Testing End-to-End Self-Management in a Wireless Future Internet Environment

Apostolos Kousaridas[1], George Katsikas[1], Nancy Alonistioti[1], Esa Piri[2], Marko Palola[2], and Jussi Makinen[3]

[1] University of Athens
Athens, Greece
scan.di.uoa.gr
{akousar, katsikas, nancy}@di.uoa.gr
[2] VTT Technical Research Centre of Finland
Oulu, Finland
{Esa.Piri, Marko.Palola}@vtt.fi
[3] Octopus Network
Oulu, Finland
www.octo.fi
jussi.makinen@octo.fi

Abstract. Federated testbeds aim at interconnecting experimental facilities to provide a larger-scale, more diverse and higher performance platform for accomplishing tests and experiments for future Internet new paradigms. In this work the Panlab experimental facilities and specifically the Octopus network testbed has been used in order to experiment on the improvement of QoS features by using the Self-NET software for self-management over a WiMAX network environment. The monitoring and configuration capabilities that different administrative domains provide has been exploited in order to test network and service layers cooperation for more efficient end-to-end self-management. The performance results from the experiments that have been performed prove that the proposed self-management solution and the mechanisms for the selection of the appropriate network or service level adaptation improve end-to-end behaviour and QoS features.

Keywords: Experimentation, Testing Facilities, self-Management, Future Internet, WiMAX, Quality of Service

1 Introduction

Several network management frameworks have been specified during the last two decades by various standardization bodies and forums, like IETF, 3GPP, DMTF, ITU, all trying to specify interfaces, protocols and information models by taking into consideration the respective network infrastructure i.e., telecom world, the Internet and cellular communications. The current challenge for the network management systems

J. Domingue et al. (Eds.): Future Internet Assembly, LNCS 6656, pp. 259–270, 2011.

is the reduction of human intervention in the fundamental management functions and the development of the mechanisms that will render the Future Internet network capable of autonomously configuring, optimizing, healing and protecting itself, handling in parallel the emerging complexity. In the autonomic network vision, each network device (e.g., router, access point), is potentially considered as an autonomic element, which is capable of monitoring its network-related state and modifying it based on policy rules that the network administrators have specified.

The scope of this work is to experiment on the improvement of QoS features (e.g., packet loss, delay, jitter) by using a self-management framework over a live network environment and exploiting monitoring and configuration capabilities that different administrative domains provide (i.e. access network and service layer). The effectiveness and the feasibility of various parameters optimization of existing network protocols avoiding manual effort are also tested. The implemented and tested self-management framework has been designed by the Self-NET project [1]. It is based on the so called closed control loop or Monitor-Decide-Execute Cycle (MDE) and consists of the Network Element Cognitive Manager (NECM) and the Network Domain Cognitive Manager (NDCM) [2].

The experimentation work has been carried out as cooperation with Self-NET and PII projects [3] by utilizing Octopus Network [4] testing resources, which are part of Panlab federation [5] of interconnected testing facilities.

The remainder of the paper is organized as follows: The Panlab experimental facilities that have been used as well as their configuration are described in section 2. Section 3 presents the mechanisms that have developed for service-aware network self-management framework. Finally, the experimentation results that have been collected from the tests and the improvement of the performance by using the self-management mechanisms are highlighted in section 4, while section 5 concludes this paper.

2 Experimental Facilities Decription

The testing facility connecting a fixed WiMAX network to the service-aware network is shown in Fig. 1. The WiMAX network environment consists of Airspan MicroMAX base station (BS) [7] and Airspan ProST subscriber station (SS) located on the Octopus testbed at Oulu [4]. The BS and SS operate in a laboratory environment with short distance direct line-of-sight condition, which keeps the signal strength relatively stable and strong throughout the measurement cases. As regards the Self-NET provision side at Greece Distributed Internet Traffic Generator (D-ITG) [8] has been used, which is a software tool that generates traffic at both UoA end machines. This is a Java based platform that manipulates two independent entities, the first is ITGSend process that undertakes the traffic generation and the latter is ITGRecv process that captures the packets to the receiver. Traffic sender can concurrently generate multiple flows with user-defined parameters that can be analyzed from the receiver to extract traffic QoS features (e.g. packet loss, delay, jitter). There are also some contributory entities that assist in improving the traffic simulation by providing log information

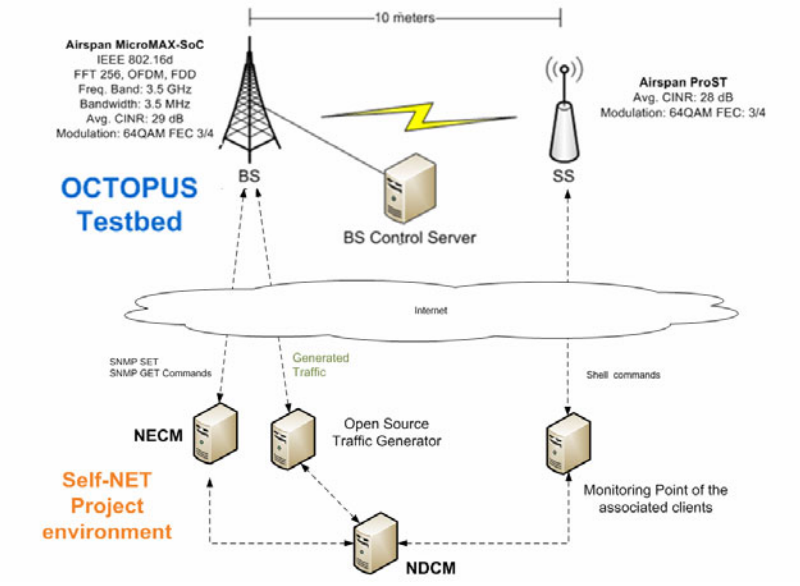

Fig. 1. Octopus testbed WiMAX and Self-NET software federation

(ITGLog), printing and plotting specific metrics (ITGDec, ITGPlot) and remotely controlling the traffic generation (ITGApi). The most well-known network, transport, and application layer protocols are supported by this platform such as TCP, UDP, ICMP, DNS, Telnet, and VoIP (G.711, G.723, G.729, Voice Activity Detection and Compressed RTP).

The Self-NET project carries out experiments over the WiMAX testbed, remotely via the Internet. The experiment required development of an additional BS control software and deployment of IP routing and tunneling between Octopus and Self-NET environments.

We implemented a BS control software (i.e. NECM) to allow dynamically collect WiMAX link information from the BS and to control Quality of Service (QoS) settings on the fly. The NECM changes QoS service classes by setting a new configuration to the BS using Simple Network Management Protocol (SNMP).

IEEE 802.16 standards specify various packet scheduling schemes to ensure required QoS of different traffic types. For example, transmission delay constraints of real-time multimedia streaming are much stricter than that of bulk data transfer. IEEE 802.16d [5], the employed WiMAX testbed is based on, specifies four different scheduling types, namely Unsolicited Grant Service (UGS), Real-time Polling Service (rtPS), Non-real-time Polling Service (nrtPS), and Best Effort (BE). UGS and rtPS are for real-time traffic where maximum latency and jitter can be set in addition to minimum reserved and maximum sustained traffic rates. BE and nrtPS are for delay-tolerant data transmission. However, nrtPS provides assured bandwidth for the traffic flow whereas BE does guarantee nothing for the traffic flow but packets are transmitted if bandwidth available.

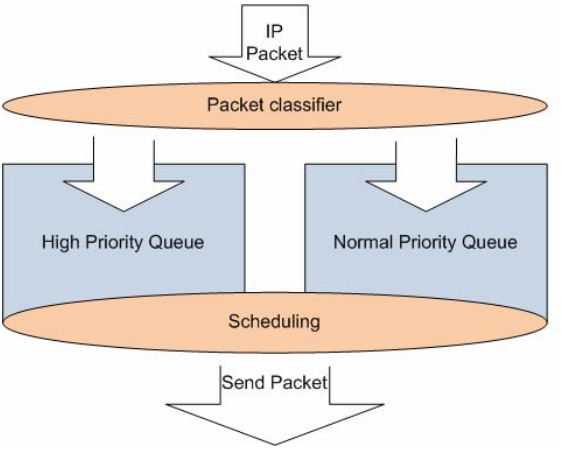

Fig. 2. Downlink packet scheduling

In the employed BS, the aforementioned scheduling types are supported only in the uplink through a request/grant scheduling. In the downlink, the BS supports a scheduling type where several traffic flows can simply be treated with different priorities only, not assuring delay or bandwidth requirements. This downlink scheduling type is capitalized on in our experiments. During default scheduling operation of the BS downlink, all traffic is treated equally by the packet classifier and put to the same normal priority transmission queue where BE scheduling is employed to. The BS controller can be commanded to configure the BS to handle particular traffic flows with higher priority. In this case the BS has two transmission queues of different priorities, as illustrated in Fig. 2. In the downlink scheduling, the packets can be classified to different transmission queues of various priorities based on the IP packet's source and/or destination MAC address, IP address, or port number. In our experiments, we used port numbers to classify the IP traffic flows. We found that during the reconfiguration of the BS service classes packet transmission between BS and SS was temporarily stagnated, however, resulting in break times constantly below a second.

The Self-NET project experiments also required setting up IP routing and tunneling from and to the WiMAX link. Two routers are dedicated on the Octopus testbed for tunneling and routing IP traffic. The user traffic from the Self-NET experimentation is tunneled by using two IP tunnels over the Internet and rerouted over the WiMAX air interface at the Octopus testbed. For the test environment provisioning, the IP tunneling (IPIP) and routing was setup at both ends, which requires two routers at the user premises – one for sending data to the uplink and receiving the downlink flows and one for sending to the downlink and receiving from the uplink.

As depicted in Fig. 3, there are two IPIP tunnels established at the overall topology in order to deploy the federation of these two testbeds. The first tunnel connects the WiMAX BS with the UoA BS Connector (10.1.3.3 – 10.1.3.1) while the second one connects the WiMAX SS with the UoA SS Connector (10.1.3.4 –10.1.3.2), creating an internal 10.1.3.0/24 network between these network entities. The traffic sent from the UoA BS Connector (10.1.1.1) is routed over the IPIP tunnel to the WiMAX BS

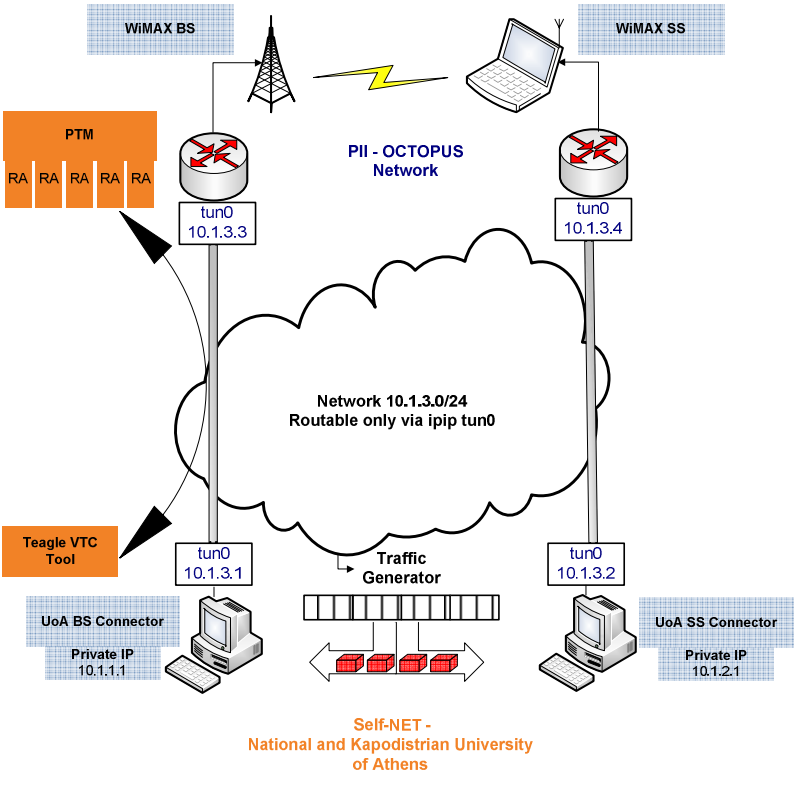

Fig. 3. Network topology and IPIP tunneling

and after the Wireless transmission (DL) to the WiMAX SS, the UoA SS Connector (10.1.2.1) receives the packets via the second tunnel. The respective procedure occurs for the UL, while the UoA SS Connector traffic is tunneled to the WiMAX SS, transmitted to the WiMAX BS and routed again through IPIP tunnel to the UoA BS Connector. During the traffic exchange, the public IPs' are opaque, as the routing procedure explicitly uses the private addresses.

Fig. 3 illustrates also the Panlab federation tools [5] such as Panlab Testbed Manager (PTM), which was installed on Octopus Network to allow Teagle Virtual Customer Testbed (VCT) tool to carry out the topology setup operation. Resource Adapter Description Language (RADL) [9] was used to generate source code for each Resource Adaptor (RA), where, for example, the WiMAX network elements can be considered as available and configurable resources. We decided to use a separate RA for each IP tunneling machine, BS and SS. The RAs managing tunneling send commands to respective machines via SSH to setup both tunneling and routing. The default values are stored in each RA and the user of the VCT tool needs to input only public IP addresses and user credentials for the two external tunneling machines in order to setup the IP tunnels and routes.

3 Mechanism for Service-Aware Network Self-Management

The allocation of Monitoring-Decision Making-Execution (Cognitive) Cycle phases at the NECM and NDCM agents is presented in this section, in order to enable network and service layers cooperation for more efficient end-to-end self-management (Fig. 1). The term cooperation is used to describe the collection of the service-level monitoring data and the usage of service-level adaptation actions for efficient network adaptation.

The NECM of the WiMAX BS constantly **monitors** network device statistics (e.g., UL/DL used capacity, TCP/UDP parameters, service flows), which are periodically transmitted to the corresponding NDCM. The latter one retrieves also associated clients perceived QoS (delay, packet loss, and jitter), the type of service (VoIP, FTP, Video) that each client consumes as well as service profile information from the service providers. The Service-level NECM undertakes to collect service-level data. The Service-level NECM could be placed at the service provider's side, even at premises of network operators. We should point that the Service-level NECM performs also service management tasks (e.g., service composition, discovery) by exploting the Cognitive Cycle (Monitoring-Decision Making-Execution) paradigm. This type of functionality is not part of this work.

The **decision making** engine of the NDCM filters the collected monitoring data from the network and the service level in order to identify faults or optimization opportunities (e.g., high packet loss) according to the specified rules or QoS requirements. In the specific use case the goal of the NDCM Decision making engine is the identification of high average packet error rate (PER) values for the end clients that consume a VOIP service. The second step is the selection of the appropriate configuration action. The following actions are taken into consideration by the NDCM:

- Change the codec that $k_1 \in \Re$ flows use.
- Change the priority of $k_2 \in \Re$ flows at the WiMAX BS.
- Change the priority of $k_3 \in \Re$ flows at the WiMAX BS and the codec of $k_4 \in \Re$ flows.

Two schemes for the selection of the optimal action have been proposed and they are described below (Fig. 4 and Fig. 5).

According to the decision making output the configuration action is transferred either to the WiMAX BS NECM in order to **execute** the change priority action via SNMP set command or to the Service-level NECM in order to execute the codec update. Our scheme is based on the available monitoring and configuration capabilities that network elements provide.

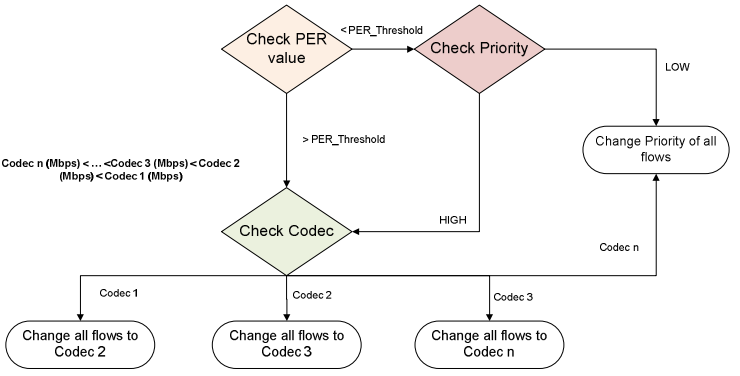

Fig. 4. Decision-making algorithm for configuration action selection – Simple

Fig. 4 presents the simple version of the decision taking scheme Firstly, the PER value is checked in order to select the 'Change Priority' or 'Change Codec' action. If the PER is lower than a pre-defined threshold (PER-threshold) the NDCM decides to change all flows from low priority to high priority service class at the WIMAX BS side. If the priority value is already set as high, then the NDCM proceeds to the 'Change Codec' action. In that case NDCM will check the specific codec that all flows use. According to the Codec type the NDCM decides the transition to a codec that achieves higher data compression, resulting in less data rate requirements; thus reducing packet error rate value. If the clients use the less demanding codec, then the change priority solution is checked. Finally, if none of the above actions are effective then the NDCM will search for an alternative configuration action.

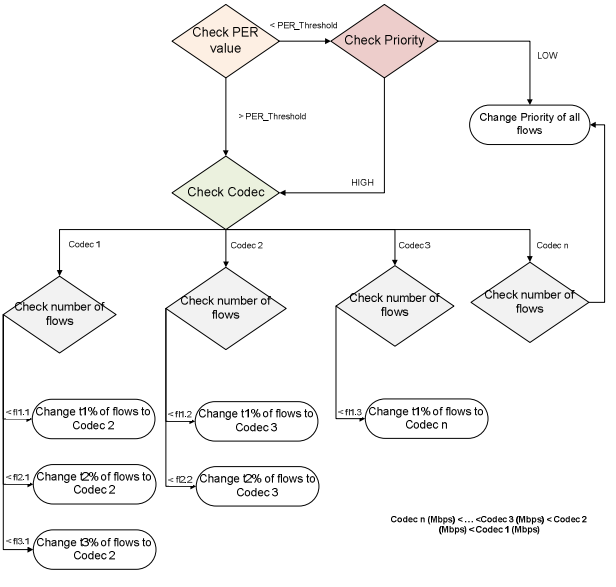

Fig. 5. Decision making algorithm for configuration action selection – Advanced

The above figure (Fig. 5) illustrates the advanced version of the scheme presented above. Specifically, each 'Change codec' action takes into consideration the number of flows $fla.b$, where b denotes the codec type and a the flow threshold of type b which traverse the network and adapts only a percentage of the underlying flows ($ta.b\%$).

4 Performance Results

In this section we provide the performance results that prove the QoS features improvement (e.g., average delay, average jitter, packets dropped) after the reconfiguration actions (e.g., due to an increase of the packet loss rate of VoIP traffic). The configuration actions that have been used are:

- The change of the prioritization scheme at the WiMAX BS side (e.g., from low priority to high priority service class). The following port-based priorities have been set:
  - High Priority: Port range [9850, 10100]
  - Low Priority: Port range [10101, 10250]
- The change of the VoIP codec between the service provider and the end user (service-level adaption). The data rates of each VoIP codec are:
  - G.711.1: 48 kbps
  - G.711.2: 40 kbps
  - G.729.3: 8 kbps
  - G.729.2: 7 kbps
  - G.723.1: 5 kbps

Table 1. Critical thresholds of Packet Loss sharp increment

Codec Type	Threshold of Flows Number
G.711.1 - (fl1.1)	29
G.711.2 - (fl1.2)	46
G.729.2 - (fl1.3)	63
G.729.3- (fl1.4)	97
G.723.1 - (fl1.5)	120

As it is described in Section 3, the decision making schemes that have been proposed for the selection of the appropriate action use a list of thresholds (i.e. *PER-threshold*, $fla.b$, $ta.b\%$). In order to estimate these thresholds accurately and to avoid setting arbitrary values, a first phase of testing took place. Specifically, various number of VoIP flows have been injected into the Octopus Network and different combinations of codec types and priorities (high, low) have been set in order to measure the arising packet error rate, and consequently calculate the appropriate threshold values. The packet loss rate increases, while the number of VoIP flows does, too. However, the

increase rate is not linear since there is a critical value for the number of flows that causes a sharp increase of the Packet Loss (over *PER-threshold* = 4%). This value varies among the different codec types, as it is depicted in Table 1, where the codec thresholds for different flow numbers are presented (*fla.b*).

The following tables depict the improvement on specific QoS features after the re-configuration actions due to an increase of the packet loss rate of VoIP traffic. Table 2 presents the reduction of the packet loss rate after the change of the prioritization (from low priority to high priority service class) at the WiMAX BS of the 28 VoIP flows that use G.711.1 codec.

Table 2. QoS features improvement using high priority service class – Simple scheme

	G.711.1 – Low Priority	G.711.1 – High Priority
Number of flows	28	28
Total packets	28607	26767
Average delay	1.028651 s	1.018491 s
Average jitter	0.012321 s	0.013235 s
Average bitrate	2546.360118 Kbit/s	2580.231640 Kbit/s
Average packet rate	2491.719455 pkt/s	2301.527932 pkt/s
Packets dropped	**573 (2.004 %)**	**12 (0.045 %)**

Table 3. QoS features improvement after total VoIP codec change from G.711.1 to G.711.2 (in the case that service class prioritization change is not effective) – Simple scheme

	G.711.1 – Low Priority	G.711.1 – High Priority	G.711.2 – Low Priority
Number of flows	32	32	32
Total packets	30565	30602	19558
Average delay	**0.514 s**	0.761 s	**0.42 s**
Average jitter	**0.012 s**	0.012 s	**0.016 s**
Average bitrate	2789.06Kbit/s	2717.74Kbit/s	3148.41Kbit/s
Average packet rate	2485.90pkt/s	2504.07 pkt/s	1582.36 pkt/s
Packets dropped	**3442 (10.12 %)**	7929 (20.58 %)	**20 (0.10 %)**

Table 3 depicts the QoS features improvement after a service level adaption of the 32 G.711.1 VoIP flows that traverse the WiMAX BS and face high packet error rate. The modification of the service class prioritization at the BS side (from low priority to high priority class) is not effective, thus an alternative configuration action has been deduced. Specifically, the change of all VoIP codecs between the service provider and the end user, selecting the G.711.2 codec, reduces the number of the dropped packets.

Since the total codec change may be a simple but greedy solution, an advanced adaptation scheme is also proposed and deployed in order to reduce Packet Loss ratio

without sacrificing the provided QoS. This scheme is based on the partial codec adaptation according to the number of the VoIP flows (Fig. 5).

The three tables below showcase the QoS features improvement after the exploitation of the advanced scheme for the selection of the adaptation (Fig. 5). It should be mentioned that the adaptation ratios presented are indicative, as there is a wide range of such ratios according to the codec type and the number of VoIP flows (from 10% to 100%). More specifically, in Table 4, the 27 G.711.1 flows are adapted to 21 G.711.1 and six G.711.2 flows, so this rational adaptation (20%) results to a satisfactory Packet Loss ratio without changing all the codecs.

Table 4. QoS features improvement after partial (20%) VoIP codec change from G.711.1 to G.711.2 – Advanced scheme

	G.711.1 – Low Priority	80% G.711.1 – 20% G.711.2 – Low Priority
Number of flows	27	27
Total packets	29158	27668
Average delay	**0.996881 s**	**1.019370 s**
Average jitter	**0.012377 s**	**0.013391 s**
Average bitrate	2719.482 Kbit/s	2600.848 Kbit/s
Average packet rate	2558.313 pkt/s	2380.823 pkt/s
Packets dropped	**1301 (4.461%)**	**79 (0.285%)**

Table 5. QoS features improvement after partial (50%) VoIP codec change from G.711.1 to G.711.2 – Advanced Scheme

	G.711.1 – Low Priority	50% G.711.1 – 50% G.711.2 – Low Priority
Number of flows	29	29
Total packets	29494	25126
Average delay	**1.075899 s**	**1.070250 s**
Average jitter	**0.013444 s**	**0.014543 s**
Average bitrate	2502.232 Kbit/s	2596.203 Kbit/s
Average packet rate	2539.245 pkt/s	2152.005 pkt/s
Packets dropped	**2621 (8.886%)**	**13 (0.05173%)**

Table 5 presents the changes of the traffic measurements after a 50% codec adaptation. The 29 G.711.1 flows are replaced with 14 G.711.1 and 15 G.711.2 flows and this adaptation contributes to about 8.5% Packet Loss reduction.

The last partial adaptation example is depicted in Table 6, where the adaptation ratio reaches 70% of the flows. The 35 G.711.1 flows are altered to 11 G.711.1 and 25 G.711.2 flows while the resulted Packet Loss scores a 40% reduction.

Table 6. QoS features improvement after partial (70%) VoIP codec change from G.711.1 to G.711.2 – Advanced scheme

	G.711.1 – Low Priority	30% G.711.1 – 70% G.711.2 – Low Priority
Number of flows	35	35
Total packets	31308	25220
Average delay	**1.085282 s**	**1.161476 s**
Average jitter	**0.013925 s**	**0.016809 s**
Average bitrate	2758.579 Kbit/s	2534.948 Kbit/s
Average packet rate	2646.066 pkt/s	2092.565 pkt/s
Packets dropped	**13338 (42.6%)**	**613 (2.43%)**

5 Conclusion

In this paper, we have presented the cooperation between Self-NET and Panlab projects and specifically the usage of Panlab testing facilities (i.e. Octopus testbed) for the experimentation on networks self-management, by using the mechanisms that the Self-NET project has designed. The experiments that have been carried out by using the Octopus wireless network environment prove both the feasibility of the proposed architecture and the QoS improvement (e.g., packet error rate reduction) that could be achieved by applying the appropriate adaptation considering the network conditions.

Different wireless links and networks have different capabilities and often service implementers and providers do not have a possibility to test their service over various networks of different access technologies. Our empirical experiments show how a remote wireless link such as WiMAX can be remotely used. However, in order to provide a wireless link as a bookable resource for a large set of customers, the establishment of the tunnels between the wireless link and the remote user of the link and a correct configuration of the routes need to be automated. This can be achieved by using the tools developed by Panlab testbed federation. Scalability issues and interactions with other network management tasks is part of our future work.

References

1. Self-NET project, `http://www.ict-selfnet.eu`
2. Kousaridas, A., Nguengang, G., Boite, J., Conan, V., Gazis, V., Raptis, T., Alonistioti, N.: An experimental path towards Self-Management for Future Internet Environments. In: Tselentis, G., Galis, A., Gavras, A., Krco, S., Lotz, V., Simperl, E., Stiller, B. (eds.) Towards the Future Internet - Emerging Trends from European Research, pp. 95–104 (2010)

3. Website of Panlab and PII European projects, supported by the European Commission in its both framework programmes FP6 (2001-2006) and FP7 (2007-2013): http://www.panlab.net
4. Octopus Network test facility, http://www.octo.fi
5. IEEE 802.16 Working Group (ed.): IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Fixed Broadband Wireless Access Systems. IEEE Std. 802.16-2004 (October 2004)
6. Wahle, S., Magedanz, T., Gavras, A.: Conceptual Design and Use Cases for a FIRE Resource Federation Framework. In: Towards the Future Internet - Emerging Trends from European Research, pp. 51–62. IOS Press, Amsterdam (2010)
7. Airspan homepage, http://www.airspan.com
8. Distributed Internet Traffic Generator, http://www.grid.unina.it/software/ITG/index.php
9. Resource Adapter Description Language, http://trac.panlab.net/trac/wiki/RADL