The Impact of the Most Common Assumptions When Modelling SDN-based Mobile Networks

Strahil Panev and Pero Latkoski

Abstract – **In this paper we discuss and evaluate the impact of some of the assumptions commonly used when mathematically modelling the performance of SDN-based mobile networks. Using queuing theory, we model and quantify the impact of the following assumptions: load independent service rate, and limited buffer size. The model is numerically evaluated in MATLAB. The results show that considering a limited buffer significantly impacts the accuracy of the modelling, while at load dependent service rate we notice visible service time deterioration at high loads.**

Keywords – **SDN, Mobile network, Queuing theory, Performance Modelling.**

I. INTRODUCTION

Software Defined Networking (SDN) is a recent paradigm that has occupied the attention of the research community and is already massively used in data centers and commercial mobile networks. It introduces a concept of decoupling the user and control plane, with a centralized network intelligence handled by the controllers, and network infrastructure that resides on commodity hardware which is abstracted from the applications running on top of the network operating system (NOS). SDN is used by the mobile operators to tackle some of the traditional challenges, such as: ossification, reduce operational and capital costs, introduce innovative services, allow for easier network maintenance, and open new opportunities that were unimaginable previously [1].

In cellular mobile networks, the concept of SDN initially was introduced in the Evolved Packet Core (EPC) in existing LTE networks. The mobile core exchange that implemented the concept of decoupling the control and user plane was the Evolved Packet Gateway (EPG), where the two logical functions of Serving Gateway (SGW) and Packet Data Network Gateway (PGW) were split in centralized intelligence residing in the control plane, and distribution of the user plane components on dumb and commercially available switches. In mobile networks, due to the large number of users, a single controller cannot handle the enormous number of flows, which is why the concept of multiple controllers was introduced [2]. This distributed control plane concept allows for a more effective flow management, however it does not come without challenges. In this paper we are interested in both single and multiple controllers' architectures and their application in today's mobile networks.

Strahil Panev is with the Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University, Skopje, R. Macedonia, E-mail: strahil22@gmail.com

Pero Latkoski is with the Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University, Skopje, R. Macedonia, E-mail: pero@feit.ukim.edu.mk

In general, there are three most common approaches used for performance modelling of SDN networks: analytical models, software simulations, and physical testbeds. The most popular approach is the modelling via software simulators/emulators, and least common is by using physical testbeds (due to cost and time consumptions). Mathematical modelling is widely used by the research community due to properties of fast and cost-effective investigation and model validation before moving to real hardware implementations. When mathematically modelling SDN-based networks, several assumptions are almost always considered: load independent service rate, and unlimited buffer [3]. Realistic SDN hardware implementations have mean service rate that deteriorates as the controller load increases, and both the switches and the controller have finite buffer size.

The aim of this paper is to analyze and evaluate the impact that these assumptions have on the accuracy of the performance modelling of SDN-based mobile networks. By using queuing theory, we propose an analytical model to incorporate the effect of limited buffer and the load-dependent mean service rate, and we compare our model with the existing analytical approaches that did not consider these properties. We analyze and quantify the impact on the controller's average service time, and by using MATLAB, we numerically validate and discuss the proposed model. Our conclusions can be used by SDN designers to identify the most critical dependencies and evaluate the impact on the accuracy of the performance modelling. The most important contributions of our work are listed as follows:

1. Investigate the SDN applicability in today's LTE networks and identify the most critical latency factor in the SDN-based mobile core network.
2. Propose analytical modelling based on M/M/m/K and M/M/1 queues to quantify the effect of limited buffer and load-dependent service rate,
3. By using MATLAB, numerically evaluate, and compare the proposed modelling results with the traditional analytical approaches and draw the conclusions that will help in understanding the deviations.

The remainder of this paper is organized as follows: In Section II, we describe the applicability of the SDN concept in mobile core networks. Section III is about analytical modeling of the effect of the discussed assumptions, while in Section IV we numerically evaluate the proposed analytical model. Finally, we state our conclusions in Section V.

II. SDN IN MOBILE NETWORKS

In today's mobile networks, SDN is already widely implemented in backhaul and core networks. The scenario of interest in this paper is presented on Fig. 1, [4]. The concept

used is called SDN-based EPC with partial virtualization, which virtualizes only the control plane 3GPP applications, while the user plane is implemented on physical OF-switches. This makes a lot of sense since the control plane functions have lesser requirements in terms of throughput, but strict requirements on latency and computation. On the other hand, user plane nodes must achieve very high throughput. In a virtual EPC, the Mobility Management Entity (MME), Home Subscriber Server (HSS) and Policy and Charging Rules Function (PCRF) are pure Control Plane (CP) functions. The control software of PGW and SGW is logically centralized, and via the north-bound Application Programming Interface (API) runs on top of an OF SDN Controller (SDNC), as an application. The User Plane (UP) functions of SGW and PGW are installed over SDN switches and controlled by SDNC on the southbound API.

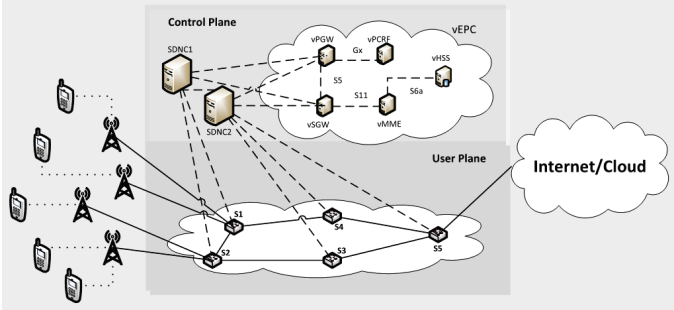

Fig. 1. SDN-based EPC in existing LTE networks

When analyzing the major latency contributors in SDN-based mobile networks, the authors in [5] clearly show that the average controller service time is the most critical factor, while all other have a limited contribution. This is the reason why we are interested in modelling the controller service time, and in the next section we will analyze the impact that the most common assumptions have to this parameter of interest.

III. Main Assumptions when Modelling SDN

In this section we will discuss and mathematically model the most frequent hypothesis used when analytically modelling SDN networks.

A. Unlimited Buffer Size

With aim to obtain closed-form expressions, taking the assumption of unlimited buffer may be one of the most common hypotheses. In a real OF-switch, there are multiple ingress and output ports, and every ingress port has a fix buffer size. The incoming packets from each buffer are then processed against the same flow tables, and there is no control traffic prioritization (each packet is processed as FIFO). It must be stressed that some OF-switches do incorporate traffic priority, but those type of implementations are not subject of our analysis [6]. Using queuing theory, our aim is to mathematically model the controller service time in the case of infinite and finite buffer and analyze the difference.

We assume an SDN network with n switches, $S = \{s_1\ s_2,...,s_n\}$, and m controllers $C=\{c_1,\ c_2,....c_m\}$. We assume that the incoming packets obey Poisson distribution and we model each controller as a single queue, and not per interface. We align with existing work [7], and we model the controller using M/M/m/K (multi-server, finite capacity system) queuing model. The service model between the switches and the controller is shown on Fig. 2.

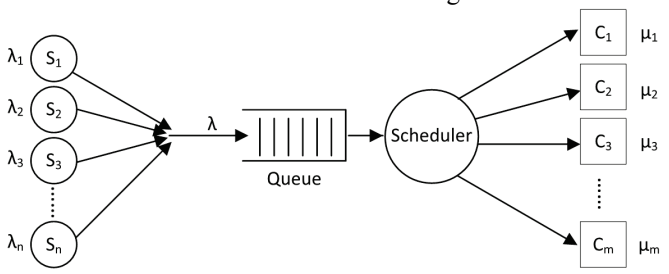

Fig. 2. Service model between controllers and switches

When an incoming packet is not matched against an existing flow table entry in the switch, the packet is sent to the controller via a *Packet-In* message. The controller then installs the necessary new flow table entries in the path of the packet. Each controller has independent service rate μ_k, which follows an exponential distribution, and the incoming packets obey a Poisson arrival distribution. The controller system has a limited queue of K packets, m controllers, and constant arrival rate λ. For an M/M/m/K queue, the values for the arrival and mean service rates are

$$\lambda_n = \begin{cases} \lambda, & 0 \le n < K \\ 0, & n \ge K \end{cases} \quad (1)$$

$$\mu_n = \begin{cases} n\mu & 1 \le n \le m \\ m\mu & m \le n \le K \end{cases} \quad (2)$$

The probability of packets waiting in the queue is given by

$$p_n = \begin{cases} p_o \frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n & 0 \le n \le m \\ p_o \frac{1}{m^{n-m}m!}\left(\frac{\lambda}{\mu}\right)^n & m \le n \le K \end{cases}. \quad (3)$$

where p_o is the probability of no packets in the queue. Using

$$\sum_{i=0}^{K} p_i = 1, \quad (4)$$

we find that p_o is

$$p_o = \left[\sum_{n=0}^{m-1} \frac{1}{n!}(\rho)^n + \sum_{n=m}^{K} \frac{1}{m^{n-m}m!}(\rho)^n\right]^{-1}, \quad (5)$$

where $\rho = \lambda / m\mu \le 1$. The average time a single flow spends in the system, $E(t_c)$ is

$$E(t_c) = \frac{E(N)}{\lambda'} + \frac{1}{\mu} = \frac{E(N)}{\lambda(1-p_k)} + \frac{1}{\mu}, \quad (6)$$

where $E(N)$ is the average number of queued packets in the system, and $\lambda' = \lambda(1-p_K)$ is the effective arrival rate. After using several substitutions in Eq. (6) as given in [8], for $E(N)$ we get

$$E(N) = p_0 \frac{(m\rho)^m \rho}{m!\,(1-\rho)^2}\left(1 - \rho^{K-m+1} - (1-\rho)(K-m+1)\rho^{K-m}\right). \quad (7)$$

Our aim is to compare $E(t_c)$ and p_0 from Eq. (6) and Eq. (5) with the same parameters calculated in the case of M/M/m queue (infinite buffer). The relevant equations for the M/M/m

queue are given as

$$E(t_c) = \frac{1}{\lambda}\left(m\rho + \rho\frac{(m\rho)^m}{(m-1)!}\frac{p_0}{(1-\rho)^2}\right) \quad (8)$$

$$p_o = \left[1 + \sum_{n=1}^{m-1}\frac{(m\rho)^n}{n!} + \frac{(m\rho)^m}{m!}\frac{1}{1-\rho}\right]^{-1} \quad (9)$$

B. Load-Independent Service Rate

In this section we use analytical modelling to consider the fact that a real controller implementation will have a mean service rate degradation as the controller load increases. We use M/M/1 queue with state dependent service [8], which means that we consider a system in which the server works with mean service rate μ_1 until there are l packets in the system, at which point it changes to a different rate $\mu 2$. The values for the arrival and service rates are

$$\lambda_n = \lambda, \quad for\ all\ n, \quad (10)$$

$$\mu_n = \begin{cases} \mu_1, & 1 \le n \le l \\ \mu_2, & n \ge l \end{cases} \quad (11)$$

The probability of packets in the queue is

$$p_n = \begin{cases} p_o\left(\frac{\lambda}{\mu_1}\right)^n & 0 \le n \le l \\ p_o\frac{\lambda^n}{\mu_1^{l-1}\mu_2^{n-l+1}} & n \ge l \end{cases} \quad (12)$$

The probability of no packets in the queue p_0, and the controller service rate $E(t_c)$ are given with the following equations

$$p_o = \left[\frac{1-\rho_1^l}{1-\rho_1} + \frac{\rho_2\rho_1^{l-1}}{1-\rho_2}\right]^{-1}, \quad (13)$$

$$E(t_c) = \frac{E(N)}{\lambda} = \frac{\sum_{n=0}^{\infty} n p_n}{\lambda} =$$

$$\frac{1}{\lambda}p_0\left(\sum_{n=0}^{l-1} n\rho_1^n + \sum_{n=l}^{\infty} n\rho_1^{l-1}\rho_2^{n-l+1}\right) =$$

$$= \frac{p_0}{\lambda}\left(\frac{\rho_1(1+(l-1)\rho_1^l - l\rho_1^{l-1})}{(1-\rho_1)^2} + \frac{\rho_2\rho_1^{l-1}(l-(l-1)\rho_2)}{(1-\rho_2)^2}\right). \quad (14)$$

where $\rho_1 = \lambda/\mu_1$ and $\rho_2 = \lambda/\mu_2$. We want to compare Eq. (13) and Eq. (14) with the equations of a general M/M/1 queue with a single mean service rate of μ. The expressions are as followed

$$E(t_c) = \frac{E(N)}{\lambda} = \frac{1/\mu}{1-\rho} = \frac{1}{\mu-\lambda}, \quad (15)$$

$$p_0 = 1 - \rho. \quad (16)$$

Finally, we want to use the opportunity to explore the controller's average service rate dependency from the number of switches. We can do that easily if we modify Eq. (15) according to [9], we get

$$E(t_c) = \frac{s+1}{2(\mu - \lambda s)}, \quad (17)$$

where s is the number of switches.

C. Poisson Arrival Rate Distribution

In the analytical models proposed to study the OF networks, the most frequent assumption considered is the Poisson arrival rate. In most of the references that we analyzed, this assumption was applied, however we know for a fact that the real network packet traffic exhibits self-similar characteristics. The self-similar traffic is a widely spread traffic phenomenon in modern networks, and its basic characteristic is: segments of the process have the very same statistical properties at different scales. The important definitions and properties of self-similarity are given by Leland et.al. [10], where they define an important parameter to quantify the degree of self-similarity, called the Hurst parameter. For this work, our model assumes Poisson arrival rate, which is better suited for circuit switched networks, while for packet arrivals there is an impact on the accuracy of modelling. In our future work, we plan to better investigate the effect of traffic self-similarity as our theoretical analysis shows that the effect of this phenomenon can be impactful when evaluating the performance of SDN-based mobile networks.

IV. Numerical Results

In this section, we present the performance results of the proposed analytical model. The evaluation is performed using MATLAB. Our focus is on analyzing the controller service time as it is the highest impact contributor to the overall latency in SDN-based mobile networks.

In [3], the authors used a testbed to evaluate the mean service rate of the controller. They measured a mean value of 4166, and in our analysis for μ we try to use values that are in the region of this reference value, already highly exploited in most of the research work in the area. We take the average packet size to be 128 B. They measured a mean value of On Fig. 3 we can see the controller service rate dependency from the average controller load. We substitute Eq. (7) in Eq. (6), and we compare with Eq. (8). We take m=2, and μ=5000. The general analysis shows that if the load is lower, the service rate is also low, however when the load increases, especially at high loads > 0.7, the service time grows exponentially. We can also conclude that in the case of an unlimited buffer, the average service time is lower, while when the buffer has finite value, the average service time deteriorates faster. This is clearly noticeable for load higher than 0.8. We conclude that at high loads, there is a deviation in the analytical model when comparing the case of infinite and finite controller buffer.

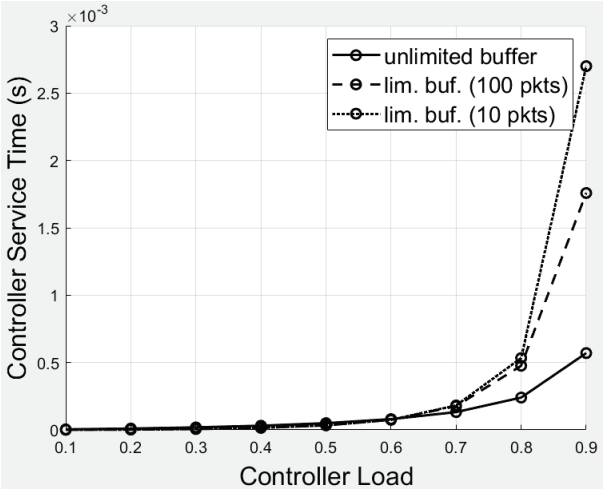

Fig. 3. Controller service time vs. controller load (limited buffer)

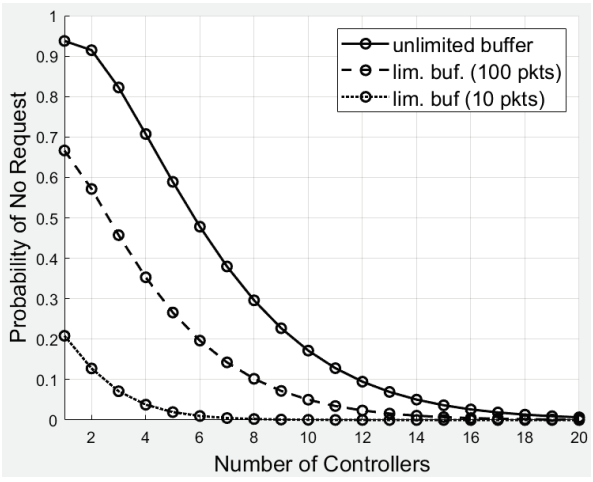

Fig. 4. The relationship between p_0 vs number of controllers (limited buffer)

In Fig. 4, we compare the results from Eq. (5) and Eq. (9). We try to analyze the probability of no packets in the queue in the cases of finite and infinite buffer. If the buffer is unlimited, then for low number of controllers, the probability of no packets in the buffer is very high. As the controller number increases, this probability goes to smaller values. If the buffer is finite, then the probability of no packets at low number of controllers is lower than in the case of unlimited buffer. If the number of controllers is high, this probability goes down. We conclude that as expected, the limited buffer influences the accuracy of the modelling, and it is an important parameter that must be considered.

We continue our analysis for the load dependent service rate. We use Eq. (14) and Eq. (15) to analyze the impact of this assumption. On Fig. 5, we compare the case of a single mean service rate of μ_1=5000, with two cases where μ_1=5000, but after the buffer is filled with l=4 packets, the service rate degrades and has the value of μ_2. We choose the value l=4 packets, as using this value creates a clear visual conclusion of the effect when changing μ_1 to μ_2. For the arrival rates we are using, we will get similar effect also for l=2 to l=5, but increasing l to higher values will actually blur the conclusions that we want to stress. As the arrival rate increases, the controller's service rate also increases, however for the case of a single mean service rate, this increase is very moderate. If μ_2 is lower, then for the case of state dependent service rate, the controller service time deterioration is much more visible. In case we vary l, we can easily conclude that if l is smaller, the controller service time will deteriorate faster and even at smaller arrival rates. As l increases, the degradation is less visible. Finally, we can comment that the variable mean service rate impact is not negligible, and it is an important factor that should be taken into consideration.

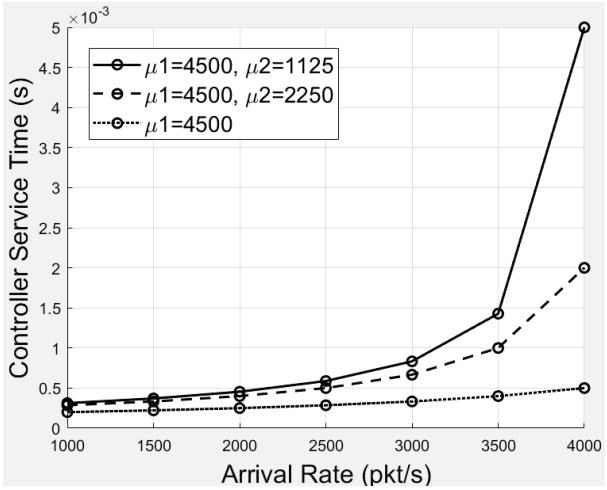

Fig. 5. Controller service time vs. controller load (load-dependent service rate)

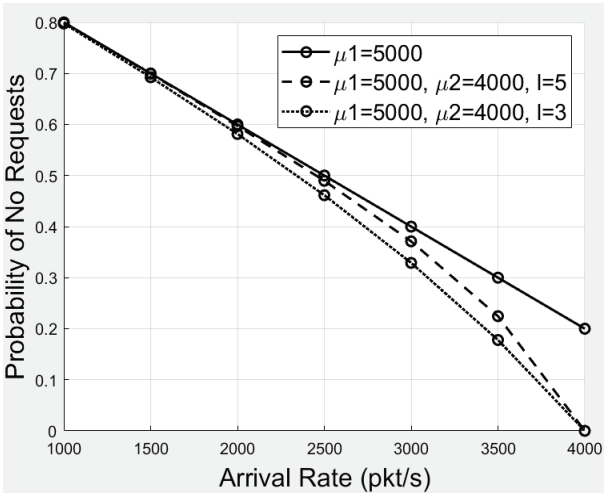

Fig. 6. The relationship between p_0 vs arrival rate (load-dependent service rate)

Fig. 6 shows similar analysis as Fig. 4, however this time we use the arrival rate for the x-axis. We compare Eq. (13) with Eq. (16). As the arrival rate increases, the probability of no requests decreases, as expected. However, in case of a variable mean service rate, this degradation is more visible, especially if l is lower, as shown on Fig. 6. The cases of l=3 and l=5 are plotted and analyzed. Even if we take higher values for l, the results on Fig. 6 will not change much. We conclude that the number of the packets in the system, at which point the service rate changes, is an important parameter that impacts the performance modelling.

Finally, the analysis of controller service time from the number of switches is shown on Fig. 7. We use Eq. (17), and we analyze this dependency for three different values of the controller mean service rate. As the number of switches gets higher, the controller service time increases, and as the mean service rate μ decreases, then this increase in the service time is more visible. This is in-line with our expectations, and we conclude that in this scenario, a distributed control plane should be considered (as opposed to a single controller), as the service degrades rapidly when the network consists of many switches.

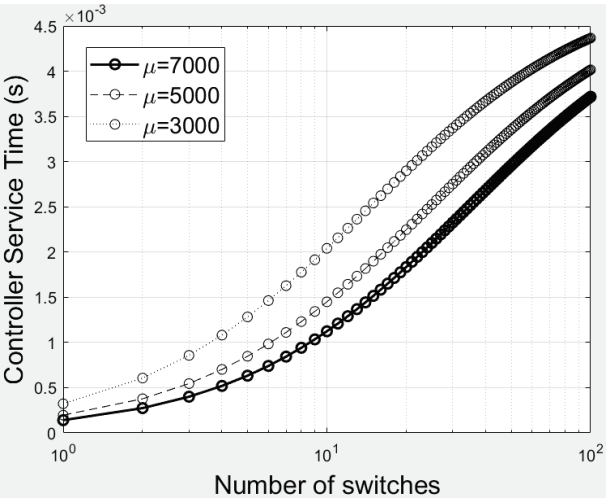

Fig. 7. Controller service time vs number of switches

V. CONCLUSION

In this paper we analyzed the most common assumptions used when modelling SDN-based mobile networks. We proposed an analytical model for state dependent service rate and the limited buffer, and we try to assess the impact when comparing to the standard mathematical models that do not account for these hypotheses. For future work, we plan to expand our analysis by using network calculus theory, and possibly employ HW considerations into our analytical modelling.

REFERENCES

[1] J.H. Cox and J. Chung, "Advancing Software-Defined Networks: A Survey", *IEEE Access*, vol. 5, pp. 25487-25526, 2017.

[2] Y. Oktian, S. Lee, H. Lee, and J. Lam, "Distributed SDN Controller System: A Survey on Design Choice", *Computer Networks*, vol. 121, pp.100-111, 2017.

[3] M. Jarschel et al., "Modeling and Performance Evaluation of an Openflow Architecture", *Proceedings of the 23rd International Teletraffic Congress*, pp. 1-7, 2011.

[4] S. Panev and P. Latkoski, "SDN-based Failure Detection and Recovery Mechanism for 5G Core Networks", *Transaction on Emerging Telecommunication Technologies*, Wiley, 2019.

[5] G. Wang and Y. Zhao, "An Effective Approach to Controller 'Placement in Software Defined Wide Area Networks", *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 344-355, 2018.

[6] D. Singh et. al., "Modelling Software-Defined Networking: Switch Design with Finite Buffer and Priority Queueing", *42nd Conference on Local Computer Networks*, 2017.

[7] K. Sood and S. Yu, "A General QoS Aware Flow-Balancing and Resource Management Scheme in Distributed Software-Defined Networks", *IEEE Access*, vol. 4, pp. 7176-7185, 2016.

[8] W. Stewart, "Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling", Princeton University Press, 2009.

[9] L. Yao et. al., "Evaluating the Controller Capacity in SDN", *23rd International Conference ICCCN*, 2014.

[10] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)", *IEEE/ACM Transactions on Networking*, pp. 1-15, 1994.