Multiwavelet based-approach to detect shared congestion in computer networks

Ass. Prof. Dr. Tarik Zeyad Ismaeel
University of Baghdad
Electrical Engineering Department

Eng. Ahmed A. Mahdi A. Kareem
University of Baghdad
Electrical Engineering Department

Abstract

Internet paths sharing the same congested link can be identified using several shared congestion detection techniques. The new detection technique which is proposed in this paper depends on the previous novel technique (delay correlation with wavelet denoising (DCW) with new denoising method called Discrete Multiwavelet Transform (DMWT) as signal denoising to separate between queuing delay caused by network congestion and delay caused by various other delay variations. The new detection technique provides faster convergence (3 to 5 seconds less than previous novel technique) while using fewer probe packets approximately half numbers than the previous novel technique, so it will reduce the overload on the network caused by probe packets. Thus, new detection technique will improve the overall performance of computer network.

Keywords: congestion, computer network, shared congestion, DMWT, DWT.

Introduction

Congestion in computer networks is becoming an important issue due to the increasing mismatch in link speeds caused by Intermixing between old and new technology of computer networks (CHIU, 1989). Flow control, congestion control and congestion avoidance are algorithms that have been addressed by several researchers in the past which are used to process congestion phenomena in computer network (Ramakrishnan, 1990). Generally speaking, there are two types of congestion that may occur in the computer network: first, *Independent congestion* may occur in the terminal of the network caused by one source which sends data through the links associated with it exceeds the capacity of these links or caused by two sources send data exceeds the capacity of individual link which is connect between them (Fall, 2009, Kim, 2003). The second type is *shared congestions* which occurs when two or more flows sharing the same link in the network and these flows send data exceeds the capacity of shared link (Balakrishnan, 1999, Rubenstein, 2000). All algorithms used to manage the performance of computer networks will first need algorithms to detect the level of congestion in these networks, thus detecting congestion became very important issue in the field of computer network. Better utilization of network resources is achievable with cooperation between flows. For example, the Congestion Manager examines all flows of the host where it resides, and groups flows passing through the same bottleneck link into a single flow aggregate. By performing congestion control over flow aggregates, rather than over each individual flow separately, the Congestion Manager can improve fairness and efficiency significantly (CHIU, 1989, Kim, 2003).

The basic primitive required for cooperative congestion control is to decide whether two flows are sharing a bottleneck link or not. Techniques for inferring shared congestion use two kinds of information from feedback: packet loss and delay. Techniques based on packet loss assume bursty packet loss (Rubenstein, 2000), thus they work well with drop-tail queues and lossy links but it slow and inaccurate with low loss rate or with other queueing disciplines, such as RED. Techniques based on delay (Rubenstein, 2000, Katabi, 2001) show more robust behavior in such an environment.

This paper will propose previous novel technique (delay correlation with wavelet denoising or DCW) with new denoising technique (discrete multiwavelet transform or DMWT) to detect shared congestion between two computer network paths. Like previous techniques, it is based on a simple observation: two paths sharing congested links (as shown in figure 1) have high correlation between their one-way delays. However, naive correlation measurements may be inaccurate, due to random fluctuation of queuing delay and mild congestion on non-shared links. In new proposed technique, these interfering delay variations are filtered out with *multiwavelet denoising*, which is signal processing method to separate signal caused by shared congestion delay from other delay signals (noise).

New detection technique is evaluated through extensive simulation by using the following programing languages: network simulation ver.2 (NS2) (Fall, 2010), AWK ver.2 (Close, 1995), MATLAB ver. R2008b. It takes at most 5 s to detect shared congestion with both drop-tail and RED queues, while previous techniques often takes longer time or fail.

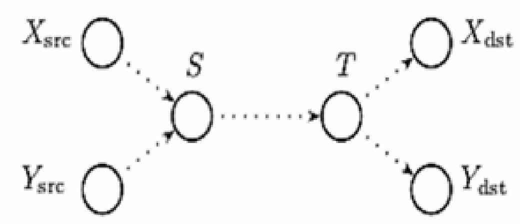


Fig. 1 Two paths sharing same link

Metrics Used To Detect Shared Congestion

There are two important metrics used with most techniques to detect shared congestion in computer network.

- **One-way delay time**:
  Time needed to send each packet from source to destination (Wen, 2008).
- **One-way Packet Loss:**
  Represents the number of packets lost from overall packets sent from source to destination (Wen, 2008).

Related Work

Rubenstein *et. al* (2000) (Rubenstein, 2000). Proposed two techniques to detect shared congestion in computer network, one based on one-way delays and the other is based on packet losses. The delay-based technique uses to generate a sequence of delay samples. An auto-measure Ma (auto correlation coefficients) is computed from pairs of adjacent packets of the first sequence. A cross measure Mx (cross correlation coefficients) is computed from a new delay sequence obtained by merging the two delay sequences. If Ma < Mx, it is mean that there is shared congestion. In their loss-based technique, Ma and Mx are conditional probabilities that a packet is lost when its following packet is lost.

Harfoush *et. al* (2001) (Harfoush, 2000). Harfoush technique explores the effects of concurrency on diagnosing network conditions. In this technique, a common source sends a packet pair back to back at 15 Hz. The probability that only the second packet is lost is computed from packet losses. If the probability exceeds the threshold (0.4), two paths are sharing a bottleneck.

Katabi *et. al* (2002) (Katabi, 2001). The approach of this technique relies on the observation that the correct clustering minimizes the entropy (Entropy: The concept of entropy is used as a measure of the uncertainty in a random variable) of the inter-packet spacing seen by the observer. This technique is highly accurate for detecting shared bottlenecks when the observer is strategically located, otherwise it becomes inaccurate.

Min S. Kim *et. al* (2003) (Kim, 2003). Proposed a novel technique (delay correlation with wavelet denoising or DCW) to detect shared congestion in Internet paths. Like previous techniques, it is based on a simple observation: two paths sharing congested links have high correlation between their one-way delays. If the correlation converges to 1, this means that there is shared congestion and other wise there is no shared congestion.

O. Younis and S. Fahmy (2005) (Fahmy, 2005). This technique is called Flowmate which uses the comparison test for delay based techniques from Rubenstein. At their development status they got the data from TCP **(**Transmission Control Protocol) flows only, since the majority of Internet traffic is TCP. The accuracy of bottleneck detection strongly depends on the lifetime of the TCP connections. The longer the lifetime of a connection the better the detection results are.

Simulation Procedure

Block diagram shown in figure (2) is represents the procedures of new shared congestion detection technique.

Delay Sampling Stage

Represents the first stage which will be used to get delay samples signal for path 1 and 2 for two types of queue (RED and DropTail). This stage is implemented on network topology shown in figure (3) which is created by NS2 program with the following parameters (Kim, 2003):

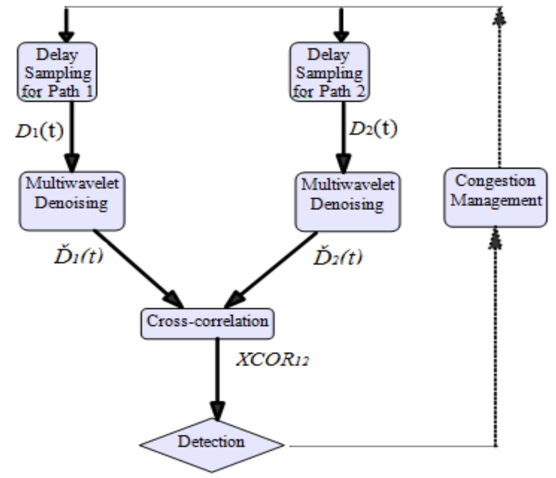
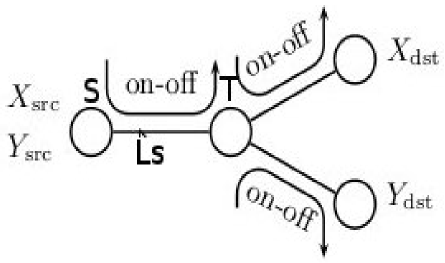


Fig. 2 Shared congestion detection



procedure

Fig. 3 Simple topology with a common source

(1) Each link has a bandwidth of 1.5 Mb/s. This value is chosen depending on real network measurements (Kim, 2003).

(2) The propagation delay of each link was chosen randomly between 20 and 30 ms for each simulation to make the behavior of simulation links similar to the behavior of real network links.

(3) Pareto ON-OFF constant-bit-rate (CBR) flows were used as background traffic, so that the congestion level could be controlled easily by changing the number of CBR flows. Background traffic will make some load on the simulation network links to make it near from the real network (Kim, 2003, Singh, 2004). The average ON and OFF states (ON-time is the time interval of sending packets by background traffic and OFF-time is the time interval of stopped (idle) sending packets) where selected uniformly between 0.2 and 3s. This step makes the behavior of simulation network near from the behavior of real network to increase the reliability of simulation results.

(4) The CBR rate was selected uniformly between 20 and 40 kb/s, and its pareto shape parameter was 1.2.

(5) Put 100 ON-OFF CBR flows on the shared link and 60 ON-OFF CBR on the independent links. With 60 flows, no congestion occurred and no packet loss was observed but the congestion will occur with 100 flows as well as the loss rate will be varied between 2% and 12%.

(6) The source node have two sources Xscr and Yscr which sends CBR packets through shared link to destination nodes Xdst and Ydst respectively at constant rate 10Hz (10 packets/S) in all simulations.

(7) The size of probe packet is 500 kB. (kB = kByte)

(8) The queueing size at each node is 50 packets.

(9) Background flows will start sending data at time 0 and after 1 second the sources Xscr and Yscr will start sending probe packets to ensure that the links are not empty.

(10) The time of each simulation starts from 0 to 105 s.

Dmwt Denoising Stage

The block diagram shown in figure (4) shows the procedure of denoising stage by DMWT.

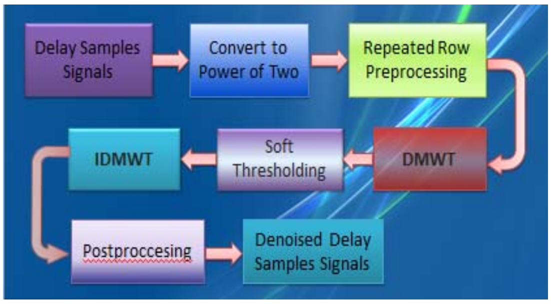


Fig. 4 Procedure of DMWT denoising stage

The important sub stages in the DMWT denoising stage is: DMWT, soft thresholding and IDMWT (inverse of DMWT). The other sub stages are assistant stages.

DMWT Operation

DMWT sub stage will analyze the input signal to two types of coefficients, the first one is approximation coefficients (represent the main information about original signal) which can be obtained by the following equation (Strela, 1998):

$$v_{\mathrm{j},k} = \sum_{m} G_{m-2\mathrm{k}} v_{\mathrm{j-1},m} \qquad (1)$$

Where

$v_{\mathrm{j},k}$ = approximation coefficients

$G_{m-2\mathrm{k}}$ = low pass filter of DMWT

$v_{\mathrm{j-1},m}$ = samples of input signal

The second one is detail coefficients (represent noise components) which can be obtained it by the following equation (Strela, 1998):

$$w_{\mathrm{j},k} = \sum_{m} H_{m-2\mathrm{k}} v_{\mathrm{j-1},m} \qquad (2)$$

Where

$w_{\mathrm{j},k}$ = detail coefficients

$H_{m-2\mathrm{k}}$ = high pass filter of DMWT

Soft Thresholding Operation

This sub stage is very important because it is used to remove noise from analyzed signal to get original signal. To get denoised signal, soft thresholding algorithm should be applied on the detail coefficients only for each decomposition level. Threshold value T is needed to apply thresholding algorithm. The idea of universal thresholding for scalar sequence of white noise wavelet coefficients with mean zero and variance σ^2 as in equation (3) (Strela, 1998).

$$T = \sigma \sqrt{2\log_{e} N} \qquad (3)$$

where N is the length of original input signal and σ represents the variance of input signal.

The universal threshold is attractively simple, but it is strictly suitable only when thresholding a gaussian scalar sequence of white noise wavelet (detail) coefficients of mean zero and variance σ^2. The repeated row preprocessing and the DMWT will change substantially the variances of the wavelet coefficients $w_{j,k}^{(0)}$ and $w_{j,k}^{(1)}$, thus 'average variances' have been defined ; for example, GHM filter banks with repeated row preprocessing the noise variance σ^2 is deflated by a factor c = 0.75 to $0.75\sigma^2$. Hence, for GHM with repeated row preprocessing, $\tilde{N} = 2\mathrm{N}$, T will be calculated by the following equation (Strela, 1998):

$$T = \sqrt{2 * 0.75\, \sigma_{\varepsilon}^{2} \log\left(2\mathrm{N}\right)} \qquad (4)$$

IDMWT Operation

This sub stage will use approximation coefficients and denoised detail coefficients to create denoised delay samples signal for path 1 and 2 by the following equation (Strela, 1998):

$$v_{j-1,k} = \sum_{m} \tilde{G}_{k-2\mathrm{m}}^{T} v_{j,m} + \sum_{m} \tilde{H}_{k-2\mathrm{m}}^{T} w_{j,m} \qquad (5)$$

Where

$v_{j-1,k}$ = approximation and denoised detail coefficients for j-1 of decomposition level. If j=1 then $v_{j-1,k}$ will represent denoised delay samples signal.

Cross-Correlation Stage

In signal processing, cross-correlation is a measure of similarity of two waveforms as a function of a time-lag (Wikipedia, 2010). The general shared congestion detection technique is based on the observation that measured delays of two paths showing strong correlation if the paths share one or more congested links, and little correlation if they don't share any congested links. Cross-correlation coefficients are calculated between delay samples signal for path 1 and 2 by the following equation (Rubenstein, 2000):

$$XCOR_{XY} = \frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2 \sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2}} \qquad (6)$$

Where

X_i and Y_i are represent one way delay signals for path X and path Y.

$\overline{X}$ and $\overline{Y}$ represent the average of one way delay signals for path X and path Y.

Detection Stage

This stage depends on the value of cross-correlation coefficients obtained from cross-correlation stage to decide whether shared congestion occurred or not in the tested path. If the values of cross-correlation coefficients converge to {1}, this leads that shared congestion occurred in the tested path of computer network, while if the cross-correlation coefficients converge to {0}, this leads that there is no shared congestion in the tested path of computer network.

Limitation

The important limitation for the new detection technique is *synchronization offset* which is defined as the time difference between arrivals of two probe packets at *S* (see figure 3), one sent by *Xsrc* at time *t* with *Xsrc*'s clock and the other by *Ysrc* at time *t* with *Ysrc*'s clock. As shown in figure (5), when synchronization offset increases, the delay sequences collected by the two nodes show less and less correlation. Figure (5) plots the cross-correlation coefficients without using any denoising signal techniques for two paths sharing a congested link as synchronization offset rises from 0 to 1s (Kim, 2003).

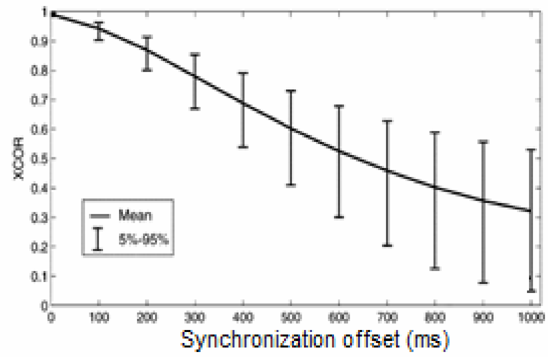


Fig. 5 Cross-correlation coefficient between two delay sequences versus synchronization

Simulation Results

After implementing the steps in delay sampling stage with RED queue technique and the source codes were co-located and their clocks were synchronized, the following signals will be obtained as shown figures (6) and (7) which represent the one way delay samples D1 and D2 of two sources X_{src} and Y_{src} sharing the same bottleneck L_s as shown in figure (3) for 100 seconds simulation. Each sample in the two delay signals represents the average of 500 simulations as explained in delay sampling stage to make the new detection technique is similar to the real network behavior.

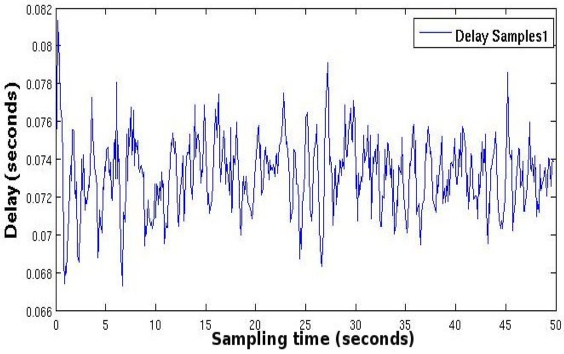


Fig. 6 Delay samples signal for path 1

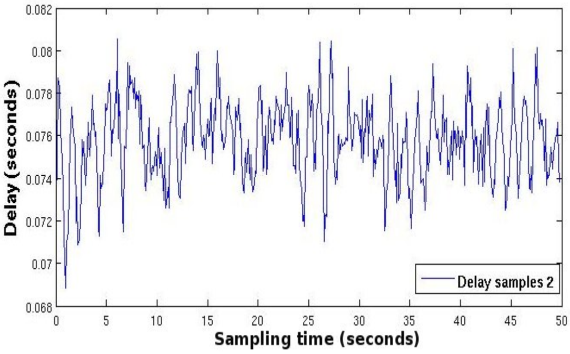


Fig. 7 Delay samples signal for path 2

To show if there is a shared congestion in the link L_s or not, these two signals D1 and D2 will be denoised by discrete multiwavelet denoising technique to get denoised signals $\tilde{D}1$ and $\tilde{D}2$ then calculating series of cross-correlation coefficients by using these two denoised signals. If cross-correlation coefficients converge to one, this indicates that there is a shared congestion and if cross-correlation coefficients converge to zero, this indicates that there is no shared congestion.

Figures (8) and (9) show the denoised of delay samples signals for path 1 and path 2 which are obtained from DWT and DMWT for three levels of decomposition. From these two figures, it can be shown that the power dropped from denoised signal by DMWT (star curve) is smaller than the one from signal denoised by DWT (dotted curve). This property is useful to keep fluctuations larger than noise in delay samples similar or near from fluctuations in original signal.

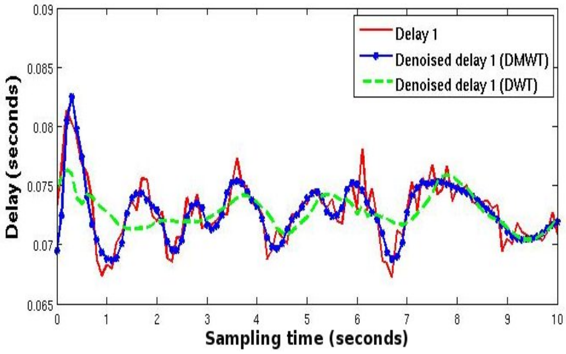


Fig. 8 Part of denoised delay 1 by DWT and DMWT

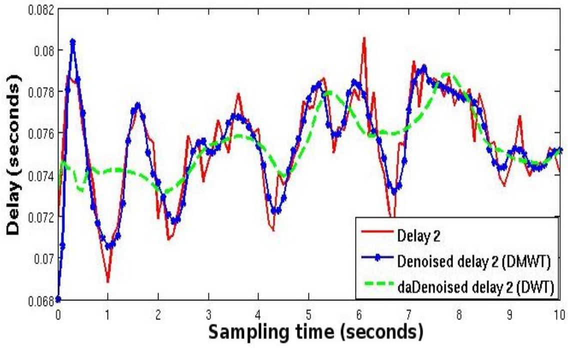


Fig. 9 Part of denoised delay 2 by DWT and DMWT

There is small difference between delay samples 1 and delay samples 2 caused by the behavior of RED queue technique which is used by network devices (router, switch, etc). DMWT will decrease the effect of this difference on the new detection technique by removing noise from these two delay signals and increase the similarity between them better than DWT does. Cross-correlation technique is used to calculate the similarity between the two collected signal, thus, DMWT will detect shared congestion faster than DWT because DMWT increases the similarity between the two collected signal and makes cross-correlation coefficients converge to one faster than coefficients obtained from DWT with small number of collected samples.

Figure (10) shows the mean of cross-correlation coefficients over 500 experiments with sending rate of 10Hz and three level of decomposition for delay samples which denoised by DWT and DMWT.

Figure (10) shows that the cross-correlation coefficients calculated from two delay samples which denoised by DMWT (continuous curve) is better and faster to converge to 1 than the cross-correlation coefficients calculated from same delay samples which is denoised by DWT (dotted curve).

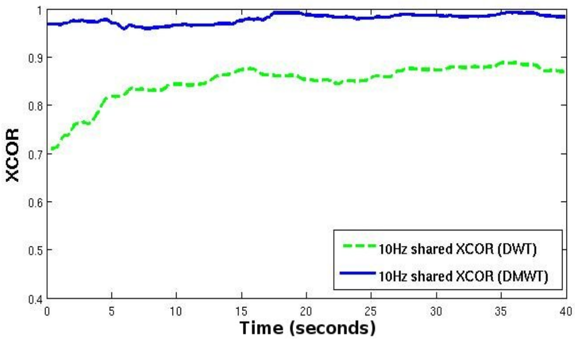


Fig. 10 Shared cross-correlation coefficients with RED queue

This means that the new detection technique is faster from previous novel technique (DCW) and need less than 7 seconds to detect shared congestion and this will improve overall network management.

Now we will test the new detection technique if there is independent congestion in the tested path.

Figures (11) and (12) represent mean of independent congestion delay samples of 500 simulations.

It can be shown that the two delay samples signals in figures (11) and (12) are not similar especially in large fluctuation as shared congestion because delay samples that flow through independent link 1 (from node T to Xdst node in figure (3)) will be exposed to situations differs from delay samples flow through independent link 2 (from node T to Ydst node in figure (3)).

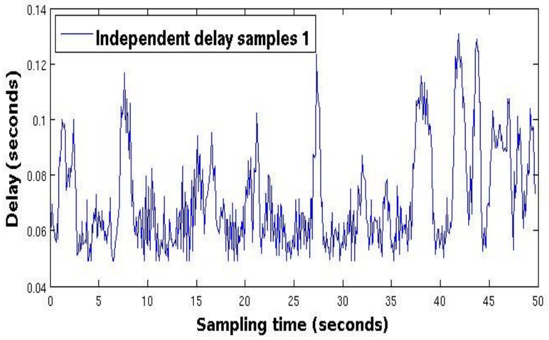


Fig. 11 Mean of independent congestion delay samples for path 1

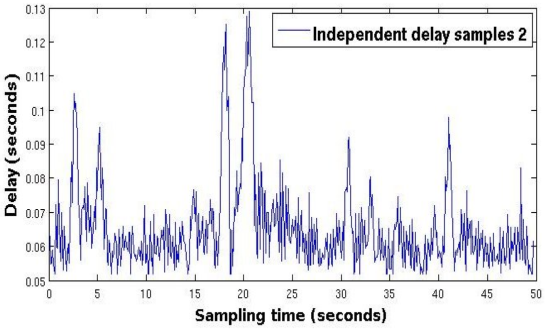


Fig. 12 Mean of independent congestion delay samples for path 2

Figure (13) shows cross-correlation coefficients between delay samples 1 and delay samples 2. Figure (13) shows that both cross-correlation coefficients obtained from denoising independent delay samples 1 and delay samples 2 by DWT and DMWT are reliable, similar and converge to zero after 3 seconds but cross-correlation coefficients obtained by DMWT are more stable from cross-correlation coefficients obtained by DWT.

The important factor which has a negative effect on the new detection technique and previous novel technique is synchronization offset which was explained in limitation section. When synchronization offset increases, the negative effect on both DWT and DMWT will be increased. Figure (14) shows the negative effects of increasing of synchronization offset on cross-correlation coefficients and make both new detection technique and previous novel technique less reliable.

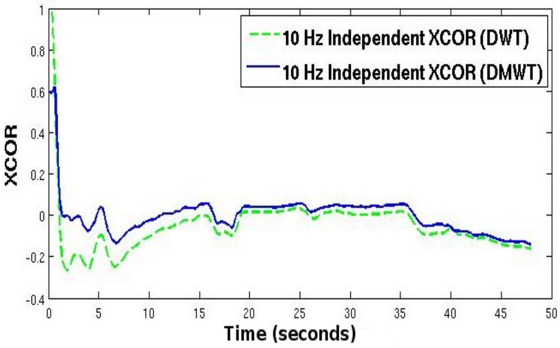


Figure 13 Independent cross-correlation coefficients

Figure (14) shows the effect of synchronization offset on the cross-correlation coefficients calculated from delay sequences 1 and 2 and how denoising techniques improves the algorithm of detecting a shared congestion in computer network by decreasing the effect of synchronization offset. Red dotted curve represents the mean of cross-correlation coefficients for non denoised delay samples which shows fast decays with the increase of synchronization offset and with 500 ms offset, the mean coefficient approaches to (0.5) which is the cut off values.

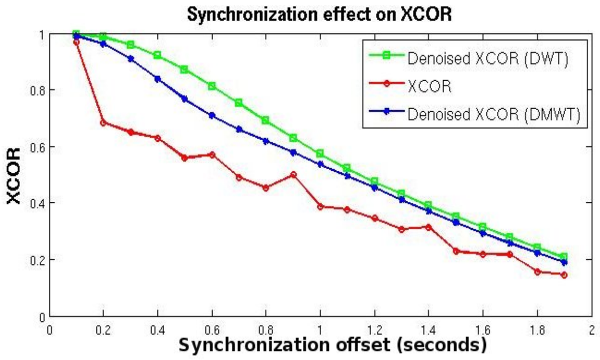


Figure 14 The effects of DWT and DMWT On the synchronization offset

It can be shown, with denoised cross-correlation coefficients by DWT and DMWT, the technique became more reliable and robust with 1 second of synchronization offset. So the maximum offset is roughly the maximum round-trip time (RTT) on the computer network. Measurement studies including one done by CAIDA (CAIDA, 1999) confirm that round-trip time is less than 1 second for the majority of paths on the computer network.

From all previous results, It can be shown that DMWT improves the new detection technique and make it faster from previous novel technique (DCW) to detect shared congestion in computer networks.

Conclusions

A new shared congestion detection technique based on the previous novel technique except DWT is replaced with DMWT which has more robust properties than wavelet transform in signal denoising and image compression. DMWT improve signal denoising technique in new detection technique and maks it faster (need less than 7 second) from previous novel technique (DCW) to detect shared congestion in computer network with approximately half number of probe packets which is needed with previous novel technique. Thus, new detection technique will decrease the effect of overload on the network caused by probe packets.

References

CAIDA, “The Cooperative Association for Internet Data Analysis”, Source: CAIDA, skitter-f.isc.org, available at http://www.caida.org/tools/measurement/skitter/RSSAC/ , 8 October 1999.

D. B. Close, et al., “The AWK Manual”, Free Software Foundation, Inc, Edition 1.0, December 1995.

D. CHIU, *et al.*, “Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks”, Digital equipment corporation, computer network and ISDN systems, Elsivier science publishers, 1989.

D. Katabi, et al., “A passive approach for detecting shared bottlenecks”, In Proceedings of the 10th IEEE International Conference on Computer Communications and Networks, Oct. 2001.

D. Rubenstein, et al., “Detecting Shared Congestion of Flows Via End-to-end Measurement ”, Working draft of version to appear in ACM SIGMETRICS’00, University of Massachusetts at Amherst, Department of Computer Science, June 2000.

H. Balakrishnan, et al., “An Integrated Congestion Management Architecture for Internet Hosts”, technical report, M.I.T. Laboratory for Computer Science and IBM T. J. Watson Research Center, 1999.

H. C. Wen, et al., “method for detecting congestion in internet traffic”, Technical report, United States Patent (San Jose, CA, US), 2008.

H. Tammerle, “Implementation and Evaluation of a Shared Bottleneck Detection System in Computer Networks ”, Master Thesis, University of Innsbruck, Institute of Computer Science , June 19, 2009.

K. Fall, et al., “The ns Manual”, The VINT Project A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, 2010.

K. Harfoush, et al., “Robust Identification of Shared Losses Using End-to-End Unicast Probes”, In Proceedings of the 8th IEEE International Conference on Network Protocols, Nov. 2000.

K. K. Ramakrishnan, et al., “A Binary Feedback Scheme for Congestion Avoidance in Computer Networks ”, ACM Transactions on Computer Systems, Vol. 8, No. 2, May 1990, Pages 158-181.

M. S. Kim, et al., “A Wavelet-Based Approach to Detect Shared Congestion”, Technical Report, University of Texas at Austin, Electrical and Computer Engineering, 2003.

M. Singh, P. Pradhan, P. Francis, “MPAT: Aggregate TCP Congestion Management as a Building Block for Internet QoS ”, International conference, Cornell University, IBM T.J. Watson Research Center, 2004.

O. Younis and S. Fahmy, “FlowMate: Scalable On-line Flow Clustering”, Networking, IEEE/ACM, University of Purdue, Department of Computer Science, 2005.

V. Strela, A. T. Walden, “Orthogonal and Biorthogonal Multiwavelets for Signal Denoising and Image Compression ”, Department of Mathematics, Dartmouth College, USA, 1998.

Wikipedia,“Cross-correlation”, Available at http://en.wikipedia.org/wiki/Cross-correlation#Properties,September 2010.