Corrective Actions at the Application Level for Streaming Video in WiFi Ad Hoc Networks

E. M. Macías, A. Suárez, and J. Martín
Grupo de Arquitectura y Concurrencia, Departamento de Ingeniería Telemática
Universidad de Las Palmas de G. C.
Campus Universitario de Tafira, 35017 Las Palmas de G. C. (Spain)

Abstract—**Efficient video streaming in a mobile ad hoc network (MANET) is a challenging problem due to the dynamic nature of the network that leads to high bit error rates, unpredictable delay, jitter, throughput and packet delivery ratios and frequent short, intermittent and long-term link failures. Despite the MANET research community's efforts, there are still open problems. For example, protocols and mechanisms that hide these issues to the video streaming applications users are still in early stages. However, these applications must tolerate transparently the dynamic behavior of the network and be able to progress in presence of disconnections. In practice, this is the exception rather the rule. In this paper, we present a multilayer cooperative solution to detect disconnections and reconnections between a video streaming server and a client and we propose corrective actions at the application level. With our transparent approach to the user, the video streaming sessions can tolerate frequent long and short disconnections and use more efficiently the shared wireless bandwidth.**

I. Introduction

Achieving multimedia communications over MANET pose many challenges [1] and profitable business [2]. Video streaming is a very useful technique for devices with low storage capacity such as mobile phones and *Personal Digital Assistants* (*PDA*) that use cellular [3], WiFi [4] or WiMax [5] wireless communication technologies. Among others, the following situations degrade the video streaming performance on MANET: i) interruption in packet delivery when a link breaks (e.g. sender, receiver or intermediate node goes out of coverage or their batteries go down), ii) efficient alternative path discovering without degrading jitter [6], iii) high error rates due to the multipath fading, iv) limited bandwidth combined with variable network latency [7]. The efficient election of the transport or application level protocol for video streaming and the efficient control of intermittent wireless channel disruption are also very important issues.

Usually *User Datagram Protocol* (*UDP*) is used to transmit live streaming video [8]. Over UDP, the couple *Real-time Transport Protocol* (*RTP*) and *Real-time Transport Control Protocol* (*RTCP*) is used for real time streaming video [9]. The server continuously sends frames and the client usually does not pause the streaming. A path break implies the server continues with the frame transmission but the client will not receive any frame (data is lost and the bandwidth and battery are not used efficiently).

The persistent version of *HiperText Transfer Protocol* (*HTTP*) can support streaming for *Video on Demand* (*VoD*) so a client sends a request and gets a response, and then sends additional requests and gets additional responses without *Transmission Control Protocol* (*TCP*) connection release. *Real Time Streaming Protocol* (*RTSP*) can use any of the above protocols for transmitting video data and TCP client commands to control the user session on the server.

Although TCP reliability mechanism will retransmit the missing data, the TCP socket will become invalid to the server or the client if an abort occurs and with high probability the user session will end abruptly. Aborts primarily occur when data goes unacknowledged for a period of time that exceeds the limits on retransmission defined by TCP. Other causes for an abort include a request by the application, too many unacknowledged TCP keepalive probes, receipt of a TCP reset packet and some types of network failures reported by the IP layer. We do not consider the improved versions of TCP explained in [10] because of they require: modifications to existing TCP (e.g *TCP-F* and *split-TCP*), more bandwidth and power consumption during a path failure (*TCP-ELFN*), dependency on a particular routing protocol to improve its performance (*TCP-Bus*), addition of layers to the TCP/IP protocol stack (*ATCP*). We do not also consider the protocols overviewed in [11] due to their performance is not well enough [12].

Cross-layer techniques have been applied to solve the above challenges, for example in [13] it is proposed the adaptation of the retry limit parameter at the 802.11 *Medium Access Control* (*MAC*) level to avoid triggering the TCP congestion control mechanism during short-term link failures (it is not appropriated for long-term disruptions). We consider a multilayer cooperative solution: a particular efficient network routing algorithm, any of the above transport protocols, a mechanism to support short and long-term disruptions for TCP based connections, and an application level software that implements the corrective actions when appropriate to robustly tolerate short and long-term disruptions. In order to consider any kind of video streaming client and server we implement a client agent and a proxy server. In this way we

Research partially supported by the Spanish CICYT (MEC) and European Regional Development Fund (FEDER) under Contract TSI2005-07764-C02-01 and The Canaries Regional Education, Cultural and Sports Ministry and FEDER under Contract PI042004/164.

T. Sobh (ed.), Innovations and Advanced Techniques in Computer and Information Sciences and Engineering, 525–530.

tested the good performance of our multi-protocol and multi-client and server solution for *WiFi* [14] ad hoc networks.

The rest of the paper is organized as follows: section 2 is devoted to discuss the related work. Section 3 reviews the software architecture. Section 4 presents the corrective actions. In section 5 we describe some experimental results. Finally, concluding remarks are summarized in section 6.

II. Related Work

The distributed and self-organizing nature of a MANET stems from having a routing protocol installed in each wireless node. The major routing protocols for MANET are classified into *on-demand* or *reactive* and *proactive* routing algorithms. The former initiate route discovery only after a path breaks incurring a high cost to establish a new route whereas the latter initiate route discovery early and before the path breaks at the cost of higher routing load.

Proactive protocols show more benefits to send video over ad hoc wireless networks than reactive protocols [15]. Due to the reactive behavior, the delay, jitter, throughput and packet delivery ratio for the communication flow may vary a lot in quantity. We use a proactive protocol named *Optimized Link State Routing Protocol* (*OLSR*) [16] that consists of: i) a neighbor sensing mechanism that detects changes in its neighborhood injecting and receiving HELLO messages periodically, ii) an efficient flooding of control traffic, i.e. OLSR packets injected into the network for the quick reconfiguration of path breaks. All nodes receive the messages and there are not duplicated messages thanks to the use of multipoint relays. This is an important property that favors its use in a wireless network which is by nature prone to mobility of nodes and collisions due to the hidden terminal problem, iii) diffusion of topological information necessary to obtain optimal routes in terms of the number of hops. This information is valid for a period of time so expired information is removed. All the traffic in OLSR is UDP and it is transmitted by broadcast or multicast on port 698.

Ref. [17] proposes a hybrid mechanism that consists of an early warning to a reactive protocol in order to initiate route discovery only when a path is likely to break. With this approach, the authors try to reduce the time to detect the disconnection and find a new path, and also reduce the routing load. The signal strength is used as the preemptive trigger. However, as the authors recognize, this physical parameter is not optimal because the value reported differs among 802.11 cards vendors [18]. Therefore, the signal strength values read from a 802.11 card should not be assumed to be particularly accurate [19].

Ref. [20] presents an architecture for detecting and diagnosing faults in IEEE 802.11 infrastructure wireless networks. One of its contributions is enabling bootstrapping and fault diagnosis of disconnected clients to report information to network administrators and support personnel. This work does not provide any support for disconnected clients during on-going video streaming sessions. On the contrary, we provide corrective actions at the application level as well as detection of disconnected clients.

References [15], [17] and [20] are concerned about providing a route between the client and the server in terms of the quick reconfiguration of paths but they are not concerned in providing solutions in the scope of video streaming sessions when a path can not be established.

Multimedia data replication at several servers in a MANET is proposed in [21]. The client establishes a connection to the nearest server once a data block has been received or it is renewed to the same server if it is still the nearest one. To our knowledge, this approach will not perform well with standard streaming protocols because a new connection for VoD implies starting the streaming from its beginning.

The factors causing the low communication quality of current WiFi ad hoc networks and its derived implications for application development is the main concern of [18]. For example, the authors state that applications must tolerate frequent disconnections and the programmers must define when a link is considered to fail but they do not provide a practical solution. We have programmed a proactive mechanism that detects when a link between the client and the server is not available (there is not an alternative route according to the routing protocol) and we do the corrective actions at the application level described in section 4.

Ref. [22] presents a proactive adaptation to the UDP-based streaming video sent by a fixed server to a mobile client connected to one 802.11b infrastructure wireless network. The adaptation consists of increasing the buffer size on the client to store more frames just before entering the low quality area (termed *trouble spot*) in the hope that the mobile client will exit the trouble spot before the buffer runs out. In this paper we consider not only UDP-based streaming video but also TCP-based. On the other hand, we consider both the client and the server mobiles. In this scenario, path breaks take place frequently and quickly so the proactive adaptation proposed in [22] could not be viable.

III. the Software Architecture

Fig.1.a. shows our target MANET consisting of any number of hops. The server node (S) communicates with the client node (C) via zero or more intermediate nodes (I).

The shaded parts in Fig. 1.b to d are the new software elements we introduce (in the protocol stack) to avoid modifying the client application, the server application and the streaming protocols:

- *olsrd* (OLSR daemon) [23] is an implementation of OLSR. OLSR routes efficiently the packets into the network according to the number of hops between the sender and the client. Due to the time-varying characteristics of the wireless links, olsrd can be configured to calculate the optimal routes defined as the number of attempts by a node on average to