Internet Engineering Task Force
Internet Draft
Intended status: Informational
Expires: January 2016

Attila Mihaly
Szilveszter Nadas
Ericsson

July 6, 2015

Middlebox Communication Enabling for Enhanced User Experience
draft-mihaly-spud-mb-communication-00.txt

Abstract

In this draft we address some of the key discussion points related to the scope of Substrate Protocol for User Datagrams (SPUD). Specifically, we show how we can define the middlebox communication framework such that it allows uneven resource sharing on the path among the endpoints enhancing in this way the user experience. Issues related to trust and incentives as well as how to support user decisions in this eco-system are clarified.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on January 6, 2016.

Table of Contents

1. Introduction

This internet draft relates to the recent IETF discussion around a new prototype protocol called Substrate Protocol for User Datagrams (SPUD)[SPUD_ML]. There is a wide range of opinions for the role of such a substrate protocol, in the scale for "indication of session start and stop for NATS" to "possibility of authenticated in-band signaling channels" with no clear consensus. [SPUD92] describes that SPUD is an extensible in-band channel that allows endpoints to signal traffic meta-data to the middleboxes on the path. It also provides a mechanism for the middleboxes to signal back to the same endpoint using the same in-band channel.

The discussion around SPUD has identified a number of constraints on the information exposed

- It is based on declarations only, thus no negotiation is needed between the parties which reduces the communication latency. There is also no assumption on what action (if any) will follow a given declaration.

- Endpoints/middleboxes may trust, but can verify the information received. The best way to prevent cheating is to remove incentives to do so.

- Incremental usefulness, no mandatory minimum vocabulary needed, i.e. SPUD need not be supported by all nodes on a path before a benefit is seen. All parties must ignore (and not delete) what they don't understand. The sender must also assume it may not be understood. This facilitates incremental deployment

Another constraint related to these was mentioned several times: the exposed information shall not change the total share of the user (from a bottleneck resource). This constraint highly decreases any incentive of the user to cheat, but it also makes QoS solutions much less efficient, because the networks must meet the QoS demand of the most resource demanding service for all users all the time. This might be possible in some networks (e.g. fixed line), but this is much more challenging for the costly resources of cellular networks.

QoS solutions are available in cellular networks [CQOS]. It is the role of Traffic Detection Function (TDF) to map the existing flows to the cellular QoS. This function is challenged by encryption and the current scope of SPUD does not help solving this challenge much.

In this draft we intend to explore how to make accessible the existing QoS framework for encrypted traffic. We introduce incentive frameworks which allow to both increase and decrease the resource share of the user and which also have consequences on future shares. We also explain how the endpoints can control their share without having too much bothersome user interactions.

2. Beyond the equal share

Declarative markings that middleboxes may trust but can easily verify originally aimed "to treat all markings on packets and flows as relative to other markings on packets and flows from the same sender". The main reason for this limitation was avoiding some of the trust issues. Indeed, as long as the communication cannot influence the overall resource share that a given user gets, there could be little incentive for the user to send false information to the path as any improper treatment would likely affect the user traffic in a negative way.

We believe that this constraint restricts the communication vocabulary too much. There is a significant potential in optimizing the overall utility for both the network and the subscribers by allocating resources unevenly when there is a congestion event. For

example, by giving more resources to a web download than to a background file transfer, the overall user satisfaction increases, because the impact on user experience improves for the former without significantly affecting the experience of the latter. An eco-system would therefore be needed that allows temporal deviations from the equal share of the different users sharing the same resource.

There are a few pre-requisites for such eco-system to happen. The first challenge is providing the right incentives for both the endpoints and the network to cooperate along this paradigm. To achieve this, the endpoints shall be able to select between service options and perceive the benefits of their selection. A further challenge is avoiding mis-use of a high resource-share service by endpoints: by default the endpoints' interest would be to select to use that service in all cases, regardless whether they really need it or not. It should be possible by the network to impose by some means that the endpoints select the more favorable service only if they really need the respective treatment for their traffic. Examples on how these targets can be achieved are detailed in sections 3.1. and 3.2. Example use cases showing the benefits of using the incentive frameworks are given in section 4.

A further aspect of uneven resource share allocation is the endpoint control of how its different flows should be treated. Details on this are in section 5.

3. Incentive frameworks

There are different potential solutions for cooperation between endpoints and middleboxes that enables unequal resource sharing and avoids mis-use by the endpoints. The one we describe here is designed with the following features in mind:

- It builds on existing QoS architectures, i.e., it does not require building a new QoS architecture.
- It requires a minimal update of the existing subscription models (e.g. bucket-based charging)
- It is based on the decision of the endpoints, i.e., the users may control which applications and when to use a specific service delivery option
- It may be based on a declarative communication.

- It is fair to the users: all users may have access to the same type of service delivery option. Moreover, there are no negatively discriminated users, including users that are not willing to participate (they will receive the default service delivery option all the time).

3.1. Economic incentive based cooperation

One possibility to avoid mis-use is to apply some penalty for using the more favorable service. Examples of such penalties are

- Applying higher price for the better service. For example, the Internet Service Provider (ISP) could introduce new service delivery options besides the existing Interactive (I) one, e.g.,
  - A real-time (RT) class (with low delay and jitter guarantees up to a certain throughput threshold) and a higher cost per bit
  - A Lower-than-best-effort (LBE) class with looser throughput delay and loss guarantees but a low cost per bit
- Applying limited caps for the better service. E.g. the same service delivery options I/RT/LBE as before but having different monthly caps (in bytes) for the new classes.
- Another option is to keep the single bucket per user solution, but apply different bit-multipliers depending on the requested treatment, for example:
  - Real-time class: 3-10x multiplier (i.e. a single RT octet results in decreasing the user's bucket by 3-10 octets)
  - Interactive class: 1x multiplier, similar to today's BE handling. Note that the operator may incentivize sending flow meta-data by applying a smaller, e.g., 0.9x multiplier when flow metadata is provided
  - LBE: 0-0.3x multiplier

The above service offerings provide endpoints the possibility to access to new services (RT class), generate (near) toll-free data (LBE), and get access to better/cheaper interactive/video service, giving the users incentives to use, but not to mis-use a given class, because of the penalty applied. We consider the above service offerings fair to the user as long as everyone has access to the

same service categories for a fair price, and a service class is handled the same way for all endpoints.

3.2. Non-economic incentive based cooperation

The incentive framework presented in section 3.1. provides a light-weight solution from the points of view of service description (SLAs), policy control and charging. Still, it may have issues with ensuring service guarantees especially in cellular network due to the shared, limited and costly radio resources. In this proposal the operator provides soft service guarantees which do not imply extra 'cost' for the subscriber and therefore no strict guarantees are provided for the different service options. Thus it does not face the difficulties of the SLA based service offerings.

The basic idea is that the ISP offers different options in "service delivery", e.g., a "background" service delivery option when there is a need for additional network resources for some other traffic. The users of this service delivery option are given relatively lower resource share than if they were using the generic "best-effort" (BE) service - thus other users can benefit from the fact that the users using the "background" service delivery option can be down-prioritized when accessing shared resources, e.g. a radio cell. In turn, the ISP offers tokens, which are objects that represent the right to perform a certain action. In this context, the users can request a so-called "priority" service delivery option given that they have accumulated tokens by using the "background" service delivery option. Users thus are motivated to request the "background" service delivery option when their traffic copes with more relaxed network QoS.

The main concept is depicted in the example figures Figure 1 (indication and usage of "background" service delivery option) and Figure 2 (request, acceptance and usage of "prioritized" service delivery option).

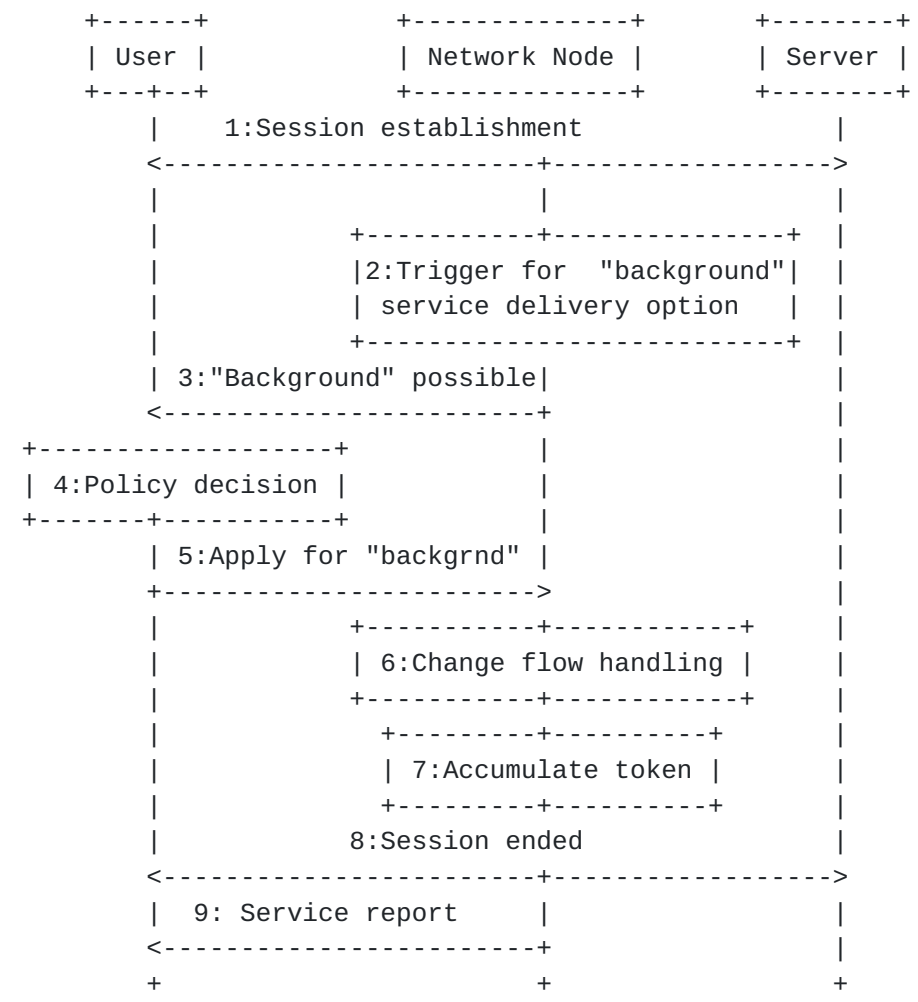

Figure 1 Example sequence diagram illustrating the background service delivery option offering and accumulation of the tokens by user by using the background service delivery option

Description of Figure 1 for the indication and usage of "background" service delivery option:

1. There is a session established between the User and Server that is susceptible for "background" service delivery option, but currently using the normal service
2. At a given time, the network experiences congestion in the region that involves also the session previously mentioned. Based on this, a trigger is generated for the possibility of using "background" service delivery option

3. A notification that the "background" service delivery option is possible is sent to impacted users, thus also to the user in this sequence chart.
4. There is a policy decision by the user whether the conditions are such that the given session may use the "background" service delivery option
5. If the decision is "yes", then the user applies for the "background" service delivery option for this session
6. The network changes the flow handling of the given session according to the "background" policy
7. At this point the network also starts to accumulate tokens for the given user
8. When the session ends,
9. The network may send a service report to the user including e.g., the session identification that used the background service delivery option, the time elapsed, traffic volume sent, tokens accumulated etc.)

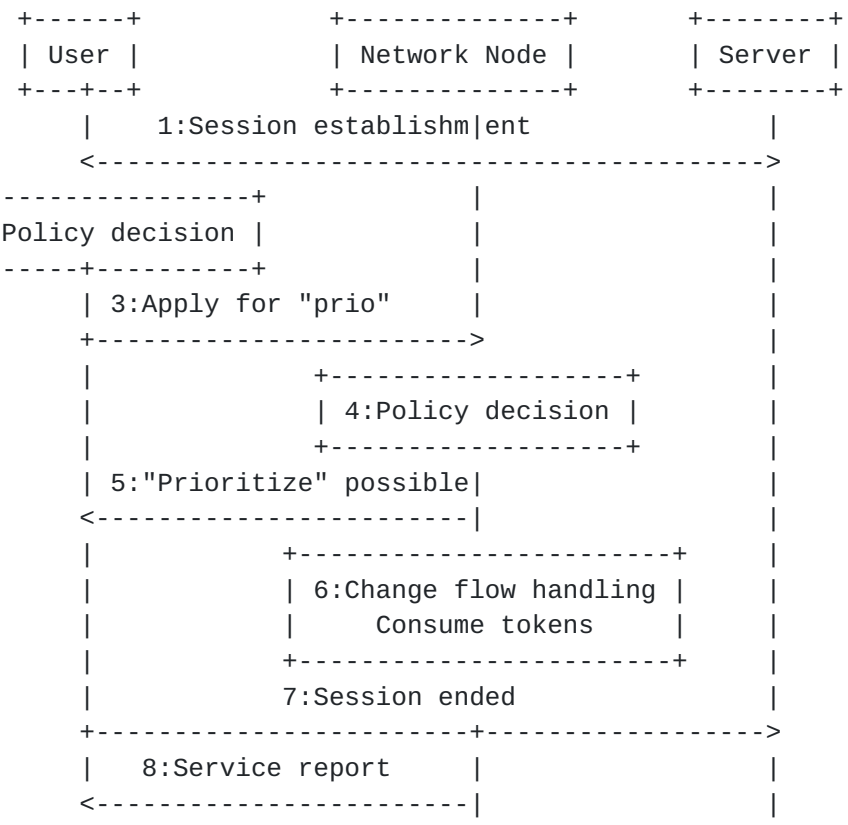

Figure 2 Example sequence diagram illustrating the request and utilization of prioritized service delivery option by the user based on accumulated tokens

Figure 2 depicts an example of requesting, acceptance and usage of 'prioritized' service delivery option:

1. There is a session established between the User and
2. At a given time the policy decision in the client triggers that the given session may benefit from "prioritized" service delivery option for improved QoE
3. A request for "prioritized" service delivery option is sent to the network
4. The network takes a policy decision on applying the "prioritized" service delivery option for user

5. If the decision is "yes", then the network send a "Prioritized service delivery option acknowledged" notification to the user
6. At the same time, the network changes the flow handling of the given session according to the "prioritized" policy and it also starts to consume tokens for the given user
7. When the session ends
8. The network may send a service report to the user including e.g., the session identification that used the "prioritized" service, the time elapsed, traffic volume sent, tokens spent etc.)

4. Use cases

The purpose of this section is to show that there are common use cases where the incentive frameworks previously defined give benefits for the endpoints.

4.1. Web acceleration

A simple example using the non-economic incentive framework in 3.2. is when a user downloading software updates starts gathering tokens (given that is notified that due to high cell load he is eligible for "background" service delivery option). Afterwards it uses the accumulated tokens for his critical traffic, e.g. for the prioritized download of the critical content of a web page to shorten the time until rendering starts.

4.2. Video streaming

Below we give another example related to streaming video, also using the incentive framework in 4.2. The idea is that the user asks for "background" or "prioritized" service delivery option depending on the current play-out buffer level. An example selection algorithm is described in Figure 3, where the service delivery option decisions in the client are taken based on the threshold levels in Figure 4 (here we made it apparent that the concept may be applied also for adaptive i.e., DASH videos; the client has a buffer-based algorithm for choosing a specific quality representation in this example). The "prioritized" service delivery option provides relative prioritization for low buffer levels. In this way video freeze events due to buffer underrun may be avoided or at least reduced and pre-buffering times also reduced, improving the QoE of the users. Once the buffer occupancy reaches a level that is considered safe for avoiding the video freeze the client may switch back to the normal service. If the buffer occupancy further increases reaching comfortable values then the user/application may apply for

```
   "background" service delivery option in order to accumulate tokens
   for further potential low-buffer events.

   normalService:
     if (buffer > Max1Threshold and networkOfferAvailable) {
       askForService("background");
       goto backgroundService;
     }
     if (buffer < Min1Threshold and isTokenAvailable()) {
       askForService("priority");
       goto priorityService;
     }
     wait();
     goto normalService;

   backgroundService:
     if (buffer < Max2Threshold) {
       askForService("normal");
       goto normalService;
     }
     wait();
     goto backgroundService;

   priorityService:
     if (buffer > Min2Threshold) {
       askForService("normal");
       goto normalService;
     }
     wait();
     goto priorityService;
```

Figure 3 Example pseudocode for different service delivery options for the streaming video case

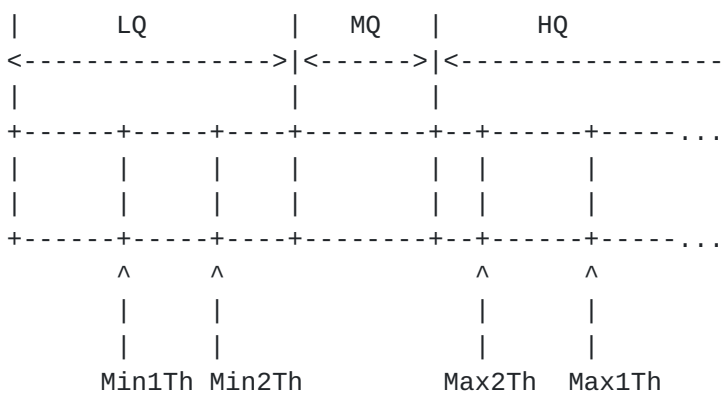

Figure 4 Example streaming video buffer thresholds for service delivery option and representation switches (LQ/MQ/HQ= buffer size ranges where the media client requests Low/Medium/High Quality representation chunks, respectively)

5. Endpoint control

The use cases for non-equal share treatment illustrated that the treatment to be selected for different applications may vary from application to application and may also vary during a session using a certain application.

In order to be fair to users and applications the users should have the control on which applications and when to use a specific service option. Indeed, having a stipulated treatment of certain application may result in positive or negative discrimination of the corresponding service provider. The endpoints should be provided the opportunity to opt in for certain treatment for different applications, or ultimately to not require any of the different service options. For example, any of the following user types should be possible in the eco-system:

- Bit-pipe enthusiast: wanting to get the same equal-share bit-pipe service as today for all his applications at all times

- Default user: being aware of the mutual advantage of using unequal share and wanting to cooperate, but not bothering about or not having the necessary technical knowledge to control its different applications. There should be some default operator or community settings that such types of users may rely on

- Biased user: considers only one type of service as important and wants to enjoy it with highest possible QoE. For example, a video fan being offered the service options presented in 3.2. would accept all offers for "background" service delivery option for all its applications except video, and would spend all its accumulated tokens on enhancing its preferred video application.

- Premium user: wanting to get as much quality out of the network service as possible and wanting to pay extra for it.

A common need for all these different user types is the ability of monitoring of the KPIs of the service delivery and possibility for understanding and verification of the advantages received by choosing a certain service delivery option.

Both service control and service monitoring would be quite cumbersome for the users so it requires some support, i.e., means to delegate the user choices either to user operating system (OS) or a separate application. Let us call this application the Quality of Experience Controlling Application (QCA). An example on QCA functionality is shown in Figure 5: QCA takes the role of communication with the network, i.e., it receives network notifications and sends service delivery option requests. In order to be able to send meaningful service delivery option requests, QCA should have access to the information about the applications that may use different service options, e.g., when they use the network resources, their identification (e.g., five-tuple) and potentially about the experienced QoE for these applications. This is shown by steps 4 (identification of APPs with on-going flows that are potential candidates for the background service delivery option) and 5 (inquiring their state) in Figure 5.

The QoE Controlling Application (QCA) makes the policy decision as in step 6 in Figure 5 and may also keep a statistics about different service options usage based on the service reports received in step 11. QCA should have a user front-end, where the user may configure its desired policy settings, i.e. which applications and in which conditions may be downgraded to the "background" service delivery option and which are those and in which conditions that could benefit from the "prioritized" service delivery option. The complexity of this user interface should be reduced such that QCA should be able to run in the background, i.e., the user should not be forced to acknowledge a certain policy behavior during its session. There may be different options to achieve this, for example, the user could be offered a number of default settings provided e.g., by the operating system vendor, or the ISP when installing the QCA to the user device so that the user is only

required to configure these settings if he wants to have some particular changes. Community databases for default user settings similar to AdBlockPlus plugins [ABPF] could also be quite useful options to select from.

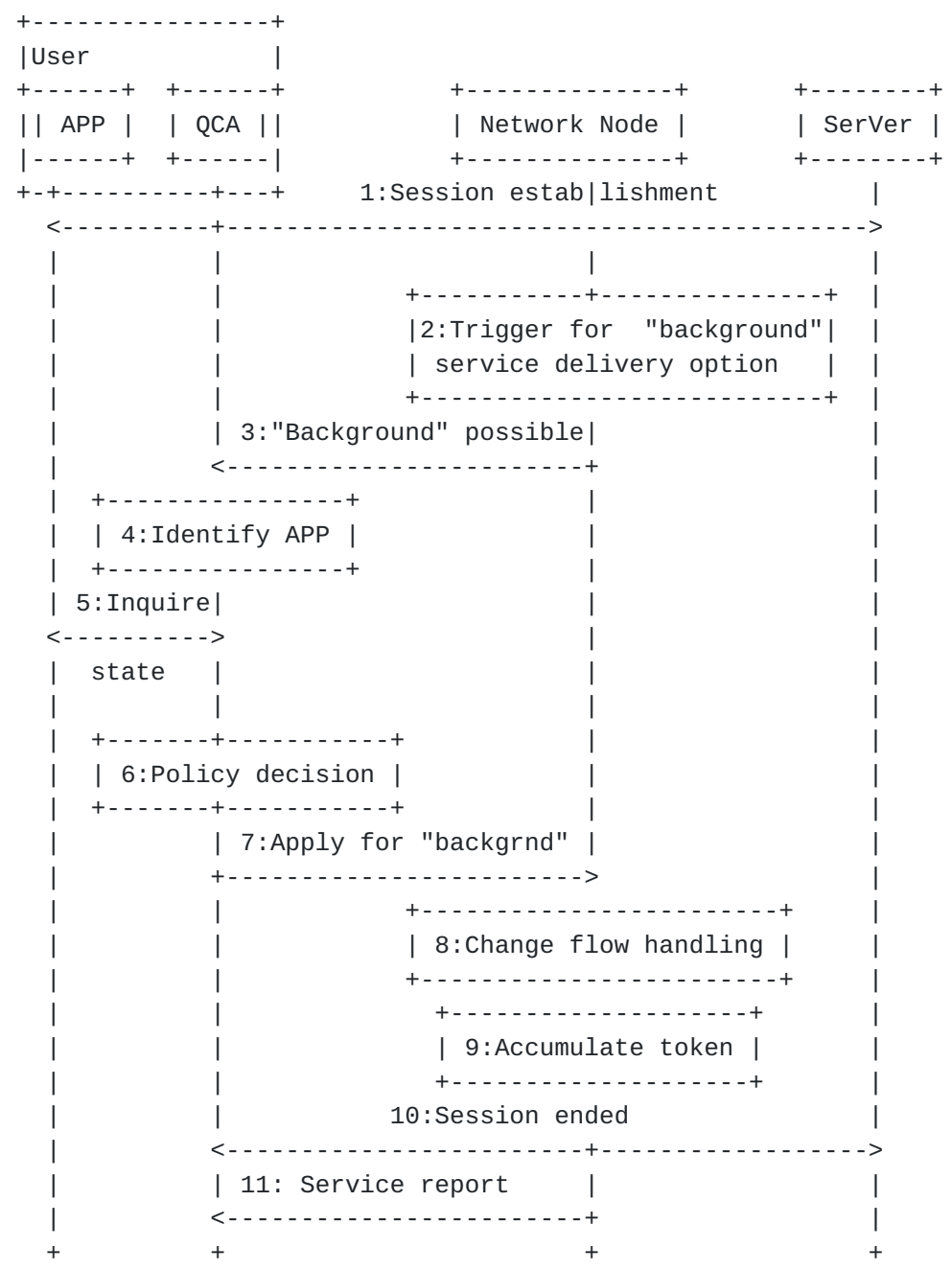

Figure 5 Example sequence chart showing the role of QCA in the example background service delivery option offering and usage by an application APP in Figure 1

6. Security Considerations

There are a number of security considerations, which is TBD to clarify and write down. The general consensus within SPUD discussion is to experiment first and then deal with security considerations. This draft has been written in this spirit.

7. IANA Considerations

The current draft does not pose any IANA considerations

8. Conclusions

In this draft we explore how to make the existing QoS framework accessible for encrypted traffic, based on authenticated declarative communication that may be carried over SPUD. We defined an incentive framework that allows the usage of different service options and discourages the mis-use of them. We also showed how the users can control access to the different service options without having too much bothersome user interactions.

This work is not complete, future work is needed in the following areas:

- How could the framework be extended to service providers to declare their intent and receive different service options
- What are the required changes to endpoint architectures for a simplified endpoint control
- Community discussion on how the proposed framework relates to the net neutrality principle, and what measures are needed to ensure that
- How to provide security for the communication
- Identification of the right forum for this discussion, if not SPUD.

9. References

9.1. Normative References

9.2. Informative References

[SPUD_ML] SPUD mailing list, see http://www.ietf.org/mail-archive/web/spud/current/maillist.html

[SPUD92] Brian Trammel, Substrate Protocol for User Datagrams https://www.ietf.org/proceedings/92/slides/slides-92-spud-1.pdf

[CQOS] Reiner Ludwig et all, "An Evolved 3GPP QoS Concept", VTC 2006 Spring

[ABPF] https://adblockplus.org/subscriptions, checked 2015-07-06

10. Acknowledgments

This document is the result of discussion in the SPUD mailing list and internally within Ericsson. We thank all who participated.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Attila Mihaly
Ericsson
Budapest
Hungary

Email: attila.mihaly@ericsson.com

Szilveszter Nadas
Ericsson
Budapest
Hungary

Email: szilveszter.nadas@ericsson.com