Neural-based TCP performance modelling

X.D.Xue and B.V.Ghita

Network Research Group, University of Plymouth, Plymouth, United Kingdom
e-mail: info@network-research-group.org

Abstract

Web users are expecting shorter response time when they are using Internet. However, Internet traffic is continued growth by adding more services and functions, the traffic will cause congestion problems and delays on Internet.

In this paper, we introduce the TCP protocol theory, the different between TCP long-lived and short-lived connection. And the neural network structure, algorithm and the parameters. Also described the current state of TCP short-lived connection traffic analysis and performance modelling. We explored and compared existing models for TCP long-lived and short-lived connection data transfer latency, the advantages and disadvantages are discussed.

This paper proposes to use mathematical model and neural network to predict TCP transfer latency for short-lived TCP connection for non-packet loss and packet loss situations, the results are compared by using the relative error and the overall comparison.

Keywords

Networks, TCP, Mathematical model, Neural network, Project, Paper

1. Introduction

Web users are expecting shorter response time when they are using Internet. However, Internet traffic is continued growth by adding more services and functions, the traffic will cause congestion problems and delays on Internet. Internet routers sometimes drop up to 5% of the incoming packets because of local buffer overflows (tcpipguide, 2004). For the TCP connections, it can be classified into two types: steady state and short-lived connections. This research will focus on the short-lived TCP connections by using steady-state flows extended mathematical model. On the other hand, in this project, the neural network simulation will be trained to estimate the short-lived TCP connections. The results will be compared with the mathematical model estimation results, and the conclusion will be made at the end of this research paper.

2. TCP, mathematical model and neural network

The Internet has become a more and more complex collection of network through its exponential increase in terms of users. With this increasing, there are also many

Internet applications, because such as the World Wide Web (WWW), usnet news, file transfer and remote login, have opted Transmission Control Protocol (TCP) as the transport mechanism. TCP is a very complex protocol, and the fast-changing network conditions make the development of an accurate TCP stochastic model to be a very challenging task. The TCP operates at the transport layer of the Open System Interconnection (OSI) network reference model. TCP connections drive the performance of Internet, because 90% of the Internet connections use TCP, 95% of the Internet traffic is carried by TCP (Garetto and Cigno, 1999).

2.1 TCP protocol

TCP is a full duplex protocol that each TCP connection supports a pair of byte streams, one flowing in each direction. TCP includes a flow-control mechanism for each of these byte streams that allow the receiver to limit how much data the sender can transmit. TCP also implements a congestion-control mechanism (Sinha & Ogielski, 1998). TCP also provides reliable data delivery. A key to provide reliability is that all transmissions in TCP are acknowledged. The recipient must tell the sender "yes, I got that" for each piece of data transferred. This is in stark contrast to typical messaging protocols where the sender never knows what happened to its transmission (Tcpipguide, 2004). This technique requires the TCP sender to assign unique sequence numbers to packets, also each packet sent is recorded until received the receiver sending back acknowledgements (ACKs) (Figure 1). On the sender side, the retransmission timer is also started whenever it sends a packet, if the packet is not arrival before the timer is expired, the sender assumes that the packet has been lost, and the TCP will arrange retransmission. The time between a sender sending data and receiving the acknowledgement is called the round-trip time (RTT) (Figure 1).

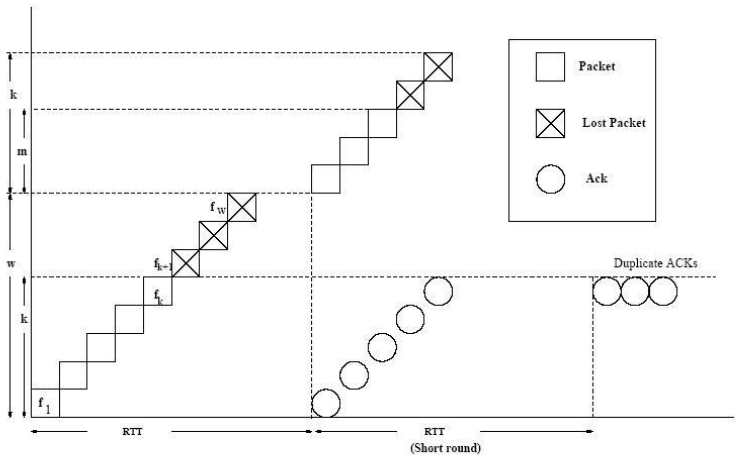

Figure 1: TCP connections (Padhye et al. 2000)

To utilise the network bandwidth effectively, TCP uses a sliding window flow control (Stevens, 1993) to send multiple segments at a time before it stops to wait for an ACK. TCP using flow control to regulate the sending and receiving rate, the sending rate depends on the receiver's processing capacity and buffer size.

Therefore, flow control is achieved by the receiver advertising a maximum window size (W_{max}) along with each ACK, thus limiting the transmission window size of the sender (Li, 2004).

This research aims to estimate TCP short-lived connections e.g. for Web transferring (smaller size date). However the mathematical modeling of long lived TCP connection is less suitable for the short lived TCP connections, because the whole date transferring process could be finished in the slow start period (Figure 1). In another word, viewing a short-lived TCP flows as an initial connection establishment handshake.

2.2 Cardwell-00 model

Neural network works on Cardwell presents the Cardwell-00 model in 2000, which is extended the steady-state results by accounting for the connection establishment phase and an approximate analysis of the initial slow start. This model added the observation that most of current TCP data transfers are short-lived and carry a small amount of data. There is a high probability for such flows, when they follow a path with low loss rate. To have a zero packet loss and, implicitly, to remain in slow-start for their duration. (Ghita, 2004). For the short-lived connection, in the case where the data transferring during the TCP short-lived connections with no packet losses (when the P value equal to 0), all data segment will be sent in the slow start phase. As the result, the model for these connections used only three inputs (Table1): the amount of data to transfer (Data), the estimated round trip delay (RTT) and the initial value of the congestion window (W1). The equation will be reduced to 1, which is the model for the time to send data segment in slow start. The data transfer latency E [T] is equal to below rather than the equation 2:

$$E\,[T] = E\,[Tss] + E\,[Tdelack] \qquad (1)$$

On the other hand, in the case where the data transferring during the TCP short-lived connections with losses (when the P is greater than 0) had several loss characteristics that may have been added to the inputs. As the result, the model for these connections used five inputs (Table 1): the amount of data to transfer (Data), the estimated round trip delay (RTT), the data loss rate (the fast retransmit loss rate), the timeout loss rate, the initial value of the congestion window (W1) and the first occurrence of loss also need to be considered. The equation of this case is the equation 2

$$E\,[T] = E\,[Tss] + E\,[Tloss] + E\,[Tca] + E\,[Tdelack] \qquad (2)$$

E [T]: the data transfer latency; E [T_{SS}]: the expected latency for the initial slow start phase; E [T_{loss}]: the expected cost for any RTOs or fast recovery that happens at the end of the initial slow start phase; E [T_{ca}]: the expected time to send the remaining data (the time spent in congestion avoidance) and E [T_{delack}]: the expected delay between the reception of a single segment and the delayed ACK for that segment.

2.3 Neural network

Neural network works on artificial neural networks, commonly referred to as ‘Neural network’, has been motivated right from its inception by the recognition that the human brain computes in an entirely different way form the conventional digital computer (Simon, 1998).

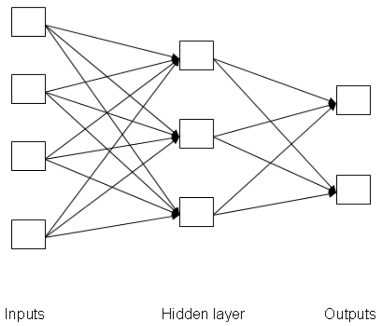

Figure 2: An example of a simple feed-forward network

Once a neural network is built to be of any use, the commonest type of artificial neural network consists of three layers (Figure 2): a layer of "**input**" units is connected to a layer of "**hidden**" units, which is connected to a layer of **"output**" units. Inputs and outputs correspond to sensory and motor nerves such as those coming from the eyes and leading to the hands. The hidden layer plays an internal role in the network. The input, hidden and output neurons need to be connected together.

This simple type of network is interesting, because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents. A neural network is the ability of the network to learn from its environment, and to improve its through learning. The network becomes more knowledgeable about its environment after each iteration of the learning process (Simon, 1999). This research will involve the neural network and mathematical model (Cardwell-00 Model), neural network (Matlab) can do different learning to estimate TCP connections and simulate the real network conditions also can be compared with the result of mathematical model, and the results will be more accurate.

2.4 Data pre-processing

The inputs in the equation 1, the first set of input for the mathematical model is including 1266 connection samples with non packet lost rate (P =0) and the inputs of mathematical model are Data, RTT, and Congestion window. The inputs in the equation 2, the first set of input for the mathematical model is including 5934 connection samples with packet lost rate (P > 0) and the inputs of mathematical model are Data, RTT, P, T0, and Congestion window.

3. TCP Short-lived connection estimation

The Cardwell-00 Model and the neural network as mentioned above will be tested in two stages below:

Stage 1 (Figure 3 and 4): Extract network parameters from TCP connections using TCP analysis software (e.g. tcptrace), those parameters are The amount of data to transfer (Data), the estimated round trip delay (RTT) and the initial value of the congestion window (W1) will be set as the first mathematical model inputs

The input parameters of the second mathematical model are the amount of data to transfer (Data), the estimated round trip delay (RTT), the data loss rate (the fast retransmit loss rate) (P), the timeout value (T0), the initial value of the congestion window (W1) and the first occurrence of loss also need to be considered.

Stage 2 (Figure 5 and 6): Input the resulting parameters, together with a performance estimate resulted from a mathematical model, into a neural network

Finally, the resulting performance estimate (i.e. throughput) for each connection, based on the trained neural network, will be compared with the current existing mathematical models and the actual performance value obtained for that particular connection in order to asses whether they are superior in terms of accuracy and robustness. Therefore, the comparison will be made in terms of relative error.

Factor	For equation 2	For equation 1
The inputs in the equation 1 and 2		
(RTT): Round-trip time	Between 1.13 and 5888.91 milliseconds	Between 1.13 and 23578.19milliseconds
(P): Data segment loss rate	The data segment loss rate is between 0 to 7%	N/A
(Data): Data segment size	Between 118 and 1460 bytes	Between 120and 1460 bytes
(T0): The average duration of the first timeout in a sequence of one or more successive timeouts	0 to 3588.177msec	N/A
(W): Congestion windows size	Between 240 and 64240 bytes	Between 240 and 59860 bytes
The mathematical result	Which is the input only for neural network estimation	
Target in neural network		
(Time): The data transfer time	Between 0.032478 and 103.4157 seconds	Between 0.016377 and 103.4141 seconds

Table 1: The inputs in the mathematical model and neural network

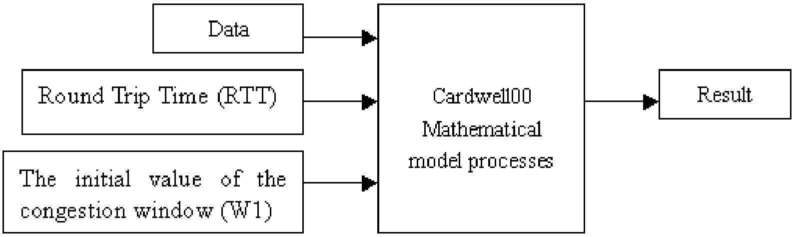

Figure 3: The mathematical model for TCP connections without packet loss

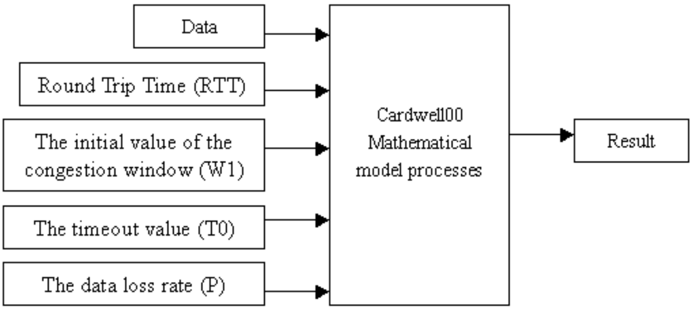

Figure 4: The mathematical model for connections with packet Loss

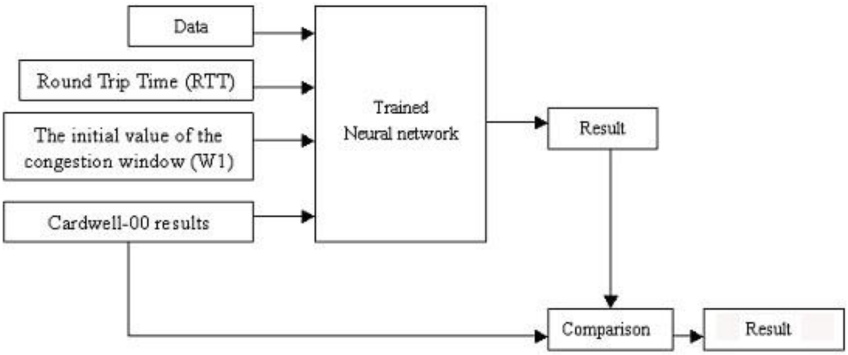

Figure 5: The neural network for TCP connections without packet loss

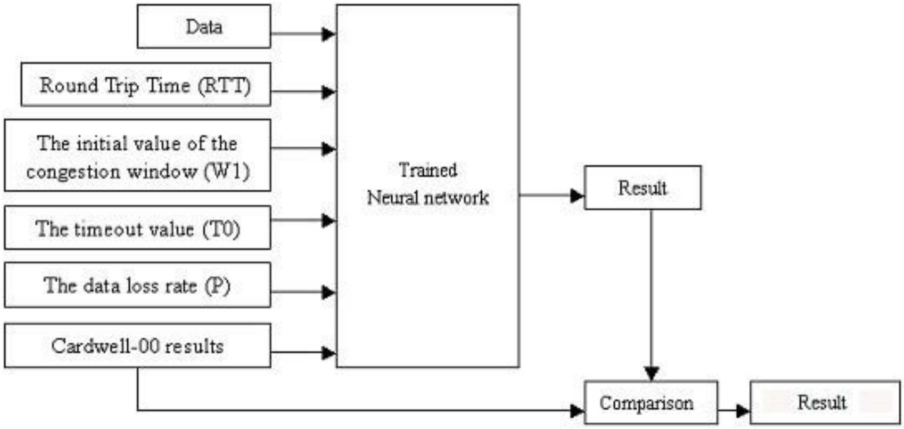

Figure 6: The neural network for TCP connections with packet loss

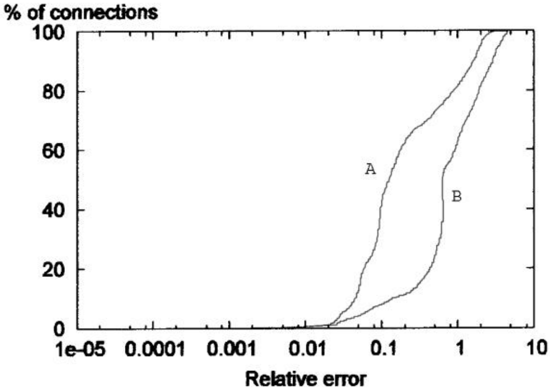

Figure 7: Cumulative distributions of the relative error resulting from using the mathematical model (B) and neural network (A) without packet loss

Figure 7 and Figure 8 compared the relative error produced by mathematical models and neural networks for TCP shot-lived with and without packet loss connections. As shown in the Figure 7 and 8 below that the neural network model outperformed the mathematical model. It shows that the relative error of connections estimated by neural network is around 5% - 20% less than the estimation of Cardwell-00 model. Although, the figure 8 shows that the accuracy of the neural network decreased, the neural network still performed better than the mathematical model. The average figures for mathematical model and neural network are presented in Table 2 and 3.

The next step is to compare the table shows that the average relative error, standard deviation of relative error and the correlation between the predicted values and the real values through the statistical results of the comparison, which are presented in Table 2 and 3 in order to estimate that how accurate is the mathematical model, the neural network model.

In Table 2 and 3 show that the neural network models perform on average better than the math model in both loss-free and loss estimation. It shows a 25% higher figure for the average relative error when compared with the results from 4-8-4-1 neural network for the connection without loss.

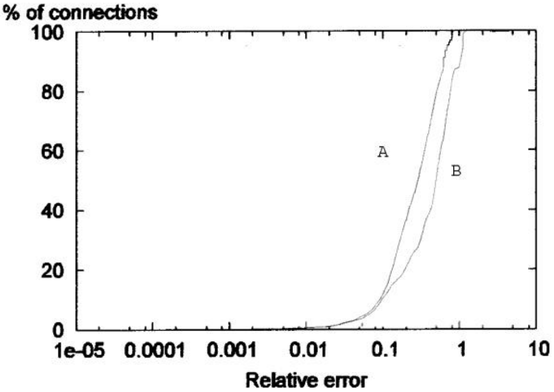

Figure 8: Cumulative distributions of the relative error resulting from using the mathematical model (B) and neural network (A) with packet loss

Model	Average relative error	Standard deviation of relative error	Correlation
Mathematical	0.596473	0.422372	0.732826
Neural network	0.352271	0.384017	0.783274

Table 2: Comparison of the resulting average figures for the short-lived TCP non-packet loss connections, using mathematical and neural network model

Model	Average relative error	Standard deviation of relative error	Correlation
Mathematical	0.557832	0.469931	0.448371
Neural network	0.530782	0.483257	0.498734

Table 3: Comparison of the resulting average figures for the short-lived TCP packet loss connections, using mathematical and neural network model

In Table 2, the neural network provided better accuracy than the mathematical model for the TCP short-lived connection without packet loss modelling results from mathematical model and neural network, and in the Table 3, the results from mathematical model and neural network model are very close; the different between the mathematical model and neural network model is smaller than the results in Table 2.

4. Conclusion and future work

We have presented our proposed neural network modelling TCP short-lived connections for different type of data sets (with and without packet loss). All the data sets are collected from real world TCP transfers, and compared with the measured results from mathematical model and neural network.

For the TCP short-lived connections, we only separately estimated the data set as non-packet loss and packet loss conditions, to estimate the packet loss rate bigger than 10% and more should be the next step for this research.

For the neural network modelling, it only may improve the modelling accuracy of the method in some of the aspects. In another words, the proposed neural based model provides a better alternative to mathematical models in terms of accuracy. Therefore, it is hard to identify the error in neural network inputs. To estimate the TCP short-lived connections in different type of neural network with different MSE parameters should be the next step for this research. And this could improve the accuracy of the TCP short-lived connections modelling.

5. References

Garetto, M. and Cigno, R. Lo (1999) "*Modeling Short Modeling Short-Lived TCP*", http://portal.acm.org/citation.cfm?id=987127, (Accessed 10-04-2005)

Ghita, B.V., (2004) "Performance characterisation of IP networks", *PhD Thesis*, University of Plymouth, UK

Li, Y.j. (2000) "Modeling Web/TCP Transfer Latency", *Master Thesis*, the University of Calgary, Canada

Padhye, J., Firoiu, V., Towsley, and D., Kurose, J. (2000) "*Modeling TCP Throughput*", http://www.cs.ucsd.edu/~savage/papers/Infocom2000tcp.pdf, (Accessed 02-02-2005)

Stevens, W. (1993) "*The Protocols (TCP/IP Illustrated, Volume 1)*", Addison-Wesley Professional; 1st edition

Simon, H., (1999) "*Neural networks : a comprehensive foundation*", Englewood Cliffs, N.J., Prentice Hall

Sinha and Ogielski., (1998) "*A TCP Tutorial*", www.ssfnet.org/Exchange/tcp, (Accessed 17-06-2005)

Tcpipguide., (2003) "*the TCP/IP Guide*", www.tcpipguide.com, (Accessed 07-06 2005)