This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2011 proceedings

A TCP prediction scheme for enhancing performance in OBS networks

Kostas Ramantas and Kyriakos Vlachos

Computer Engineering and Informatics Department & Research Academic Computer Technology Institute
University of Patras, Rio, Greece
{ramantas, kvlachos}@ceid.upatras.gr

Abstract— The efficient transmission of TCP traffic over OBS networks is a challenging problem, due to the high sensitivity of TCP congestion control mechanism to losses. In this paper, a traffic prediction scheme is proposed that exploits TCP traffic dynamics to optimize the performance of TCP transmission over OBS networks. Due to the TCP flow control mechanism, traffic dynamics can be accurately predicted in at least one RTT-long prediction window. In the proposed scheme, the prediction process is tightly coupled with the burst assembly process since the edge node is capable of inspecting incoming TCP traffic, keeping traffic statistics in parallel to the assembly process. These statistics are then used for traffic predictions. In this way burst size can be predicted and thus in advance reserve the appropriate resources. In this paper, we detail the traffic prediction mechanism and we also provide simulation results to assess its performance.

Keywords-Optical Burst Switching; TCP; Traffic Prediction

I. INTRODUCTION

TCP transmission over OBS [1] has been extensively studied in the related literature. The bufferless nature of OBS networks and the best-effort transmission of bursts has numerous side effects on TCP performance, which has led many researchers to cope with the performance evaluation and enhancement of TCP over OBS. TCP performance suffers from the high number of segments that are lost upon a single burst drop, which falsely triggers TCP congestion avoidance. This usually results in many sources timing out and entering the slow start phase, which may also result in synchronizing TCP transmissions with an imminent effect on link utilization. Further, the introduction of an unpredictable aggregation delay challenges the window mechanism used by the TCP protocol, with TCP flows characterized as medium or slow suffering the highest performance degradation [2].

One class of solutions proposed in the literature relies on the OBS domain and assumes no modifications to the TCP protocol. These solutions include burst retransmission techniques that "mask" the effect of bursts losses to the TCP sender [3], and adaptive burst assembly algorithms that adjust the assembly period dynamically per edge node [4]. A second class of solutions focuses on improving TCP protocol so that it detects false congestion events and does not trigger congestion avoidance mechanism on every packet loss. These techniques are based either on feedback from burst losses communicated to the TCP sender as in BTCP [2], or analyzing RTT statistics in a short time frame, as in SAIMD [5], to infer congestion. However, the efficient transmission of TCP over OBS without significant enhancements in the TCP protocol is still considered an open problem that must be solved before the commercial deployment of OBS.

In this work, we rely on burst size predictions for achieving performance enhancements for TCP traffic over OBS. We argue that if it was possible to accurately predict TCP flows' throughput, it would also be possible to predict burst sizes for the future assembly cycles. That would allow making reservations of the appropriate resources in advance, enhancing network performance and improving bandwidth utilization. Several research efforts have focused on predicting aggregate network traffic. Among them, the LMS adaptive filter has been previously used for obtaining burst size predictions in OBS networks [7] [8] due to its high efficiency and good performance. However, using linear predictive filters for traffic forecasting also bears a significant limitation. Aggregate traffic predictions lack information at the flow level, as traffic measurements only reflect network conditions at the time the traffic measurements were taken. Thus, changes in the network state (like burst loss ratio) are enough to invalidate them, taking time for the filter to converge.

In this work, we propose a novel approach for burst size predictions, based on intimate flow-level knowledge of TCP traffic, to obtain traffic predictions in a short control time interval. Flow-level traffic statistics are extracted by a TCP flow profiler that is integrated with the burst-assembly process, via passive traffic measurements. These are then used for the online estimation of traffic parameters for predicting future burst sizes. The rest of the paper is organized as follows. Section II discusses TCP traffic profiling issues, while Section III presents the burst prediction mechanism along with its detailed analytic modeling. Finally, in Section IV, the predictor's accuracy and its effect in achieving enhanced performance for TCP flows over OBS is investigated using ns-2 network simulator.

II. TCP TRAFFIC PROFILING

Network traffic characterization is a vital step for traffic prediction and dynamic bandwidth allocation. In this work, network load is represented with a flow-level model, whose parameters are computed according to the underlying traffic structure. We propose the estimation of traffic parameters to

978-1-61284-233-2/11/$26.00 ©2011 IEEE

be carried out by a TCP flow profiler, which extracts flow-level statistics from active TCP connections. In this section, the traffic model and the main architectural issues of the TCP profiler for OBS networks are presented.

A. Traffic model

In this work, a flow-level model is employed for representing network traffic, as opposed to more common time series models. Network traffic is modeled as the superposition of heavy-tailed TCP connections, whose sizes are drawn from a Pareto distribution. This is a commonly accepted model that explains the self-similarity of internet traffic [9]. To "calibrate" the model, a TCP flow profiler [10] is employed so that to compute a small set of parameters, regarding the connection size and the rate process of active TCP flows. With respect to TCP flow rate, it is known to have a high degree of predictability, especially for long-lived flows, that can be analytically predicted. Formula based prediction of TCP flow rate only requires up-to-date measurements of burst loss ratio *p*, flow round trip time RTT and number of segments per transmitted burst [6]. These parameters are measured and kept constantly updated by the flow profiler. With respect to flow connection size, as mentioned above, it is drawn from a Pareto random variable X, whose complementary CDF, defined as $P(x) = Pr(X > x)$ is:

$$P(x) = \left(\frac{x}{x_{min}}\right)^{-\alpha} \tag{1}$$

It can be seen that *P(x)* is parameterized with x_{min} parameter and tail index parameter α. A Maximum Likelihood Estimator (MLE) is used for estimating the tail index parameter α proposed in [11], for given connection size samples $x_i \geq x_{min}$:

$$\hat{\alpha} = n\left[\sum_{i=1}^{n} ln\left(\frac{x_i}{x_{min}}\right)\right]^{-1} \tag{2}$$

It must be noted here that connection size samples are output by the flow profiler, whose architecture is detailed in the following section. With respect to x_{min} parameter (i.e. the minimum connection size), we assume that it is known or estimated from direct data observation. In what follows, we refer to *P(x)* as the flow *survival function*, as it captures the probability of a flow "surviving", i.e. staying active, to transmit at least x bytes. The survival function is used in the calculation of the *expected residual life* of flows, defined as:

$$e(x) = \int_{x}^{\infty} \frac{P(u)}{P(x)}\, du \tag{3}$$

B. TCP flow profiler architecture

In this section the main features of the TCP profiler architecture for OBS networks are presented**.** The proposed profiler is integrated with the burst assembly unit and thus it is capable of extracting TCP flow-level characteristics, as it has access to all incoming TCP segments. Segment headers are inspected before being assigned to the appropriate assembly queue. Thus, keeping traffic statistics at the edge nodes of an OBS network does not bear a significant overhead, since no computationally intensive packet processing is needed. However, due to memory size and bandwidth limitations, it is infeasible for the profiler to store traffic information for all active flows on a packet by packet basis. Thus, the well-known traffic sampling technique is used, which is common in traffic monitoring architectures like Cisco Netflow.

Flow-level measurements are performed on a subset of active flows, called the "*flow sample*". Sampled flows are selected according to Bernoulli sampling scheme, as soon as the flow SYN packet is received from the profiler, during the TCP three-way handshake phase. The flow is selected with a probability p_s*=1/N,* also known as the *sampling rate*. The flow profiler, using passive traffic monitoring, calculates a set of per-flow counters and empirical distributions for the following parameters on a packet by packet basis:

- *Flow length (i.e. number of segments transmitted so far)*
- *Flow Round-Trip-Time (RTT)*
- *Flow Segment per burst ratio (SPB)*

Additionally, the profiler keeps track of the burst loss ratio per source-destination pair, by monitoring signaling messages received by the edge router. It must be noted that SYN-based sampling technique employed by the profiler, is known to guarantee an independent selection of the flow sample [12]. Thus, sample-based traffic statistics can serve as unbiased estimates of unsampled ones.

C. Active flow histogram

All monitored TCP flows are organized in a flow histogram, based on their current flow length, which is updated online on a packet by packet basis. Inactive flows, i.e. the ones exceeding a time threshold of inactivity, are removed from the histogram. The objective of the active flow histogram is to count the number of overall active flows along with the service they have received so far from the network, which is represented by the number of packets they have successfully transmitted. On this basis, it is possible to estimate the overall number of active flows and their expected residual life, which constitute the two fundamental metrics for characterizing any work-conserving queuing system and predicting its evolution.

To construct the histogram, the flow sample is partitioned in a number of bins, with each bin corresponding to a range of flow lengths. For example the k-th bin, denoted as bin-k, corresponds to the flows with length in the interval (r_k, r_{k+1}], and the number of sampled flows that fall into bin-k is denoted as NF_k. Long-lived flows exceeding a certain threshold (to be discussed in the following section) are pooled together in the rightmost edge bin of the histogram, that extends to infinity. The number of unsampled flows is simply calculated by multiplying the number of sampled flows in each bin with the reciprocal of the sampling rate, *N* which is called the *sampling period*. $N \cdot NF_k$ provides an unbiased estimate of the number of unsampled active flows with a length of (r_k, r_{k+1}]. This is due to the independent selection of sampled flows, which is guaranteed by the SYN-based sampling technique detailed in the previous section. Regarding the variance of the scaling estimate, it is bounded by the number of active flows on each bin, as well as the profiler sampling rate. The standard error of the estimator for *bin-k* is [12]:

$$\frac{\sqrt{Var(\widehat{NF}_k)}}{NF_k} = \sqrt{\frac{N}{NF_k}} \qquad (4)$$

From the above formula, it is clear that under-populated bins (i.e. with a small NF_k as compared to the sampling rate) lead to a high variance. Several methods have been proposed in the literature to analytically determine optimal bin widths [13] but make a lot of a priori assumptions about the flow length distribution. In this work, the widths of the bins were empirically selected so that the standard error of the estimation is bounded, and stays under 5%.

III. Online TCP Traffic Prediction

One very promising approach for achieving performance enhancements for TCP traffic over OBS is the advance reservation of network resources. This can be achieved with measurement-based prediction of burst sizes that allows burst-level reservations at the core before the burst size is actually known. This approach, given a reasonably accurate prediction method, can combine the performance and QoS guarantees of two-way signaling with the low latency of one-way protocols as in [7]. In this work, we take advantage of traffic predictability at the flow level to obtain aggregate traffic and eventually burst size predictions. Specifically, we exploit the fact that newly arrived flows have to "*probe*" for network capacity, by starting their transmission with a congestion window of one segment. This is then doubled in every lossless round (i.e. a round trip with no packet loss) before the flow reaches its maximum transmission speed. This delay allows notifying in advance the core nodes of the upcoming traffic variations.

A. Formula-based prediction of TCP flow rate

In this section, we show how the evolution of the sending rate of an individual TCP flow can be predicted based on well-established TCP over OBS performance models. Specifically we predict the sending rate of a TCP flow, denoted as flow-i, as the number of segments S_i transmitted in the next time interval, termed *prediction interval*. S_i is derived as a function of flow parameters SPB_i, $\mathrm{RTT_i}$ and L_i, i.e. the flow segments per burst ratio, round trip time and flow length respectively:

$$S_i = S(SPB_i, R_i, L_i) \qquad (5)$$

We take into account both slow-start and steady-state performance of TCP flows, and we assume an idealized scenario where the trajectory followed by the TCP congestion window is totally deterministic. We assume that each TCP flow starts its transmission in the slow-start phase with a congestion window of one segment. It then exponentially grows until it reaches the steady-state window and stays there until it concludes its transmission. The fast-retransmit and time-out periods are not taken into account, as they have a small contribution to the TCP throughput. The steady state TCP window is obtained from [6], a formula which is accurate for small values of burst loss ratio p:

$$\overline{cwnd} = \min\left\{\frac{\sqrt{1.5 * SPB_i}}{\sqrt{p}}, W_m\right\} \qquad (6)$$

W_m is the TCP maximum window size (in segments) that is constrained from advertized TCP window as well as flow access rate. The instant flow congestion window during the slow-start phase, denoted as $cwnd_i$, can be directly estimated from its flow length [14], as:

$$cwnd_i = L_i/2 \qquad (7)$$

In what follows, given τ the duration of one prediction interval, we denote $r = \tau/RTT_i$ the duration of the prediction interval in TCP rounds. The number of segments transmitted from the flow during one prediction interval at the slow-start phase, starting with a congestion window of $cwnd_i$, is [14]:

$$S_i = cwnd_i * (2^r - 1) \qquad (8)$$

As soon as the flow reaches the steady state, $\overline{cwnd}$ segments are transmitted on average per round, or $r \cdot \overline{cwnd}$ segments per prediction interval. Thus, the overall number of segments transmitted in a prediction interval from *flow-i* either in the slow-start or the steady-state phase is:

$$S_i = \begin{Bmatrix} cwnd_i * (2^r - 1), & cwnd_i \leq \overline{cwnd} \\ r * \overline{cwnd}, & cwnd_i > \overline{cwnd} \end{Bmatrix} \qquad (9)$$

B. Burst Size prediction

For predicting burst sizes, we rely on the flow statistics of active TCP connections, such as flow length distribution, which are estimated in real time by the flow profiler. The flow profiler provides online estimates of the pairs $\{NF_k, length_k\}$ that correspond to the number of active flows NF_k assigned to the k-th bin of the histogram, and have a flow length of $length_k$. The distribution of active flows to bins depends on the network state and input load variations. For example, an increase on connection arrivals "propagates" from the leftmost bins to the rightmost ones, as the newly arrived flows gradually increase their congestion windows.

In the proposed burst size prediction scheme, time is divided in prediction intervals, whose duration is denoted with τ. The predictor at the end of the n^{th} prediction interval, at time $t = n\tau$, predicts the aggregate number of bytes to be transmitted in the next prediction interval that is up to $t = (n+1)\tau$. Since a lot of active flows may conclude their transmission before the ending of the prediction interval, we have to consider flow *survival probability*. Assuming X the Pareto random variable that corresponds to the flow length, and $P(x)$ the flow survival function output by the flow profiler, the *survival probability* that a flow (denoted as *flow-i*) transmits x bytes before completion, is:

$$\Pr[X > x + y_i \mid X > y_i] = \frac{P(y_i + x)}{P(y_i)} = \left(\frac{y_i}{y_i + x}\right)^{\alpha} \qquad (10)$$

where $y_i = MSS \cdot L_i$ is the byte length of flow-i and MSS the network maximum segment size. It can be seen that for long-lived flows, survival probability approaches '1', or

$\lim_{y_i \to \infty} \Pr[X > x + y_i \mid X > y_i] = 1$, which is a fundamental property of heavy-tailed distributions. Long-lived flows exceeding a pre-defined threshold (typically of a few MBs) are assigned to the rightmost histogram bin by the profiler, regardless of their actual flow length, and their survival probability is approximated as '1'. Equation (10) shows that a higher threshold leads to higher accuracy and thus a smaller overestimation in the number of active flows, but also requires a higher number of histogram bins increasing resource consumption.

The number of bytes transmitted from an active flow within one prediction interval depends on its sending rate and its expected residual life. Given SPB_i, $\mathrm{RTT_i}$ and $\mathrm{L_i}$ the parameters of flow-i, and $S(SPB_i, R_i, L_i)$ its sending rate modeled in the previous section, the expected number of bytes it transmits in the following time interval, before concluding its transmission, is obtained by integrating the flow survival function *P(.)*:

$$B(\mathrm{SPB_i}, \mathrm{RTT_i}, \mathrm{L_i}) = \int_{y_i}^{y_i + \mathrm{S(SPB_i,RTT_i,L_i)*MSS}} \frac{P(u)}{P(y_i)}\, du \qquad (11)$$

Next, we analytically derive the expected number of bytes transmitted from all active flows assigned to one histogram bin, denoted as *bin-k*. For doing so, we assume that all flows in a bin have the same length, which corresponds to the mid value of the bin range and is *length_k*. Further, using the empirical distribution of round-trip times and segments per burst estimated by the profiler, i.e. *RTT(i)* corresponds to the frequency of RTT_i value and SPB(i) to the frequency of $\mathrm{SPB_i}$ value, the expected number of bytes transmitted from all flows assigned to bin-k histogram bin, $\mathrm{H_k}$ is derived as:

$$E[H_k] = NF_k * \sum_{i=1}^{\infty} SPB(i) * \sum_{j=1}^{\infty} RTT(j) * B\big(SPB_i, RTT_j, \mathrm{L_k}\big) \qquad (12)$$

Thus, the predicted number of bytes transmitted from all active flows, assuming *M* bins in the flow histogram is:

$$\tilde{B} = \sum_{k=1}^{M} NF_k * E[H_k] \qquad (13)$$

Finally, the predicted burst size is obtained by equally dividing $\tilde{B}$ with the number of bursts created during the prediction interval τ assuming T_b is the assembly time:

$$\tilde{L} = \left(\frac{T_b}{\tau}\right) * \tilde{B} \qquad (14)$$

Clearly, this scheme assumes constant burst sizes within the same prediction interval. This assumption is valid when traffic profile within the prediction interval is relatively smooth, which depends on input load variations. However TCP traffic is expected to exhibit a degree of sub-RTT burstiness, which can't be captured by the proposed scheme. Nevertheless the accuracy in the burst size prediction can be improved using smaller interval periods and a worst case correction parameter.

IV. Performance Evaluation in OBS Networks

In this section, we measure the accuracy of the proposed TCP prediction mechanism and evaluate its effect in enhancing performance at an OBS network. In particular, we measure the rate of successful reservations for different prediction intervals and sampling rates. We argue that predicting traffic increases (or decreases) over larger time intervals in an OBS network, can lead to a lower burst loss ratio and minimize edge delays.

A. Advance reservation of resources using traffic predictions

The prediction of future burst sizes using TCP dynamics can be used for advance reservation of resources at the burst level. This is possible using either two-way OBS signaling, or even one-way signaling as detailed in [7]. The bandwidth allocated for a source destination pair is negotiated and updated based on traffic predictions. In particular, a refresh (or a new setup) message is transmitted downstream per prediction interval to communicate to the core nodes the upcoming traffic changes and to modify in advance their burst-level reservations. However prediction errors can result in under- or over-estimated burst sizes. An over-estimation would simply waste an amount of resources, but even a small under-estimation would result in an insufficient resource reservation and lead to a dropped burst. This is usually compensated with a correction parameter δ added to the predicted value $\tilde{L}$, as $(1+\delta)\tilde{L}$. The use of a correction margin is common to predictive bandwidth reservation schemes, like [8], and it reflects the percentage of the reserved bandwidth that is wasted. In the following section, we first evaluate the proposed prediction scheme and then assuming a correction parameter related to the coefficient of variation of the predicting error, we further investigate its performance in OBS networks.

B. Evaluation results

For evaluating the TCP prediction scheme, we used a simple 3-node topology in ns-2 simulator. The topology consisted of two edge nodes denoted as E1 and E2 and one core router denoted as C. Clients are assigned to edge router E1 and initiate connection requests on servers assigned to E2. The core node is a bufferless OBS node, where reservations are made using a standard two-way signaling reservation protocol, and reservations take place only for the duration of the burst starting from their arrival time. The burst assembly process is performed at the edge nodes, using a timer-based aggregation scheme ($\mathrm{T_{MAX}}$) with a threshold of 3ms. The network round trip time was set equal to 15ms, while all clients had a uniformly selected access rate of 20Mpbs, 50Mpbs and 100Mpbs.

A synthetic traffic scenario that imitates internet traffic was modeled in this topology, that consists of a mix of short-lived connection requests, with a mean size of 50KB, and a mix of long-lived file transfers with sizes drawn from a Pareto distribution and an average of 3MB. Short-lived traffic was generated with packmime HTTP 1.1 traffic generator [15], so as to model interactive web requests. The TCP profiler was developed as a separate ns-2 module and was incorporated in

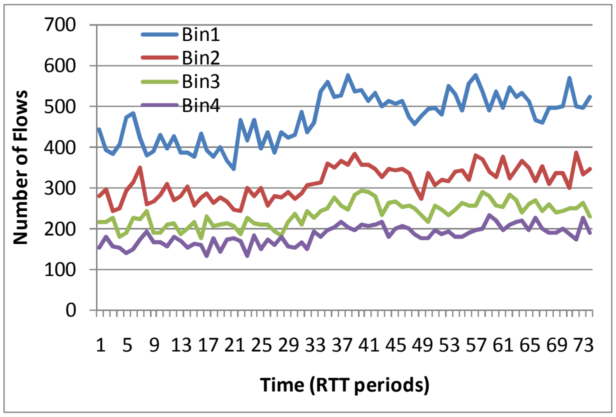

Figure 1: Evolution of number of active flows per bin

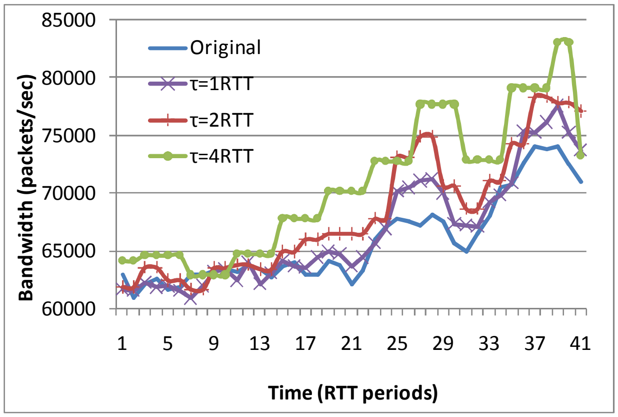

Figure 2: Evolution of TCP aggregated throughput for prediction intervals of 1, 2, 4 times the RTT.

the edge nodes. The profiler outputs the predicted burst size using flow statistics obtained from the flow profiler, analytically calculating the expected throughput of all active flows for the next prediction interval. This predicted burst size is then compared with the real value.

Figure 1 displays flow histogram evolution over time, in the form of number of flows assigned to bins 1 to 4 (further bins were omitted for clarity). The increases in the arrival rate can be clearly seen, propagating from bin-1 to bin-4, with a small lag that corresponds to the RTT. At the same time, the flow congestion window doubles per RTT, until it reaches steady-state (or the flow concludes its transmission). This lag between the bins as well as the predictability of the evolution of the congestion window is exploited by the proposed scheme to perform accurate predictions. As expected, a percentage of flows (depending on flow size density function) will conclude before being assigned to the next bin, and thus flow population decreases when moving from bin-1 to bin-4.

Figure 2 displays the aggregated TCP throughput evolution along with a snapshot of the predictor's output, for prediction intervals of 1, 2 and 4 times the RTT. It can be seen that the predicted value closely follows the true value of the aggregated throughput, while it converges fast to the changes in the flow arrival rate. However, longer prediction intervals come at the expense of a higher variance, which can be significant for long prediction intervals (i.e. 4 RTTs). In order to quantify the prediction accuracy for different prediction intervals, we also compiled the cumulative density function (CDF) of the relative prediction error of TCP throughout (see Figure 3) with a high sampling rate of N=2. From Fig. 3, it can be seen that with 1RTT prediction interval, the error in the predicted throughput is less than 4% for the majority (95%) of the prediction intervals. This deteriorates to 10% and 15% error for 2 and 4 RTT prediction intervals, because of the limitations in predicting TCP throughput evolution, which becomes more unpredictable after a few TCP rounds.

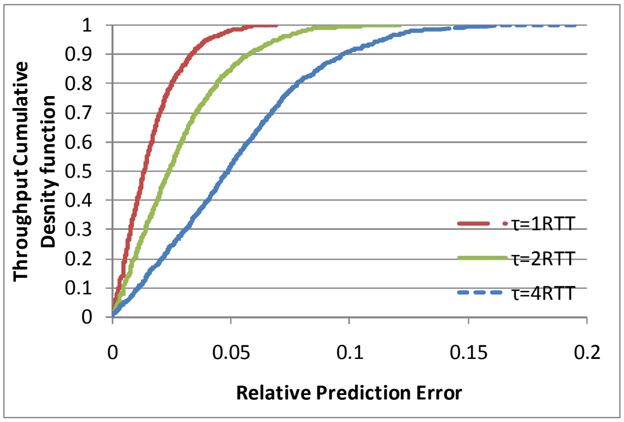

Figure 3: CDF of the aggregated throughput relative prediction error for prediction intervals of 1, 2, 4 times the RTT.

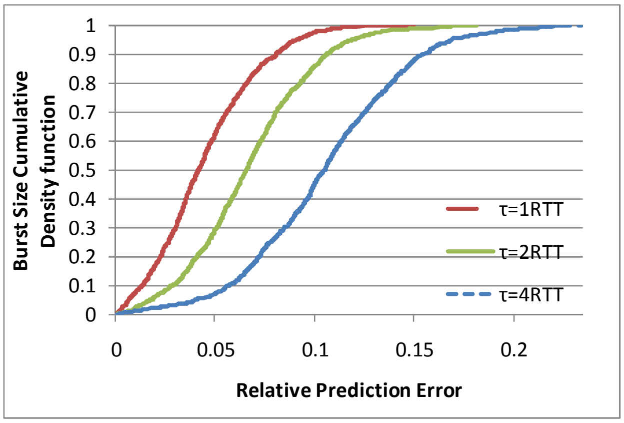

Figure 4: CDF of the burst size relative prediction error for prediction intervals of 1, 2, 4 times the RTT.

Next, we evaluated the burst size prediction accuracy of the proposed scheme. The relative prediction error between the real and the predicted burst sizes was quantified with the cumulative density function (CDF) of the relative prediction error (see Figure 4) for prediction intervals of 1, 2, 4 times the RTT. It can be seen that the error in predicting the burst sizes exceeds the error in predicting aggregated throughput (see Figure 3 and 4). This is due to the sub-RTT burstiness effect, as we have assumed the same predicted burst size for all the assembly cycles within a prediction interval. Thus, even for the smallest prediction interval of 1 RTT, there is a certain degree of variability. To this end, from Figure 4, it can be seen that the relative prediction error is less than 9% for the 95% of the bursts for a prediction interval of 1RTT. For long prediction intervals (i,e. 4 RTTs) there is a sharp increase in the prediction error, since our assumption of static traffic profile within the prediction interval does not hold any more.

TABLE I. COV METRIC VS PREDICTION INTERVAL LENGTH AND SAMPLING PERIOD

	N=2	*N=5*	*N=10*	*N=20*
1RTT	0,023	0,041	0,07	0,11
2RTT	0,038	0,05	0,076	0,11
4 RTT	0,064	0,073	0,094	0,13

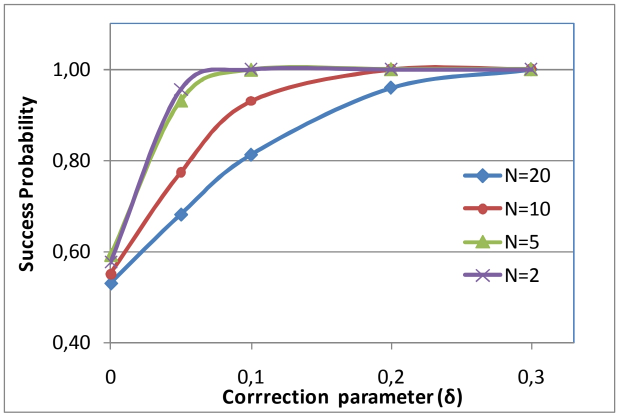

Figure 5: Probability of successful reservation for different sampling rates.

Finally, we evaluated the effect of sampling rate on burst prediction accuracy by compiling a table (see TABLE I) of the coefficients of variation (CoV) for different sampling periods N={2,5,10,20} and prediction intervals. As expected, a smaller sampling rate leads to lower prediction accuracy, due to the loss of information incurred by traffic sampling.

CoV is an important metric, as it is measures the dispersion of the burst prediction error, and can be used to derive the correction margin. As mentioned earlier, underestimating the real burst size due to prediction error may lead to data loss. For the burst reservation to succeed, prediction error must not exceed the correction margin. Given L_n the length of the n^{th} burst transmission and e_n the burst prediction error, the probability of successful reservation can be defined as $P_s = Pr(e_n < \delta * L_n)$, where δ is the correction parameter.

Assuming that the prediction error follows zero-mean Gaussian distribution (i.e. resembles white noise), the success probability can be derived as $P_s = 1 - Q(\frac{\delta}{CoV})$ where Q(.) is the distribution Q-function [8]. The correction parameter δ can then be selected according to the desired performance. For example, for $\delta = 2 \cdot CoV$ the success probability is 95%.

In Figure 5, we have calculated how the success probability varies for different correction parameters and different sampling rates. It can be seen that even with low sampling rates, it is possible to achieve high success probabilities, at the expense however of wasting part of the reserved bandwidth. In any case, we may argue that the proposed TCP specific prediction mechanism is capable of in advance modifying bandwidth reservations in the core network, thus achieving low edge delays, as in one-way signaling schemes, with the minimum data losses in the core network.

V. CONCLUSIONS

In this paper a TCP specific prediction scheme has been detailed, for enhancing performance of TCP transmission over OBS networks. The prediction process relies solely on flow statistics, estimated by a flow profiler. The latter performs running estimations of the number of active flows to obtain flow-level statistics like segments-per-burst distribution and RTTs, which are then used to estimate aggregated TCP throughput and the corresponding burst sizes. Performance evaluation results have shown that the proposed mechanism adequately profiles flow dynamics and estimates burst sizes at sub-RTT time intervals with an error of less than +/- 10% for the majority of the bursts. Burst size prediction is used to in advance reserve the appropriate resources, which can lead to a lower burst loss ratio and minimize edge delays.

ACKNOWLEDGEMENTS

The authors thank the European Social Fund (ESF), Operational Program for EPEDVM and particularly the Program Herakleitos II, for financially supporting this work. This work was also supported by the BONE-project, a NoE funded by the EU through the 7th ICT-Framework.

REFERENCES

[1] C. Qiao and M. Yoo, "Optical burst switching (OBS)-A new paradigm for an optical internet," J. High Speed Networks, vol. 8, pp 69-84, 1999.

[2] "X Yu, C Qiao, Y Liu, "TCP implementations and false time out detection in OBS networks", IEEE INFOCOM, vol.2, pp. 774- 784, 2004

[3] Qiong Zhang, V.M Vokkarane, Yuke Wang, J.P. Jue, "Analysis of TCP over optical burst-switched networks with burst retransmission," IEEE GLOBECOM , vol.4, no., pp.6 pp.-1983, 2005

[4] X. Cao, J. Li, Y. Chen, C. Qiao, "Assembling TCP/IP packets in optical burst switched networks", IEEE GLOBECOM, vol3, pp. 2808+, 2002

[5] B. Shihada, Pin-Han Ho, Qiong Zhang, "A Novel Congestion Detection Scheme in TCP Over OBS Networks," Journal of Lightwave Technology, vol.27, no.4, pp.386-395, 2009

[6] X Yu, C Qiao, Y Liu, D Towsley, "Performance evaluation of TCP implementations in OBS networks", Technical Report 2003-13, CSE Dept., SUNY, Buffalo, 2003

[7] K. Vlachos and D. Monoyios, "A Virtual One-Way Signaling Protocol With Aggressive Resource Reservation for Improving Burst Transmission Delay", IEEE/OSA Journal of Lightwave technology, 27(4), 2869+, 2009.

[8] J. Liu et al., "FRR for latency reduction and QoS provisioning in OBS networks", IEEE Sel. Areas Commun., vol. 21, no. 9, pp. 1210+, 2003

[9] T Karagiannis, M Molle, M Faloutsos, "Long-range dependence ten years of Internet traffic modeling", Internet Computing, IEEE , vol.8, no.5, pp. 57- 64, 2004.

[10] K Ramantas, K Vlachos, "Profiling TCP traffic in Optical Burst Switching Networks", ICST BROADNETS 2010, in press

[11] Aaron Clauset, Cosma R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data", SIAM Review, 51(4):661+, Feb 2009

[12] N.Duffield, C. Lund, and M. Thorup, "Properties and Prediction of Flow Statistics from Sampled Packet Streams", in Proc. ACM/SIGCOMM Internet Measurement Workshop, 2002.

[13] D. W. Scott, "Multivariate Density Estimation: Theory, Practice, and Visualization". New York: Wiley, 1992.

[14] N. Cardwell, S Savage, T Anderson, "Modeling TCP Latency" IEEE INFOCOM, vol.3, pp.1742-1751, 2000.

[15] J. Cao, W.S. Cleveland, Y. Gao, K. Jeffay, F.D. Smith, and M.C. Weigle , "Stochastic Models for Generating Synthetic HTTP Source Traffic", IEEE INFOCOM, vol.3, pp.1546-1557, 2004