Data traffic differentiation and QoS on the train, in fast parameter varying, heterogeneous wireless networks

Milos Rovcanin, Dries Naudts, Daan Pareit, Erwin Van de Velde, Johan Bergs, Ingrid Moerman, Chris Blondia

***Abstract*—Although Internet on the train and train to wayside communication in general becomes more and more available for train operators, there are still a lot of challenges for future research. We previously developed a network platform that is responsible for an uninterrupted and seamless connectivity from the train to the wayside through heterogeneous wireless access technologies. This paper mainly focuses on the concept for providing sufficient Quality of Service (QoS) guarantees in a dynamic train environment. Within this network platform, IPv6 strategies are adopted for QoS, exploiting multi-homing and intelligent aggregation techniques. The implementation that has been done in the Click Modular Router programming environment will also be presented in details.**

***Index Terms* — Click Modular Router, IPv6, railway, data traffic aggregation**

I. Introduction

Providing a communication system between fast moving trains and the ground involves some major challenges [1]. These are mainly caused by a very dynamic behavior of communication channels due to high speed of a train. Most notable are Doppler shifts, variation in line-of-sight (LoS) between train and base stations, frequency selective fading, handover etc. They cause variations in conditions of communication channels that are both spacial and temporal.

Additionally, limiting the offered services to only onboard Internet is not a feasible business case. A viable business case should extend to a broad spectrum of railway communication services like: train control, diagnostics, real time passenger information and entertainment, security services (CCTV surveillance) etc.

Previous research [2,4] suggested that three different types of wireless communication technologies (satellite links, wireless local area networks, mobile operator networks) can be used simultaneously to provide an uninterrupted and seamless connectivity between train and wayside. A typical scenario would be to use a satellite link as the main communication channel, with a backup in public 2G/3G networks when there is no LoS. WiFi communication would be used when the train is in a railway station.

In order to guarantee a user friendly experience, it is inevitable to provide good QoS mechanisms in the network architecture. Specific QoS research is performed in IEEE wireless technologies, such as 802.11e, 802.16e [10] and network protocols by the IETF, e.g. Diffserv, IntServ, RSVP. Currently, some QoS support is already available in commercial solutions that provide wireless networks connectivity and mobility in harsh environments, such as the products of the Icomera and T-Systems.

However, no real integrated solution exists to provide QoS in heterogeneous networks. In literature, there is already a research covering QoS in wireless heterogeneous networks [3]. However, fast changing wireless characteristics during train mobility and the use of different wireless technologies still poses huge challenges in this domain. Moreover, the requirements of the on-board applications can rapidly alter due to the fact that the number of users/clients and applications can quickly change. Thus, it is still a big challenge to provide an adequate integrated QoS solution that can deal both with a heterogeneous network environment and dynamic application demands. In Section II we present the general architecture of the network platform where our software solution is implemented. Section III describes the data and control software modules. Finally, Section IV will describe the software implementation of all the configuration modules and submodules.

II. General architecture

We previously designed [4] a new and modular architecture for the Train-To-Wayside-Control-System (TWCS). All traffic flows from the Mobile Control Equipment (MCE) – a standard onboard gateway for all the outgoing traffic, to the Wayside Control Equipments (WCE) at the wayside, through the modules of the data plane, which is depicted in Fig. 1. We differentiate between connections for reliable transport (straight lines), e.g. TCP (Transmission Control Protocol) connections, and connections for unreliable transport (dotted lines), e.g. UDP (User Datagram Protocol) streams, as they have different requirements.

In the control plane, shown in Fig. 2, some modules (elliptic shape) provide configuration information while others (rectangular shape) process control information which is needed during the operation of the data modules (thick rectangular shape). The information exchange between the control modules is shown with unidirectional or bidirectional arrows. Information that is passed from MCE to the WCE or vice versa needs to be sent over the data plane and is depicted as a dashed line. Within the train-to-wayside communication system (TWCS), we aim at offering an optimized connected experience by prioritizing important

 September 12, 2011.

M. Rovcanin, D. Naudts, D. Pareit and I. Moerman, Department of Information Technology, Internet Based Communication Networks and Services, Ghent University, B-9050 Gent, Belgium. (Phone: +3293314946 ; e-mail: milos.rovcanin@intec.ugent.be, dries.naudts@intec.ugent.be, daan.pareit@intec.ugent.be, ingrid.moerman@intec.ugent.be)

E. Van de Velde, J. Bergs and C. Blondia, PATS Research Group, Dept. Mathematics and Computer Science, University of Antwerp, Middelheimlaan 1, B-2020 Antwerpen, Belgium (Phone: +3232653519; email: erwin.vandevelde@ua.ac.be, johan.bergs@ua.ac.be, chris.blondia@ua.ac.be)

traffic flows, enforcing Service Level Agreement (SLA) levels, respecting traffic flow characteristics (e.g. low latency), traffic shaping according to available bandwidth etc. This is jointly referred to as the QoS aspect of this system.

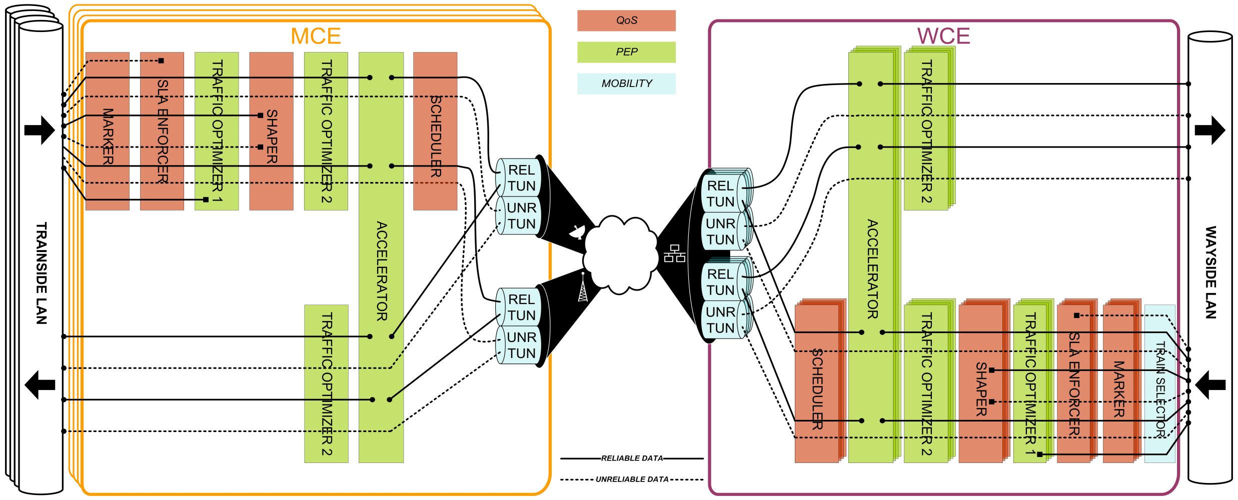


Fig. 1. General architecture – data plane

Concerning provision of Quality of Service in the TWCS, the functionality is logically split into the Marker, SLA Enforcer, Shaper and Scheduler (Fig. 1). They are presented as a part of a data plane of the system. Elements in the control and signaling plane are depicted on Fig. 2. Firstly, the Marker marks packet flows with a service class and priority by using the DiffServ architecture [8], according to different services and their flow characteristics. Next, the SLA Enforcer ensures that all flows that belong to the same SLA (one can use a VLAN per SLA) comply to the SLA stipulations (e.g. maximum data rate, data volume). Then, the Shaper shapes all flows to the available capacity on the wireless train-to-wayside link by dropping packets of flows, with respect to the relative priority of the different flows.

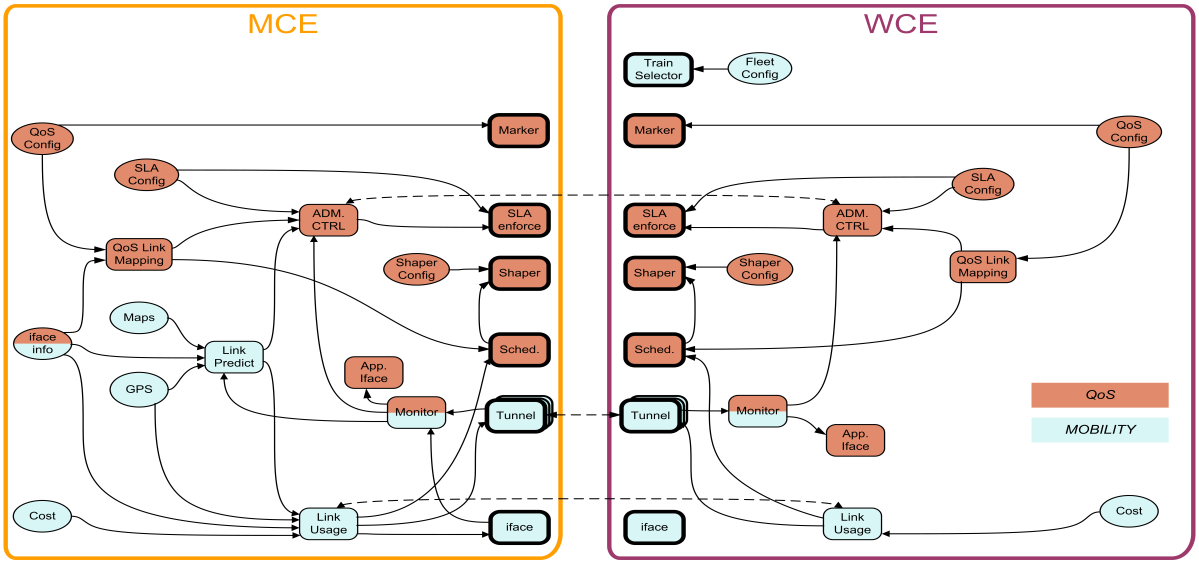


Fig. 2. General architecture – control plane

Finally, the Scheduler needs to schedule all flows on an appropriate link, considering the service class of each flow (e.g. low latency requirement for VoIP).

III. QoS modules

A.1 Data modules

Although mentioned in the brief overview of the system, Scheduler and Shaper elements only use the fact that the traffic has been prioritized to shape it and schedule it to an appropriate link. They are not involved in the data traffic differentiation process itself. Therefore, they are out of the scope of this paper. We will describe in details the main functionalities and implementation of the Marker and SLAEnforcer elements.

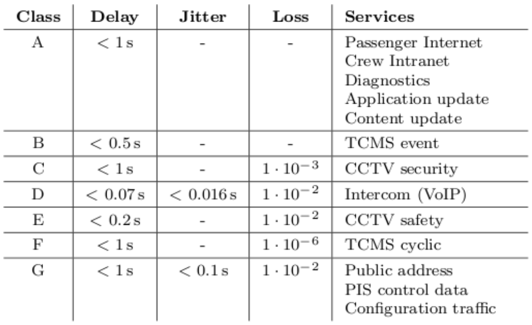
A.1.1 Marker

Both IETF (Internet Engineering Task Force) and ITU-T (International Telecommunication Union) have described classification options, indicating that classes are differentiated by three parameters: delay, delay variation (jitter) and information loss. Note that bandwidth demand is thus not included, as bandwidth shortage can be translated into these parameters. We use the IETF Differentiated Services (DiffServ) architecture for classification within the TWCS. DiffServ is a set of enhancements to the Internet protocol to enable QoS between hosts in different networks.

Traffic is classified into a limited set of service classes, which are treated differently. This allows greater scalability than end-to-end QoS, as used in IntServ, for example.

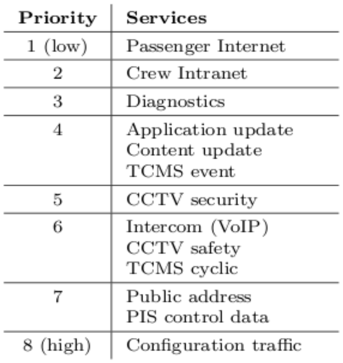
We therefore identify the characteristics of the train-to-wayside (T2W) services and categorize them in different 'service classes' (i.e. data traffic that requires specific delay, jitter and loss characteristics from the network), as stated in Table 1. Next to the network characteristics, a second aspect to consider is the relative priority of the different T2W services. This is given in Table 2.

TABLE I
SERVICE CLASSES



Class	Delay	Jitter	Loss	Services
A	$< 1\,\text{s}$	-	-	Passenger Internet Crew Intranet Diagnostics Application update Content update
B	$< 0.5\,\text{s}$	-	-	TCMS event
C	$< 1\,\text{s}$	-	$1 \cdot 10^{-3}$	CCTV security
D	$< 0.07\,\text{s}$	$< 0.016\,\text{s}$	$1 \cdot 10^{-2}$	Intercom (VoIP)
E	$< 0.2\,\text{s}$	-	$1 \cdot 10^{-2}$	CCTV safety
F	$< 1\,\text{s}$	-	$1 \cdot 10^{-6}$	TCMS cyclic
G	$< 1\,\text{s}$	$< 0.1\,\text{s}$	$1 \cdot 10^{-2}$	Public address PIS control data Configuration traffic

TABLE II
PRIORITY OF SERVICE CLASSES



Priority	Services
1 (low)	Passenger Internet
2	Crew Intranet
3	Diagnostics
4	Application update Content update TCMS event
5	CCTV security
6	Intercom (VoIP) CCTV safety TCMS cyclic
7	Public address PIS control data
8 (high)	Configuration traffic

A third and the last aspect concerning T2W services is the Service Level Agreement (SLA) a device is subjected to. The SLA can e.g. restrict the type of services that a device is allowed to use. Within this architecture, all devices that are subject to the same SLA are put into a separate Virtual Local Area Network (VLAN). This way, SLA identification is indicated in the VLAN header.

All IP packets entering the MCE or WCE are first inspected by the Marker, which needs to determine:

- what flow each packet belongs to,
- service class a flow belongs to
- priority of a flow

A flow is basically a sequence of IPv6 packets [5] from one host to another, which is like an artificial logical equivalent to a call or connection [6]. A non-zero Flow Label (20 bits) value in the IPv6 header (see Fig. 3) uniquely defines a flow. If the Flow Label field has not been set by the source node, the Marker has to determine the flow each packet belongs to.The Marker therefore inspects n-tuple of parameters in the packet header, typically including IP source address, IP destination address, source port number, destination port number and protocol identification. A Flow Label is set by means of a pseudo random generator, so chances that incoming flows have the same Flow Label should be very small [7]. The assignment of a Flow Label to an n-tuple expires when a timer expires after some idle time. This timer can e.g. be based on the typical maximum TCP time-out (a number of minutes or hours).

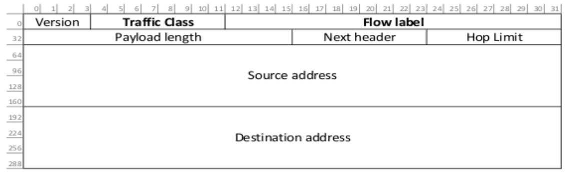


Fig. 3. IPv6 packet header

The packets will also be assigned a value to the Differentiated Service Code Point (DSCP) bits (6 bits) of the IPv6 Traffic Class (8 bits) in the IPv6 header (see Fig. 3). This value indicates into what service class the packets are classified and what priority they have. The other two bits of the Traffic Class field are used for Explicit Congestion Notification (ECN).

A.1.2 SLA Enforcer

After packets have been marked in the Marker, they are entering the SLA Enforcer. The SLA Enforcer will:

- shape all traffic flows according to the applicable SLA,
- drop flows for which the service class requirements cannot be met

The SLA Enforcer jointly shapes all flows with the same SLA. Therefore, the SLA Enforcer needs to know which flows are actually bound to the same SLA. As stated in Section A.1.1, flows are considered to belong to the same SLA if they have the same VLAN tag. The actual SLA specifications and the mapping of VLAN tags to applicable SLAs need to be stated in the special control element, SLA Config module, which is contacted by the SLA Enforcer. Some SLAs will be shaped more rigidly by the SLA Enforcer. This could be the case for SLAs for which the aggregate flows have e.g. exceeded the agreed data volume. For a certain amount of time, e.g. the rest of the month, new flows belonging to this SLA could be blocked or their bandwidth could be decreased or they could be charged for the excessive data volume. Furthermore, the SLA Enforcer will drop all flows that belong to a service class for which the requirements cannot be met. This is all signaled by the control element called Admission Control.

A.2 Control modules and signaling

A.2.1 Interface info

The Interface Information module defines the type and typical characteristics of the on-board interfaces.

A.2.2 QoS Config

The QoS Config contains the requirements per service class and priority, as well as the rule set how to determine what flows will be categorized into what service class and priority.

A.2.3 SLA Config

The SLA Configuration contains information on mapping of VLANs on SLAs ,restrictions for each SLA, maximum allowed data rate and maximum allowed monthly or weekly data volume, allowed flow priorities, allowed flow service classes

A.2.4 QoS Tunnel Mapping

The QoS Config contains the requirements per service class, while the Interface Information determines what each link can offer. Based on this combination, the QoS Link Mapping deducts the supported service classes per link.

A.2.5 Admission Control

The Admission Control will signal the service classes that are currently not supported to the SLA Enforcer, which will drop the relevant flows. The Admission Control knows which service classes are no longer supported by combining information from the QoS Link Mapping, which states what services classes are supported over which link, from the Monitor, which reveals which links are currently available, and from the Link Prediction, which calculates what tunnels are likely to disappear within very short time. Based on the information of these three modules, the Admission Control can calculate the service classes that are currently supported and those that are not. Additionally, the Admission Control checks the SLA Configuration and it can signal to the SLA Enforcer that all flows within a certain SLA need additional shaping, when the SLA stipulations were breached. E.g. if the allowed data volume of a SLA has been surpassed, all flows within this SLA can be rejected or given very limited bit rate by the SLA Enforcer.

IV. Implementation

Complete implementation has been done using the GIT version of the Click Modular Router [9]. Basic router functions as classification, bandwidth shaping, queuing etc. are implemented inside the the elements that consist a kernel of the Click. However, most of the specific functionality of our configuration is incorporated inside the elements we developed.

B.1 Marker

Marker (see Section A.1.1) is designed as a compound element. Its main building blocks are SetFlowLabel, VLANTagger and setIP6DSCP elements (see Fig. 4).

B.1.1 VLANTagger

Instead of using the IEEE 802.1q protocol, better known as VLAN tagging, we developed an element that sets up VLAN designators inside a Flow Label (see Section A.1.1) field of an Ipv6 header. By using four most significant bits of the Flow Label field for this purpose, Ethernet header had become obsolete, so it is stripped off at the moment packet enters a router configuration. The benefits are: lower processing power per each packet and smaller overhead. A drawback is the fact that only 16 different VLANs could be defined and not more than 65536 different traffic flows at the same time. We estimate that our system will remain within these constraints if used in a real life scenario.

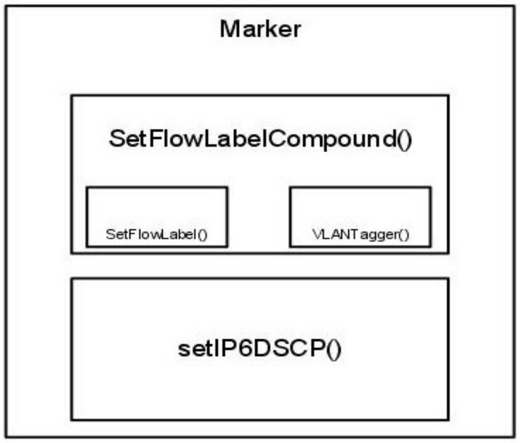


Fig. 4 The Marker module and it's submodules

VLAN tags are added according to the source address of packet. Mapping is statical and predefined and mapping tables are kept inside the SLAConfig element. Different VLAN mean a different sort of a user (1^{st} class passengers, 2^{nd} class passengers, train crew etc.).

B.1.2 SetFlowLabel

SetFlowLabel element is accounted for adding unique flow labels to each packet of every individual flow (see Section A.1.1) and maintaining a hash table of all the existing flows. Every hash table entry contains necessary information about the specific flow (Table III). Hash keys are calculated using the data from the table belwow (destination and source port numbers and a protocol id). Every new flow ID is generated using a pseudo-random generator that takes values form the interval from 0 to 65535 (see Section B.1.1). Due to a specific mechanism of the Accelerator module, flow IDs must stay the same in both directions in the case of a TCP transmission. Therefore, there is an element that incorporates only a part of the SetFlowLabel's functionality and it captures all the packets coming from the MCE, if installed on WCE or vice versa, in order to update a hash table with the first received packet of a TCP flow. This way, the ACK packets will have the same flow ID as the data packets. This element is not presented on the Fig. 4 since it is not technically a part of the Marker, but more of a support stand alone module.

Table III
Hash table entry

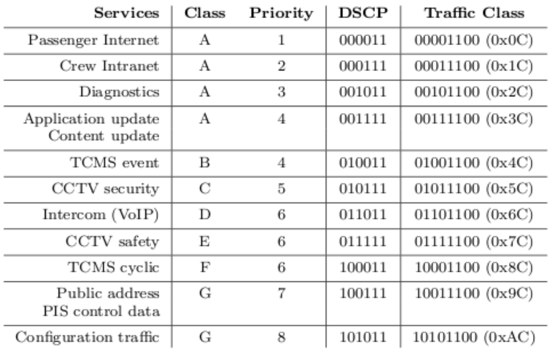
unsigned short	src_port
unsigned short	dst_port
unsigned short	protocol_id
int	flow_id
int	duplicated
int	alive
int	key

SetFlowLabel is allowed to block a flow if it receives a signal from a Shaper element. This happens if there is a significant packet loss on a link. SetFlowLabel will also remove an entry of a flow from a hash table if it has been idle for a predefined period of time (see Section A.2.1).

B.1.3 setIP6DSCP

This element uses users annotations containing destination and source port, set up in the SetFlowLabelElement, to determine a traffic class of a flow and add a designator into the IPv6 header of a packet. Six DSCP bits result in maximum 64 different service classes, but IANA has allocated some pools for standardized service classes [7].

TABLE IV
DSCP bits for T2W services



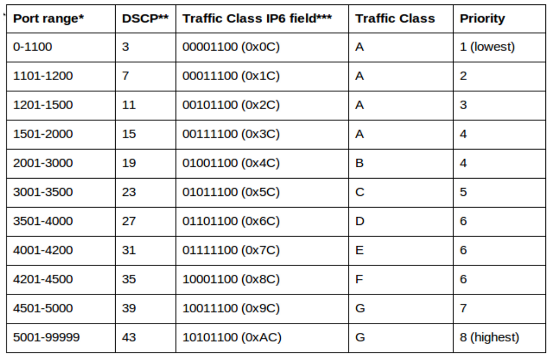
Services	Class	Priority	DSCP	Traffic Class
Passenger Internet	A	1	000011	00001100 (0x0C)
Crew Intranet	A	2	000111	00011100 (0x1C)
Diagnostics	A	3	001011	00101100 (0x2C)
Application update Content update	A	4	001111	00111100 (0x3C)
TCMS event	B	4	010011	01001100 (0x4C)
CCTV security	C	5	010111	01011100 (0x5C)
Intercom (VoIP)	D	6	011011	01101100 (0x6C)
CCTV safety	E	6	011111	01111100 (0x7C)
TCMS cyclic	F	6	100011	10001100 (0x8C)
Public address PIS control data	G	7	100111	10011100 (0x9C)
Configuration traffic	G	8	101011	10101100 (0xAC)

When merging Table I and Table II, we propose to use the DSCP values for the T2W services as stated in Table IV to indicate both the service class and the priority. By labeling each packet with the flow label, service class and priority, the Marker’s decisions are passed via the data plane to all subsequent modules which need to make decisions based on those parameters. For local use, the ‘xxxx11’ bit pattern can be used [8], which allows for 16 different service classes within the TWCS.

Traffic class designator is usually set up by an application that initiates a flow. In our case, in the absence of such an application and for the purpose of an easier parameter changeability, we determine a TC depending of a destination port of a flow (Table V).

Combining information from tables IV and V we can simulate e.g. a VOIP call if we send an UDP flow to a port between numbers 3501 and 4000. This table is placed inside the QOSConfig element. SetIP6DSCP can access that data due to fact that it's been initialized with the QOSConfig as a configuration parameter. Click's SetIP6DSCP element is used to place the designator of a class inside an IPv6 header once it has been determined.

Table V
Destination port to TC mapping



Port range*	DSCP**	Traffic Class IP6 field***	Traffic Class	Priority
0-1100	3	00001100 (0x0C)	A	1 (lowest)
1101-1200	7	00011100 (0x1C)	A	2
1201-1500	11	00101100 (0x2C)	A	3
1501-2000	15	00111100 (0x3C)	A	4
2001-3000	19	01001100 (0x4C)	B	4
3001-3500	23	01011100 (0x5C)	C	5
3501-4000	27	01101100 (0x6C)	D	6
4001-4200	31	01111100 (0x7C)	E	6
4201-4500	35	10001100 (0x8C)	F	6
4501-5000	39	10011100 (0x9C)	G	7
5001-99999	43	10101100 (0xAC)	G	8 (highest)

With the VLAN tag, flow label and TC designator placed inside a header, packet is ready to be sent to the next configuration module – SLAEnforcer.

B.2 SLAEnforcer

This element is also designed as a compound one (Fig. 5)

B.2.1 BWShaper

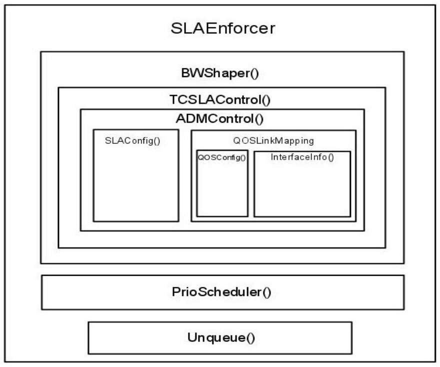


Fig. 5 SLAEnforcer configuration module and it's submodules

This element's purpose is to check whether a traffic class of a flow is allowed per SLA it belongs to, if the maximum allowed data volume per SLA has been reached and if there is a suitable link for transmitting such a flow. When all the conditions are met, packet is being sent towards the links. All above mentioned functionalities are implemented inside different submodules.

B.2.2 TCSLAControl

Initial state of the system is that all traffic classes are allowed for each SLA except for one which is dedicated for hosts that can generate a highest priority traffic (SLA number 6 has been randomly chosen). Of course, there has to be a distinction between different sorts of users (users that belong to SLA for the 2nd class passengers cannot generate a TCSMA configuration traffic). TCSLAControl element checks what traffic class is allowed per certain SLA and if there has been a breach of a maximum data volume allowed per SLA.

B.2.3 ADMControl

ADMControl contains table of traffic classes that are allowed per each SLA. It's content can be changed during the run time by two different write handlers. One for deleting a TC from an SLA and the other one for adding it.

B.2.4 SLAConfig

VLAN tag to SLA mapping is included inside this element. It should be defined in cooperation with the Network Operator (NO). This element also maintains counters for counting the amount of data that has been sent inside every SLA. If the maximum value has been reached, the available bandwidth per SLA will be limited to a certain value until the SLA is renewed. Maximum allowed volume per SLA should also be defined in cooperation with the NO.

B.2.5 QOSConfig

Typical properties regarding delay, packet loss and jitter per every traffic class (see Section A.2.1) are defined inside this element. Properties (Table VI) can be changed during the run time using dedicated write handlers.

Table VI
Properties of different traffic classes

TC	delay	jitter	loss
A	1	1	1
B	0.5	1	1
C	1	1	0.001
D	0.07	0.16	0.01
E	0.2	1	0.01
F	1	1	0.000001
G	1	0.1	0.01

B.2.5 InterfaceInfo

InterfaceInfo element contains typical values of properties for all the available links. The most important properties are: highest possible data rate, delay, jitter and packet loss. It also contains information about the ID of a link and whether a link is active or not. There are two links available at the initial state of the system: satellite and 3G (Table VII). Currently available links can be set down or up at will and the new ones can be added easily using dedicated write handlers.

Table VII
Satellite and 3G link properties

ID	state	rate	delay	jitter	loss
1	1	10Mbps	0.9	0.003	0.00001
2	1	28Mbps	0.7	0.002	0.00001

B.2.6 QOSLinkMapping

Combining the functionalities of the two above mentioned elements, QOSLinkMapping checks if there is an active link capable of transmitting an incoming packet of a certain traffic class. This element is directly used by the Scheduler to list the IDs of all the links that are currently active.

Only if there is a link that meets all the demands of a traffic class that a packet belongs to (delay, loss, jitter), a packet will be sent further on, towards the links.

V. Conclusion

We elaborated the QoS aspects for the network platform of the novel and modular IPv6-enabled TWCS architecture we previously designed.

The basic step, in order to ensure sufficient QoS in a system like this, is a proper traffic class differentiation. This concept will ensure, for example, low latency to critical network traffic such as voice or streaming media, while providing simple best-effort service to non-critical services such as web traffic or file transfers. The implementation explained here will aid future research for testing and implementing QoS in the TWCS systems. Experimental results will be presented at the workshop.

References

[1] W. van Brussel, “Bringing ICT services to trains: technical and economical challenges” 9th Conference on Telecommunications Internet and Media Techno Economics (CTTE), 2010, 1 -7

[2] B. Lannoo, J. Van Ooteghem, D. Pareit, T. Van Leeuwen, D. Colle, I. Moerman and P. Demeester, “Business model for broadband internet on the train”, The Journal of The Institute of Telecommunications Professionals, 2007, 1, 19-27

[3] D.T. Fokum, V.S. Frost, "A Survey on Methods for Broadband Internet Access on Trains," Communications Surveys & Tutorials, IEEE , vol.12, no.2, pp.171-185, Second Quarter 2010.

[4] L. Verstrepen, W. Joseph, E. Tanghe, J. Van Ooteghem, B. Lannoo, M. Pickavet, L. Martens and P. Demeester, "Making a well-founded choice of the wireless technology for train-to-wayside data services," 9th Conference on Telecommunications Internet and Media Techno Economics (CTTE), 2010, 1-7

[5] S. Deering, R. Hinden. Internet Protocol, Version 6 (IPv6) Specification (1998). URL http://tools.ietf.org/html/rfc2460

[6] N. Brownlee, C. Mills, G. Ruth. Traffic Flow Measurement: Architecture (1999). URL http://tools.ietf.org/html/rfc2722

[7] IANA. Differentiated Services Field Codepoints. Website (2010). URL http://www.iana.org/assignments/dscp-registry/dscp-registry.xml

[8] J. Babiarz, K. Chan, F. Baker. Configuration Guidelines for DiffServ Service Classes (2006). URL http://tools.ietf.org/html/rfc4594

[9] E. Kohler, The Click Modular Router. Ph. D. thesis, Massachusetts Institute of Technology (2001).
URL http://pdos.csail.mit.edu/papers/click:kohler-phd/thesis.pdf

[10] Lim, W., Kim, D., Suh, Y., and Won, J. 2009. Implementation and performance study of IEEE 802.21 in integrated IEEE 802.11/802.16e networks. Comput. Commun. 32, 1 (Jan. 2009), 134-143.

[11] K. Nichols, S. Blake, F. Baker, D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (1998). URL http://tools.ietf.org/html/ rfc2474