Transport Information Collection Protocol with clustering of information sources

Mohamed Karim Sbai, Chadi Barakat

Transport Information Collection Protocol with clustering of information sources

Mohamed Karim SBAI
Projet Planete, INRIA Sophia Antipolis, France
National School of Computer Science (ENSI), Tunisia
Email: Mohamed_Karim.Sbai@sophia.inria.fr

Chadi BARAKAT
Projet Planete, INRIA Sophia Antipolis, France
Email: Chadi.Barakat@sophia.inria.fr

***Abstract*— We improve and validate TICP, a TCP-friendly reliable transport protocol to collect information from a large number of sources spread over the Internet [1]. A collector machine sends probes to information sources, which respond by sending back report packets containing their information. TICP adapts the probing rate in a way to avoid implosion at the collector and network congestion. To ensure smooth variation of the congestion control parameters and to probe sources behind the same bottleneck at the same time, we add to TICP a mechanism that clusters information sources. This mechanism is based upon the Global Network Positionning (GNP) Internet coordinate system. By running simulations in ns-2 over realistic network topologies, we prove that TICP with clustering of information sources has shorter collect session duration and causes less packet losses than the initial version that probes sources independently of their locations.**

I. Introduction

Nowadays, collecting information from a large number of network entities has more and more applications. The collected data can be availability of network entities, statistics on hosts and routers, quality of reception in a multicast session, numbering of population, votes, etc. In this work, we improve the Transport Information Collection Protocol (TICP) [1] which collects information entirely from a set of sources spread over the Internet. A collector machine sends probes to information sources, which send back report packets containing their information. However, some difficulties come into play:

- There is a risk of network congestion due to bandwidth limitation and the large number of sources. Furthermore, all sources are not behind the same bottleneck which makes the congestion control more difficult.
- The collection traffic can be aggressive towards traffic generated by other applications. In particular, it must not penalise concurrent TCP traffic.
- The loss of probes or reports lengthens the duration of the collect session, which urges for an efficient retransmission scheme.

TICP does not only adapt the probing rate as a function of network conditions, but also tries to minimize the collect session duration by deploying an efficient retransmission strategy. Moreover, it shares network resources fairly with concurrent traffic, namely TCP traffic by adapting its probing rate in a way similar to how TCP does.

The collector in the former version of TICP [1] probes information sources in a random order. We show in this paper that this choice causes many problems when moving into large networks like increasing collect session duration, causing high loss rates and generating a traffic out of control. To ensure a smooth variation of the congestion control parameters of TICP and to probe sources behind the same bottleneck at the same time, we add to TICP a mechanism to gather information sources into clusters. This mechanism is based on the modeling of the Internet by an euclidean space and its decomposition into clusters. We use for this the Global Network Positionning system (GNP) [4] which provides Internet host coordinates. The new mechanism makes it possible to traverse sources from the closest to the farthest from the collector in terms of RTT(Round Trip Time). This way sources located behind the same bottleneck are probed together which improves the efficiency of the congestion control. To evaluate the performances of the protocol thus obtained, we have run simulations with the NS-2 simulator [6]. These simulations have shown that TICP with the new mechanism of clustering has better performances. Indeed, there is a decrease in the loss rate and the collect session duration.

In the first section, we describe the main functionalities of TICP. We show in the second section that the former version of TICP has many problems and that we need to cluster sources. In the third section, we explain our approach of clustering. The fourth section discusses simulations results and the last one concludes the paper.

II. Transport Information Collection Protocol

TICP [1] is a reliable transport information collection protocol implementing diverse functionalities. We focus here on those related to error recovery and network congestion control.

A. Error recovery

The TICP collector has a list of all information sources that it probes to get reports containing information. Every source is distinguished by an identifier (eg. IP address). The sources whose reports are lost are probed again. To ensure this retransmission mechanism, the collect session is a succession of rounds. In the first round, the collector sends request packets to all sources following their ranking in the list it maintains. In the second round, the collector sends requests to sources whose reports were not received in the previous round. The collector continues in rounds until it receives all reports. This behavior in rounds is meant to wait for transitory network congestions to disappear from one round to another and to absorb the excessive delay that some reports may experience.

B. Congestion control

To control the rate of requests and reports across the network, TICP is based on a report-clocked window based congestion control similar to the TCP one [2]. The collector maintains one variable *cwnd* indicating the congestion window size in number of requests or reports. New requests are transmitted only when the number of expected reports *pipe* is less than *cwnd*. TICP adapts *cwnd* to the observed loss rate of reports. It proposes two algorithms to do so: Slow start and Congestion Avoidance.

1) Slow start: The collector starts a collect session by setting *cwnd* to *RS* (protocol parameter) and sending *RS* request packets. After some time, reports starts to arrive. Some of these reports come on time, others are delayed. A timely report indicates that the network is not congested and that the collector can continue increasing its congestion window: $cwnd = cwnd + 1$. A delayed report does not cause any change to *cwnd*. The window grows in this way until the network becomes congested. At this point, the collector divides its congestion window by two and enters the congestion avoidance phase. The protocol comes back to slow start mode whenever a severe congestion appears.

2) Congestion avoidance: The congestion avoidance phase represents the steady state of TICP. During this phase, the collector increases slowly *cwnd* in order to probe the network for more capacity. Upon each timely report, the congestion window is increased by the following amount: $cwnd = cwnd + \frac{RS}{cwnd}$. When congestion is detected, *cwnd* is divided by two and a new congestion avoidance phase is started.

C. Congestion detection mechanism

TICP uses the congestion detection mechanism to compute report loss rates and to decide whether a report is on time, delayed or lost. This mechanism is based upon a timer *TO* scheduled at the begining of the session and rescheduled again every time it expires.

1) Round-trip time estimator: TICP sets the timer of the mechanism to an estimate of RTT, using the samples of RTT seen so far. The value of the timer is computed using estimates of the average RTT and its variance. Let *srtt* and *rttvar* be the estimates of the average and mean deviation of RTT. Let *rtt* be the measured RTT when a report arrives. Inspired from TCP behavior, The collector updates the values of the estimates and the timer (*TO*) in the following way :

$$rttvar = \frac{3}{4}.rttvar + \frac{1}{4}.|srtt - rtt|$$
$$srtt = \frac{7}{8}.srtt + \frac{1}{8}.rtt$$
$$TO = srtt + 4.rttvar$$

2) Detecting network congestion: TICP computes the report loss rate during a time window equal to *TO*. When the timer is scheduled, the collector saves in the variable *torecv* the number of reports to be received before the expiration of the timer. Let *recv* be the number of timely reports received between the scheduling of the timer and its expiration. The collector considers then that $torecv - recv$ reports were lost in the network. It estimates the loss rate to $1 - \frac{recv}{torecv}$.

The network is considered as congested if the loss rate exceeds the Congestion Threshold (*CT*) and severely congested if it exceeds a higher threshold $SCT > CT$ (Severe Congestion Threshold). *CT* and *SCT* are two parameters of the protocol. TICP set them as follows:

$$CT = min(0.1, \frac{RS}{cwnd})$$
$$SCT = max(0.9, \frac{cwnd - RS}{cwnd})$$

3) Delayed and timely reports: A timely report is a report received before its deadline. The deadline of a report is given by the timer. A report not received before its deadline is assumed to be lost. If it arrives later than the deadline, it is considered to be delayed.

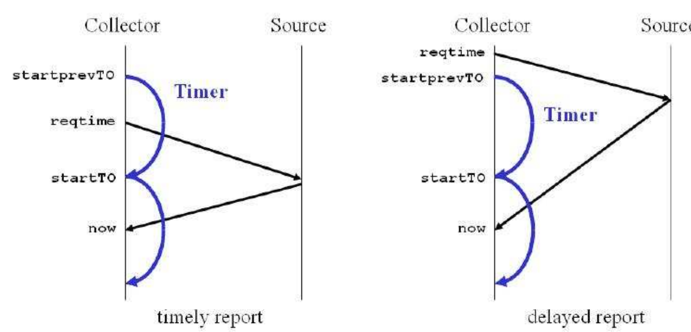

Fig. 1. The two types of reports

Figure 1 explains how the deadline of a report is set. Let *startTO* be the scheduling time of the timer. Let *startprevTO* be the previous scheduling time of the timer. When a report is received, the collector extracts from its header the timestamp *reqtime* indicating the time by which the corresponding probe has been sent. The report is received on time if and only if $startprevTO < reqtime$. The report is a delayed one in the opposite case.

III. Need for clustering of information sources

In this section, we present the drawbacks of the former version of TICP that motivated our present work. As we described earlier, the collector has a complete list of sources' identifiers. A collect session is a succession of rounds. In a given round, the collector begins by probing the source at the top of the list, then the following one and so on till the end of the list. The ranking of sources in this list has been so far done randomly and independently of any topology information. In reality, sources are more or less far from the collector. The random ordering results in variable non correlated RTTs during the collect session. Since the estimate of RTT at a given instant depends on its previously measured values, which in the case of random ordering are unrelated, this estimate seldom gives a good idea on the RTT of the next pair probe/report. This causes several problems. First, an overvaluation of RTT results in a delay in the detection of network congestion; the collector waits more than necessary for already lost reports. This delay means a waste of time and an aggravation of network congestion since the probing rate will not be reduced on time. On the other hand, an undervaluation of RTT can cause errors in the computation of report loss rate since the timer expires prematurely. Thus, some reports are declared lost while they are not. In this case, we reduce unnecessarily the size of the congestion window (*cwnd*) and hence, we increase the collect session duration.

Furthermore with random ranking of sources, packets generated can circulate everywhere in the network. At a given moment, this traffic can participate in the congestion of many bottlenecks. Since it is difficult to adapt congestion window size to network conditions on all paths from sources to collector, the Internet is considered by the original version of TICP as a single bottleneck. This version of TICP does not ensure fairness with concurrent traffic and its mechanism of congestion control is not efficient in case of large networks.

All the drawbacks described above are due to the random ordering by which information sources are probed. It is then important to cluster sources so that those close to each other are probed simultaneously. Also it is important to rank clusters from the nearest to the most distant of the collector so that to ensure that the network conditions vary smoothly and hence TICP congestion control can track them efficiently. The contribution of the current work is to add to TICP such a clustering and ranking mechanism together with its validation with extensive simulations.

A cluster is a group of sources located in the same neighborhood. Our idea is that the more sources are close to each other the more their reports meet the same network conditions on their paths to the collector and the more probable they are located behind the same bottleneck. In this case, the loss of reports indicates that the common bottleneck is congested, hence the collector can handle this congestion efficiently by decreasing the probing rate.

The collector probes clusters from the nearest to the farthest. This ensures a smooth variation of the congestion control parameters of TICP, for instance the rate of sending probes and the estimate of RTT. This again results in an efficient network congestion control.

IV. Clustering of information sources

In this section, we describe our approach to cluster information sources. For this, we use the Global Network Positionning (GNP) system to model the Internet by a 2-dimensional euclidean space [3]. A host is represented by a point in this space. The mathematical distance function gives an approximative value of the RTT between any 2 hosts. To ensure this, a small set of hosts called landmarks distributed across the Internet first compute their own coordinates in this geometric space. These coordinates are then disseminated to any ordinary host willing to compute its own coordinates relative to the coordinates of the landmarks [4].

The collector and information sources participate in GNP as ordinary hosts. At the end of the GNP operations, each source has a couple of coordinates $H(x_H, y_H)$ and the collector has also its own coordinates $C(x_C, y_c)$.

We define a cluster as being a set of information sources whose representing GNP points are located in a square area. The side of the square is denoted a, which is a parameter of the protocol. The central cluster is the square whose center is the point representing the collector $C(x_C, y_c)$.
A cluster is completely defined by a couple of coordinates (X,Y) being integer values. These coordinates are those of the center of the corresponding square relative to the collector coordinates and normalised by a. An information source whose coordinates equals to $H(x_H, y_H)$ belongs to the cluster (X,Y) given by:

- $X = round(\frac{x_H - x_C}{a})$
- $Y = round(\frac{y_H - y_C}{a})$

In order to probe information sources from the nearest to the farthest, the collector begins with the central cluster and then follows a spiral trajectory. Figure 2 gives an idea on this trajectory. One can with a simple algorithm find the coordinates of the next cluster during the collection knowing the coordinates of the current cluster.

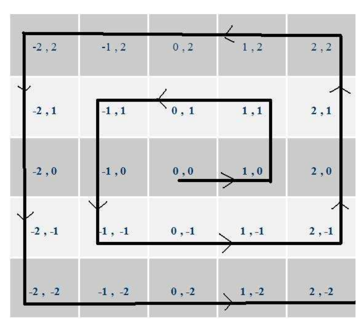

Fig. 2. Order of probing information sources

V. Simulation results

In this section, we discuss the results of our simulations. We have run these simulations in ns-2 [6] in order to evaluate the performance of TICP with and without clustering of information sources. That is why we have implemented GNP and TICP in ns-2.
We generate realistic network topologies for simulations using GT-ITM (Georgia Tech-Internet Topology Modeling) [7][8]. We choose to work on transit-stub (TS) topologies which

give the ability to model the complexity and the hierarchical structure of the real Internet. TS topologies model networks using a 2-level hierarchy of routing domains with transit domains interconnecting lower level stub domains. To these TS topologies, we assign latencies of 35ms for intra-transit domain links, 10 ms for stub-transit links and 5ms for intra-stub domain links. Figure 3 gives an example of a TS topology. Table I shows the parameters of the TS topologies used in ours simulations. In each simulation, we choose randomly 500 sources of information and a collector among the nodes that compose each TS topology. The parameters of TICP are set as in Table II.

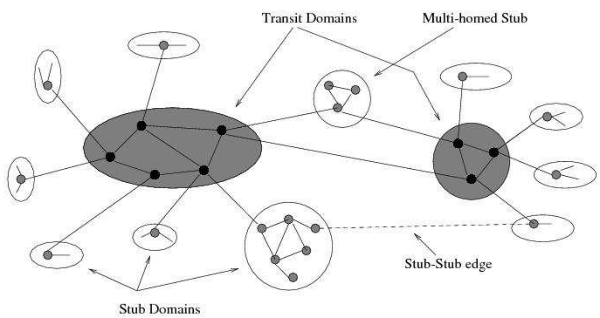

Fig. 3. transit-stub topologies

TABLE I
TRANSIT-STUB MODEL PARAMETERS

Parameter	Signification	Scenario
T	Number of transit domains	5
Nt	Average Number of nodes / transit domain	7
K	Number of stub domains / transit node	8
Ns	Average number of nodes / stub domain	7

TABLE II
TICP PARAMETERS

Parameter	Value
RS	10
probe size	100 b
report size	1500 b
a	50 ms

A. Network Congestion

We compare between the both versions of TICP with and without clustering of information sources. The comparison criterion is network congestion. Figure 4 illustrates the evolution of the congestion window size (*cwnd*) as a function of simulation time for TICP without clustering. We notice that at time 17s there was a reset of *cwnd* to *RS* following a severe network congestion (loss rate $> SCT$). TICP can not adapt the probing rate to the available bandwidth in several bottlenecks simultaneously. The network is seen by TICP as a single bottleneck. Figure 5 plots the same result but this time for TICP with clustering. In this case, TICP remains in the congestion avoidance phase and the severe congestion does not appear. This illustrates that TICP with clustering adapts the window size to the right network available resources. One can see that TICP with clustering treats bottlenecks one by one rather than at once.

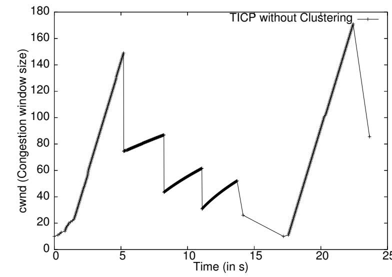

Fig. 4. Cwnd as a function of time for TICP without clustering

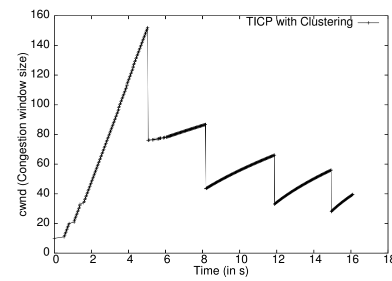

Fig. 5. Cwnd as a function of time for TICP with clustering

B. Collect session duration

We continue the comparison between the two versions of TICP. This time we concentrate on collect session duration. Figure 6 shows this duration for several simulations of TICP without clustering. In each simulation, the order of sources in the list of the collector is different ,that is why we obtain each time a different collect session duration. For TICP with clustering, the result is the same since the topology does not change. TICP with clustering finds the good order of information sources and has the shortest collect session duration. We save on average 30% of the collect session duration by moving from TICP without clustering to TICP with clustering.

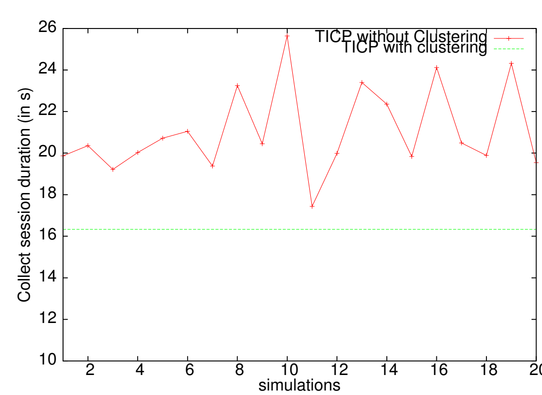

Fig. 6. Collect session duration for different ordering of sources

To evaluate the optimality of TICP more generally, we implement in ns-2 an information collection protocol having a constant congestion window size. For each window size, we run 10 simulations and we record the minimum of the collect session duration over them. Figure 7 presents the evolution of this duration as a function of *cwnd*. The curve has a parabolic shape: for small congestion window sizes, collect session duration is long because we have a low probing rate. For large window sizes, the network is congested which lengthens the collect session duration. The role of TICP is to find dichotomicly the good congestion window size that minimizes the collect session duration. We notice in Figure 7 how TICP with clustering manages to reach the optimum unlike TICP without clustering which yields longer durations.

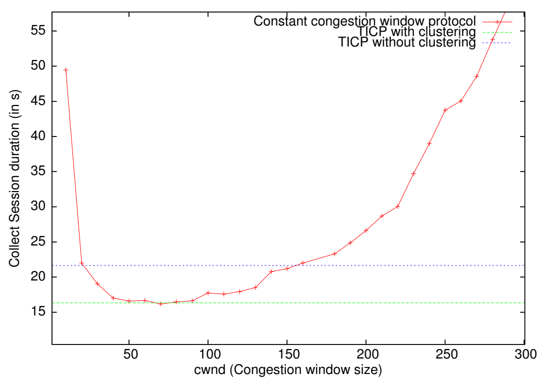

Fig. 7. Optimality of the protocol

C. Impact of cluster size

We vary the cluster size and we study its impact on collect session duration. Taking a very large a is equivalent to TICP without clustering since sources will be probed independently of their locations within the large cluster. Taking a very small results in clusters empty or with few number of sources which is not efficient since there will be no clustering of sources behind common bottlenecks. There should be some average a that provides the best performance. Figure 8 validates this intuition where we can see that in the network topologies we considered, a value of a around 50ms is optimal. Each point in the curve of Figure 8 is the average over 5 simulations run on different network realizations statisfying the characteristics in Table I. The number of sources is taken equal to 500.

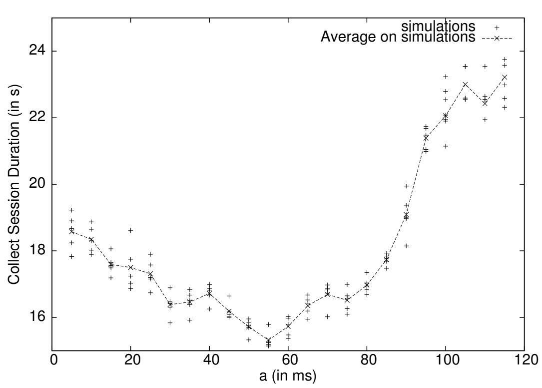

Fig. 8. Impact of a on collect session duration

Figure 9 studies how the number of sources impacts the choice of optimal a. We can clearly see that the optimal cluster size decreases when the number of sources increases. Compared to the value used above, the optimal a is equal to 85 ms for 300 sources and 45 ms for 700 sources . For small number of souces, one needs to increase a to group sources behind same bottlenecks together. At the opposite, for more sources one needs to decrease a so that the collector can better probe them depending on their locations. If we continue increasing the number of sources, the optimal a will stabilize and become equal to a minimum depending on the topology. One can safely use this values for applications collecting from a very large number of sources.

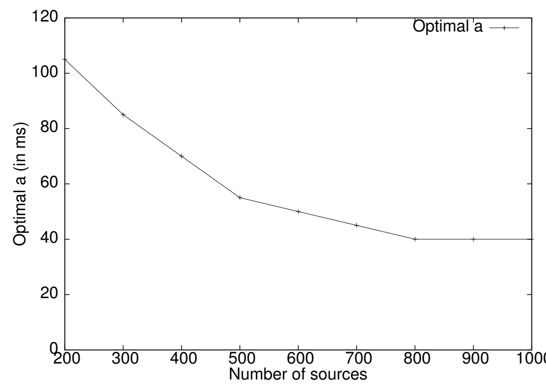

Fig. 9. Optimal a as a function of the number of sources

VI. Conclusion and Perspectives

TICP is a transport protocol to collect information from a large number of network entities. It aims to control the congestion of the network and to minimize collect session duration. To ensure a smooth variation of the congestion control parameters, we have added to TICP a mechanism to cluster information sources. The simulation results showed that this mechanism ameliorates the performances of TICP. In fact, it reduces loss rate and yields shorter collect session duration. However, the work on TICP is not yet achieved. Our current research focuses on the implementation of the protocol and on its extension to account for sources of large amounts of data.

References

[1] Chadi Barakat, Mohamed Malli, Noamichi Nonaka,"TICP:Transport Information Collection Protocol", in Annals of Telecommunications, vol.61, no. 1-2, pp. 167-192, January-February, 2006.

[2] M.Allman, V.Paxson, W.Stevens,"TCP Congestion Control", RFC 2581, April.1999.

[3] V.Paxson, M.Allman, "Computing TCP's Retransmission Timer", Internet Draft, April 2000.

[4] T.S Eugene Ng and Hui Zhang, "Predicting Internet Network Distance with Coordinates-Based Approaches", INFOCOM'02,New York,NY,June 2002.

[5] T. S. Eugene Ng and Hui Zhang, "Towards Global Network Positioning", Extended Abstract, ACM SIGCOMM Internet Measurement Workshop 2001, San Francisco, CA, November 2001.

[6] The Network Simulator ns-2, http://www.isi.edu/nsnam/ns/

[7] Ellen W.Zegura, Ken Calvert and S. Bhattacharjee,"How to Model an Internetwork". Proceedings of IEEE Infocom '96, San Francisco, CA.

[8] Ken Calvert, Matt Doar and Ellen W. Zegura, "Modeling Internet Topology, IEEE Communications Magazine, June 1997.