EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

# A novel network architecture for train-to-wayside communication with quality of service over heterogeneous wireless networks

Daan Pareit[1*], Erwin Van de Velde[2], Dries Naudts[1], Johan Bergs[2], Jan Keymeulen[1], Ivan De Baere[3], Walter Van Brussel[4], Christophe Vangeneugden[4], Patrick Hauspie[5], Gerd De Vos[5], Ingrid Moerman[1], Chris Blondia[2] and Piet Demeester[1]

**Abstract**

In the railway industry, there are nowadays different actors who would like to send or receive data from the wayside to an onboard device or vice versa. These actors are e.g., the Train Operation Company, the Train Constructing Company, a Content Provider, etc. This requires a communication module on each train and at the wayside. These modules interact with each other over heterogeneous wireless links. This system is referred to as the Train-to-Wayside Communication System (TWCS). While there are already a lot of deployments using a TWCS, the implementation of quality of service, performance enhancing proxies (PEP) and the network mobility functions have not yet been fully integrated in TWCS systems. Therefore, we propose a novel and modular IPv6-enabled TWCS architecture in this article. It jointly tackles these functions and considers their mutual dependencies and relationships. DiffServ is used to differentiate between service classes and priorities. Virtual local area networks are used to differentiate between different service level agreements. In the PEP, we propose to use a distributed TCP accelerator to optimize bandwidth usage. Concerning network mobility, we propose to use the SCTP protocol (with Dynamic Address Reconfiguration and PR-SCTP extensions) to create a tunnel per wireless link, in order to support the reliable transmission of data between the accelerators. We have analyzed different design choices, pinpointed the main implementation challenges and identified candidate solutions for the different modules in the TWCS system. As such, we present an elaborated framework that can be used for prototyping a fully featured TWCS.

**Keywords:** railway, train, quality of service, performance enhancing proxy, network mobility, SCTP, TWCS

## 1 Introduction

Wireless voice communication with moving trains has already been studied for decades [1-4]. More recent studies focus on the offering of data services on board of the trains [5], excluding the dedicated safety signaling systems (e.g., European Rail Traffic Management System, ERTMS [6]). Provisioning train-to-wayside (T2W) data services is nowadays one of the booming railway business opportunities. This allows for optimizing operational processes of the train operating company (TOC) and for offering new services to passengers (commuters, travelers etc.). The involved cost reduction and/or additional revenues can yield a positive business case [7,8]. Multiple companies have conducted trials and/or commercial deployments [9].

To offer these T2W services, a centralized communication system offers a more flexible and scalable solution, compared to direct communication from every single onboard device with the wayside base stations. This approach allows for better coverage on board, joint bandwidth optimizations, traffic conditioning and traffic differentiation. The system comprises the actual communication equipment on board and at the wayside, jointly referred to as the Train-to-Wayside Communication System (TWCS). Within this article, we will elaborate the architecture for this TWCS. Note that other

* Correspondence: daan.pareit@intec.ugent.be
[1]Department of Information Technology, Ghent University-IBBT, Gaston Crommenlaan 8 box 201, 9050 Ghent, Belgium
Full list of author information is available at the end of the article

related topics are also important for T2W data services, but they are out of scope for this article. This includes e.g., the design of the onboard network [10-12], the aggregation network of the network operators [13-17] etc.

Research on several TWCS aspects has been extensive 18, but the architectures that have been described so far [12,19-24] only provide a high level view on the complete design or they focus on a specific aspect (e.g., the mobility protocol). Within this article we therefore present a novel TWCS architecture with a more fine grained design.

Furthermore, the importance of Quality of Service (QoS) is often mentioned [12,16,20,22] but has, to the best of our knowledge, never been elaborated within the TWCS context. To this end, we specify the appropriate architectural components and their relationships.

In addition to the QoS aspect, we also elaborate the components for a so called Performance Enhancing Proxy (PEP). The PEP will optimize the bandwidth usage over the wireless T2W links, as these links are typically the capacity bottleneck of the end-to-end connection.

The correct interaction of these components and the design of a network mobility solution are also tackled within this article. We explain the design choices within the architecture and we indicate implementation challenges for anyone that aims at prototyping this architecture. Finally, note that this TWCS architecture is completely designed for use with IPv6 [25], but most of the design could also be applied to IPv4 networks.

We start the remainder of this article by providing an overview of the different actors, services and technologies involved in T2W communication in Section 2. Next, an overview of the network topology and the novel modular TWCS design is given in Section 3. The modules that concern the PEP, QoS, and Network Mobility are elaborated in Sections 4, 5 and 6, respectively. To summarize the complete processing of a packet, the modifications in the packet headers are illustrated in Section 7. Finally, conclusions are drawn in Section 8.

## 2 Overview of actors, services and technologies in train-to-wayside communication

Within this section, we present an overview of the different actors, services and technologies involved in T2W communication in Sections 2.1, 2.2, and 2.3, respectively.

### 2.1 Actors

For communication between an onboard device and a wayside device, different actors are involved:

(1) Network Operator (NOP)
(2) Integrator (INT)
(3) Railway Stakeholders (RST)

(a) Train Constructing Company (TCC)
(b) Train Operating Company (TOC)
(c) Railway Infrastructure Owner (RIO)
(d) Train Constructor Subsupplier (TCS)
(e) Content Provider (CPR)
(f) Train Maintainer Company (TMC)
(g) Train Owner (TOW)
(h) Security Authority (SAU)
(i) ...

The TCC, TOC, RIO, TCS, CPR, TMC, TOW, and SAU are all considered as RST who want to have remote data access to devices on the trains. Note that, in a specific scenario, one company can have the role of multiple actors. Below, we describe these different actors briefly:

- *Network Operator (NOP):* a company that provides trains with wireless access to the wayside. NOPs can be cellular incumbent operators, satellite operators or dedicated wireless data access providers.
- *Integrator (INT):* a company that brings all components together and ensures that these subsystems function together. A key functionality of the INT is to support the routing of data from the trains to the RSTs and vice versa. Therefore, the wayside local area network (LAN) of the INT has two main functions. Firstly, it is the intermediate network between the RSTs, where the wayside users or devices are located who want to access data of onboard devices, and the NOPs, to which the trains are connected. The networks of the RSTs and NOPs are linked to the network of the INT by tunnels over the core Internet or via leased lines. Secondly, it can house wayside devices which need to be shared by multiple RSTs.
- *Train Constructing Company (TCC) or Train Constructor:* manufactures trains which are sold to TOWs. The TCC is typically responsible for repairs during a limited warranty period.
- *Train Operating Company (TOC) or Train Operator:* the entity that operates the trains for passenger and/or freight transport. Such a company can either be private or public.
- *Railway Infrastructure Owner (RIO):* takes care of maintenance and extensions of the railway network infrastructure (usually excluding the metro or tram), of allocating rail capacity and of traffic control.
- *Train Constructor Subsupplier (TCS):* a subsidiary or a supplier of the TCC.
- *Content Provider (CPR):* an organization that creates informational, commercial, educational or entertainment content that is accessible on the train.

- *Train Maintainer Company (TMC):* an organization that is responsible for the maintenance of the trains of a certain TOC.
- *Train Owner (TOW):* an organization that owns the trains. It leases them to a TOC.
- *Security Authority (SAU):* an organization that is responsible for the security on the train.

### 2.2 Services

Following T2W services were distinguished:

- *Passenger Internet:* This includes web browsing, emailing, virtual private network (VPN) access to the corporate network of the business traveler etc. This type of service is typically offered to devices that are owned and carried by the passengers themselves.
- *Crew intranet:* This includes web browsing on the intranet that is available for crew members and which can contain manuals and procedure guidelines, time tables, e-ticketing, an internal telephone directory etc.
- *Diagnostics:* Diagnostic information from onboard components can be sent to the TOC, the TCC, etc. to analyze performance and to pro-actively replace a component before it breaks. This can include e.g., the time it takes for a door to close, temperature of onboard screens etc. Monitoring information of the track [26] is also possible.
- *Application update:* Provisioning of software updates for the applications that are running the T2W services on onboard devices, or updates of the firmware of these devices.
- *Content update:* Provisioning of sporadic content updates for onboard servers for information and entertainment, e.g., annual time tables, advertising, news headlines, touristic information, movies etc. Some of these updates can be very large (e.g., in the case of multimedia files).
- *Train Control and Monitoring System (TCMS) event:* Events that contain sporadic monitoring information, will be sent to the wayside. These messages could be triggered when a sensor reaches a critical (alarm) level (e.g., when the train speed is too high).
- *Closed-Circuit Television (CCTV) security:* Streaming of CCTV security images from the train to a wayside operating center can be used to detect acts of violence or vandalism on board of the train.
- *Intercom:* The TWCS could also provide capabilities for voice calls over IP (VoIP) for communication between crew on board and dispatching personnel at the wayside.
- *Closed-Circuit Television (CCTV) safety:* This includes streaming of camera views to the train driver for train safety applications, e.g., a view on the platform when approaching a station or a view on the railroad when approaching a level crossing.
- *TCMS cyclic:* The wayside actors can get cyclic monitoring information from onboard devices. This includes GPS location, trip number selected, current train state (maintenance, trip running etc.)
- *Public Address:* The Public Address system is used to make announcements by wayside dispatching personnel to passengers on board.
- *Passenger Information System (PIS) data:* Onboard displays of a PIS are updated with live information on connection delays/cancellations, changed platforms, etc.
- *Configuration traffic:* Remote configuration of onboard services and devices to steer their actions, e.g., to switch on or off.

Note that this list covers the applications that are currently known and required for by railway industry. However, if needed, this list can be extended for new services. This will require additional adaptations in Section 4.1.

### 2.3 Access technologies

We can differentiate between three kind of wireless access technologies to provide T2W connectivity: satellite, cellular and dedicated wireless data networks [8].

A combination of some of these technologies is typically considered to be used by the TWCS [7,12,19,22,27-31]. Table 1 gives an overview of the different characteristics of these access technologies [7,32]. Using these values one can obtain a rough idea of what the network will be able to provide in terms of bandwidth and latency.

## 3 Network architecture

In Figure 1, a global overview of the T2W communication topology is depicted. The network of each actor (see Section 2.1) is represented by a separate cloud. Each actual deployment fits into this generic picture, although some alterations might be necessary to reflect the actual physical topology. One company can e.g., have the role of multiple actors (e.g., the INT is also a NOP) or multiple companies could have the same actor role (e.g., multiple TOCs accessing their fleet through the same INT and NOPs).

Multiple services (see Section 2.2) are running on devices on board of the train or on the wayside at the INT, at the RST or at a third party connected to the Internet. The generated data needs to reach its destination on the wayside or on board. Therefore, the data travels from the train over a wireless link to the network of a NOP. Via a leased line or a secure tunnel over the Internet core network the data flows from the NOP to the network of the INT. From the INT, the data is further routed to the RST or the Internet. Data from wayside to

**Table 1 High level overview of the characteristics of different wireless technologies**

| Parameter | Satellite network | Cellular networks | Wireless data networks |
|---|---|---|---|
| Bandwidth | High | Low-high | High-very high |
| | (2-50 Mbps) | (0.17-14.4 Mbps) | (2-50 Mbps) |
| | | (future: 0.1-1 Gbps) | (future: 0.1-1 Gbps) |
| Delay | High | Low-very high | Very low |
| | (500 ms) | (100-1000 ms) | (<50 ms) |
| | | (future: <50 ms) | |
| Current coverage | International | National | Limited |
| | (but no coverage in dense urban areas, tunnels) | (for most recent standards not yet fully achieved) | (new networks needed) |
| Maximum train speed | Very high | High | Low - high |
| | (up to 500 km/h) | (up to 250 km/h) | (120-250 km/h) |
| | | (future: >250 km/h) | (future: >250 km/h) |
| Technologies | DVB-S [78] | GPRS (2G) [79] | Wi-Fi [80] |
| | DVB-S2 [81,82] | EDGE (2.75) [83] | Flash-OFDM [84] |
| | DVB-RCS [85,86] | UMTS (3G) [87] | WiMAX [88,89] |
| | | HSPA (3.5G) [90] (future: LTE (4G) [91], LTE-Advanced) | (future: WiMAX 2) |

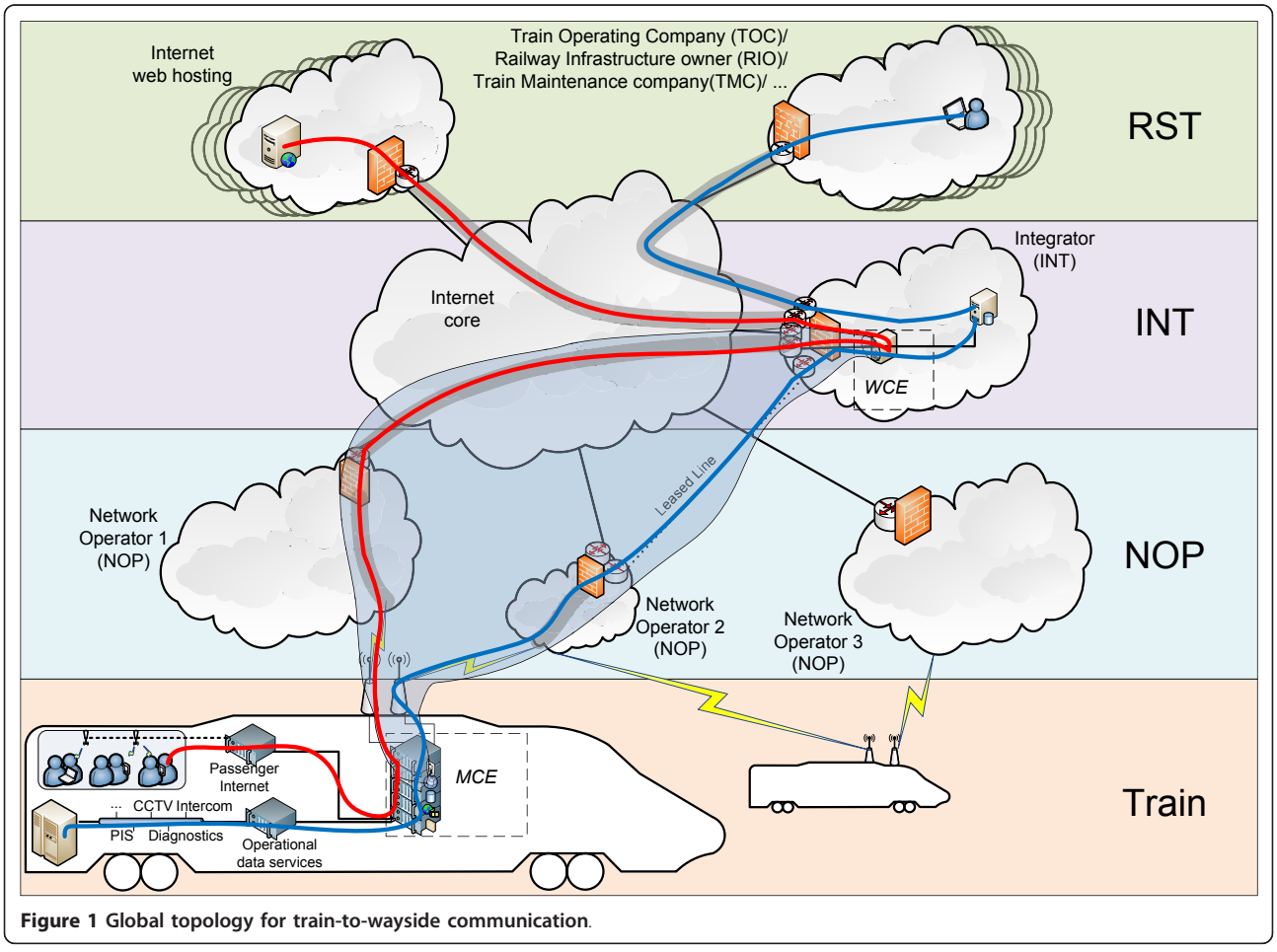



**Figure 1 Global topology for train-to-wayside communication**.

train travels along the same route, but in reverse direction.

To establish communication, each consist, which is a fixed combination of cars of a train, has a mobile communication equipment (MCE) on board, while the INT hosts a wayside communication equipment (WCE) at the wayside, which are indicated in Figure 1. The MCE is the onboard standard gateway for all outgoing traffic, originating from the train, while the WCE is the standard gateway for all traffic towards the trains, originating from the wayside.

The onboard devices are unaware of the fact that they are part of a mobile network and they do not need to implement any special protocols or algorithms. They simply connect to a local access point within the onboard local network (wired or wireless). Outdoor antennas are placed on the roof of the train, which maintain a wireless connection with base stations on the wayside or with a satellite in space. They are physically connected to the MCE inside the train. The MCE links the onboard local network to the outdoor antennas. This way, all passenger and operational data traffic can be transmitted via the MCE and via the external antennas to the wayside.

The MCEs and WCE are jointly referred to as the Train to Wayside Communication System (TWCS) and their interaction, as depicted in Figure 1, is represented schematically in Figure 2.

As 3G and 4G data subscriptions are becoming more popular on consumer devices, direct communication from a device on board with a wayside NOP is also possible. One could therefore question the viability of installing an integrated TWCS system. However, direct communication from a device on board with a wayside NOP is mostly not the preferred way of communication. One of the reasons is the poor coverage of mobile networks inside a train, which also leads to frequent voice call drops on trains. Another reason is that the TWCS relieves all onboard devices from the burden of maintaining a connection at vehicular speeds.

We have designed a new and modular architecture for the TWCS, as shown in Figure 3 and 4. All traffic flows through the modules of the data plane, which is depicted in Figure 3. We differentiate between connections for reliable transport (straight lines), e.g., Transmission Control Protocol (TCP) connections, and connections for unreliable transport (dotted lines), e.g., User Datagram Protocol (UDP) traffic flows, as they have different requirements.

In the control plane, shown in Figure 4, 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 WCE or vice versa needs to be sent over the data plane and is depicted as a dashed line.

Modules can be individual nodes, although they will most probably be implemented on the same machine. Furthermore, note that for the WCE we drew multiple wayside instances of each module in Figure 3, in order to represent a separate process thread per communicating train. Another approach would be to have a single instance of each module, which processes jointly the traffic for all trains at the WCE. It is an implementation choice whether to use a thread per train in a larger program or a single process for all trains. We believe this has a rather limited impact on the architecture, as most

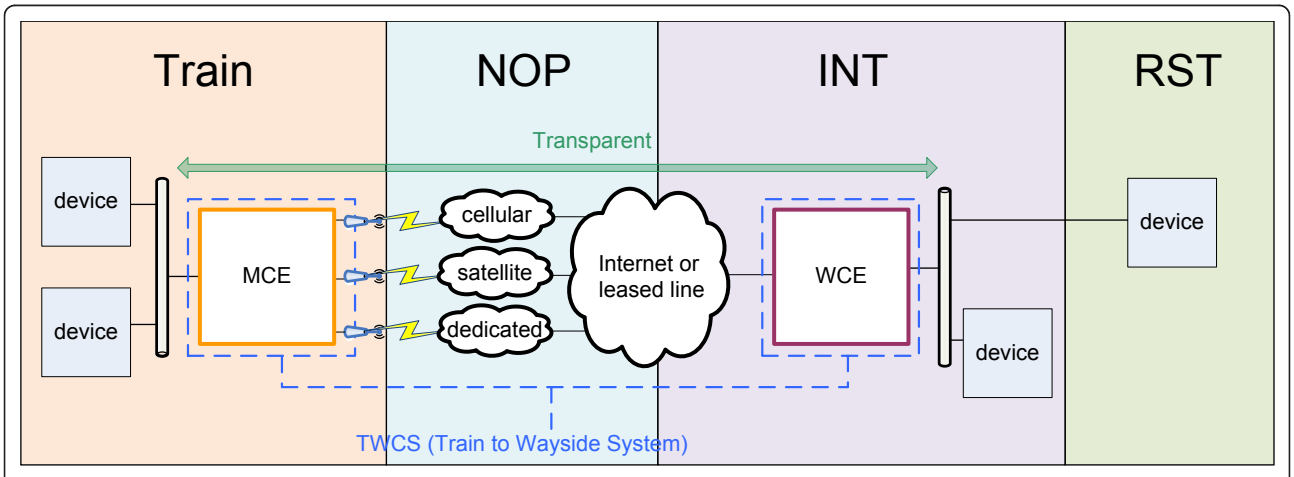



**Figure 2** Schematic representation of interaction between the MCE and WCE, which are the subsystems of the TWCS on board and at the wayside, respectively.

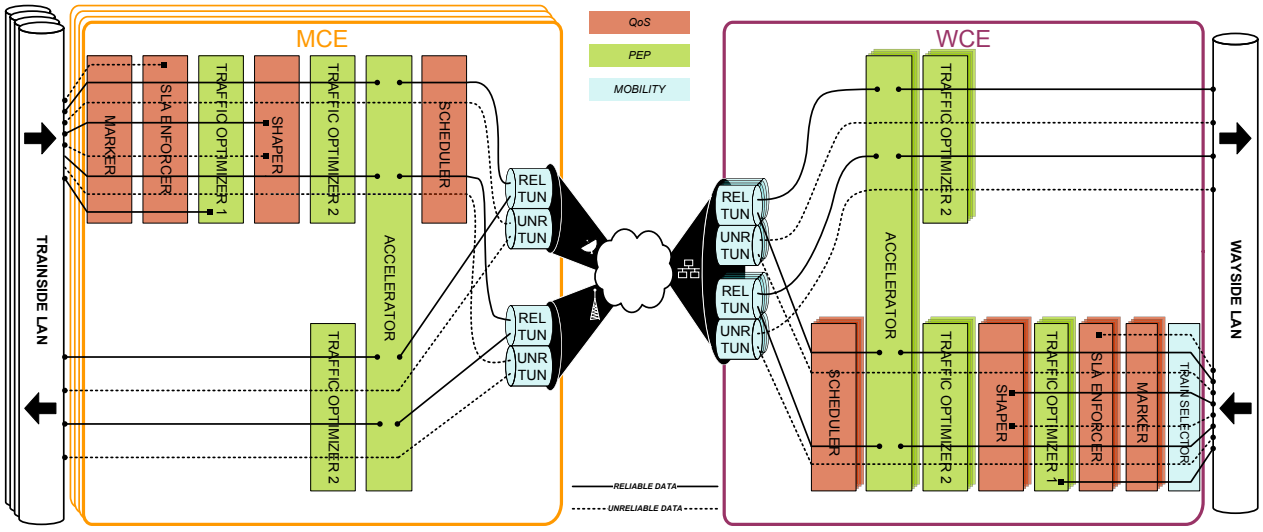



**Figure 3** Modular architecture of the MCE and WCE, which are the subsystems of the TWCS on board and at the wayside, respectively (data plane).

components could be easily implemented to process data traffic flows for multiple trains instead of having dedicated threads per train.

Each module in Figure 3 is colored according to one of the three main functionalities the TWCS provides:

- *Quality of Service* for an optimized connected experience by prioritizing important traffic, enforcing Service Level Agreements (SLA), respecting traffic characteristics (e.g., low latency), traffic shaping according to available bandwidth, etc.

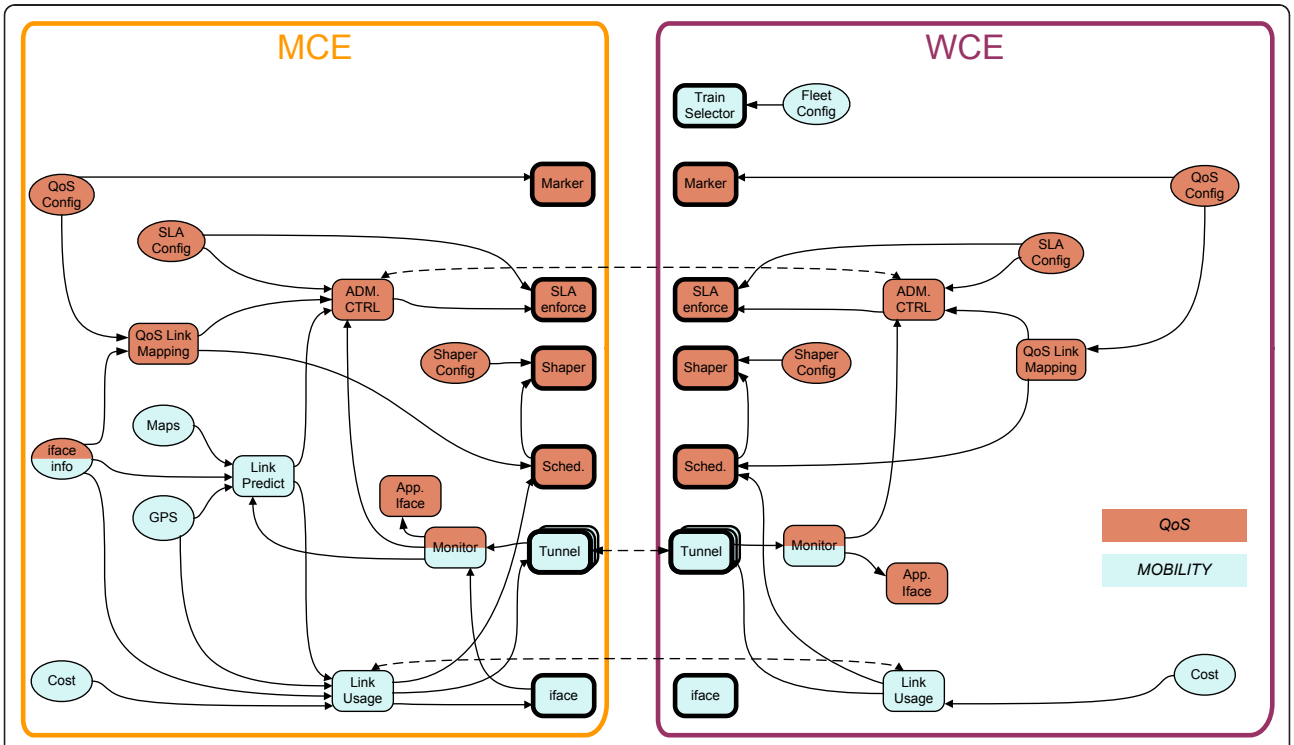



**Figure 4** Modular architecture of the MCE and WCE, which are the subsystems of the TWCS on board and at the wayside, respectively (control plane).

- *Performance Enhancing Proxy (PEP)* for optimizing the overall bandwidth usage of the system with functionalities that include data caching (e.g., a web proxy server), compression, TCP accelerating etc.
- *Network Mobility* for ensuring seamless heterogeneous handovers by setting up multiple tunnels, link failure prediction etc.

The design and implementation suggestions for all modules within each of those functional categories are elaborated in Sections 4, 5, and 6, respectively.

## 4 Quality of service

Within the TWCS, we aim at delivering an optimized connected experience by prioritizing important traffic, enforcing SLA levels, respecting traffic characteristics (e.g., low latency), traffic shaping according to available bandwidth etc. These functions are referred to as the QoS aspect of this system.

Firstly, we describe the classification of the services in Section 4.1. Then, we discuss the data and control modules, concerning QoS provision in Sections 4.2 and 4.3, respectively. Next, implementation challenges are tackled in Section 4.4.

### 4.1 Service classification

Both Internet Engineering Task Force (IETF) [33,34] and International Telecommunication Union (ITU-T) [35] 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 [33,34,36] 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 for greater scalability than per flow end-to-end QoS, as used in IntServ for example.

We therefore identified the characteristics of the T2W services in Section 2.2 and categorized the services in different 'service classes' (i.e., data traffic that requires specific delay, jitter and/or loss characteristics from the network [34]), as stated in Table 2.

Next to the network characteristics, a second aspect to consider is the relative priority of the different T2W services. This is given in Table 3, which was determined jointly with partners in the railway industry [37]. Note that there is no one-to-one mapping of service classes and priorities.

A third and last aspect concerning T2W services is the SLA a device is subject 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 VLAN. This way, SLA identification is indicated in the VLAN header.

### 4.2 Data modules

In this section, we discuss the relevant modules in the data plane (see Figure 3) that deal with QoS: the Marker, the SLA Enforcer, the Shaper and the Scheduler.

#### *4.2.1 Marker*

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

- what traffic flow each packet belongs to,
- the service class the traffic flow belongs to and
- the priority of the traffic flow

A traffic flow is a portion of traffic, delimited by a start and stop time, that is originated from a particular IPv6 source address with a particular transport port number and destined for a particular IPv6 destination address with a particular transport port number [38]. The combination of source address and a non-zero Flow Label (20 bits) value in the IPv6 header (see Figure 5) uniquely defines a traffic flow. If the Flow Label field has not been set by the source node, the Marker has to determine the traffic flow each packet belongs to.

The Marker therefore inspects a 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. As

**Table 2 Service class requirements for T2W services**

| Class | Delay | Jitter | Loss | Services |
|---|---|---|---|---|
| A | < 1s | - | - | Passenger internet; crew intranet; diagnostics; application update; content update |
| B | < 0.5 s | - | - | TCMS event |
| C | < 1s | - | $< 1{\cdot}10^{-3}$ | CCTV security |
| D | < 0.07 s | < 0.016 s | $< 1.10^{-2}$ | Intercom (VoIP) |
| E | < 0.2 s | - | $< 1{\cdot}10^{-2}$ | CCTV safety |
| F | < 1s | - | $< 1{\cdot}10^{-6}$ | TCMS cyclic |
| G | < 1s | < 0.1 s | $< 1{\cdot}10^{-2}$ | Public address; PIS data; configuration traffic |

(a dash (-) means that this metric is no hard requirement for a given service)

**Table 3 Priority of T2W services**

| 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 data |
| 8 (high) | Configuration traffic |

it inspects multiple fields, the Marker is considered as a 'a multi-field classifier' [36]. The Marker then assigns the same Flow Label value to all packets that belong to the same traffic flow (although this field should normally only be set by the source node).

A Flow Label is set by means of a pseudo random generator, so chances that incoming traffic flows have the same Flow Label should be very small [25]. The assignment of a Flow Label to an *n*-tuple expires when termination messages (e.g., TCP FIN) are signaled within the traffic flow or when a timer expires after some idle time. This timer can e.g., be based on the typical maximum TCP time-out time (a number of minutes or hours).

The packets will also be assigned a value (known as a 'codepoint') to the DSCP bits (6 bits) of the IPv6 Traffic Class (8 bits) in the IPv6 header (see Figure 5). 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), which are set to zero when not in use. Six DSCP bits result in maximum 64 different service classes, but IANA has allocated some pools for standardized service classes [33,39]. For local use, the 'xxxx11' bit pattern can be used, which allows for 16 different service classes within the TWCS. When merging Tables 2 and 3, we propose to use the DSCP values for the T2W services as stated in Table 4 to indicate both the service class and the priority.

To determine what T2W service a traffic flow belongs to, the Marker can inspect:

- the MAC address of the source node, if e.g., all traffic of a certain node needs a certain priority,
- the VLAN of the traffic flow, if e.g., all nodes within a certain SLA need a certain priority,
- deep packet inspection to determine the application protocol or even to analyze the application payload (e.g., I-frame versus P or B-frame in video codecs)
- ...

The actual rules that determine how to identify a traffic flow as belonging to a certain T2W service (and thus to determine the service class and priority), will need to be stated in the QoS Config module and are used by the Marker while operating.

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.

The described functionality can be implemented with e.g., the Click Modular Router [40] (using e.g., IPClassifier element) or with Linux netfilter [41].

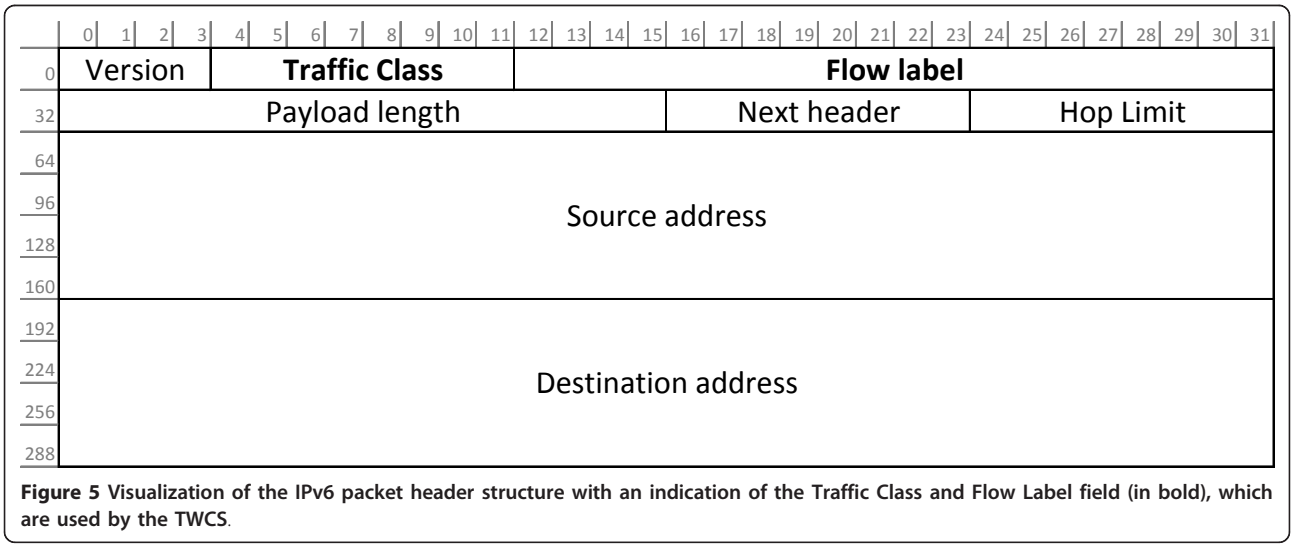



**Figure 5 Visualization of the IPv6 packet header structure with an indication of the Traffic Class and Flow Label field (in bold), which are used by the TWCS.**

**Table 4 Allocation of DSCP bits for T2W services**

| Services | Class | Priority | DSCP | Traffic Class |
|---|---|---|---|---|
| Passenger internet | A | 1 | 000011 | 00001100 (0 × 0C) |
| Crew intranet | A | 2 | 000111 | 00011100 (0 × 1C) |
| Diagnostics | A | 3 | 001011 | 00101100 (0 × 2C) |
| Application update | A | 4 | 001111 | 00111100 (0 × 3C) |
| Content update | | | | |
| TCMS event | B | 4 | 010011 | 01001100 (0 × 4C) |
| CCTV security | C | 5 | 010111 | 01011100 (0 × 5C) |
| Intercom (VoIP) | D | 6 | 011011 | 01101100 (0 × 6C) |
| CCTV safety | E | 6 | 011111 | 01111100 (0 × 7C) |
| TCMS cyclic | F | 6 | 100011 | 10001100 (0 × 8C) |
| Public address | G | 7 | 100111 | 10011100 (0 × 9C) |
| PIS data | | | | |
| Configuration traffic | G | 8 | 101011 | 10101100 (0 × AC) |

#### *4.2.2 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 traffic flows for which the service class requirements cannot be met

The SLA Enforcer jointly shapes all traffic flows with the same SLA. Therefore, the SLA Enforcer needs to know which traffic flows are actually bound to the same SLA. As stated in Section 4.1, traffic flows are considered to belong to the same SLA if they have the same VLAN tag. For VLANs, IEEE 802.1Q [42] is widely used and adds an extra field in Ethernet frames. The actual SLA specifications and the mapping of VLAN tags to applicable SLAs need to be stated in the SLA Config module, which is contacted by the SLA Enforcer.

Some SLAs (signaled by the Admission Control, see Section 4.3.5) will be shaped more rigidly by the SLA Enforcer. This could be the case for SLAs for which the aggregate traffic flows have e.g., exceeded the agreed data volume. For a certain amount of time, e.g., the rest of the month, new traffic flows belonging to this SLA could be blocked, their bandwidth could be decreased or they could be charged for the excessive data volume.

Furthermore, the SLA Enforcer will drop all traffic flows that belong to a service class for which the requirements cannot be met (signaled by the Admission Control, see Section 4.3.5).

Implementation of the SLA Enforcer can be done with Click Modular Router (e.g., Shaper element) or Linux Traffic Control [43] (e.g., with qdisc).

#### *4.2.3 Shaper*

If the aggregate capacity of the available wireless links is smaller than the sum of the total load that needs to be sent from train to wayside (or vice versa), the Shaper will shape the different traffic flows by adapting their aggregate data rate to match the available capacity, based upon the link occupation that is signaled (per service class) by the Scheduler (see Section 4.2.4).

The Scheduler also signals to the Shaper what traffic flows are mapped on what links, so the Shaper will know which traffic flow rates to adapt in order to avoid queue overflows in the Scheduler. This mechanism, which causes a transmitting device to back off from sending data packets until the bottleneck has been eliminated is sometimes referred to as 'backpressure'. This means certain packets will get dropped while others can pass through. Traffic flows with higher priority are favored over those with lower priority. Traffic flows with equal priority should be shaped in a way that each traffic flow gets a fair share of the available bandwidth. The drop probability distributions per priority need to be defined in the Shaper Config. It also includes for which priorities the starvation of traffic flows with lower priorities is allowed, which is typically only for the highest priority class. It also needs to include a minimum bandwidth per traffic flow. If this threshold is reached within the same priority class, it is better to drop a complete traffic flow, rather than to shape all traffic flows equally.

Whilst the SLA Enforcer (see Section 4.2.2) has already shaped all incoming traffic to meet the SLA restrictions, the Shaper thus shapes traffic per traffic flow, depending on the traffic flow priority and available channel capacity (bandwidth). The service class of the traffic flows is not considered for traffic flow prioritizing, as differentiation based on service class is done in the Scheduler (see Section 4.2.4).

Still, the Shaper can look at the service class of each packet to inspect whether a packet can be dropped if service class requirements cannot be met for the specific packet. While the SLA Enforcer (see Section 4.2.2) already drops all traffic flows that belong to a service class for which the requirements cannot be met by the network, the Shaper can additionally drop a particular packet for which the service class requirements cannot be met, although its service class was supported by the network. This could happen e.g., for a packet in a low latency service class that was buffered too long before arriving at the Shaper. In this case, the packet would arrive at its destination too late anyway and could already be dropped at the Shaper in order not to spoil bandwidth on the wireless links.

Implementation of the Shaper can be done with Click Modular Router (e.g., Shaper element) or Linux Traffic Control [43] (e.g., with qdisc).

#### *4.2.4 Scheduler*

The traffic that has successfully passed all preceding modules finally arrives at the Scheduler, which allocates

the traffic to the appropriate link, based on matching the service class of the traffic flow with the delay and jitter properties of the link. The QoS Link Mapping component, see Section 4.3.4, defines what service classes can be supported on what links. The Scheduler will signal to the Shaper what traffic flows are mapped onto what links, so the Shaper knows what traffic flows to shape when a link is becoming overloaded. For the Shaper to know when a link is becoming overloaded, the Scheduler signals the load on each link. Therefore, it uses the principle of active queue management (AQM) [34]. When the queue occupation for a link is lower than a certain minimum threshold, the Scheduler signals the Shaper to allow more traffic. When the queue occupation exceeds a certain maximum threshold, the Scheduler signals to allow less traffic. This is similar to the random early detection (RED) [34] mechanism (which can also be used in the Shaper, see Section 4.4.2) but with signaling information over the control plane instead of marking or dropping packets on the data plane.

Traffic flow priority is no longer considered here as this was already done in the Shaper. Service classes that require e.g., low latency or low loss rate will need to be mapped on a link with similar characteristics and a scheduling algorithm (e.g., weighted round robin) could prioritize certain service classes.

Once a traffic flow has been mapped onto a certain link, all following packets of the traffic flow will be allocated to the same link. This is done in order to reduce jitter. Only when this link goes down, the traffic flow will be rescheduled.

### 4.3 Control modules & signaling

In this section, we discuss the modules on the control plane that deal with QoS: Interface Information, QoS Config, SLA Config, QoS Link Mapping, Admission Control and Application Interfacing. They are indicated in Figure 4.

#### *4.3.1 Interface information*

The interface information module defines the type and characteristics of the interfaces that are needed to connect to NOPs. As the WCE has no wireless interfaces, this does not need to be implemented at the WCE. Instead, other measures will have to be taken at the wayside (see Sections 4.4.6 and 6.5.3).

#### *4.3.2 QoS config*

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

#### *4.3.3 SLA config*

The SLA Configuration contains information on

- the mapping of VLANs on SLAs
- the restrictions for each SLA
  - maximum allowed data rate
  - maximum allowed monthly or weekly data volume
  - allowed traffic flow priorities
  - allowed traffic flow service classes

The above only lists the SLA restrictions. Performance guarantees per SLA, on the other hand, can only be given if there is a dedicated network connection or if the INT also has a SLA with the NOP.

#### *4.3.4 QoS link mapping*

The QoS Config contains the requirements per service class, while the interface information determines what each T2W link can offer. Based on this combination, the QoS Link Mapping deducts the supported service classes per link.

#### *4.3.5 Admission control*

All traffic flows for which the service class requirements cannot be met, need to be pro-actively rejected. There is no point in sending them over the wireless links as they will be discarded at their destination. Therefore, the Admission Control will signal the service classes that are currently not supported to the SLA Enforcer, which will drop the relevant traffic 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 will still be supported in near future) and those that are not.

Additionally, the Admission Control checks the SLA Configuration and it can signal to the SLA Enforcer that all traffic 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 traffic flows within this SLA can be rejected or given very limited bit rate by the SLA Enforcer.

#### *4.3.6 Application interfacing*

The design of the MCE/WCE allows to offer an API (application programming interface) to end devices on board or at the wayside. This way, end devices can optionally subscribe to events that the Application Interface will generate to indicate the availability of e.g., high throughput or low delay with general network condition messages (e.g., “no network available”, “low bandwidth”, “average bandwidth”, “high bandwidth”). The end devices

can use this information for their internal reasoning to find a suitable moment to start a certain application (e.g., only transfer movie files for the entertainment system when high bandwidth is available).

These events will be generated based on the input from the Monitor. This component has a view on the performance of the wireless links and tells the Application interface what the available bandwidth, jitter and delay is.

### 4.4 Implementation challenges & suggested solutions

In this section, we discuss various implementation issues concerning QoS that might occur.

#### *4.4.1 Timely dropping of traffic flows for unsupported service classes*

Each new traffic flow that belongs to a service class that cannot be supported by the network will be dropped by the SLA Enforcer (see Section 4.2.2). This functionality needs to be performed in the SLA Enforcer, before any other processing. If this would only be done in any of the next modules, those traffic flows would unfairly be taken into account by the SLA Enforcer and add up to the consumed data volume or data rate. This would result in less bandwidth or data volume than end users are entitled to.

#### *4.4.2 TCP synchronization*

When the data rate of the data traffic flows is adapted by the Shaper, its buffers will fill up and the source node should need to throttle back. To this end, AQM [34] could be a solution by using RED [34] queues. When the queue's occupation has reached a certain threshold, it can start dropping some random packets. This way, the congestion control of the relevant source node will react and decrease its send rate. This will also decrease the rate at which the Shaper's buffers are filled up. Instead of dropping packets, they could also be marked using Explicit Congestion Notification (ECN), which would lead to the same data rate decrease but without overhead retransmission. AQM is a better solution than just waiting for buffer overflows ('tail drop') to happen, as the latter would lead to TCP synchronization among the different source nodes. All nodes would take measures for congestion and the network will become under-utilized firstly and flooded afterwards when all nodes are increasing their send rate once again.

#### *4.4.3 Encryption*

When the data payload is encrypted, no deep packet inspection can be performed by the Marker. The Marker can only look at the headers (if these are not encrypted) to determine the traffic flow, service class and priority.

#### *4.4.4 Packet rescheduling upon link failure*

When a link goes down, the packets that were scheduled for this link need to be rescheduled to another link. The question rises how to make an appropriate data structure for the buffer implementation for the scheduled packets within the Scheduler, in order to still allow rescheduling.

A first suggestion to implement the buffers of the Scheduler, would be to have a FIFO queue per service class. When a link polls for a packet, a scheduling algorithm will then select the queue from which the first packet could be popped and sent. However, as we do not spread a traffic flow across multiple links, we need to check if the considered packet belongs to a traffic flow that is mapped to this link. If this was not the case, the next service class queue should be considered. If the first packet in each of the service class queues belongs to a traffic flow that is mapped to another link, the traffic would stall on the link that polled for a packet.

In order for a tunnel/link not unnecessarily to remain idle, another suggestion for implementation would be to have separate queues per service class and per link. However, when a link goes down, the packets that are still present in the relevant queues would need to be moved into the queues of another link. Putting them at the end of the queue would not be fair, as they should rather be merged based on their time of arrival in the queues.

A solution to avoid the disadvantage of possibly stalling links, as in the first suggestion, and of having to merge queues, as in the second suggestion, is to use a data structure per service class which allows to select any packet rather than only the first one, e.g., a linked list, a hashmap. This way, the Scheduler will first select the hashmap of the appropriate service class and then take the next packet out of it that belongs to any of the traffic flows that are mapped on the link that polls for a packet. Conceptually, this resembles to a 'virtual queue' per service class per tunnel/link (as in the second suggestion), but when a link disappears, the service class queues are automatically merged into the ones of the remaining links.

#### *4.4.5 VLAN persistency*

The VLAN marking of each packet needs to be passed from MCE to WCE and vice versa if it is required to still know the SLA of the packet at the other side of the wireless links. Sending the header with the VLAN tag of each packet results in quite some overhead over the wireless T2W links. Therefore, it would be better to strip this header and instead use some bits in the Flow Label field of the IP header to indicate the VLAN tag and thus the corresponding SLA.

#### *4.4.6 Discrepancy between MCE and WCE*

If a SLA allows a certain amount of bandwidth or data volume to be used, with e.g., a free to choose downlink/uplink ratio, communication between the Admission Control on the MCE and the WCE could be needed.

## 5 Performance enhancing proxy

Within this TWCS architecture, we aim to centrally optimize the overall bandwidth usage within modules that are jointly referred to as the 'PEP'. We discuss the

relevant data modules and their order in Sections 5.1 and 5.2, respectively. The control modules are described in Section 5.3. Finally, we tackle the implementation challenges concerning the PEP in Section 5.4.

### 5.1 Data modules

In this section, we discuss the relevant modules in the data plane (see Figure 3) that deal with bandwidth optimization: the traffic optimizers and the accelerator.

#### *5.1.1 Traffic optimizer 1*

Traffic optimizer 1 (TO1) is a module which tries to decrease the load on the wireless links. It can instantly reply to a device with the information it requested, without always having to send data over the T2W link by

- monitoring information requests from devices and locally caching the information responses
- responding to future identical information requests with the cached information

The functioning of TO1 will thus mostly be situated at the application layer of the open systems inter-connection (OSI) model [44] and includes typically some kind of transparent caching proxies, such as a web proxy, a domain name server (DNS) cache and a simple mail transfer protocol (SMTP) proxy. If a traffic flow is eligible for this kind of traffic optimization, the connection is terminated here and the TO1 replies with locally cached content or it sets up a new connection with the destination server on the other side if there was no cached data available or if the cached data was outdated.

The web proxy can be especially useful for the 'Passenger Internet' and 'Crew Intranet' services (see Section 2.2). It will monitor hypertext transfer protocol (HTTP) traffic requests and keep the HTTP responses from the web server in a local cache. Web browsers do not need to explicitly configure the web proxy in their settings (this would not be scalable and would be too difficult for passengers to configure), but the proxy will operate transparently.

Furthermore, all services could benefit from a DNS cache in TO1, as DNS is an ideal candidate for caching, and thus for performance gain, as it is designed as a hierarchical distributed naming system with DNS records having a long lifetime, typically in the order of a couple of hours. A negative cache, which maintains unresolvable records, could also be kept. As the slow propagation in the whole DNS system does not support fast addition and deletion of records, the cache should neither.

Whereas the web proxy and DNS cache will decrease the network load by responding with locally cached copies, an SMTP proxy is meant for email relaying and will always have to forward the email that originated from an end user. However, the SMTP proxy provides a convenient 'local email buffer' as SMTP offers no timely delivery guarantee. Therefore, the email can be locally stored in the SMTP proxy and be forwarded at a slower rate or only when there is enough free capacity available over the wireless links.

The TO1 module will likely prove to be the most useful within the onboard MCE (rather than in the WCE), as it is more likely that the server application will reside on the wayside instead of on the train. Nevertheless, there might be some specific uses for a caching on the wayside, so this module can be implemented at both sides.

The most widely used, open source HTTP caching proxy is called Squid [45]. For DNS caching, BIND [46] and Dnsmasq are available open source software. For SMTP proxying, a widely known open source agent is sendmail [47].

#### *5.1.2 Traffic optimizer 2*

The second traffic optimizer (TO2) is a module which aims to reduce the actual bandwidth of the traffic flows by using data compression.

The counterpart on the receiving end does exactly the opposite, it decompresses data so the receiver perceives the data traffic flow as unaltered.

Whereas TO1 thus tried to decrease the traffic load that is to be sent over the wireless networks, TO2 will now forward all incoming traffic, but it will optimize the given load in order to consume less bandwidth.

TO2 should inspect the data traffic flows to check whether it is useful to perform data compression, as there are a number of cases where compression by TO2 is unwanted:

- Data compression makes sense for e.g., HTTP traffic, but not for network time protocol (NTP) traffic. The latter type of data traffic consists of small packets which should not be delayed by data compression.
- When original data is encrypted, e.g., in VPN tunnels, data compression will have little or no effect. Data compression tries to remove statistical redundancy. However, encrypted data appears to be completely random data without any statistical redundancy.
- Similarly, it could be useful to check whether the data has been compressed already, as additional compression is in this case not likely to further decrease the data size but could even be counterproductive due to the extra control information.

For the 'Passenger Internet' and 'Crew Intranet' services (see Section 2.2), web pages can e.g., be compressed using classic compression algorithms such as LZW (LempelZivWelch) and Huffman encoding. However, modern implementations of web servers and browsers tend to do this themselves. Statistics [48] show that this is the case

for 66% of web pages, including embedded resources such as images, scripts and stylesheets. This means that compression by the TWCS will only be useful in about a third of the cases for web pages. When considering the increasing importance of multimedia content over traditional webpages, the importance of compression by the TWCS will decline as multimedia content is typically already compressed.

Another option could be to introduce lossy compression. For pictures, one could e.g., reapply the lossy image codec to obtain lower picture quality or one could reduce the image resolution. When considering web pages, about two thirds [48] of the actual transmitted size of a web page is made up by images, so a rather significant performance gain could be expected. However, introducing (additional) lossy compression requires adequate knowledge about the considered use case and whether this degraded content provision is acceptable for the end user. Furthermore, as content is now actually being altered, one finds oneself in a jurisdictional gray zone.

For data compression, an open source implementation of a compression proxy exists, called ZIP Proxy [49]. It compresses text using the gzip algorithm and uses JPEG of JPEG2000 to compress images.

#### *5.1.3 Accelerator*

The accelerator will try to optimize incoming TCP traffic flows by

- mitigating performance degradation resulting from the rather large round trip time between MCE and WCE for some wireless access technologies
- mitigating performance degradation resulting from multiple competing TCP connections

For this end, TCP Acknowledgments (ACKs) are sent by the Accelerator at the transmitting side to halt the TCP traffic flows (which is known as 'TCP ACK spoofing'), leading to a lower perceived round trip time at the node were the traffic flow originates from. Next, the data of the traffic flow is disposed of its TCP mechanism and the payload is encapsulated in UDP datagrams (to still maintain the necessary control information, e.g., source and destination addresses and ports) and sent over a tunnel which guarantees reliable transport (see Section 6.2.2). This way, there is no competition for bandwidth by the TCP congestion mechanisms of each individual TCP data traffic flow. This leads to an increased overall system throughput. On the receiving side, the Accelerator sets up a new TCP connection for each traffic flow, based on the encapsulated original control information. This way, the destination endpoint will not notice that the TCP connection was split by the Accelerators. The behavior of this Accelerator is thus distributed between a sending module and a receiving module, contrary to typical TCP accelerators.

PEP for Satellite Links (PEPsal) [50] is a known implementation for TCP acceleration over high latency links, but it is implemented as a single component. Space communication protocol standards (SCPS) is another TCP accelerator, originally mainly intended for satellite links, which consists of a transmitter and a receiver component. TCP Speaker is an implementation of similar functionality in the Click Modular Router framework.

### 5.2 Module order

We put the SLA Enforcer (see Section 4.2.2) in front of TO1, but one could argue that it is unnecessary to let the SLA Enforcer shape the data traffic that is locally provided by TO1 because the onboard network will typically have enough capacity and the local data traffic does not use any resources on the wireless links. However, as depicted in Figure 6, it would still be unfair not to restrict the achievable data rates on the onboard network. The reason for this is that the cached data has once been transferred over the wireless link on request of a certain user, say Alice. This happened most probably at a slower rate than the achievable onboard data rate that is obtained when another user, say Bob, is requesting the same data and is provided with a locally cached copy. Suppose user Alice's SLA allows for higher data rates than Bob's, Bob would have gotten higher data rates than Alice in this example, see Figure 6a. As this is unacceptable, the SLA enforcer is placed before TO1, see Figure 6b.

Furthermore, TO1 is to be put before the Shaper. Within TO1 (see Section 5.1.1), some traffic flows containing information requests might be terminated without having to go over the wireless T2W links as TO1 itself can immediately reply with a locally cached copy of the requested information. Therefore, the information traffic flow carrying the request does not need to be shaped onto the outgoing wireless links and TO1 is placed before the Shaper.

On the other hand, TO2 needs to be put behind the Shaper, in order for the SLAs to be applied correctly, see Figure 7. This is explained as follows. Suppose TO2 was put before the Shaper and two users, say Alice and Bob, are sending the same amount of data and having the same SLA (concerning maximum data rate). Furthermore suppose Alice is sending a compressible data traffic flow, allowing for TO2 to optimize the traffic by decreasing the amount of data to be sent, while Bob is sending an incompressible data traffic flow, for which TO2 optimizations are superfluous. In this case the Shaper receives less bytes in the optimized traffic flow that originated from Bob than in the one from Alice. However, it will send an equal amount of bytes for both traffic flows that it receives from TO2, as the SLAs of Alice and Bob are the same. When considering the original data that Alice and Bob sent, this results in a higher effective throughput for Alice,

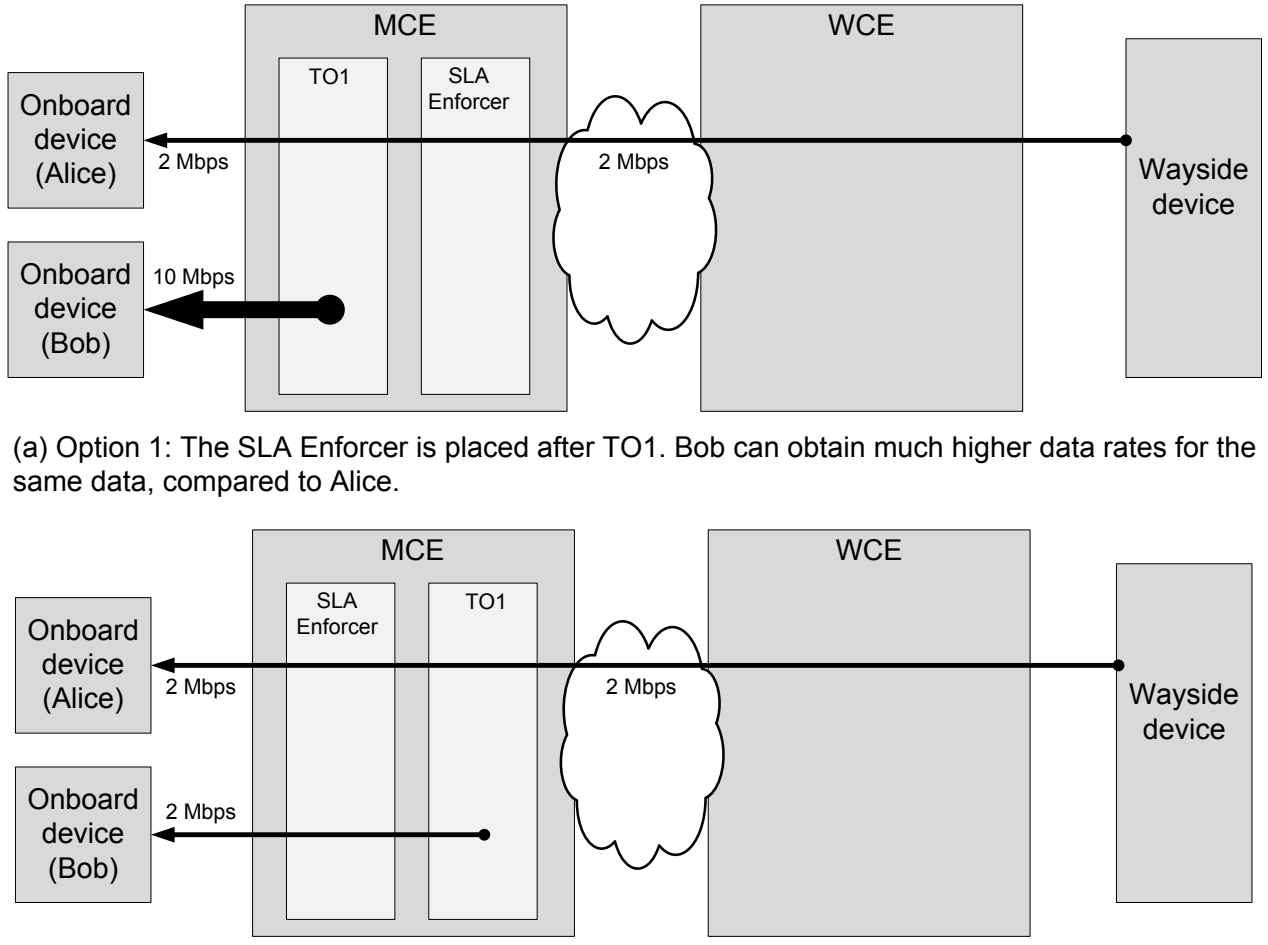



**Figure 6 Comparison of two options for the relative positioning of the SLA Enforcer and TO1, which has a caching and/or proxy functionality, while considering two users, Alice and Bob, who are requesting the same data at different moments in time**.

compared to Bob, see Figure 7a, which would be unfair. Therefore, TO2 is placed after the Shaper, see Figure 7b. One can still try to argue that the Shaper (see Section 4.2.3) has shaped all incoming traffic flows to a certain available aggregate capacity of the wireless links and that applying TO2 after the Shaper will result in a suboptimal use of the capacity of the aggregated links. However, the excess capacity will be signaled from the Scheduler (see Section 4.2.4) to the Shaper (in relative terms, not in absolute terms) and the Shaper will gradually adapt the aggregate load it passes through accordingly. The Shaper can thus be shaping for a total capacity that is higher than the effective available aggregate capacity on the links. In this way, all data traffic flows benefit from optimizations in TO2, rather than a single user if the TO2 would be placed before the Shaper.

### 5.3 Control modules & signaling

TO1 does not have to exchange information with other modules, as it can work perfectly on its own.

On the other hand, the TO2 in the MCE passes information to TO2 in the WCE or vice versa, in order for the receiving module to know what compression algorithm was applied and how to decompress the data.

Likewise, the Accelerator in the MCE also passes information to the Accelerator in the WCE or vice versa, in order for the receiving module to know what the end destination of the original TCP traffic flow is. For this end, the UDP protocol could e.g., be used.

Communication with other modules is unnecessary and the TO1, TO2 and the Accelerator were therefore not included in Figure 4.

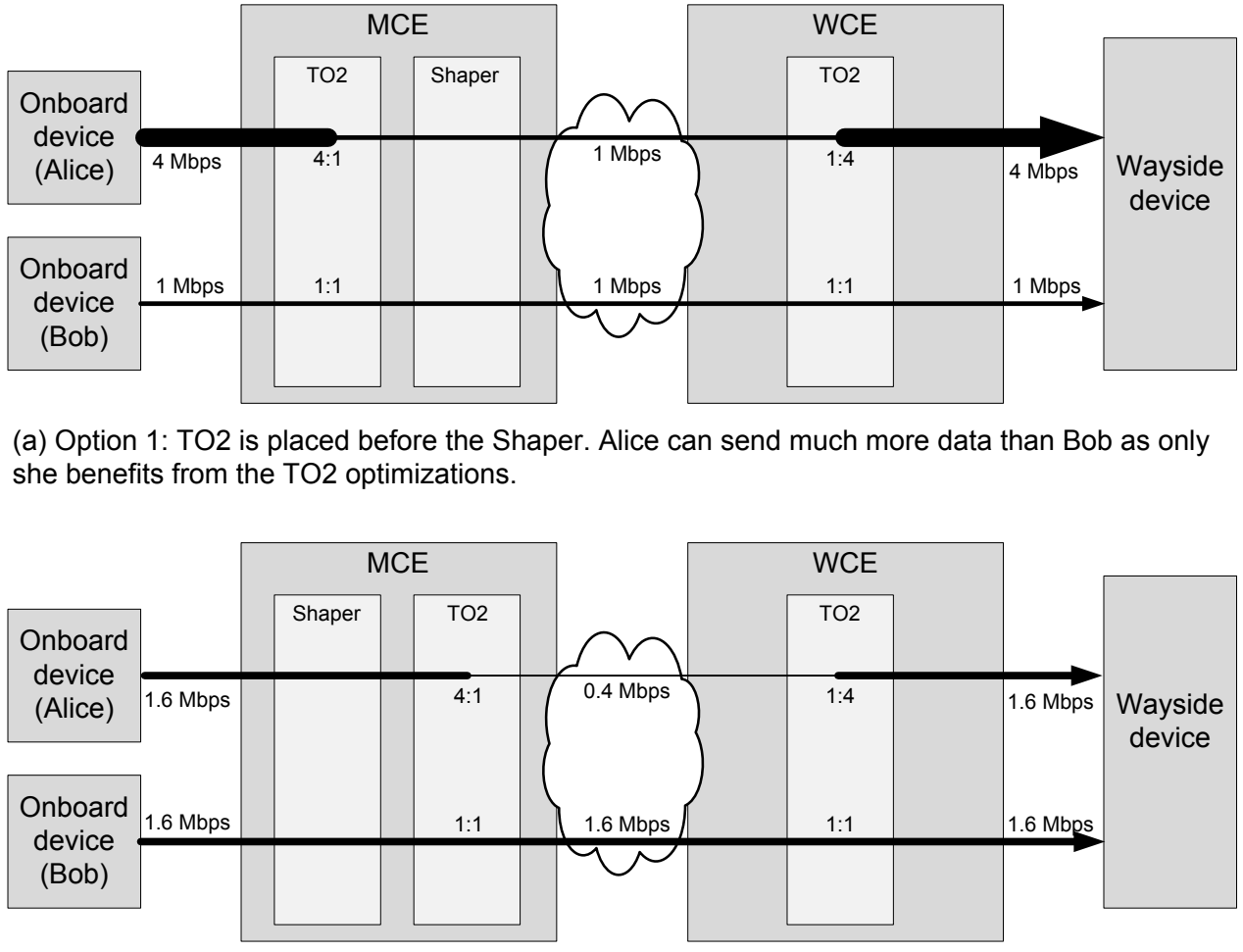



**Figure 7 Comparison of two options for the relative positioning of the Shaper and TO2, which has a data compression functionality, while considering two users with the same SLA, Alice and Bob, who are sending a compressible and incompressible data traffic flow, respectively, at the same time**.

### 5.4 Implementation challenges & suggested solutions

In this section, we discuss various implementation issues, concerning the PEP, that might occur.

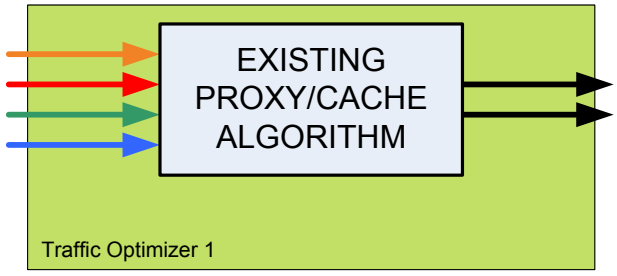
#### *5.4.1 Hardware constraints*

When implementing caching proxies (in TO1), hardware constraints are to be considered. A caching web proxy can easily take a significant quantity of storage, which may not be available in a restricted environment as is common in ruggedized railway equipment. A good trade-off should be made between the decrease in load on the wireless links and the storage cost.

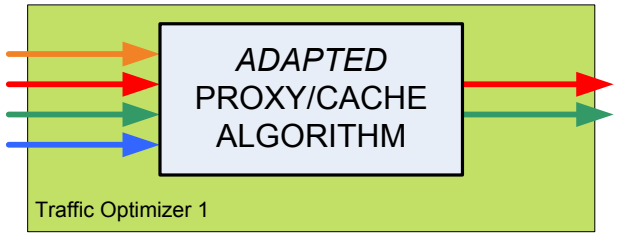
Likewise, for compression proxies (in TO2) a trade-off should be made between the decrease in load on the wireless links and the cost of a more powerful processor. This is because, depending on configuration and choice of algorithms, a compression proxy can consume a considerable amount of processing power.

#### *5.4.2 Marking persistency*

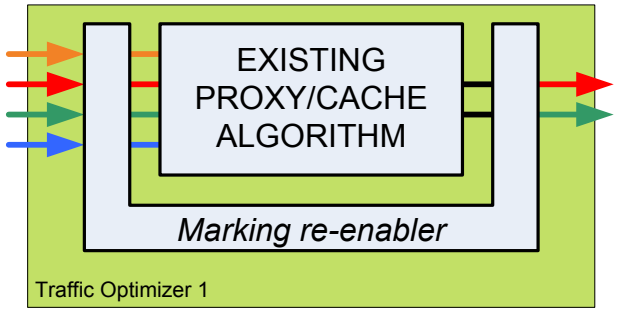
A problem that occurs with caching proxies is that they break the end-to-end principle. This implies that any marking given to a certain input traffic flow by the Marker module (see Section 4.2.1) is likely to get lost for traffic which is handled by TO1, see Figure 8a. A first solution is to set the cache algorithm to have the same marking on its outgoing traffic flows as for its related incoming traffic flows, see Figure 8b. When the implementation is not capable of doing so or when modification of the cache algorithm is not possible, a system could be designed which tracks incoming and outgoing traffic to and from the cache and reassigns the correct




(a) Original: The labels that were marked onto packets of a traffic flow are lost after TO1 because new traffic flows are created within TO1.




(b) Option 1: Adapt the proxy/cache algorithm to have the same marking on its outgoing traffic flows as for its related incoming traffic flows




(c) Option 2: Track incoming and outgoing traffic to and from the proxy/cache and reassign the correct marking

**Figure 8 Different options in order to maintain packet marking in TO1.**

marking to each outgoing traffic flow, as shown in Figure 8c. However, it would nevertheless be a difficult task to match the correct incoming traffic flow to the outgoing one.

#### *5.4.3 Encryption*

When encryption hides the TCP header (e.g., IPsec VPN), TCP acceleration will be impossible. On the other hand, when the TCP header is not encrypted but its payload is, the TO1 proxy servers will not be able to function as they e.g., do not know the type of application that is used. Furthermore the compression that occurs in TO2 will have no effect on encrypted data as encryption has removed any statistical redundancy. We do not elaborate on these aspects, but they are briefly mentioned here as they should be reckoned with when deployed on the field.

## 6 Network mobility

While a train operator can have a preferred main wireless access network, switching to another technology and/or NOP will be unavoidable on certain parts of the track to obtain better coverage (e.g., satellite access in rural areas, cellular networks in urban areas) or to avoid roaming costs (e.g., for an international train). To increase the available data capacity, multiple wireless links can even be used simultaneously ('load balancing').

When the train travels (at high speed) through the coverage area of different wireless networks, a mobility solution is required in order to tunnel data from T2W and for the trains to remain accessible from the wayside: the train needs to signal to the wayside on what IP addresses it can be reached and to notify when any of these change.

We firstly describe the addressing scheme in Section 6.1. Next, we describe the data and control modules in Sections 6.2 and 6.3, respectively. The choice of an appropriate tunneling protocol is elaborated in Section 6.4 and implementation challenges concerning mobility are tackled in Section 6.5.

### 6.1 Addressing

Within this architecture we assume that all actor networks are IPv6 capable [25]. Furthermore, the following configuration is necessary:

- each train has one or multiple predefined/64 subnet(s).
- a 'INT Gateway' in the INT's wayside network acts as a router to each of these subnets for the 'outside world'
- the train always sets up its outgoing tunnel connections to the (preferably fixed) IP address of the WCE.
- the standard gateway for all devices on board of the train is the MCE.

Assume that the INT has been assigned the 2001:db8:1::/48 subnet[a] and this network is reachable from the IPv6 Internet via the INT Gateway with IP address 2001:db8:2::1 and acts as a router for 2001:db8:1::/48. The INT has further subdivided 2001:db8:1::/48 in several/64 subnets. One of these subnets is for internal use in the wayside LAN, the others are assigned to the trains. We assume that 2001:db8:1::/64 acts as the INT wayside LAN subnet, and 2001:db8:1:1::/64 - 2001:db8:1:ffff::/64 are assigned to the trains. This is shown in Figure 9.

The train that has been assigned the 2001:db8:1:1::/64 subnet, powers up. It brings up all of its interfaces and tries to connect to the different networks. As soon as at least one interface has established a connection and has received an IP address of the service provider it is connected to, the train will contact the INT. Suppose the train has one active interface, which has been assigned the following IP: 2001:db8:ffff::1. The train connects to the INT by setting up one or multiple tunnels. The MCE on the train establishes a connection with the WCE at the INT by opening the tunnel connections to WCE's IP address, assume 2001:db8:2::2. The traffic that is destined for this train is automatically routed correctly to the INT Gateway, as it acts as a router for this 2001:db8:1:1::/64 subnet. Within the INT network, the traffic is handled by the WCE, which will send it in the tunnel to the MCE, by routing it to the 2001:db8:ffff::1 address, which is also depicted in Figure 9.

This method does require a higher-level authentication mechanism to make sure no unauthorized tunnel connections are set up to the WCE. Every time an interface transitions from the "not connected" to the "connected" state, this process is repeated. Additionally, the Tunnel Module informs the INT of the state of each interface of the train. This allows the train to notify the INT that an interface is up or down, and notify the INT of IP address changes.

### 6.2 Data modules

In this section, we discuss the relevant modules in the data plane (see Figure 3) that deal with network mobility: the Train Selector, the Tunnel and the Network Interface.

#### *6.2.1 Train selector*

This module is only necessary in the WCE, at the wayside. It finds out to which MCE (and thus which train) a certain data traffic flow should be sent. Typically, this will be decided based on the destination IP address, as one or more subnets can be allocated to the onboard network of a train (see Section 6.1). Routing to those subnets is advertised to the outside world by the INT. When traffic destined for one of those subnets arrives at the INT's network, it is routed internally to the WCE. The Train Selector module will then look up to what train the destination IP address belongs to. For this end, the Train Selector contacts the Fleet Configuration database, which contains a mapping of onboard IP subnets to a train ID and the associated processing thread. Next, it allocates the traffic to the corresponding processing thread. Each module within the same processing thread is dedicated to a single train (and thus a single MCE).

#### *6.2.2 Tunnel*

The Tunnel module provides the

- forwarding of the data traffic from the onboard LAN to the wayside LAN (at the INT) and vice versa
- protocol to maintain a connection with the wireless network of a NOP

All data traffic flows are tunneled towards the WCE before effectively being routed to their destination. The Tunnel component on the receiving side will take the data out of the tunnel and forward it to its destination. The appropriate protocol is discussed in Section 6.4.

#### *6.2.3 Network interface*

Once packets, encapsulated in the tunneling protocol, finally arrive at the Network Interface, they will be encapsulated by the data link layer protocol of the

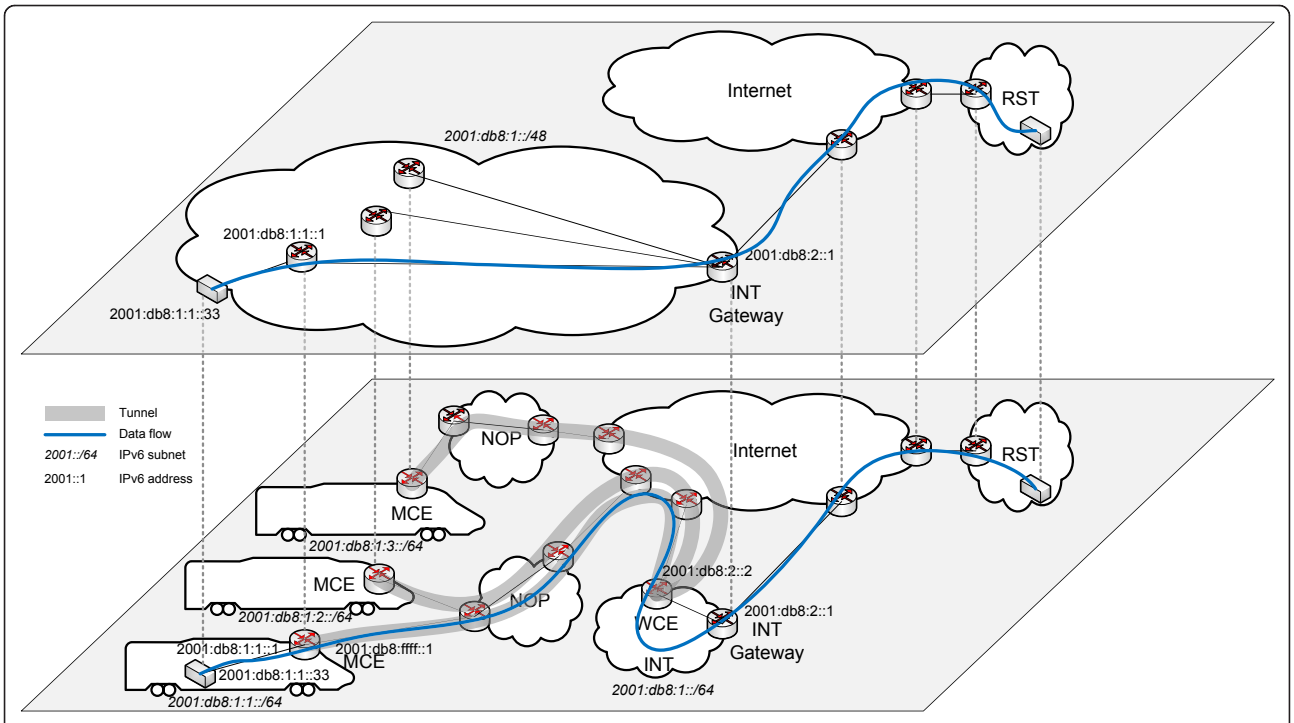



**Figure 9 IPv6 addressing in TWCS**. The upper plane shows the network *as perceived* by communicating nodes at the wayside at the RST, while the lower plane reveals the *actual routing* over the TWCS tunnels.

wireless link when sent from the MCE, or in an Ethernet packet when sent from the WCE.

If the (wireless) technology and the operator that are used, support multiple QoS levels for the transmission, the service classes that are used within the TWCS can be mapped onto the appropriate QoS levels on the link.

### 6.3 Control modules & signaling

In this section, we discuss the modules on the control plane that are related to network mobility: Link Usage, Link Prediction and Monitor. They are indicated in Figure 4.

#### *6.3.1 Link usage*

During the train journey, the Link Usage continuously decides what links can currently be used for data transmission.

This decision is based on inputs from the Interface Info, Cost and Link Prediction modules. Firstly, the Link Usage has to know what interfaces are on board. This information is obtained from the Interface Info module.

Next, the Cost module gives information about the current economic cost of using a certain link, based upon which the Link Usage might avoid using some links, in order to mitigate high data costs. The use of a satellite link with a fixed cost per month might e.g., be preferred above a link with a variable rate, like a volume billed mobile network link. It could also e.g., be useful to know when the monthly data volume with a wireless service provider would have been consumed before the end of the month. In this case, the Link Usage might favor the use of other links over this one, in order to mitigate additional high data costs for this provider, or the Link Usage might command the Network Interface to use another Subscriber Identity Module (SIM) card of the same NOP. Another example is the use of a cellular link of the national operator abroad (e.g., on an international train), which yields high roaming costs and is preferably avoided by switching to a cellular link of an operator of the visited country. Next to the information from the Interface Info module and the Cost module, the Link Prediction will tell the Link Usage which link is likely to go down, so the latter can take the appropriate measures to gradually stop scheduling traffic flows onto this link and to disable this link.

Besides optimizing the connectivity of a particular train, it can also try to optimize bandwidth utilization for the whole fleet, by e.g., not using a medium that is shared amongst the complete fleet (e.g., satellite connection) when it has enough capacity available on other wireless links.

The Link Usage will notify several modules of its decision. Firstly, it will notify the Scheduler (see Section 4.2.4) to stop or start scheduling traffic flows on the concerning link. Secondly, the Link Usage will notify the Tunnel module to tear down or to start up tunnels over this link and the tunneling protocol (see Section 6.4) of

the Tunnel module will do the necessary signaling to the other side. The Link Usage might also need to notify the Link Usage at the other side about the links that are in use. Finally, it also enables or disables the specific Network Interface with the appropriate parameters (e.g., SSID, Service Set Identifier, on a Wi-Fi link). When a train travels through many stations and depots where hotspots are used to connect to, it can often take a considerable time for the Wi-Fi client on board to scan all the different SSIDs used by these hotspots. Location information from the GPS module can speed up the process by directing the client to connect to one specific SSID in that location.

#### 6.3.2 Link prediction

The Link Prediction estimates future link availability, based on a pre-defined or adaptive link model, based on historical data and based on current measurement data (provided by the Monitor module).

Using positioning from a GPS device and relating this to geographical maps, it is possible to foresee network interruptions or 'blind spots' (e.g., satellite connections will go down in a tunnel). Speed information from a GPS device is also useful. For instance on parts of the tracks where trains are running at more than 250 km/h it makes no sense to connect using EDGE technologies, so for these parts of the tracks the modems can be inhibited for using this technology.

This way, the Link Prediction is able to calculate what link is likely to go down soon. Those link(s) are signaled to the Link Usage and Admission Control (see Section 4.3.5).

As Link Prediction can only be done properly at the train and not at the wayside (as the WCE has no direct access to the wireless links), this module does not need to be implemented at the WCE.

#### 6.3.3 Monitor

The Monitor continuously monitors the condition of all active interfaces and of the tunnels running over those interfaces. It reads and processes information from the Network Interfaces about the up or down status, the physical data rate, BER, signal quality etc. This way, it can estimate how the links are performing. From the Tunnel module, the Monitor retrieves information from the tunneling protocol (e.g., heartbeats, 'going down' messages, packet error rate (PER)) on how the tunnels are performing. Other mechanisms such as dedicated Operations, Administration, and Maintenance (OAM) traffic flows and IEEE 802.21 (Media Independent Handover, MIH) messages could be used to gain additional information about the wireless networks (e.g., 'going down' messages) [30,51].

The Monitor provides the Link Prediction with relevant measurement data (e.g., signal quality and messages concerning upcoming changes in link availability) that needs to be processed to predict future link quality for each of the interfaces. Measurements concerning the available bandwidth and the round trip time are given to the Application Interface. The tunnel up/down status, bit error rate (BER), packet loss, round trip time etc are signaled to the Admission Control (see Section 4.3.5).

### 6.4 Tunnel protocol

We propose to use Stream Control Transmission Protocol (SCTP) [52] as the protocol for the Tunnel module (Section 6.2.2). There are different reasons to opt for this protocol: it supports multi-homing, mobility, unreliable traffic flows, multiple traffic flows etc. This is elaborated within the TWCS context below. Note that SCTP also has some other advantages which are not tackled here, e.g., protection against SYN flooding and selective acknowledgments (SACKs).

#### 6.4.1 Mobility support

There are many ways to support mobility for the nodes on board of the train. In this article, we assumed that the network mobility is managed solely by the MCE on board of the train and that the onboard devices are unaware of the fact that they are part of a moving network. Moreover, all third party communicating nodes are also unaware of any mobility of the onboard devices. Therefore, all data traffic is routed to the WCE at the wayside. This way, mobility is transparent in the complete system (which was reflected in the addressing scheme in Figure 9) and it allows to centrally optimize bandwidth capacity, prioritize some data traffic flows etc.

A logical choice would be to use the NEtwork MObility (NEMO) protocol [53], which implies that a router in the mobile network (the Mobile Router, MR) is using Mobile IP (MIP) [54] to connect via a bidirectional tunnel with a Home Agent (HA). In our case, the MCE would act as the MR, which connects to the WCE, which acts as the HA.

MIP-based solutions for TWCS have already been studied [12,22,30,55,56] and extensions to support multi-homing, by using multiple egress interfaces on the MR, have been developed [57-59] within the IETF Working Group MEXT (Mobility EXTensions for IPv6) or in the former Working Groups MIP6 (Mobility for IPv6), NEMO or MONAMI6 (Mobile Nodes and Multiple Interfaces in IPv6). Recently, progress was also made on how to actually spread the data traffic flows over the different interfaces [60,61]. Furthermore, the network prefix that is allocated by the MR to the onboard network does not have to be the same prefix that was allocated on the egress interfaces of the MR, but it can be one or more logical subnet(s) from the network of the INT [62], conform the addressing scheme that was proposed in Section 6.1. This setup is depicted in Figure 10a, where the reliable transport layer connections are indicated by a double arrowed solid black line.

The NEMO solution is a network layer solution and simply tunnels the data traffic flows in an IP-in-IP tunnel. In our scenario however, this solution does not fit our requirements. To ensure an optimized bandwidth consumption, we have included an Accelerator (see Section 5.1.3) which converts TCP data traffic flows into simple UDP datagrams. However, these packets need to be reliably transmitted to the other side of the TWCS. The tunneling mechanism thus needs to support reliable transmission, which is typically a transport layer functionality. We thus need to use a transport layer protocol as a tunneling protocol, which can be regarded as acting on layer 3.5. Another way of looking at this particular situation, is to regard all data traffic flows of the mobile network as the single data traffic flow of a single mobile node (the MCE), rather than as a complete mobile network. While the onboard nodes believe that they have an end-to-end transport layer connection with a wayside node, this connection is actually intercepted at the onboard Accelerator of the MCE, which sets up a UDP connectionless traffic flow to the wayside. All those traffic flows need to be jointly tunneled in a single transport layer connection. With the NEMO solution, shifting a traffic flow from one egress interface to another is done by Mobile IP at the MCE, while retransmissions are managed by the end-to-end TCP connection. However, when using a 3.5 layer transport connection directly between MCE and WCE, this transport connection will need to be persistent during the handover from one interface to another. Standard TCP is incapable of doing so. Therefore, we propose to use SCTP [52] (with its ADDIP extension [63] for mobility), as it supports multi-homing for these handovers. It was developed in the IETF Working Group Sigtran (Signaling Transport) and has been studied to be used within a TWCS context [22,64]. This tunneling solution is depicted in Figure 10b, where the reliable transport layer connections are indicated by a double arrowed solid black line. The virtual transport layer connections, as perceived by the end devices or by the Accelerators, are indicated by a curved dark gray line with unreliable transport connections indicated by a dashed line.

SCTP also allows data to be partitioned into different "streams", so that a message lost in one stream will not stall delivery of message in other streams. In the TWCS scenario, each data traffic flow can be mapped to a different stream.[b] This way, the tunneling transport protocol will not introduce a mutual dependency amongst the data traffic flows during packet loss within a single traffic flow.

A summary of the properties of SCTP and NEMO, within the TWCS context, is given in Table 5.

#### 6.4.2 Tunnel instances

An important choice to be made is how many transport layer tunnels (see Section 6.4.1) we envisage and where they are related to. There are two options to consider: a tunnel per service class or a tunnel per wireless interface. Both options are visualized in Figure 11.

The first option is to implement the tunnels per service class (see Figure 11a). If a service class is supported by multiple wireless links, the tunnel needs to be able to transparently span multiple links, which requires a multi-path protocol to simultaneously send data over multiple links, e.g., Multi-Path TCP (MP-TCP) [65] or MMP-SCTP (Mobile Multi-Path Stream Control Transmission Protocol) [66]. This option does not require to implement any scheduling over the different links, as the tunneling protocol takes care of this. However, this also results in some significant drawbacks. Firstly, packets that belong to the same data traffic flow could be spread over multiple links, increasing the incurred jitter for this data traffic flow. Secondly, paths of different tunnels can run over the same link, which leads to contending congestion mechanisms over this link, resulting in a suboptimal usage of the link capacity.

The second option, and the one that is chosen within this architecture, is to implement a tunnel per link (see Figure 11b). In this case, the tunneling protocol only actively sends data traffic over one link, eliminating the need for a multi-path protocol. However, reliable transport is still required in the case of a link failure and packets that were lost need to be retransmitted over another link. Therefore, multi-homing support is required to enable persistency of the transport connection that acts as the tunnel between MCE and WCE. The SCTP protocol can provide this functionality, see section Section 6.4.3.

When using a tunnel instance per link, one will need to implement an external scheduling algorithm to distribute the different data traffic flows (with different service classes) over the different links. Within this architecture, this is the responsibility of the Scheduler module, see Section 4.2.4 and depicted in Figure 3. The advantage of this solution is that one now has full control to avoid spreading one traffic flow over multiple wireless links. Furthermore, there is only one tunnel transport connection per link, which avoids competing congestion mechanisms. A comparison between the two options is given in Table 6.

An additional advantage of having a tunnel instance per link, is the ability to optimize data transmission for this specific link. Depending on the link characteristics, performance could be improved by e.g., aggregating small packets, adapt window sizes and counter expirations of the sending algorithm and of the congestion control etc.

#### 6.4.3 Packet retransmission upon link failure

As stated in Section 6.4.2, we need a multi-homing protocol for the transport connections that act as tunnels per wireless interface. When a link goes down, the packets that were lost over this link can then be retransmitted over another link within the same transport connection.

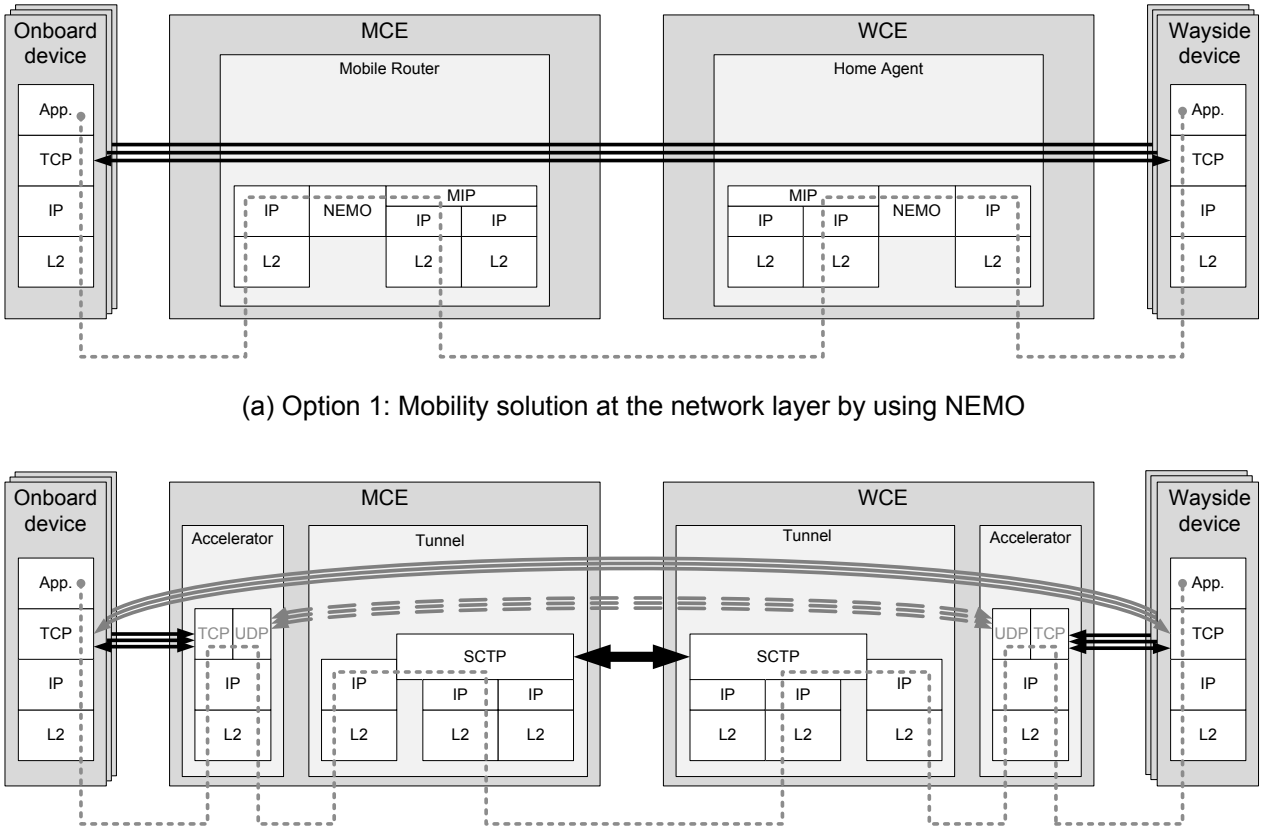



**Figure 10 Comparison of two network mobility solution architectures within the TWCS context**. The actual reliable transport layer connections are indicated by a double arrowed solid black line, while the perceived reliable (unreliable) transport layer connections are indicated in a solid (dashed) grey line. The packet flows through different OSI layers is indicated in a dotted line.

SCTP provides a suitable solution, as it was initially designed with multi-homing built-in.

When the link goes down, the packets that need reliable transport and that were not yet acknowledged will be retransmitted via the original SCTP association over any of the other links. They will thus be sent in parallel with the traffic from another SCTP association that uses this link as its primary path. Meanwhile, the Scheduler should no longer schedule traffic flows on the link that went down. This way, new data traffic is no longer sent via the SCTP association of the link that went down. This setup is visualized in Figure 12.

**Table 5 Comparison of SCTP and NEMO within the TWCS context**

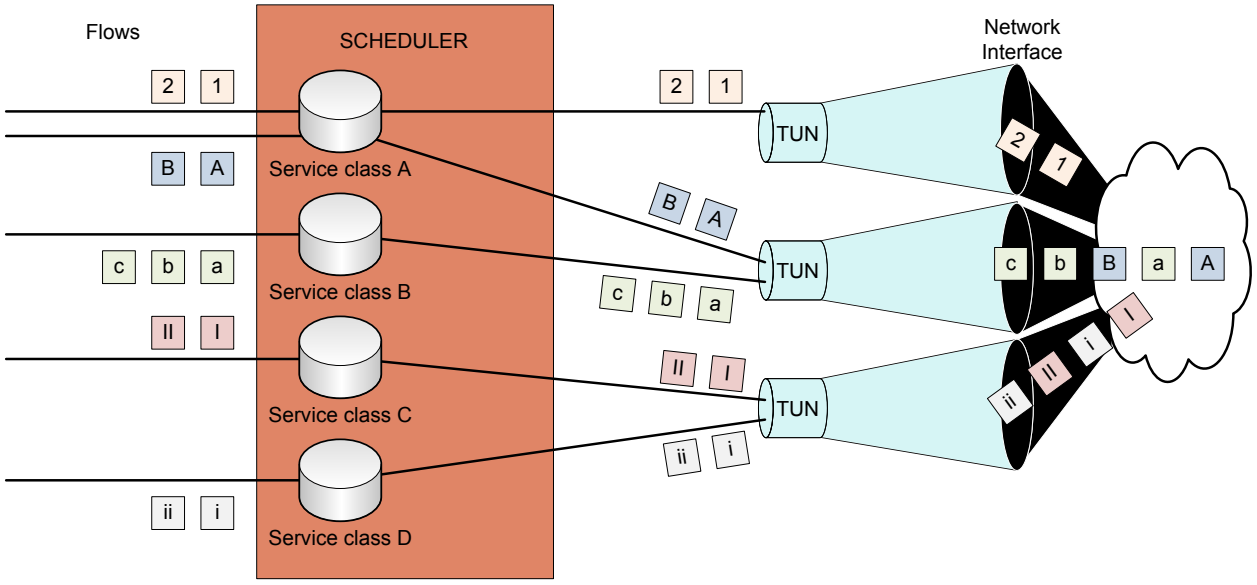
| | SCTP | NEMO |
|---|---|---|
| OSI layer | L4 | L3 |
| Tunnel per | Socket | Interface |
| Mobility | Yes* | Yes |
| Multi-homing | Yes | Yes* |
| Reliable transport | Yes | No |
| Unreliable transport | Yes* | Yes |

(* supported via protocol extensions)

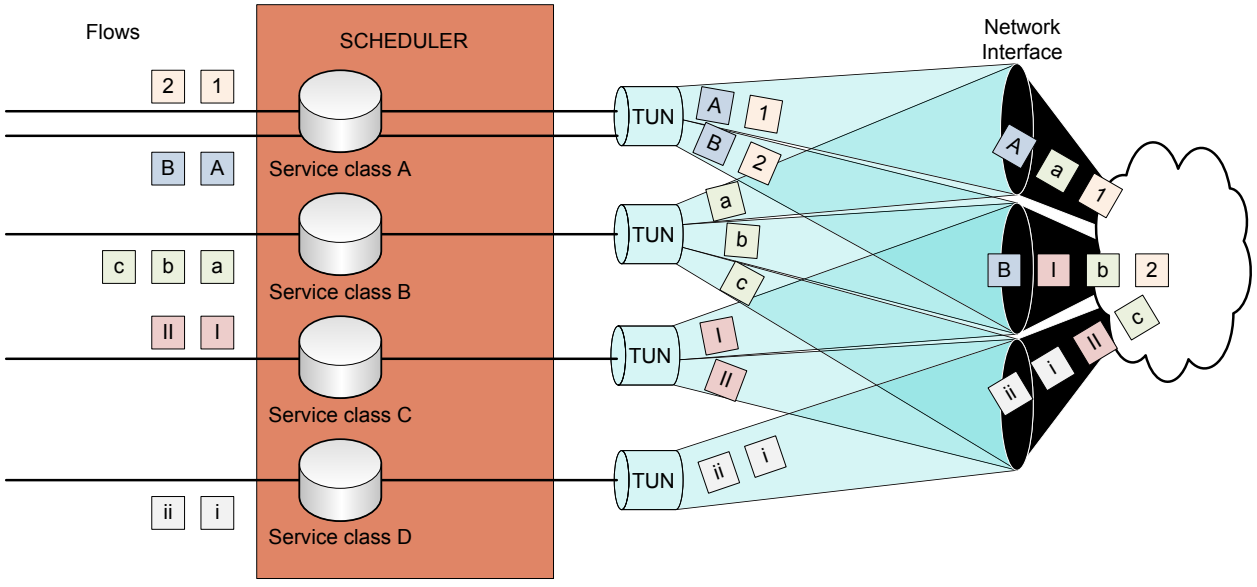
#### *6.4.4 Support for unreliable transport*

So far, we have focused on tunnels for reliable transport, which is required for e.g., the data between the two Accelerator endpoints (see Section 5.1.3). However, retransmissions are irrelevant and therefore a waste of bandwidth for other data traffic flows. Therefore, a second tunnel should run in parallel to the tunnel for reliable transport or the tunneling protocol should support both reliable and unreliable transport, as depicted in Figure 3.

The unreliable tunnel also needs to incorporate a congestion control mechanism (in order not to fully occupy the link and push the reliable tunnel out). It thus needs to be a so called network-friendly protocol, which does not guarantee reliability, but throttles itself according to




(a) Option 1: a tunnel per service class. This allows for an easy mapping of service classes to tunnel instances, but the tunnels are more complex and traffic flows can be spread over multiple network interfaces.




(b) Option 2 (preferred): a tunnel per network interface. This requires more effort for mapping service classes to tunnel instances, but the tunnels are more simple and one can avoid spreading a traffic flow over multiple network interfaces.

**Figure 11** Two options to implement tunnels between MCE and WCE for mobility support.

the current network situation. Simple UDP encapsulation, which offers no congestion control, would thus be insufficient for the unreliable tunnel. Datagram Congestion Control Protocol (DCCP) is such a protocol [67,68], included in the mainline Linux kernel [69,70] and IPv6 compatible.

Rather than using separate protocols for the reliable and unreliable tunnels per link, one could also opt to

**Table 6 Comparison of the two options to implement tunnels between MCE and WCE for mobility support as depicted in Figure 11**

| | Option 1 | Option 2 (preferred) |
|---|---|---|
| Tunnel socket per | Service Class | Network interface |
| Scheduling by | Tunneling protocol | Scheduler module |
| Links per Multiple active socket | links (multi-path) | One link + back-up links (multi-homing) |
| Contention per link | Yes | Only during link failure |
| Data traffc fow over multiple links | Yes (risk for jitter) | No |
| Protocol suggestions | MP-TCP | SCTP |
| | MMP-SCTP | |

use a single protocol to do both per link. This also typically means that only one congestion control mechanism is used for both tunnels, which is expected to maximize bandwidth utilization, compared to two competing mechanisms. Here again, SCTP is an eligible protocol. SCTP has an extension, called PR-SCTP [71], which enables some parts of a SCTP stream to be sent unreliably.

### 6.5 Implementation challenges & suggested solutions

In this section, we discuss various implementation issues, concerning Network Mobility, that might occur.

#### *6.5.1 Congestion control over a wireless link*

For a link with a lot of bandwidth or delay fluctuations, it is not beneficiary for the tunnels to use 'slow-start' mechanisms as the bandwidth fluctuation is mistaken for congestion. SCTP has unfortunately a poor congestion control mechanism in its current form in the Linux kernel when used over wireless links. It could be adapted similar to e.g., the TCP Hybla [72] congestion algorithm, which is designed with wireless links in mind and performs adequately over such links.

#### *6.5.2 Refragmentation*

There are no issues with fragmentation. Firstly, SCTP itself will fragment packets that are too large to send in single IP packet. Therefore, it will use the smallest maximum transmission unit (MTU) of all available links (the Path MTU, PMTU) to avoid IP fragmentation when retransmitting over another link. However, once a message is fragmented by SCTP, it cannot be refragmented [52]. If the PMTU decreases when a new wireless link is enabled, the previously created SCTP packets that are (re)transmitted over this link will be too large, but they will be fragmented on the IP layer.

#### *6.5.3 Discrepancy between MCE and WCE*

As the WCE has no direct access to the wireless network interfaces and only the MCE knows the availability of the wireless links, the initiative for setting up tunnels will always be taken on the MCE side.

While the MCE knows which tunnels it is managing, the WCE does not know which tunnels are belonging to the same MCE. Therefore, the MCE needs to signal to the WCE which tunnels are belonging to the same 'association', i.e., originating from the same train. This involves the exchange of control messages over the data plane towards the other side. This signaling is included in SCTP [52] and its ADDIP [63] extension.

The WCE has no view on the availability and capacity of the wireless links that are part of its connection with the MCE. The WCE's wired interface could continuously pull data to be sent, as the first wired links of its connection towards the trains will be quite fast in comparison with the last wireless links. The WCE will therefore need to tune its sending rate to the buffer size of the tunneling protocol(s) instead of using the data rate of its wired Ethernet interface. The MCE could also communicate the current available capacity on the wireless links to the WCE, but this mechanism will be too slow to react to changing conditions.

#### *6.5.4 Tunnel overhead*

Small packets can cause a lot of overhead, especially when in tunnels, where each packet will get an additional tunnel header. Aggregating a number of small packets in a larger packet can give some performance gain. SCTP allows to bundle multiple 'chunks' into a larger packet. However, this results in some extra delay, since an aggregated packet will have to be composed before it can be sent. The appropriate caution should be taken to ensure that this happens only with traffic flows which do not need timely delivery.

Another option to reduce the amount of header overhead, is to apply header compression. For header compression, RObust Header Compression (ROHC) [73,74] is very well known and free libraries exist [75].

## 7 Example packet traffic flows

Within this section we illustrate the modifications in the packet header due to the complete processing of a packet in this architecture. Firstly, we give an overview of the OSI layers that the different components, as described in the previous sections, are acting on in Section 7.1. Next we show the header modifications for a UDP and TCP traffic flow in Sections 7.2 and 7.3, respectively.

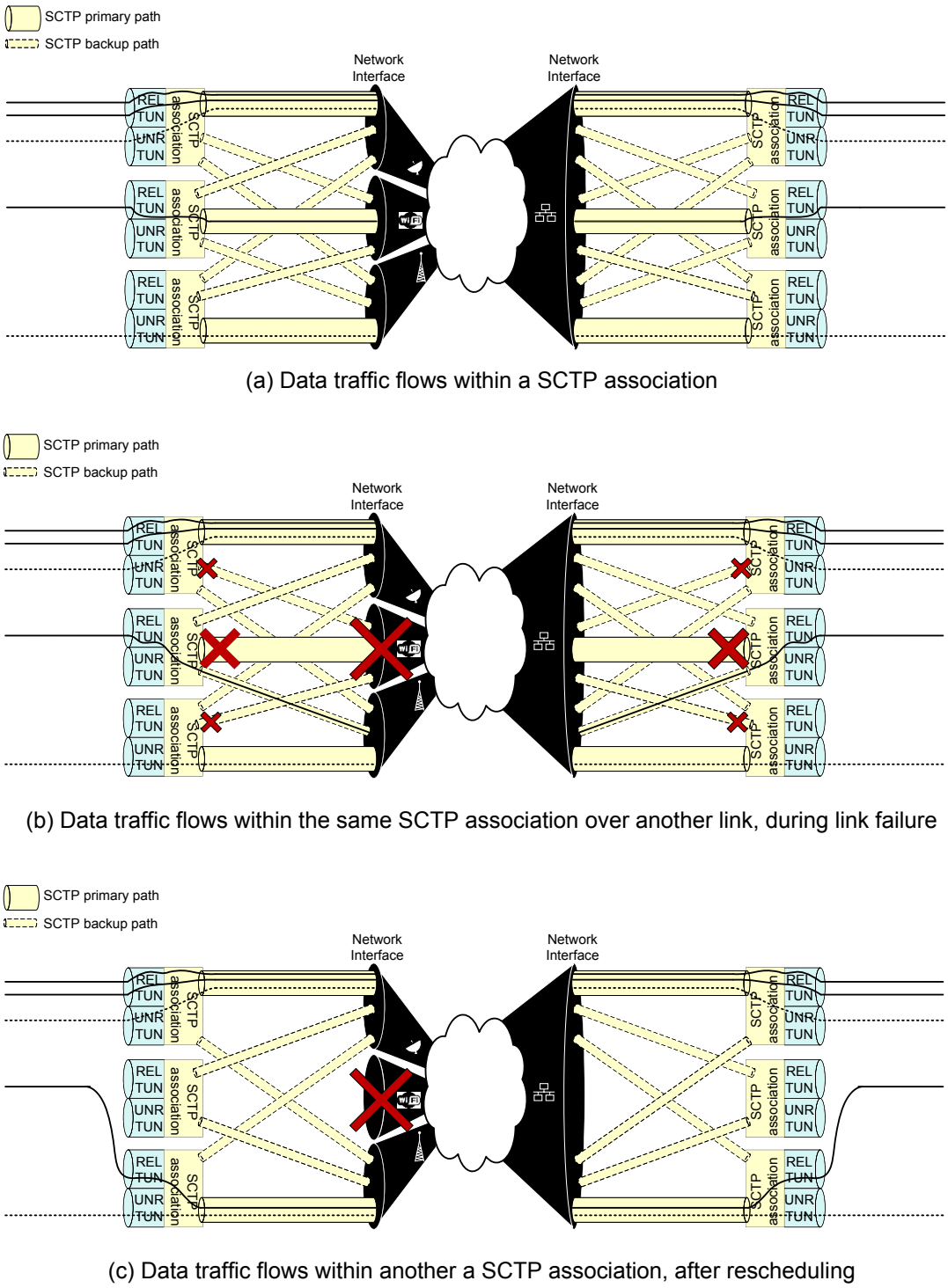



**Figure 12 Using SCTP for reliable and unreliable traffic flows per link and for retransmission over other links.**

### 7.1 OSI layers

In Sections 4.2, 5.1, and 6.2 we have described the functionality of the modules that are acting on the data plane. This functionality is situated on different levels of the OSI [44] model. An overview thereof is given for unreliable and reliable traffic flows in Figure 13.

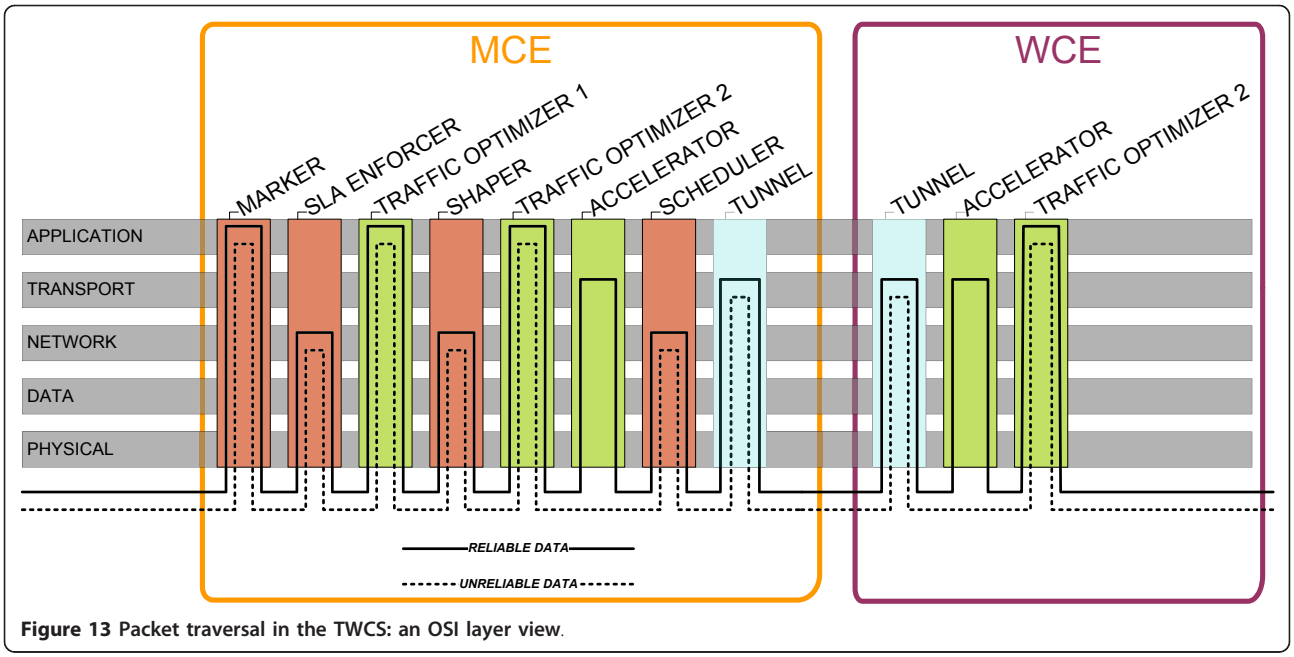



**Figure 13** Packet traversal in the TWCS: an OSI layer view.

### 7.2 UDP traffic flow

For an unreliable UDP traffic flow, we show how its packets will pass the different modules (on the data plane) and how the packet headers will change accordingly in Figure 14.

Suppose a packet, originating from an onboard device with IP address 2001:db8:1:1::33 (see Figure 9) is destined for a wayside server with IP address 2001:db8:a::a. The packet first needs to be routed to the onboard standard gateway, which is the MCE with IP address 2001:db8:1:1::1 (see Figure 9). The onboard device knows this via fixed configuration or via e.g., DHCPv6 (Dynamic Host Configuration Protocol). The onboard device then resolves the link layer address for 2001:db8:1:1::1 (via neighbor discovery [76]), and the packet is sent over the onboard network (e.g., via Wi-Fi or Ethernet) to the MCE.

When the packet arrives at the MCE, the Marker will assign a value to the DSCP bits of the Traffic Class and to the Flow Label field (see Figure 5). Suppose the packet has not been dropped by the Shaper or the SLA Enforcer, it will eventually arrive at the tunnel for unreliable transport. The tunnel will encapsulate this packet as payload in its tunneling protocol (with SCTP, see Section 6.4). The source IP address in the outer IP header of the new packet is the IP address of the wireless interface, which is here assumed 2001:db8:ffff::1 (see Figure 9) and the destination address is the IP address of the WCE, which is here assumed 2001:db8:2::2 (see Figure 9).

This packet will then be routed via the tunnel to the WCE, where the Tunnel module will decapsulate the packet, and the original UDP packet reappears in the processing thread, which is forwarded to its final destination at a RST or on the Internet (see Figure 1).

### 7.3 TCP traffic flow

For a reliable TCP traffic flow, we show how its packets will pass the different modules (on the data plane) and how the packet headers will change accordingly in Figure 15.

Suppose a packet, originating from an onboard device with IP address 2001:db8:1:1::33 (see Figure 9) is destined for a wayside server with IP address 2001:db8:a::a. Just as in the previous scenario, the packet first needs to be routed to the onboard standard gateway, which is the MCE with IP address 2001:db8:1:1::1 (see Figure 9). The onboard device knows this via fixed configuration or via e.g., DHCPv6. The onboard device then resolves the link layer address for 2001:db8:1:1::1 (via neighbor discovery [76]), and the packet is sent over the onboard network (e.g., via Wi-Fi or Ethernet) to the MCE.

When the packet arrives at the MCE, the Marker will assign a value to the Traffic Class and the Flow Label field (see Figure 5). When the packet arrives at the Accelerator, an additional change in the packet header occurs, which was not applicable for unreliable traffic flows (Section 7.2). The Accelerator will send a TCP ACK to the onboard device (TCP ACK spoofing, see Section 5.1.3) and will convert the TCP payload into raw data, encapsulated in e.g., UDP, in order to terminate the TCP mechanisms of each independent traffic flow here.

Next, the packet is sent over a tunnel for reliable transport (with SCTP, see Section 6.4) to the WCE. Therefore, the packet is encapsulated into another packet, similar to what happens for unreliable traffic

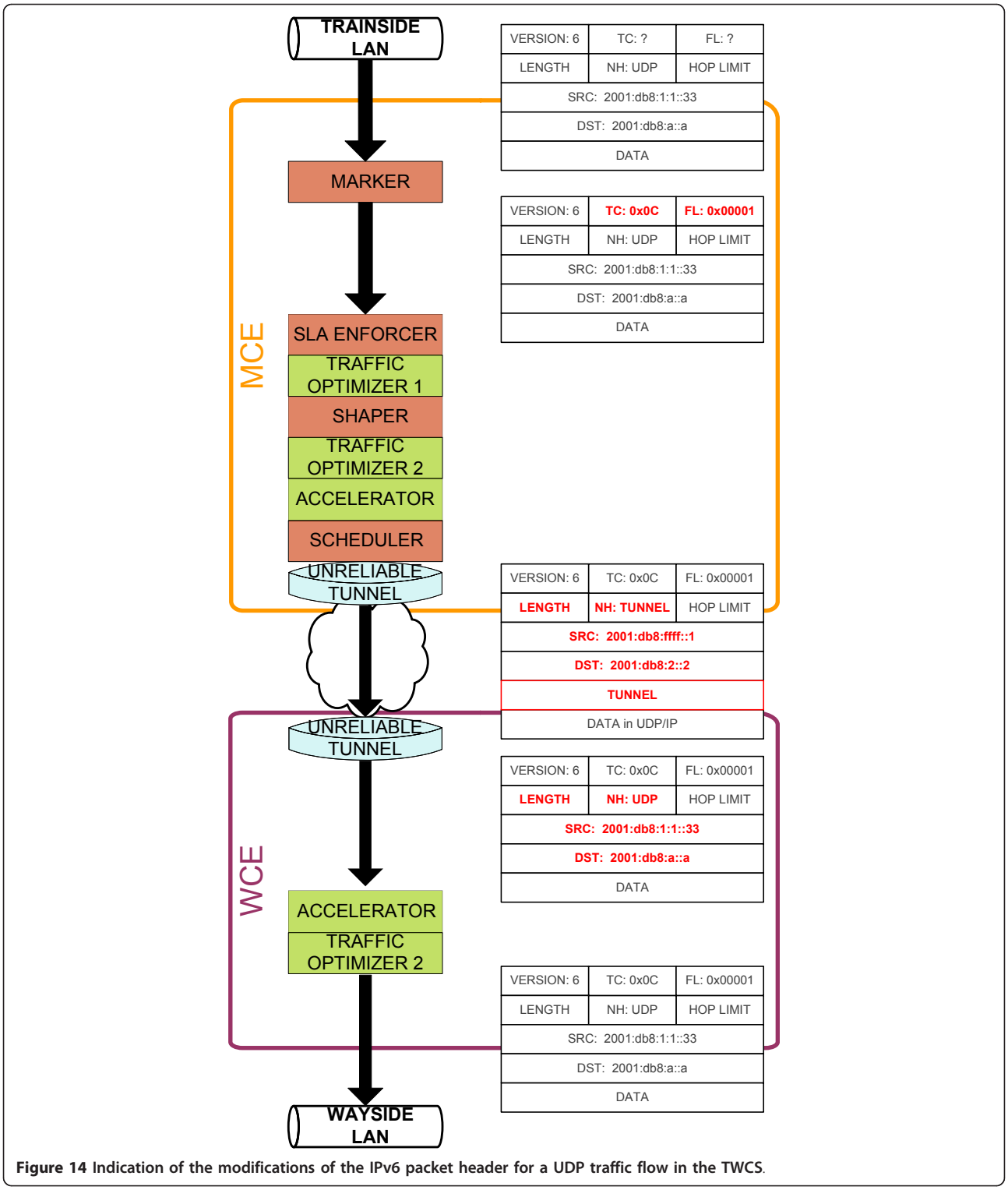



**Figure 14** Indication of the modifications of the IPv6 packet header for a UDP traffic flow in the TWCS.

flows. The source IP address of the new packet is the IP address of the wireless interface, which is here assumed 2001:db8:ffff::1 (see Figure 9) and the destination address is the IP address of the WCE, which is here assumed 2001:db8:2::2 (see Figure 9).

In the WCE, the packet is decapsulated, and the receiving Accelerator will convert the packet into the original TCP packet and use a local TCP socket to rebuild the TCP mechanism and to forward the packet to its final destination at a RST or on the Internet (see

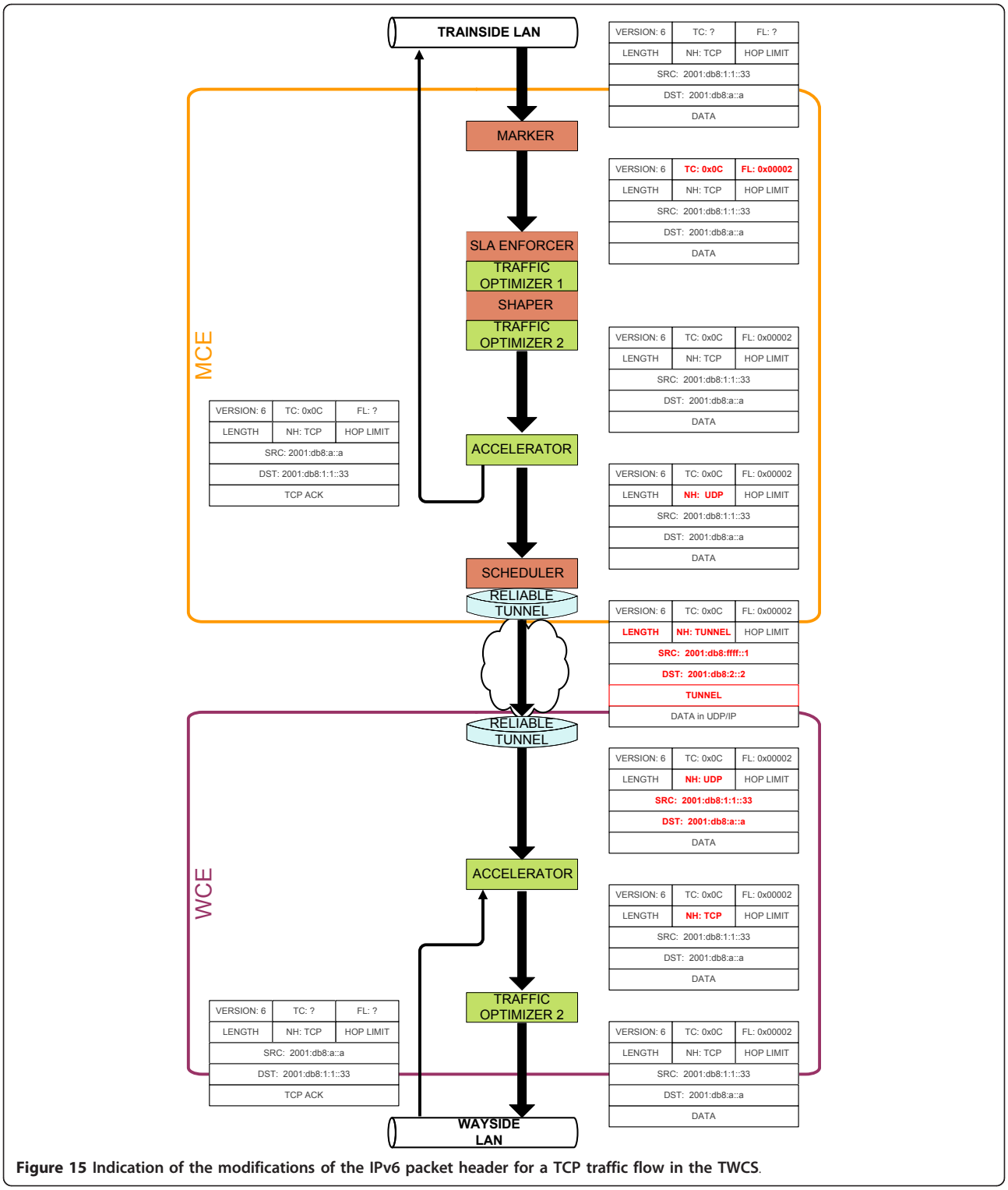



**Figure 15** Indication of the modifications of the IPv6 packet header for a TCP traffic flow in the TWCS.

Figure 1). TCP ACKs that are received as a reply to this packet, will be discarded by the Accelerator at the WCE, as the packet was already acknowledged by the transmitting Accelerator at the MCE.

## 8 Conclusion

We designed a novel and modular IPv6-enabled TWCS architecture to jointly tackle QoS, overall bandwidth optimizations and the network mobility in TWCS.

*Concerning provision of QoS* in the TWCS, the functionality is logically split into the Marker, SLA Enforcer, Shaper and Scheduler. Firstly, the Marker marks packet traffic flows with a service class and priority by using the DiffServ architecture, according to the different services and their traffic flow characteristics. Next, the SLA Enforcer ensures that all traffic flows that belong to the same SLA comply to the SLA stipulations (e.g., maximum data rate, data volume). Then, the Shaper shapes all traffic flows to the available capacity on the wireless T2W link by dropping packets of traffic flows, with respect to the relative priority of the different traffic flows. Finally, the Scheduler needs to schedule all traffic flows on an appropriate link, considering the service class of each traffic flow (e.g., low latency requirement for Voice-over-IP (VoIP)). A 'backpressure' mechanism, based on queue occupation, is suggested for signaling the available capacity from the Scheduler to the Shaper.

For *overall bandwidth optimization*, a PEP is inserted in the design, which consists of different modules: TOs and an Accelerator. The TOs include caching proxies and data compression. The Accelerator locally intercepts TCP connections to mitigate performance degradation over high latency links. We propose a distributed design for the Accelerator (one component on board and one at the wayside), in order to avoid multiple competing TCP mechanisms over the same link. To maintain SLAs and fairness among the onboard devices, we furthermore paid much attention to the correct order of the different QoS and PEP components.

For the *network mobility* aspect, we propose to use the Stream Control Transmission Protocol (SCTP) protocol (with extensions for mobility and partial reliability) to create a tunnel per wireless link. This way, the traffic between the Accelerators can be put in a single reliable connection per link, data traffic flows can be mapped to SCTP streams that do not stall each other, UDP traffic flows can be sent without reliability but with congestion control, and multi-homing allows packets to be retransmitted over another link in the case of a link failure.

By considering all of these aspects together, we are able to indicate the mutual dependencies and relationships between the different functionalities, as well as the possible implementation issues. This led us to propose this fully featured and integrated TWCS design.

## Endnotes

[a]We use the 2001:db8::/32 IPv6 address prefix as this is the prefix that is reserved for documentation purposes [77].

[b]A stream identifier is 16bits, so 65,536 data traffic flows can be mapped on a unique stream id.

**Acknowledgements**
This research was carried out as part of the IBBT TRACK ICON project. Daan Pareit would like to thank the IWT-Vlaanderen (Institute for the Promotion of Innovation through Science and Technology in Flanders) for financial support through his Ph.D. grant.

**Author details**
[1]Department of Information Technology, Ghent University-IBBT, Gaston Crommenlaan 8 box 201, 9050 Ghent, Belgium [2]Department of Mathematics and Computer Science, University of Antwerp - IBBT, Middelheimlaan 1, 2000 Antwerp, Belgium [3]Newtec Cy, Laarstraat 5, 9100 Sint-Niklaas, Belgium [4]Nokia Siemens Networks, Atealaan 34a, 2200 Herentals, Belgium [5]Bombardier Transportation, Vaartdijkstraat 5, 8200 Brugge, Belgium

**Competing interests**
The authors declare that they have no competing interests.

Received: 29 April 2011 Accepted: 22 March 2012
Published: 22 March 2012

**References**

1. H Forbes, A Radio Inter-Communicating System for Railroad Train Service. *Proceedings of the Institute of Radio Engineers*. **15**(10), 869–878 (1927)
2. N Monk, S Wright, Technical Aspects of Experimental Public Telephone Service on Railroad Trains. *Proceedings of the IRE*. **36**(9), 1146–1152 (1948)
3. N Monk, Experimental Radio-Telephone Service for Train Passengers. *Proceedings of the IRE*. **39**(8), 873–881 (1951)
4. F Abrishamkar, J Irvine, Comparison of current solutions for the provision of voice services to passengers on high speed trains, in *IEEE VTS-Fall 52nd Vehicular Technology Conference*, Volume **4**, 1498–1505 (2000)
5. S Scalise, Summary of Conclusions and Outlook ESA ARTES 1 project Internet to Trains Initiative presentation, ESA-ESTEC, Noordwijk NL, (2008)
6. ERTMS. Website (2011) http://www.ertms.com/
7. V Riihimaki, T Vaaramaki, J Vartiainen, T Korhonen, Techno-economical inspection of high-speed internet connection for trains. *IET Intelligent Transport Systems*. **2**, 27–37 (2008). doi:10.1049/iet-its:20070014
8. B Lannoo, J Van Ooteghem, D Pareit, T Van Leeuwen, D Colle, I Moerman, P Demeester, Business model for broadband internet on the train. *The Journal of The Institute of Telecommunications Professionals*. **1**, 19–27 (2007)
9. W van Brussel, Bringing ICT services to trains technical and economical challenges, in *9th Conference on Telecommunications Internet and Media Techno Economics (CTTE)*, 1–7 (2010)
10. B Jooris, P Verhoeve, F Vermeulen, I Moerman, Mobile communication & service continuity in a train scenario, in *Proceedings of the 12th Annual Symposium of the IEEE/CVT Symposium on Communications and Vehicular Technology in the Benelux (SCVT2005)* (2005)
11. B Jooris, A Schoutteet, F Vermeulen, I Moerman, Access network controlled fast handoff for streaming multimedia in WLAN, in *16th IST Mobile and Wireless Communications Summit*, 1–5 (2007)
12. X Liang, F Ong, P Chan, R Sheriff, P Conforto, Mobile Internet access for high-speed trains via heterogeneous networks, in *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications. PIMRC 2003*. **1**, 177–181 (2003)
13. F De Greve, F Van Quickenborne, F De Turck, I Moerman, P Demeester, Rapidly recovering Ethernet networks for delivering broadband services on the train, in *The IEEE Conference on Local Computer Networks*, 294–302 (2005)
14. F Van Quickenborne, F De Greve, F De Turck, I Moerman, P Demeester, I Chong, K Kawahara, A Tunnel-Based QoS Management Framework for Delivering Broadband Internet on Trains, in *Information Networking. Advances in Data Communications and Wireless Networks, Volume 3961 of Lecture Notes in Computer Science*, (Springer Berlin / Heidelberg, 2006), pp. 552–561
15. B Lannoo, D Colle, M Pickavet, P Demeester, Comparison of two optical switching architectures to provide a broadband connection to train passengers, in *Optical Fiber Communication Conference and the 2006 National Fiber Optic Engineers Conference. OFC 2006*, 3 (2006)
16. B Lannoo, D Colle, M Pickavet, P Demeester, Radio-over-fiber-based solution to provide broadband internet access to train passengers [Topics in Optical Communications]. *IEEE Communications Magazine*. **45**(2), 56–62 (2007)

17. F van Quickenborne, F de Turck, P Demeester, Advanced Multimedia Services for Fast Moving Users on Trains, in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks. WoWMoM 2007*, 1–8 (2007)
18. D Fokum, V Frost, A Survey on Methods for Broadband Internet Access on Trains. *IEEE Communications Surveys Tutorials*. **12**(2), 171–185 (2010)
19. KD Lin, JF Chang, Communications and entertainment onboard a high-speed public transport system. *IEEE Wireless Communications*. **9**, 84–89 (2002)
20. F Zou, S Zhang, An Open Onboard Internet System Model for the M-Commerce on Train, in *International Conference on Management and Service Science. MASS '09*, 1–5 (2009)
21. V Schena, F Ceprani, FIFTH Project solutions demonstrating new satellite broadband communication system for high speed train, in *IEEE 59th Vehicular Technology Conference, 2004. VTC 2004-Spring*. **5**, 2831–2835 (2004)
22. D Pareit, N Gheysens, T Van Leeuwen, I Moerman, W Van Brussel, W Torfs, P De Cleyn, C Blondia, QoS-enabled Internet-on-train network architecture: inter-working by MMP-SCTP versus MIP, in *7th International Conference on ITS Telecommunications. ITST '07*, 1–6 (2007)
23. J Conti, Hot spots on rails. *Communications Engineer*. **3**(5), 18–21 (2005). doi:10.1049/ce:20050502
24. M Karlsson, M Bergek, M Agervald, K Axelsson, A system for data transmission via several communication routes (2005) http://www.freepatentsonline.com/EP1175757B1.html
25. S Deering, R Hinden, in Internet Protocol, Version 6 (IPv6) Specification, (Network Working Group, 1998) http://tools.ietf.org/html/rfc2460
26. PF Weston, C Roberts, CJ Goodman, CS Ling, Condition Monitoring of Railway Track using In-Service Trains, in *International Conference on Railway Condition Monitoring*, 26–31 (2006)
27. B Matuz, G Liva, C Niebla, N Diaz, S Scalise, P Kim, DI Chang, HJ Lee, Link Layer Coding for DVB-S2 Interactive Satellite Services to Trains, in *IEEE Vehicular Technology Conference. VTC Spring 2008*, 2922–2926 (2008)
28. G Ohta, F Kamada, N Teramura, H Hojo, 5 GHz W-LAN verification for public mobile applications - Internet newspaper on train and advanced ambulance car, in *First IEEE Consumer Communications and Networking Conference. CCNC 2004*, 569–574 (2004)
29. M Hempel, H Sharif, T Zhou, P Mahasukhon, A wireless test bed for mobile 802.11 and beyond, in *Proceedings of the 2006 international conference on Wireless communications and mobile computing*, (New York, NY, USA: ACM, 2006), 1003–1008. IWCMC '06
30. K Kumar, P Angolkar, D Das, R Ramalingam, SWiFT: A Novel Architecture for Seamless Wireless Internet for Fast Trains, in *IEEE Vehicular Technology Conference. VTC Spring 2008*, 3011–3015 (2008)
31. M Aguado, O Onandi, P Agustin, M Higuero, E Jacob Taquet, WiMax on Rails. *IEEE Vehicular Technology Magazine, IEEE*. **3**(3), 47–56 (2008)
32. L Verstrepen, W Joseph, E Tanghe, J Van Ooteghem, B Lannoo, M Pickavet, L Martens, P Demeester, Making a well-founded choice of the wireless technology for train-to-wayside data services, in *9th Conference on Telecommunications Internet and Media Techno Economics (CTTE)*, 1–7 (2010)
33. K Nichols, S Blake, F Baker, D Black, in Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, Network Working Group, (1998), http://tools.ietf.org/html/rfc2474
34. J Babiarz, K Chan, F Baker, in Configuration Guidelines for DiffServ Service Classes, Network Working Group, (2006), http://tools.ietf.org/html/rfc4594
35. ITU-T, End-user multimedia QoS categories, (ITU, 2001). Series G: Transmission systems and media; digital systems and networks G.1010
36. S Blake, D Black, M Carlson, E Davies, Z Wang, W Weiss, in An Architecture for Differentiated Services, Network Working Group, (1998), http://tools.ietf.org/html/rfc2475
37. IBBT, Train Applications over an advanced Communication Network. Website (2011), http://www.ibbt.be/en/projects/overview-projects/p/detail/track-2
38. N Brownlee, C Mills, G Ruth, in Traffic Flow Measurement: Architecture, Network Working Group, (1999), http://tools.ietf.org/html/rfc2722
39. IANA, Differentiated Services Field Codepoints. Website (2010), http://www.iana.org/assignments/dscp-registry/dscp-registry.xml
40. E Kohler, The Click Modular Router, *PhD thesis*, (Massachusetts Institute of Technology, 2001), http://pdos.csail.mit.edu/papers/click:kohler-phd/thesis.pdf
41. About the netfilter/iptables project. Website (2011), http://www.netfilter.org/
42. IEEE Standard for Local and metropolitan area networks- Virtual Bridged Local Area Networks, Park Avenue, New York, (2006), http://standards.ieee.org/getieee802/download/802.1Q-2005.pdf
43. MA Brown, Traffic Control HOWTO. Website (2005), http://linux-ip.net/articles/Traffic-Control-HOWTO/
44. H Zimmermann, OSI Reference Model-The ISO Model of Architecture for Open Systems Interconnection. *IEEE Transactions on Communications*. **28**(4), 425–432 (1980). doi:10.1109/TCOM.1980.1094702
45. squid. *squid: Optimising Web Delivery* (2011), http://www.squid-cache.org/
46. BIND – Internet Systems Consortium. *Website* (2011), http://www.isc.org/software/bind
47. Open Source - Sendmail.com. *Website* (2011), http://www.sendmail.com/sm/open_source/
48. S Ramachandran, Web metrics: Size and number of resources, Google, (2010), http://code.google.com/speed/articles/web-metrics.html
49. Ziproxy homepage. *Website* (2011), http://ziproxy.sourceforge.net/
50. SourceForge.net: PEPsal - Project Web Hosting - Open Source Software. *Website* (2011), http://pepsal.sourceforge.net/
51. IEEE Standard for Local and Metropolitan Area Networks: Part 21: Media Independent Handover Services, Park Avenue, New York, (2009), http://standards.ieee.org/getieee802/download/802.21-2008.pdf
52. R Stewart, Stream Control Transmission Protocol, Network Working Group, (2007), http://tools.ietf.org/html/rfc4960
53. V Devarapalli, R Wakikawa, A Petrescu, P Thubert, in Network Mobility (NEMO) Basic Support Protocol, Network Working Group, (2005), http://tools.ietf.org/html/rfc3963
54. D Johnson, C Perkins, J Arkko, Mobility Support in IPv6, Network Working Group, (2004), http://tools.ietf.org/html/rfc3775
55. F Zou, X Jiang, Z Lin, IEEE 802.20 Based Broadband Railroad Digital Network - The Infrastructure for M-Commerce on the Train, in *Proceedings of 4th International Conference on Electronic Business (ICEB2004)* (2004)
56. E Hernandez, A Helal, Examining Mobile-IP performance in rapidly mobile environments: the case of a commuter train, in *26th Annual IEEE Conference on Local Computer Networks. LCN 2001*, 365–372 (2001)
57. J Abley, B Black, V Gill, in Goals for IPv6 Site-Multihoming Architectures, Network Working Group, (2003), http://tools.ietf.org/html/rfc3582
58. C Ng, T Ernst, E Paik, M Bagnulo, Analysis of Multihoming in Network Mobility Support, Network Working Group, (2007), http://tools.ietf.org/html/rfc4980
59. R Wakikawa, V Devarapalli, G Tsirtsis, T Ernst, K Nagami, Multiple Care-of Addresses Registration, Network Working Group, (2009), http://tools.ietf.org/html/rfc5648
60. G Tsirtsis, G Giaretta, H Soliman, N Montavont, Traffic Selectors for Flow Bindings, Network Working Group, (2011), http://tools.ietf.org/html/rfc6088
61. G Tsirtsis, H Soliman, N Montavont, G Giaretta, K Kuladinithi, Flow Bindings in Mobile IPv6 and Network Mobility (NEMO) Basic Support, Network Working Group, (2011), http://tools.ietf.org/html/rfc6089
62. R Droms, P Thubert, F Dupont, W Haddad, C Bernardos, DHCPv6 Prefix Delegation for NEMO, *Internet-Draft*, Network Working Group, (2010), http://tools.ietf.org/html/draft-ietf-mext-nemo-pd-07
63. R Stewart, Q Xie, M Tuexen, S Maruyama, M Kozuka, in Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration, Network Working Group, (2007), http://tools.ietf.org/html/rfc5061
64. D Pareit, I Moerman, P Demeester, W Torfs, P De Cleyn, C Blondia, SCTP as mobility protocol for enhancing internet on the train, in *8th International Conference on ITS Telecommunications. ITST 2008*, 323–327 (2008)
65. A Ford, C Raiciu, M Handley, S Barre, J Iyengar, Architectural Guidelines for Multipath TCP Development, Network Working Group, (2011), http://tools.ietf.org/html/rfc6182
66. CM Huang, CH Tsai, Mobile Multi-path Transmission using SCTP, Network Working Group, (2006), http://tools.ietf.org/id/draft-huang-tsai-mmp-sctp-00.txt
67. E Kohler, M Handley, S Floyd, Datagram Congestion Control Protocol (DCCP), Network Working Group, http://tools.ietf.org/html/rfc4340
68. E Kohler, Generalized Connections in the Datagram Congestion Control Protocol, in Internet-Draft, Internet Engineering Task Force, (2006), http://tools.ietf.org/html/draft-kohler-dccp-mobility-02
69. The Linux Foundation, DCCP. *Website* (2009), http://www.linuxfoundation.org/collaborate/workgroups/networking/dccp
70. DCCP-TP. *Website* (2008), http://www.phelan-4.com/dccp-tp/tiki-index.php

71. R Stewart, M Ramalho, Q Xie, M Tuexen, P Conrad, Stream Control Transmission Protocol (SCTP) Partial Reliability Extension, Network Working Group, (2004), http://tools.ietf.org/html/rfc3758
72. CC N, R Firrincieli, TCP hybla: a TCP enhancement for heterogeneous networks. *International Journal of Satellite Communications and Networking*. **22**, 547–566 (2004). doi:10.1002/sat.799
73. C Bormann, C Burmeister, M Degermark, H Fukushima, H Hannu, LE Jonsson, R Hakenberg, T Koren, K Le, Z Liu, A Martensson, A Miyazaki, K Svanbro, T Wiebe, T Yoshimura, H Zheng, Robust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed, Network Working Group, (2001), http://tools.ietf.org/html/rfc3095
74. K Sandlund, G Pelletier, LE Jonsson, in The RObust Header Compression (ROHC) Framework, Network Working Group, (2010), http://tools.ietf.org/html/rfc5795
75. RObust Header Compression (ROHC) library in Launchpad. Website (2011), https://launchpad.net/rohc
76. T Narten, E Nordmark, W Simpson, H Soliman, Neighbor Discovery for IP version 6 (IPv6), Network Working Group, (2007), http://tools.ietf.org/html/rfc4861
77. G Huston, A Lord, P Smith, IPv6 Address Prefix Reserved for Documentation, Network Working Group, (2004), http://tools.ietf.org/html/rfc3849
78. ETSI, Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services (1997)
79. ETSI, Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Overall description of the GPRS radio interface; Stage 2 (2010), http://pda.etsi.org/exchangefolder/ts_143064v090000p.pdf
80. IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Park Avenue, New York, (2007) http://standards.ieee.org/getieee802/download/802.11-2007.pdf
81. ETSI, Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (2006)
82. ETSI, Digital Video Broadcasting (DVB) User guidelines for the second generation system for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2) (2005)
83. ETSI, Digital cellular telecommunications system (Phase 2+); GSM/EDGE Radio Access Network (GERAN) overall description; Stage 2 (2011), http://pda.etsi.org/exchangefolder/ts_143051v100000p.pdf
84. Qualcomm, Qualcomm Successfully Demonstrates Mobile Broadband using FLASH-OFDM at the Wireless Broadband East Africa Conference. Website (2006), http://www.qualcomm.com/news/releases/2006/12/05/qualcomm-successfully-demonstrates-mobile-broadband-using-flash-ofdm
85. ETSI, Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems (2005)
86. ETSI, Digital Video Broadcasting (DVB); Interaction channel for Satellite Distribution Systems; Guidelines for the use of EN 301 790 (2006)
87. ETSI, Universal Mobile Telecommunications System (UMTS); UTRAN overall description (2011), http://pda.etsi.org/exchangefolder/ts_125401v100100p.pdf
88. WiMAX Forum, WiMAX Forum™Mobile System Profile Specification Release 1.5 Common Part, (WiMAX Forum, 2009). Technical Specifications WMF-T23-001-R015v01
89. IEEE Standard for Local and metropolitan area networks- Part 16: Air Interface for Broadband Wireless Access Systems, Park Avenue, New York, (2009) http://standards.ieee.org/getieee802/download/802.16-2009.pdf
90. ETSI, Universal Mobile Telecommunications System (UMTS); High Speed Downlink Packet Access (HS-DPA); Overall description; Stage 2 (2011), http://pda.etsi.org/exchangefolder/ts_125308v100400p.pdf
91. ETSI, LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (2011), http://pda.etsi.org/exchangefolder/ts_136300v100300p.pdf

doi:10.1186/1687-1499-2012-114
**Cite this article as:** Pareit *et al.*: A novel network architecture for train-to-wayside communication with quality of service over heterogeneous wireless networks. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:114.