

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Pursuing Credibility in Performance Evaluation of VoIP Over Wireless Mesh Networks

Edjair Mota<sup>1</sup>, Edjard Mota<sup>1</sup>, Leandro Carvalho<sup>1</sup>,  
Andréa Nascimento<sup>1</sup> and Christian Hoene<sup>2</sup>

<sup>1</sup>*Federal University of Amazonas*

<sup>2</sup>*Tübingen University*

<sup>1</sup>*Brazil*

<sup>2</sup>*Germany*

## 1. Introduction

There has been an increasingly interest in real-time multimedia services over wireless networks in the last few years, for the most part due to the proliferation of powerful mobile devices, and the potential ubiquity of wireless networks. Nevertheless, there are some constraints that make their deployment over Wireless Mesh Networks (WMNs) somewhat difficult. Due to the dynamics of WMNs, there are significant challenges in the design and optimization of such services. Impairments like packet loss, delay and jitter affects the end-to-end speech quality (Carvalho, 2004). Experimenters have been proposing solutions to the challenges found so far, and comparing them before implementation is a mandatory task. There exists a necessity of designing efficient tools for enhancing the computational effort of the performance modeling and analysis of VoIP over WMNs. Structural complexity of such highly dynamic systems causes that in many situations computer simulation is the only way of investigating their behavior in a controllable manner, allowing the experimenter to conduct independent and repeatable experiments, in addition to the comparison of a large number of system alternatives. Stochastic simulation is a flexible, yet powerful tool for scientifically getting insight into the characteristics of a system being investigated. However, to ensure reproducible results, stochastic simulation imposes its own set of rules. The credibility of a performance evaluation study is greatly affected by the problem formulation, model validation, experimental design, and proper analysis of simulation outcomes.

Therefore, a fine-tuning of the parameters within a simulator is indispensable, so that it closely tracks the behavior of a real network. However, the lack of rigor in following the simulation methodology threatens the credibility of the published research (Pawlikowski et al., 2002; Andel & Yasinac, 2006; Kurkowski et al., 2005).

The aim of this chapter is to provide a detailed discussion of these aspects. To do so, we used as a starting point the observation that the optimized use of the bandwidth of wireless networks definitely affects the quality of VoIP calls over WMN. Since the payload size of VoIP packets is usually smaller than the header size, much network resource is spent for conveying control information instead of data information. Hence, VoIP header compression is an alternative to reduce the use of the bandwidth needed to transmit control information, thereby increasing the percentage of bandwidth used to carry payload information.

However, this mechanism can make the VoIP system less tolerant to packet loss, which can be harmful in WMN, due to its high rate of packet loss. Additionally, in a multi-hop wireless environment, simple schemes of header compression may not be enough to increase or maintain the speech quality. An interesting alternative approach in this context is the use of header compression in conjunction with packet aggregation (Nascimento, 2009), aiming to eliminate the intolerance to packet loss without reducing the compression gain.

Although these issues are not unique to simulation of multimedia transmission over wireless mesh networks, we focus on issues affecting the WMN research community interested in VoIP transmissions over WMN. After modeling thoroughly the issues of VoIP over WMN, we built a simulation model of a real scenario at the Federal University of Amazonas, where we have been measuring the speech quality of VoIP transmissions by means of a tool developed by our groups. Then, we modeled a bidirectional VoIP traffic, and proposed a carefully selected set of experiments and simulation details such as the sources of randomness and analysis of the output data, closely following sound methodology for each phase of the experimentation with simulation.

## 2. Background

### 2.1 Wireless mesh networks

Wireless mesh network (WMN) is a promising communication technology that has been successfully tested in academic trials and is a mature candidate to implement metropolitan area networks. Compared to fiber and copper based access networks, it can be easily deployed, maintained and expanded on demand. It offers network robustness, and reliable service coverage, besides its low costs of installation and operation. In many cities, such as Berlin and Bern, WMN has been used to provide Internet access for many users.

In its more general form, a WMN consists of a set of wireless mesh routers (WMRs) that interconnect with each other via wireless medium to form a wireless backbone. These WMRs are usually stationary and work as access points or gateways to the Internet to wireless mesh clients (WMCs). High fault tolerance can be achieved in the presence of network failures, improper operation of WMRs, or wireless link inherent variabilities. Based on graph theory, (Lili et al., 2009) suggested a method to analyze the fault-tolerant and communication delay in a wireless mesh network, while (Queiroz, 2009) investigated the routing table maintenance issue, by proposing and evaluating the feasibility of applying the Bounded Incremental Computation model (Ramalingam & Reps, 1996) to satisfy scalability issues. Such kind of improvement is essential to real time multimedia application in order to reduce the end-to-end delay.

Even being accepted as a good solution to provide access to the telephone service, the WMN technology poses some problems being currently investigated such as routing algorithms, self-management strategies, interference, to say a few. To understand how to achieve the same level of quality of multimedia applications in wired networks, it is imperative to grasp the nature of real-time multimedia traffic, and then to compare it against the problems related to the quality of multimedia applications.

### 2.2 Voice over IP

A VoIP call placed between two participants requires three basic types of protocol: signaling, media transmission, and media control. The signaling protocols (e.g. H.323, SIP) establish, maintain and terminate a connection, which should be understood as an association between applications, with no physical channel or network resources associated with it. The media

transmission protocols (e.g. RTP) are responsible for carrying out the actual content of the call – the speaker’s voice – encoded in bits. Finally, the media control protocols (e.g. RTCP) convey voice packet transmission parameters and statistics, ensuring better end-to-end packet delivery. In this work, our attention is focused upon the voice stream. So, lets briefly introduce the main logical VoIP components of the media transmission path.

As illustrated in Figure 1, the sender’s voice is captured by a microphone and digitalized by an A/D conversor. The resulting discrete signal is then encoded and compressed by some codec into voice frames. One or more frames can be encapsulated into a voice packet by adding RTP, UDP and IP headers. Next, the voice packets are dispatched to the IP network, where they can get lost due to congestion or transmission errors. The transmission delay of packets – i.e, the time needed to deliver a packet from the sender to the receiver – is variable and depends on the current network condition and the routing path (Hoene et al., 2006).

At the receiver, the arriving packets are inserted into a dejitter buffer, also known as playout buffer, where they are temporarily stored to be isochronously played out. If packets are too late to be played out in time, they are discarded and considered as lost by the application. After the dejitter buffer the speech frames are decoded. If a frame is missing, the decoder fills the gap by applying some Packet Loss Concealment (PLC) algorithm. Finally, the digital signal is transformed into an acoustic signal.

Since IP networks were not designed to transport real-time traffic, an important aspect in VoIP communications is the assessment of speech quality. It is imperative that new voice services undergo a significant amount of testing to evaluate their performance. Speech quality is a complex psychoacoustic outcome of the perception process of the human auditory system (Grancharov & Kleijn, 2008). Its measurement can be carried out using either subjective or objective methods.

Subjective methods, specified in ITU-T Rec. P.800 (ITU-T, 1996), require that a pool of listeners rates a series of audio files using a five-level scale (1 – bad, 2 – poor, 3 – fair, 4 – good, 5 – excellent). The average of all scores thus obtained for speech produced by a particular system represents its Mean Opinion Score (MOS). The main reason for the popularity of this test is its simplicity (Grancharov & Kleijn, 2008). However, the involvement of human listeners makes them expensive and time consuming. Moreover, subjective tests are not suitable to monitor the QoS of a network on a daily basis. This has made objective methods very attractive for meeting the demands for voice quality measurement in communications networks.

Among the objective (or instrumental) methods, the E-model, defined in the ITU-T Rec. G.107 (ITU-T, 1998), is one of the most used. It computes, in a psychoacoustic scale, the contribution of all impairment factors that affect voice quality. This does not imply that the factors are uncorrelated, but only that their contributions to the estimated impairments are independent and each impairment factor can be computed separately (Myakotnykh & Thompson, 2009). Although initially designed for transmission planning of telecommunication systems (Raake, 2006; ITU-T, 1998), the E-model was modified by (Clark, 2003; Carvalho et al., 2005) to be used for VoIP network monitoring.

The output of the E-model is the *R* factor, which ranges from 0 (worst) to 100 (excellent) and

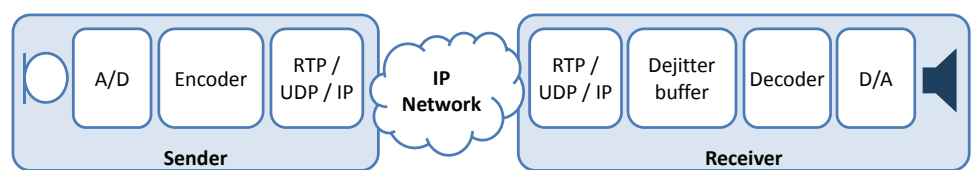


Fig. 1. VoIP components of the media transmission path.

can be converted to the MOS scale. Voice calls whose  $R$  factor value is below 60 (or 3.6 in MOS scale) are not recommended (ITU-T, 1998). For VoIP systems, the  $R$  factor can be obtained by the following simplified expression (Carvalho et al., 2005):

$$R = 93.2 - Id(codec, delay) - Ie, eff(codec, loss, PLB) \quad (1)$$

where  $Id$  represents the impairments associated with end-to-end delay, and  $Ie, eff$  represents the impairments associated with codec compression, packet loss rate and packet loss behavior (PLB) during the call.

The measurement tool proposed in (Carvalho et al., 2005) for speech quality evaluation based on the E-model was adapted as a patch to the Network Simulator code (McCanne & Floyd, 2000). It was used for validating our simulation models concerning the transmission of VoIP over WMNs.

### 2.3 Performance evaluation

Measurement and stochastic simulation are the main tools commonly used to assess the performance of multimedia transmission over WMNs. Success in the development of complex wireless networks is partially related to the ability of predicting their performance already in the design phase and subsequent phases of the project as well. Dynamic increasing of the complexity of such networks and the growth of the number of users require efficient tools for analyzing and improving their performance. Analytical methods of analysis are neither general nor detailed enough, and in order to get tractability, they need sometimes to make assumptions that require experimental validation. On the other hand, simulation, formerly known as a last resort method, is a flexible and powerful tool adequate for prototyping such complex systems. To the factors that have additionally stimulated the use of simulation, one could include: faster processors, larger-memory machines and trends in hardware developments (e.g. massively parallel processors, and clusters of distributed workstations). Straightforward simulation of complex systems, such as WMNs, takes frequently a prohibitively amount of computer time to obtain statistically valid estimates, despite increasing processing speed of modern computers. It is not rare the simulation take some hours to estimate a performance metric corresponding to a few seconds of real time. (Mota et al., 2000) investigated the influence of jitter on the quality of service offered by a wireless link, and reported a simulation time as long as 180 hours using just 9 wireless terminals, though simulation has been executed in a fast workstation dedicated to that purpose.

Such phenomenon results from the statistical nature of the simulation experiments. Most simulation models contain stochastic input variables, and, thereby, stochastic output variables, the last ones being used for estimating the characteristics of the performance parameters of the simulated system. In order to obtain an accurate estimate with known (small) statistical error, it is necessary to collect and analyze sequentially a substantial amount of simulation output data, and this can require a long simulation run.

Efficient statistical tools can be used to impact the running time of an algorithm by choosing an estimator with substantially lower computational demand. It would be a mistake to think that more processing power can replace the necessity for such tools, since the associated pitfalls can be magnified as well (Glynn & Heidelberger, 1992). The need for effective statistical methods to analyze output data from discrete event simulation has concerned simulation users as early as (Conway, 1963). Development of accurate methods of statistical analysis of simulation output data has attracted a considerable scientific interest and effort.



Even though, credibility of stochastic simulation has been questioned when applied to practical problems, mainly due to the application of not robust methodology for simulation projects, which should comprise at least the following:

- The correct definition of the problem.
- An accurate design of the conceptual model.
- The formulation of inputs, assumptions, and processes definition.
- Build of a valid and verified model.
- Design of experiments.
- Proper analysis of the simulation output data.

### 3. Model credibility

#### 3.1 Problem definition

To formulate a problem is so important as to solve it. There is a claim credited to Einstein that states: "The formulation of a problem is often more essential than its solution, which may be merely a matter of mathematical or experimental skill". The comprehension of how the system works and what are the main specific questions the experimenter wants to investigate, will drive the decisions of which performance measures are of real interest.

Experts are of the opinion that the experimenter should write a list of the specific questions the model will address, otherwise it will be difficult to determine the appropriate level of details the simulation model will have. As simulation's detail increases, development time and simulation execution time also increase. Omitting details, on the other hand, can lead to erroneous results. (Balci & Nance, 1985) formally stated that the verification of the problem definition is an explicit requirement of model credibility, and proposed high-level procedure for problem formulation, and a questionnaire with 38 indicators for evaluating a formulated problem.

#### 3.2 Sources of randomness

The state of a WMN can be described by a stochastic or random process, that is nothing but a collection of random variables observed along a time window. So, input variables of a WMN simulation model, such as the transmission range of each WMC, the size of each packet transmitted, the packet arrival rate, the duration of periods ON an OFF of a VoIP source, etc, are random variables that need to be:

1. Precisely defined by means of measurements or well-established assumptions.
2. Generated with its specific probability distribution, inside the simulation model during execution time.

The generation of a random variate - a particular value of a random variable - is based on uniformly distributed random numbers over the interval  $[0, 1)$ , the elementary sources of randomness in stochastic simulation. In fact, they are not really random, since digital computers use recursive mathematical relations to produce such numbers. Therefore, it is more appropriate to call them pseudo-random numbers (PRNs).

Pseudo-random numbers generators (PRNGs) lie in the heart of any stochastic simulation methodology, and one must be sure that its cycle is long enough in order to avoid any kind of correlation among the input random variables. This problem is accentuated when there is

a large number of random variables in the simulation model. Care must be taken concerning PRNGs with small periods, since with the growth of CPU frequencies, a large amount of random numbers can be generated in a few seconds (Pawlikowski et al., 2002). In this case, by exhausting the period, the sequence of PRNs will be soon repeated, yielding then correlated random variables, and compromising the quality of the results.

As the communication systems become even more sophisticated, their simulations require more and more pseudo-random numbers which are sensitive to the quality of the underlying generators (L'Ecuyer, 2001). One of the most popular simulation packages for modeling WMN is the so called ns-2 (Network Simulator) (McCanne & Floyd, 2000). In 2002, (Weigle, 2006) added an implementation of the MRG32k3, a combined multiple recursive generator (L'Ecuyer, 1999), since it has a longer period, and provides a larger number of independent PRNs substreams, which can be assigned to different random variables. This is a very important issue, and could be verified before using a simulation package. We have been encouraging our students to test additional robust PRNGs, such as Mersenne Twister (Matsumoto & Nishimura, 1998) and Quantum Random Bit Generator – QRBG (Stevanović et al., 2008).

### 3.3 Valid model

Model validation is the process of establishing whether a simulation model possesses a satisfactory range of accuracy consistent with the real system being investigated, while model verification is the process of ensuring that the computer program describing the simulations is implemented correctly. Being designed to answer a variety of questions, the validity of the model needs to be determined with respect to each question, that is, a simulation model is not a universal representation of a system, but instead it should be an accurate representation for a set of experimental conditions. So, a model may be valid for one set of experimental conditions and invalid for another.

Although it is a mandatory task, it is often time consuming to determine that a simulation model of a WMN is valid over the complete domain of its intended applicability. According to (Law & McComas, 1991), this phase can take about 30%–40% of the study time. Tests and evaluations should be conducted until sufficient confidence is obtained and a model can be considered valid for its intended application (Sargent, 2008).

A valid simulation model for a WMN is a set of parameters, assumptions, limitations and features of a real system. This model must also address the occurrence of errors and failures inherent, or not, to the system. This process must be carefully conducted to not introduce modeling errors. It should be a very good practice to present the validation of the used model, and the corresponding deployed methodology so independent experimenters can replicate the results. Validation against a real-world implementation, as advocated by (Andel & Yasinac, 2006), it is not always possible, since the system might not even exist. Moreover, high fidelity, as said previously, is often time consuming, and not flexible enough. Therefore, (Sargent, 2008) suggests a number of pragmatic validation techniques, which includes:

- Comparison to other models that have already been validated.
- Comparison to known results of analytical models, if available.
- Comparison of the similarity among corresponding events of the real system.
- Comparison of the behavior under extreme conditions.
- Trace the behavior of different entities in the model.

- Sensitivity analysis, that is, the investigation of potential changes and errors due changes in the simulation model inputs.

For the sake of example, Ivanov and colleagues (Ivanov et al., 2007) presented a practical example of experimental results validation of a wireless model written with the Network Simulator (McCanne & Floyd, 2000) package for different network performance metrics. They followed the approach from (Naylor et al., 1967), to validate the simulation model of a static ad-hoc networks with 16 stations by using the NS-2. The objective of the simulation was to send a MPEG4 video stream from a sender node to a receiving node, with a maximum of six hops. The validation methodology is composed of three phases:

**Face validity** This phase is based on the aid of experienced persons in the field, together with the observation of real system, aiming to achieve high degree of realism. They chose the more adequate propagation model and MAC parameters, and by means of measurements on the real wireless network, they found the values to set up those parameters.

**Validation of Model Assumption** In this phase, they validated the assumptions of the shadowing propagation model by comparing model-generated and measured signal power values.

**Validation of input-output transformation** In this phase, they compared the outputs collected from the model and the real system.

### 3.4 Design of experiments

To achieve full credibility of a WMN simulation study, besides developing a valid simulation model, one needs exercise it in valid experiments in order to observe its behavior and draw conclusions on the real network. Careful planning of what to do with the model can save time and efforts during the investigation, making the study efficient. Documentation of the following issues can be regarded as a robust practice.

**Purpose of the simulation study** The simple statement of this issue will drive the overall planning. Certainly, as the study advances and we get deeper understanding of the system, the ultimate goals can be improved.

**Relevant performance measures** By default, most simulation packages deliver a set of responses that could be avoided if they are not of interest, since the corresponding time frame could be used to expand the understanding of the subtleties of WMN configurations.

**Type of simulation** Sometimes, the problem definition constraints our choices to the deployment of terminating simulation. For example, by evaluating the speech quality of a VoIP transmission over a WMN, we can choose a typical conversation duration of 60 seconds. So, there is no question about starting or stopping the simulation. A common practice is to define the number of times the simulation will be repeated, write down the intermediate results, and average them at the end of the overall executions. We have been adopting a different approach based on steady-state simulation approach. To mitigate the problem of initialization bias, we rely on Akaroa 2.28 (Ewing et al., 1999) to determine the length of the warm-up period, during which data collected during are not representative of the actual average values of the parameters being simulated, and cannot be used to produce good estimates of steady-state parameters. To rely on arbitrary choices for the run length of the simulation is an unacceptable practice, which compromises the credibility of the entire study.



**Experimental Design** The goal of a proper experimental design is to obtain the maximum information with the minimum number of experiments. A factor of an experiment is a controlled independent variable, whose levels are set by the experimenter. The factors can range from categorical factors such as routing protocols to quantitative factors such as network size, channel capacity, or transmission range (Totaro & Perkins, 2005). It is important to understand the relationship between the factors since they impact strongly the performance metrics. Proper analysis requires that the effects of each factor be isolated from those of others so that meaningful statements can be made about different levels of the factor.

As a simple checklist for this analysis, we can enumerate:

1. Define the factors and their respective levels, or values, they can take on;
2. Define the variables that will be measured to describe the outcome of the experimental runs (response variables), and examine their precision.
3. Plan the experiments. Among the available standard designs, choose one that is compatible with the study objective, number of design variables and precision of measurements, and has a reasonable cost. Factorial designs are very simple, though useful in preliminary investigation, especially for deciding which factors are of great impact on the system response (the performance metric). The advantages of factorial designs over one-factor-at-a-time experiments is that they are more efficient and they allow interactions to be detected. To thoroughly know the interaction among the factors, a more sophisticated design must be used. The approach adopted in (C.L.Barrett et al., 2002) is enough in our problem of interest. The authors setup a factorial experimental design to characterize the interaction between the factors of a mobile ad-hoc networks such as MAC, routing protocols, and nodes' speed. To characterize the interaction between the factors, they used ANOVA (analysis of variance), a well-known statistical procedure.

### 3.5 Output analysis

A satisfactory level of credibility of the final results cannot be obtained without assessing their statistical errors. Neglecting the proper statistical analysis of simulation output data cannot be justified by the fact that some stochastic simulation studies might require sophisticated statistical techniques.

A difficult issue is the nature of the output observations of a simulation model. Observations collected during typical stochastic simulations are usually strongly correlated, and the classical settings for assessing the sample variance cannot be applied directly. Neglecting the existence of statistical correlation can result in excessively optimistic confidence intervals. For a thorough treatment of this and related questions, please refer to (Pawlikowski, 1990). The ultimate objective of run length control is to terminate the simulation as soon as the desired precision of relative width of confidence interval is achieved. There is a trade-off since one needs a reasonable amount of data to get the desired accuracy, but on the other hand this can lengthen the completion time. Considering that early stopping leads to inaccurate results, it is mandatory to decrease the computational demand of simulating steady-state parameters (Mota, 2002).

Typically, the run length of a stochastic simulation experiment is determined either by assigning the amount of simulation time before initiating the experiment or by letting the simulation run until a prescribed condition occurs. The latter approach, known as sequential

procedure, gather observations at the output of the simulation model to investigate the performance metrics of interest, and a decision has to be taken to stop the sampling. It is evident that the number of observations required to terminate the experiment is a random variable since it depends on the outcome of the observations.

According to this thought, carefully-designed sequential procedures can be economical in the sense that we may reach a decision earlier compared to fixed-sample-sized experiments. Additionally, to decrease computational demands of intensive stochastic simulation one can dedicate more resources to the simulation experiment by means of parallel computing. Efficient tools for automatically analyzing simulation output data should be based on secure and robust methods that can be broadly and safely applied to a wide range of models without requiring from simulation practitioners highly specialized knowledge. To improve the credibility of our simulation to investigate the proposal of using bandwidth efficiently for carrying VoIP over WMN, we used a combination of these approaches, namely, we applied a sequential procedure based on spectral analysis (Heidelberger & Welch, 1981) under Akaroa-2, an environment of Multiple Replications in Parallel (MRIP) (Ewing et al., 1999).

Akaroa-2 enables the same sequential simulation model be executed in different processors in parallel, aiming to produce independent and identically distributed observations by initiating each replication with strictly non-overlapping streams of pseudo-random numbers. It controls the run length and the accuracy of final results.

This environment solve automatically some critical problems of stochastic simulation of complex systems:

1. Minimization of bias of steady-state estimates caused by initial conditions. Except for regenerative simulations, data collected during *transient phase* are not representative of the actual average values of the parameters being simulated, and cannot be used to produce good estimates of steady-state parameters. The determination of its length is a challenging task carried out by a sequential procedure based on spectral analysis. Underestimation of the length of the transient phase leads to bias in the final estimate. Overestimation, on the other hand, throws away information on the steady state and this can increase the variance of the estimator.
2. Estimation of the sample variance of a performance measure and its confidence interval in the case of correlated observations in equilibrium state;
3. Stopping the simulation within a desired precision selected by the experimenter.

Akaroa-2 was designed for full automatic parallelization of common sequential simulation models, and full automated control of run length for accuracy of the final results Ewing et al. (1999). An instance of a sequential simulation model is launched on a number of workstations (operating as simulation engines) connected via a network, and a central process takes care of collecting asynchronously intermediate estimates from each processor and calculates conveniently an overall estimate.

The only things synchronized in Akaroa-2 are substreams of pseudo-random numbers to avoid overlapping among them, and the load of the same simulation model into the memory of different processors, but in general this time can be considered negligible and imposes no obstacle.

Akaroa-2 enables the same simulation model be executed in different processors in parallel, aiming to produce IID observations by initiating each replication with strictly non-overlapping streams of pseudo-random numbers provided by a combined multiple recursive generator (CMRG) (L'Ecuyer, 1999).

Essentially, a master process (*Akmaster*) is started on a processor, which acts as a manager, while one or more slave processes (*akslave*) are started on each processor that takes part in the simulation experiment, forming a pool of simulation engines (see Figure 2). Akaroa-2 takes care of the fundamental tasks of launching the same simulation model on the processors belonging to that pool, controlling the whole experiment and offering an automated control of the accuracy of the simulation output.

At the beginning, the stationary Schruben test (Schruben et al., 1983) is applied locally within each replication, to determine the onset of steady state conditions in each time-stream separately and the sequential version of a confidence interval procedure is used to estimate the variance of local estimators at consecutive checkpoints, each simulation engine following its own sequence of checkpoints.

Each simulation engine keeps on generating output observations, and when the amount of collected observations is sufficient to yield a reasonable estimate, we say that a checkpoint is achieved, and it is time the local analyzer to submit an estimate to the global analyzer, located in the processor running *akmaster*.

The global analyzer calculates a global estimate, based on local estimates delivered by individual engines, and verifies if the required precision was reached, in which case the overall simulation is finished. Otherwise, more local observations are required, so simulation engines continue their activities.

Whenever a checkpoint is achieved, the current local estimate and its variance are sent to the global analyzer that computes the current value of the global estimate and its precision.

NS-2 does not provide support for statistical analysis of the simulation results, but in order to control the simulation run length, ns-2 and Akaroa-2 can be integrated. Another advantage of this integration is the control of achievable speed-up by adding more processors to be run

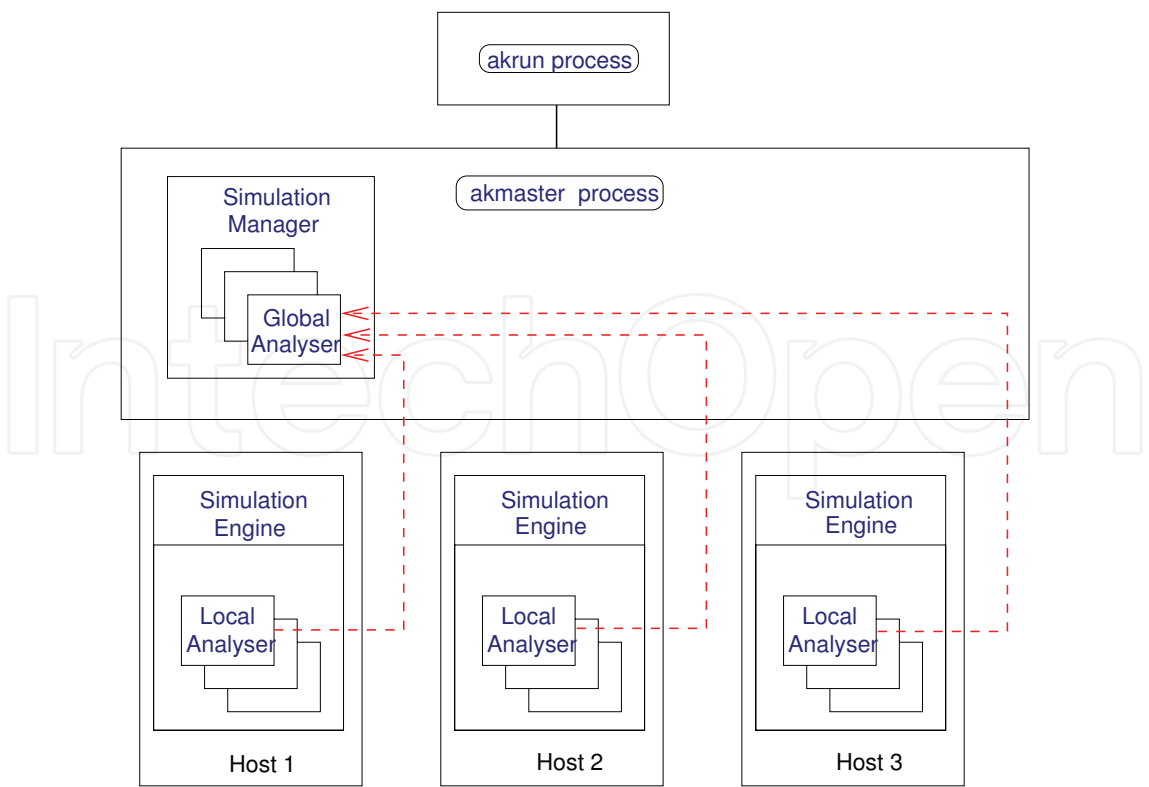


Fig. 2. Schematic diagram of Akaroa.

in parallel. A detailed description of this integration can be found in (*The ns-2akaroa-2 Project*, 2001).

4. Case study: header compression

4.1 Problem definition

One of the major challenges for wireless communication is the capacity of wireless channels, which is especially limited when a small delay bound is imposed, for example, for voice service. VoIP signaling packets are typically large, which in turn could cause a long signaling and media transport delay when transmitted over wireless networks (Yang & Wang, 2009). Moreover, VoIP performance in multi-hop wireless networks degrades with the increasing number of hops (Dragor et al., 2006).

VoIP packets are divided into two parts, headers and payload, that travel on RTP protocol over UDP. The headers are control information added by the underlying protocols, while the payload is the actual content carried out by the packet, that is, the voice encoded by some codec. As Table 1 shows, most of the commonly used codecs generates packets whose payload is smaller than IP/UDP/RTP headers (40 bytes).

In order to use the wireless channel capacity efficiently and make VoIP services economically feasible, it is necessary to apply compression techniques to reduce the overheads in the VoIP bearer and signaling packets. The extra bandwidth spared from control information traffic can be used to carry more calls in the same wireless channel or to allow the use of better quality codec to encode the voice flow.

Header compression in WMNs can be implemented in the mesh routers. Every packet received by a router from a mesh client should be compressed before being forwarded to the mesh backbone, and each packet forwarded to a mesh client should be decompressed before being forwarded out of the backbone. This guarantees that only packets with compressed headers would be transported among mesh backbone routers.

Header compression is implemented by eliminating redundant header information among packets of the same flow. The eliminated information is stored into data structures on the compressor and the decompressor, named context. When compressor and decompressor are under synchronization, it means that both compressor context and decompressor context are updated with the header information of the last sent/received packet of the flow. Figure 3 shows the scheme of header compression.

Codec	Bit rate (kbps)	Packet duration (ms)	Payload size (bytes)
G.711	64.0	20	160
G.726	32.0	20	80
G.728	16.0	20	40
G.729a	8.0	20	20
G.723.1 (MP-MLQ)	6.3	30	24
G.723.1 (ACELP)	5.3	30	20
GSM-FR	13.2	20	33
iLBC	13.33	30	50
iLBC	15.2	20	38

Table 1. Payload size generated by the most used codecs.

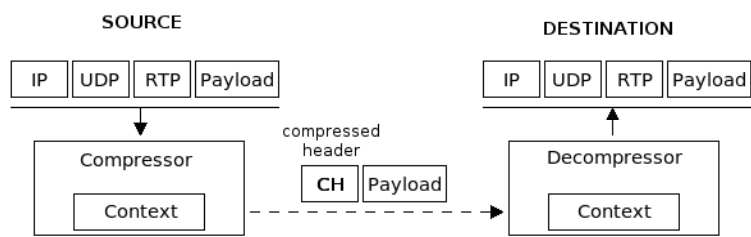


Fig. 3. General header compression scheme.

When a single packet is lost, the compressor context will be updated but the decompressor context will not. This may lead the decompressor to perform an erroneous decompression, causing the loss of synchronization between the edges and lead to the discard of all following packets at the decompressor until synchronization is restored. This problem may be crucial to the quality of communication on highly congested environments.

WMNs offer a high error rate in the channel due to the characteristics of the transmission media. Since only a device can transmit at a time, when more than one element transmits at the same time a collision occurs, as in the problem of the hidden node, which can result in loss of information in both transmitters. Moreover, many other things can interfere with communication, as obstacles in the environment, and receiving the same information through different paths in the propagation medium (multi-path fading). With these characteristics, the loss propagation problem may worsen, and the mechanisms of failure recovery by the algorithms may not be sufficient, especially in the case of bursty loss. Furthermore, the bandwidth in wireless networks is limited, making the allowed number of simultaneous users also limited. The optimal use of available bandwidth can maximize the number of users on the network.

4.2 Robust header compression – RoHC

The Compressed RTP (CRTP) was the first proposed header compression algorithm for VoIP, defined in the Request for Comments (RFC) 2508 (Casner & Jacobson, 1999). It was originally developed for low-speed serial links, where real-time voice and video traffic is potentially problematic. The algorithm compresses IP/UDP/RTP headers, reducing their

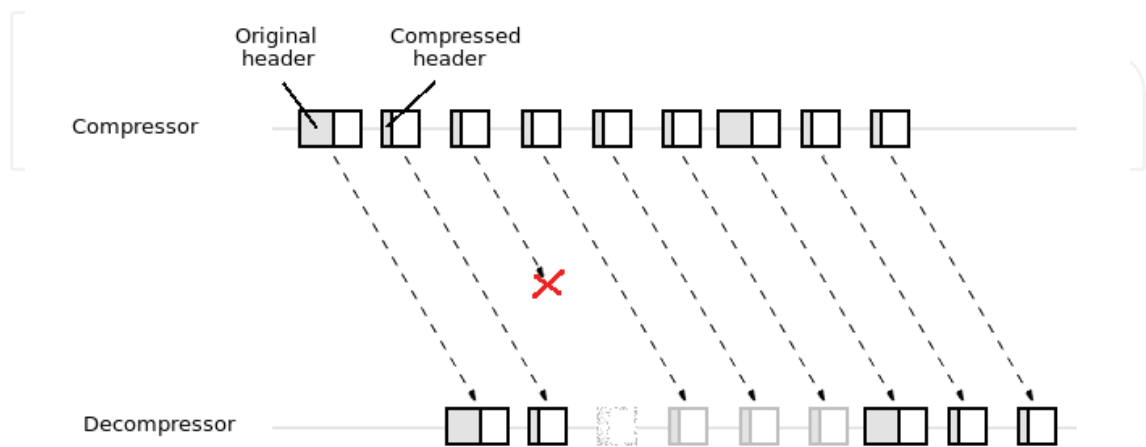


Fig. 4. Loss propagation problem.



size to approximately 2 bytes when the UDP checksum header is not present, and 4 bytes otherwise.

CRTP was designed based on the unique header compression algorithm available until that date, the Compressed TCP (CTCP) Jacobson (1990), which defines a compression algorithm for IP and TCP headers in low-speed links. The main feature of CRTP is the simplicity of its mechanism.

The operation of CRTP defines sending a first message with all the original headers information (FULL\_HEADER), used to establish the context in the compressor and decompressor. Then, the headers of following packets are compressed and sent, carrying only the delta information of dynamic headers. FULL\_HEADER packets are also periodically sent to the decompressor, in order to maintain synchronization between the contexts, or when requested by the decompressor through a feedback channel, if the decompressor detects that there was a context synchronization loss.

CRTP does not present a good performance over wireless networks, since it was originally developed for reliable connections (Koren et al., 2003), and characteristic of wireless networks present high packet loss rates. This is because the CRTP does not offer any mechanism to recover the system from a synchronization loss, presenting the loss propagation problem. The fact that wireless networks do not necessarily offers a feedback channel available to request for context recovery also influences the poor performance of CRTP.

The Robust Header Compression (RoHC) algorithm (Bormann et al., 2001) and (Jonsson et al., 2007) was developed by the Internet Engineering Task Force (IETF) to offer a more robust mechanism in comparison to the CRTP. RoHC offers three operating modes: unidirectional mode (U-mode), bidirectional optimistic mode (O-mode) and bidirectional reliable mode (R-mode). Bidirectional modes make use of a feedback channel, as well as the CRTP, but the U-mode defines communication from the compressor to the decompressor only. This introduces the possibility of using the algorithm over links with no feedback channel or where it is not desirable to be used.

The U-mode works with periodic context updates through messages with full headers sent to the decompressor. The B-mode and R-mode work with request for context updates made by the decompressor, if a loss of synchronization is detected. The work presented in (Fukumoto & Yamada, 2007) showed that the U-mode is most advantageous for wireless asymmetrical links, because the context update does not depend on the request from the decompressor through a channel that may not be available (by the fact that it is asymmetric link).

The RoHC algorithm uses a method of encoding for the values of dynamic headers that are transmitted in compressed headers, called Window-Least Significant Bits (W-LSB). This encoding method is used for headers that present small changes. It encodes and sends only the least significant bits, which the decompressor uses to calculate the original value of the header together with stored reference values (last values successfully decompressed). This mechanism, by using a window of reference values, provides a certain tolerance to packet loss, but if there is a burst loss that exceeds the window width, the synchronization loss is unavoidable.

To check whether there is a context synchronization loss, the RoHC implements a check on the headers, called Cyclic Redundancy Check (CRC). Each compressed header has a header field that carries a CRC value calculated over the original headers before the compression process. After receiving the packet, the decompressor retrieves the headers values with the information from the compressed header and from its context, and executes again the calculation of the CRC. If the value equals the value of the CRC header field, then the compression is considered

successful, otherwise it is found a synchronization loss.

The RoHC offers a high compression degree, and high robustness, but its implementation is quite complex compared to other algorithms. Furthermore, RoHC has been implemented for cellular networks, which typically have one single wireless link, and it considers that the network delivers packets in order.

#### 4.3 Static compression + aggregation

A header compression algorithm that does not need synchronization of contexts could eliminate any possibility of discarding packets at the decompressor due to packet loss, and eliminate all necessary process for updating and context re-synchronization. However, the cost to implement such an algorithm may be reflected in the compression gain, which may be lower with respect to algorithms that require synchronization.

If it is not possible to maintain the synchronization, the decompressor cannot decompress the headers of received packets. Whereas usually the decompressor is based on the information of previously received packets of the same stream to update its context, the loss of a single packet can result in the context synchronization loss, and then the decompressor may not decompress the following packets successfully, even if they arrive on time and without errors at the decompressor, and it is obliged to discard them. In this case we say that the loss was propagated as the loss of a single packet leads to the decompressor to discard all the following packets (Figure 4).

To alleviate the loss propagation problem, some algorithms use context update messages. Those messages are sent periodically, containing all the complete information of the headers. When the decompressor receives an update message, it replaces the entire contents of its current context for the content of the update message. If it is unsynchronized, it will use the information received to update its reference values, and thus restore the synchronization. One way to solve the problem of discarding packets at the decompressor due to context desynchronization was proposed in (Nascimento, 2009), by completely eliminating the need of keeping synchronization between compressor and decompressor. The loss propagation problem can be eliminated through the implementation of a compression algorithm whose contexts store only the static headers, and not the dynamic ones. If the contexts store static information only, there is no need for synchronization. This type of compression is called static compression.

The static compression has the advantage of no need of updating the context of compressor and decompressor. It only stores the static information, i.e., those that do not change during a session. This means that no packet loss will cause following packets to be discarded at the decompressor, thus eliminating the loss propagation problem. Another advantage presented by the static compression is the decrease in the amount of information to be stored in points where compression and decompression occur, as the context stores only the static information. However, the cost of maintaining contexts without the need for synchronization is reflected in the compression gain, since the dynamic information is sent in the channel and is not stored in context, as in conventional algorithms (Westphal & Koodli, 2005). This causes the compressed header size increase in comparison to conventional compression algorithms headers size, reducing the compression gain achieved.

The static compression can reduce the headers size to up to 35% of its original size. Some conventional algorithms, which require synchronization, can reduce the headers size to less than 10%. Experiments with static compression in this work showed that even though this algorithm does not present the loss propagation problem, its compression gain is not large

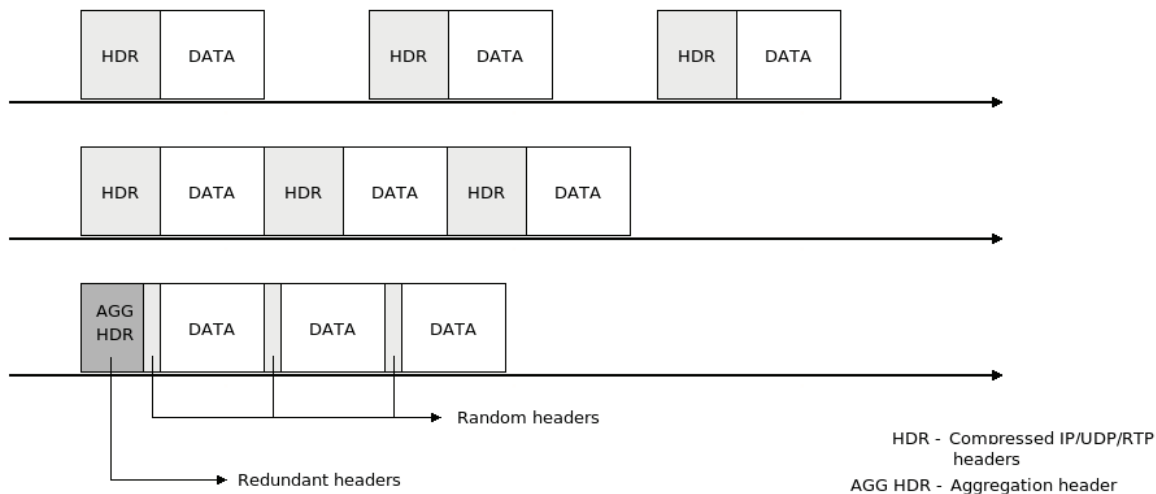


Fig. 5. Cooperative solution: compression + aggregation.

enough to offer significant gains in comparison to more robust algorithms. Therefore, it is suggested the use of technical aids to increase the compression gain achieved while using the static compression mechanism.

The static header compression use headers whose values do not change between packets of the same voice stream. However, some dynamic headers most of the time of a session also presents some redundancy between consecutive packets, because they follow a pre-established behavior pattern. One way to provide greater compression gain for the static header compression can take advantage of that redundancy often present. To use the dynamic information redundancy without returning to the problem of contexts synchronization and loss propagation, after the static compression process we can use a simple aggregation packet mechanism. The packet aggregation is a technique also used to optimize the bandwidth usage in wireless networks. Its main goal is, through the aggregation of several packets, to reduce the overhead of time imposed by the 802.11 link layer wireless networks MAC, reduce the number of packet loss in contention for the link layer, and decrease the number of retransmissions (Kim et al., 2006). In addition, aggregation also helps to save bandwidth consumption for control information traffic by decreasing the amount of MAC headers sent to the network.

An effective cooperation between the packet aggregation and packet header compression techniques requires that only packets of the same flow can be aggregated. The packet aggregation introduces a delay of the queuing process, since the compressor needs to expect the arriving of  $k$  packets to form an aggregation packet, where  $k$  is called aggregation degree. This additional delay reflects on the quality of the call, and that means that this type of mechanism is not the best option in environments with few wireless hops, or low traffic load. It is therefore important to use a low aggregation degree, since this value is directly proportional to the delay to be imposed on the traffic.

After the aggregation, the dynamic redundant information among the packets headers of the aggregated packets are taken from the compressed headers and kept into a single external header called aggregation header (Figure 5). By redundant information we mean that ones assuming sequential values or the same value among the aggregated packets. The aggregation header contains the IP/UDP/RTP headers which value is equal for all aggregated packets. So when the aggregation packet reaches the destination, the

decompressor will be able to rebuild the compressed header of each aggregated packet, from the aggregation header, and thus may continue with the process of deaggregation and subsequent static decompression. The experiments conducted in this study showed that the mechanism of compression and aggregation can increase the compression gain from about 60% (static compression only) to more than 80%.

#### 4.4 Objective of the study

The main objective of this study is to evaluate the performance of the proposed approach based on the combination of static header compression and packet aggregation. We also aim to assess the performance of the algorithm RoHC U-mode, since it is an algorithm standardized by IETF, presenting a high compression gain, and presenting the loss propagation problem.

The objective of this chapter is to suggest a sound simulation methodology aiming to get reliable results of simulations of VoIP over WMN. To achieve this goal, we started by selecting an experimental environment based on two well-known simulation tools: ns-2 and Akaroa-2. The first one was selected due to its widely use in the scientific community, which enables the repeatability of the experiments. Moreover, ns-2 receives steadily support from active forums of developers and researchers. We used the version 2.29, which received a patch with improvements on physical and link layers modeling capabilities.

Akaroa-2 was deployed to guarantee the statistical quality of the results. We are interested in measures of the steady-state period, and Akaroa-2 is in charge of detecting the end of the transient period. Observations of that period are discarded by Akaroa-2, mitigating the bias effects that should appear in the final results otherwise. The carefully design of Akaroa-2 for detecting the end of the transient period is based on a formal method proposed in (Schruben et al., 1983), as opposed to simple heuristics. By integrating ns-2 and Akaroa-2, sources of randomness in our simulation model make use of the pseudo-random number of the latter, which we analyzed and accepted as adequate to our purposes.

#### 4.5 Experimental design

For this study we opted for the use of end-to-end header compression, for no extra cost in the intermediate nodes between a source-destination pair. To make the use of end-to-end header compression over a WMN, it is necessary that the routers of the network are able to route packets with compressed headers. Since the header compression is applied also to the IP header, that means that the routers must implement the packets routing without extracting information from IP headers.

We decided to use routing labels, implemented with the Multi-protocol Label Switching (MPLS) (Rosen et al., 2001). The MPLS is known to perform routing between the network and link layers, thus performing routing on layer 2.5 (Figure 6). The MPLS works primarily with the addition of a label in the packets (and it is indifferent to the type of data transported, so it can be IP traffic or any other) in the first router of the backbone (edge router) and then the whole route through the backbone will be made by using labels, which are removed when the packets leave the backbone.

We used the implementation of MPLS for NS-2.26 available in (Petersson, 2004), it is called MPLS Network Simulator (MNS) version 2.0. It required a small adjustment on the module for use in version 2.29, and the structure of the wireless node of NS-2, because the original module only applies to wired networks.

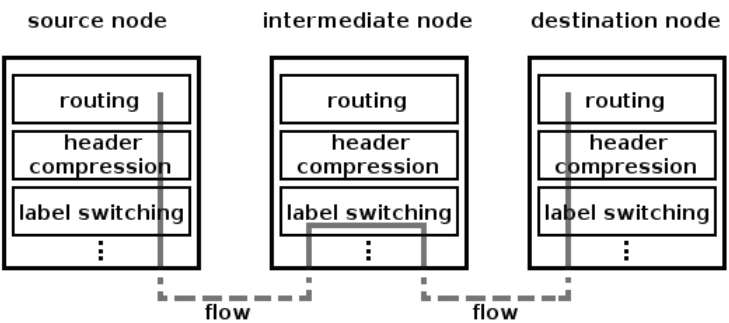


Fig. 6. Label routing performed by MPLS.

4.5.1 Factors

We compared the proposed scheme based on the combination of static header compression and packet aggregation (SHC+AG) against ROHC, and the static header compression approach (SHC). Decisive values for state transition on compressor and decompressor state machines, like number of sent packets before changing to a higher state, or number of decompress failures before changing to a lower state are not discussed in the RoHC Request for Comments. In our experiments, those values were established in according to (Seeling et al., 2006; Fukumoto & Yamada, 2007).

4.5.2 Performance measures

- Packet loss** A factor that influences the quality of real-time applications is the packet loss in the network. VoIP applications offer some tolerance to packet loss, since small losses are imperceptible to the human ear. However, this tolerance is very limited, and high rates of packet loss could impose a negative impact on the speech quality and harm the understanding of the interlocutors.
- Network delay** It is a primary factor of influence on the speech quality. The time limit for the human ear does not perceive delay on the speech reproduction is 150 ms. Therefore, if the network imposes very large delays, the impact of this factor in the quality of the call will be noticeable.
- MOS** The MOS is intended to quantitatively describe the speech quality, taking into account several factors, including packet loss, delay, codec, compression, etc. Therefore, the MOS, presented in our work together with the metric of loss and delay, will give an idea of how those metrics affect the quality of the call as a whole. It also make it possible to check if the average quality of calls can be considered acceptable.
- Compression gain** This measure indicates how effective is the compression mechanism with respect to its ability on decreasing the headers size. The higher the algorithm compression gain, the greater is its ability on compressing the headers.
- Bandwidth efficiency** This measure indicates how much of bandwidth was used for payload transmission, thus quantifying the contribution of each header compression algorithm to a more optimal usage of available bandwidth. It is obtained through the ratio between the total number of transmitted bytes of payload and the total bytes effectively used for the transmission, including payload and headers.



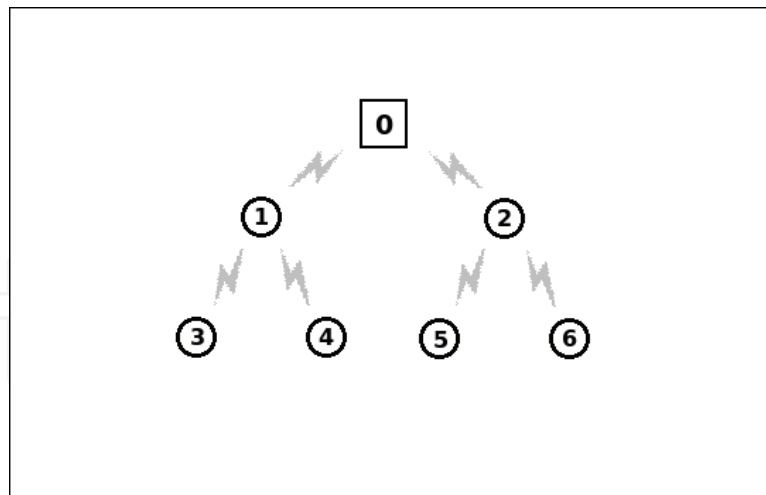


Fig. 7. Tree scenario used in the WMN simulation.

#### 4.5.3 Scenario

In our experiments we simulated the IEEE 802.11b channel (IEEE, 2004), and we used the log-normal shadowing propagation model, with the changes suggested in (Schmidt-Eisenlohr et al., 2006), and the wireless channel was set based on (Xiuchao, 2004), customized according to measurements. The shadowing model was set to use pathloss exponent of 3.5 and standard deviation of 4.0 (Rappaport, 2001).

The selected scenario represents a mesh backbone with 7 routers positioned on tree form (Figure 7). In this scenario, the main idea is to evaluate the algorithms on a network whose routers need to handle traffic from different sources. On mesh networks such behavior is common. The VoIP calls were generated from the leaf nodes (nodes 3, 4, 5, and 6) destined to the node 0. This traffic behavior is usual in many mesh networks which have a gateway, a device that gives access to external networks or to the Internet.

#### 4.5.4 Traffic model

Bidirectional VoIP calls were modeled as CBR ON/OFF traffic with 60 seconds of duration, configured to represent a voice stream coded by G.729a codec with 20ms of frame duration, 8 kbps of bit rate and with static dejitter buffer of 50ms. The codec G.729a was used, because it offers good quality in low transmission rate conditions.

#### 4.5.5 Statistical plan

The mean value of the performance measures and the corresponding confidence interval were obtained by applying the sequential version of the Spectral Analysis method of estimation implemented by Akaroa-2. (Pawlikowski et al., 1998) concluded that this method of analysis under MRIP is very accurate. A method is said to be accurate when final confidence interval is quite close to the theoretical confidence interval of a simulation model whose analytical solution is known in advance.

Given the desired confidence level, Akaroa-2 collects a number of stead-state observations (samples) after deleting observations of the transient period. At predefined checkpoints determined by the Spectral Analysis method, Akaroa-2 calculates the estimative of the performance measure of interest, computes the confidence interval and then check the relative precision. If the relative precision is less than the maximum relative precision set by the experimenter, the simulation is finished, otherwise the simulation keeps generating samples

Algorithm	Compression gain
Robust Header Compression (RoHC)	0.8645
Static Header Compression (SHC)	0.6384
Static Header Compression + Aggregation (SHC+AG)	0.8274

Table 2. Compression gain of the header compression algorithms used in the simulation.

until the next checkpoint. For the experiments in this study, we set a confidence level of 95%, and a maximum relative precision of 5%. The run length control is automatically done by Akaroa-2.

4.6 Results analysis

We are going to depict only the main results, but the interested readers can access <http://grcm.dcc.ufam.edu.br> to get more details and, of course, the source code.

Table 2 shows the values obtained for the compression gain of the evaluated algorithms. The high compression gain presented by Robust Header Compression (RoHC) algorithm is due to the fact that header compression eliminates the static and dynamic information, leaving only the context identification information and the dynamic information when there is a change in its values.

The RoHC algorithm is able to decrease the headers size up to 2 bytes, which could provide an even greater compression gain. However, since it eliminates the dynamic information from the headers, the RoHC U-mode algorithm periodically needs to send context update messages with the objective of recovering the decompressor from a possible loss of synchronization. Those update messages have a larger size than the compressed headers, reaching almost the size of the original headers.

The frequency on which update messages are sent is a trade-off for header compression algorithms that need to update the context. The shorter this frequency is, the lower is the possibility of the decompressor context being outdated, however, the lower the compression gain. Therefore, the act of sending update messages and the frequency on which they are sent directly influence the RoHC compression gain. In our experiments, we sent messages to update the headers on every 10 packets of compressed headers according to the work presented in Seeling et al. (2006).

The static compression algorithm showed the smallest compression gain. Static compression eliminates from IP/UDP/RTP headers only the information classified as static and inferred, maintaining the dynamic information in the headers. Therefore, as expected, the static compression algorithm does not offer a compression gain so high as the RoHC algorithm, that also compresses the dynamic headers. The impact of this difference in compression gain on voice traffic behavior will be evaluated with the analysis of packet loss, delay and MOS.

The static compression and packet aggregation approach showed a compression gain almost as high as the RoHC algorithm. It means that the aggregation process has fulfilled the task of increasing the compression gain of the static compression. Although the compression gain did not exceed that obtained by the RoHC, the value displayed is approaching and that means that the static header compression and packet aggregation approach has generated headers of size almost as small as the headers compressed by RoHC.

The static compression showed a compression gain of 0.6384. The mechanism of packet aggregation offered an extra compression gain due to elimination of redundant dynamic header information of the aggregated packets. In this case, we can say that the compression gain of this approach is also influenced by the aggregation degree, which in our experiments

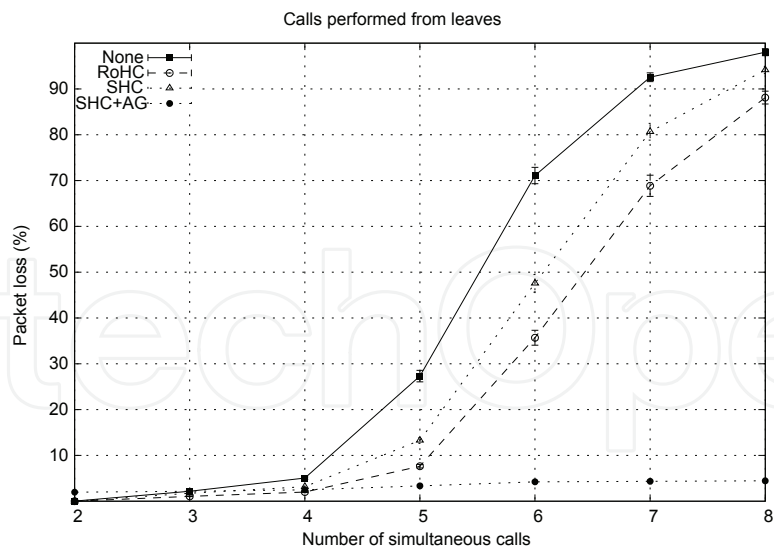


Fig. 8. Packet loss of calls performed over the tree scenario, using different header compression settings.

was two packets per aggregation packet. The aggregation degree poses a trade-off for overall speech quality, because the greater it is, the greater is the extra compression gain, but the packetization delay will be greater.

Figure 8 shows the values of packet loss obtained in the tree scenario. The number of simultaneous calls shown in the graph is the amount of simultaneous calls for each source-destination pair. Experiments were carried out with the number of simultaneous calls ranging from 2 to 8. For five or more simultaneous calls, the majority of the settings showed high packet loss rates.

The calls without header compression (*none*) showed the highest packet loss rate values. The packet loss for the SHC algorithm was higher than for the other compression algorithms.

The SHC+AG approach showed lower packet loss rate values if compared to the other algorithms. Although the aggregation increases the size of the packets, which could also increase the loss, this procedure also reduces the amount of packets sent to the network in a such a way proportional to the aggregation degree used. The packet aggregation on every two packets, as used in our experiments, resulted in the creation of slightly larger packets but not large enough to negatively impact the packet loss.

Then, aggregation reduces the amount of packets sent to the network, reducing the effort of the link layer. In addition, it provides a decrease in the amount of bytes sent to the network, key feature of the header compression. Therefore, besides the high compression gain offered by SHC+AG approach, the positive impact on the packet loss was also due to aggregation itself, which was primarily responsible for maintaining the packet loss rate below than those provided by other algorithms.

In this experiment, VoIP calls were performed to node 0, from all network nodes, and from the leaf nodes, at different times. Figure 9 shows the MOS values calculated on those calls.

The RoHC and SHC algorithms showed higher values of MOS, but showed no significant difference between them, which can be explained by the no significant difference between their packet loss rate. The SHC+AG approach showed the lowest MOS values for 2, 3, and 4 simultaneous calls, and the highest values with the increase of the simultaneous calls. Again, the MOS showed a more stable behavior with the increase on the number of calls, compared

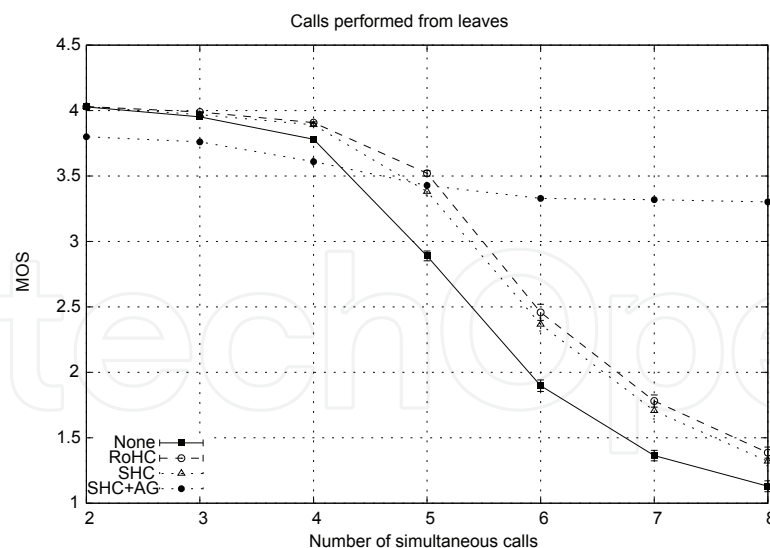


Fig. 9. MOS of calls performed over the tree scenario, using different header compression settings.

to other algorithms. This is justified by the behavior also more stable presented by metrics of delay and packet loss.

## 5. Conclusion

In this chapter we have considered a set of necessary conditions that should be fulfilled to give credibility to performance evaluation studies of VoIP transmission over WMN based on stochastic simulation. Since we have followed a sound methodology formed by the carefully choices in every stage of the simulation methodology, we can be sure that our results are reliable, no matter which results we have obtained. Certainly, the proposed compression scheme deserves additional fine tuning, but we are sure that future versions of it can be compared in an unbiased manner.

## 6. References

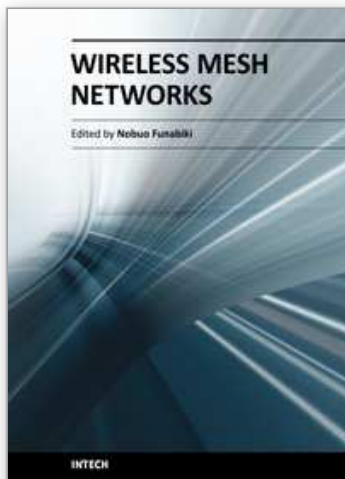
- Andel, T. R. & Yasinac, A. (2006). On the Credibility of Manet Simulations, *Computer* 39(7): 48–54.
- Balci, O. & Nance, R. (1985). Formulated Problem Verification as an Explicit Requirement of Model Credibility, *Simulation* 45(2): 76–86.
- Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L.-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T. & Zheng, H. (2001). Robust Header Compression: Framework and four profiles, Request for Comments 3095.
- Carvalho, L. S. G. (2004). *An e-model implementation for objective speech quality evaluation of voip communication networks*, Master's thesis, Federal University of Amazonas.
- Carvalho, L. S. G., Mota, E. S., Aguiar, R., Lima, A. F., de Souza, J. N. & Barreto, A. (2005). An e-model implementation for speech quality evaluation in voip systems, *IEEE Symposium on Computers and Communications*, Cartagena, Spain, pp. 933–938.
- Casner, S. & Jacobson, V. (1999). Compressing IP/UDP/RTP Headers for Low-Speed Serial Links, Request for Comments 2508.
- Clark, A. D. (2003). Modeling the Effects of Burst Packet Loss and Recency on Subjective Voice

- Quality, *IP Telephony Workshop*, Columbia University.
- C.L.Barrett, Marathe, A., Marathe, M. & Drozda, M. (2002). Characterizing the interaction between routing and mac protocols in ad-hoc networks, *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, ACM, New York, NY, USA, pp. 92–103.
- Conway, R. (1963). Some tactical problems in digital simulation, *Management Science* 10, 1: 47–61.
- Dragor, N., Samrat, G., Kyungtae, K. & Rauf, I. (2006). Performance of voip in a 802.11 wireless mesh network, *Proceedings of the IEEE INFOCOM*, Barcelona, Spain, pp. 49–52.
- Ewing, G. C., Pawlikowski, K. & McNickle, D. (1999). Akaroa2: Exploiting network computing by distributing stochastic simulation, *Proceedings of the 13th European Simulation Multi-Conference*, Warsaw, Poland, pp. 175–181.
- Fukumoto, N. & Yamada, H. (2007). Performance Enhancement of Header Compression over Asymmetric Wireless Links Based on the Objective Speech Quality Measurement, *SAINT '07: Proceedings of the 2007 International Symposium on Applications and the Internet*, IEEE Computer Society, Washington, DC, USA, p. 16.
- Glynn, P. & Heidelberger, P. (1992). Experiments with initial transient deletion for parallel replicated steady-state simulations, *Management Science* 38(3): 400–418.
- Grancharov, V. & Kleijn, W. B. (2008). *Speech Quality Assessment*, Springer, chapter 5, pp. 83–99.
- Heidelberger, P. & Welch, P. D. (1981). A spectral method for confidence interval generation and run length control in simulations, *Communications of the ACM* 24(4): 233–245.
- Hoene, C., Karl, H. & Wolisz, A. (2006). A perceptual quality model intended for adaptive VoIP applications, *Int. J. Commun. Syst.* 19(3): 299–316.
- IEEE (2004). IEEE 802.11TM Wireless Local Area Networks.  
URL: <http://grouper.ieee.org/groups/802/11/>
- Ivanov, S., Herms, A. & Lukas, G. (2007). Experimental validation of the ns-2 wireless model using simulation, emulation, and real network, *In 4th Workshop on Mobile Ad-Hoc Networks (WMAN07)*, pp. 433–444.
- Jacobson, V. (1990). Compressing TCP/IP Headers for Low-Speed Serial Links, Request for Comments 1144.
- Jonsson, L.-E., Sandlund, K., Pelletier, G. & Kremer, P. (2007). Robust Header Compression: Corrections and Clarifications to RFC 3095, Request for Comments 4815.
- Kim, K., Ganguly, S., Izmailov, R. & Hong, S. (2006). On Packet Aggregation Mechanisms for Improving VoIP Quality in Mesh Networks, *Proceedings of the Vehicular Technology Conference, VTC'06*, IEEE, pp. 891–895.
- Koren, T., Casner, S., Geevarghese, J., Thompson, B. & Ruddy, P. (2003). Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering, Request for Comments 3545.
- Kurkowski, S., Camp, T. & Colagrosso, M. (2005). Manet simulation studies: the incredibles, *SIGMOBILE Mob. Comput. Commun. Rev.* 9(4): 50–61.
- Law, A. M. & McComas, M. G. (1991). Secrets of successful simulation studies, *Proceedings of the 23rd conference on Winter simulation*, IEEE Computer Society, Washington, DC, USA, pp. 21–27.
- L'Ecuyer, P. (1999). Good parameters and implementations for combined multiple recursive random number generators, *Operations Research* 47(1): 159–164.
- L'Ecuyer, P. (2001). Software for uniform random number generation: Distinguishing the good and the bad, *Proceedings of the 33rd Conference on Winter Simulation*, IEEE Computer



- Society, Virginia, USA, pp. 95–105.
- Lili, Z., Huibin, W., Lizhong, X., Zhuoming, X. & Chenming, L. (2009). Fault tolerance and transmission delay in wireless mesh networks, *NSWCTC '09: Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, IEEE Computer Society, Washington, DC, USA, pp. 193–196.
- Matsumoto, M. & Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator, *ACM Trans. Model. Comput. Simul.* 8(1): 3–30.
- McCanne, S. & Floyd, S. (2000). The Network Simulator.  
URL: <http://www.isi.edu/nsnam/ns/>
- Mota, E., Fitzek, F., Pawlikowski, K. & Wolisz, A. (2000). Towards Credible and Efficient Network Simulation Experiments, *Proceedings of the High Performance Computing Symposium, HCPS'2000*, Washington, DC, USA, pp. 116–121.
- Mota, E. S. (2002). *Performance of Sequential Batching-based Methods of Output Data Analysis in Distributed Steady-state Stochastic Simulation*, PhD thesis, Technical University of Berlin.
- Myakotnykh, E. S. & Thompson, R. A. (2009). Adaptive Rate Voice over IP Quality Management Algorithm, *International Journal on Advances in Telecommunications* 2(2): 98–110.
- Nascimento, A. G. (2009). *Header compression to achieve speech quality in voip over wireless mesh*, Master's thesis, Federal University of Amazonas.
- Naylor, T. H., Finger, J. M., McKenney, J. L., Schrank, W. E. & Holt, C. C. (1967). Management Science, *Management Science* 14(2): B92–B106. Last access at 23 Feb. 2009.  
URL: <http://www.jstor.org/stable/2628207>
- Pawlikowski, K. (1990). Steady-state simulation of queueing processes: survey of problems and solutions, *ACM Comput. Surv.* 22(2): 123–170.
- Pawlikowski, K., Ewing, G. C. & McNickle, D. (1998). Coverage of Confidence Intervals in Sequential Steady-State Simulation, *Journal of Simulation Practise and Theory* 6(3): 255–267.
- Pawlikowski, K., Jeong, H.-D. & Lee, J.-S. R. (2002). On Credibility of Simulation Studies of Telecommunication Networks, *IEEE Communications Magazine* 40(1): 132–139.
- Petersson, M. (2004). MPLS Network Simulator verso 2.0 para NS-2 2.26.  
URL: <http://heim.ifi.uio.no/johanmp/ns-2/mns-for-2.26.tar.gz>
- Queiroz, S. (2009). *Evaluation of incremental routing in 802.11 wireless mesh networks*, Master's thesis, Federal University of Amazonas.
- Raake, A. (2006). *Speech Quality of VoIP: Assessment and Prediction*, John Wiley & Sons.
- Ramalingam, G. & Reps, T. (1996). An incremental algorithm for a generalization of the shortest-path problem, *J. Algorithms* 21(2): 267–305.
- Rappaport, T. S. (2001). *Wireless Communication Principles and Practice*, Prentice Hall PTR.
- Rosen, E., Viswanathan, A. & Callon, R. (2001). Multiprotocol Label Switching Architecture, Request for Comments 3031.
- Sargent, R. G. (2008). Verification and validation of simulation models, *Proceedings of the 40th Conference on Winter Simulation*, pp. 157–169.
- Schmidt-Eisenlohr, F., Letamendia-Murua, J., Torrent-Moreno, M. & Hartenstein, H. (2006). Bug Fixes on the IEEE 802.11 DCF module of the Network Simulator Ns-2.28, Technical Report.  
URL: <http://www.telematica.polito.it/fiore/index.html>

- Schruben, L., Singh, H. & Tierney, L. (1983). Optimal tests for initialization bias in simulation output, *Operations Research* 31, 6: 1167–1178.
- Seeling, P., Reisslein, M., Madsen, T. K. & Fitzek, F. H. (2006). Performance Analysis of Header Compression Schemes in Heterogeneous Wireless Multi—Hop Networks, *Wirel. Pers. Commun.* 38(2): 203–232.
- Stevanović, R., Topić, G., Skala, K., Stipčević, M. & MedvedRogina, B. (2008). Quantum random bit generator service for monte carlo and other stochastic simulations, pp. 508–515.
- The ns-2akaroa-2 Project* (2001). Last access at 24 Aug. 2010.  
URL: [http://www.tkn.tuberlin.de/research/ns-2\\_akaroa-2/ns.html](http://www.tkn.tuberlin.de/research/ns-2_akaroa-2/ns.html)
- Totaro, M. W. & Perkins, D. D. (2005). Using statistical design of experiments for analyzing mobile ad hoc networks, *MSWiM '05: Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, ACM, New York, NY, USA, pp. 159–168.
- Union, I. T. (1996). Methods for subjective determination of transmission quality, *Recommendation P.800*, Telecommunication Standardization Sector of ITU, Geneva, Switzerland.
- Union, I. T. (1998). The E-model, a computational model for use in transmission planning, *Recommendation G.107*, Telecommunication Standardization Sector of ITU, Geneva, Switzerland.
- Weigle, M. C. (2006). Improving confidence in network simulations, *Proceedings of the Winter Simulation Conference*, Monterey, CA, pp. 2188–2194.
- Westphal, C. & Koodli, R. (2005). Stateless IP Header Compression, *Proceedings of IEEE International Conference on Communication*, Mountain View, CA, USA, pp. 3236 – 3241.
- Xiuchao, W. (2004). Simulate 802.11b Channel within Ns-2.  
URL: [http://www.comp.nus.edu.sg/wuxiucha/research/reactive/report/80211ChannelinNS\\_2new.pdf](http://www.comp.nus.edu.sg/wuxiucha/research/reactive/report/80211ChannelinNS_2new.pdf)
- Yang, Y. & Wang, X. (2009). *Compression Techniques for VoIP Transport over Wireless Interfaces*, CRC Press, chapter 5, pp. 83–99.



## **Wireless Mesh Networks**

Edited by Nobuo Funabiki

ISBN 978-953-307-519-8

Hard cover, 308 pages

**Publisher** InTech

**Published online** 14, January, 2011

**Published in print edition** January, 2011

The rapid advancements of low-cost small-size devices for wireless communications with their international standards and broadband backbone networks using optical fibers accelerate the deployment of wireless networks around the world. The wireless mesh network has emerged as the generalization of the conventional wireless network. However, wireless mesh network has several problems to be solved before being deployed as the fundamental network infrastructure for daily use. The book is edited to specify some problems that come from the disadvantages in wireless mesh network and give their solutions with challenges. The contents of this book consist of two parts: Part I covers the fundamental technical issues in wireless mesh network, and Part II the administrative technical issues in wireless mesh network,. This book can be useful as a reference for researchers, engineers, students and educators who have some backgrounds in computer networks, and who have interest in wireless mesh network. It is a collective work of excellent contributions by experts in wireless mesh network.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Edjair Mota, Edjard Mota, Leandro Carvalho, Andrea Nascimento and Christian Hoene (2011). Pursuing Credibility in Performance Evaluation of VoIP over Wireless Mesh Networks, *Wireless Mesh Networks*, Nobuo Funabiki (Ed.), ISBN: 978-953-307-519-8, InTech, Available from: <http://www.intechopen.com/books/wireless-mesh-networks/pursuing-credibility-in-performance-evaluation-of-voip-over-wireless-mesh-networks>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen