray Ground propogation model. Figure 5 shows the stability of session prediction algorithm for μ = 2 at different transmission ranges. It can also be seen that the proposed technique is robust enough to withstand very high speeds. We believe that the variation in prediction level is because of the topology and randomness in the movement of the client. The accuracy of RTT

deviates with the number of hops. In [16] the authors have recommended a technique to increase the accuracy of RTT measurement. However, on an average the prediction level is kept as low as six, which is better than in [17]. Note that, in [17] the authors use slow moving clients and a centralized approach. We watched the video that was received by the client using different video players, and found that it can sustain redundant video frames up to 30 and can sustain a maximum loss of six. Figure 6 shows the probability of the proposed method to produce flawless session handoff (when prediction value is 0). This probability signifies the accuracy of the proposed scheme. This result shows that the proposed scheme has more than 20% probability of producing seamless session.

