The Troubled Journey of QoS: from ATM to Content Networking, Edge-Computing and Distributed Internet Governance

George Pavlou, Ioannis Psaras

University College London

Abstract

Network Quality of Service (QoS) and the associated user Quality of Experience (QoE) have always been the networking "holy grail" and have been sought after through various different approaches and networking technologies over the last decades. Despite substantial amounts of effort invested in the area, there has been very little actual deployment of mechanisms to guarantee QoS in the Internet. As a result, the Internet is largely operating on a "best effort" basis in terms of QoS. Here, we attempt a historical overview in order to better understand how we got to the point where we are today and consider the evolution of QoS/QoE in the future.

As we move towards more demanding networking environments where enormous amounts of data is produced at the edge of the network (*e.g.,* from IoT devices), computation will also need to migrate to the edge in order to guarantee QoS. In turn, we argue that distributed computing at the edge of the network will inevitably require infrastructure decentralisation. That said, *trust* to the infrastructure provider is more difficult to guarantee and new components need to be incorporated into the Internet landscape in order to be able to support emerging applications, but also achieve acceptable service quality.

We start from the first steps of ATM and related IP-based technologies, we consider recent proposals for content-oriented and Information-Centric Net-

Email address: {g.pavlou, i.psaras}@ucl.ac.uk (George Pavlou, Ioannis Psaras)

working, mobile edge and fog computing, and finally we see how distributed Internet governance through Distributed Ledger Technology and blockchains can influence QoS in future networks.

Keywords: QoS, QoE, mobile edge computing, fog computing, distributed ledger technology, blockchain

---

1. 1980s: The First Steps

Back in the 1980s, when the Internet was still essentially a research network, there was a view that the future global networking technology would emerge from the telecommunications world through the evolution of the telephone network towards the multi-service network of the future. Digital transmission infrastructure in the form of SDH/SONET allowed the introduction of the Integrated Services Digital Network (ISDN) in the late 1980s as the first form of integrated infrastructure for voice, narrowband video and data, with the aim to evolve towards Broadband ISDN in the 1990s. This evolution, or more accurately revolution, would come through the universal introduction of Asynchronous Transfer Mode (ATM) [1] as the underlying packet-based network technology.

The key design principle behind ATM was the reservation of network resources per micro-flow, i.e. application-to-application flow, through end-to-end network signalling. This approach has its roots to the Public Switched Telephone Network (PSTN), where system 7 signalling (SS7) reserves two 64Kbps channels, one in each direction, in order to carry the encoded voice of a telephone call. The micro-flow resource reservation approach works well in the PSTN where every switch deals with exactly the same type of resource, i.e., exactly 64Kbps for every channel. However, in the context of ATM-based multi-service networks, applications could reserve any amount of bandwidth according to their needs and switches would have to deal with a very large number of highly different resource allocations. Despite extensive research and development efforts in this area at the time, it has not been definitively shown that core network switches could cope successfully with a potentially very high number of concurrent number of resource reservations by applications with widely different characteristics.

2. 1990s-mid 2000s: Attempting to Scale Up (and largely failing)

The advent of the World-Wide Web (WWW) in the early 1990s was the key evolution that changed the networking landscape and catapulted the Internet to the global multiservice network and the substrate of the current information infrastructure. Subsequently, there has been interest to offer services with guaranteed QoS/QoE over the Internet Protocol (IP) and the first take resulted in the Integrated Services (IntServ) framework [2], which was conceived as a relatively lightweight ATM-like IP technology. The approach chosen was to still reserve resources per micro flow through signalling in a similar fashion to ATM, and suffered from the same scalability problems for core network routers, so it saw no deployment. IntServ was followed in the late 1990s/early 2000s by the much more scalable Differentiated Services (DiffServ) framework [3], in which off-line reservations were done through provisioning for a limited number of classes of service, i.e. essentially for aggregate macro-flows. In DiffServ, customers establish Service Level Agreements (SLAs) with a provider beforehand and the provider provisions the network for the next period based on the technical characteristics of the SLAs i.e. the Service Level Specifications (SLSs) [4]. This framework found some use and is still in use today, mainly for corporate customers, but it was another QoS/QoE technology that was not as widely used as it was originally anticipated.

A marked difference in the direction and search for QoS/QoE took place in the late 1990s through the extensive use of network traffic engineering [5]. Instead of reserving resources to provide the desired quality of service to users/applications that are prepared to pay a premium, the target became to utilise the overall network resources more efficiently though traffic engineering, resulting in better end-to-end delay, eliminating packet loss and resulting effectively in better QoE for all applications, i.e. a "better than best effort" service. This was possible through the advent and wide deployment of Multi-Protocol Label Switching (MPLS) [6], which provided support for explicit routing with potentially multiple paths from ingress to egress nodes, allowing to spread traffic evenly through suitable routing plans and result in relatively uniform maximum link utilisation (MLU). The traffic-engineered routing plans were produced based on the anticipated traffic, i.e. the traffic matrix, over the next provisioning period which was produced based on measurements and gave rise to the advent of technologies such as NetFlow [7] which monitor and report/store all flows in a provider network.

From the mid-2000s onwards, video streaming started dominating the Internet and soon became the largest component of the overall traffic and kept growing continuously, with this tendency continuing today and predicted to continue increasing exponentially. The need to deliver video content with reduced latency and reduced network cost in terms of traffic carried by network links gave rise to Content Distribution Networks (CDNs), which placed content objects in various locations at the edge of networks and applications provided redirection so that the closest copy to a requesting user is accessed. This technology provides some form of QoS/QoE to the consuming user, and most importantly, it keeps the global network load manageable given the amount of video content that is continuously downloaded/streamed. In parallel, there has been research on in-network adaptation according to the state of the network based on video stream meta-data and on media-aware routing.

3. Current Practices: Network Difficult to Tame - Bring Content Closer to Reduce Latency

Given the fact that the Internet was increasingly being used for multimedia object access, researchers in the mid-to-late 2000s came up independently with similar ideas of taking the content-aware routing and future CDNs to a radical conclusion: they suggested making the network a global CDN which would route packets based on content IDs, instead of network addresses [8], [9], [10]. Given that content is addressed explicitly (as opposed to addressing the end-hosts of the communication channel through IP addresses), content could be cached anywhere and popular content would be eventually accessed from close locations by subsequent users, providing increased QoS/QoE [11], [12], [13]. This became the research area of Information Centric Networking (ICN) ([14], [15]) which is still going strong, although at this point, it is still difficult to think of global network-layer deployment given the massive investment in IPv4 (with some IPv6) in the current global Internet. Extensive research efforts in the area of ICN in-network caching [16], [17], [18], [19] and the native multicast supported by the ICN paradigm [9], [10] have shown significant QoS/QoE improvement, especially in the case of video delivery. This fact, together with the pressure that content publishers face with the increasing costs of timely video delivery and mobility support [20], [21] is foreseen to bring ICN closer to deployment [22]. Research on ICN in-network caching has also found applicability in other areas such as Telco CDNs [23] and some aspects of it may find deployment in 5G and the Internet of Things (IoT).

Another technology that emerged around 2010 and relates to some extent to QoS/QoE is Software Defined Networking (SDN) [24]. The key target of SDN is to decouple the control from the data plane and move it outside network devices to a logically centralised controller, providing programmability and cost reduction and allowing easy evolution of networking technologies and enabling innovation [25]. It is, to an extent, a similar idea to the PSTN Intelligent Network (IN), which also places control functionality in a centralised computing node outside the network and redirects intelligent call signalling there. SDN applies a similar principle to packet networks, *i.e.,* the fixed Internet and future 5G cellular networks. Forwarding rules installed by the controller in routers/switches guide data plane packet forwarding, but radical approaches might also be possible, where the forwarding rules are installed dynamically on a per flow basis based on the controller's view of the state of the network. In general, SDN promises to provide the ability to manage a network in near real-time and subsequently enable better, more fine-grained resource management, which should result in better QoS/QoE.

4. New Expectations and Requirements: Mobile Edge and Fog Computing

In parallel to the quest for QoS/QoE in communication networks in the last 3-4 decades, there has been a parallel shift in the computation model that supports the Internet. This move was mainly motivated by technology advances in the area of computing, computer architectures, but most importantly network virtualisation. The most recent evolution in the quest for QoS/QoE that emerged recently, *i.e.,* circa 2015 with the first studies appearing as early as 2009 [26], is Mobile Edge Computing (MEC) [27], [28], [29]. Despite not having agreed on a widely acceptable and usable QoS/QoE solution for the core Internet infrastructure, advances in the area of IoT are pushing the bar higher. Applications covering a wide spectrum from Virtual and Augmented Reality, to autonomous vehicles, swarms of Unmanned Aerial Vehicles (UAVs) for goods' delivery and flying taxis will require communication with the infrastructure in some form or another. The response times required in those cases is in the order of a few msecs in terms of round trip times (RTTs). Given that from the computation, hardware and software perspectives such applications are ready to be implemented, there is growing pressure towards the networking community to deliver in terms of standards and protocols for the communications part.

During the past twenty years (or so) we have been witnessing a continuous trend towards centralising Internet content delivery and application-oriented computation. Centralisation led to the development of massive scale datacentres (commonly referred to as "the cloud"), which is the place where 90% of user requests end up being executed. Although this trend served well the purpose of the Internet as we know it today, and was also inline with the demand of economies of scale, it is certainly not fit for purpose for future applications. The 5G architecture, currently under design, standardization and development, will demand for applications that respond in sub-msec latencies. Such applications cannot tolerate centralized computation siloed behind closed walls, typically in far-away data-centers.

Edge-/Fog-computing has been proposed as a complementary paradigm to the cloud [27], [29]. Its main idea is the de-centralisation of the cloud into multiple smaller scale computing devices, or cloudlets (ranging from mini-data centres to WiFi APs, to Raspberry Pis), which we refer to as "computation spots".

The expectation from the mobile edge-/fog-computing paradigm presents, to a certain extent, similarities to the caching era of the 90s. That is, similarly to the move from servers acting as the sole providers of static content to proxy caches, and more recently ubiquitous in-network caches (in the ICN area), the edge-/fog-computing paradigm is attempting a shift of computation closer to the users. By and large, the rationale behind deploying proxy caches was to reduce: *i)* response delay to end-users, *ii)* core-network traffic, and, *iii)* server load. Moving to the edge-/fog-computing paradigm, we could realistically argue that the motivation and expectation is roughly similar: move network functions and user-facing applications closer to the users to reduce response delay, network traffic, e.g., in case of heavy data that needs to be uploaded to the cloud [30], [31], and the ever-increasing stress placed on data centres [32].

Interestingly, there is one more dimension that can severely affect QoS and that needs to be addressed in case of edge-computing - as opposed to proxy-caching functionality. That is, resolving functions, i.e., computation functionality, on the fly is impossible to handle by the current DNS infrastructure. Functions can get instantiated and dissolved in the order of seconds and need to be resolved and executed in msecs, while normal DNS entries are updated a few orders of magnitude slower, *i.e.,* in the order of minutes, if not much longer. A computation-centric paradigm, where functions are packaged in lightweight virtualisation environments, e.g., Unikernels [33], [34], and are explicitly named with individual IDs - similar to the content objects discussed earlier - again presents several advantages. Requests carry input parameters for the function, while functions are stateless, meaning that they can be executed at any network node (see Serverless architecture [35]).

New computation-centric architectures are needed to address the need for fast resolution of network functions that are executed at the edges of the network [36], [37]. The ultimate purpose of such architectures is to alleviate the need for costly, in terms of RTTs, communication to DNS-like resolution services and therefore, improve end-user and application QoS.

5. The New Challenge: Removing Trust

There is one extra element worth raising in case of distributed edge and fog computing, which is likely to influence significantly the system performance in terms of QoS/QoE: **Internet infrastructure governance**, or in order words, *who is owning and managing the edge-computing infrastructure and who to trust when using Internet services* [38]. In the current Internet landscape, infrastructure is owned and operated (in obscure ways) by an oligopoly of "tech giants" - the likes of Google, Microsoft, Amazon, Facebook, Akamai etc. Although this model has worked relatively well in terms of performance[1] so far, it is questionable whether a similar model would work well in case of a distributed MEC network.

Firstly, it is easy to technically manage (and provide relatively acceptable QoS for) a few centralised computation factories, but almost impossible to manage and administer billions of computation spots. Secondly, innovation reaches a threshold difficult to pass when infrastructure stays behind closed doors. Thirdly, it is embarrassing to witness that after 40 years of intense research, engineering and development, if a link to the centralised infrastructure fails the most basic Internet functionalities break[2]. Instead, it is reasonable to assume that a Mobile Edge Computing infrastructure, which will be responsible for vital

[1] It has failed its users hugely in terms of privacy, for example.

[2] Amazon Web Services (AWS) holds a 40% share of the cloud-server market. When AWS's Virginia datacenter had an outage, a significant part of the web went offline [39].

applications, such as e.g. driving our cars, will be run by a multitude of players operating closer to the end-user. Clearly, QoS needs to come to the forefront of attention, as it will soon be responsible for extremely latency-sensitive applications that will manage central and in many cases life-threatening aspects of our lives, e.g., autonomous driving.

To remove centralisation is to remove the trust from the infrastructure provider. Whether one trusts the tech giants or not, by using their infrastructure one silently accepts that they will do their best to provide high performance, guarantee security and preserve privacy. Moving to a decentralised and distributed governance model and in order to achieve acceptable QoS, users will have to trust unknown operators/companies, effectively removing trust from the Internet ecosystem [40] of the last 20 years, when the current tech giants scaled up.

Recent advances in cryptography and Distributed Ledger Technology (*aka* blockchain) can be of significant contribution at this point. Distributed ledgers can track and record any transaction between any two entities in a trustless manner in an immutable history record. Security can be improved and privacy can be guaranteed. Despite performance issues of current blockchain systems [41], [42] (which are receiving significant attention at the moment and are expected to be solved in the near future), the important point is this: *computing infrastructure can be distributed to billions of computation spots, operated by anyone who can innovate on it, while governance can become decentralised and guarantee higher levels of security and privacy than the current infrastructure.* [43]

The impact of such developments on QoS is enormous. Distributed computing between trustless nodes is an enabler for ubiquitous computing where any spare computation cycle can be exploited to execute latency-sensitive applications in geographically-close locations. In turn, latency to reach the computation spot is reduced, execution time within the computation spot is kept to a minimum and applications are guaranteed to receive timely responses.

6. A timely use-case: Autonomous Vehicles

One of the big game changers in the automotive industry is the introduction of automated driving which will heavily rely on timely information and computation of results (e.g., traffic flow coordination to avoid accidents). It is expected that each car will generate approximately 4,000 Gigabytes of data per day,[3] a figure that will undoubtfully challenge future networks.

Automated driving relies on information such as very detailed maps about the vicinity combined with Machine Learning in order to avoid collisions. Vehicular systems will be equipped with sensors (e.g., front/rear/blind spot cameras, radar systems or brightness sensors) to monitor the environment. Autonomous cars will communicate with each other as well as with infrastructure components to share sensed information.

Processing such big amounts of data by the vehicles themselves may be impossible, while pushing everything up to the cloud requires excessive amounts of bandwidth [30], but most importantly induces prohibitive round-trip latencies [44], [45].

In contrast, a network that supports execution of in-network functions can aggregate vehicular sensor data at nodes with sufficient computational power placed at the edge of the network. Processing data closer to the required geolocation reduces latency and the load in the core network. Edge-network functions are used to process the incoming information and compute a detailed map of the vicinity. The returned result can be reused by all involved vehicles.

When a vehicle sends a request for a computation result, the network orchestrates the computation. This procedure includes splitting the task into sub-computations, scheduling and assigning these sub-tasks to suitable execution locations and integrating intermediate results into a final reply.

However, for such a system to be deployed, an efficient and secure payment system is essential and can determine its future success. In an open, non-walled

[3] https://newsroom.intel.com/editorials/krzanich-the-future-of-automated-driving/

garden cloud computing environment, execution nodes are owned by multiple stakeholders, while requestors do not know which nodes will execute their tasks and thus whom to pay in advance. What is more, even with this knowledge they do not want to pay for yet unfinished or unverified tasks. On the other hand, an execution node receiving a request does not want to use its resources without making sure that it will eventually receive the corresponding payment.

To make a payment system truly distributed, one needs to include result verification techniques to ensure its correctness. Thus, a distributed and secure payment system is needed in order to allow for the transfer of funds between mutually distrusting requesting and executing nodes. Requestors submit tasks to a blockchain and allow any node to claim the tasks for execution. The blockchain does not belong to any central entity and its integrity is assured by thousands of miners charging only a minimal fee [46]. When the computations are finished, the result is returned to the requestor and the executing node is paid for its work. Such solutions can leverage deposits, payment channels [47], smart contracts and *trusted execution environments* (TEEs) [48] to ensure proper behaviour of all parties involved without establishing any trust relation between them.

7. Concluding Remarks

Despite the extensive research and development efforts to build a Quality of Service framework for static communications in the core Internet, there has largely been very little consensus on the appropriate solution. As such, there is no widespread solution deployed to date.

At the same time, as the Internet grew in the last few decades, the infrastructure is required to adapt to a ubiquitous connectivity, communication and computation paradigm. New applications require stringent latencies at the edge of the network (where applications are mostly needed, i.e., in users' devices) and computation cannot be contained in remote data-centres anymore.

New developments in the areas of Information-Centric Networks, Mobile

Edge Computing, but also in the general area of Information Security have the potential to complement the needs and requirements of such applications. *The current challenge is therefore, to bridge the gap between the established Internet infrastructure and related protocols and the new development activities in the areas of security, privacy and distributed ledger technology.* The amalgamation of these areas at large can together provide the QoS expected by end-users and the industry investing in future technologies and applications.

References

[1] S. E. Minzer, Broadband isdn and asynchronous transfer mode (atm), IEEE Communications Magazine 27 (9) (1989) 17–24. doi:10.1109/35.35508.

[2] R. Braden, D. Clark, S. Shenker, Integrated services in the internet architecture: an overview, Request for Comments RFC 1663, Internet Engineering Task Force (IETF) (1989).

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An architecture for differentiated services, Request for Comments RFC 2475, Internet Engineering Task Force (IETF) (1998).

[4] P. Trimintzios, I. Andrikopoulos, G. Pavlou, P. Flegkas, D. Griffin, P. Georgatsos, D. Goderis, Y. T'Joens, L. Georgiadis, C. Jacquenet, R. Egan, A management and control architecture for providing ip differentiated services in mpls-based networks, IEEE Communications Magazine 39 (5) (2001) 80–88. doi:10.1109/35.920861.

[5] N. Wang, K. H. Ho, G. Pavlou, M. Howarth, An overview of routing optimization for internet traffic engineering, IEEE Communications Surveys Tutorials 10 (1) (2008) 36–56. doi:10.1109/COMST.2008.4483669.

[6] E. Rosen, A. Viswanathan, A. Callon, Multiprotocol label switching architecture, Request for Comments RFC 3031, Internet Engineering Task Force (IETF) (2001).

[7] B. Claise, Cisco systems netflow services export version 9, Request for Comments RFC 3954, Internet Engineering Task Force (IETF) (2004).

[8] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, I. Stoica, A data-oriented (and beyond) network architecture, in: Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '07, ACM, New York, NY, USA, 2007, pp. 181–192. doi:10.1145/1282380.1282402.
URL http://doi.acm.org/10.1145/1282380.1282402

[9] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, R. L. Braynard, Networking named content, in: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT '09, ACM, New York, NY, USA, 2009, pp. 1–12. doi:10.1145/1658939.1658941.
URL http://doi.acm.org/10.1145/1658939.1658941

[10] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, Named data networking, SIGCOMM Comput. Commun. Rev. 44 (3) (2014) 66–73. doi:10.1145/2656877.2656887.
URL http://doi.acm.org/10.1145/2656877.2656887

[11] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, P. Nikander, Lipsin: Line speed publish/subscribe inter-networking, in: Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication, SIGCOMM '09, ACM, New York, NY, USA, 2009, pp. 195–206. doi:10.1145/1592568.1592592.
URL http://doi.acm.org/10.1145/1592568.1592592

[12] W. K. Chai, N. Wang, I. Psaras, G. Pavlou, C. Wang, G. G. de Blas, F. J. Ramon-Salguero, L. Liang, S. Spirou, A. Beben, E. Hadjioannou, Curling: Content-ubiquitous resolution and delivery infrastructure for next-

generation services, IEEE Communications Magazine 49 (3) (2011) 112–120. doi:10.1109/MCOM.2011.5723808.

[13] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, H. Karl, Network of information (netinf) - an information-centric networking architecture, Comput. Commun. 36 (7) (2013) 721–735. doi:10.1016/j.comcom.2013.01.009.
URL http://dx.doi.org/10.1016/j.comcom.2013.01.009

[14] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, G. C. Polyzos, A survey of information-centric networking research, IEEE Communications Surveys Tutorials 16 (2) (2014) 1024–1049. doi:10.1109/SURV.2013.070813.00063.

[15] D. Kutscher, et al., Rfc 7927: Information-centric networking (icn) research challenges.
URL https://tools.ietf.org/html/rfc7927

[16] W. K. Chai, D. He, I. Psaras, G. Pavlou, Cache "less for more" in information-centric networks, in: Proceedings of the 11th International IFIP TC 6 Conference on Networking - Volume Part I, IFIP'12, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 27–40. doi:10.1007/978-3-642-30045-5_3.
URL http://dx.doi.org/10.1007/978-3-642-30045-5_3

[17] I. Psaras, W. K. Chai, G. Pavlou, Probabilistic in-network caching for information-centric networks, in: Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking, ICN '12, ACM, New York, NY, USA, 2012, pp. 55–60. doi:10.1145/2342488.2342501.
URL http://doi.acm.org/10.1145/2342488.2342501

[18] M. Zhang, H. Luo, H. Zhang, A survey of caching mechanisms in information-centric networking, IEEE Communications Surveys Tutorials 17 (3) (2015) 1473–1499. doi:10.1109/COMST.2015.2420097.

[19] A. Araldo, D. Rossi, F. Martignon, Cost-aware caching: Caching more (costly items) for less (isps operational expenditures), IEEE Transactions on Parallel and Distributed Systems 27 (5) (2016) 1316–1330. doi:10.1109/TPDS.2015.2433296.

[20] J. Augé, G. Carofiglio, G. Grassi, L. Muscariello, G. Pau, X. Zeng, Anchor-less producer mobility in icn, in: Proceedings of the 2Nd ACM Conference on Information-Centric Networking, ACM-ICN '15, ACM, New York, NY, USA, 2015, pp. 189–190. doi:10.1145/2810156.2812601.
URL http://doi.acm.org/10.1145/2810156.2812601

[21] J. Aug, G. Carofiglio, G. Grassi, L. Muscariello, G. Pau, X. Zeng, Map-me: Managing anchor-less producer mobility in content-centric networks, IEEE Transactions on Network and Service Management 15 (2) (2018) 596–610. doi:10.1109/TNSM.2018.2796720.

[22] J. Takemasa, Y. Koizumi, T. Hasegawa, Toward an ideal ndn router on a commercial off-the-shelf computer, in: Proceedings of the 4th ACM Conference on Information-Centric Networking, ICN '17, ACM, New York, NY, USA, 2017, pp. 43–53. doi:10.1145/3125719.3125731.
URL http://doi.acm.org/10.1145/3125719.3125731

[23] D. Tuncer, V. Sourlas, M. Charalambides, M. Claeys, J. Famaey, G. Pavlou, F. D. Turck, Scalable cache management for isp-operated content delivery services, IEEE Journal on Selected Areas in Communications 34 (8) (2016) 2063–2076. doi:10.1109/JSAC.2016.2577319.

[24] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, Openflow: Enabling innovation in campus networks, SIGCOMM Comput. Commun. Rev. 38 (2) (2008) 69–74. doi:10.1145/1355734.1355746.
URL http://doi.acm.org/10.1145/1355734.1355746

[25] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford,

C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, D. Walker, P4: Programming protocol-independent packet processors, SIGCOMM Comput. Commun. Rev. 44 (3) (2014) 87–95. doi:10.1145/2656877.2656890. URL http://doi.acm.org/10.1145/2656877.2656890

[26] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for vm-based cloudlets in mobile computing, IEEE Pervasive Computing 8 (4) (2009) 14–23. doi:10.1109/MPRV.2009.82.

[27] M. Satyanarayanan, The emergence of edge computing, Computer 50 (1) (2017) 30–39. doi:10.1109/MC.2017.9.

[28] X. Sun, N. Ansari, Edgeiot: Mobile edge computing for the internet of things, IEEE Communications Magazine 54 (12) (2016) 22–29. doi:10.1109/MCOM.2016.1600492CM.

[29] P. Mach, Z. Becvar, Mobile edge computing: A survey on architecture and computation offloading, IEEE Communications Surveys Tutorials 19 (3) (2017) 1628–1656. doi:10.1109/COMST.2017.2682318.

[30] I. Psaras, O. Ascigil, S. Rene, G. Pavlou, A. Afanasyev, L. Zhang, Mobile data repositories at the edge, in: Proceedings of the 1st USENIX Workshop on Hot Topics in Edge Computing (HotEdge'18), 2018.

[31] E. M. Schooler, D. Zage, J. Sedayao, H. Moustafa, A. Brown, M. Ambrosin, An architectural vision for a data-centric iot: Rethinking things, trust and clouds, in: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017, pp. 1717–1728. doi:10.1109/ICDCS.2017.243.

[32] O. Ascigil, T. K. Phan, A. G. Tasiopoulos, V. Sourlas, I. Psaras, G. Pavlou, On uncoordinated service placement in edge-clouds, in: 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 2017, pp. 41–48. doi:10.1109/CloudCom.2017.46.

[33] A. Madhavapeddy, D. J. Scott, Unikernels: Rise of the virtual library operating system, Queue 11 (11).

[34] A. Madhavapeddy, T. Leonard, M. Skjegstad, T. Gazagnaire, D. Sheets, et al., Jitsu: Just-in-time summoning of unikernels., in: NSDI, 2015.

[35] S. Hendrickson, S. Sturdevant, T. Harter, V. Venkataramani, A. C. Arpaci-Dusseau, R. H. Arpaci-Dusseau, Serverless computation with openlambda, in: Proceedings of the 8th USENIX Conference on Hot Topics in Cloud Computing, HotCloud'16, USENIX Association, Berkeley, CA, USA, 2016, pp. 33–39.
URL `http://dl.acm.org/citation.cfm?id=3027041.3027047`

[36] M. Krol, I. Psaras, Nfaas: Named function as a service, in: ACM ICN'17, ACM, 2017, pp. 1–11.

[37] M. Sifalakis, B. Kohler, C. Scherb, C. Tschudin, An information centric network for computing the distribution of computations, in: ACM ICN'14, 2014, pp. 137–146.

[38] I. Psaras, Decentralised edge-computing and iot through distributed trust, in: Proceedings of the 1st ACM Open IoT Day, MobiSys 2018, 2018.

[39] J. Swearingen, http://nymag.com/selectall/2018/03/when-amazon-web-services-goes-down-so-does-a-lot-of-the-web.html (2018).

[40] M. Krol, I. Psaras, Secure payments for outsourced computations, in: 2018 NDSS Workshop on Decentralised IoT Security and Standards, 2018.

[41] M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, G. Danezis, Chainspace: A Sharded Smart Contracts Platform, in: In Proceedings of the Network and Distributed System Security Symposium (NDSS), 2018.

[42] J. Lind, I. Eyal, P. R. Pietzuch, E. G. Sirer, Teechan: Payment channels using trusted execution environments, CoRR abs/1612.07766. `arXiv:`

1612.07766.
URL http://arxiv.org/abs/1612.07766

[43] M. Al-Bassam, A. Sonnino, M. Krol, I. Psaras, Airtnt: Fair Exchange Payment for Outsourced Secure Enclave Computations (2018).
URL https://arxiv.org/abs/1805.06411v1

[44] D. Grewe, M. Wagner, M. Arumaithurai, I. Psaras, D. Kutscher, Information-centric mobile edge computing for connected vehicle environments: Challenges and research directions, in: Proceedings of the Workshop on Mobile Edge Communications, 2017.

[45] D. Grewe, M. Wagner, H. Frey, ICN-based open, distributed data market place for connected vehicles: Challenges and research directions, in: ICC2017: WS06-Convergent Internet of Things- On the synergy of IoT systems (ICC2017-WS06), 2017.

[46] M. Swan, Blockchain: Blueprint for a new economy, " O'Reilly Media, Inc.", 2015.

[47] J. Lind, I. Eyal, P. Pietzuch, E. G. Sirer, Teechan: Payment Channels Using Trusted Execution Environments (2017).

[48] S. Gueron, A memory encryption engine suitable for general purpose processors., IACR Cryptology ePrint Archive 2016 (2016) 204.