Understanding TCP Cubic Performance in the Cloud: a Mean-field Approach

Sonia Belhareth*, Lucile Sassatelli†, Denis Collange*, Dino Lopez-Pacheco†, Guillaume Urvoy-Keller†
*Orange Labs, Sophia Antipolis, France
†Laboratoire I3S - Université Nice Sophia Antipolis - CNRS, 06903 Sophia Antipolis, France
Emails: {sassatelli,lopezpac,urvoy}@i3s.unice.fr, {sonia.belhareth,denis.collange}@orange.com

***Abstract*—Cloud networking typically leads to scenarii where a large number of TCP connections share a common bottleneck link. In this paper, we focus on the case of TCP Cubic, which is the default TCP version in the Linux kernel. TCP Cubic is designed to better utilize high bandwidth-delay product path in an IP network. To do so, Cubic modifies the linear window growth function of legacy TCP standards, e.g., New Reno, to be a cubic function. Our objective in this work is to assess the performance of TCP Cubic in a cloud setting with a large number of long-lived TCP flows. We rely on a mean-field approach leading to a fluid model to analyze the performance of Cubic. After a careful validation of the model through comparisons with ns-2, we evaluate the efficiency and fairness of Cubic as compared to that of New Reno for a set of typical cloud networking scenarii.**

I. Introduction

To meet the changing requirements of Internet networks, various strategies for congestion control in TCP have been designed, such as Bic-TCP [1], TCP Cubic [2], Compound TCP [3] and TCP Reno [4]. Bic-TCP and TCP Cubic are designed specifically for high bandwidth delay products links. TCP Cubic is the most widely used version of TCP nowadays [5]. It is characterized by a cubic window growth function and implemented by default in the Linux kernel since version 2.6.19.

The kind of networks for which TCP Cubic (and other high-speed TCP flavors) has been been designed call for analytical models to allow to predict performance in general cases and study the impact of various parameters. Indeed, using experimental testbeds does not usually provide enough flexibility to explore a wide range of parameter values, while the computational and storage cost of simulation (with, e.g., ns-2) increases quickly when considering high-performance networking environments such as those met in cloud networking.

In this paper, we aim at developing an analytical model for TCP Cubic to analyze its performance in typical cloud scenarii where a large number of long-lived TCP connections, e.g., HTTP streaming or back-up flows, share a bottleneck link. Specifically, we consider three scenarii: (i) an intra data-center (DC) scenario with a lot of ongoing traffic between physical servers (intra-DC scenario), (ii) an inter data-center scenario where high provisioned links are used to synchronize or back up data (inter DC scenario), (iii) a content distribution scenario where a large number of high speed clients, e.g., FTTH clients, simultaneously download content from the data center (FTTH scenario). Our contributions are twofold:

- Based on a mean-field approximation, we derive a fluid model for TCP Cubic, that allows to predict performance in terms of several metrics. We carefully validate this analytical model against ns-2 simulations for our cloud scenarii and exhibit its domain of validity.
- We provide an extensive comparison of TCP Cubic and New Reno for our cloud scenarii, assessing their efficiency/fairness trade-off as well as the impact of the buffer size on their performance.

II. Related Work

Several analytical models have been proposed in the literature to analyze the performance of legacy TCP versions, but there are fewer for TCP Cubic. The authors of [6], [7], [8] consider a single long-lived flow. In [9], Poojary and Sharma investigate the cases of three TCP Cubic connections as well as the competition between a Cubic and New Reno connection. In [8], Blanc et al. compare the performance of Cubic, Compound TCP, HighSpeed TCP and New Reno under a simple loss model, using Markov chains. A few studies [10], [11] have also investigated TCP Cubic in vivo using experimental testbeds and are discussed in Sec. VI-B.

Mean field approximations, or equivalently mean-field limits, date back to the seventies [12] and are used to analyze the limiting behavior of systems made of N objects, as N tends to infinity. As the limit process is the solution of a deterministic ordinary differential equation (ODE), it is referred to as a fluid limit, or fluid model. Baccelli et al. introduced a mean-field model for a set of N TCP Reno connections in [13]. In this paper, the authors consider a bottleneck router implementing the RED (Random Early Discard) active queue management policy, the TCP Reno version of TCP at equilibrium, i.e., TCP operates in congestion avoidance mode and does not experience time-outs. The model is derived, but the focus is then put on the fixed points of the mean field equations. We build on this work to obtain a mean field model of TCP Cubic. The model for TCP Cubic is an extension of [13] in that it is more complex: two parameters instead of one now define the state of an object, and the time of the last loss is an additional quantity that must be approximated. Furthermore, we validate extensively our model on cloud networking scenarii, trying to assess its domain of validity by identifying simulation behavior that it cannot capture.

III. System description

In this work, we assume the network is in steady state, i.e., the TCP Cubic sender has reached equilibrium. Thus, we neglect, in line with the approach in [13], the slow start phase of TCP.

A. Window variation in TCP Cubic

In the congestion avoidance phase, TCP Cubic features two modes of operations, the so-called TCP and Cubic modes [2]. The TCP mode corresponds to low bandwidth delay products (BDPs), while the Cubic mode is triggered for high BDPs. Each mode corresponds to a specific way of increasing the window size and is determined by the following pair of equations:

$$w_c(t) = C(t - V_{cubic})^3 + w_{max} \quad (1)$$

$$w_{tcp}(t) = w_{max}(1-\beta) + \frac{3\beta}{(2-\beta)}\frac{t}{R(t)} \quad (2)$$

where w_{max} is the congestion window prior to the last loss event[1], $R(t)$ is the estimated RTT of the connection, β and C are constant values usually set to 0.2 and 0.4, respectively, and $V_{cubic} = \sqrt[3]{\frac{\beta w_{max}}{C}}$. The state of a TCP Cubic connection at time t is defined by the pair $< w(t), w_{max}(t) >$ where $w(t)$ is set to $\max(w_c(t), w_{tcp}(t))$ upon each ACK reception. If the maximum is $w_c(t)$ (resp. $w_{tcp}(t)$), TCP Cubic is said to operate in Cubic mode (resp. TCP mode). The variables $w(t)$ and $w_{max}(t)$ can take any positive real values, though in the implementation they are upper-bounded by the maximum possible window size defined in the stack. Let E denote the space spanned by $< w, w_{max} >$ in the remainder of the article. When a loss occurs (the sender detects it thanks to a DUPACK), the state of the connection is changed from $< w(t-\delta t), w_{max}(t-\delta t) >$ to $< w(t), w_{max}(t) > = < (1-\beta)w(t-\delta t), w(t-\delta t) >$.

B. TCP Cubic mode of operation

From equations 1 and 2, it appears that for fixed network set-up (the physical path between the sender and the receiver) and stationary load conditions, TCP Cubic operates (in stationary regime) in either Cubic or TCP modes but not both. We present in this part the relationship between the mode of operation of TCP Cubic and the network parameters. We thereby find out that among the three scenarii we focus on (described in the next part), namely high speed clients to DC, intra-DC and inter-DC, Cubic operates in the TCP mode for the first two scenarii, while it is in Cubic mode only for the last one.

Let us consider the difference $D(t, RTT, w_{max}) = w_c(t) - w_{tcp}(t)$. Let $t_0(RTT, w_{max}) > 0$ denote the first value for which the derivative of $D(t, RTT, w_{max})$ with respect to t is zero:

$$t_0(RTT, w_{max}) = \left(\frac{\beta}{C(2-\beta)RTT}\right)^{\frac{1}{2}} + V_{cubic} \; .$$

It can be shown (details are omitted by lack of space) that $D(t_0, RTT, w_{max})$ is the minimum value of $D(.)$ in t. It increases in RTT and in w_{max}. In the steady state, the value of w_{max} is $BDP + BS$, where BS denotes the buffer size available for a TCP connection, and the RTT is lower-bounded by the end-to-end path latency denoted by $baseRTT$. Hence, for given network settings of $baseRTT$, BS and BDP, the sign of $F(baseRTT, BS, BDP) = D(t_0(baseRTT, BS + BDP), baseRTT, BS + BDP)$ allows to determine the mode of operation of TCP Cubic.

C. Network scenarii

We consider a classical dumbbell topology with N TCP senders, N TCP receivers and a shared bottleneck with fixed capacity $N \times L$ and fixed buffer size $N \times B$. The latency of the path between each pair of sender and receiver is fixed and equal to $baseRTT$. L and B can thus be seen as the allocated server and buffer capacities per flow. We assume a FIFO/droptail server/queue management policy for the queue of the bottleneck, as it is the prevalent policy in todays network, including data-centers. The three scenarii we focus on correspond to the following choices of L and B:

- Scenario A - FTTH-client: this scenario models the case of high-speed clients, with FTTH access, that are simultaneously downloading from a DC. We thus consider $L = 100Mb/s$ and $baseRTT = 20ms$, and take the buffer size $BS = 50$ packets. The value of $baseRTT$ corresponds to typical RTTs observed for FTTH clients [14], especially when they access well-provisioned servers. This is in contrast with DSL access where the latency on the last mile typical represent around 50 ms of the total RTT.
- Scenario B - Intra-DC scenario: we consider $B = 1Gb/s$ and $baseRTT = 1ms$, as servers in a typical DC are equipped with 1 Gb/s NICs and the end-to-end delay observed in DC are in the order of a ms [15]. We also take $BS = 50$ packets.
- Scenario C - Inter-DC scenario: we consider a dedicated link connecting to DC that are far apart. Hence, we take $B = 1Gb/s$ (remember that it corresponds to the average bandwidth per flow), $baseRTT = 50ms$ and $BS = 500$ packets.

IV. A FLUID MODEL FOR TCP CUBIC

We build on [13] and consider N TCP Cubic connections routed through a bottleneck link whose aggregate capacity is NL packets per second. The queue size at the sending buffer of the bottleneck link router is denoted by $Q^{(N)}(t) = Nq^{(N)}(t)$, the buffer size being NB. Let $S_n^{(N)}(t) =< w^{(n)}(t), w_{max}^{(n)}(t) >$ be the state of connection n, for $n = 1, \ldots, N$. In order to express all quantities governing the connection states in terms of an absolute time variable, t is changed to $t - s_{loss}^{(n)}(t)$ in equations 1 and 2, where $s_{loss}^{(n)}(t)$ denotes the elapsed time since the last loss seen by the n-th TCP sender. Our goal is to predict the performance of the system of N TCP Cubic connections thanks to a fluid model, stemming from a mean-field approximation. For the sake of space, we only give a sketch of the formal proof which allows us to consider the limit behavior of the system when N tends to infinity, so as to get fluid model of the performance. Considering $\mathbf{Y}^{(N)}(t) = (S_1^{(N)}(t), \ldots, S_N^{(N)}(t))$ as the state of the system, that is that composed of the N connections, $\mathbf{Y}^{(N)}(t)$ is an homogeneous Markov chain that can be shown to be a mean-field interaction model with N objects, as defined in [16]. We define the occupancy measure as the fraction of connections in each state at each time t, and denote it

[1]Note that w_{max} is varying over time but is constant between two loss events. This is also the case for V_{cubic}.

by $p^{(N)}(t, w, w_{max})$ for time t and state $< w, w_{max} >$. Theorem 3.1 of [12] ensures that, as $N \to \infty$, for any $t > 0$ and $< w, w_{max} > \in E$, $p^{(N)}(t, w, w_{max})$ converges uniformly almost surely to the solution $p(t, w, w_{max})$ of the coupled Ordinary Differential Equations (ODE) below, with initial condition $p(0, (1-\beta)x, x) = 1$ (x is set to 5 in the experiments of Section V). Additionally, the other quantities of interest, namely $q^{(N)}(t)$, $R^{(N)}(t)$ and $s^{(n)}_{loss}(t)$ can be expressed thanks to their deterministic fluid limits $q(t)$, $r(t)$ and $s_{loss}(t)$, respectively. It is worth noting that the convergence to the fluid limit holds in the presence of $q^{(N)}(t)$ that can be considered as a resource, as defined and proven in [16].

$$\begin{aligned}
&\frac{dp(t, w, w_{max})}{dt} = \\
&\qquad \Bigg\{ \frac{w}{(1-\beta)} \frac{1}{r(t)} \delta\left(w_{max}, \frac{w}{(1-\beta)}\right) \sum_{v=1}^{W} p\left(t, \frac{w}{(1-\beta)}, v\right) \\
&\qquad - \frac{w}{r(t)} p\left(t, w, w_{max}\right) \Bigg\} k(t - r(t)) + \Bigg\{ - Ap(t, w, w_{max}) \\
&\qquad + A \frac{(w-1)}{r(t)} p\left(t, (w-1), w_{max}\right) \Bigg\} (1 - k(t - r(t))) \qquad (3)
\end{aligned}$$

All the equations describing the system involve per-connection quantities. In the above equation, W denotes the maximum possible value of w and w_{max}. The probability that a packet be dropped by the bottleneck buffer at time t is denoted by $k(t)$, whose expression is the same as in [13] for droptail, as that of $q(t)$. The parameter A denotes the increase of the congestion window $w(t)$ between t and $t + dt$. Depending on the mode of operation (either Cubic or TCP), A is hence the time derivative of $w_c(t)$ or $w_{tcp}(t)$ given in equations 1 and 2: $A = 3C(t - s_{loss}(t) - V_{cubic})^2$ or $A = \frac{3\beta}{2-\beta} \frac{1}{r(t)}$. The parameter $s_{loss}(t)$ denotes the average absolute time of the last loss before time t. It is estimated thanks to the intensity $i(t)$ of the loss process, assumed to be Poisson as in [13], and we take:

$$s_{loss}(t) = \begin{cases} 0 & \text{, if } i(t) < 1 \\ t - \frac{t}{i(t)} & \text{, otherwise} \end{cases} \qquad (4)$$

V. Numerical results

In this section, we present validation results based on the three scenarii presented in Section III-C. Our approach for validation is to compare the fluid model results against ns-2 simulations. The former are obtained using a numerical ODE solver of matlab. The TCP packet size is set to 1480 bytes. We set the number of connections in ns-2 to $N = 10$. We consider the time-series of average window size and instantaneous queue size, and the marginal distribution of the window size.

Both for FTTH and intra-DC scenarii (only the latter is represented), TCP Cubic operates in the TCP mode. Figures 1 and 2 show a very good temporal match both in terms of the variation of amplitudes and in the frequency of oscillations of the two metrics for intra DC scenario.

In the Inter-DC scenario, owing to the large bandwidth delay product of the path and the high RTT, TCP Cubic operates in the Cubic mode. The matching between the model and the simulation is less good in this scenario, as it can be observed from Figures 3 and 4. Our model does not capture the loss synchronization effect among the sources that occurs in the simulation. Indeed, the shape of the average window time

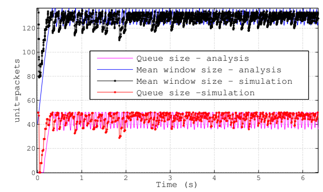

Fig. 1. Time series of queue size and average window size - Intra-DC scenario - Cubic

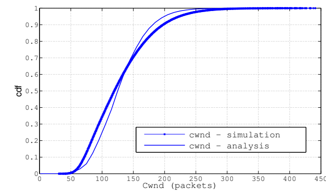

Fig. 2. Congestion window - Intra-DC scenario - Cubic

series in Figure 3 indicates that almost all connections experience loss simultaneously and repeatedly. We have checked in the ns-2 simulations that all the 10 connections have indeed synchronized loss events, in spite of the various techniques we tried to avoid synchronization of sources (increase of the buffer size and the level of multiplexing, i.e., number of active connections). It seems to be a fundamental feature of TCP Cubic to exhibit this loss event synchronization as already observed by Hassayoun and Ros in [17]. In this paper, the authors studied several high speed version of TCP and observed, through simulation, the existence of synchronization among sources even when using several counter-measures like RED policy, traffic on the backward path or time-varying RTT. They also observed that while a lot of sources experience losses simultaneously, the utilization of the link remain close to the maximum. This is confirmed by our simulations and somewhat captured by our model. A deeper analysis will be performed in a future work to identify whether the infinite number of connections the model considers is the root cause for not capturing the synchronization effect.

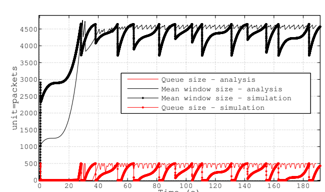

Fig. 3. Time series of queue size and average window size - Inter-DC scenario - Cubic

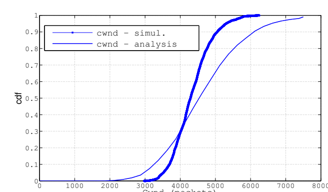

Fig. 4. Congestion window - Inter-DC scenario - Cubic

VI. Study of Fairness and the impact of the buffer size

In this section, we present applications of our fluid models to the study of two key problems. The first one is the fairness of TCP Cubic as compared to that of TCP New Reno. While TCP Cubic is able to take advantage of paths with larger bandwidth delay products, one can question its ability to share the bandwidth evenly between flows. We use New Reno as a reference here, as this version of TCP is known to achieve a good level of fairness when the flows share the same path.

The second issue that we investigate is the impact of the buffer size on the efficiency of Cubic (and also New Reno). The question of buffer sizing has received a lot of attention, e.g., [18], [17], [19]. Rules derived from [17], [19] recommend using buffer sizes whose range is between 10% and 60% of the bandwidth delay product of the path. However, some measurements studies focusing specifically on TCP Cubic, observe a detrimental effect of small buffer [11]. However, the authors in [11] pinpointed that the jury was still out concerning the root cause of the inefficiency that they observed as it could be an intrinsic feature of TCP Cubic or an artefact of their testbed.

Throughout this section, we restrict ourselves to the intra-DC and FTTH scenarii, as we obtained good match with simulations for those cases. While TCP Cubic operates in TCP mode in these scenarii, note that the algorithms that govern TCP Cubic in TCP mode and TCP New Reno are not the same.

A. Fairness analysis

For the case of TCP Cubic and New Reno, we assess the fairness of the protocol by two metrics. To capture the time variation of the distribution of congestion windows, we compute, at each time instant the coefficient of variation[2] (CoV) of the window size distribution and we report the cdf of CoV over a large time period. It is clear from Figure 5 that TCP Cubic achieves a better level of fairness than TCP New Reno over the two scenarii of interest, as (i) the CoVs for Cubic are both smaller and span also over a smaller set of values and (ii) the marginal cdfs (not represented here) span over a smaller set of values for Cubic.

Fairness and efficiency have to be assessed jointly. We report in Figures 6 the distribution of utilization for the FTTH and intra-DC scenarii for both TCP Cubic and New Reno. We can now conclude that the better fairness of Cubic is not achieved at the expense of a lower link utilization.

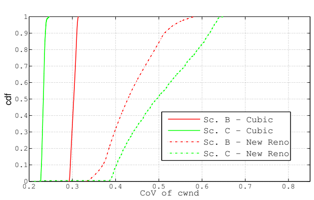

Fig. 5. CoV of congestion window - Intra-DC and FTTH - Cubic and New Reno

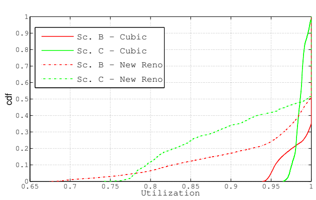

Fig. 6. Utilization - Intra-DC and FTTH - Cubic and New Reno

B. Impact of the buffer size

In this section, we investigate the impact of the buffer size on the utilization of the queue (and consequently of the server). We report results only for the intra-DC scenario owing to space constraints. We vary the buffer size at the bottleneck from 10% of the BDP to 100% of the BPD for both Cubic and New Reno - see Figures 7 and 8, where we present the normalized occupancy of the queue. Several conclusions can be drawn from these figures. First, both Cubic and New Reno are greedy in the sense that the larger the buffer size, the larger the queue occupancy. It is not necessarily a good news as larger queue occupancy means larger set-up latency for new incoming flows and larger jitter for time sensitive traffic, e.g. Web searches in a DC [20]. Second, Cubic is more greedy than New Reno. Third, TCP New Reno is clearly less efficient than Cubic for buffer sizes smaller than 60% of the BDP as we observe a significant fraction of mass at zero, meaning that the buffer is often empty, hence the server is likely to be underutilized. Overall, for the case of Cubic, our model suggests that this version of TCP is able to survive with buffer sizes as small as 20% of the BDP. The experimental results obtained in [11] are thus not pathological behaviors of Cubic, but are likely to be due to another cause, e.g., a bad implementation (the author in [11] used an early implementation of Cubic in the Linux kernel). Note however that when the buffer size becomes very low, other technical problems might appear in real network appliances (such as competition between reading and writing into buffers). Hence, while the behavior observed in [11] does not seem to be due to Cubic itself, it is likely to be observed with other real experimental networks.

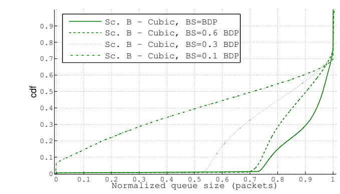

Fig. 7. Impact of buffer size - Intra-DC - Cubic

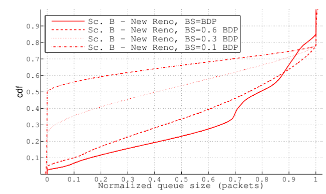

Fig. 8. Impact of buffer size - Intra-DC - New Reno

VII. Conclusion

In this paper, we have derived a fluid model for TCP Cubic, that allowed to predict the values of various metrics such as distribution of the window sizes of N connections, throughput, RTT, loss rate and queue size. The model accuracy is very good for the intra-DC and FTTH scenarii, while it is less good for the inter-DC scenario where TCP Cubic operates in Cubic mode and causes loss synchronization amongst the connections, as observed in [17]. Future work includes to identify whether such behavior of TCP arises from a too low level of multiplexing, that cannot be captured by the model that is a limit when the number of connections tends to infinity. Finally, we show that TCP Cubic is at once more efficient and fair than TCP New Reno, in particular in the case of low buffer sizes. In contrast to TCP New Reno, TCP Cubic is able to survive with buffer sizes as small as 20% of the BDP, thereby shedding some light on the possible cause of bad utilization observed in experimental works for such buffer sizes. Our future work aims at having a more general model, encompassing the slow-start phase, and mixing TCP New Reno and Cubic connections, to assess wider results about these TCP versions.

[2]The CoV is the ratio of the standard deviation to the mean of a distribution. It can be seen as a normalized measure of its variability.

References

[1] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control (bic) for fast long-distance networks," in *IEEE INFOCOM*, vol. 4, 2004, pp. 2514–2524.

[2] S. Ha, I. Rhee, and L. Xu, "Cubic: a new TCP-friendly high-speed TCP variant," *SIGOPS Oper. Syst. Rev.*, vol. 42, no. 5, pp. 64–74, Jul. 2008.

[3] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A compound tcp approach for high-speed and long distance networks," in *IEEE INFOCOM*, 2006, pp. 1–12.

[4] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," RFC 2001 (Proposed Standard), Internet Engineering Task Force, Jan. 1997, obsoleted by RFC 2581. [Online]. Available: http://www.ietf.org/rfc/rfc2001.txt

[5] P. Yang, W. Luo, L. Xu, J. Deogun, and Y. Lu, "TCP congestion avoidance algorithm identification," in *International Conference on Distributed Computing Systems (ICDCS)*, Washington, DC, USA, 2011, pp. 310–321.

[6] H. Cai, D. Y. Eun, S. Ha, I. Rhee, and L. Xu, "Stochastic ordering for internet congestion control and its applications." in *IEEE INFOCOM*, 2007, pp. 910–918.

[7] W. Bao, V. W. S. Wong, and V. C. M. Leung, "A model for steady state throughput of TCP CUBIC," in *IEEE Global Telecommunications Conference (GLOBECOM)*. IEEE, 2010, pp. 1–6.

[8] A. Blanc, K. Avrachenkov, and D. Collange, "Comparing some high speed TCP versions under Bernoulli losses," in *International Workshop on Protocols for Future, Large-Scale and Diverse Network Transports (PFLDNet)*, Tokyo, Japan, May 2009.

[9] S. Poojary and V. Sharma, "Analytical model for congestion control and throughput with tcp cubic connections," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2011, pp. 1–6.

[10] S. Jain and G. Raina, "An experimental evaluation of cubic TCP in a small buffer regime," in *National Conference on Communications (NCC)*, 2011, pp. 1–5.

[11] D. Leith, R. Shorten, and G. McCullagh, "Experimental evaluation of cubic-TCP," *PFLDNet*, 2007.

[12] T. G. Kurtz, "Solutions of Ordinary Differential Equations as Limits of Pure Jump Markov Processes," *Journal of Applied Probability*, vol. 7, no. 1, pp. 49–58, 1970.

[13] F. Baccelli, D. R. McDonald, and J. Reynier, "A mean-field model for multiple TCP connections through a buffer implementing red," *Perform. Eval.*, vol. 49, no. 1-4, pp. 77–97, Sep. 2002.

[14] A. Hafsaoui, G. Urvoy-Keller, D. Collange, M. Siekkinen, and T. En-Najjary, "Understanding the impact of the access technology: the case of web search services," in *International conference on Traffic Monitoring and Analysis (TMA)*, 2011, pp. 37–50.

[15] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Internet Measurement Conference*, 2010, pp. 267–280.

[16] M. Benaïm and J.-Y. Le Boudec, "A class of mean field interaction models for computer and communication systems," *Performance Evaluation*, vol. 65, no. 11-12, pp. 823–838, 2008.

[17] S. Hassayoun and D. Ros, "Loss synchronization and router buffer sizing with high-speed versions of TCP," in *INFOCOM Workshops 2008, IEEE*, 2008, pp. 1–6.

[18] D. Wischik and N. McKeown, "Part I: buffer sizes for core routers," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 3, pp. 75–78, Jul. 2005.

[19] S. Chen and B. Bensaou, "Can high-speed networks survive with droptail queues management?" *Elsevier Comput. Netw.*, vol. 51, no. 7, pp. 1763–1776, May 2007.

[20] B. Vamanan, J. Hasan, and T. Vijaykumar, "Deadline-aware datacenter TCP (D2TCP)," in *ACM SIGCOMM*, 2012, pp. 115–126.