A Dynamic Delay Equalization VoIP over Wireless Module for NS-2

Padraig O Flaithearta, Hugh Melvin

p.oflaithearta1@nuigalway.ie, hugh.melvin@nuigalway.ie

Discipline of Information Technology,
College of Engineering & Informatics,
National University of Ireland, Galway
Ireland

Both authors are members of the Performance Engineering Laboratory (PEL)
http://pel.it.nuigalway.ie

This research is funded by Údarás na Gaeltachta, and carried out at Acadamh na hOllscoilaiochta Gaeilge, Cárna, and at NUI Galway main campus.

Abstract

The Network Simulator NS-2 is a versatile extendable tool that is widely used in the networking community. It provides support for simulation of many protocols over networks including 802.11e. To investigate network performance, researchers can configure a network scenario with an easy-to-use scripting language, and then observe results generated by NS-2. As part of our overall research, this paper details our module extension for NS-2 that will dynamically prioritize within multiple VoIP sessions in order to equalize overall M2E delays. Our module will thus attempt to equalize QoS for all sessions within an Access Category using R-factor values from the ITU-T E-model to determine session prioritization.

Keywords

NS-2, E-Model, VoIP, Time Synchronization, 802.11e

1 Introduction

There is an ever increasing demand for delay sensitive applications over 802.11 among both personal and business users. The potential savings that VoIP in particular can provide are contributing to the demand for improvements in voice traffic Quality of Service (QoS). The 802.11e protocol was developed with time sensitive applications in mind, however, severe congestion leading to unacceptable delays and packet loss can still occur. Our research investigates how synchronized time implemented in end terminals can aid in optimizing 802.11e parameters to improve QoS for VoIP. In our earlier research, we have shown that by prioritizing wireless MAC access for certain VoIP sessions that have large overall one-way delays over VoIP sessions that have smaller one-way delays, we can reduce MAC and overall delay and thus improve their QoS, without significantly impacting on the QoS of the remaining VoIP sessions. We are employing a combination of running simulations using the NS-2 network simulator and, simultaneously, building a practical test bed to back up our results.

This paper details our NS-2 extension that dynamically optimizes 802.11e EDCA MAC parameters in order to equalize overall one-way delays among multiple VoIP sessions. The algorithm to dynamically vary parameters is informed by the E-Model R-factor, based on per-session each-way delay information. It will implement and maintain, where appropriate, an equalization of overall one-way delays within the Voice access category (AC_VO). Although this paper presents an all-VoIP scenario as the use case example, this extension is applicable to any EDCA access category or traffic makeup, and future work will include experiments using different traffic characteristics.

The remainder of the paper is structured as follows. Section 2 provides some background information including a general overview of our research. Section 3 looks at our method of QoS measurement namely the ITU-T E-Model, and also gives some background information and a description of the general structure of the NS-2 Network Simulator. Section 4 looks at our NS-2 Module in more detail, including the adaptive algorithm and some of its example working scenarios. Some information on the 802.11e NS-2 implementation is also covered here. Finally our conclusion is provided in Section 5

2 Background

Our research broadly involves an investigation into the improvements synchronized time can bring to VoIP over wireless networks. As outlined by ITU-T recommendation G.114, the acceptable limit of one-way delay for VoIP applications is approx 150ms. Although wireless MAC delays for multiple VoIP sessions through a single access point will be roughly equivalent, the overall one-way delay will typically vary greatly. For example, some calls take place over a LAN, with typical network delays < 10ms, and others over long distance with network delays > 100ms (Fig. 1).

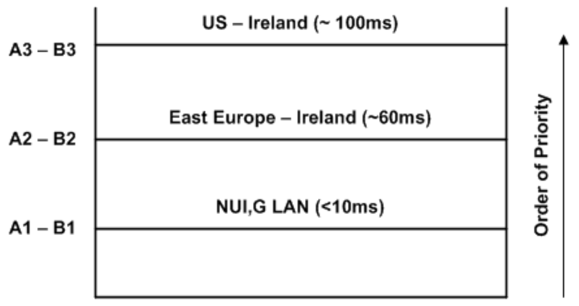

Fig. 1: Different Baseline Delays

With such latter network delays the total one-way delay (Sender including MAC contention, Network and Receiver) of packets could be close to, or may exceed the ITU-T G114 recommendation of 150msec (Fig. 2).

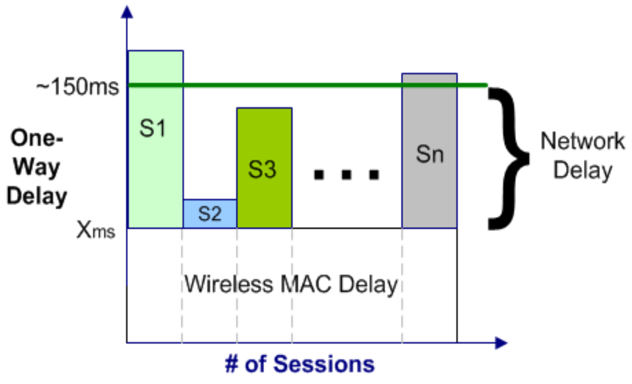

Fig. 2: Wireless MAC delay similar for all sessions. Total one-way delay can vary

All sessions within the 802.11e EDCA voice category have similar delays at the wireless MAC layer. We propose that if the precise one-way delay information for each VoIP session is known in both directions, EDCA parameters can be configured differently in real-time between VoIP sessions in order to optimize the channel delays, so that the sessions with the higher one-way network delays receive higher priority treatment at wireless MAC level relative to other VoIP sessions with lower delays. Essentially priority is assigned to voice sessions within the voice access category.

For a session that has a small one-way delay, packets can afford to wait for a longer time at the wireless MAC layer, on the condition that they don't cause a significant degradation in QoS for that session and that they are delivered within 150ms (e.g. "S2" in Fig. 3), whereas for a session that has a long network delay, packets have a shorter contention delay at the wireless MAC layer due to prioritization, thus reducing the overall one-way delay for that voice session (e.g. "S1" in Fig. 3).

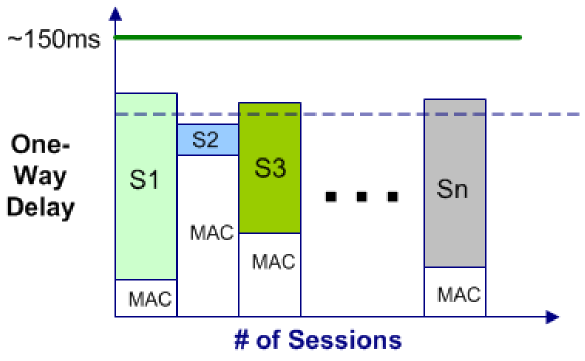

Fig.3: With Knowledge of Each way delays, we can achieve acceptable QoS by equalizing the delays among Voice Sessions by dynamically tuning EDCA parameters

Our earlier simulations have shown that we can improve the delay values for VoIP sessions with large existing non-Wireless MAC delays, by prioritizing these sessions over VoIP sessions with smaller non-Wireless MAC delays (Fig. 4).

For our simulations, an NS-2 topology runs multiple VoIP sessions over an 802.11b network. Each session is set up between a wireless and a wired node via an Access Point (AP) and a wired router. We have simulated different geographical distances by setting varying delays (Fig.1) in the wired network. The VoIP sessions running over the long-haul wired links were then prioritized over other VoIP sessions on the wireless side by tuning their 802.11e EDCA parameters.

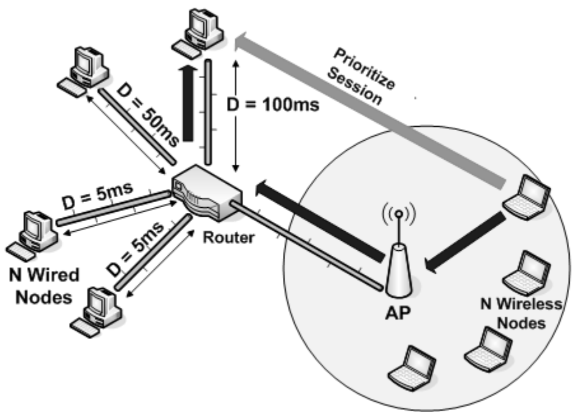

Fig. 4: NS-2 simulation topology. Sessions to be prioritized have preset wired link delays of 50ms and 100ms respectively.

Our methodology is to employ a mixture of simulation and real test-bed experiments. With NS-2, we are currently implementing an NS-2 extension module that will dynamically alter EDCA parameters between VoIP sessions based on precise one way delay information and the R-factor values. Along with our simulations, we are simultaneously building a real world wireless test-bed on which to carry out experimental tests to back up our simulation results.

3 Measurement of Quality

3.1 E-Model

QoS (Quality of Service) can be defined as a metric that measures the overall user satisfaction with network performance. QoS is application dependent and the QoS requirements may vary widely depending on the application. There are different metrics available to quantify the QoS for VoIP applications. One widely used metric is the ITU recommendation G.107 E-model.

The E-model computes a predictive estimation of the subjective quality of the packetized voice from transmission parameters. The e-model outputs a number called the R-factor, which is computed as a function of delay, packet loss, equipment impairment factors, and user quality call expectation. The value is calculated as follows:

$$\mathbf{R = Ro - Is - Id - Ie,eff + A} \qquad [1]$$

where **Ro** is the basic signal-to-noise ratio (received speech level relative to circuit and acoustic noise). **Is** accounts for the impairments which occur with the voice signal. **Id** is the sum of all impairments due to delay and echo effects. With respect to M2E delay (Fig. 5), **Id** (in presence of good echo cancellation) can be approximated by 10 units per 100msec up to 150msec, and by 12-13 units per 100msec thereafter [2]. The **Ie,eff** is the effective equipment impairment factor, taking into account the codec and its tolerance to packet losses [1]. **A** is a bonus factor that models the user expectation of the technology employed and can be used to take into account the fact that the user will tolerate some decrease in transmission quality when using certain technologies.

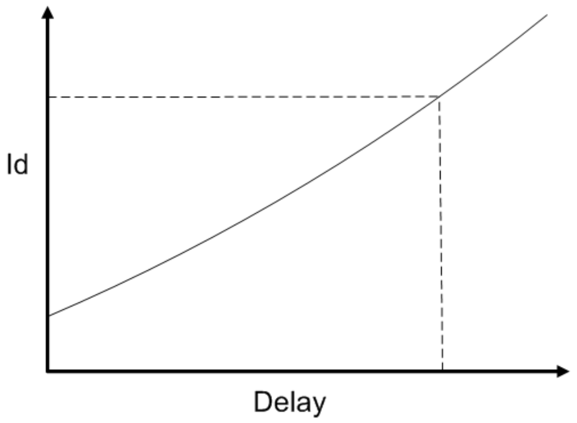

Fig. 5: Id takes Delay and Echo into consideration. However, for our research we are more concerned with the effect of delay

The major factors affecting QoS in a wireless network are throughput, packet loss, packet delays and jitter. Some of the main problems associated with achieving good QoS in a wireless environment are that the wireless channel can change with time and so it is possible that links between two nodes could break during a voice session.

The R-factor values are defined and categorized as shown in Table 1.

$90 \le R < 100$	Best: Very satisfied
$80 \le R < 90$	High: Satisfied
$70 \le R < 80$	Medium: Some users dissatisfied
$60 \le R < 70$	Low: Many users dissatisfied
$50 \le R < 60$	Poor: Nearly all users dissatisfied

Table 1: Definition of Categories of Speech transmission Quality

Our NS-2 module will employ an R-factor based system as a measurement of quality for dynamically determining the necessary parameters in order to carry out its functionality. When the network is monitored during a simulation, an R-factor will be calculated for each individual session, and these R-factor values will be then used to calculate which sessions require prioritization over other sessions.

3.2 The Network Simulator NS-2

The NS-2 is an object oriented, discrete event simulator written in C++, with an OTcl (Object-Oriented Tool Command language) interpreter as a frontend [3]. A set of discrete events are executed in order by a simulator object. OTcl is its primary command and configuration language, it implements network protocols such as TCP and UDP over wired and wireless networks, and also traffic behaviour such as FTP, Telnet, Web, CBR (constant bit rate) & VBR(variable bit rate). NS-2 implements router queue management mechanisms, multicasting and some of the MAC layer protocols (which include 802.11b MAC layer specification) for LAN simulations. The NS-2 package also includes tools for analysis and display of the simulation results including NAM(Network Animator).

The backbone of NS-2 is basically written in C++, with an OTcl script interpreter forming the front-end. It supports a class hierarchy in C++, called Compiled hierarchy & a similar one within the OTcl interpreter, called interpreter hierarchy. A one to one correspondence between classes of these two hierarchies exists (Fig. 6). The root of the hierarchy is Class TclObject. The users create new simulator objects through interpreter.

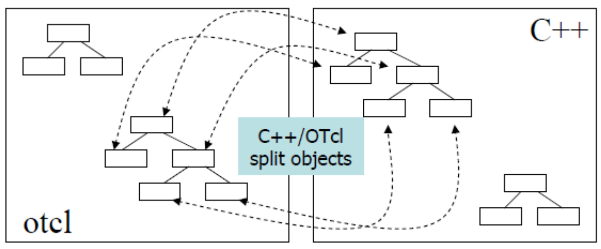

Fig. 6

The simulator performs two main functions, it performs detailed and efficient simulations of protocols, for which C++ is required for handling bytes, packet headers and efficient algorithm implementation. The other function is allowing users to vary parameters quickly to compute different outputs from simulations, which is controlled by the scripting language Tcl.

3.3 General Structure of NS-2

The NS developer works on Tcl, runs simulations in Tcl using the simulator objects in OTcl library. The event scheduler and most of the components are implemented in C++ and available to OTcl through a one-to-one OTcl linkage. NS is basically an Object Oriented extended Tcl interpreter which uses the network component libraries and the network setup libraries (also known as the plumbing modules).

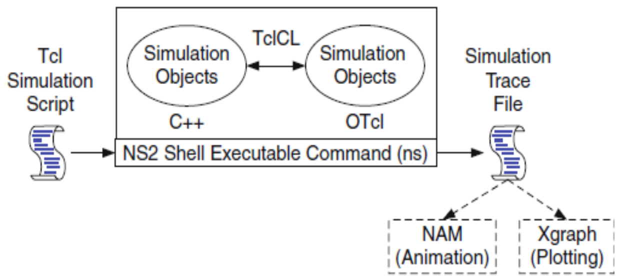

Fig. 7 [4]

In order to run a basic simulation, an OTcl script language program does the following;

- Initiates an event scheduler,
- Sets up a network topology using the Network Component Objects
- Connects the network together using the functions in the Plumbing library
- Co-ordinates the timing of the traffic source.

The event scheduler is used by those network components that handle packets to issue an event for a packet. Essentially a packet can be viewed as an event which triggers at a particular time. Simulation results are stored in trace files, which are used for analysis of the simulated data, and graphs relating to the quality of the network can be generated from these trace files. A graphical tool known as Network Animator (NAM) also uses these files to automatically generate a topological view of the network.

Given the two language structure of NS-2, certain tasks can be performed in a Tcl script, such as configuring network topology or assigning traffic characteristics, whereas more complex tasks involve editing OTcl and C++ code which requires recompilation. In order for our module to work dynamically and seamlessly, it is being built in C++, and will be located in the ns_2_x/mac/802_11e directory.

4 NS-2 VoIP Module

The goal of our NS-2 module is to dynamically tune 802.11e EDCA parameters for wireless nodes (within an Access Category) that are hosting real time traffic sessions, in order to equalize the overall M2E delay for all sessions where appropriate. This will be done based on information provided by the E-model.

4.1 Adaptive algorithm

Our extension module involves adapting EDCA parameters according to an algorithm that computes the optimal settings based on each way delay calculations. This algorithm will calculate in real-time the R-factor (including delay) for each individual session, and then use this information to tune parameters in order to gradually equalize the overall delays for all sessions (Fig. 3).

The first goal of our algorithm will be to check whether a session has a large one way delay relative to other sessions, and if so, it will aim to reduce the delay for that session(s), once it does not compromise the QoS of the remaining sessions. The objective thereafter will be to concurrently pursue and achieve a state of equalization of delay for all remaining sessions.

Depending on the scenario and the desired outcome, there are varying combinations of EDCA parameters that can be tuned. In a scenario where there are multiple VoIP sessions running on an 802.11 network (such as in Fig. 4), and the goal is to equalize M2E delays for all sessions, as mentioned earlier, the algorithm uses the R-factor to measure the QoS for each session. Where R-factor is below 70, a user would experience a "medium" level of QoS, and some users would be dissatisfied according to the E-model. QoS for individual sessions would then be calculated in order to determine how 802.11e EDCA parameters will be tuned. We are building this into our module as the basis of our delay equalization algorithm.

4.2 Example working scenario

Where a specific VoIP call has an R-factor rating below 70 based on a M2E delay of >150ms due largely to a non-Wireless MAC delay, a simple prioritization of this call over others could render it satisfactory for a user simply by altering the AIFSN or CW_Max values for that session. To achieve this (and repeat if necessary for other sessions with similarly large delays) would be the first goal of our module. The aim would then be to gradually achieve a state of equalization of the M2E delays for all sessions in the network.

However, in a scenario (Fig. 8) where a VoIP call has a M2E delay of ~500ms and the remaining sessions have delays of approx 100ms, it would not make sense to sacrifice the QoS of most sessions in order to try to satisfy the QoS of a session that cannot be significantly improved.

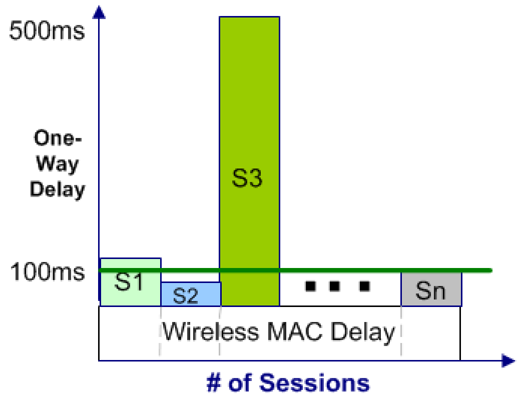

Fig. 8

As per Fig. 9, no significant improvement would be achieved for "S3", however the remaining sessions would have a significantly degraded QoS.

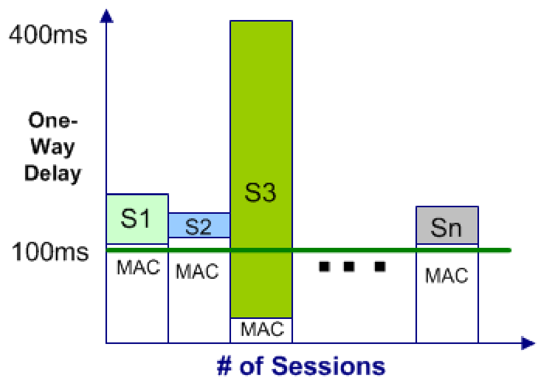

Fig. 9

4.3 802.11e in NS-2

All of our development takes place within the *...ns2.x/mac/802.11e/* folder [5]. It is here where our algorithm is being implemented. We are developing a C++ procedure that implements our algorithm to periodically adapt 802.11e parameters for each individual wireless node, based on R-factor information associated with the real-time session being hosted on that node.

At the beginning of a simulation the MAC requests the 802.11e parameters for each priority [6]. The AIFS, CW_MIN, CW_MAX and TXOP Limit values are assigned to each queue before traffic sessions begin to transmit. After a certain period, R-factor values will be calculated, and 802.11e parameters will be re-assigned if required. This process will be repeated periodically for the duration of the simulation. On completion of the development of our module, we must complete validation testing within NS-2. This is to ensure that our code changes do not impact on other parts of the simulator.

5 Conclusion

As part of our overall research, this paper details our module extension for NS-2 that aims to equalize M2E delays for multiple VoIP sessions. Our module, which is currently under development, gradually equalizes the overall one way delays for all sessions within an Access Category using information provided by R-factor values from the ITU-T E-model to determine session prioritization.

6 References

[1] ITU-T Recommendation G.107 – “The E-Model: A Computational Model for Use in Transmission Planning”

[2] J. Janssen, D. De Vleesschauer, M. Buchli, G. H. Petit, Assessing Voice Quality in Packet-Based Telephony, IEEE Internet Computing, May/June 2002

[3] The Network Simulator ns-2, Documentation, http://www.isi.edu/nsnam/ns/ns-documentation.html

[4] T. Issariyakul, E. Hossain, Introduction to the Network Simulator, Springer Science+Business Media, LLC. 2009

[5] An 802.11e EDCA and CFB Simulation Model for NS-2, S. Wieholter, C. Hoene. http://www.tkn.tu-berlin.de/research/802.11e_ns2/

[6] S. Wieholter, C. Hoene, Design and Verification of an IEEE 802.11e EDCF simulation model in NS-2, TKN Berlin, 2003