

Flash Crowd Effect in RCP

Filipe Abrantes, João Taveira Araújo and Manuel Ricardo

Abstract—The Rate Control Protocol (RCP) [1] is an explicit congestion control mechanism that, amongst other characteristics, reduces the average flow completion time (AFCT) metric by one order of magnitude when compared to TCP NewReno. RCP reduces the AFCT by allowing new flows to instantly use the same rate as existing flows in the network. This results in link utilization temporarily exceeding available capacity when flows enter a network, inducing queue build-up. As such, RCP is particularly vulnerable to *flash crowds*, whereby a system witnesses a significant increase in the number of flows over a short period of time. In this paper we analyze RCP’s response to varying rates of increase in the number of flows. We conclude that, for a given arrival growth rate, RCP is able to stabilize queue length as long as this rate does not exceed well defined limits. We quantify the queue length required to stabilize the system response and the limit arrival growth rate using a model of RCP that incorporates the effect of new flow arrivals. Finally, we validate our analysis through ns-2 simulations.

Index Terms—RCP, congestion control, flash crowds

I. INTRODUCTION

In recent years congestion control algorithms which rely on routers to adjust the rate of connections have become increasingly researched. Amongst these explicit congestion control algorithms is the Rate Control Protocol (RCP) [1]. An RCP router calculates a rate that is to be used by all flows bottlenecked in that router. This rate is updated according to the link utilization and the current queue length of the router. By specifying a single rate across all flows, the RCP system exhibits (1) perfect flow bandwidth fairness at all times and (2) removes the adaptation phase that new flows typically go through in other congestion control protocols. The result is that the average flow completion time (AFCT) is greatly reduced, i.e. by one order of magnitude compared to New Reno, which is argued by RCP authors as being the right metric for congestion control [2] since most flows in the Internet have a short duration. The reduced AFCT comes at the cost of utilization overshoot whenever the number of flows in the system increases - and under-utilization when the inverse happens. This reflects the design philosophy behind RCP [3], which assumes it works well for most cases, whilst not performing as successfully under extreme, rare conditions. In this paper we dig exactly into one of these cases where RCP might struggle to perform. The objective of the paper is not to detract RCP, far from it, but rather to present an analysis and results that help us predict RCP behaviour under specific, extreme conditions and, based on our findings, aid in the design of a robust RCP system. We study the significant and persistent increase in the number of flows in a network, also known as *flash crowds*, and its implications on RCP. Flash crowds appear typically at the beginning of a popular event, e.g. sports match, live concert streaming, *Slashdot* article,

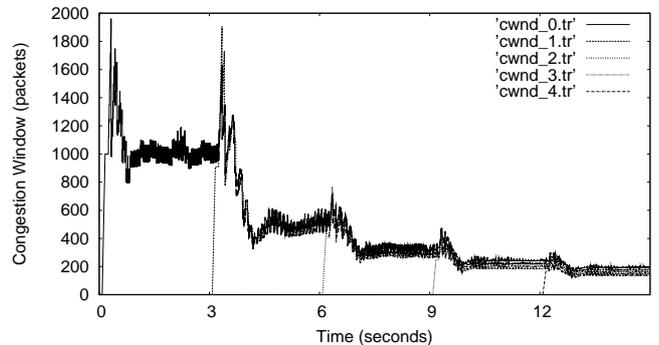


Fig. 1. Flow congestion windows. A new flow enters the network every 3 seconds. Whenever a flow enters the network there is a period of increased queuing delay due to capacity overshoot. The spikes in the congestion window and the slow convergence reflect this increased queuing delay.

where the number of participants, read flows, is likely to increase significantly over a short period of time.

We introduce the variation of the number of flows in the equations that characterize RCP’s behaviour, and by doing that we are able to define system steady-state properties. We find that there is a maximum growth rate of the number of flows for which the bottleneck queue will stabilize. We find this stabilizing queue length and also the maximum growth rate for which the system is stable. These results are useful to the design of an RCP system as parameters may be chosen in order to make RCP more robust in the presence of flash crowds.

The paper structure is as follows: in this section we motivate and explain the scope of our study. In Section II we model the variation of the number of flows and present our analysis, and in Section III we validate the results from the analysis through ns-2 simulations. We conclude the paper in Section IV.

II. FLASH CROWD EFFECT IN RCP

The RCP router periodically calculates a common rate R to be used by all flows. This rate is passed to the sources using the rate field of a header placed between the network and transport headers. The RCP router only fills the rate field if its common rate R is lower than the value contained in the rate field. As a result, on arrival the rate field will have been filled by the path bottleneck. The rate R is updated by the router at every control interval d so a fraction of the unused bandwidth is distributed amongst flows and a portion of the standing queue is drained. R is defined as:

$$R(t) = R(t-d) + \left(\frac{\alpha \cdot (C - y(t-d)) - \beta \cdot \frac{q(t-d)}{d}}{\hat{N}(t)} \right) \quad (1)$$

where $y(t)$ is the sum of the incoming bandwidth, d is the delay of the communication between the router and the sources (an approximate average at least), and C is the link capacity. α , and β are system constants tuned to ensure system stability for any capacity, delay and number of sources. \hat{N} represents an estimate of the number of flows traversing the router, which is calculated by the router as $\hat{N}(t) = \frac{y(t-d)}{R(t-d)}$. The accuracy of the estimate of the number of flows is of extreme relevance, as it directly impacts network utilization. Moreover, if during an update interval the number of flows entering the network differs from the number of flows leaving the network, link utilization will probably also differ from 1. To better understand this phenomenon let's think on what happens when flows enter the network. The new flows will be informed to use a certain rate R that was calculated taking into account less flows. As a result, the entrance of the new flow will cause temporary over-utilization of the network, leading to queue build-up, until the estimate of the number of flows in the router becomes more accurate, and R is set accordingly. Fig. 1 refers to an experiment where a new flow enters the network every 3 seconds. After the arrival of a new flow there is an increase in the congestion window of all flows, caused by the increase in queuing delay due to over-utilization, and then a few RTTs are required until RCP drains the bottleneck queue and adjusts the common rate R with the correct number of flows. Furthermore, we can see that the effect of the arrival of new flows is proportional to the ratio between new and existing flows.

A. Modeling Flow Arrivals

An RCP system can be studied using a fluid model. Following Eq. 1, and a) assuming a constant number of flows in the network, b) considering all flows have the same RTT and c) ignoring queue boundaries, the set of equations below characterizes an RCP system:

$$F(t) = \alpha \cdot (C - y(t-d)) - \beta \cdot \frac{q(t-d)}{d} \quad (2)$$

$$\dot{y}(t) = \frac{F(t)}{d} \quad (3)$$

$$\dot{q}(t) = y(t) - C \quad (4)$$

where the system delay d can be expressed by the sum of the propagation RTT d_0 and queuing delay:

$$d = d_0 + \frac{q(t)}{C} \quad (5)$$

To introduce the effect of the variation of the number of flows, we need to write Eq. 3 as:

$$\dot{y}(t) = \frac{1+L(t)}{d} \cdot F(t) + \frac{L(t)}{d} \cdot y(t-d) \quad (6)$$

where $L(t)$ represents the growth rate of the number of flows:

$$L(t) = \frac{N(t) - N(t-d)}{N(t-d)} \quad (7)$$

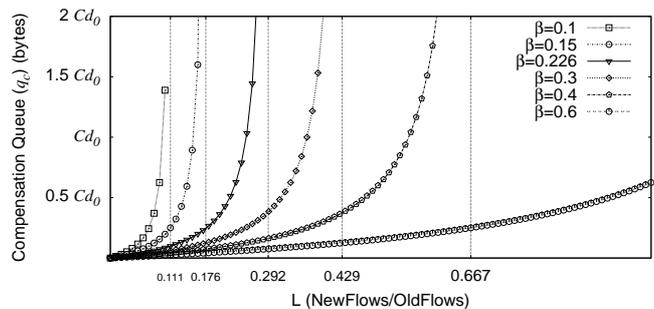


Fig. 2. The compensation queue q_c required to neutralize a growth rate L of the number of flows for various values of β .

Note that we define the growth rate of the number of flows as being normalized to the system delay d . As such it represents the ratio between the number of new flows during an interval of d seconds and the number of active flows in the previous interval.

B. Finding the Limits of RCP

To understand the limits of RCP, we analyze its behaviour in the presence of a constant growth rate of the number of flows. This means that we consider $L(t) = L$ to be constant and establish steady-state properties and limits as a function of L . For example, considering $L = 0.5$ results in an increase in the number of flows by 50% over each interval of d seconds. As we will show soon enough, L itself influences the system delay d and for that reason we will also define stationary properties of the system as a function of L_0 . L_0 is a particular case of L calculated for the network minimum delay d_0 which is constant, allowing us to define flow growth more objectively.

We start the analysis by assuming steady-state conditions of the system represented by Eq. 6, 2, 4, 5. Steady-state conditions are $y(t) = C$, $\dot{y}(t) = 0$, $L(t) = L$. Under these conditions, we can rewrite Eq. 6 as:

$$q_c = \frac{C \cdot L}{\beta \cdot (1+L)} \cdot d \quad (8)$$

which, using $d = d_0 + \frac{q}{C}$ from Eq.4, results in:

$$q_c = \frac{C \cdot L}{(\beta - 1) \cdot L + \beta} \cdot d_0 \quad (9)$$

where d_0 is the network RTT excluding queuing delay at the router. This is an interesting result, assuming that the system is able to achieve steady-state. In the presence of a constant growth rate in the number of flows in the network, the queue length of the bottleneck router will grow to a point where it neutralizes the effect of the arrival of new flows. We call this queue length the *compensation queue* or q_c . The compensation queue q_c required to balance flow growth rate is proportional to the network bandwidth delay product $C \cdot d_0$, and grows with the flow growth rate L , while decreasing with an increase of β . An interesting remark is that the parameter α does not influence the compensation queue. This is somewhat expected as α controls the weight given to the spare bandwidth in the feedback given to the sources. In steady-state the link is fully

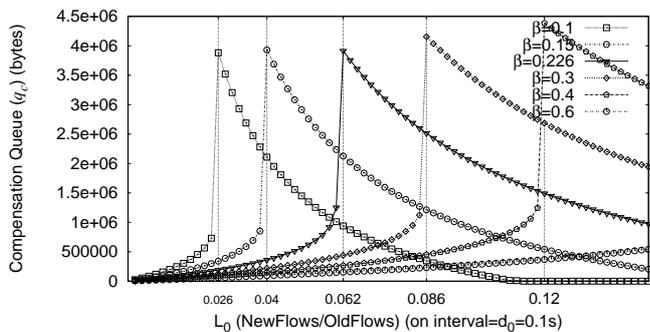


Fig. 3. The compensation queue q_c required to neutralize a growth rate L_0 of the number of flows for various values of β . This refers to the particular case of $C = 100$ Mbit/s and $d_0 = 0.1$ s.

utilized, thus there is no spare bandwidth. Another interesting conclusion is that the RCP system can only sustain the flash crowd if $(\beta - 1) \cdot L + \beta > 0$. If this condition is not met, q_c will tend to infinity, meaning that utilization will be persistently above the network capacity and the system will be unstable. Fig. 2 shows q_c as a function of L , β . The stability limits are shown in the figure as vertical lines. We have seen that the RCP system tries to neutralize the growth of the number of flows by building up the queue, stabilizing queue length up to a certain growth limit. These results also show a more subtle connection. We have established a relationship between the compensation queue q_c and the growth rate L . The growth rate L , however, is defined as the growth rate of the number of flows each d seconds, while d itself depends of L . This does not allow us to define a constant growth rate. To overcome this problem we define L_0 , which has the same meaning as L , but refers to the growth on a fixed interval of d_0 seconds. Additionally, we can represent L as a function of L_0 :

$$1 + L = (1 + L_0)^{\frac{d}{d_0}} \quad (10)$$

upon simplification:

$$L = (1 + L_0)^{1 + \frac{q_c}{C \cdot d_0}} - 1 \quad (11)$$

d_0 , as previously stated, is the network RTT excluding queuing delay. Using this new definition of L in Eq. 9 we obtain:

$$q_c = \frac{C \cdot \left[(1 + L_0)^{1 + \frac{q_c}{C \cdot d_0}} - 1 \right]}{(\beta - 1) \cdot \left[(1 + L_0)^{1 + \frac{q_c}{C \cdot d_0}} - 1 \right] + \beta} \cdot d_0 \quad (12)$$

and now we have q_c defined only in terms of initial conditions, allowing us to determine q_c for a given constant growth rate of the number of flows. Unfortunately, this equation is not easily reducible to a closed form so we will just leave it as is, solving it numerically. The resulting plot is shown in Fig. 3, which exhibits a similar pattern to that of Fig. 2. One difference is the marking of stability limits. In Fig. 3 the maximum y vertex of each curve corresponds to the highest growth rate L_0 for which RCP is able to absorb the flash crowd.

In conclusion our analysis shows that, within certain limits, RCP is able to stabilize queue length even in the presence of a

constant L or, in other words, if an exponential growth of the number of flows occurs. We have shown how to calculate the length at which the queue stabilizes given a growth rate L_0 , a network minimum RTT of d_0 , a link capacity C , and the β parameter of RCP. Likewise, we have shown how to calculate the maximum growth rate L_0 which RCP is able to sustain whilst remaining stable.

C. Model and Analysis Limitations

The model we used to study RCP response to the variation of the number of flows has two major simplifications. It considers that all flows have the same RTT and does not consider queue boundaries. We don't expect significant impact from these simplifications, apart from the obvious - i.e. if the queue buffer is smaller than the compensation queue q_c required, the RCP system will not be able to neutralize the flash crowd, resulting in an ever increasing packet drop rate. A deeper study on the implications such simplifications have should be the subject of future work.

The main limitation of our analysis of the RCP model is that we assumed convergence to the steady-state. This may not happen however, as the RCP system is only stable for certain pairs of values of α, β . Therefore, the results of our analysis are only valid for pairs of α, β that enable system stability. A previous study [4] has defined the area of α, β for which RCP is stable.

D. Response to Typical Arrival Distributions

We have derived steady-state properties and conditions as a function of a constant growth rate of the number of flows L_0 . As such we can calculate the compensation queue q_c if the number of flows in the network grows by a factor of $(1 + L_0)$ in each interval of d_0 seconds - an exponential increase. It is equally interesting to understand how an RCP system responds to other types of growth of the number of flows, namely in the presence of typical flow arrival distributions. To this end, we need to find how $L_0(t)$ behaves for these distributions. We analyze $L_0(t)$ for 3 types of flow arrival distributions: Laplace, Normal, and Erlang. The Laplace and the Normal distributions refer to the case of scheduled events, e.g. sports match, where arrivals may start before the event. The Erlang distribution refers to the case of unplanned events, e.g. *Slashdot* article, where there is a strong ramp-up reaction shortly after the event occurs, which then fades away in time. The probability density function (PDF) of the Laplace distribution is defined as:

$$f(x) = \frac{1}{2b} \cdot e^{-\frac{|x-\mu|}{b}} \quad (13)$$

the PDF of the Normal distribution is defined as:

$$f(x) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}} \quad (14)$$

the PDF of the Erlang distribution is defined as:

$$f(x) = \frac{\lambda^k \cdot x^{k-1} \cdot e^{-\lambda \cdot x}}{(k-1)!} \quad (15)$$

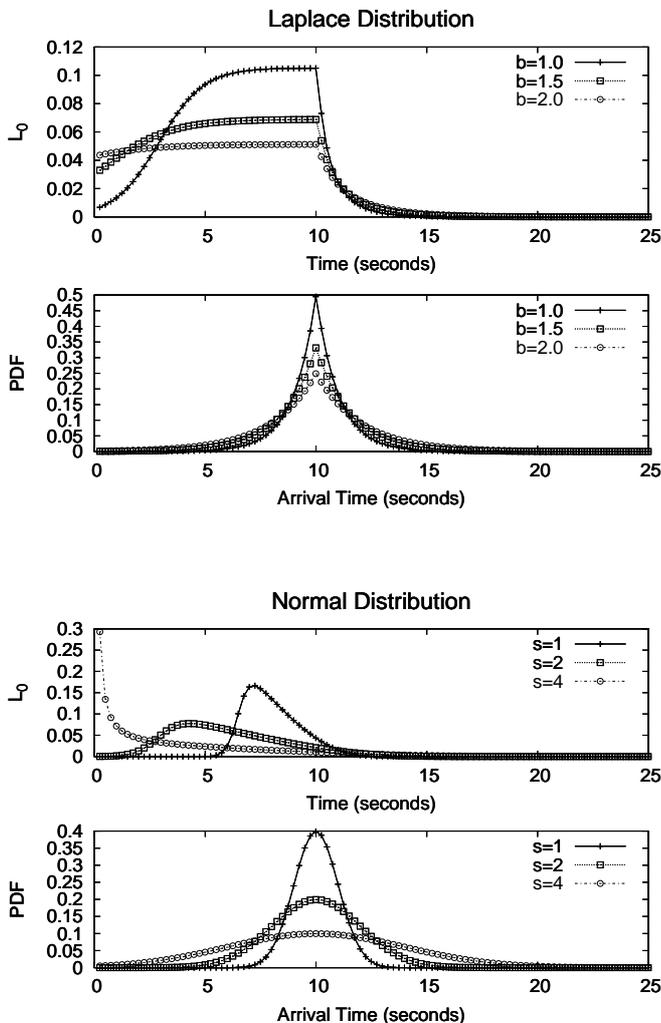


Fig. 4. The relationship between L_0 and the PDF of the Laplace, and the Normal distributions.

where x represents the arrival time. In Fig. 4, 5 we plot the evolution of $L_0(t)$ over time for some cases of the 3 distributions. Those plots are obtained for flash crowds of 5000 flows, and considering $d_0 = 0.1$ s. Also, the initial number of flows in the system, i.e. before the flash crowd, is set to 1. The results obtained can be generalized for a flash crowd with any number of flows, as long as the ratio between the number of flows of the flash crowd and the initial number of flows in the network is kept constant. Analyzing $L_0(t)$ for a PDF of an arrival distribution allows us to infer the queue response to that PDF. Queue length will follow $L_0(t)$ dynamics if $L_0(t)$ is below the stability limit (shown in Fig. 3), however if $L_0(t)$ is above the stability limit, then the queue length will increase exponentially. We will see this in more detail in the next section.

III. SIMULATION RESULTS

The purpose of our simulation results is twofold: we wish to both 1) validate the theoretical limits extracted from the model presented in the previous section as well as 2) understand the limitations such a model has in fully representing an actual RCP system. To this end, we present results performed with ns-2 using our own implementation of the RCP algorithm

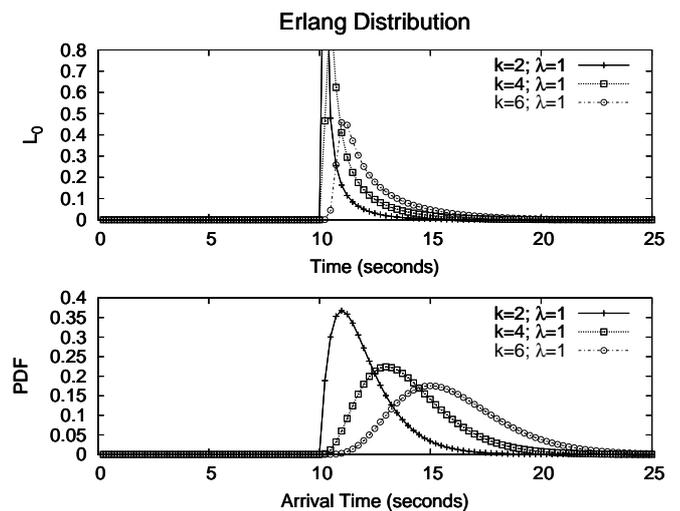


Fig. 5. The relationship between L_0 and the PDF of the Erlang distribution.

based on the existing XCP source code included in the ns-2 package. The setup, shown in 6, is composed of wired nodes connected to a sink S via a router, R . To ensure the same bottleneck is shared across all flows, nodes connected to R have twice the bandwidth available between R and S , which was set at 100Mbit/s. The propagation delay, d , was set to 25ms unless otherwise stated, resulting in a total round trip time of 100ms for each flow.

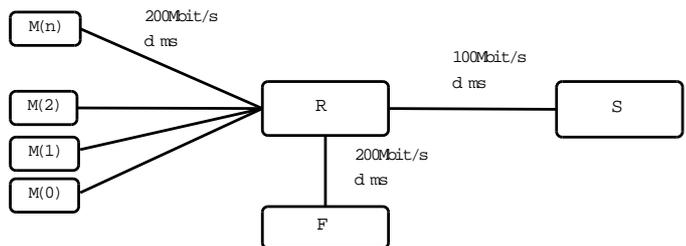


Fig. 6. Simulation setup

Since our main emphasis is on understanding queue dynamics under a sustained increase of flows, we first populate the system with flows from nodes M_i to the sink S . This allows the system to both stabilise the flow rate attributed to every flow and drain the queue, which naturally builds up as the first flows enter the network. The number of initial nodes m is the minimum value of flows to which the growth rate L can be successfully applied, obtained using $m = \text{round}(\frac{1}{L} - 1)$. Once the queue has been depleted, the node responsible for simulating the flash crowd effect, F , initiates flows at the desired rate.

A. Model Validation

We begin by validating the relationship between the compensation queue q_c and L_0 present in Eq. 3. We run a set of experiments for several values of L_0, β . In these experiments $\alpha = 0.4$ Fig. 7 plots the values obtained through ns-2 simulation overlapped with the theoretical values. The values

obtained through simulation are represented by lines with points, while the theoretical results are shown in simple lines. The simulation results support that our analysis is valid and accurate. The bottom plot of Fig. 7 represents the queue length dynamics over time for various L_0 and $\beta = 0.226$, where we observe that the queue length converges to a vicinity of the value we have derived in the analysis. The plot above, shows the compensation queue required for a given pair of L_0, β . The curves obtained through simulation mirror those obtained through the analysis with only a small error.

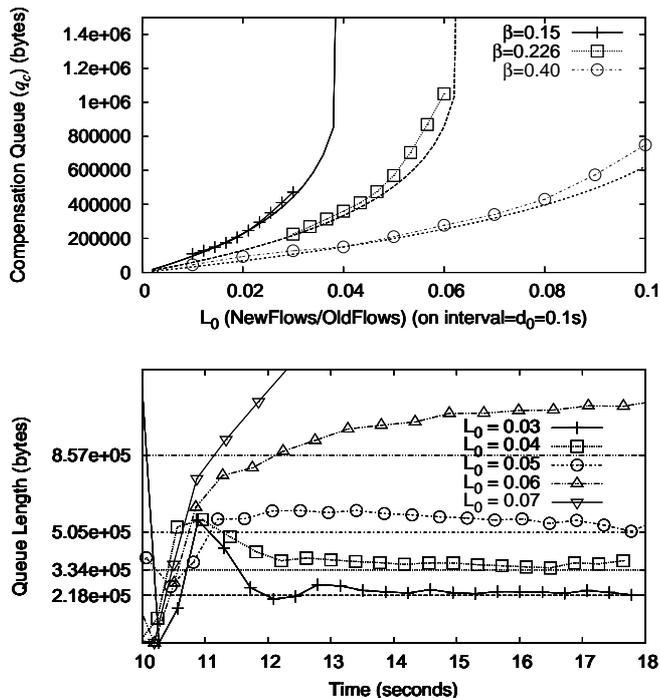


Fig. 7. **Above)** The compensation queue q_c as a function of L_0 for various β . The theoretical values are plotted with lines without points. **Below)** Queue length vs. time for $\beta = 0.226$ and a range of L_0 . The compensation queue theoretical values are represented by horizontal lines.

B. Laplace Distribution

We analyze how RCP adapts to a surge in the number of flows following a Laplace distribution (Eq. 13). We center the peak of the Laplace distribution at instant $t = 20$ s and experiment with various values of the scale parameter b of the Laplace distribution. b controls the degree of concentration of flow arrivals around $t = 20$ s, being that for $b = 1$ arrivals are more concentrated than for $b = 2$. 5000 flows are injected in the network for each simulation run and RCP was configured with $\beta = 0.226, \alpha = 0.4$. The resulting figures (Fig. 8) show how $L_0(t)$ evolves over time, and also the consequent evolution of the queue length. It is observable that Laplace distributions tend to produce $L_0(t)$ function that tends to converge to a constant value. The queue length follows the dynamics of $L_0(t)$ if $L_0(t)$ stays below the stability limit, which for this case is $L_0 \approx 0.062$ (marked in the figure as an horizontal line). If $L_0(t)$ goes above the stability limit, then the queue will grow exponentially, as it happens in this

experiment for $b = 1, b = 1.5$. For $b = 1.5$, $L_0(t)$ tends to 0.068 which is only slightly above the stability limit. For this reason the exponential growth of the queue in this case is a bit timid.

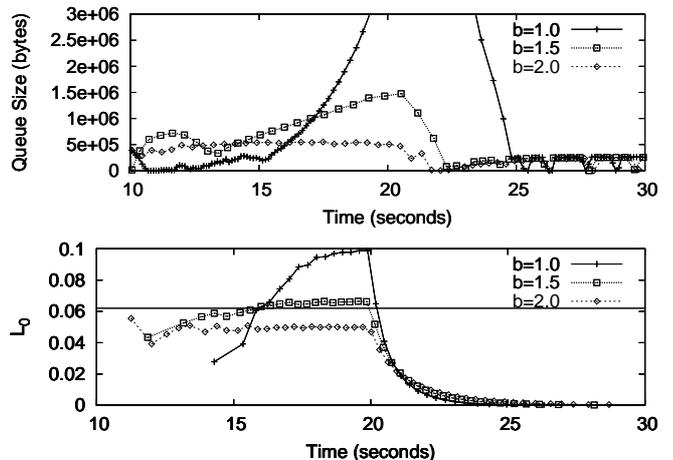


Fig. 8. **Above)** The queue response to flash crowds following Laplace distributions with $b = 1, b = 1.5, b = 2$. **Below)** The growth rate $L_0(t)$ of the number of flows throughout time.

C. Normal & Erlang Distributions

Finally, we analyze how RCP adapts to a surge in the number of flows following Normal and Erlang distributions (Eq. 14, and Eq. 15, respectively). We center the peak of the Normal distribution at $t = 10$ s, while the peak of the Erlang distribution varies between 5 and 10 s. We also vary the parameters of the distributions that regulate the concentration of flow arrivals. The number of injected flows was, again, 5000. The results, shown in Fig. 9 (Normal), and Fig. 10 (Erlang), indicate that both distributions produce periods of higher acceleration of the number of flows than the Laplace distribution. The value of $L_0(t)$ required to inject 5000 flows easily surpasses the stable limit of $L_0 < 0.06$ for the tested scenarios. The Erlang distribution is the worst in this aspect as it may even cause large spikes of $L_0(t)$ right in the beginning of the flash crowd. The periods of high acceleration experienced in these distributions, even if only for a short time, prove to be much harder to control by RCP, than the continued acceleration experienced by flash crowds following Laplace distributions. Whenever $L_0(t)$ goes above its stable limit, the queue length grows exponentially causing system delay to increase accordingly. For the system to recover from this unstable period, $L_0(t)$ must dive well below the initially established stability limit, because that limit was valid assuming a much lower base delay. How low $L_0(t)$ must go after an unstable period, depends on how much the system delay has grown during the period of instability. This fact is most clearly observable in the results from the experiment with a Normal distribution of flow arrivals (Fig. 9), where the longer the unstable period is, the lower $L_0(t)$ must go before queue length starts to decrease.

Another aspect of these experiments that stands out, is the high value of the queue length (in the order of tens of

Mbyte) obtained when the stability limit of L_0 is breached. In our experiments we did not limit the size of queue length, mainly because our objective was to validate the mathematical model. However, we do not expect real systems to have such large buffers. If the stability limit is breached, massive packet loss is expectable and additional measures are required to guarantee decent network performance. Such measures might include adopting some sort of admission control mechanism, or increasing the value of the parameter β of the RCP controller, whenever $L_0(t)$ crosses to the unstable region.

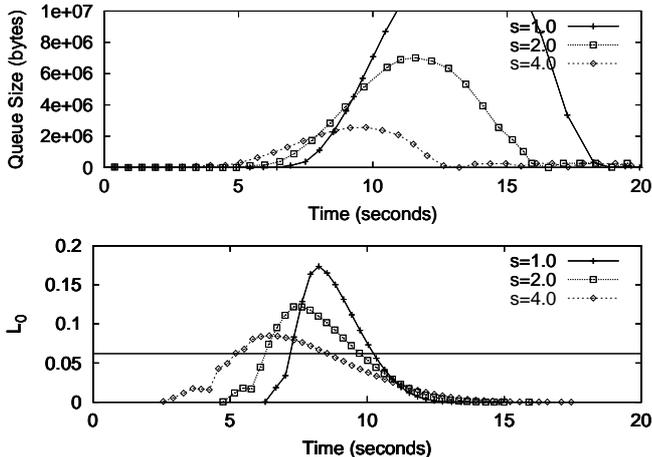


Fig. 9. **Above)** The queue response to flash crowds following Normal distributions with $s = 1, s = 2, s = 4$. **Below)** The growth rate $L_0(t)$ of the number of flows throughout time.

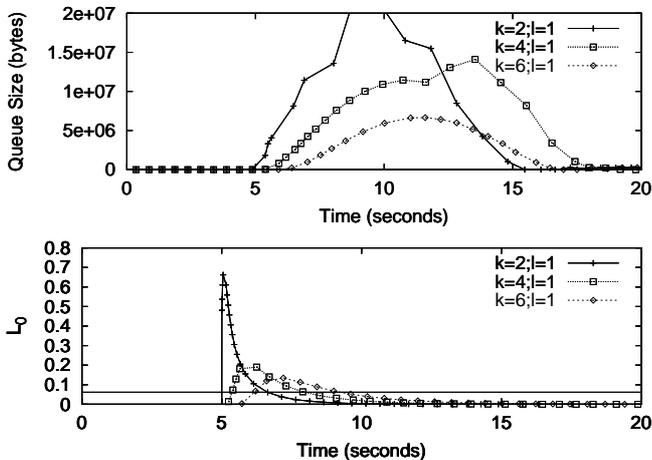


Fig. 10. **Above)** The queue response to flash crowds following Erlang distributions with $k = 2, k = 4, k = 6$, and $\lambda = 1$. **Below)** The growth rate $L_0(t)$ of the number of flows throughout time.

IV. CONCLUSION

In this paper we have studied the effect that the persistent and significant increase of the number of flows has in an RCP system. We have introduced the variation of the number of flows in the differential equations that characterize the behaviour of RCP, and by that we were able to determine properties of the steady state of RCP. We found that RCP is

able to stabilize queue length if the growth rate $L_0(t)$ of the number of flows does not exceed a certain limit. The queue length required to stabilize the system is proportional to the BDP of the network and decreases with β . The maximum growth rate for which RCP is stable is obtained by identifying the maximum of Eq. 3. With the results we have presented, the designer of an RCP system is better prepared to choose RCP parameters and also to predict the system response in the presence of a flash crowd. Additionally, we have studied how the growth rate $L_0(t)$ behaves for the case of three typical arrival distributions: Laplace, Normal, and Erlang. Flash crowds following a Laplace distribution have shown to be the easier to control by RCP, while those following an Erlang distribution where more prone to drive the system to instability - assuming the flash crowds are composed by the same number of flows and have similar duration.

REFERENCES

- [1] N. Dukkupati, M. Kobayashi, R. Zhang-Shen, and N. McKeown. Processor Sharing Flows in the Internet. In *IEEE IWQoS*, June 2005.
- [2] N. Dukkupati and N. McKeown. Why Flow-Completion Time is the Right Metric for Congestion Control. *ACM Computer Communication Review*, 36, 2006.
- [3] Y. G. Nandita Dukkupati and R. Zhang-Shen. Typical versus Worst Case Design in Networking. In *Proc. of HotNets-IV*, 2005.
- [4] H. Balakrishnan, N. Dukkupati, N. McKeown, and C. Tomlin. Stability Analysis of Explicit Congestion Control Protocols. Technical report, Stanford University Department of Aeronautics and Astronautics Report: SUDAAR 776, September 2005.