A Quality of Experience based Approach for Wireless Mesh Networks[*]

Anderson Morais, and Ana Cavalli

Télécom SudParis, France
{anderson.morais, ana.cavalli}@it-sudparis.eu

Abstract. Wireless Mesh Network (WMN) is an emerging technology that is gaining importance among traditional wireless communication scenarios. Wireless Mesh Networks are becoming a popular way of offering end-to-end services, such as Voice over IP (VoIP) and Video on Demand (VOD), in an inexpensive, practical, and fast manner. However, those applications require a minimum network performance level in order to provide an acceptable Quality of Experience (QoE) level to users. In this paper, we present a QoE based approach, which analyzes key QoS parameters like delay, jitter, and packet loss in a mesh network executing the Better Approach To Mobile Ad hoc Network (BATMAN) proactive routing protocol. The proposed mechanism evaluates the actual QoE level from the point of view of each mesh node taking into account the calculated QoS parameters, and then proposes a response in case of QoE degradation as reducing the node broadcasting rate.

Keywords: QoE, Wireless Mesh Network, QoS parameters.

1 Introduction

Wireless Mesh Network (WMN) has emerged in the wireless communication scenario as a new technology to fulfill the requirements of Next Generation Wireless Networks (NGWN), such as offering adaptive, flexible, and reconfigurable network architecture while providing cost-effective solutions to wireless Internet Service Providers (ISPs) [1]. ISPs are choosing Wireless Mesh Network (WMN) to offer Internet connectivity since it allows an easy, fast, and cheap network deployment.

WMNs are characterized by dynamic self-organization, self-configuration, and self-healing whereas the mesh nodes automatically establish an ad hoc network and maintain the connectivity among the nodes. The mesh nodes transmit traffic from others nodes to reach a destination that they could not reach by themselves. The mesh structure assures the availability of multiple paths for the node in the network. If a mesh node crashes or loses the connection, its neighboring nodes simply find another route and the network continues operating. Rather than traditional wireless networks, WMNs do not rely on dedicated wireless infrastructure, but the nodes count on each other to maintain the network entirely connected and transmit data traffic to the destination. As a result, the network is very reliable and has a good coverage because

there is often more than one route between a source node and a destination node. Wireless community networks and municipal wireless networks are good examples of real-life WMNs, which offer low-cost Internet access via Wi-Fi to large areas by using inexpensive IEEE 802.11 wireless mesh routers.

However, streaming multimedia applications including VoIP, Internet Protocol Television (IPTV), and multiplayer online games, require a minimum level of performance, for example, fixed bit rate, low latency, and low jitter, in order to provide an acceptable usability, reliability, and end-to-end quality level to users. Quality of Experience (QoE) is the user perceived service performance, or the required degree of satisfaction of the user. A typical QoE measurement method is the Mean Opinion Score (MOS) [2], which can be determined from subjective ratings by real subjects, for instance, subjective quality assessment of audio listening where a number of users judge the quality of the receive audio. Alternatively, the MOS can be predicted from objective analysis, which uses an original signal and the degraded received signal as input. The MOS provides a numerical indication of the perceived quality of received media after compression and/or transmission.

In this paper, we describe a QoE based approach, which analyzes key QoS parameters such as delay, jitter, and packet loss for a VoIP service. The aim of the proposed mechanism is to show how the current network performance level can impact on the service quality level. We assume the service is running in a mesh network, where each node executes BATMAN proactive routing protocol. The proposed mechanism calculates the objective QoS parameters for each mesh node by applying active and passive monitoring techniques. Then these parameters are used to calculate the MOS, which represents the user perceived performance, i.e., the QoE. The approach can also carry out responses in case of QoS and QoE degradation as reducing the broadcasting rate of the node to improve the network performance and bandwidth, or updating the node's routing table to redirect the data traffic to a different path and consequently avoid taking congested routes that provoke collisions.

The main contributions of the paper are: (i) a QoE based approach that analyzes performance metrics for a mesh network; (ii) a mechanism to calculate passively and actively the QoS parameters and the corresponding QoE level in a mesh network; (iii) a reaction mechanism in case of QoS/QoE deterioration for QoE adjustment.

The remainder of the paper proceeds as follows. Section 2 describes the related work. Section 3 introduces the QoE based approach composed of a monitoring module and a controlling module. Finally, Section 4 concludes the paper.

2 Related Work

Research efforts have been focused on obtaining QoE measures from objective metrics offered by QoS assessment in order to overcome the difficulties and drawbacks of performing a subjective QoE assessment. Those approaches can be classified in intrusive and non-intrusive methods. The Perceptual Evaluation of Speech Quality (PESQ) [3] is an intrusive method for automated assessment of speech quality, which predicts the subjective quality of speech codecs by comparing the source signal with the degraded received signal. PESQ was developed to model

subjective tests commonly used in telecommunications to assess the voice quality by real users, thus using true voice samples as test signals.

The E-model [4] and the ITU-T Recommendation P.563 [5] are common examples of non-intrusive techniques. The E-model tool assess the voice quality taking into account telephony impairments, e.g., low bit-rate codecs, one-way delay, loss, noise, and echo, and outputs a "Transmission Rating Factor R". The ITU-T P.563 evaluates the voice quality executing three steps: (i) preprocessing of the voice signal, (ii) extraction of distortion and speech parameters from the voice signal parts, and (iii) determination of distortion class and generation of the MOS.

Moreover, new non-intrusive approaches for VoIP and Video over IP have been proposed to objectively estimate the QoE from QoS parameters. In [6], the authors propose a framework to provide online QoE estimates for Voice and Video over IP (VVoIP) on the network paths. The QoE model is expressed as a function of the measurable network factors: bandwidth, delay, jitter, and loss. In [7], a model that characterizes the relationship between packet loss and video distortion is developed. That model is used to develop a video quality evaluation method that is independent from the video content characteristics. The work [8] extends the E-model considering the effects of packet loss and delay jitter in VoIP scenarios. A new formula is proposed to quantify these effects and incorporated into the E-model.

However, approaches that perform subjective QoE assessment based exclusively on QoS analysis using real-time traffic in real network environments are hardly found, and even less for multimedia applications based on IPTV and VOD technologies.

3 The QoE Based Approach

The goal of the proposed mechanism is to monitor the performance level of a mesh network in order to ensure a stable and proper QoE level for each mesh node. Objective QoS parameters are measured, and then mapped to the user-perceived QoE, which is expressed through the *Mean Opinion Score* (MOS). Performance and service quality problems should be detected as quickly as possible by the approach, and a precise reaction should be provided to maintain a high QoS level and a good quality level for the services. Every relay mesh node is equipped with monitoring capacities, which evaluate the current network conditions, and provides a response accordingly.

The approach provides QoS support at MAC Layer by using BATMAN Layer 2 routing functionality. BATMAN protocol operates entirely on ISO/OSI Layer 2, so not only the routing information is transported using raw Ethernet frames, but also the data traffic is handled by BATMAN. BATMAN encapsulates and forwards all data traffic until it reaches the destination, thus emulating a virtual network switch of all mesh nodes. Hence, all nodes seem to be link local and not aware of the network's topology. The BATMAN algorithm [9] is detailed as follows.

Each node (also referenced as Originator) periodically broadcasts hello messages, known as Originator Messages (OGMs), to tell its neighbors about its existence. An OGM owns at least an Originator address, a Source address, a Time To Live (TTL), and a unique Sequence Number value. As a neighbor receives an OGM, it modifies the Source address to its own address and rebroadcast this OGM in accordance to

BATMAN forwarding rules to tell its neighboring nodes about the existence of the node that originated the OGM, and so on and so forth. Hence, the mesh network is flooded with OGMs until every node has received it at least once, or until they got lost because of communication links packet loss, or until their TTL value has expired. The Sequence Number value of the OGM is utilized to verify how fresh the message is, i.e., to discern between new OGMs and duplicated OGMs in order to guarantee that each OGM is only counted once. The amount of OGMs, i.e., the total number of Sequence Numbers, received from an Originator via each neighboring node is utilized to calculate the route quality, i.e., the Transmission Quality (TQ). Thus, BATMAN chooses as next-hop to the Originator the neighboring node from which it has received the highest amount of OGMs from that Originator within a sliding window.

Basically, the proposed approach consists of two modules, the *Traffic Monitoring Module* that analyzes the network performance status from the node, and the *Routing Control Module* that offers different possibilities to influence on the traffic that goes through the node in order to provide appropriate QoS. These two components must communicate with each other in the node for QoS/QoE information exchange. Furthermore, all the *Routing Control Modules* throughout the network should be able to exchange QoE and performance information in a distributed way.

Figure 1 shows the general structure of the proposed approach. The main part is formed by the BATMAN Layer 2 protocol. Running BATMAN on every node enables all the nodes to connect to each other and to form a mesh cloud. The *Traffic Monitoring Module* is implemented as a hybrid probe in each node, which passively captures packets passing through to the BATMAN interface and actively injects BATMAN traffic in the network, when necessary, to collect measurements such as packet loss, and round trip time. In addition, the module is executed independently of network layer (ex.: IPv4, IPv6, and DHCP) and the transporting protocol. The *Routing Control Module* is implemented as an extension to BATMAN protocol. It can make use of BATMAN OGMs or generate specific messages to allow communication between different *Routing Control Modules*. The *Routing Control Module* and *Traffic Monitoring Module* are connected to each other via communication sockets. Sections 3.1 and 3.2 explain these two modules in more detail.
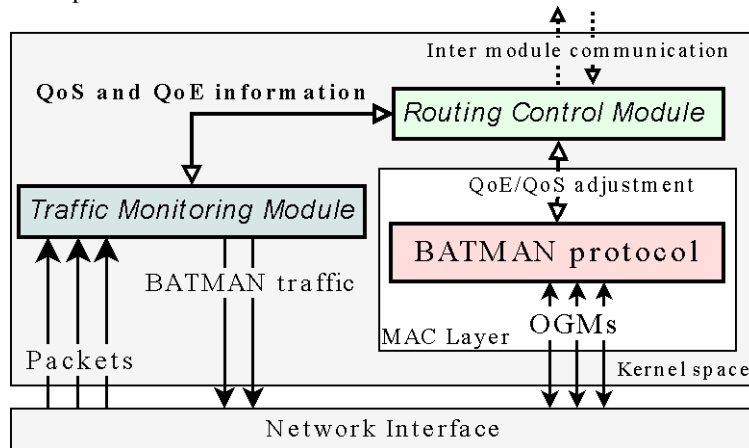

Fig. 1. Scheme for QoE/QoS control.

3.1 Traffic Monitoring Module

The *Traffic Monitoring Module* is an active and passive network-monitoring tool that is customized for a BATMAN mesh network to monitor QoS parameters from the mesh network traffic.

The *Traffic Monitoring Module* obtains some basic information about a certain packet stream flow that it intercepts and analyzes. Table 1 illustrates the explicit information provided by the *Traffic Monitoring Module* that is extracted from BATMAN packets and from the node's routing table. For instance, the node receives an OGM from its neighbor `fe:fe:00:00:02:01` (Source address), which was originated by node `fe:fe:00:00:01:01` (Originator address) with a unique Sequence Number value. According to the node's routing table its neighbor `fe:fe:00:00:02:01` is the current best ranking neighbor towards the Originator. The routing table also provides a "Potential Nexthop" towards the Originator.

Table 1. Explicit information extracted from BATMAN packets and the routing table.

BATMAN Packet			BATMAN Routing Table		
Source	Originator	Sequence Number	Best Nexthop	Potential Nexthop	Transmission Quality (TQ) to the Originator
fe:fe:00:00: 02:01	fe:fe:00:00: 01:01	125	fe:fe:00:00: 02:01	fe:fe:00:00: 03:01	245/255

Active Approach. Table 2 presents the statistical information obtained by the active monitoring mechanism. The Round Trip Time (RTT) is the length of time it takes for a packet to be sent plus the length of time it takes for an acknowledgment of that packet to be received.

Table 2. Statistical information obtained by the active monitoring mechanism.

Active monitoring				
Packet Loss (%)	Minimum RTT (ms)	Maximum RTT (ms)	Mean RTT (ms)	RTT Standard Deviation (ms)
0.0	10.912	68.917	12.640	5.330

The active monitoring approach is applied to obtain the *Packet Loss*, the *Minimum RTT* (the shortest RTT), the *Maximum RTT* (the longest RTT), the *Mean RTT* (the average of measured RTTs), and the *RTT Standard Deviation* (an indication of how regular or varied the RTTs were). For that, BATMAN Advanced Control and Management tool (*batctl*) is employed [10]. The reason to use BATMAN *batctl* tool is since BATMAN operates on Layer 2, thus all nodes participating in the virtual switch are completely transparent for all protocols above Layer 2. Therefore, the common diagnosis tools do not work as expected in a BATMAN mesh network.

The *batctl* ping diagnostic tool is a useful way to determine if a network connection is of sufficient quality, for example, to carry out a VoIP call without voice quality degradation. For that, the *batctl* ping tool injects custom ICMP echo request packets into the data flow demanding an immediate response. Then it measures the

time it takes for the response to be received. From this, the RTT is calculated. If a response is not detected, then a lost packet is recorded. Note that absolute values of RTT is of less importance since it does not matter at all to voice quality if the a RTT is 1ms or 100ms, what matters for VoIP calls are the following indicators [11]:

- If the *Mean RTT* is less than 150ms, then the latency itself will not be an issue to users. VoIP calls can still be carried on networks with RTTs as high as 500ms and above (satellite network frequently have RTTs higher than 700ms) but the delay is noticeable to callers.
- If the *Packet Loss* is less than 2%, then the voice quality is not affected. In fact, *Packet Loss* should to be close to 0%, but voice quality degradation will only become an issue as *Packet Loss* goes beyond 2%.
- If the *RTT Standard Deviation* is less than 10ms, then the voice quality is not deteriorated. The main problem with networks carrying voice is jitter [12], the variability in packet latency times. A network with constant latency has no variation, and consequently no jitter. If RTTs vary abruptly then voice quality will suffer from jitter. If *RTT Standard Deviation* goes beyond 20ms one will start to have problems with voice traffic because of jitter.
- If the difference between the *Minimum RTT* and *Maximum RTT* values is less than 20ms then jitter is not a problem. If it is above 20ms, then a more detailed analysis of the responses is required to determine if the *Maximum RTT* is an isolated anomaly or an indication of significant jitter. Frequent variation in the RTT values over a range greater than 20ms indicates jitter issues and voice quality problems.

The equations to calculate the RTT statistical information are given below.

- For a packet p_i that is generated by *batctl* tool, the RTT is defined as:

$$RTT_i = te_{p_i} - ts_{p_i} \,, \tag{1}$$

where ts_{p_i} is the start time, i.e., the time packet p_i is sent to destination, and te_{p_i} is the end time, i.e., the time packet p_i is received back on the sender.

- The *Mean RTT* is given by:

$$\overline{RTT} = \frac{1}{n} \sum_{i=1}^{n} RTT_i \,, \tag{2}$$

where n is the number of measured RTTs for n packets.

- The *RTT Standard Deviation* (or jitter) is calculated as:

$$RTTSD = \sqrt{\frac{1}{n} \sum_{i=1}^{n} RTT_i^2 - \overline{RTT}^2} \,. \tag{3}$$

- The *Packet Loss* is calculated as:

$$P_{Loss} = 1 - \frac{n}{N} \quad , \tag{4}$$

where N is the total number of packets injected by *batctl* tool.

Passive Approach. Table 3 shows the statistical information that can be obtained by passive monitoring, which simply observes the packets passing through the BATMAN node interface. The *Packet Loss*, the *Mean Inter Packet Delay*, the *Inter Packet Delay Standard Deviation*, the *Bit Rate*, and the *Packet Rate* can be obtained.

Table 3. Statistical information obtained by the passive monitoring mechanism.

Passive monitoring				
Packet Loss (%)	*Mean Inter Packet Delay* (ms)	*Inter Packet Delay Standard Deviation* (ms)	*Bit Rate* (bit/s)	*Packet Rate* (pkt/s)
0.2	20.640	5.330	120.5	130.0

For every BATMAN packet pkt_i that traverses the node the following implicit and explicit information can be obtained:
- seq_i, that is the unique sequence number of pkt_i generated by Originator O_j, for $i = \{0, \dots, n\}$ and $j = \{0, \dots, m\}$.
- t_i, the absolute arrival time of pkt_i.
- $\Delta t_i = t_i - t_{i-1}$, the relative arrival time of pkt_i compared to pkt_{i-1} expressed in seconds, for $0 < i \leq n$.
- len_i, the total length of pkt_i in bytes.

Using these definitions, the statistical information can be obtained as follows.

- The *Mean Inter Packet Delay* is defined as:

$$MIPD = \overline{\Delta t} = \frac{1}{n} \sum_{i=1}^{n} \Delta t_i \quad , \tag{5}$$

where n is the number of the packets analyzed for Originator O_j.

- The *Inter Packet Delay Standard Deviation*, i.e., the jitter, is calculated as:

$$IPDSD = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \Delta t_i^2 - \overline{\Delta t}^2} \quad . \tag{6}$$

- The *Packet Loss* is defined as:

$$PKT_{Loss} = 1 - \frac{(n+1)}{seq_n - seq_0} \quad , \tag{7}$$

where $(n + 1)$ is the number of the captured packets, seq_n is the maximum sequence

number received from Originator O_j, and seq_0 is the minimum sequence number received from Originator O_j.

- The *Bit Rate* in kbit/sec (kbps) is defined as:

$$BR = \frac{1}{t_n - t_0} \sum_{i=0}^{n} len_i \quad . \tag{8}$$

- While the *Packet Rate* in pkts/s is defined as:

$$PKT_{Rate} = \frac{(n + 1)}{t_n - t_0} \quad . \tag{9}$$

3.2 Routing Control Module

The key part of the proposed approach is the *Routing Control Module*. It is in charge of evaluating whether the actual network situation is acceptable according to the performance measurements supplied by the *Traffic Monitoring Module*, and if this is not the case, it must decide how to react to the traffic QoS problems. To perform this task, certain thresholds are required.

Threshold Mechanism. Monitoring of the services alone is not enough to provide QoS/QoE enhancement. It is also necessary a mechanism that evaluates the monitored information and responds in the case of a possible performance quality decrease. To perform this task, a *threshold* mechanism is necessary which indicates a *good*, *medium*, or *bad* service quality level.

Moreover, key parameters have to be defined in order to establish the *thresholds* and attribute correct quality levels to them. The key parameters selected to evaluate the QoS and possible QoS impairment are the previously ones introduced for the active and passive approaches:
- Active approach: the *Mean RTT – $\overline{RTT}$* (latency), the *RTT Standard Deviation – RTTSD* (jitter), and the *Packet Loss – P_{Loss}*.
- Passive approach: the *Mean Inter Packet Delay – MIPD* (latency), the *Inter Packet Delay Standard Deviation – IPDSD* (jitter), and the *Packet Loss – PKT_{Loss}*.

In this work, *thresholds* are defined to evaluate the QoS level, which are based on key performance parameters, and should be configured according to each service. Each BATMAN packet transporting data for a specific service can be configured with 12 *thresholds* values, which 6 for the active approach and 6 for the passive approach, as shown in Table 4. The *thresholds* values for the active approach take into account a VoIP service, as explained in Section 3.1. For instance, if $0 \leq \overline{RTT} < Th_{\overline{RTT}}^{m-g}$ the call service will not be affected, and the output of quality level is certainly a *good* rating. If $Th_{\overline{RTT}}^{m-g} \leq \overline{RTT} \leq Th_{\overline{RTT}}^{m-b}$ the delay will notably deteriorate the transmission quality, providing a *medium* quality level. If $\overline{RTT} > Th_{\overline{RTT}}^{m-b}$ the user experience will be practically unacceptable, then providing a *bad* quality level.

Table 4. Thresholds defined for the key QoS parameters of the active and passive approaches.

Active approach		Passive approach	
medium-good	*medium-bad*	*medium-good*	*medium-bad*
$Th_{RTT}^{m-g} = 150\text{ms}$	$Th_{RTT}^{m-b} = 500\text{ms}$	$Th_{MIPD}^{m-g} = 150\text{ms}$	$Th_{MIPD}^{m-b} = 400\text{ms}$
$Th_{RTTSD}^{m-g} = 10\text{ms}$	$Th_{RTTSD}^{m-b} = 20\text{ms}$	$Th_{IPDSD}^{m-g} = 10\text{ms}$	$Th_{IPDSD}^{m-b} = 20\text{ms}$
$Th_{P_{Loss}}^{m-g} = 0.4\%$	$Th_{P_{Loss}}^{m-b} = 2\%$	$Th_{PKT_{Loss}}^{m-g} = 0.4\%$	$Th_{PKT_{Loss}}^{m-b} = 2\%$

The *thresholds* of the passive approach are defined in accordance with ITU-T G.114 [13], which recommends that a one-way delay of 400ms should not be exceeded, although highly interactive services such as voice calls and video conferences can be affected by much lower delays. In addition, if delays are kept below 150ms, most applications, both speech and non-speech, will experience essentially transparent interactivity.

The *thresholds* could become less demanding in case of a larger number of services running in the network, which demand more bandwidth. Alternatively, they can be more claiming in an empty network, which means more bandwidth resource is available. The *threshold* adaptation process is based on the defined *Bit Rate BR* and *Packet Rate PKT_{Rate}* values. Therefore, the *thresholds* can be adapted to different network situations. Moreover, they could assume dynamic values according to each type of service and taking into consideration the network load.

The monitored information calculated by the *Traffic Monitoring Module* is available to the *Routing Control Module* at any time. More precisely, the statistical information is computed and saved internally for the last n captured packets (for the passive approach) or for the last N injected packets (for the active approach). Thus, at the moment the *Routing Control Module* makes a request, the calculated statistical information (the key QoS parameters) is instantly provided to it.

In the active approach, the key parameters are calculated for N generated packets at every T seconds. These measurements provide an indication of the network quality at a single point in time. That frequency (N/T) can be configured by the user to provide more accuracy measurements of the network at more regular times. In the passive approach, the key parameters are calculated for n newer received packets. This value n can also be configured, so that the key parameters can be calculated using more fresh packets. Then, more importance is given to recently received packets because they can provide a more precise indication of the current network situation. Afterwards, the calculated key parameters are compared to the *thresholds*, which are in turn mapped to the QoE expressed as MOS.

In particular, we use the *Mean Opinion Score* (MOS) to represent the QoE of the VoIP service. In this approach, PESQ method is applied to compare an original audio file with the "received" audio file, which supposes suffered some deterioration in its route due to packet loss or other impairments factors. The PESQ algorithm provides raw scores in the range $0.5 - 4.5$, which represent a prediction of the perceived quality that would be given to the received audio by real users in a subjective listening test. The resulting PESQ value is then mapped into a MOS objective listening quality value in the range $1 - 4.5$ according to the mapping function in ITU-T P.862.1 [14].

The MOS values can assume the following interpretation: (1) bad; (2) poor; (3) fair; (4) good; (5) excellent.

In work [15], an exponential relation is established between the MOS and an impairment factor (QoS parameter) such as jitter, or packet loss, for VoIP services. The authors apply the PESQ method and the MOS mapping function respectively using audio files degraded by packet loss ratios along with the reference audio files. The following exponential function is retrieved from the obtained MOS values and the packet loss ratios used in the experiments.

$$QoE \cong MOS = 2.866 \cdot e^{-26.335 \cdot P_{Loss}} + 1.122 \ . \tag{10}$$

We can use the Equation (10) to calculate the MOS values taking into the account the *thresholds* defined for *Packet Loss* P_{Loss}. Table 5 illustrates the calculated MOS *thresholds* representing the QoE level for the *Packet Loss threshold* $- Th_{P_{Loss}}$, which are mapped to the VoIP service quality level.

Table 5. MOS values for packet loss representing the QoE mapped to the service quality level.

Quality level	MOS (1 – 4.5)	$Th_{P_{Loss}}$ (%)	P_{Loss} (%)
good	$4.0 \geq \text{MOS} > 3.7$	0.4	$0 \leq P_{Loss} < 0.4$
medium	$3.7 \geq \text{MOS} > 2.8$		$0.4 \leq P_{Loss} < 2$
bad	$2.8 \geq \text{MOS} \geq 1.1$	2	$2 \leq P_{Loss} \leq 100$

We can observe that MOS has maximum value of 4.0 when P_{Loss} is 0%, i.e., no packet loss is detected and VoIP call has a *good* quality level. As P_{Loss} reaches critical level of 2%, the MOS value drops to 2.8, then signaling a *medium* quality level. If P_{Loss} achieves the maximum 100%, MOS has the minimum value of 1.1, meaning that the quality level is *bad*.

Traffic Control Mechanism. Quality deterioration can occur for several reasons, for instance, packet loss, jitter, and long end-to-end delays. In particular, for WMNs, a cause to these impairment factors could be the high number of collisions between neighboring nodes. A possible reaction to collisions is reducing the amount of transmitted traffic, which causes interference between neighbor transmissions and congests the links. For example, by increasing the *OGM Interval* (the time that defines how often the node broadcasts OGMs, e.g., the default value is 1000ms) to a higher value but still acceptable for a mesh network (to allow BATMAN to recognize route changes in its near neighborhood), the frequency of possible colliding packets is automatically decreased, and the link interference as well.

New fields should be added to BATMAN OGMs, or specific messages should be exchanged between the *Routing Control Modules* of the affected nodes, in order to inform the disturbed nodes that the key QoS parameters and MOS *threshold* have been surpassed, and the respective QoE level has been deteriorated.

If the detected *Packet Loss* PKT_{Loss} is high between neighboring nodes, that means the *thresholds* $Th_{PKT_{Loss}}^{m-b}$ and $Th_{P_{Loss}}^{m-b}$ were exceeded, and also the corresponding MOS *threshold* was overpassed. Therefore, the *Routing Control Module* of those nodes can

apply the following corrective actions.

- Increase the *OGM Interval* of the nodes. Consequently, the *Bit Rate BR* and *Packet Rate* PKT_{Rate} values (bandwidth) will automatically decrease for these nodes, and the delay *MIPD*, the jitter *IPDSD*, and the *Packet Loss* PKT_{Loss} should decrease subsequently. Nonetheless, altering the *OGM Interval* can have effects on the discovery process of the other nodes in highly mobile scenarios.
- Apply a Hop Penalty to the Transmission Quality (TQ) field of each OGM forwarded by the nodes. A high Hop Penalty (the maximum value is 255) will make it more unlikely that other nodes will choose this node as the "Best Nexthop" towards any given destination. Thus, the data traffic will diminish on the nodes' interface. As a result, the inter node interference and collisions will be attenuated on the communication links, and *Packet Loss* PKT_{Loss} should decrease.
- Force the nodes to update its routing table to change the actual "Best Nexthop" to the alternative "Potential Nexthop" towards the Originator. That would force the node to redirect the current data traffic to other route via other neighbor, then possibly avoiding collisions and interference with the problematic neighboring node. Hence, the *Packet Loss* PKT_{Loss} would eventually decrease.

As the delay *threshold* Th_{MIPD}^{m-g} and the jitter *threshold* Th_{IPDSD}^{m-g} have been exceed, the quality level is possibly turned *medium*. If we are dealing with a bandwidth sensitive service, the QoE must be restored to the *good* level. Thus, we apply a bit rate based approach to overcome that issue. The *Routing Control Module* of the node will use the calculated *Bit Rate BR* and *Packet Rate* PKT_{Rate} as a quality metric to select the best route towards the destination (Originator), i.e., the rote that offers the highest throughput. For example, in Figure 2 node O_1 will choose O_2 as "Best Nexthop" to node O_3 since this path has the best TQ. Nevertheless, the route that provides the most bandwidth to node O_1 is through neighbor O_4.
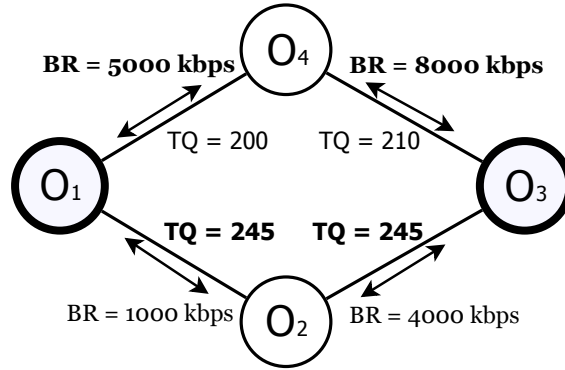
Fig. 2. Bit Rate approach to increase the QoE level.

Therefore, the *Routing Control Module* will update the node's routing table using as "Best Nexthop" neighbor node O_4, which offers the route toward the destination with the highest *Bit Rate BR*. Using this route will certainly decrease the delay *MIPD*, and the jitter *IPDSD* values, then improving the quality level.

4 Conclusion

In this paper, we presented an approach that provides the required Quality of Experience (QoE) level for a VoIP service in a BATMAN mesh network. The approach is comprised of two modules, the *Traffic Monitoring Module* and the *Routing Control Module*. The *Traffic Monitoring Module* measures the status of the network situation by calculating key QoS parameters, which are then mapped to a QoE level. If the QoS performance and the corresponding QoE level are degraded, the *Routing Control Module* executes a response in the node to recover the QoE level.

In a future work, we envisage to implement the proposed monitoring and control modules in a real mesh network environment to evaluate the approach effectiveness for IPTV and VoIP services. We also plan to design a QoE model for IPTV service taking into account the key QoS parameters presented in this work.

References

1. Zhang, W., Wang, Z., Das, S. K., Hassan, M.: Security issues in wireless mesh networks. In Book Wireless Mesh Networks: Architectures and protocols. New York: Springer, 2008.
2. ITU-T, "Methods for subjective determination of transmission quality", ITU-T Recommendation P.800 (1998).
3. ITU-T, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs", ITU-T Recommendation P.862 (2001).
4. ITU-T, "The E-model, a computational model for use in transmission planning", ITU-T Recommendation G.107 (2009).
5. ITU-T, "Single-ended method for objective speech quality assessment in narrow-band telephony applications", ITU-T Recommendation P.563 (2004).
6. Calyam, P., Ekici, E., Lee, C.-G., Haffner, M., Howes, N.: A "GAP-Model" based framework for online VVoIP QoE measurement. In: Journal of Communications and Networks, 9(4), 446–456 (2007).
7. Tao, S., Apostolopoulus, J., Guerin, R.: Real time monitoring of video quality in IP networks. IEEE/ACM Transactions on Networking, 16(5), 1052–1065 (2008).
8. Ding, L., Goubran, R. A.: Speech Quality Prediction in VoIP Using the Extended E-model. In: Proceedings of the IEEE Global Telecommunications Conference (Globecom'03), 2003.
9. Neumann, A., Aichele, C., Lindner, M., Wunderlich, S.: Better approach to mobile ad-hoc networking (B.A.T.M.A.N.). IETF Internet-Draft, expired October 2008 (April 2008).
10. Open-Mesh.net, B.A.T.M.A.N. (better approach to mobile ad-hoc networking), http://www.open-mesh.net/.
11. Interpreting Ping Response Times, http://wiki.netvoice.ca/doku.php?id=trouble:ping-interpretation.
12. RTP: A Transport Protocol for Real-Time Applications, RFC 3550, http://tools.ietf.org/html/rfc3550 (2003).
13. ITU-T, "One-way transmission time", ITU-T Recommendation G.114 (2003).
14. ITU-T, "Mapping function for transforming P.862 raw result scores to MOS-LQO", ITU-T Recommendation P.862.1 (2003).
15. Fiedler, M., Hoßfeld, T., Tran-Gia P.: A Generic Quantitative Relationship between Quality of Experience and Quality of Service. In: IEEE Network Special Issue on Improving QoE for Network Services, 2010.