Self-learning Prioritization of End-User Services to Eliminate Overload Situations

Marek Kocan
SilverEngine GmbH, Munich, Germany
marek.kocan@silverengine.de

***Abstract* – Telecommunication network providers employ various strategies to protect from and to mitigate overload situations caused by signaling storms to minimize end-user service loss. The common approach is the deployment of additional hardware above the engineered capacity combined with resource intensive operational recovery procedures. While signaling storms are relatively rare in its occurrence, they usually have serious consequences – loss of end-user service resulting in negative publicity and business damage. Adaptive overload management emphasizes end-user service as its primary goal in addition to the protection of a network function. The communication dialogs necessary to establish the end-user service are automatically detected and the involved requests are appropriately prioritized. Combining these two processes, the probability of service establishment and its eventual restoration is increased, which contributes to the reduction of overload situation as more end-users can receive its service. Self-learning request prioritization can reduce the time and complexity needed to restore service for all end-users during signaling storms. Through its automatic and self-learning operation it is suited for current and upcoming cloudified and 5G core networks.**

Keywords* – *overload; overload protection; robustness; signaling storm; adaptive prioritization; automatic network recovery; machine learning

I. INTRODUCTION

Telecommunication networks provide a variety of services to devices and applications that are consumed by end-users. End-user may be a human person using its (mobile) device to use internet-based services but also a software application or internet of things (IoT) device running without any human interaction. This experience is primarily a seamless one, the end-user is not aware of the technical details of the services, given the services are available. These services (voice, data, voice over LTE, voice over WIFI…) are delivered through a wide range of network functions (Fig. 1). For each service the communication may be different and may involve additional or less network functions (NF).

The approach of end-user service prioritization focuses on control plane network functions that hold the volatile end-user state information as well as semi-permanent subscription data necessary to provide the required services. Failure scenarios, like loss of (a portion of) end-user state information may trigger signaling storms within the network generating stress on network functions. Self-learning end-user service prioritization can, without significant investment into NF over-capacity and complex operational procedures, increase the service restoration rate and reduce the overall recovery time in overload situations.

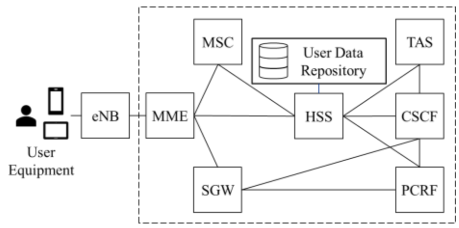

Figure 1. Interaction of network functions to deliver end-user service

II. THE CHALLENGE – SIGNALING STORMS

The telecommunication network is largely a multivendor environment. Different network functions from various vendors interact via standardized interfaces with each other. Decomposition into network functions supports and promotes innovation and competition. Vendors are competing to provide the best features, efficient resource utilization and carrier-grade availability. The downside of the decomposition is vendors focused attention on a specific network function (“silo” view). The network function itself is made highly available with its own approach to overload management (robustness) – primarily protecting itself against “misbehavior” of other NFs (vendors). Such effort is costly and uncoordinated between NFs. Considering a single NF in isolation may rationalize this strategy from vendors perspective, however from the telecommunication network operator point of view this is an unsatisfactory and an insufficient solution.

Telecommunication network operator’s motivation and goal is to provide end-user service availability (and not a single NF availability in isolation) where all interacting network functions are included. This can be achieved only if the complete call flow (e.g. for device attach, service registration, session or service request) across the whole chain of different network functions is successfully executed (Fig. 2). Within a single end-user service call flow (e.g. device attach) certain NFs may be invoked multiple times.

Even if signaling storms are rare in its occurrence, they still happen more often that one would like to experience. NF overload protection is the foundation to master this kind of events. Overloaded network function performs individual decisions which requests will be processed, and which requests will be rejected (negative response). But as previously stated, all (call flow) requests across all network functions must succeed to provide service to the end-user. With NF silo overload decision behavior this can hardly be achieved, and service is provided either with low probability or not at all. It is just a coincidence whether end-user will obtain its service or not.

Common approach to mitigate negative consequences is to engineer network functions capacity for the worse possible scenario – deploying enough capacity to manage (the expected) traffic storms and thus avoid – if possible – overload conditions in the network in the first place. This approach has still the weakness that NF dimensioning is a theoretical exercise and the traffic in the network is not fully under network function control or under control of the planning personal. Rather it is driven by end-user devices, applications and failure modes of individual network functions (incl. faulty behavior). For this reason, additional operational procedures are developed to recover network with manually controlled procedures. Despite all the engineering efforts the risk remains, traffic storms are (still) possible and may emerge by factors higher than the planned NF over-capacity. The economic implications of such approach are significant – higher capital investment as well as increased ongoing operational costs.

III. PRIORITIZATION OF END-USER SERVICE

Carrier-grade network functions must provide overload protection on its external interfaces. A typical overload management would monitor selected performance indicators about its performance and utilization. If the utilization or key performance indicators (KPI) would cross predefined thresholds, the NF overload protection would be activated. Basic implementation would process as many incoming requests as it can successfully serve within a rolling time window while adhering to KPIs. The additional, randomly selected requests would be rejected with negative response.

The responsibility to recover from the negative responses is passed onto the NF client. The effect of this behavior is that:

- The NF client has very limited options and ability to recover (its capacity is also limited) and is likely to pass the failure back in the call flow chain.
- Rejection due to overload results in a service loss for end-user.
- Rejection of arbitrary requests decreases the prospects that an end-user will receive its service.
- Every rejection generates a retry process within the network which may eventually amplify the overload condition even further (end-users having service may be dropped off the network as well).

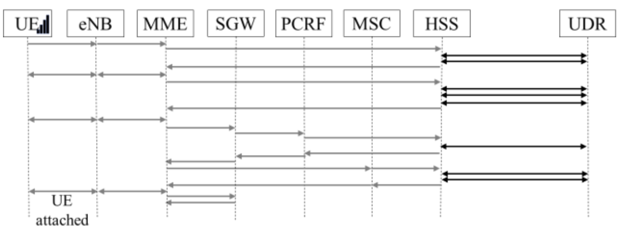

Figure 2. Exemplary attach call flow for UDR

An improvement to the situation can be achieved by supporting request priorities on network functions external interface. The NF client can, to a limited extent, indicate to NF service the priority of its request. NF service can make an informed decision to satisfy client priorities and reject lower priority requests initially during overload condition. This has impacts on NF clients that will need to implement the prioritization of its requests towards NF service fitting into the end-user call flow scenarios.

The generic principle of the end-user service prioritization is exemplary explained on a device network attach scenario. The Fig. 2 is providing such an illustrative example of a network attach call flow. We will focus on User Data Repository (UDR) network function [1] and its interaction with Home Subscriber Server (HSS). HSS is the NF client and UDR is the NF service. There are several key points that can be observed:

(a) The end-user service establishment requires several interactions to occur across multiple network functions.

(b) All these interactions must succeed to deliver the service to end-user.

(c) HSS NF is invoked several times in an end-user call flow (not necessarily triggered by the same NF) prior to service establishment.

HSS using UDR is a data less application. In general, it does not store states between multiple invocations of its external interface. HSS can adjust priorities for UDR requests within a single external invocation. The first UDR request with lower priority is followed by additional requests with increasing priority and finally the last request with relatively the highest priority. With staggered request priority within an HSS single external invocation, UDR being in overload, the first request with low priority would be more likely rejected than the requests with higher priority. This has the benefit that once the initial request (with lower priority) succeeds the next requests are more likely to succeed as well. Such behavior attempts to utilize the overloaded resources more efficiently and increases the probability of success for the particular HSS external interaction. However, overall end-user service establishment probability is not increased.

Enhancing the approach with end-user service prioritization requires HSS to prioritize UDR requests across multiple HSS external invocations (Fig. 3). The first HSS request in that call flow will have the lowest priority while the following requests towards UDR would be issued with increased priority. Likelihood of service establishment is increased under UDR overload conditions compared to the previous approach. The expected consequence is reduced traffic and lowered resources consumption.

It is technically feasible, for a network function (e.g. HSS), to track internally end-to-end dialogs and determine appropriate request priorities. However, this is incomplete and complex:

- The prioritization is focused on single network function (e.g. HSS) impacting only a part of the end-user service delivery. The NF client may not be aware of other NF client's functionality which may cause additional interaction with the same NF service (e.g. UDR) which was not engineered in.
- The end-user call flow will vary based on network service and its characteristics itself and may not be constant (operators may enable, disable various features for groups of end-users).
- The NF (e.g. HSS) needs to store additional (highly volatile) internal state across interactions and thus introduce additional transient state information. Additional state information would eliminate data-less simplicity of NF client (i.e. necessary state replication across NF client instances).
- The prioritization depends heavily on deep engineering and implementation knowledge of the network function (which may change over time) and requires careful complex design consideration and on-premise configurability. Utilization of various NF client's features may alter the call flows and service invocations that must be considered in the implementation phase (increasing complexity and error probability).

The described approach can improve the situation at the costs of complexity. Self-learning end-user service prioritization provides a simpler and more generic approach.

A. End-user state information availability

Telecommunication network is providing the same services to a large group of end-users. The same (similar) communication patterns in the core network are repeated over and over. The information about provided service is represented as a state information that can be a registration state, attachment state or other kind of session state information. Initially, when a service is first-time requested by end-user, an initial state information must be created (registration or session data record). This operation is the most expensive one as it involves lot of interdependent communication across a variety of NFs (authentication, collecting service information, provisioned subscription data, applying policies…). The resource demands for initial end-user service establishment are very high. Once the service is established usually only smaller updates (e.g. service requests, location updates) are necessary to keep the service connected.

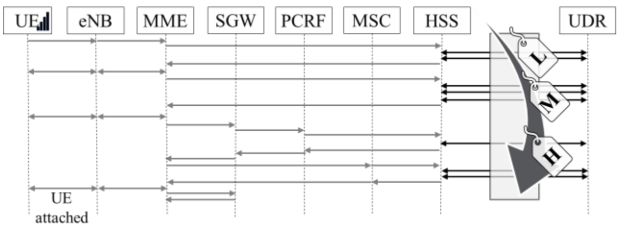

Figure 3. Exemplary attach call flow with UDR prioritization

There are many potential triggers of overload within the core network. However, majority is triggered by a failure scenario or application faults on the end-user devices. This may be a failure in radio network, failure in network connectivity, failure of network function instance or even a disaster situation (e.g. power outage, flood, construction work). Loss of or invalidation of the respective end-user state information within the network results in service interruption and end-user disconnection. End-users losing their service will attempt to re-establish the service as soon as possible. If many end-users are impacted there is a significant risk of overload to occur in the network.

The state information within the network represents the vulnerable resource that needs dedicated protection. In the exemplary scenario (Fig. 3) pressure is generated on semi-permanent (subscription information) data store (data needed to re-establish the service). The request priorities must be determined outside of the NF client and the prioritization of end-user service has to occur close to the NF service holding the state information (Fig. 4). Then it is possible to comprehend all interactions that are necessary to establish a service. This has the benefits of reducing the client complexity and increasing the flexibility in the deployment.

B. Enhancing prioritization with automatic learning

The amount of services provided by the network with its variations (service features) is limited. There are (large) groups of end-users using those services and the resulting communication patterns are following a very similar lifecycle – establish service (e.g. attach), update service information (e.g. service request) and dismantle service (e.g. detach). As these patterns repeat in the network, it is possible to analyze the traffic towards NF service and learn the interaction dialog patterns (end-user call flows) automatically in real-time directly from the network.

Fig. 5 is showing a logical diagram enhancing network function service overload management. All requests send towards the NF service would be analyzed and eventually modified. There are two main roles of such functionality:

(1) **Learning role** – scanning the incoming requests and learning end-user dialog sequences.

(2) **Prioritizing role** – labeling the requests with

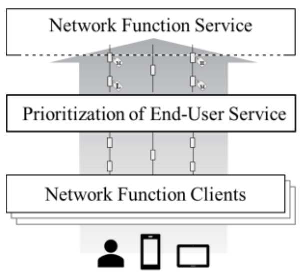

Figure 4. Logical diagram enhancing NF service overload protection

priority based on the learned end-user dialogs.

The first role (1) can be defined as the learning phase and is consisting of a sequence of activities:

- Request **sampling** – collecting request samples for analysis.
- **Extraction** of key features from the requests that are considered as important distinguishing factors. Using those factors end-user interaction dialogs are detected in the stream of requests.
- **Scoring** of the detected end-user dialogs and establishing end-user dialog patterns. Dialog patterns include the requests expected for end-user interaction including their priority based on the request position in the dialog sequence.
- The detected dialog patterns are then merged with existing known dialog patterns to continually **learn** active interaction patterns.

The learning phase is proceeding independently of the main traffic stream and their results can be periodically merged with the so far learned dialog patterns. Regular or constant learning phase will ensure that it can automatically adapt to possible changes in the network that may be caused by software updates or by new feature/service introductions. This functionality also significantly reduces the engineering efforts required for configuration of such functionality – it will simply be automatically configured for the needs in the network by itself.

The second role (2) of the service can be defined as the prioritization phase. In this phase each incoming request is **matched** against the active dialog patterns and if a match is found a **priority label** is attached to the request. The request with the priority is send to the NF service that uses this information in its overload protection mechanism. This way it will be ensured that the priority is in line with the end-user service delivery goals – the more progressed the end-user dialog (call flow) is the higher priority will be assigned.

C. Measurements

To validate the benefits and impacts of the end-user service prioritization the self-learning prioritization has been developed and tested in various scenarios. The setup consists of three components – a client, a prioritization service and the protected NF service. The client represents the end-users (UEs) demanding service and can simulate a

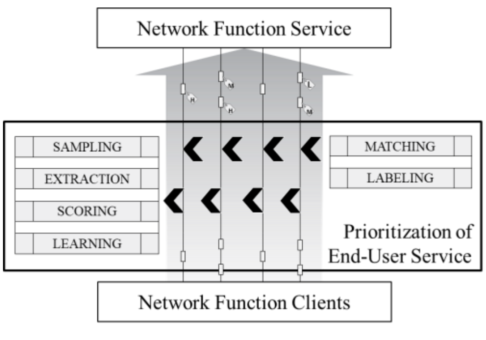

Figure 5. Logical diagram enhancing NF service overload protection

configurable number of end-users. The client sends all requests to prioritization service (without priorities). The prioritization service implements the end-user dialog detection mechanism (the learning) and the prioritization of the incoming requests which are forwarded towards the NF service for processing. NF service provides the protected network function service that is used by the client. It can support a priority driven overload protection and rejects the lower priority requests prior to higher priority requests (based on overload level).

The tests focused on comparing three specific scenarios with increasing number of end-users:

- Scenario (A) where NF service had enough capacity to process the peak traffic originated from signaling storm without NF service entering overload situation. This scenario was used to baseline maximum peak capacity.
- Scenario (B) where NF service has limited capacity and supports overload protection without priority consideration.
- Scenario (C) where NF service has limited capacity (as in previous scenario) implementing and using request priority to accept or reject requests in overload situation.

The initial situation for each test scenario is the same – loss of all end-user states (services) – all end-user services are disconnected. This represents the worst-case scenario where a traffic storm reaches the network function service requesting the service for all end-users nearly simultaneously. For scenarios with overload protection (with and without prioritization) the NF service (limited) capacity was exactly the same.

For an end-user to *attach* successfully to the network (service establishment) a sequence of requests must succeed on NF service. Once end-user is attached to the network a periodic *service request* is issued for each. If the *attach* use case fails it is aggressively retried after a small delay. If *service request* use case fails, the end-user loses its service and *attach* use case is initiated again. Each request to NF is accompanied by a timeout and its expiration leads to a use case failure with a subsequent retry starting again with *attach* use case. The *attach* use case produces factor 4 more requests than the *service request* use case.

The tests were executed with increasing number of end-users. Initial test with specific number of end-users was baselined at 100%. The number of end-users has been then increased by a factor up to 6 (600% of the initial number of end-users). Each test generated certain level of overload on the network function service (except the baseline scenario tests measuring the maximum peak capacity without overload).

For each number of end-users (from 100% to 600%) a baseline peak capacity was determined (scenario (A)). This is the capacity that is needed to be supported by NF service in order to avoid overload situation at all (i.e. to deploy additional hardware accommodating the expected signaling storms). This peak capacity is shown in Fig. 6 as gray bars on the right vertical axis. If the NF would have to sustain the load from the 100% of end-users (initial number of end-users) without overload the NF would have to support 10x higher peak capacity (10 instances of the same NF service); for 6 times (600%) more end-users it would require even more than 30 times more capacity to avoid overload situation (compared to the limited capacity for scenarios (B) and (C)).

The Fig. 6 shows two curves (on a logarithmic scale – left axis) monitoring the average network *attach* success ratio representing successful service establishment – for overload protection with prioritization (black squares – scenario (C)) and without prioritization (dark gray triangles – scenario (B)). The value represents the likelihood that an attempt of end-user to *attach* to network will succeed. At the start of the test the ratio is lower and as the recovery progresses it is increasing. It can be observed that even with relatively low overload level the probability of network *attach* with end-user service prioritization is higher than without prioritization. This difference is even more significant with increasing overload on the NF service. End-user service prioritization on NF increases the probability of service establishment significantly under extreme overload situations. The side-effect of the increased probability is that the strength of the signaling storm is being faster reduced and less and less *attach* attempts are being retried which speeds up the recovery process.

It is important to note that the scenario without prioritization (B) has not recovered for 300% of end-users and more. Only a minimal portion of end-users received service which was subsequently lost (failed *service request*). The (simulated) network never recovered in these test scenarios.

The ratio of recovery duration is shown in Fig. 7. The chart is showing the recovery duration compared with baseline scenario (peak capacity available without overload). It can be observed that overload protection with end-user prioritization recovers faster. Not only that, overload protection with priorities is able to recover from significantly higher overload situations. This is especially caused by the increased *attach* probability, it can recover even 6 times more end-users with the same capacity (average probability of a successful end-user network *attach* with prioritization is still 0.55% compared to 0.0001% without prioritization).

Self-learning end-user service prioritization brings major benefits in managing signaling storms:

- It increases the probability that end-user service can be established or preserved during overload situations.
- The overall traffic is reduced faster as more users have service recovered. The duration of the overall recovery is shorter.
- Its automatic and self-learning nature can recover the service without manual intervention.
- It can recover from higher levels of overload.
- The deployment costs are reduced (lower peak capacity required).
- It provides a safety net for unexpected events in the network.

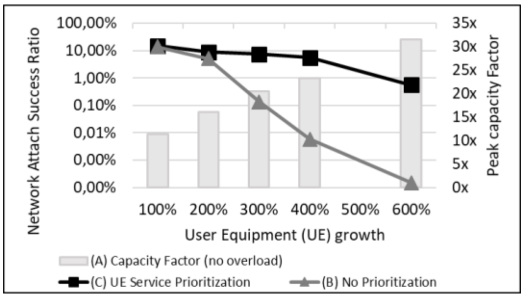

Figure 6. Network attach success ratio

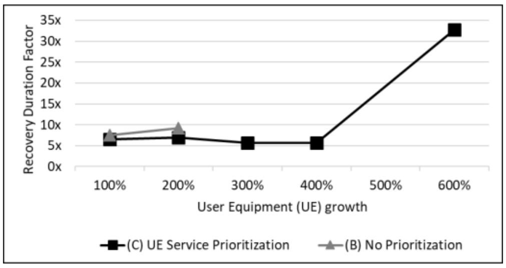

Figure 7. Recovery duration ratio of all UEs

IV. 5G CORE AND CLOUD

So far in our exemplary situation, the assumption has been that “somewhere” in the network end-user state is lost and a signaling storm is reaching UDR. UDR is the end-point that needs to deliver all the information that is needed to establish/restore services (UDR stores semi-permanent subscription and policy information that is the basis of majority of services in the network). But many of the current network functions also hold internal state information. For example, Mobile Management Entity (MME) holds registrations status, bearer information, IP Multimedia System (IMS) holds session information… The network functions are stateful and loss of any of this state may result in a signaling storm that impacts also all other network functions in the end to end call flow chain. State information and stateful network functions ideally require specific overload protection.

There are currently two significant developments in the telecommunication networks that are worth to analyze in terms of state information and the relevance of specific overload protection. The first area is related to network cloudification or Network Function Virtualization (NFV) of the core network. The second development is related to new 5G core network architecture and its service-based architecture.

A. *Cloudification of network functions*

NFV [2] is significantly changing the landscape of telecommunication networks. The cloudified network functions will have to adopt a different architecture – NFs must implement so called cloud native principles. Critical principle for NF cloudified architecture is the separation of state information from the NF business logic. The NF state information is being externalized and the network function itself is becoming stateless and thus simpler, easily scalable (elastic) and maintainable. Stateless network functions can process any incoming request as the data can be queried and updated in real time. Externalized data need to be made highly available, scalable and include appropriate overload protection to minimize the likelihood of state information loss.

B. *5G Core*

5G is introducing significant changes into the telecommunication core network. The standard leverages best practices from web scale companies and defines new interfaces. There are two main aspects relevant for managing state information – (1) disintegration of network function to separate data from business logic and (2) introduction of service-based architecture [3]. Disintegration of network functions is preferring stateless network functions with externalized state information. 5G defines [4] two data storage network functions that are available. Unified data repository (UDR) for storing structured data (e.g. subscriber profile, policy data, session information…) and unstructured data storage network function (UDSF) for storing unstructured private NF state information. Service based interfaces (SBI) [5] introduce a common interface basis for all network functions. All interfaces are based on the same protocol with different data information being exchanged. SBI is also introducing (an optional) request priority parameter that can be supported by network functions for overload management.

C. *Data Layer*

Based on the increasing needs for external data storage a data layer architecture has emerged. 5G service-based architecture [3] is introducing Unified Data Layer (UDL) that provides storage services for NFs, for semi-permanent (subscription, policy, context) information as well as for volatile user information (session data). In [6] the UDL has been refined into a Network Data Layer (NDL) with more detailed architecture and functional requirements. NDL combines the trends related to state/data externalization into a single network data layer. One of the NDL goals is increased stateful resiliency of network functions. The data layer is state-full and therefore its immediate ability to react to signaling storms (elasticity) is limited and it comes with significant costs (replication). Self-learning end-user service prioritization is a complement to the data layer. The end-user prioritization shall be overlaid (Fig. 8) on top of NDL external interfaces. NDL must support and respect request priorities on its external interface as part of NDL overload protection. This way, the state information would be appropriately protected even in overload – signaling storm – situations while state-less network functions can elastically adapt and accommodate to incoming traffic.

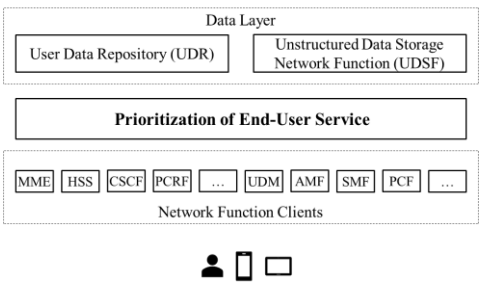

Figure 8. Enhancing data layer robustness in 4G and 5G networks

V. Conclusion

Telecommunication network operators expect that end-users will receive their service whenever they need and thus look for solutions how to improve the service delivery even in imperfect network conditions. Loss of (access to) state information leads to loss of end-user service and attempts to re-establish it as soon as possible. Signaling storms cannot be excluded in current and even in future networks. Introduction of new technology comes with unknowns and risks. The behavior of network functions and new 5G devices will need to be learned and tuned. Internet of Things (IoT) market is growing and it is expected still to significantly grow. IoT devices reveal different traffic patterns than consumer traffic patterns. The growing number of different devices raises the risk of uncontrolled traffic storms and makes traffic capacity planning unreliable.

There are many unknowns and the networks must be prepared for it. The implemented self-learning end-user service prioritization can significantly improve robustness of telecommunication network and help to manage signaling storms. It serves as a defense against the unexpected and recovers customer service even under high overload conditions automatically. It reduces engineering and implementation costs needed by vendors and operators through its self-learning and automatic operation. It permits to reduce the dimensioned peak capacity of network functions (thus the capital and operational expenses) while providing faster recovery times in overload situations. The approach fits very well into existing telecommunication networks as well as it matches with new 5G and cloudified networks.

References

[1] 3GPP TS 23.335 – User Data Convergence: http://www.3gpp.org/DynaReport/23335.htm

[2] ETSI Network Functions Virtualisation; Architectural Framework: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf

[3] NGMN Alliance, D. Wang, TSun – Service-Based Architecture in 5G: https://www.ngmn.org/fileadmin/ngmn/content/downloads/Technical/2018/180119_NGMN_Service_Based_Architecture_in_5G_v1.0.pdf

[4] 3GPP TS 23.501 – System Architecture for the 5G System: http://www.3gpp.org/DynaReport/23501.htm

[5] 3GPP TS 29.500 – Technical Realization of Service Based Architecture: http://www.3gpp.org/DynaReport/29500.htm

[6] NGMN Alliance, A. Hörnes, S. Langer - A network data layer concept for the telco industry: https://www.ngmn.org/fileadmin/ngmn/content/downloads/Technical/2018/180831_NDL_White_Paper_v1.0.pdf

[7] B. Beyer, Ch. Jones, J. Petoff, N. Murphy; Google: Site Reliability Engineering, chapter Handling Overload by A. Cuervo and S. Chavis: https://landing.google.com/sre/books/

[8] D. Hixson, K. Guliani, Capacity Planning, ;login:, vol. 40, no. 1, February 2015, https://www.usenix.org/publications/login/feb15/capacity-planning

[9] J. Hamilton: On Designing and Deploying Internet-Scale Services, Proceedings of the 21st Large Installation System Administration Conference, November 2007: https://www.usenix.org/legacy/event/lisa07/tech/hamilton.html