

Rate adaptation in 3GPP video streaming using track switching over a multihop WLAN

Eliseo Catalán

Jordi Casademont

Xavier Sánchez-Loro

Josep Lluís Ferrer

Technical University of Catalonia

{eliseo.catalan, jordi.casademont, xsanchez, jlferrer}@entel.upc.edu

Abstract

In nowadays communication technologies, one of the main challenges to overcome is how to translate theoretical algorithms, designed protocols, existing hardware products from different manufacturers and the already knowhow into innovative applications that can produce a benefit for the user and for the commercializing firm. Following this topic, the present paper introduces an application that combines different emerging technologies: wireless mesh networks, multihop routing and adaptation of streaming delivery with the objective to create a tool for effective marketing campaigns.

1. Introduction

At present time, many people carry in their pockets sophisticated communication tools that are able to do much more than their usual role: talk by phone. Many of our mobile phones, apart from their cellular 3G interface, are equipped with other network technologies as IEEE 802.11 and Bluetooth. The objective of the application presented in this paper is to make use of these facilities implementing a coordinated network of local information points that would give spaces such as airports, tourist areas, shopping malls, open environments, and so on extra information capabilities due to the higher available bandwidth of these technologies. Once the user enters under the coverage area of the network it will be able to access personalized and adapted video content. For instance, if the situation is a shopping mall, users get multimedia content of the different shops on the mall.

Another aspect to consider is the maximization of user's quality of experience (QoE) making use of an efficient distribution system, able to adapt multimedia data to the capabilities of user's devices and network conditions. In video transmission, QoE can be basically mapped to the quality and smoothness of the

video play. Adapting the transmitted content to network circumstances leads to a better experience for the user, especially when working over WLAN.

WLAN are especially difficult to plan because available bandwidth varies a lot depending on the number of users or propagation distances. They also provide no guarantee in terms of QoS (Quality of Service), i.e. bandwidth, delay or jitter. This variability leads to a low QoE when trying to distribute fixed bitrate streaming video over multicellular WLAN with mobile users. Nevertheless, nowadays mobile devices are able to provide information related to buffer usage or existing delay using latest 3GPP Releases [1] and loses using RTCP messages. The server can use this feedback to adapt the content transmitted to the client.

Latest Darwin Streaming Server release provides 3GPP-Rel6 compatibility and performs some rate adaptation by removing specific frames. This adaptation solves buffering problems in client side and helps to reduce network traffic and packet losses, but does not avoid QoE problems, as the video is not fluently played due to frame drops.

As a single 3GP file supports the inclusion of multiple tracks containing the same video but with different bitrates, we have developed a track swapping system, which allows the server to switch between tracks in a total transparent way for the client. The goal is to adapt active connection's bitrate according to network conditions to benefit the whole network. One of the main challenges is to make this track-switching system compatible with as many commercial clients as possible using an existing open source streaming server. It is also necessary to develop a module that provides information about user's device capabilities.

This document is structured in five major parts; firstly it introduces system's architecture, secondly the involved technology, thirdly it describes the implementation, then practical results and finally conclusions.

2. System architecture

The developed application is based in a competitive platform (Wilico by Futurlink [2]) that provides the minimal hardware to achieve the expected requirements. This platform is an off-the-shelf Linux computer with IEEE 802.11 and Bluetooth interfaces designed to work as a high capabilities Access Point and distribute multimedia content. In the present application, Wilico nodes are programmed with software that enables them to set up a multihop wireless network able to relay data between them using the AODV protocol, they can also manage client mobility redirecting the data to the node where the client is associated. There are two different networks: the access network between user's devices and Wilico nodes, and the backbone network to connect relay nodes. Both networks use 802.11 interfaces with different channels (Figure 1).

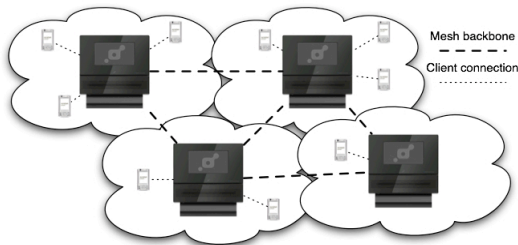


Figure 1: Multihop network.

Another point is the fact that when a user enters for the first time into the system, there are some tools to discover many of the capabilities and characteristics of his device [3]. The detector module is set in one of the Wilico nodes and using the multihop network interrogates new clients. It obtains the maximum possible information about their hardware and software characteristics as well as to completely characterize the used browser in order to create a consistent, valid and parsable profile. Detector module executes different detection mechanisms depending on the user's device. The module makes all possible tests on the device and then it matches the results.

The final required module is the streaming server. It can be installed in any of the Wilico nodes and stream video content to all clients using the backbone network but, in order to increase system performance, we have installed one streaming server in each Wilico node that only serves its associated clients, so the practical experiments will be performed in a one-hop scenario as it is topology independent. The streaming server also uses the information provided by the detector module to adapt the contents to the different devices.

3. Involved Technology

3.1. PSS: Packet-Switched Stream Service

Proposed by 3GPP, PSS specification scope is to provide a framework for streaming applications over 3G data networks. Different PSS versions have been released so far, the Release-6 is especially relevant because it includes tools to perform rate adaptation in mobile transmissions. Although this framework was intended to work in cellular telephony scenarios, the protocols and solutions implemented include any kind of network such as WLAN.

3.2. 3GP Multimedia Container

3GPP also proposed a multimedia container format to provide a standard container for different available mobile devices. 3GP video files are very similar to MPEG-4 part 14. Multiple video coding schemes can be found inside this container, typically MPEG-4 part 2, H.263 and H.264 for video and AMR NB/WB, AAC for audio. Those codifications were chosen considering the typical narrow bandwidth that is available in packet-switched networks over cellular networks.

3.3. ISO Boxes

3GP files are based in ISO Boxes, like MP4. These boxes contain not only audio and video tracks, but also additional control information. They are ordered hierarchically and divided in two major groups as shown in Figure 2.

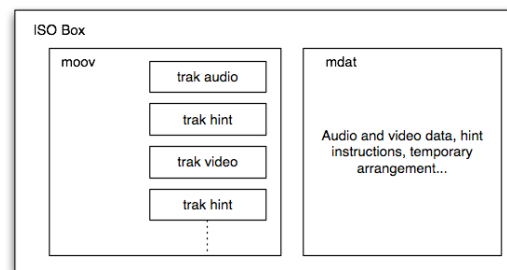


Figure 2: Example of ISO Boxes.

The first group, called *moov*, is related to the movie and stores as many *trak* boxes as tracks available in the file, these tracks have many fields used to define its characteristics and properties. Hint tracks are a special kind of *trak* that contains useful information about a video or audio track; the streaming servers use this information to properly package data to the clients. In hinted media there are as many hint tracks as media tracks. In addition of this *moov* group, there is the *mdat* group that contains media information.

Since 3GP 6th Release, one of the fields included in *traks* is the alternate group, that indicates which tracks could be chosen instead of the others.

3.4. RTP/RTCP

RTP (Real-Time Transmission Protocol) is used to transmit media with real-time requirements providing functionalities as timestamps or payload identifiers. Transmitted flows sources are identified using a field called SSRC (Synchronization source); there is also an identifier called CSRC (Contributing source) used to identify contributor sources in scenarios where an intermediate device as a mixer is defined. Each transmitted track has a RTP flow so, typically, in a two-tracked file, two RTP sessions will be started.

The most relevant fields of RTP packets are: Payload Type (PT), that will generally be in 96-127 range reserved for dynamic coding; Sequence Number (SQ), incremented sequentially in each session, usually starts with a random value; Timestamp, that indicates the sampling time of the first octet in the data, it keeps flow synchronization and it's useful to evaluate jitter.

Information provided by RTP packets is interpreted by the client and server who generate Receiver Report (RR) and Sender Report (SR) messages respectively; these messages are part of the RTCP (Real Time Control Protocol). RTCP messages include percentage of losses, cumulative losses, latest received sequence number, jitter, last sender/receiver report and delay since last report. In these reports additional fields can be added as, for instance, application specific messages known as APP. The PT field marked with the value 204 identifies these APP messages and then the NAME field identifies the kind of specific data carried.

The APP reports used in 3GPP are known as NADU, which stands for Next Application Data Units and are identified when this name is set to PSS0. Much of the information included in these packets is referent to ADU, which vary depending on the coding of the carried payload.

NADU packets include different fields as SSRC of the reported source; NSN and NUN indicating the RTP sequence number and the number of the next ADU to decode respectively. Free buffer size indicates the available data room in 64 bytes blocks, so a limited size can be announced. Finally the DELAY is useful to calculate the time between the stored data and its playing time.

RTP is transmitted over UDP, as this transport protocol has not delivery guarantee neither network congestion mechanisms, all server sessions are allowed to go into the network with equal probability. The

result is a high packet loss when aggregate throughput goes over network capacity [5].

3.5. RTSP: Real Time Streaming Protocol

RTSP is a fundamental part of the protocol stack in PSS transmission as it is used to establish and control sessions, providing a request/reply system similar to HTTP. RTSP uses a well-known TCP port (554) in order to start message exchange; these messages are typically the following:

- OPTIONS is sent by the client in order to obtain all server supported RTSP commands.
- DESCRIBE is sent by the client in order to obtain a description of a video file.
- PLAY and PAUSE are used to start, seek and temporary stop the data flow.
- TEARDOWN is used to finish the session.
- SETUP is used to ask the server for a track;

When a track is setup, clients expect the server to send information with some session parameters: time, sequence number, SSRC and timestamp fields. Some clients have shown more tolerance to malformed packets such as wrong sequence numbers, but in all cases respecting the PT field is mandatory.

3.6. SDP: Session Description Protocol

SDP is the protocol used to describe the file characteristics such as contained tracks, coding, bitrate, length, alternate groups, 3GP adaptation support, ... The client can interpret this information in order to choose its more suitable tracks. In practical situations most terminals ignore this information choosing the first pair of tracks or them all.

3.7. DSS: Darwin Streaming Server

The presented solution is developed over DSS 6.0.3. This server was chosen because is open source, provides a good developing framework and tools, and is 3GPP Rel6 compatible. DSS 6.0.3 introduces a rate-adaptation mechanism using RR and NADU packets based on dropping video frames.

4. Implementation

This point presents the main modifications carried out in the DSS in order to work with most commercial devices and to improve the adaptation algorithm.

4.1. 3GP file generation

The first step of the implementation is the generation of a proper 3GP/MP4 file, with multiple tracks with the

same content but coded at different bitrate. Having the same video coded at different bitrates enables the client to continue playing after track-switching. In order to work with DSS, this file must be hinted in advance.

4.2. Describe

Typical client-server dialog (Figure 3) starts when clients ask for a resource, typically a 3GP/MP4 file.

By default, the original DSS sends information of all tracks inside the SDP Session Description message. Nevertheless, many times this is not the best option because clients choose the first track by default, independently of system capacity, device capabilities or track bitrate. Other clients (VLC or QuickTime to the date releases) ask for all available tracks, downloading simultaneously all of them, as they are not 3GP release 6 compatible.

Our system improves this situation making use of some known information: main capacities of user devices, acquired during the association phase, and network load. The sever has been provided with an algorithm that using this known information selects the most suitable tracks of a requested file for each specific situation. After that, it sends a SDP Session Description message with only the optimum tracks. This solution forces the client to select the tracks that the server algorithm considers most favourable.

4.3. Setup: Adding alternative tracks

When the client receives the SDP Session Description message, it sends one Setup request for each track (usually one for video and one for audio). Then, the server establishes one client session that contains not only requested tracks but also alternative tracks containing the same information but coded at different bitrate. Alternative tracks are those included in the 3GP file that have not been announced in the SDP Session Description message. Alternative tracks may be used for dynamic adaptation if network conditions change during the transmission. All alternative tracks share the parameter values of the original track as SSRC, port, first RTP sequence number, ... These alternative tracks remain inactive so the server does not send packets belonging to them. The response message to the client only includes information related to the active tracks as it won't accept any other information.

Tracks (active and alternative) belonging to a client session are ordered inside the server according to their bitrate. This management is done in basis of some new attributes added to each track.

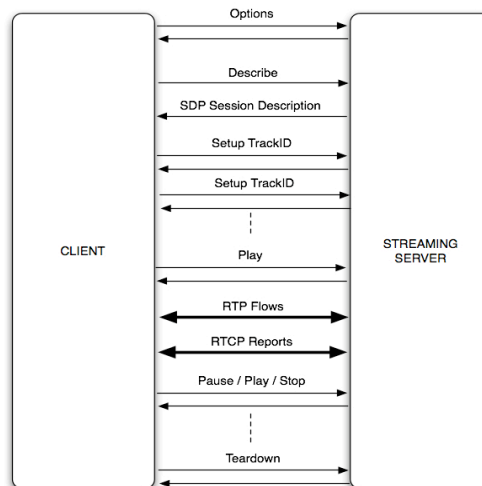


Figure 3: Client-Server Dialog.

4.4. DSS existing algorithm: frame dropping

Latest Darwin Streaming Server 6.0.3 release provides a rate adaptation algorithm based on the transmission of a single high bitrate coded video track that, in case of congestion, is forced to drop a certain percentage of frames in order to decrease the outgoing bitrate [4].

DSS defines 6 different quality levels that have associated different percentages of frame dropping. For instance, the highest quality level drops no frames, a medium level drops 70% of B frames, and the lowest level drops all B and P frames. In order to change the active quality level, DSS applies an algorithm that works with reported losses, RTT values, buffer usage ratio and delay when available. All this information is obtained from NADU and RR packets.

Initially, the algorithm allows a certain amount of time for the client to buffer the video. During this phase the outgoing bitrate is higher than codification bitrate to shorten buffering time, usually the double. After that it starts the adaptation phase.

Another problem is client's buffer management. The buffer can get full (overflow) or run out of data (under-run). Therefore, packet transmission is delayed or advanced according to client buffer feedback. This is controlled by the variable called Transmission Time.

This solution helps dealing with buffering problems such as overflow and under-run on client sides, also it benefits the rest of the network elements modifying the overall transmission rate; the main disadvantage is the poor quality of the displayed video due to frame dropping. The video becomes discontinuous too often and QoE is acceptable but not good.

4.5. Modified algorithm: track switching

The main advantage of the developed adaptation function is that instead of dropping frames of a video flow coded at a high bitrate, we transparently switch between alternative tracks containing the same information but coded at different bitrates. In terms of video quality and smoothness this second situation is much better than the first and results in a higher quality of experience as shown in [6]. Each track is associated to a quality level, which will vary depending on the client reports.

In order to change the quality level, and consequently of video track, we have defined a Quality Indicator (QI). When the QI overcomes a certain threshold, it changes to a higher quality level. When decreases under another threshold it switches to a lower quality level. To prevent repeated bouncing quality level changes in the boundaries of a threshold, QI is modified accordingly to additive increments when client reports are positive: RTCP RR messages report no losses and NADU messages report stable buffers and no delays. This QI shows multiplicative decrements when one of the previous conditions is not fulfilled. The QI parameters can be adjusted for each scenario and the update frequency depends on RTCP RR. The algorithm also controls the Transmission Time to accommodate client's buffers.

4.6. Swapping function

The main problem to perform track switching to mobile terminals like phones or PDA is doing it in a completely transparent way, as no protocols are currently defined. As introduced before, RTP clients only expect to receive information about the initially described tracks, all RTSP responses and RTP packets must be consistent with that information. To achieve this consistence, a swap function has been designed. The goal of the swap function is to assign the proper values to each track when the switching is performed; it must keep tracks time correct, adjust RTP sequence numbers, Payload Type and Timestamp and reflect that in all RTP packets, and finally manage track activation and deactivation.

5. Practical Experiments

Practical experiments carried out include the comparison of the system performance using: no adaptation at all, default adaptation system based in frame dropping, and track switching adaptation in a single hop architecture. In order to test a scenario

which requires many adaptation situations, we have forced the 802.11 network to work at 1 Mbps. Clients are two 3GPP-Rel6 compatible terminals (Nokia N95 and N80) and one non-compatible. The idea is to compare bandwidth distribution and video quality in each case; smoothness in video playing is also taken into account. The adaptable files include two video tracks (about 300 Kbps and 150 Kbps in average) and one audio track, so only video tracks will be switched if needed; the requested 3GP file is the same for all the terminals.

Following figures show the outgoing of Darwin Streaming Server. Blue and green lines are 3GPP compatible terminals and red line the non-compatible one. These lines represent outgoing data flows generated by the streaming server application belonging to sessions that are established and finished to test the system behaviour. Later, they are forwarded to the 1 Mbps IEEE 802.11 network interface of the streaming server Wilico node. Therefore, if the addition of all of them (black line) is higher than 1 Mbps there will be losses and many delays in the network interface. In fact, transmission problems begin earlier than 1 Mbps due to the 802.11 efficiency.

5.1. Non-adaptation scenario

Non-adaptation scenario presented in Figure 4 shows a fair bandwidth distribution among all clients. All new connections are accommodated in the server without decreasing the bitrate of already established ones. This leads that when the third connection is established ($t = 150s$) the aggregate output is much higher than 1 Mbps. The consequence is a very poor received video quality: bad image quality, frequent pauses to enter in a buffering period and also disconnections. Main problems are delays and loss of referenced frames that generate a chain of errors in the video play.

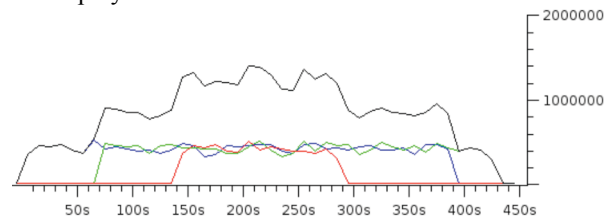


Figure 4: Server output bitrate (bps) - without adaptation.

5.2. Default adaptation scenario.

This scenario (Figure 5) provides adjustments in active session bitrates at server output. When green

client establishes a connection ($t = 50s$), blue client reduces its bitrate. The same happens at $t = 130s$ when red client enters. Note that the aggregate throughput never goes up to 1 Mbps. Server response to small condition variation is very fast, which can lead to a bouncing effect. The main advantage of this system is the fast adaptation to available network capacity and bandwidth equally distribution among clients.

In terms of video quality it has a relatively good behaviour because avoids disconnections due to buffering problems and reduces decoding errors (malformed frames or black pixel groups) due to referenced I frames losses. Nevertheless, QoE suffers from a very noticeable frame dropping and the experience is not very satisfactory.

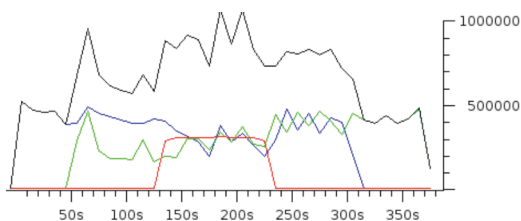


Figure 5: Server output bitrate (bps) - with default algorithm.

5.3. Track-switching scenario

In this scenario track switching and Transmission Time variation are performed. Figure 6 shows a situation similar to Figure 5 where new connections are established and the system adjusts the global outgoing throughput under 1 Mbps. Therefore rate adaptation works properly, like the DSS original adaptation algorithm. The system avoids bouncing changes, note that quality level decreasing is almost immediate when problems are found (new connection enters $t = 770s$) but opposite procedure takes a longer time (one client leaves $t = 540s$).

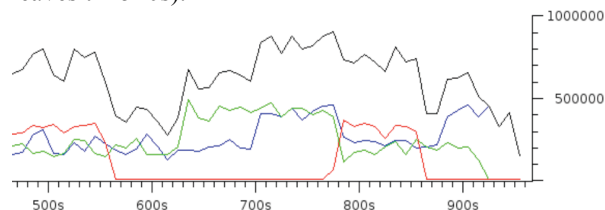


Figure 6: Server output bitrate (bps) - with track-switching algorithm.

The main advantage of this method is its higher level of QoE. The transaction from one quality level to another is smooth on the client player, so it is not a harmful situation at all and no forced drops are performed. In situations where bandwidth restrictions

apply, it is better to have all frames of a lower quality video, than a higher quality video with many lost frames [6].

6. Conclusion

This paper presents an innovative application that makes use of the newest communication technologies at different protocol architecture levels: new hardware platform (Wilico nodes), multihop IEEE 802.11 mesh network, dynamic AODV routing protocol, application to discover client devices capabilities and finally a modified open source streaming server to adapt video delivery to both device capabilities and network conditions.

Experimental results have shown that the system is nearly ready for commercial usage, and that the new track-switching stream adaptation algorithm is a proper solution to congested wireless networks that gives better QoE than the one provided with the to the date newest release of Darwin Streaming Server (6.0.3). The system has proved to work fine with different 3GPP compatible terminals.

7. Acknowledgements

This work has been supported in part by UPC Càtedra Red.es, FEDER and the Spanish Government through projects TEC2006-04504 and PROFIT Infopoints-NET. We also acknowledge the unconditional support from Futurlink.

8. References

- [1] 3GPP TS 26.234, "Transparent End-To-End Packet-Switched Streaming Service (PSS): Protocols and Codecs (Release 6)".
- [2] Futurlink: www.futurlink.com.
- [3] X. Sanchez-Loro, J. Casademont, J. L. Ferrer, J. Paradells. "A Proxy-based Solution for Device Capabilities Detection". Internet and Multimedia Systems and Applications – EuroIMSA 2007. Chamonix (France) 2007.
- [4] L. Cicco, S. Mascolo, V. Palmisano. "An Experimental Investigation of the End-to-End QoS of the Apple Darwin Streaming Server. ". Int. Conf. on Wired/Wireless communications, WWIC 2008. Tampere (Finland), 2008.
- [5] P. Frojdh, U. Horn, M. Kampmann, A. Nohlgren, M. Westerlund, "Adaptive streaming within the 3GPP packet-switched streaming service," IEEE Network, vol.20, no.2, pp.34-40, 2006.
- [6] O. Verscheure, P. Frossard and M. Hamdi, "MPEG-2 video Services over Packet Networks: Joint Effect of Encoding Rate and Data Loss on User-Oriented QoS". 8th Int. Workshop on Network and Operating Systems Support for Digital Audio and Video, NOSSDAV' 98. Cambridge, (United Kingdom), 1998.