Service Boost: Towards On-demand QoS Enhancements for OTT Apps in LTE

K. Samdanis, F. G. Mir, D. Kutscher, T. Taleb
NEC Europe Ltd, Germany
{samdanis, mir, kutscher, taleb}@neclab.eu

This paper introduces the concept of dynamic Service Boost and proposes deployment solutions in mobile networks focusing on the 3GPP Long Term Evolution (LTE) architecture. The main idea is to introduce a time bound preferential service to subscribers based on predefined service contracts. By applying a light-weight, dynamic Quality of Service (QoS) control, operators achieve both, efficient network utilization and adequate QoS for users and content/application providers. This helps operators to use resources more efficiently, for example to enable a more efficient capacity sharing in the presence of increasing mobile traffic. We initially investigate the impact of Service Boost on Over-The-Top (OTT) traffic transmitted without any specific QoS guarantees over the so-called default bearer in LTE. Then we consider the design and analysis of a Service Boost architecture and framework for managing and prioritizing service requirements for certain applications within LTE. Finally, we elaborate the realization of Service Boost through congestion accountability.

I. Introduction

Application and service provision has changed significantly since the early days of 3G and its walled-garden services. Originally, the 3GPP architecture has relied on Internet Multimedia Subsystem (IMS) and its application control mechanisms for service provisioning with potential charging. In LTE, QoS is implemented between the User Equipment (UE) and the Packet-Data-Network (PDN)-Gateway (P-GW), relying on the bearer concept, a virtual resource that maps a certain QoS class (e.g., guaranteed bitrate for video streaming) to particular resource reservations. Today, there is a widespread usage of different types of mobile devices (smart-phones, portable computers, tablets etc.) with the dominant traffic generated by the so-called OTT applications, which run independently on the top of the mobile operator's network. Given the trend towards flat rate user contracts and the exponential data traffic growth, it is realized that a significant fraction of the revenue is created outside the mobile operator's network. Interestingly, all OTT traffic is transferred by LTE without any specific QoS guarantees, i.e. it is sent over the default bearer, which is used for best-effort traffic.

Converging data and service access in a single bearer approach can significantly simplify network design and operation. However, the vast amount of applications calls for a service differentiation mechanism. The Service Boost concept presented in this paper is intended to address two problems: (i) provide a way for users to request the preferred network services for selected applications and (ii) enable operators to monetize these preferred services, i.e., by charging users and/or Application Content Providers (ACP). The advantages of this approach are numerous. Firstly, it provides the means for offering users with a mechanism to request a better service for selected flows and secondly it enables the network to letting the user decide about important applications, so that network operators do not have to resort "traffic detection functions" and Deep Packet Inspection (DPI). Hence, Service Boost contributes to simplify network design and operation, while at the same time it is introducing a different kind of resource sharing other than TCP-like "fair sharing", in where network resources are shared equally between concurrent and competing flows. This would let the user differentiate and prioritize "more important" flows such as video streaming against "less important" ones such as software update download. Service Boost can also be (though not necessarily) associated to charging and thus can create additional revenue sources for operators.

In summary, Service-Boost is reflecting the current developments in mobile networks wherein network services are mostly substituted by OTT ones. Effectively, Service-Boost can: (i) enable users to request a better service on demand and (ii) enable operators to participate in the revenue creation that today benefits OTT application providers. The rest of the paper is structured as follows: Section II summarizes the related work. Section III describes the Service Boost concept and business models for revenue sharing, while Section IV presents the details of the Service Boost concept and describes a deployment scenario based-on Congestion Exposure (ConEx) mechanisms. Finally, Section V concludes this work and describes our future research directions.

II. Related Work

The LTE QoS mechanism for managing and allocating network resources is based on the bearer concept [3]. LTE specifies a number of bearers each associated with a different QoS class. As of today, the use of dedicated bearers is the closest means for materializing the Service Boost functionality within the current LTE networks. The process of bearer establishment and modification may be handled either by the network or user. Particularly, early 3GPP releases, i.e. Rel.5, introduced a UE initiated bearer management via the use of a vendor specific Application Programming Interface (API). Such paradigm shares certain similarities with the subscriber initiated Service Boost request, which concentrate on the fact that an API is essential towards the network to communicated QoS demands, while the network performs the admission control. For Service Boost such a terminal API is important since the user has a better perception of the QoS, but this

would impose certain standardized requirements for "opening" such API to App developers. An alternative way for requesting a Service Boost from the device side may be handled by the application via the Application Function (AF), which provides the Policy Charging and Rule Function (PCRF) the corresponding rules associated with the requested QoS. Such an approach may be used for requesting a Service Boost once establishing a new session.

Service Boost may also be realized during a session using the bearer resource modification request [4] to modify the resources for a traffic flow aggregate with a specified QoS demand or the Traffic Flow Template (TFT) installed both on the UE and P-GW. The Service-Boost approach, may utilize the UE bearer modification to enhance an application or the generic user performance in the following two ways: (i) to boost the performance for particular applications the UE may provide the required parameters of QoS Class Identifier (QCI) as well as Guarantee Bit Rate (GBR) values along with the flow description that needs to be boosted and (ii) to remove already mapped flows from a particular dedicated bearer while creating additional capacity for the boosted flow.

The realization of Service Boost described above implies the use of dedicated bearers for boosting the QoS of specific flows. However, nowadays most of the traffic is mapped on the default bearer. Without knowledge about the content inside the default bearer, operators employ DPI techniques to inspect traffic before introducing policies at the P-GW, which aim to differentiate traffic flows without modifying bearers [3] or use offload mechanisms [5][6] to route traffic towards the Internet directly without traversing the core network. Such approaches may alleviate traffic volumes handled by the mobile operator but are not addressing the fundamental problem of how an operator may expose its network resources in a way that it can offer better and innovative services that would also offset dwindling revenues. Our goal is to provide mechanisms that may offer both the user and APC the capability to request service differentiation on the fly satisfying user preferences under changing network conditions e.g. under congestion.

3GPP User Plane Congestion Control (UPCON) [7] outlines a set of use cases considering the user plane congestion due to high load users on particular cells or due to a high volume of control traffic generated for example by frequent keep alive messages related to certain applications. The objective is to explore congestion control by rate limiting certain type of subscribers, traffic types or applications etc. Considering the Service Boost, such user plane congestion control could provide the means for efficient resource management and for assuring the extra resources need to boost particular applications and users. However, a concrete Service Boost solution for applications that continuously use the default bearer is still open. In addition, LTE QoS mechanisms can provide limited Service Boost capabilities only within the operator's domain, lacking an end-to-end perspective.

Beyond the conventional 3GPP based approaches, an Open API framework is introduced in [2] to provide application developers with a rich set of APIs, where operators can expose their network capabilities to application developers for making the maximum use out of the network and delivering better services to the end users. The various valuable network assets that can be used for customizing and enriching existing services may include QoS functions, subscriber location and presence information. Eventually, network programmability through technologies such as OpenFlow could create new opportunities for providing QoS in relation with certain applications in a dynamic and flexible manner, where application developers and users have more control of the network resources. Such an approach is still at an early state, yet to be further explored in terms of integrating it with 3GPP mobile networks. This paper elaborates a Service Boost scheme for LTE and analyzes a ConEx based deployment focusing on the default bearer, while considering an end-to-end perspective.

III. Service Boost Concept

To manage the increasing data volume, operators employ various mechanisms for throttling traffic that are loosely classified into static and dynamic. A commonly used static method referred to as "volume cap" provides a rate-limit for user's traffic based on a pre-determined data volume. Such a policy is not considering network congestion, penalizing users even under low load situations. In contrast, dynamic mechanisms consider the network resource availability with the objective to police traffic based on resource consumption. In [1] Comcast outline a protocol agnostic way of throttling user traffic that is based on user resource consumption patterns to adjust offered capacity accordingly. However, none of the priory described methods actively involve the end host and service provider in the process of prioritizing a particular traffic type for enhancing Quality of Experience (QoE).

The Service Boost concept introduced here would enable users to request a preferential service for current applications, i.e., for a specific flow, on demand (a possible implementation approach is described in this paper). This would not only be beneficial for users interested to prioritize specific flows over others, but would also be beneficial for application and/or content providers who want to ensure or increase the probability that accessing their particular service provides the best possible service quality. A network service provider could thus enable better control forwarding for services according to these preferences and effectively act as a QoS broker [2]. A subscriber may request a generic QoS enhancement from the network provider, e.g. a bandwidth increase, or a QoS enhancement in relation to a particular service for a given time interval by paying an additional amount. In this way network providers can charge users by commoditizing the network resources for a short service boost related to particular applications or flows. Such an extra per user charging, provides the means for network operators to increase their average revenue per connection.

For a Service Boost request from the ACP, the desired network QoS is not acquired from the user, but indirectly via the ACP. The difference is the fact that the ACP uses an application API that specifies the related QoS parameter to the network, instead of a user API. In addition, the boost in this

case is only related to a particular service and application, but the user only needs to utilize conventional signaling to query it. For this paradigm two distinct business models can be envisioned. One were the users pay the application and then the ACP ensures the adequate QoS by purchasing the required commodities form the network provider and another, where the subscriber pays the network provider to provide both the application and the required QoS and in turn the network provider gives the application content provider the appropriate share to offer the desired application to the user. Hence, operators can enter the revenue loop between the subscribers and ACP by providing mechanisms for dynamic QoS upgrade through some Open/Standardized Interfaces.

Service Boost requests from subscribes or ACPs may be realized by one of the following charging models:

- **Token-based Model** charges users according to the QoS enhancement. Such tokens may accompany the purchase of a particular application or may be purchased by the user separately for an on demand use, associated with set of applications.
- **Contract-based Models** assume that operators offer charging plans following a tiered approach with different bandwidths and quota limits, i.e. gold, silver, platinum subscribers, preserving the QoE of important customers respectively by assigning special boosting quota with their subscription profiles.
- **Service-based Models** as a pay as you go paradigm to allow access with a certain QoS in relation with a particular service e.g. online gaming, where a group of users are playing interactive either on a social website or at some gaming server hosted.
- **Roaming-based Models** that offer selected services and content access combined with specified QoS on visited networks.

IV. Resource Aware Service Boost

This section presents our proposal for materializing Service Boost in LTE. Initially, we demonstrate the operation of Service Boost and then we elaborate its integration into the LTE architecture detailing also the main functional elements of the proposed scheme, before analyzing a deployment solution based on ConEx mechanisms.

A. Service Boost Depolyment Architecture

The main idea behind the operation of Service Boost is the need for maintaining a dynamic view of the user plane traffic within the Evolved Packet System (EPS). There are several alternatives for tracking user plane traffic and congestion in a network, which can be broadly classified as *"in-band"* and *"out-of-band"* signaling mechanisms. The Explicit Congestion Notification (ECN) [10] and ConEx [11] based-on in-band congestion signaling make the congestion information visible on the data path, while in out-of-band congestion singling is carried out separately from data [1], while OpenFlow [12] offers an out-band-control for the forwarding elements.

The rationale here is to embed soft-state in the user plane, default bearer, to maintain information about congestion/load. Given that the state is maintained on the user plane, a centralized entity referred to as Boost Server may query load/congestion events via the control plane. Such a Boost Server needs to build a global view of the user data plane and for this reason it utilizes observation points at the radio, backhaul and core network. The granularity of required state is an important design consideration. Naturally, a fine granular per flow based state requires a significant amount of resources. Other alternatives are to maintain statistics that are aggregated on Users, Groups, and Application that can be aggregated at coarser Cell level. Given the dynamic network view, operators have sufficient information for deciding which parts of the network are experiencing congestion and even if additional performance boost will be useful for a particular applications/sessions. The rest of the path that is beyond operator domain (e.g. Internet) can be handled by a proxy [8] or could use load/congestion information across peering domains [9].

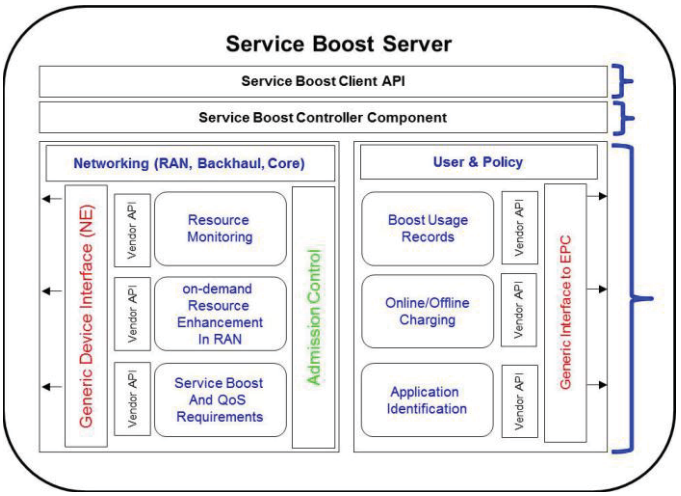

Fig.1: Service Boost Server Architecture

Fig.1 outlines the generic architecture for the Service Boost server illustrating the main functional elements, which can be divided into three distinct layers named (i) network management layer, (ii) boost controller layer and (iii) service boost client interface layer.

1) Network Management Layer

The network management layer is divided into two distinct parts, the networking responsible for managing network resources for Service Boost requests across the EPS (RAN, backhaul, core) and the user and policy part that conveys Service Boost user state information within the operator's network. The networking functional part assumes that the forwarding elements expose their *"*internal state*"* through some open or proprietary API and it is composed by the following functional elements including:

- **Resource monitoring** that keeps track of the network resources and distinguishing best-effort ones, which are the prime candidate for securing extra bandwidth to boost service requests, assuming that a dynamic resource provision mechanism is in place.

- **On-demand resource enhancement in RAN**, which utilizes the information regarding available capacity at each cell in order to explore path diversity and load balancing among neighbor eNBs for assuring the desired QoS for Service Boost requests.
- **Service Boost and QoS requirements** that focus on analyzing the QoS requirements, e.g. bandwidth, delay and jitter, in relation with a specific application once a Service Boost is requested.
- **Admission Control,** which determines if a particular incoming request can be served successfully by the network, boosting the user's performance. For the EPS, the Boost Server may rely on the admission control functionality already offered by the different network elements.

The user and policy part assumes communications towards the Home Subscriber Server (HSS) for reading user profiles and updating boost related information, while containing the following functional element including:

- **Boost Usage Records** that maintains user profiles in relation with boost usage. Usage information depends on how the boost is commoditized. One option is to introduce tokens representing a boost instance, which can be augmented with certain policies that limit the usage at certain times, cell sites or applications.
- **Online/Offline Charging** that offers on the fly boost tokens purchases through online/offline charging. The Boost Server provides the necessary binding for purchasing tokens, but such functionality can also be exposed to UEs by "In-App Payments" support.
- **Application Identification** is added to identify flows when the boost is initiated and assist how the necessary state should be communicated to various network elements for delivering the desired QoS.

2) The Boost Controller Layer

The boost controller is a wrapper entity around the two main components described above. It controls the execution of certain functions in the system and it also holds the relevant network state, which can include statistics at some pre-defined granularity, topology information about parts of the network for exploiting path diversity etc. In short, it holds the dynamic view of the network for providing the Service Boost.

3) Service Boost Client Interface

The client API facilitates the communication once a user requests from the Boost Server specific service parameters that the server knows by actively maintaining a network state. For example, a UE may request the current congestion level for reaching a particular gateway in the core network, in order to decide whether to start a particular application. The exact anatomy of such requests needs further research.

Once a UE requests a Service Boost the following steps, as illustrated in Fig.2, need to take place inside an LTE network, before obtaining the desired service associated with a particular flow. As an alternative deployment scenario we are considering a separate Boost Server and Boost Controller, which would be responsible for maintaining the network resource state, an entity equivalent to the SDN controller.

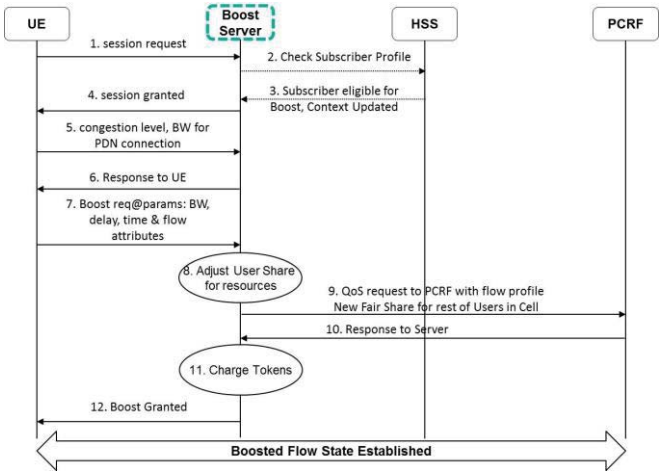

Fig.2: Process for requesting a Service Boost in LTE

Initially, to establish a connection between a UE and the Boost Server, a session request is transmitted from the UE as shown in step 1 on Fig.2. The server contacts the HSS for authenticating the UE in step 2 and 3, to ensure that it is eligible for the Service Boost. Once the UE is authenticated the Boost Server informs it in step 4 and a communication channel is then established between the two entities for querying Service Boost requests. The UE may query the Boost Server about the congestion state of a specific PDN connection as shown in steps 5 and 6.

A Service Boost request issued by the UE towards the Boost Server containing particular application parameters is shown in step 7. If the request is accepted the Boost Server adjusts the best-effort share of resources in order to accommodate the Service Boost request in step 8 and then informs the PCRF accordingly establishing the necessary policy rules in the network, in steps 9 and 10. For granted requests, the Boost Server charges the user e.g. via token mechanisms, and updates the user records in step 11. The boost grant response is send to user in step 12.

B. Service Boost by Congestion Accountability

In resource management mechanisms like ConEx [9], user traffic accountability is not coupled with the network usage. Instead, another cost metric is used that captures the impact of user traffic on other users, while sharing the network resources. This is based on co-operative sharing, where only the end users at a particular time know the utility for an offered bit rate by the network. From the protocol perspective, the network signals resource congestion by probabilistically marking ECN bits in the arriving packets that are conveyed to the sender via the feedback loop. In response, the sender declares its congestion contribution back to the network by setting a special code point in outgoing packets for balancing arriving congestion notifications. ConEx employs two functional entities the Policer and the Audit Function for correct protocol operation. In short, under network congestion,

it is expected that users with less important traffic back-off to accommodate traffic of important users.

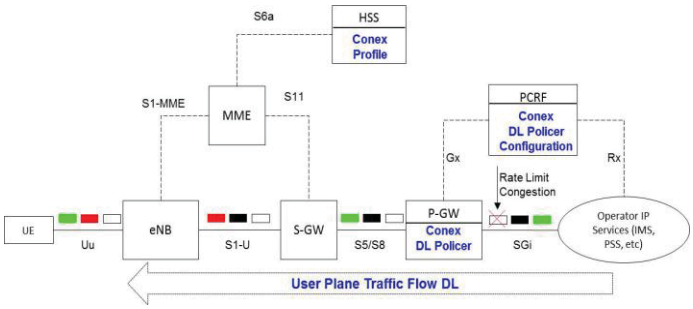

Fig.3: Downlink policer configuration

A ConEx Policer configuration is assigned for each user that contains the congestion allowance and the operator's policy for rate limiting traffic under congestion scenarios. Within the mobile operator's domain, the Policy is stored along with user profile at the HSS. When a UE attaches the network, the PCRF instantiates a user specific policer configuration for the user plane traffic. For policing traffic in the mobile network, separate policer configurations are employed for uplink and downlink directions that optimize access network resources under congestion situations.

Fig.3 depicts the downlink configuration where the Policer is deployed at the P-GW to rate limit traffic as soon as the user congestion quota is exceeding some pre-defined limit. For manipulating the Policer state, a configuration layer is added at the PCRF that can dynamically change its runtime state. Early packet drops by the Policer at the P-GW conserves capacity in the access network by rate limiting traffic of users who are exceeding their congestion allowance in a given time interval. Similarly, Fig.4 depicts a configuration where the policer is deployed at the eNB for rate limiting user traffic for the uplink direction. For this configuration, the Policer at the eNB rate limits user traffic as soon as the user is exceeding some predefined soft threshold.

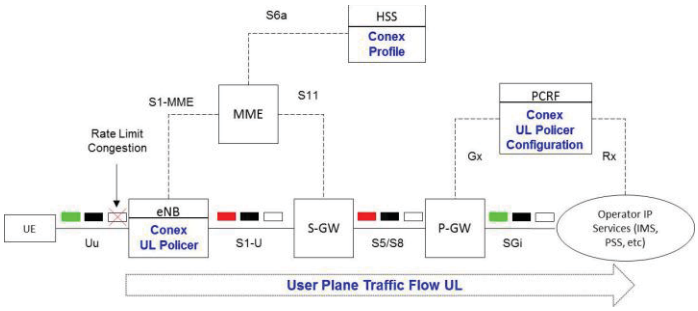

Fig.4: Uplink policer configuration

The Policer implementation can be implemented in different ways; for example, a simple token bucket can be employed for rate limiting the allowed user congestion in the network. Packet drops can be performed based-on some soft threshold at the Policer, which signals impending excess allowance usage to the sender. A typical problem with such a process is the non- differentiation of traffic type from the bulk of use plane traffic passing through the Policer. In one potential solution the end user already knows the Policer configuration and monitors how various applications consume the congestion allowance. By terminating unimportant applications, allowance can be shifted to important flows. However, for Service Boost support such an end host specific solution might not be sufficient, since the network is not involved.

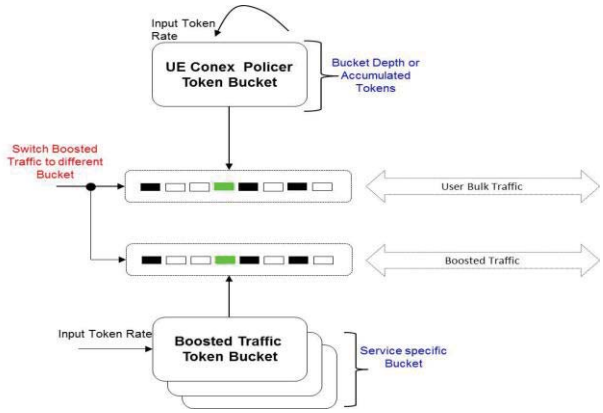

Fig.5: Differentiated Traffic Bucket

For introducing Service Boost support, we propose to define a network path specific token bucket that offers a differentiated forwarding behavior to selected user traffic. Instead of using the UE specific ConEx Policer, the boosted traffic is forwarded via another bucket achieving differentiated traffic treatment. The use of separate queues for conventional and boosted traffic policed by a different service bucket as illustrated in Fig.5 is also recommended. Two Differentiated Services Code Points (DSCP), should be used by the network for identifying the conventional and boosted traffic. For triggering the boost functionality, the UE should be able to convey the traffic profile with boost specific parameters to the Boost Server as elaborated in Fig.2.

As the Boost Server maintains user profile information and the necessary interfaces to the EPC, it can easily install the necessary rules in the PCRF that modifies the Policer configuration for boosted traffic. For charging, we envision that the Service Boost functionality is incorporated as an "application" in the UE. Third party applications that need to use Service Boost support may employ an "in-App" payment procedure that dynamically top up the tokens in the differentiated bucket according to a payment that is charged on the UE account. For cases where OTT providers pay on behalf of the user, there is a need for some kind of business relationship with the corresponding mobile operator as discussed in Section III.

The use of ConEx at the boarder of the mobile operator's network, i.e. at the P-GW, could also be used to determine the congestion level outside the operator's network towards neighboring domains. One important benefit for such a scheme is that it gives an end-to-end perspective of the path that spans to a corresponding node on the Internet. Typically,

mobile operators are mostly interested in knowing the level of congestion in their own system rather than in the Internet. For constructing an end-to-end view, proxy based solutions, as described in [8] should be used at the edges of the autonomous domains, e.g. operator network. Then the state of the path can be maintained by TCP connection monitoring, knowing in this way which segment of the path is congested. Enabling congestion exposure between two network domains would enable two interesting functions in the context of congestion management and Service Boost:

- Enabling the mobile operator to select peering networks according to observed congestion.
- Reacting to Service Boost requests, taking currently observed congestion in peering networks into account.

Assuming a mobile network is connected to more than one "uplink" peering network, it can monitor bulk congestion levels for traffic coming from and sent to a specific peering network. If congestion levels exceed a certain maximum threshold, the operator could apply traffic engineering to shift traffic to other peering candidate networks. In similar ways, a data center operator could do monitor congestion levels at its different Internet Service Providers (ISPs) and decide to shift traffic accordingly. If congestion is mainly caused by bottlenecks outside the operator network, the operator cannot assure that a Service Boost request would have a positive impact on the end user performance. In case Service Boost is associated with charging or some other form of compensation it would not be correct to charge the users in such situations. With ConEx, the operator would at least be aware of the congestion situation outside its network, so that it can react to service boost requests in a more appropriate and fair way. For example, the operator could respond to Service Boost requests that cannot be fulfilled to at least inform the user or application accordingly.

In summary, the ConEx-based approach discussed has the advantage that it does not require extensive network monitoring for implementing Service Boost. A Service Boost for a particular flow can be implemented by changing the congestion policing rules on a policing function on a network element. This could be implemented in a single-operator domain only, without require a global ConEx deployment. Assuming that there would be inter-domain congestion exposure, this could be used to make congestion visible that is not caused in other parts of the networks such as OTT data centers, other access networks and peering networks. This would enable operator to at least know when Service Boost is possible – and when it's not. However, this approach requires ConEx to be performed globally, so it is not likely to be deployed early.

V. Conclusions

This paper outlines the key mechanisms for materializing the Service Boost support in a mobile operator's network, which include:

- Service Boost through dedicated bearer based on both vendor specific API or AF signaling.
- Service Boost for a flow/application/session/flow in the default bearer by DPI and policy control.
- Service Boost through end-to-end ConEx mechanism.

Given the diverse application requirements along with different QoS support within the underlying mobile network, there should be a standardized mechanism for providing Service Boost. In this regard it is important the processof mapping the LTE QoS Class Identifier (QCI) on the DSCP domain code points. By having a standardized mechanism, Service Boost can be provided by mapping boosted traffic on a particular DSCP code point.

Another aspect that is also important to consider is the end-to-end Service Boost support. Such a view is significant in deciding about the admission control of a Service Boost request. If mobile operators can ensure that a Service Boost request will not be impacted on parts that are beyond the operator control then the request is accepted otherwise, valuable resources would be wasted without a significant performance benefit for the user. Alternatively, if mobile operator's need to assure a stable end-to-end Service Boost, the role of Service Level Agreements (SLA) between the operators and peering network becomes important. Further work is needed in order to evaluate our ConEx based Service Boost proposal against conventional QoS mechanisms.

References

[1] C. Bastian, et.al, Comcast protocol agnostic Congestion Management System, IETF RFC 6057, Dec 2010.

[2] Beyond Core Telecommunication Business, Growing a service provider's brand with Application Enablement, Strategic White Paper, 2011.

[3] H. Ekstrom, "QoS Control in 3GPP Evolved Packet System", IEEE Communications Magazine, Vo.47, No.2, Feb 2009.

[4] 3GPP TS 23.401, GPRS enhancements for E-UTRAN Access, v12.1.0 Rel. 12, Jun 2013.

[5] L.J. Ma, "Traffic Offload mechanisms in EPC based on Bearer Type", IEEE 7th WiCOM, Sept. 2011.

[6] K. Samdanis, T. Taleb, S. Schmid, "Traffic Offload Enhancements for eUTRAN", IEEE Communications Surveys and Tutorials, 3rd Quarter 2012.

[7] 3GPP TR 22.805, Feasibility study on User Plane Congestion Control, v.12.1.0, Rel.12, Dec 2012.

[8] S. Kopparty, et.al., "Split TCP for Mobile Ad-Hoc Networks", IEEE GLOBECOM, Taipei, Nov. 2002.

[9] B. Briscoe, et.al, "Policing Congestion Response in an Internetwork using Re-feedback", ACM SIGCOMM, Aug 2005.

[10] K. Ramakrishnan, et.al., Explicit Congestion Notification, IETF, RFC 3168, Sep 2001.

[11] M. Mathis, B. Briscoe, Congestion Exposure (ConEx) Concepts and Abstract Mechanisms, IETF Internet Draft draft-ietf-conex-abstract-mech-07, work in progress, July 2013.

[12] N. McKeown, et.al., "OpenFlow: enabling innovation in campus networks", Computer Communication Review, Vol.38, No.2, Mar 2008.