Computer Communications 26 (2003) 366–375

computer communications

www.elsevier.com/locate/comcom

The TCP control block interdependence in fixed networks—new performance results

Michael Savorić*, Holger Karl, Adam Wolisz

Department of Electrical Engineering, Technical University Berlin, Einsteinufer 25, 10623 Berlin, Germany

Received 12 May 2002; accepted 14 May 2002

Abstract

In a today's Internet end system, the congestion and flow control of every TCP connection is done separately. This means that every TCP sender of the TCP connections of an end system has to determine current information about the network for itself and independently from other TCP senders. Due to TCP's congestion control and timer management, i.e. the slow start algorithm and timeout timer calculation, this can lead to suboptimal network utilization. In addition, separate control reduces fairness between simultaneous TCP connections. Both effects are caused by several TCP senders' different perception of current network conditions.

Therefore, it might be an interesting idea to reuse network information in an end system: information collected by existing TCP connections could be used to initialize control variables of new TCP connections with more up-to-date values. This can improve the overall network utilization of and the fairness between the TCP connections of an end system.

One such network information reuse approach is the TCP control block interdependence (TCBI, RFC 2140). In this paper, the performance of two different TCBI control algorithms is investigated and compared to standard TCP by simulations in a fixed network scenario. Since more and more end systems are connected to the Internet via wireless LANs, also the influence of packet losses in the last hop of a TCP connection on the performance of the TCBI control algorithms is considered.

Keywords: Transmission control protocol; Congestion control; Flow control; Network information reuse

1. Introduction

One of the key mechanisms the Internet is based on is the congestion control provided by the transmission control protocol (TCP) [1,2]. Its main purpose is to protect the network infrastructure from collapsing from overload by matching a sender's offered load to the capacity that is actually available in the network. Initially, a TCP connection is only allowed to inject very few packets at the sender into the network. As acknowledgements for these packets arrive, signaling not only the arrival of the packets at the receiver but also the network's ability to support this additional load, the sending rate is slowly increased. This increase continues until packets are lost due to congestion, continually probing the network and constructing an implicit estimation of the current network capacity available between the sender and the receiver.

A TCP connection's capacity estimation is stored in a number of variables, e.g. the congestion window size, the slow start threshold, the (smoothed) round trip time, or the round trip time variance. These control variables are stored in a memory block called TCP control block (TCB). At the start of a new TCP connection, the control variables are initialized with fixed and standardized values and during the connection's lifetime, these variables are continually updated to reflect the information obtained so far about the network conditions in the path to the TCP receiver (see Fig. 1). As these control variables are maintained on a per-connection basis, each connection has to undergo the network estimation process separately. However, when a new connection is opened to a receiver to which other connections already exist, these existing connections already have collected information about the network status. It might be an interesting idea to reuse this information for new connections, instead of requiring them to independently probe the network, wasting time in slowly converging to the actual network congestion status. Thus, for the new TCP connection the fixed initial values of the control variables of standard TCP can be replaced with more adequate initial values derived from given network

* Corresponding author.
E-mail addresses: savoric@ee.tu-berlin.de (M. Savorić), karl@ee.tu-berlin.de (H. Karl), awo@ieee.org (A. Wolisz).

0140-3664/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0140-3664(02)00155-X

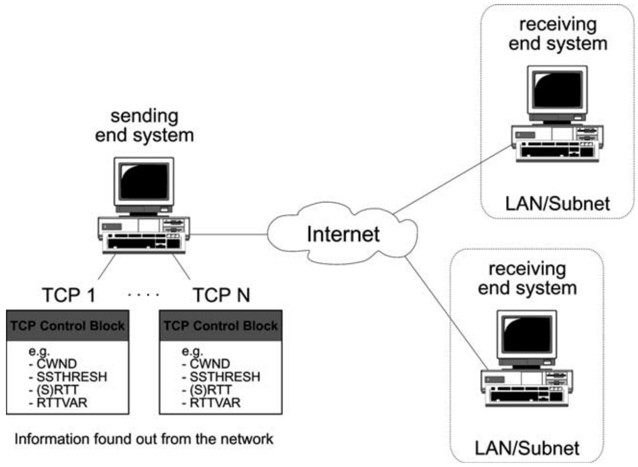

Fig. 1. The TCP control blocks of N TCP senders of an end system.

information. To compensate for this higher-than-allowed sending rate of a new TCP connection, the sending rate of already existing connections should be proportionally reduced. Since the TCP sender of the new TCP connection can directly access suitable network information, it can use an optimized sending rate from the beginning, improving its throughput. Additionally, the fairness of capacity sharing between several connections can potentially be improved.

Several approaches for reusing network information have been proposed [3–7]. These approaches are either located in the transport or in the application layer of the protocol stack. One of the transport layer approaches, the TCP control block interdependence (TCBI) [3], is considered in more detail in this paper.

In order to use TCBI, only the sending end system must have TCBI capabilities (a *TCBI end system*, for short). TCBI information sharing is only reasonable between connections from a TCBI end system to a single receiver system or to a number of receiver systems belonging to the same subnet. Such TCP connections form a *TCBI connection set*. Which, when, and how information is shared between connections belonging to the same TCBI connection set depends on the actual implementation of the algorithms of *TCBI controller* in the TCBI end system. Since in Ref. [3] only the idea of reusing network information is described and no algorithms are explained, the design principles and implementation of the TCBI controller's algorithms can be freely selected.

A network information reuse approach located in the application layer is the hypertext transfer protocol (http) in its current version 1.1 [7]. It is used in many WWW servers and provides a network information reuse mechanism for multiple http transactions to one destination. But the TCBI approach supports much more functionalities. It is able to reuse network information not only between multiple http transactions to one destination but also between multiple protocol transmissions to one destination or to many destinations in the same subnet. And this network information reuse is transparently done for all applications of an end system with TCBI capabilities. Therefore, this paper concentrates on the TCBI approach.

An earlier investigation of the TCBI approach [8] with two simple TCBI controllers, which consider only the congestion window size of the TCP connections in a TCBI connection set, has shown under what conditions and for which TCP connections the TCBI approach can result in performance gain and how large this gain can be. In this paper, the same simulation topology but with a higher background traffic load is used to investigate two more complex TCBI controllers. In addition to the congestion window size, these controllers take into account the slow start threshold, the smoothed round trip time, and the round trip time variance of the TCP connections in a TCBI connection set.

The TCBI can be used not only for new TCP connections. In a scenario with TCP receivers in mobile end systems the TCBI can be used after a handover of a mobile TCP receiver for two reasons: First and similar to the TCBI of a new TCP connection, to increase the throughput and fairness of the handover TCP connection. One possible means would be to use a congestion window size larger than the current one (if possible). And second, to reduce the probability of congestion in the new subpath of the handover TCP connection between the handover switching node in the network and the new base station. This could be accomplished by using, for example, a smaller congestion window size than the current one (if necessary). The latter case is the more important one from the network point of view, since congestion negatively influences the throughput of all TCP connections over the congested part of the network. In this paper, TCBI is only investigated in a fixed network scenario with TCP receivers in stationary end systems. The case of TCP receivers in mobile end systems and the TCBI for handover TCP connections is part of our current research activities.

The remainder of this paper is organized as follows: Section 2 shows the feasibility and usability of the TCP control block interdependence in fixed networks. In Section 3, the algorithms of the two investigated ensemble TCBI controllers are explained. The simulation model and the simulation scenarios are described in Section 4. The evaluation metrics and the statistical evaluation methods are considered in Sections 5 and 6. In Section 7, the simulation results are shown and discussed. Finally, Section 8 contains the conclusion of this paper and a description of some of our future research activities.

2. TCP control block interdependence in fixed networks

In the introduction, the general concept of sharing TCP control block information has been described, using the notion of a controller that manages the information exchange. Such a controller is an abstract entity which needs to be specified further to determine a concrete, implementable and testable functionality. In particular, the following questions need to be answered:

Which connections belong to a connection set? If the TCP connections of a TCBI end system have TCP receivers in different end systems how fine should the granularity be chosen to determine that two or more of these TCP connections have TCP receivers in end systems in the same part of the network and can form a TCBI connection set? If the precise segmentation of a class A, B, or C network in subnetworks is not known, only the network part of a class A, B, or C network address can be used for that. But this might be too inaccurate in most cases.

How many connections of an end system can benefit? The proportion of TCP connections of an end system that can benefit from a network information reuse approach like the TCBI strongly depends on the type of the end system. For example, the numerous TCP connections of a large WWW or proxy server have a higher probability of using the TCBI approach than the few TCP connections of an ordinary end system.

How long is congestion information valid? The TCBI can be used not only between a new TCP connection and TCP connections which exist in parallel (ensemble TCBI), but also between a new TCP connection and one or more recently closed TCP connections (temporal TCBI). In the latter TCBI variant the network information reuse should be combined with an aging algorithm and a lifetime limit for the cached values to avoid a reuse of outdated network information. But how long can be the duration between a terminated and a new TCP connection such that the information which the terminated TCP connection has determined is still meaningful? Otherwise, the cached network information might be totally out-of-date and detrimental for the performance of the new TCP connection. Measurements [9] have shown that a proper observation of the available bandwidth on a path seen by one TCP connection can be used for other TCP connections on the same path as a fairly good prediction for the available bandwidth up to time periods on the order of tens of minutes. But in a more dynamic network load scenario this time period for reusing network information must be drastically decreased.

When to exchange information between connections? The information reuse between TCP connections of a TCBI connection set can be done once, e.g. at the start of a new TCP connection, or dynamically during the whole lifetime of the TCP connections belonging to the same TCBI connection set. This decision can also depend on the TCP control variables for which the information reuse is done. One reasonable example is that all TCP connections of a TCBI connection set use only one common value each for the smoothed round trip time, the round trip time variance, and the derived timeout timer. Another possibility is to use an aggregated congestion window size and share it (dynamically) over all TCP connections of the TCBI connection set. This is quite similar to the ideas behind the TCP implementation in common congestion control approaches like the ensemble TCP (E-TCP) [4] or the congestion manager (CM) [5].

Depending on these decisions, temporal and/or ensemble TCBI controllers with different functionalities and behavior can be defined. In the remainder of this paper only ensemble TCBI controllers are investigated.

3. Ensemble TCBI controllers

Two different ensemble TCBI controllers are used to compare standard TCP in an end system with the ensemble TCBI approach. Both ensemble TCBI controller variants use the congestion window sizes, the slow start thresholds, the smoothed round trip times, and the round trip time variances of the existing TCP connections in a TCBI connection set to calculate initial values for the congestion window size, the slow start threshold, the smoothed round trip time, and the round trip time variance of a new TCP connection. With observations of these TCP control variables accurate values for the additional load into the network and the timeout timer of the new TCP connection can be derived.

In addition to the variables stored in the control blocks of the existing TCP connections of a TCBI connection set, both TCBI controllers need to store an aggregated smoothed round trip time and an aggregated round trip time variance per TCBI connection set.

The two TCBI controllers described here differ mainly in their treatment of existing TCP connections when a new TCP connection starts: the mean value TCBI controller leaves them unaffected, the fair share TCBI controller influences also the existing TCP connections.

3.1. Mean value TCBI (MV-TCBI) controller

The algorithms of the mean value TCBI (MV-TCBI) controller are described in the following list:

Congestion window size: The MV-TCBI controller computes the mean of the current congestion window sizes of the existing TCP connections of a TCBI connection set and assigns this value as the initial congestion window size to the new TCP connection. If no other TCP connection is in the TCBI connection set of the new TCP connection then the initial congestion window size of the new TCP connection is set to the standard value 2. Only the initial congestion window size of the new TCP connection is changed.

Slow start threshold: The MV-TCBI controller computes the mean of the current slow start thresholds of the existing TCP connections of a TCBI connection set and assigns this value as the initial slow start threshold to the new TCP connection. If no other TCP connection is in the TCBI connection set of the new TCP connection then the initial slow start threshold of the new TCP connection is set to the standard value 64. Only the initial slow start threshold of the new TCP connection is changed.

Smoothed round trip time: The MV-TCBI controller uses the current value of an aggregated smoothed round trip time of the existing TCP connections of a TCBI connection set as the initial smoothed round trip time of the new TCP connection. If no other TCP connection is in the TCBI connection set of the new TCP connection then the initial smoothed round trip time of the new TCP connection is set to the standard value. Only the initial smoothed round trip time of the new TCP connection is changed.

The aggregated smoothed round trip time is updated after every change of the smoothed round trip time of one of the n TCP connections in a TCBI connection set by a weighted calculation of $(n-1)/n$ times the last value of the aggregated smoothed round trip time plus $1/n$ times the new smoothed round trip time.

Round trip time variance: The MV-TCBI controller uses the current value of an aggregated round trip time variance of the existing TCP connections of a TCBI connection set as the initial round trip time variance of the new TCP connection. If no other TCP connection is in the TCBI connection set of the new TCP connection then the initial round trip time variance of the new TCP connection is set to the standard value. Only the initial round trip time variance of the new TCP connection is changed.

The aggregated round trip time variance is updated after every change of the round trip time variance of one of the n TCP connections in a TCBI connection set by a weighted calculation of $(n-1)/n$ times the last value of the aggregated round trip time variance plus $1/n$ times the new round trip time variance.

3.2. Fair share TCBI (FS-TCBI) controller

The algorithms of the fair share TCBI (FS-TCBI) controller are described in the following list:

Congestion window size: The FS-TCBI controller computes the sum of all current congestion window sizes of the existing TCP connections of the TCBI connection set plus the standard initial congestion window size 2. This value is used to calculate a congestion window size fair share by an arithmetic mean value computation for all TCP connections in the TCBI connection set. At the beginning of a new TCP connection, all TCP connections of the TCBI connection set get this congestion window size fair share as their new congestion window size.

Slow start threshold: The FS-TCBI controller computes the sum of all current slow start thresholds of the existing TCP connections of the TCBI connection set plus the standard initial slow start threshold 64. This value is used to calculate a slow start threshold fair share by a mean value computation for all TCP connections in the TCBI connection set. At the beginning of a new TCP connection, all TCP connections of the TCBI connection set get this slow start threshold fair share as their new slow start threshold.

Smoothed round trip time: The FS-TCBI controller uses the same algorithm for the initial smoothed round trip time calculation as the MV-TCBI controller. At the beginning of a new TCP connection all TCP connections of the TCBI connection set are assigned this smoothed round trip time calculation result as their new smoothed round trip time.

Round trip time variance: The FS-TCBI controller uses the same algorithm for the initial round trip time variance calculation as the MV-TCBI controller. At the beginning of a new TCP connection all TCP connections of the TCBI connection set are assigned this round trip time variance calculation result as their new round trip time variance.

In the current implementation of both TCBI controllers no pacing algorithm is used, i.e. the first segments of every new TCP connection are sent in a burst.

The FS-TCBI controller is less aggressive to the network than the MV-TCBI controller. With the FS-TCBI controller and after a start of a new TCP connection the sum of all congestion window sizes or slow start thresholds of the TCP connections in the TCBI connection set is only one standard initial congestion window size or initial slow start threshold higher than before the new TCP connection has started. This is equivalent to the situation after the start of a new standard TCP connection. In addition, the FS-TCBI controller is fair to all TCP connections belonging to the same TCBI connection set.

The main topic of this paper is to show the power of the FS-TCBI controller compared to standard TCP. The MV-TCBI controller is used to indicate that a too aggressive TCBI controller is detrimental for the overall performance under specific network conditions.

4. Simulation model

The performance evaluation of the TCBI approach is done by simulations. Our simulation model consists of a network topology, different traffic load models for the (optionally) TCBI-controlled and background TCP connections, and two scenarios taking into account the influence of a reliable or unreliable packet transport in the last hop of the network topology on the performance of the TCBI approach. In this section, we describe all parts of our simulation model in detail.

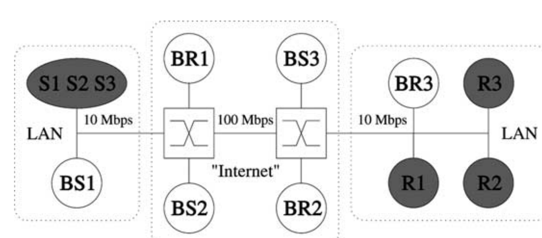

Fig. 2. Structure of the simulated fixed network.

Table 1
Distributions and parameters for the stochastic variables of the simplified WWW model

Stochastic variable	Distribution	Distribution parameter (s)
Inter-session time	Exponential	$\mu = 5.0$ s
Pages per session	Lognormal	$\mu = 25.807$ pps $\sigma = 78.752$ pps
Inter-page time (reading time)	Gamma	$\mu = 35.286$ s, $\sigma = 147.390$ s
Page size	Pareto	$\alpha = 1.7584$, $\beta = 30458$ bytes

4.1. Simulated network topology

The topology of the simulated fixed network is shown in Fig. 2. The simulation network consists of several TCP senders (S1,…, S3 and BS1,…, BS3) and TCP receivers (R1,…, R3 and BR1,…, BR3), two routers, and two Ethernet-type wired LANs. The routers are connected via links with a bit rate of 100 Mbps and a propagation delay of 10 ms. For each link the routers have a queuing capacity of 20 IP packets.

The TCP senders S1, S2, and S3 are located in an end system with TCBI capabilities. The other TCP senders BS1, BS2, and BS3 are located in different end systems and stress the network of the simulation model with background traffic. All the TCP receivers BR1, BR2, and BR3 of the background traffic and the TCP receivers R1, R2, and R3 of the TCBI end system are located in stationary end systems.

The end systems of the background traffic TCP senders BS2 and BS3 are connected to the network via links with a bit rate of 100 Mbps and a propagation delay of 0.25 ms. The end system of the background traffic TCP sender BS1 and the TCBI end system are connected to the network via the sender LAN. The end systems of the TCP receivers R1, R2, and R3 and the background traffic TCP receiver BR3 are connected to the network via the receiver LAN. Both LANs have a bit rate of 10 Mbps and a propagation delay of 0.5 μs. The propagation delay of the LANs are derived from typical LAN installations. The packet loss rate in the receiver LAN is adjustable to investigate the influence of different packet loss probabilities in the last hop of a TCP connection on the overall throughput of the considered TCP connections.

In the simulation model, two different TCP load classes are used: the first class consists of short TCP connections with five segments to send and the second class includes TCP connections whose number of segments to send is determined by a WWW traffic model [10]. This traffic model is derived from real HTTP traces in corporate and educational environments and uses three abstraction levels: the session level, the page level, and the packet level. Here, a simplified version of this WWW traffic model is used which consists only of the first two levels. In every WWW session a lognormally distributed number of WWW pages with Pareto-distributed page sizes are sent. The time between the pages, i.e. the inter-connection time or reading time, is gamma distributed. The load in the network can be easily adjusted by using different parameters for the exponentially distributed session interarrival time. The inter-connection time of TCP connections derived from the first TCP load class of TCP connections is also gamma distributed. Table 1 shows the distributions and parameters for the stochastic variables of the simplified WWW model used for some of the (optionally) TCBI-controlled TCP connections in the simulation model.

The TCBI end system establishes TCP connections from both TCP load classes: one short TCP connection and two WWW TCP connections. All the background traffic TCP connections are modeled by the second connection class with modified inter-connection time and session interarrival time distributions, where connections are immediately restarted once they have terminated. This is done to reach a higher background load in the network with these few background traffic TCP senders.

The whole simulation model is implemented in ns-2 (version 2.1b8a). For all standard TCP connections, the ns implementation of a TCP Newreno sender is used. The TCP connections of a TCBI end system are instances of a new TCP sender class (MS_TCP) derived from the ns implementation of a TCP Newreno sender.[1] This new TCP sender class provides additional information reuse and common congestion control mechanisms between the TCP connections of a TCBI connection set and some statistical performance evaluation methods.

4.2. Simulation scenarios

The simulation model is used to investigate two different simulation scenarios where the receiver LAN is either reliable or unreliable. In the simulation scenario with a reliable receiver LAN no packet losses occur in the receiver LAN. In an unreliable receiver LAN packets are lost in the receiver LAN with a given packet loss rate (PLR). In addition, due to the background traffic and the TCP congestion control behavior in both simulation scenarios also some packets can be lost in the routers of the network if they are congested.

[1] The ns implementation of the TCP Newreno sender and the TCP receiver does not perform the connection setup/teardown of a TCP connection. In future investigations of the TCBI approach also the influence of the TCP connection setup/teardown protocol mechanism on the overall throughput of a TCP connection will be considered.

With these simulation scenarios the influence of the TCP control block interdependence on the throughput of the TCP connections in fixed networks with different packet loss properties in the last hop can be investigated.

5. Evaluation metric

In each simulation and for all new TCP connections, the mean throughput, the mean initial congestion window size, the mean initial slow start threshold, the mean initial smoothed round trip time, and the mean initial round trip time variance are compared between standard TCP and the TCBI controllers. In addition, also a fairness index between TCP connections of a TCBI connection set existing in parallel is used to compare standard TCP with the TCBI TCP approach. If n TCP connections exist in parallel during a period of time and reach the mean throughputs $\bar{t}_i$, $1 \leq i \leq n$, in this period of time, then for these TCP connections a fairness index I_f for this period of time can be computed as follows [11]:

$$I_f = \frac{\left(\sum_{i=1}^{n} \bar{t}_i\right)^2}{n \cdot \sum_{i=1}^{n} \bar{t}_i^2} \text{ with } \frac{1}{n} \text{ (very bad)} \leq I_f \leq 1 \text{ (excellent)}.$$

However, only those TCP connections are considered in the following comparison which either are controlled by the ensemble TCBI approach or are not controlled but could be controlled by the ensemble TCBI approach, since at least one parallel TCP connection to the same LAN is already established and useful information found out from the network is available. This evaluation metric is used since in general the share of TCBI TCP connections on all TCP connections depends on the type of the TCBI end system (cf. Section 2). The chosen evaluation metric shows the performance of the TCBI approach independent of the type of the TCBI end system. The overall performance of the TCBI approach on the throughput of all TCP connections can then be computed by using the share of the TCBI TCP connections on all TCP connections of the considered TCBI end system.

To compare the standard TCP with the TCBI controllers, two different mean throughput computations for the considered TCP connections of the two TCP load classes are used. If one considered TCP connection has sent s segments in duration d, then the two mean throughput calculations work as follows:

- Computation of the overall mean throughput ($\bar{T}_1$) : The sum of sent segments of all n considered TCP connections is divided by the overall duration of these TCP connections, i.e.:

$$\bar{T}_1 = \frac{\sum_{i=1}^{n} s_i}{\sum_{i=1}^{n} d_i}$$

- Computation of the connection-oriented mean throughput ($\bar{T}_2$) : For each of the n considered TCP connections a mean throughput t is calculated. All these mean throughput values are then used to compute the overall mean throughput of the considered TCP connections by a normal non-weighted arithmetic mean calculation independent of the number of segments sent by each of the considered TCP connections, i.e.:

$$\bar{T}_2 = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{s_i}{d_i} = \frac{1}{n} \cdot \sum_{i=1}^{n} t_i$$

The first throughput calculation is the more important one, since with this throughput metric the overall throughput of the different TCP controllers can be evaluated. In addition, the second throughput calculation gives a connection-oriented mean throughput which can be understood as the mean throughput a single TCP connection of one of the two TCP load classes can expect.

6. Statistical evaluation method

For the simulated scenarios with either a reliable or an unreliable last hop the results of the standard TCP (no TCBI) controller are compared with the results of both the MV-TCBI controller and the FS-TCBI controller.

The statistical evaluation method used for this comparison is called the t-test for unpaired observations of two alternatives and is described in detail in [12]. The main idea of this method is to compute a confidence interval for the difference of the mean values of both alternatives for a given confidence level. Then the decision criterion is:

- If the confidence interval includes zero, then the two alternatives can not be distinguished.
- If the confidence interval is above/below zero, then the first/second alternative is the better one.

Tests with confidence intervals give not only a yes–no answer like other hypothesis tests, they also give an answer to the question how precise the decision is. A narrow confidence interval indicates that the precision of the decision is high whereas a wide confidence interval indicates that the precision of the decision is rather low.

This t-test for unpaired observations of two alternatives is used for the statistical evaluation of the simulation results for the overall mean throughput ($\bar{T}_1$) and the connection-oriented mean throughput ($\bar{T}_2$) of a new TCP connection and the overall mean throughput ($\bar{T}_1$), the connection-oriented mean throughput ($\bar{T}_2$), and the mean fairness index ($\bar{I}_f$) for parallel TCP connections of the standard TCP and both the MV-TCBI controller and the FS-TCBI controller.

In all tables showing the statistical evaluation results of the simulations, for each confidence interval also the confidence level (0.90, 0.95 or 0.99) for which the simulation results are significantly different is denoted. If the simulation results are not significantly different even for the confidence level 0.90, then the confidence interval for the confidence level 0.90 is shown.

7. Simulation results

The simulation results for a fixed network with an either reliable or unreliable last hop (packet loss rate of 5%) are shown in Sections 7.1 and 7.2.

In every following table, TCP 1 is a short TCP connection and TCP 2 and TCP 3 are WWW TCP connections. The simulation results of one simulation scenario shown in the tables are mean values computed over five independent simulation runs for this simulation scenario. Both stated mean throughput metrics ($\bar{T}_1$, $\bar{T}_2$) of the considered TCP connections are measured in TCP segments per second; in every TCP segment the payload is set to 1000 bytes. For the TCP connections entering a TCBI connection set also the mean initial congestion window size ($\overline{cwnd}$), the mean initial slow start threshold ($\overline{ssthresh}$), the mean initial smoothed round trip time ($\overline{srtt}$), and the mean initial round trip time variance ($\overline{rttvar}$) are shown. For standard TCP connections the mean initial smoothed round trip time and the mean initial round trip time variance are not applicable (N/A) for the computation of the initial timeout timer, i.e. the initial timeout timer is set to the fixed standard value. For TCP connections existing in parallel both mean throughput metrics ($\bar{T}_1$, $\bar{T}_2$) and the mean fairness index ($\bar{I}_f$) are shown.

For both simulation scenarios, the three different controllers (standard TCP/no TCBI, MV-TCBI, FS-TCBI) are investigated for a simulated time of 250 000 s in each case. In this simulated time, approximately 23 000 TCP connections starting at the TCBI end system can be observed. Only some of them are controlled or could be controlled by the ensemble TCBI approach. In the simulation model and with the chosen traffic load model the possibility to have TCP connections in parallel and use the ensemble TCBI for a new TCP connection is relatively low. Averaging over all simulations shows that approximately 4.5% of the new TCP connections in the reliable last hop scenario and approximately 9.0% of the new TCP connections in the unreliable last hop scenario are controlled or could be controlled by the ensemble TCP control block interdependence.

7.1. Scenario 1: ensemble TCBI with a reliable last hop

Table 2 shows the simulation results of simulation scenario 1, reliable last hop.

New short TCP connections benefit from the MV-TCBI controller with a large gain of approximately 49% for the overall mean throughput and of approximately 33\for the connection-oriented mean throughput compared to standard TCP. For new WWW TCP connections no difference in the overall mean throughput can be observed between the standard TCP and the MV-TCBI controller. Looking at the connection-oriented mean throughput the MV-TCBI controller achieves only a slight throughput increase.

Table 2
Simulation results for scenario 1 – $\bar{T}_1$ is the overall mean throughput, $\bar{T}_2$ is the connection-oriented mean throughput, $\bar{I}_s$ is the fairness index; TCP 1 is a short TCP connection, TCP 2 and TCP 3 are WWW TCP connections

		no TCBI	MV-TCBI	FS-TCBI
$\bar{T}_1$ of a new TCP connection [segments/s]	TCP 1	12.06	17.95	18.32
	TCP 2	73.55	73.51	78.80
	TCP 3	70.98	71.12	77.34
$\bar{T}_2$ of a new TCP connection [segments/s]	TCP 1	24.47	32.54	31.29
	TCP 2	85.52	88.59	91.61
	TCP 3	85.53	89.13	90.28
$\overline{\text{cwnd}}$ of a new TCP connection [segments]	TCP 1	2.00	19.17	10.85
	TCP 2	2.00	14.05	7.93
	TCP 3	2.00	13.65	7.85
$\overline{\text{ssthresh}}$ of a new TCP connection [segments]	TCP 1	64.00	60.36	61.88
	TCP 2	64.00	60.05	62.13
	TCP 3	64.00	60.53	62.10
$\overline{\text{srtt}}$ of a new TCP connection [ns-2 time unit]	TCP 1	N/A	7.35	7.62
	TCP 2	N/A	6.30	6.15
	TCP 3	N/A	6.33	6.15
$\overline{\text{rttvar}}$ of a new TCP connection [ns-2 time unit]	TCP 1	N/A	3.80	3.88
	TCP 2	N/A	3.28	3.16
	TCP 3	N/A	3.17	3.19
$\bar{T}_1$ of parallel TCP connections [segments/s]		60.06	60.72	59.77
		49.91	52.52	51.03
		0.8036	0.9151	0.9190

New short TCP connections benefit from the FS-TCBI controller with a large gain of approximately 52% for the overall mean throughput and of approximately 28\for the connection-oriented mean throughput compared to standard TCP. For new WWW TCP connections the FSV-TCBI controller achieves a throughput increase of approximately 6–9% for both throughput metrics.

Compared to the standard TCP, the large fairness improvement of both TCBI controllers is remarkable.

In the simulations with the TCBI controllers and with the observed probability of using the ensemble TCBI in the TCBI end system the mean throughput of the background TCP connections is not negatively affected by the TCBI TCP connections.

Table 3 shows the significance of the simulation results depending on the chosen confidence level $1 - \alpha$.

7.2. Scenario 2: ensemble TCBI with an unreliable last hop

Table 4 shows the simulation results of simulation scenario 2, unreliable last hop.

New short TCP connections benefit from the MV-TCBI controller with a large gain of approximately 95% for the overall mean throughput and of approximately 16\for the connection-oriented mean throughput compared to standard TCP. For new WWW TCP connections the MV-TCBI

Table 3
Statistical evaluation of the simulation results for scenario 1 $\bar{T}_1$ is the overall mean throughput, $\bar{T}_2$ is the connection-oriented mean throughput, $\bar{I}_f$ is the fairness index; TCP 1 is a short TCP connection, TCP 2 and TCP 3 are WWW TCP connections

		no TCBI ↔ MV-TCBI	no TCBI ↔ FS-TCBI
$\bar{T}_1$ of a new TCP connection[$1-\alpha$:conf()]	TCP 1	0.99:(− 9.52, − 2.26)	0.99: (−8.82, − 3.71)
	TCP 2	0.90: (−7.96, +8.04)	0.90: (−12.05, +1.55)
	TCP 3	0.90: (−8.39, +8.11)	0.90: (−14.34, +1.62)
$\bar{T}_2$ of a new TCP connection [$1-\alpha$:conf()]	TCP 1	0.99: (−10.47, −5.67)	0.99: (−8.80, −4.84)
	TCP 2	0.90: (−7.34, +1.20)	0.95: (−10.40, −1.78)
	TCP 3	0.90: (−6.70, −0.50)	0.99: (−8.57, −0.93)
$\bar{T}_1$ of parallel TCP connections [$1-\alpha$:conf()]		0.90: (−3.88, +2.56)	0.90: (−2.54, +3.12)
$\bar{T}_2$ of parallel TCP connections [$1-\alpha$:conf()]		0.99:(− 3.97, −1.25)	0.90: (−2.30, +0.06)
$\bar{I}_f$ of parallel TCP connections [$1-\alpha$:conf()]		0.99: (−0.13, −0.10)	0.99: (−0.13, −0.10)

controller achieves a overall mean throughput gain of approximately 25–35% compared to standard TCP. But for the connection-oriented mean throughput the standard TCP controller is slightly better.

New short TCP connections benefit from the FS-TCBI controller with a large gain of approximately 108% for the overall mean throughput and a slight gain of approximately 6% for the connection-oriented mean throughput compared to standard TCP. For new WWW TCP connections the FS-TCBI controller achieves a large throughput increase of approximately 40% for the overall mean throughput and a slight throughput increase of approximately 4% for the connection-oriented mean throughput.

There is no fairness gain of the TCBI controllers compared to the standard TCP. In such a scenario with frequent packet losses a high fairness between TCP connections of a TCBI connection set can only be achieved if the network information reuse between these TCP connections is permanently done and not only once at the start of a new TCP connection of a TCBI connection set.

In the simulations with the TCBI controllers and with the observed probability of using the ensemble TCBI in the TCBI end system, the mean throughput of the background TCP connections is not negatively affected by the TCBI TCP connections.

Table 5 shows the significance of the simulation results depending on the chosen confidence level $1-\alpha$.

Table 4
Simulation results for scenario 2 − $\bar{T}_1$ is the overall mean throughput, $\bar{T}_2$ is the connection-oriented mean throughput, $\bar{I}_f$ is the fairness index; TCP 1 is a short TCP connection, TCP 2 and TCP 3 are WWW TCP connections

		no TCBI	MV-TCBI	FS-TCBI
$\bar{T}_1$ of a new TCP connection [segments/s]	TCP 1	5.60	10.92	11.67
	TCP 2	33.71	45.51	48.17
	TCP 3	34.34	42.83	47.68
$\bar{T}_2$ of a new TCP connection [segments/s]	TCP 1	57.41	66.50	60.64
	TCP 2	93.76	91.28	97.47
	TCP 3	94.74	88.49	98.20
$\overline{\text{cwnd}}$ of a new TCP connection [segments]	TCP 1	2.00	5.50	3.80
	TCP 2	2.00	4.69	3.22
	TCP 3	2.00	4.50	3.39
$\overline{\text{ssthresh}}$ of a new TCP connection [segments]	TCP 1	64.00	25.42	44.86
	TCP 2	64.00	34.70	49.19
	TCP 3	64.00	33.69	49.24
$\overline{\text{srtt}}$ of a new TCP connection [ns-2 time unit]	TCP 1	N/A	2.63	2.61
	TCP 2	N/A	2.28	2.23
	TCP 3	N/A	2.31	2.27
$\overline{\text{rttvar}}$ of a new TCP connection [ns-2 time unit]	TCP 1	N/A	1.25	1.24
	TCP 2	N/A	1.07	1.05
	TCP 3	N/A	1.10	1.08
$\bar{T}_1$ of parallel TCP connections [segments/s]		37.47	40.75	44.12
$\bar{T}_2$ of parallel TCP connections [segments/s]		77.72	73.97	76.31
$\bar{I}_f$ of parallel TCP connections []		0.7948	0.8088	0.7998

7.3. Summary

Based on the two simulation scenarios the following statements can be made: In the simulation scenario with a reliable last hop both TCBI-controllers achieve the expected large throughput gains for new short TCP connections compared to standard TCP. New WWW TCP connections only slightly benefit from the TCBI controllers with a throughput gain of at most 9% compared to standard TCP. But the fairness between TCP connections existing in parallel is substantially improved by both TCBI controllers.

In the simulation scenario with an unreliable last hop both TCBI-controllers achieve the expected large throughput gains for new short TCP connections compared to standard TCP. Moreover, the overall mean throughput of new WWW TCP connections is significantly improved by both TCBI controllers. But no fairness gain between parallel TCP connections can be achieved by the TCBI controllers. In this case, only a common congestion control between these TCP connections can lead to an improved fairness.

In both simulation scenarios the FS-TCBI controller achieves better results than the MV-TCBI controller for the

Table 5
Statistical evaluation of the simulation results for scenario 2 − $\bar{T}_1$ is the overall mean throughput, $\bar{T}_2$ is the connection-oriented mean throughput, $\bar{I}_f$ is the fairness index; TCP 1 is a short TCP connection, TCP 2 and TCP 3 are WWW TCP connections

		no TCBI ↔ MV-TCBI	no TCBI ↔ FS-TCBI
$\bar{T}_1$ of a new TCP connection[1 − α:conf()]	TCP 1	0.99: (−6.64, − 4.01)	0.99: (−7.96, −4.19)
	TCP 2	0.99: (−16.92, −6.68)	0.99: (−20.32, −8.60)
	TCP 3	0.95: (−14.01, −2.97)	0.99: (−21.25, −5.43)
$\bar{T}_2$ of a new TCP connection [1 − α:conf()]	TCP 1	0.99: (−11.41, −6.77)	0.99: (−5.61, −0.85)
	TCP 2	0.90: (−0.31, +5.27)	0.95: (−7.36, −0.06)
	TCP 3	0.99: (+1.26, +11.24)	0.95: (−6.87, −0.05)
$\bar{T}_1$ of parallel TCP connections [1 − α:conf()]		0.99: (−5.37, −1.19)	0.99: (−8.93, −4.37)
$\bar{T}_2$ of parallel TCP connections [1 − α:conf()]		0.90: (−6.32, +13.82)	0.90: (−8.68, +11.50)
$\bar{I}_f$ of parallel TCP connections [1 − α:conf()]		0.99: (−0.03, −0.00)	0.90: (−0.01, +0.00)

overall mean throughput of new TCP connections. Since the FS-TCBI controller is less aggressive to the network than the MV-TCBI controller, the FS-TCBI controller can take advantage of its properties in such a network scenario with a higher load. And it can be expected that the FS-TCBI controller outperforms the MV-TCBI controller in all performance metrics if the background traffic load will be further increased.

In the simulation scenario with an unreliable last hop the relative difference of the connection-oriented mean throughput between the standard TCP and the MV-TCBI or FS-TCBI controller is smaller than in the reliable last hop scenario. This result has two reasons: The initial congestion window size for a TCBI TCP connection in the case of an unreliable last hop is smaller than in the case of a reliable last hop, since also the parallel TCP connections are affected by packet losses and therefore have a smaller mean congestion window size. And the first packet loss of a TCBI TCP connection reduces its congestion window size to the standard value and nearly eliminates the performance improvement which can be otherwise expected from a higher initial congestion window size.

It is remarkable that in the simulation scenario with an unreliable last hop the connection-oriented mean throughput (not the overall mean throughput) of all TCP connections which are controlled or could be controlled by the ensemble TCBI approach is higher than in the simulation scenario with a reliable last hop. But this at first astonishing result can be easily explained: The background traffic TCP connections with receivers in the receiver LAN are often affected by packet losses in the unreliable last hop. Due to the TCP congestion control algorithms these background traffic TCP connections reach a smaller overall sending rate, allocate less bandwidth, and produce a substantially lower load in the receiver LAN. The other TCP connections with receivers in the receiver LAN are also affected by packet losses in the unreliable last hop. But from a single TCP connection point-of-view the lower load in the receiver LAN can sometimes compensate for and even over-compensate for the in general negative influence of packet losses in an unreliable last hop on the mean throughput of a TCP connection. For example, most of the short TCP connections and some of the WWW TCP connections are not affected at all by packet losses during their whole lifetime. These TCP connections can highly benefit from the lower load in the receiver LAN and reach a higher connection-oriented mean throughput. Therefore, the observed higher connection-oriented mean throughput in the unreliable last hop scenario is only based on the lower load in the receiver LAN.

8. Conclusion and outlook

The simulation results show that the TCBI approach and in particular the FS-TCBI controller outperform standard TCP. New short TCP connections mostly benefit from the TCBI approach. But also new WWW TCP connections can have a remarkably higher throughput compared to standard TCP.

The two TCBI controllers discussed here use relatively simple algorithms, e.g. the new value for one particular TCP control variable in a TCBI connection set is not influenced by the current values of the other control variables in a TCBI connection set. Nevertheless, with these simple algorithms a remarkable gain of the mean throughput for some and a fairness gain for all TCP connections can be achieved.

In current and future investigations of the TCBI approach some more complex algorithms with a higher expected performance and fairness gain will be considered. The mid-term objective will be the combination of the TCBI approach with a transparent common congestion control of TCP connections belonging to the same TCBI connection set over their whole lifetime. Additionally, in terms of TCP-friendliness also the sending rate of UDP streams should be controlled with this approach.

References

[1] V. Paxson, M. Allman, W. Stevens, TCP congestion control, RFC 2581 (1999).

[2] S. Floyd, T. Henderson, The new reno modification to TCP's fast recovery algorithm, RFC 2582 (1999).

[3] J. Touch, T.C.P. Control, Block interdependence, RFC 2140 (1997).

[4] L. Eggert, J. Heidemann, J. Touch, Effects of ensemble TCP, ACM SIGCOMM Computer Communication Review 30 (1) (2000) 15–29.

[5] H. Balakrishnan, S. Seshan, The congestion manager, RFC 3124 (2001).

[6] S. Seshan, M. Stemm, R. Katz, SPAND: shared passive network performance discovery, Proceedings USITS'97 (1997).

[7] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, Hypertext transfer protocol—HTTP/1.1, RFC 2616 (1999).

[8] M. Savorić, The TCP control block interdependence in fixed networks—some performance results, Proceedings QOFIS 2001, LNCS 2156 (2001) 261–272.

[9] V. Paxson, End-to-end internet packet dynamics, IEEE/ACM Transactions on Networking 7 (3) (1999) 277–292.

[10] A. Reyes-Lecuona, E. González-Parada, E. Casilari, J.C. Casasola, A page-oriented WWW traffic model for wireless system simulations, Proceedings ITC 16 (1999) 1271–1280.

[11] R. Jain, Congestion control and traffic management in ATM networks: recent advances and a survey, Computer Networks and ISDN Systems 28 (13) (1996) 1723–1738.

[12] R. Jain, The Art of Computer Systems Performance Analysis, Wiley, New York, 1991.