An Adaptive Policy Approach to Video Quality Assurance

Joseph McNamara*, Sven van der Meer†, Liam Fallon†, John Keeney†, and Enda Fallon*
*Software Research Centre, Athlone Institute of Technology, Athlone, Ireland
j.mcnamara@research.ait.ie efallon@ait.ie
†Network Management Lab, Ericsson, Athlone, Ireland
sven.van.der.meer@ericsson.com liam.fallon@ericsson.com john.keeney@ericsson.com

***Abstract*—Video in all its forms is probably the most important service carried on networks today and few would argue that video quality assurance is one of the most daunting network management challenges. Quite often, video optimization strategies and their decisions are an integral part of either the video protocol (e.g. dynamic adaptation of rate and quality) or the distribution systems (e.g. multi-level caching architectures). A unified method of assuring video services is a formidable task, especially as the world prepares for the adoption of 5G network concepts and the associated complexity. In parallel, policy has been proposed as an approach for managing domains in a flexible and adaptive manner. In this work, we describe our approach to use adaptive policy to externalize the goals and decision making of optimization strategies in the form of a network resource evaluation and path selection experiment aimed at video service quality assurance. In this paper, we present our approach, outline our initial implementation and discuss our preliminary results.**

***Index Terms*—Service Assurance, Video Optimization, OTT, Adaptive Policy, Work in Progress**

I. INTRODUCTION

The Ericsson Mobility Report [1] indicates video as the dominant service being used in mobile networks; around 14 Exabytes per month in Q3/2017. Popular immersive video formats generate traffic 4 to 5 times that of standard video [2]. Assuming continuous growth, video traffic will account for 75% of all mobile data traffic by 2023.

As the advent of 5G networks draw closer, it brings with it the promise of larger data rates, low-latency, higher bandwidth and a new level of energy efficiency through the implementation of networking concepts such as (i) Mobile Edge Computing (MEC), (ii) Network slicing, (iii) Virtualized Network Functions (VNF), and (iv) Beamforming, to name a few. While these concepts contribute to the 5G promise, they have also compounded the complexity in regards to video management. Therefore we have seen a drive towards automation, adaptive policy and machine learning paradigms in an effort to mitigate this complexity.

In our work we propose adaptive policy as an integrated, unified video optimization approach. This unified approach is important for service assurance of all video-related services. The key requirements of this unified approach are to (1) measure the QoS capabilities of the network, (2) translate them into SLAs and QoE specifications as the basis for measuring service delivery, and (3) provide a closed-loop monitoring-and-repair policy system to alter the network if required.

We hope to move towards a mathematical model for the policy that governs the network for service assurance. A very good candidate is ΔQ (based on the model developed in [3]).

A. Usage Scenarios

The classic scenarios can be summarized as streaming videos from various content providers, on-demand, using specific applications or embedded in HTML pages. Emerging usage scenarios use video as facilitator (or main contributor) for a complete, purpose-driven user experience. One example is discussed in [4]: event experience, typically focusing on worldwide sporting events. Here, video plays an important part of an offering comprising all aspects of the event, for both participating and remote viewers. Increased deployment of video-enabled devices and better Quality of Service (QoS) promised by 5G mobile networks should create a technical environment facilitating many new, unforeseen usage scenarios.

B. Video Optimization

Approaches and solutions are available for optimizing particular network nodes, cross-layer optimization, end-to-end optimization, and optimization of OTT flows. The developed techniques include pre-selected and best-effort quality, variable bitrate based on network conditions or client Quality of Service (QoS) parameters, single or hierarchical caching of videos, optimization of local or intermediate buffers, Quality of Experience (QoE) driven methods, and of course hybrids using two or more of the above. In [5] the authors present a cross-layer optimization algorithm using cache and buffer methods, aiming for fast video delivery. The algorithm is evaluated in a mobile network using relaying stations.

Typically, video optimization is done *out-of-network* using transcoding, transrating, time-shifting, and pacing [6]. *In-network* optimization techniques are becoming popular, utilizing available information of the underlying network. An optimization of radio base stations for multi-user video streaming is detailed in [6].

HTTP-based Adaptive Streaming (HAS) is an *in-network* technique that allows for better resource utilization using multi-layer information to deliver and if required adapt the best possible video stream given network conditions. A survey of HAS can be found in [7].

A few limitations remain: solutions are client-driven, hard to direct by network operators, and not policy-driven. Combining

978-3-903176-14-0 © 2018 IFIP

QoE with HAS promises to overcome them. In [8] the authors present in-network QoE management for video streams. This approach provides an interface for the network operator to steer the optimization process, thus facilitating policy control. In [9] the authors add fairness to the QoE management using client-transparent proxies. In February 2017, ISO has published the MPEG Server and Network Assisted Dynamic Adaptive Streaming over HTTP (SAND DASH) standard for video streaming over the Internet [10]. A programmer's introduction to and a demo of SAND can be found on Github [11]. In [12], the authors develop a multi-server multi-coordinator framework, which helps to model groups of clients accessing spatially distributed edge servers for replicated video content.

C. This Paper

This paper describes our initial work in probing a unified approach to video quality assurance that aims to integrate (virtually) any usage scenario discussed in §I-A with existing *in-network* video optimization techniques introduced in §I-B in the context of 5G mobile networks.

To address the requirements of the approach we have specified a system architecture and conducted experiments which (1) generate Mean Opinion Score (MOS) to evaluate network path quality, (2) deploy adaptive policy for network path consideration, and (3) implement decisions through the network controller.

§II describes a closed-control system [13] with an adaptive policy engine (APEX, [14]) similar to the setup described in [15] with additional components for video evaluation and traffic routing. Later extensions will add base stations, edge nodes, and virtual nodes. §III details the deployment of core components and the policy designed to steer the video traffic. §IV discusses our preliminary test results and evaluates the used policy and methodology. §V describes our experience and future work in this area.

II. SYSTEM ARCHITECTURE

The architecture of our closed loop system shown in Fig.1 follows a testbed architecture described in our previous work [15] with expanded utilization of component capabilities and the addition of a video evaluation framework. Full description of the core components of the system architecture can be found at [16]. Modifications to the previous architecture include the *Context Builder* and the *EvalVid tool-set*.

Context Builder realizes a mediator between the deployed SDN controller and APEX. The main output of the context builder is default MOS and predefined paths, which become a basis for decision making. The component is realized as a script to facilitate fast changes and adaptation.

EvalVid tool-set[1] is used to evaluate the quality of the video stream. It allows for the generation of a MOS [17], which is a QoE metric. In this initial experimentation, we focus on packet loss, mainly to keep the dynamicity of the testbed low. A full QoS metric (jitter, delay, throughput) can be added later.

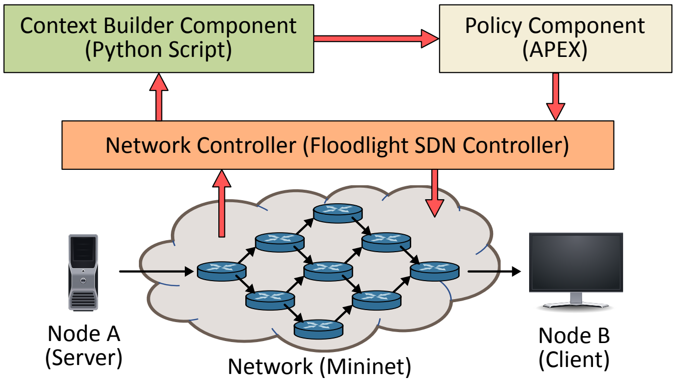

Fig. 1: System Architecture

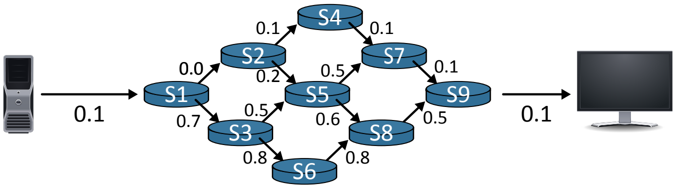

Fig. 2: Network Diagram containing packet loss information for links

Policy Component is the implementation of our APEX engine [14] to deploy, trigger, and execute policies. For this paper we have defined a single policy, which uses a video quality evaluation metric to influence network configuration to optimize video streams. Policy decisions are implemented through Floodlight to reroute network streams.

All components in the architecture are Free and Open Source Software (FOSS). This means that our experimentation can be easily repeated using the instructions from [16].

III. TESTBED AND POLICY

This section details the implementation of our approach in four algorithms, presented in pseudocode.

Algorithm 1 shows the creation of the Mininet emulated network topology displayed in Fig.2, connection of the Floodlight SDN controller and the enabling of the firewall used to enforce video stream paths. Hosts are then pinged to ensure the successful configuration of the network (lines 1-15). Access is then given to the Mininet Object Command Line for the execution of the VLC stream and client laid out in Algorithm 2. The script concludes with a messaging loop responsible for reconfiguring the firewall to adopt a new path specified by our policy running in APEX (lines 16-24). When the experiment is finished, we simply stop the MO (line 25).

Algorithm 2 shows the procedure to generate a video stream from the server (Node A) to the client (Node B) where it is recorded for analyses. VLC is executed on Node B where it is configured to receive and record the video stream (lines 2-4). VLC is also executed on Node A where it is configured to stream video to our client using the RTP/MPEG protocol (lines 5-7).

Algorithm 3 shows the procedure to generate each MOS. The original video is used to generate a Peak Signal to Noise Ratio (PSNR) (lines 1-7) and this is stored as a reference

---

[1] http://www2.tkn.tu-berlin.de/research/evalvid/fw.html

PSNR (refPsnr). A PSNR is also generated for the streamed video (streamPsnr). Both PSNRs are used to generate a MOS through the EvalVid Tool-set (lines 8-11).

Algorithm 4 consumes the MOS of a streamed video and updates the recorded MOS for the path in the policy's context. The policy then determines whether this new MOS should cause the path for video streaming in the network to be amended or not. *Match*: the MOS of the incoming event is verified to ensure an expected value is received (i.e. a value between 1 and 5). After verification the MOS is passed on to the next state. *Establish*: compares the stored max MOS against the incoming MOS and records the greater in policy context. *Decide*: examine the possible paths stored in the policy context and the path with the best MOS is selected as the active path. If a new active path is selected the path identifier is passed onto the next state. *Act*: take the path identifier for the new active path and packages it into a response. This response is used to configure the new path for the video stream through the Floodlight SDN controller. The defined policy realizes a context-aware MEDA (Match, Establish, Decide, Act) policy [18].

Algorithm 1 Configure and Run Mininet

```
procedure MININET                          ▷ Mininet, Floodlight, and Kafka
    FLC $\leftarrow$ new Floodlight Controller
    MO $\leftarrow$ Mininet Object (TCLink)
    MO $\leftarrow$ Node A, Node B & 9 switches
    MO $\leftarrow$ 14 (link $\times$ packetLossPercentage)
    MO $\leftarrow$ FLC
    start MO                               ▷ topology & controller
    while $\neg Pingall$ do                ▷ wait for nodes
        pinghosts()
    end while
    FLC $\leftarrow$ enableFirewall()
    FLC $\leftarrow$ cfgDefPath()          ▷ configure default path
    while $\neg Pingall$ do                ▷ wait for configurations
        pinghosts()
    end while
    MOCL $\leftarrow$ Mininet Object Command Line
    MOCL $\leftarrow$ Kafka Consumer Object
    MOCL $\leftarrow$ APEX output subscription
    while $message$ do                     ▷ process Kafka messages
        if $activePath \in$ message then
            ap $\leftarrow$ activePath(message)
            FLC $\leftarrow$ cfgFirewall(ap)
        end if
    end while
    stop MO                                ▷ cleanup
end procedure
```

Algorithm 2 Video Stream

```
procedure VIDEO STREAM                     ▷ Node B $\leftarrow$ Node A
    MOCL $\leftarrow$ Mininet Object Command Line
    XTB $\leftarrow$ MOCL.xterm(NodeB)
    XTB $\leftarrow$ vlcwrapper(url).record()
    XTA $\leftarrow$ MOCL.xterm(NodeA)
    XTA $\leftarrow$ vlcwrapper(stream).start(XTB.IP)
    XTA                ▷ use RTP/MPEG, deactivate transcoding
end procedure
```

Algorithm 3 Generating MOS

```
 1: procedure REFERENCE                        ▷ reference PSNR
 2:     yuv $\leftarrow$ decodeYuv(file)
 3:     craw $\leftarrow$ compRawVideo(yuv, fps, false)
 4:     refmp4 $\leftarrow$ mp4(craw)          ▷ hint RTP transport track
 5:     yuvMp4 $\leftarrow$ decodeYuv(refmp4)
 6:     refPsnr $\leftarrow$ psrn(yuvMp4)
         STORE(refPsnr)
 7: end procedure
 8: procedure MOS                 ▷ streamed vs. reference PSNR
 9:     yuvStream $\leftarrow$ decodeYuv(streamedMp4)
10:     streamPsnr $\leftarrow$ psrn(yuvStream)
         STORE(streamPsnr)
         GENERATEMOS(refPsnr, streamPsnr)
11: end procedure
```

IV. PRELIMINARY EVALUATION

Using the network configuration outline in Fig.2 we created a brute force analysis of media streaming characteristics using MOS for six network paths with the packet loss rates; Path 1=0.5%, Path 2=1.0%, Path 3=1.5%, Path 4=3.0%, Path 5=2.5% and Path 6=2.0%. The brute force analysis involved the streaming of the Akiyo video sample at 15, 30 and 60 frames per second at resolutions of 480p, 720p and 1080p for each of the six paths. This created a baseline brute force analysis of 54 separate tests illustrated in Fig.3.

Our results found that the overall maximum MOS was achieved for path 1 which had the lowest packet loss rate at 0.5%. Path 4 generated the lowest maximum MOS, 2.82, and experienced the largest packet loss rate, 3.0%. While the highest and lowest generated path MOSs align with their packet loss rates the distribution of MOSs in Fig.3 show the association is not tight, making the path selection decision more challenging.

The policy described in Algorithm 4 attempts to use a path climbing approach to optimize the video stream. In this experiment, Path 5 is arbitrarily selected and the MOS for a resolution of 480p at 15fps is calculated as 2.7. Path 5 is implemented by the adaptive path selection policy as optimal with the corresponding frame rate and resolution. In the 2nd cycle the Akiyo video sample is streamed with a resolution of 720p at 15fps. As the MOS attained is 1.58 and less than the previous optimal no path re-selection occurs and the stream configuration of 480p at 15fps remains optimal. Given that a resolution increase degraded the MOS, policy decides to increase the frame rate to 30fps and return the resolution to 480p. This configuration results in a MOS of 3.04. As this value exceeds the current maximum MOS the optimal stream configuration is updated to 480p at 30fps. No path re-selection

Algorithm 4 MOS/Active Path Policy

```
procedure MATCH                                          ▷ Match state
    $e^o_m \leftarrow e^i_m.mos$
    $e^o_m \leftarrow e^i_m.vidParam$
end procedure
procedure ESTABLISH                                      ▷ Establish state
    $a_p \leftarrow ctxt(path)$
    $p_r \leftarrow ctxt(prevRes)$
    $p_f \leftarrow ctxt(prevFps)$
    if $e^i_e.mos > ctxt(maxMos)$ then
        $ctxt(maxMos) \leftarrow e^i_e.mos$
        $ctxt(optRes)$
        $ctxt(optFps)$
    end if
end procedure
procedure DECIDE                                         ▷ Decide state
    if $p_r = e^i_d.vidParam.res$ then
        if $p_f = e^i_d.vidParam.fps$ then
            $a_p \leftarrow rand(path)$
            $e^o_d \leftarrow a_p$
        end if
    end if
    if $e^i_d.mos >= ctxt(maxMos)$ then
        if $e^i_d.vidParam.res$ != $maxRes$ then
            $e^o_d \leftarrow increaseRes$
        end if
    else if $e^i_d.vidParam.fps$ != $maxFps$ then
        $e^o_d \leftarrow increaseFps$
    end if
end procedure
procedure ACT                                            ▷ Act state
    if $e^o_d.a_p$ then
        $e^o_a \leftarrow genCmd(e^o_d.a_p)$
    else if $e^o_d.increaseRes$ then
        $e^o_a \leftarrow genCmd(e^o_d.increaseRes)$
    else if $e^o_d.increaseFps$ then
        $e^o_a \leftarrow genCmd(e^o_d.increaseFps)$
    end if
end procedure
```

is required, Path 5 is still considered optimal. Paths 3 and 1 are then considered. Path 1 with a resolution of 480p and a frame rate of 60fps results in a MOS of 3.55. This value exceeds the current optimal MOS. The optimal path is reconfigured to 1 and the optimal stream configuration is set to 480p at 60fps. The final remaining paths do not alter the optimal path selection. Fig.4 shows the MOS for each tested path along side the MOS generated from the optimal path.

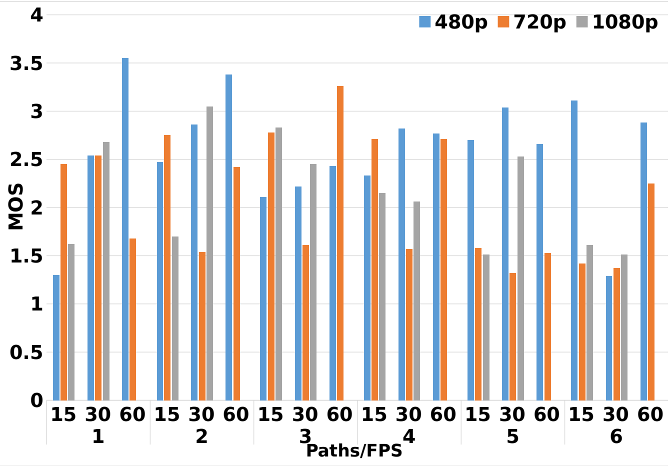

Fig. 3: Brute force analysis results

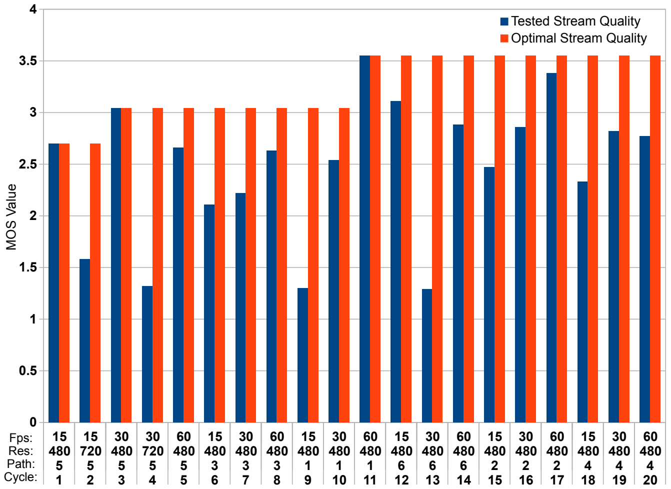

Fig. 4: Current and Optimal Stream Comparison

V. CONCLUSIONS

In this paper we have discussed the importance of current OTT and network-centric video optimization strategies while motivating the need for a closed control, policy-driven approach for 5G operators to mitigate between OTT, other service offerings, and the available video optimization techniques. In this context, we describe the starting point for the role of adaptive policy, in service assurance, for video quality control through the evaluation of network resources using MOS.

We believe policy controlled closed control loop mechanisms that can steer video optimization frameworks are important when considering new technologies such as MEC and VNFs. The network resource evaluation described in this paper, although preliminary, is one of many ways adaptive policy can be incorporated into specific optimization strategies and with the inclusion of APEX in the Open Networking Automation Platform (ONAP) we may see adaptive policy play a bigger role in 5G networks. All components, scripts, and other artifacts for the experiment detailed in this paper are available online [19]. The provided instructions allow any interested party to run the experiment described in this paper.

In future work, we will extend our initial setup to realize more complex 5G network use cases, along with policies that allow for effective and efficient video service assurance. We will continue to publish all testbeds online, to promote a wider discussion of video service assurance.

ACKNOWLEDGMENTS

This work is partly funded by the European Commission via the ARCFIRE project (Grant 687871) under the H2020 program. This work was also supported in part by Irish Research Council Enterprise Partnership Scheme Postgraduate Scholarship 2017 under Project EPSPG/2017/275.

REFERENCES

[1] Ericsson. (2017, Nov.) The Ericsson Mobility Report. [Online]. Available: https://www.ericsson.com/en/mobility-report

[2] ——. (2017, Nov.) Mobile traffic analysis by application. [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/november-2017/mobile-traffic-analysis-by-application

[3] D. C. Reeve, "A New Blueprint for Network QoS," Ph.D. dissertation, Computing Laboratory, University of Kent, Canterbury, Kent, UK, August 2003. [Online]. Available: http://www.cs.kent.ac.uk/pubs/2003/1892

[4] Ericsson. (2017, Nov.) Enhancing the event experience. [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/november-2017/enhancing-the-event-experience

[5] L. Xiang, D. Wing Kwan Ng, T. Islam, R. Schober, V. W. S. Wong, and J. Wang, "Cross-layer optimization of fast video delivery in cache- and buffer-enabled relaying networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11 366–11 382, Dec. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7959207/

[6] H. Viswanathan, D. De Vleeschauwer, A. Beck, S. Benno, R. B. Miller, G. Li, M. M. Clougherty, and D. C. Robinson, "Mobile video optimization at the base station: Adaptive guaranteed bit rate for http adaptive streaming," *Bell Labs Technical Journal*, vol. 18, no. 2, pp. 157–174, Sep. 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6772140/

[7] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hofeld, and P. Tran-Gia, "A survey on quality of experience of http adaptive streaming," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 469–492, Firstquarter 2015.

[8] N. Bouten, S. Latr, J. Famaey, W. V. Leekwijck, and F. D. Turck, "In-network quality optimization for adaptive video streaming services," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2281–2293, Dec 2014.

[9] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck, "Qoe-driven rate adaptation heuristic for fair adaptive video streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 2, pp. 28:1–28:24, Oct. 2015. [Online]. Available: http://doi.acm.org/10.1145/2818361

[10] ISO/IEC, "Information technology – dynamic adaptive streaming over http (dash) – part 5: Server and network assisted dash (sand)," Edition 1, Feb. 2017. [Online]. Available: https://www.iso.org/standard/69079.html

[11] TNO. (2017) MPEG-DASH SAND. [Online]. Available: https://tnomedialab.github.io/sand/

[12] A. Mehrabi, M. Siekkinen, and A. Ylä-Jääski, "Joint optimization of qoe and fairness through network assisted adaptive mobile video streaming," in *13th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/8115778/

[13] L. Fallon, J. Keeney, and S. van der Meer, "Using the compa autonomous architecture for mobile network security," in *2017 IFIP/IEEE International Symposium on Integrated Network Management (IM 2017)*, May 2017. [Online]. Available: https://www.researchgate.net/publication/317014658_Using_the_COMPA_Autonomous_Architecture_for_Mobile_Network_Security

[14] L. Fallon, S. van der Meer, and J. Keeney, "Apex: An engine for dynamic adaptive policy execution," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 699–702.

[15] J. McNamara, J. Keeney, L. Fallon, S. van der Meer, and E. Fallon, "A testbed for policy driven closed loop network management," in *2018 IEEE/IFIP Network Operations and Management Symposium (NOMS 2017)*, Apr. 2018.

[16] Ericsson. (2018) Apex documentation. [Online]. Available: https://ericsson.github.io/apex-docs/

[17] R. Schatz, S. Egger, and A. Platzer, "Poor, good enough or even better? bridging the gap between acceptability and qoe of mobile broadband data services," in *2011 IEEE International Conference on Communications (ICC)*, June 2011, pp. 1–6.

[18] S. van der Meer, J. Keeney, and L. Fallon, "Dynamically adaptive policies for dynamically adaptive telecommunications networks," in *11th International Conference on Network and Service Management (CNSM)*, Nov. 2015. [Online]. Available: https://www.researchgate.net/publication/282576518_Dynamically_Adaptive_Policies_for_Dynamically_Adaptive_Telecommunications_Networks

[19] Ericsson. (2018) Apex code repo on github. [Online]. Available: https://github.com/Ericsson/apex