© 2015 IJEDR | NC3N 2015 | ISSN: 2321-9939

Minimizing the Congestion for Data Transferring Tcp Impact on Rits Protocol for Wireless Network

[1]Mr.S.Mohanarangan, [2] E.Anbumalarsakthe
[1]HOD, [2]II ME CSE
[1]Arunai college of Engineering
[1]Tiruvannamalai – 606 603

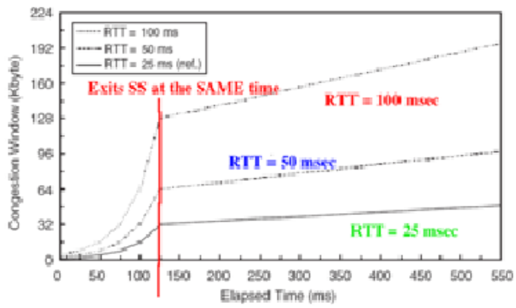
***Abstract* - TCP(Transport Control Protocol) has been widely used in the Internet and works well. It is the de-facto reliable protocol. TCP incast congestion happens in high bandwidth and low latency networks when multiple synchronized servers send data to a same receiver in parallel. Receiver side is a natural choice since it knows the throughput of all TCP connections and the available bandwidth. In the existing model, such as zigbee and content delivery network model , it just focus on the tree routing model, which has no routing table and CDN model only focus on distributed overlay of servers, It lacks in local stability. Due to the flexibility of the size of congestion window the rate of packet loss reduced but if the number of servers present is same and number of sender increases it leads to loss of packet loss because of the buffer flow. To avoid this situation in this paper we face the challenging issue of defining and implementing the RITS concept (that integrates post-facto time sync into a routing service).it is a network compensation strategy that makes RITS concept scales the network density and it is fully based on the time synchronization. It totally encounters the clock skew problem and rectify the causes for the skew problem and make the transfer of packets efficiently.**

***Keywords* - TCP, Incast congestion, RITS protocol**

---

1. INTRODUCTION

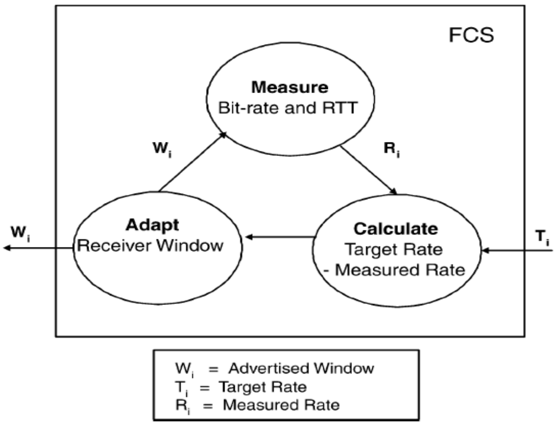
TCP does not work well for many-to-one traffic pattern on high-bandwidth, low latency networks, where congestion happens when many synchronized servers under a same Gigabit Ethernet switch send data to one receiver in parallel. It is responsible for transmission. Those connections are called as barrier synchronized .The final performance is determined by the slowest TCP connection that suffers timeout due to intense packet losses. The root cause of TCP incast collapse is that the highly bursty traffic condition of multiple TCP connections overflows the Ethernet switch buffer in a short period of time, causing intense packet losses. This paper focuses on avoiding packet losses before incast congestion, which is good than recovering after loss. Our idea is to perform incast congestion avoidance at receiver side by preventing incast congestion. Receiver side is a natural choice since it knows the throughput. Also the methods which avoids the congestion are slow start, fast retransmission, fast recovery servers which supports the parallel transmission of the packets in simultaneous manner.



Our idea is to perform incast congestion avoidance at receiver side by preventing incast congestion. TCP does not work well for many-to-one traffic pattern on high-bandwidth, low latency networks. Receiver side is a natural choice since it knows the throughput of all TCP connections and the available bandwidth. The receiver side can adjust the receive window size of each TCP connection, so the aggregate burstiness of all the synchronized senders are under control. There are also several congestion avoidance techniques: they are

1) Slow start
2) Fast retransmission
3) Fast recovery.

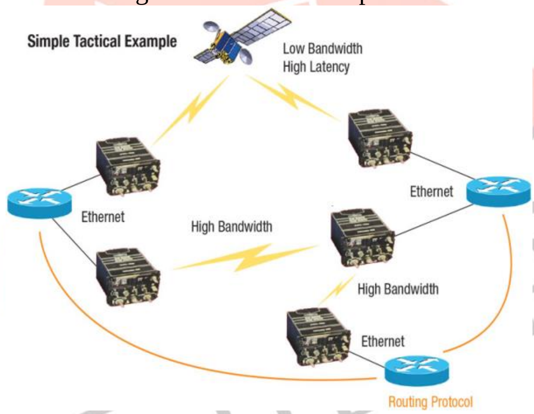
This paper addresses the above challenges by a systematically designed ICTCP. The incast can be avoided by analyzing why timeout occurs and also by reducing the idle links over the network. Reno is the concept included in the network to encounter the timeout experience when sender and receiver in the transmission mode.



The receive window is responsible for the efficient transmission of the packets throughout the network . so we are recruiting this method. The technical novelties of this work are as follows: 1) To perform congestion control at receiver side, we use the available bandwidth on the network interface. 2) per flow congestion control is performed independently in slotted time of RTT (Round Trip Time). 3) Our receive window adjustment is based on the ratio of difference of measured and expected throughput over expected one. This diagram depicts the receive window function with the help of calculating the bit-rate and the round trip time and yields the target rate and the measured rate of the work done.

2.BACKGROUND AND MOTIVATION

In distributed file systems, files are stored at multiple servers. TCP incast congestion occurs when multiple files are fetched from n number of servers. In this paper, we briefly introduce the TCP incast problem, then illustrate our observations for TCP characteristics on high-bandwidth which gives transfer of packets in the high rate(size) low-latency network (this depicts the situation where the packets lost its function when there is a idleness in the network .To avoid this situation we are heading towards the RITS approach. We seek for the RITS algorithm to avoid the packet loss and clock skew situation.



2.1ROUTING INTEGRATED TIME SYNCHRONIZATION

RITS is a reactive time sync(time synchronization) protocol, which can be used to obtain times of event detections at multiple observers in the local time of the sink node(s).It is the method of calculating the entire size of the network topology . RITS is an extension of the routing service with a well-defined interface having the remedy for the clock skew compensation. The interface defines commands to send and timestamp a data packet with a sender, a callback function is used here to notify signal the reception of a packet, and a command is used to query the timestamp (acknowledgement) of a received packet. Integrating the reactive time sync with the routing service has several benefits over a standalone procedure of the time sync service:

Coupling Of Event Data And Event Timestamps:

There is a tight logical coupling between event information and the corresponding timestamps. RITS retains this coupling: in a data packet, event data and timestamps are physically collocated. Coupling of nodes for the transmission. RITS thus implements implicit time sync, that is, the flow of time information is embedded in the flow of data flow of time information is embedded in the flow of data. Finally it encounters the events occurred.

Network-transparent event timestamps:

RITS protocol converts the corresponding time stamp hop by hop to the local time of the recipient node. As a result, all data packets received by a given node contain event timestamps in the recipient node's local time, independently from where in the network the events originated. By encountering this we can conclude the transmission is done successful or not.

Packet aggregation:

Packet aggregation helps decrease the number of message transmissions. In fact, not only does the number of radio messages decrease, but also the overall payload size. It is the method deployed to avoid the congestion by having the messages in an aggregate manner. It decrease the traffic by accumulating the messages and send it by the packets.By having this we can advertise that decrease of transmission will occur.

Packet filtering:

Filtering is the method deployed to filter the idle nodes and the packet. Through packet filtering support, it is possible to discard outdated messages at intermediate nodes enroute to the destination, thus decreasing the message load. Idle messages are discarded and get deleted.

Orthogonality to the routing policy:

DFRF allows for the customization of routing behavior via routing policies. RITS is orthogonal to the policies.

UNCERTAINTIES AND ERRORS IN TIME SYNCHRONIZATION

Time synchronization occurs or relies on message exchange on nodes .

The following factors are responsible for the time factor for the transmission .

They are

1. Send time
2. access time
3. receive time
4. probagation time
5. transmission time
6. encoding time
7. decoding time
8. interruption handling time
9. byte handling time.

Send Time :

This is the important time to send the packet from one node to the neighbor node . Receiver node sends the ack to the sender node. The time spent at the sender to construct the message.

Receive time :

This is the processing time for the receiver network to receive messages and the acknowledgement to the sender and also notifies the arrival of the messages.

Access time:

Each packet faces some delay at the MAC (Medium Access Control) layer before actual transmission.The time spent on the waiting or delayed time at the receiver channel.

Probagation time :

This is the time period spent in propagation of the messages between the network interfaces of the sender and the receiver. This is the probagation time.

Transmission time :

The time period taken to transmit the message from sender to the receiver channel. Here the speed ratio depends on the message between the two channels (sender and the receiver channel).

Interruption Handling Time :

When interrupts occurs there will be the delay in transmission time or delay in receiving time And also interrupts cause the delays in high rate.

Encoding time :

The time taken to encode the given message and send it without the interruption with the indication of the interruption.

Decoding time :

It decodes the message from the sender channel and converts into in the binary format by changing the byte can cause fluctuations and it introduce the jitter.

Byte alignment time :

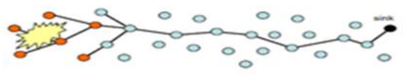
The delay incurred because of the different byte alignment of the sender and receiver. This time is deterministic and can be computed on the receiver side from the bit offset and the speed of the radio.

2.2 RITS and TDOA Measurements

In an important theory and an ass of monitoring the applications, sensor fusion works with time differences of arrival (TDOA) of events. Let us assume that the event E was detected at time uE by two nodes namely $r1$ and $r\_ 1$, and the two time tags arrived to the data fusion node along two different paths P and $P\_$, such that $P = r1, \ldots rn$ and $P\_ = r\_1 , \ldots , r\_m$. This step is taken to rectify the idle paths and to rectify the idle nodes .The variances of the skew related errors are proportional to the time the packet spend at the nodes. Sometimes it may cause the jitter condition. Another factor is that the event times need to be close to each other to recover the neighbor nodes. Otherwise, the clock skew (clock fluctuations or oscillation) of the sender node introduces a large error. To avoid the clock skew the timestamp is measured. The time event is a factor to encounter the transmission.The time stamp of messages and the events are measured. The clock rate is measured to transmit the packet in the slotted time. The offset, skew and the drift is responsible for the transfer of the packets efficiently.

Applicability of RITS:

only time differences of arrivals are required to localize an event, **fast** routing to the sink is required, because the event source is mobile.



Large scale network detects an event, the timestamps are then routed along a spanning tree to the sink node

2.3 SKEWCOMPENSATION

The main features of RITS were that it does not require any a priori information, does not need to know or maintain the skews of the nodes, and uses no additional time sync (time synchronization) messages to achieve synchronization. RITS converts the event timestamps from the local time of the sender node to that of the receiver node as the message is being passed from hop to hop. Without skew compensation, this conversion is achieved by adding the offset of the clocks of the sender and the receiver nodes to the event timestamp in the local time of the sender to yield the event timestamp in the local time of the receiver. When receiving a packet – which includes event description and event timestamp(send time and the receive time) in the sender's local time – the receiver node calculates the ***age of the packet*** in the sender's time: ***ages = ysm− ysE***, where ***ysm*** is the transmit timestamp of the message and ***ysE*** is the event timestamp, both in the sender's local time. RITS is hired to resolve the event issues.The time consumed when transferring the packet may be calculated and also it gives the time consumed by the acknowledgement. It not only yields the time but also calculates the ages of the packet. (ysm is the event timestamp of the message and yse gives the event timestamp).R denotes the receiver clock rate. It is important to measure the receive window for all the events done in the network topology.

2.4 DESIGN FACTORS FOR TIME SYNCHRONIZATION

Temperature:

It is one of the reason for the clock skews. Since sensor nodes are deployed in various places, the temperature variations throughout the day may cause the clock to speed up or slow down. For a typical sensor node, the clock drifts few *parts per million (ppm)* For low-end sensor nodes, the drifting may be even worse.

Phase noise:

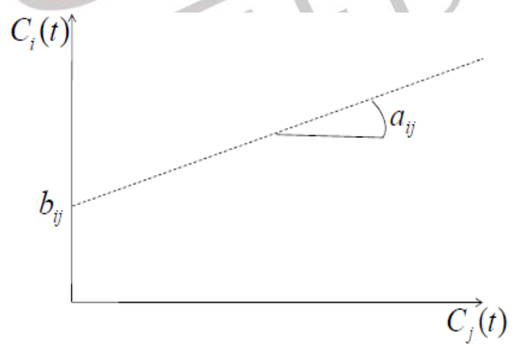
Some of the causes of phase noise are access fluctuations at the hardware interface, response variation of the operating system to interrupts, and jitter in the network delay. Jitter is an undesirable effect caused by the inherent tendencies of TCP/IP networks and components. The jitter in the network delay may be due to medium access and queueing delays.

Frequency noise:

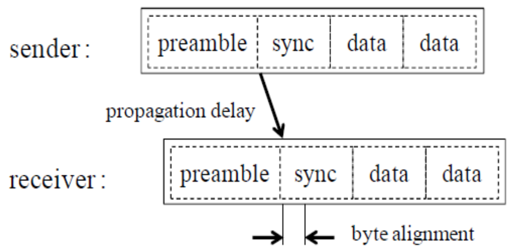
The frequency noise is due to the unstability of the clock crystal. A low-end crystal may experience large frequency fluctuation, because the frequency spectrum of the crystal has large sidebands on adjacent frequencies. This is one of the reason for causing clock skews.

Asymmetric delay:

As a result, an asymmetric delay may



cause an offset to the clock that cannot be detected by a variance type method. If the asymmetric delay is static, the time offset between any two nodes is also static.



This is the diagram depicting the delay of propagation in the alignment with the byte.

Clock glitches:

Sudden change in the clock that is clock fluctuation. Clock glitches are sudden jumps in time. This may be caused by hardware or software anomalies such as frequency and time steps.

3. DESIGN RATIONALE

Our goal is to improve TCP performance for incast congestion, instead of introducing a new transport layer protocol. Although we focus on TCP in data center network, we still require no new TCP option or modification to TCP header.TCP is responsible for the transmission of packets in the entire network topology. This is to keep backward compatibility, and to make our scheme general enough to handle the incast congestion in future high-bandwidth, low-latency network. To overcome all the inefficiencies we focus on the shortest path routing, fast retransmission, encountering the problem such as timeout, packet loss, idle links. To face these problems we just use the efficient method known as RITS. It not only encounters the timestamps but also encounters the ages of the packets to report the problem of idle packets by measuring the ages of the packet.

A. WIRELESS CHANNEL DESIGN

This module is developed to wireless network requirements, wireless equipments, Transmitter and receiver between one to another node by calculate the distance. Wireless sensor transmission ranges cover all nodes. It is more efficient than the wired network because wired environment may cause a packet drop and intense packet loss.

B. TOPOLOGY DESIGN

Topology design all node place particular distance. Without using any cables then fully wireless equipment based transmission and received packet data. Node and wireless between calculate sending and receiving packets. By having this design we can measure the distance between the nodes. By having this we can also calculates the time needed to transmit the packets.

C. NODE CREATING

Nodes are really created for the purpose of transmission of the packets from one node to the another node. To node create more than 10 nodes placed at particular distance. Wireless node placed in intermediate area. Each node knows its location relative to the sink. Nodes are responsible for the packet transmission and have the knowledge of neighbor nodes and the sink nodes.

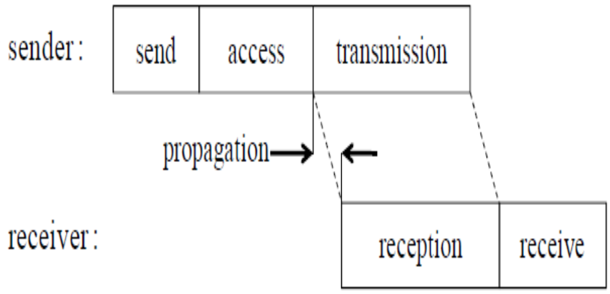
D. ATTACK DETECTION

Two sophisticated attacker models:

Probabilistic attack and variant response time delay, Attack is nothing but having malicious node in the network topology. To avoid the attack we use the signature method for the authorized transmission and also use the proxy method to compensate the congestion situation.

E. SYNCHRONIZATION OF MULTIPLE NODES

Sensor networks most often have a much more complicated topology than the simple examples and not all sensor nodes can converse with each other in directly. By having the multiple nodes in the network topology it can compensate the congestion by providing the transmission even there is the single node failure. Each and every node should know its neighbor node and the links between the nodes and the failure of the links.



F. ROUTING INTEGRATED TIME SYNCHRONIZATION PROTOCOL (RITS):

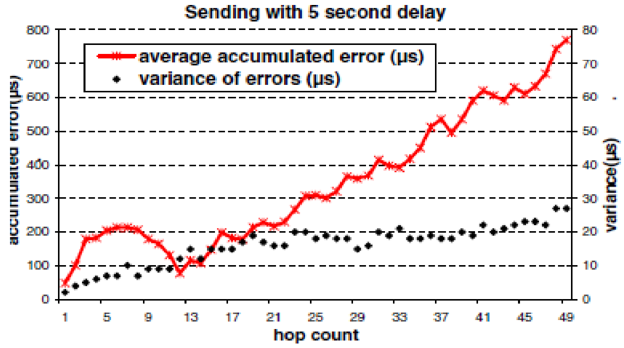
The Routing Integrated Time Synchronization protocol (RITS) provides post-facto synchronization. Detected events are time-stamped with the local time and reported to the sink. When such an event timestamp is forwarded towards the sink node, it is converted from the local time of the sender to the receiver's local time at each hop. A skew compensation strategy improves the accuracy of this approach in larger networks. RITS, as well as reactive techniques, is advanced to many proactive time sync protocols trade precision for power saving. RITS is the efficient method for estimating the timestamps. The message timestamp and the event timestamp are just estimated in this technique and it is the only method encountered the clock skew. It is the only technique which uses the timestamp for each and every hop to send and receive the packets.

4. ALGORITHM
SPR:
SHORTEST PATH ROUTING ALGORITHM:

Shortest Path Routing (SPR) algorithm route the messages over shortest paths in which the cost of links between nodes is defined by conditional intermeeting times rather than the conventional intermeeting times.

1. The every source nodes have the every node address and its distance. It delivers not only the nearest neighbor node but also location of each node in the network topology.
2. If source node hope to send data**:** To check the neighbors' node length is shortest means to transfer the data in those nodes.
3. The sensor node gathers the information about the nodes distance.(Distance of neighbor nodes).
4. Verify the node length, finally to find out the minimum path in the network. Length is an important factor.
5. Using the minimal path the data will be sending in efficient way as well as without any loss data.

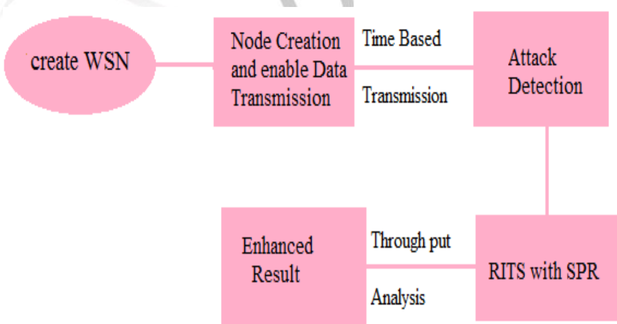
By using Shortest Path Routing **(SPR)** algorithm to decide which neighbors to keep track and which neighbors to discard. The clock speed is determined by the clock fluctuations. The clock speed agreement not achieved in networks with high neighborhood density. Network losses its connectivity due to beyond level of the packet capacity. Using Routing integrated time synchronization protocol with shortest path routing algorithm network use the SPR so each source to select individual path in network. So data transmission is fast and network performance will be increase. Data transfer will happen in an efficient Manner and in comfortable manner.



Packet loss is minimizing as well as data delivery ratio is increase. Also using the concept of the **Proxy** to authenticate the entering nodes and using the ***one way hash function to*** authenticate the nodes and use the ***node partioning*** concept for the betterment of the transmission. The one way hash function will discard the unauthorized nodes and sends the message with the signature and updates the recycle list

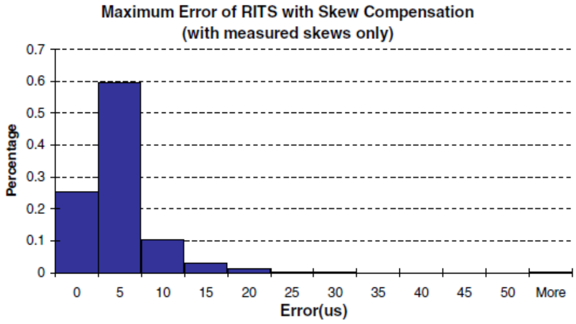
4. EXPERIMENTS AND RESULTS

Our network use the routing integrated time synchronization protocol with Shortest path routing algorithm so that each source can able to select individual path in network for data transmission. RITS can give the entire network topology size and weight. So data transmission is fast and automatically our network performance will be increase. Data Transfer in an efficient Manner. Packet loss is minimized as well as data delivery ratio is increase. at first the wireless topology is created with the nodes which is responsible for the transmission. The transmission is enabled only when the node is authenticated and ready for the transmission. The job done with the combination of RITS and the shortest path routing it creates the enhanced result of the transmission by having the method such as proxy(surrogate) , attack detection, time based, authentication etc. The transmission is enabled only when the node is authenticated and ready for the transmission.

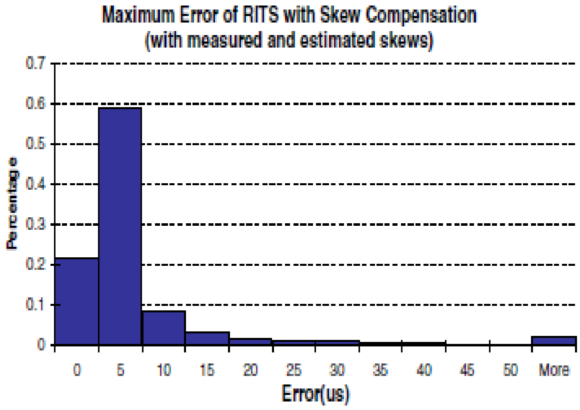


ARCHITECTURE OF THE PROPOSED METHOD

By using the shortest path routing we can conclude that the distance of the neighbor nodes. To be more secured while in transferring the nodes we are hiring the proxy concept. To check for the valid of the nodes it uses the signature method. Network



partitioning method is employed here to partition the nodes by that way we can avoid the traffic.



If the new node enters the network it is checked for the ID match with the proxy it is the valid node and if valid one it will send it to a recycle list. Network partitioning method is employed here to partition the nodes thus it will controls the congestion and gives the effective transmission.

5. CONCLUSION

To overcome the all the inefficiencies of the tcp incast we are facing the challenging issue of implementing the RITS concept .The RITS concept employed here is highly used here it reduce the traffic condition. This concept gives the time based transmission of the packets to the desired destination. It takes the responsibility of the transmission of the packets, also check for the clock skew by measuring the fluctuation of the time. Totally it regulates the size and the time synchronization of the sender and the receiver. This concept estimates the skew and shows the diagram of having send a packet with the skew and without the skew condition. Here to enhance the transmission the concept of the proxy is used.

REFERENCES

[1] M. Al-Fares, A. Loukissas, and A. Vahdat. A Scalable, Commodity Data Center Network Architecture. In *Proc. SIGCOMM*, 2008.
[2] M. Alizadeh, A. Greenberg, D. Maltz, J. Padhye, P. Patel,B. Prabhakar, S. Sengupta, and M. Sridharan. DCTCP: Efficient Packet Transport for the Commoditized Data Center. In *Proc. SIGCOMM*, 2010.
[3] L. Brakmo and L. Peterson. TCP Vegas: End to End Congestion Avoidance on a Global Internet. *JSAC*, 1995
[4] Y. Chen, R. Griffith, J. Liu, R. Katz, and A. Joseph. Understanding TCP Incast Throughput Collapse in Datacenter Networks. In *Proc. WREN*, 2009.
[5] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *OSDI'04*, 2004.
[6] T. Schmid, Z. Charbiwala, Z. Anagnostopoulou, M. B. Srivastava, and P. Dutta, “A case against routing-integrated time synchronization,” in Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, ser. SenSys ’10. *New York, NY, USA: ACM, 2010, pp. 267–280.*
[7] P. Sommer and R. Wattenhofer, “Gradient Clock Synchronization in Wireless Sensor Networks,” in 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), San Francisco, USA, April 2009.
[8] L. Schenato and F. Fiorentin, “Average timesynch: a consensus-based protocol for time synchronization in wireless sensor networks,” Automatica, vol. 47, no. 9, pp. 1878–1886, 2011.
[9] F. Sivrikaya and B. Yener, “Time synchronization in sensor networks: a survey,” Network, IEEE, vol. 18, no. 4, pp. 45–50, 2004.
[10] E. Krevat, V. Vasudevan, A. Phanishayee, D. Andersen, G. Ganger, G. Gibson, and S. Seshan. On Application-level Approaches to Avoiding TCP Throughput Collapse in Cluster-based Storage Systems.
[11] C. Lenzen, P. Sommer, and R. Wattenhofer, “Optimal Clock Synchronization in Networks,” in 7th ACM Conference on Embedded Networked Sensor Systems (SenSys), Berkeley, California, USA, November 2009.

[12] T. Schmid, Z. Charbiwala, Z. Anagnostopoulou, M. B. Srivastava, and P. Dutta, “A case against routing-integrated time synchronization,” in Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, ser. SenSys ’10. New York, NY, USA: ACM, 2010, pp. 267–280.
[13] P. Sommer and R. Wattenhofer, “Gradient Clock Synchronization in Wireless Sensor Networks,” in 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), San Francisco, USA, April 2009.
[14] L. Schenato and F. Fiorentin, “Average timesynch: a consensus-based protocol for time synchronization in wireless sensor networks,” Automatica, vol. 47, no. 9, pp. 1878–1886, 2011.
[15] F. Sivrikaya and B. Yener, “Time synchronization in sensor networks: a survey,” Network, IEEE, vol. 18, no. 4, pp. 45–50, 2004.