**ITCP**

An intelligent TCP with neural networks based end-to-end congestion control for ad-hoc multi hop wireless mesh networks.

Mentored by: Dr. Shilpa

Darshana A. Naik

Submitted by: Antariksh Guha 1MS15CS022

Ashutosh Kumar 1MS15CS029

Ankush Kumar 1MS15CS020

Darshil Vala 1MS15CS038

Abstract

WMNs or Wireless Mesh Networks face quite a challenge in reliable data transfer over conventional TCP. The problem lies in the fact that conventional TCP have various faulty features like multi hop communication over lossy and non deterministic wireless mediums, absence of base station and same traffic patterns in neighboring mesh nodes.

Thus as the conventional TCP fail to address these congestion control problem, in this paper a more intelligent TCP congestion control problem has been proposed which include a Neural Network, also known as iTCP. With the use of this method the total throughput of the experimented or given network show impressive improvement and average energy consumption over the transmitted data also decreases.

Introduction

WMNs are quite relied upon for wireless network connections in city-wide connection, emergency response, community networks, urban sensornet and the like. As discussed in the abstract section, TCP over wireless networks come with its inherent problems like multi hop problem, hidden station problem, propagation loss, absorption loss, multi-path effect for being used in a wireless medium.

If two network topologies are considered, the first one as linear and the second one as a grid topology in both the cases TCP shows the same problem of packet drop, and the packet drop rate increases with the increase in the number of hops taken into account.

So as discussed, the main aim of this paper is to develop a solution to the above problem, keeping in mind that the original TCP is not changed to a great extent. The characteristics of the conventional TCP are kept in mind while designing the iTCP.

Motivation

As discussed earlier, the various drawbacks of the conventional TCP make it mandatory for the researchers to develop a more intelligent TCP. The hop mechanism is not the sole reason for this. The conventional congestion control methods like slow start and congestion avoidance are needed to be done away with, with the design of an end to end congestion control technique.

Consecutive transmission failures are taken into consideration for the experimentation purpose of the iTCP. Synchronization is a major drawback as WMNs have no base node to due to the absence of a central node it becomes quite difficult to control the traffic at the nodes. Similar transmission patterns are shown in all the nodes in a given neighborhood of nodes. But in real instances this is not the case as there are always delays during transmission among nodes and the synchronization problem disappears.

All these drawbacks and limitation, and especially the characteristics of the conventional TCP, led to the emergence of a need for a more intelligent TCP.

Specific problem solved

Well there are a series of advantages and disadvantages of the conventional TCP but the aim of this paper is not to solve the specific problems only but to do away with them completely by proposing a completely new technique. The main problem which have been solved can be mentioned as follows:

1. Terrible Network throughput of the TCP
2. The high rate of energy consumption per bit of transmitted data.
3. Low cwnd values of the TCP

Solution

With the help of throughput analysis it can be inferred that the throughput is greatly enhanced in iTCP compared to that of TCP Reno and other of the like. This is due to the fact that the type of learning we imply here is reinforced learning. Here at any given point of time only the currently available input is taken into consideration to give an output. Even though RTT is a major attribute in congestion control, it is neglected in iTCP because in a wireless transmission there needn’t be a large RTT to mean a congestion as packets of data are being lost all the time.

In iTCP we use a fine grained timer instead of a course grained timer to generate delay to mitigate congestion. Hence the overall time of the transmission is decreased.

In discussion with regard to energy consumption it can be said that the decrease of network throughput has direct hand in the decrease of the energy consumption. As the throughput increase, the total energy consumed by the network decrease due to the fact that the network remains operational of a lesser amount of time.

Well the congestion window is the one around which everything revolves in iTCP. If we see the simulation results for the same we observe that itcp has quite higher congestion window size compared to the conventional TCP. The neural network employed in this regard can be discussed in this regard.

The neural network consists of various nodes like hidden layer and the layer where the data is inputed and has to be decided whether it has to be considered or not for a specific increment or decrement or remaining unchanged for an attribute. In this case the attribute considered is congestion window.

For this type of set up one can decide that which attributed has to be considered to update another attributed depending upon the input from the earlier node. It can be set to 1.0 if a particular node matters in the decision making of a latter node or output. For calculating the next cwnd size the size of the current cwnd is taken directly. Hence if the output has to remain same for the cwnd the weight for the node from same to dscn is retained as 0.

Also for the next layer the dynamic weights of the previous layers are considered. Hence the weight sets associated with cwnd window are always considered as 1.0 and the sensitivity factor is taken in consideration to calculate the dynamic weights of the latter or next nodes for congestion window size.

Congestion control technique

The basic idea of congestion control herein is to control the cwnd size based on previous input as a part of reinforced learning. To attain this other attributes like :

1. Number of duplicate ACKs
2. Number of consecutive timeouts
3. Current congestion window size

Are the ones taken into consideration. The neural network is also cyclic in nature as it displays a cyclic nature of the input and output cwnd size from the layers. The neural network is of forward-feed type as an input of the present cwnd size gives output to the output cwnd size. There are three layers in itcp. One input, two hidden and one output layer. The first layer is the decision making layer wherein it is decided whether the cwnd has to be increased or decreased or remained same depending on the other two attributes, hence it determines the relative strength of the layers.

The next layer which is the next hidden layer determines the type of update based on the highest relative strength of the input from the previous layer. Since we always take initial cwnd size into consideration all weights flowing from cwnd size in the first layer is taken to be 1.0. Also if the cwnd size has to remain the same we take the weight from the same node to the chng node to be 0, as we wouldn’t want any change in the output node if it is decided in the first layer that the cwnd size will remain the same as before. Similarly, we want the output to take care of the highest relative strength of recommendation at the output we take the weight of the same node to the desc node to be 1.0.

Now, the main question arises that how is it decided whether the cwnd size has to be increased decreased or remain the same. It is calculated in the first layer where we calculate a weighted sum of the number of consecutive timeouts and duplicates ACKs. The part of the cwnd size is always constant as all the weights in its end are set to one. Hence the final result has to be decided based on the inputs from the other two named attributed dynamically. We also have a sensitively factor for the final output of the cwnd size which has three cases :

1. 1.5 timeouts and 3.0 duplicate giving strength of recommendation as decreasing the cwnd size.
2. 2.0 timeouts and 4.0 duplicate giving strength of recommendation as increasing the cwnd size.
3. 3.0 timeouts and 6.0 duplicate giving strength of recommendation as the cwnd size to remain the same.

The sensitivity factor is given by current cwnd size by alpha and alpha being the sensitivity granularity.

Selecting the Activation Function

The prospect of getting ideal weights for all the nodes depends on the fact of adaptation of the best possible activation function again.

1. For the input layer nodes the function is a positive linear function

F(x) = max(x,0)

1. For the inner hidden layer and when we want the congestion window to be decreasing we would want a function which gives decreasing values for the increasing in the number of timeouts and the increasing number of the number of ACKs. Hence, if there are too many timeouts and duplicates, which satisfies the conditions of a congestion, we would get a stronger recommendation for decreasing the cwnd size.

Also, threshold values are important and we set the lower value of the decreasing function to be 0.This is to control the extent of decrease.

Fdecr (x)=min(-x,fd-max) for all x<0

Fdecr (x)=0 for all x>=0

1. For the inner hidden layer and when we want the congestion window to be increasing we would want a logarithmic function with base b =2 and fd-max to be set as the upper limit for increase of the cwnd size.

Fincr=min(1+log(x),fi-max where x>=1

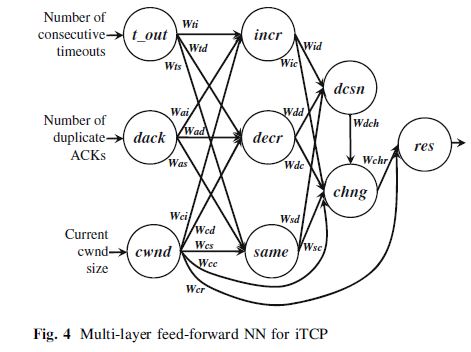
Fincr= 1 where x<1

1. For the cwnd size to remain the same we need a function which is symmetric and the f max to be zero to make it more consistent.

Fsame=fs-maxxB-|x| for all x

All these inputs go into the desc node of the second last hidden layer and that node decides whether there is a required update on the cwnd size.

On the decision making of the cwnd size all the attributes are taken into consideration such as the weights and the outputs of the previous layer too.



Performance Evaluation

The performance of the iTCP is tested on two network topologies:

1.Linear

2.Grid

The complete evaluation of the given paper is done on two topics and those are :

1. Total network throughput
2. Average energy consumed per transmitted bit

The performance evaluation of the iTCP is done with the help in ns-2. Destination Sequence distance vector is enabled as the routing protocol. The various network parameter, especially the threshold values are set first before the initiation of the experimentation.

The development of an efficient and good congestion control does not only depend on the type of congestion control we give in the network but also the flow control in the media and it also largely depends on the network topology taken into consideration. In some network topologies it may be observed that after a certain amount of flow the throughput and the average energy consumed per bit decreases, whereas in others it may increase. But no matter what the application of NN in congestion flow improvement the performance of the network in terms of network throughput and average energy consumed, in comparison to that of the other conventional TCP congestion controls.

When we run the NN set up in a grid topology and we take the cwnd size as the primal attribute. The other attributes like throughput and average energy consumed depends upon this only. Also the receiver window size is a negative factor in this whole set up as the receiver window size is fixed at 20 and it never changes.

Environmental Impact:

The network congestion can be further discussed as:

1. In case of very low number Nt, the conventional TCP fail as they are unable to differentiate between a congestion or a normal drop in the amount of incoming traffic. They display a small number of traffic as a congestion and trim the cwnd window. But iTCP is more intelligent and so they do congestion control only when it is a real congestion.
2. iTCP is also better to detect congestion in case of the hidden station problem, in case of higher flows in the network traffic.
3. The increase in coverage area and total number of nodes help iTCP to do a better job instead of losing efficiency as in the case of the other conventional TCPs.
4. The increase in the flow of the traffic and the number of nodes reversely affect the congestion control of the traffic, as it should, but still the performance on iTCP is better that most other TCP.

Bandwidth Ratio:

The variation of the result in case of varying bandwidth is same as that of the variation as discussed in the environmental impacts section discussed above. Faster accumulation of data or congestion occur in case higher bandwidths.

Large-scale dense deployment:

The large scale deployment of the iTCP have different impact on the congestion control of the instances when the network has more or less number of nodes, considering the flow of the network is constant for both cases.

The normal TCP as unable to differentiate between a congestion and a proper sparse node set up. They treat both as a case of congestion and thus waste resourses when the network and more cwnd window could have been applied but it is ignored. The iTCP helps in this regard and it also help in regard to the condition of bottleneck in TCP connections. Here application limited data transfer is applied at the application layer to overcome this problem.

Congestion window analysis:

The cwnd size in iTCP is quite large compared to the cwnd size in the other forms of TCP. Where large values of cwnd size are favorable in case of large data transfer.

Throughput Analysis:

The throughput of the conventional TCP are quite less compared to iTCP as the cwnd size helps in more passage of data.

Fairness Analysis:

The analysis of fairness of a network is the same as that of network throughput analysis. iTCP attains better fairness over the other network TCP. Fairness Index represents the normalized sum of the difference in throughput over all the possible flows of networks.

Energy consumption analysis:

The iTCP consumes lower energy per unit bit of the transmitted data. The lower consumption is in accordance to the higher network throughput.

End to end analysis:

In case of end to end delay in the network we see that TCP Vegas has the upper hand over iTCP. But it doesn’t matter as the data transfer need to be reliable and not time management specific. It doesn’t matter whether the data reaches in the given time or it reaches late in most of the cases.

Network condition analysis:

The network quality and network strength are one of the most important factors when it comes to network analysis. Both the factors play a major role in the experimentation of our assumptions.

System overhead analysis:

When we analyze the memory buffer and the cache memory we see that in case of iTCP there is significant less usage of these two structures. The other TCP structures use much more memory than iTCP.

Related Work:

There has been many suggestions regarding the congestion control of TCP:

1. TCP Tahoe uses slow start and congestion avoidance where the congestion window is cut down 1 during the state of presence of timeout.[59]
2. TCP Reno use Fast Recovery and Fast Retransmit with the help of ACKs. [100]
3. TCP Newreno improves TCP Reno by maintaining Fast Recovery. [34]
4. TCP Sack improves TCP Newreno by implementing selective acknowledgement. [35]
5. TCP Fack improves TCP Sack by implementing forward acknowledgement. [71]
6. TCP Vegas improves TCP Reno by proactive measures of congestion control and comparison between estimates and real overheads. [10]
7. Cubic TCP and Compound TCP are used in wired networks and they are better in terms of throughput in the wireless version of the TCP Cubic and TCP Compound. [42]
8. Snoop was the first attempt to improve TCP but it involved the change of the fact that TCP did not have any normal base station. [98]
9. TCP Westwood gave the idea of segment size which was not feasible with TCP. [1]
10. Several congestion control techniques suggest the presence of static nodes in place of mobile nodes and it becomes quite difficult if the nodes the dynamic or mobile in case of congestion control. [61]
11. Some variations had been proposed in this respect to minimize this effect like minimizing this effect at intermediate nodes and priority-based MACs.\
12. The first attempt to improve the TCP was to propose an approach with explicit notifications from the routing layer to optimize the network congestion. [109]
13. TCP-AP proposes incorporate adaptive pacing based on 4-hop propagationfor controlling congestion in hop transmission. [67]
14. But in case of TCP-AP there is continuous query in the other network layer which is not quite feasible always., and hence TCP-AP is not always feasible.
15. There are also other proposed network congestion techniques such as NRED and FairCast. These types are not always feasible though as they require information from the neighboring nodes which is not always possible.

Table based on iTCP:

|  |  |
| --- | --- |
| Control Pattern | End to End |
| Node Distribution | Random and Uniform |
| Evaluated by | Ns-2 |
| Compared with model | TCP Reno, Westwood, Vegas |
| Key idea | Designed to overcome two mail things:   1. Bad throughput of the network 2. More energy consumption rate per transmitted data bit |
| Key features | * Network traffic congestion control * Capable to control the various challenges of the WMNs:  1. The throughput problem in WMNs. 2. The high energy consumption problem per unit transmitted bit in the network. |
| Congestion Notification Method | * Explicit notifications about congestion from routing layer. |
| Congestion control mechanism | * Network traffic congestion control * The iTCP is implemented with the help of an intelligent neural network. * Reinforced learning is implemented in iTCP. * The next value of a node is decided based on the previous input to the node. * The input parameters in the process are number of duplicate ACKs, number of consecutive timeouts and the congestion window size. * The proposed congestion control is forward-feed and has zero bias and has multi layers in the neural network. * There is a cyclic relationship between the layers of the network between the current and the next output of the congestion window size. * Fixed and dynamic wrights of the nodes play a major role the prediction of the next cwnd size. * Choosing the activation function is one of the most important task for optimal weight selection. |
| Advantages | * The iTCP has the ability to distinguish between a case of congestion and fewer number of data in the network, as its more intelligent than the conventional TCPs. * iTCP handles traffic with network hidden station problem better. * The implementation of iTCP requires lesser energy per unit transmitted bit in the media. * It can work on a larger congestion area than the other congestion control, which is a quite advantageous thing. * The implementation doesn’t require very radical changes or we can say that it requires no changes at all in the present structure of the TCP. * The high convergence problem is a boon in disguise as it can be said that the synchronization problem is solved in this regard. * Resource constrainting is not there in WMNs. * Applicable in mobile nodes in network which in turn gives mobility. * Synchronization problem arises in ideal cases but in practical cases it is solved automatically because of time delay between the nodes. |
| Comparison | * The comparison of the iTCP is done with TCP Reno, TCP Westwood and TCP Vegas. * iTCP does better evaluation of the network in case of the hidden network problem. * It works better in case the area coverage or the number of nodes is increased. * In increase of the flow of traffic the performance is negatively impacted but its still better than the other TCPs. * The cwnd size is quite big in case of the iTCP rather than the others. * The throughput in case of iTCP is lesser than that of the others. * iTCP is fairer over the other networks. * It consumes lesser energy compared to the other TCPs available. * TCP Vegas has upper hand in case of end to end time analysis but it doesn’t matter as reliable data transfer doesn’t depend on the time constraint as much it depends on whether or not the data reaches the other end. * The memory buffer and cache memory used in case of iTCP is very less as compared to the other means of TCPs. |
| Limitations | * The main two issues of the iTCP are:  1. Training of the neural network: The training set is not fixed and hence training a model is quite a challenge. 2. The quick convergence wherein the cwnd size quickly increase or decrease is a limitation. The congestion window attain their highest value very quickly in case of both increase and decrease of the congestion window size. 3. In case of greedy data transfer congestion control , iTCP is inefficient because the congestion window retains a small size due to consecutive timeouts and duplicate ACKs. 4. Selecting the sensitivity granularity (alpha) is difficult and should be done based on the best possible outcome |
| Solution to the Limitation | * Newer TCP congestion control techniques are being proposed like Smesh, Wing. * Here gateways are implemented which may act as both destination or source but congestion occurs here also. * The field of congestion control required quite a lot of work. * Implementation of application limited data transfer (method) to prevent iTCP from retaining small congestion window size . * The time period for training may be decreased instead of keeping it in real time but again it loses efficacy or accuracy. * Doing the above thing may decrease the convergence problem also. |

Conclusion:

Hence, we see various drawbacks and also advantages of the present TCP in wireless networks. But we try to build on the drawbacks of the TCP. An intelligent TCP congestion technique named as iTCP is suggested herein. And it is also kept in mind that not radical changes are made in the present basis structure of the TCP.

A complete design is proposed for the new congestion control system and it is run as an experiment in ns-2. A real bed test is also done on the same. An overall of improvement in the network throughput and energy consumed per transmitted bit is obtained in iTCP. Moreover, there are no system overheads in this as compared to the other network congestion technique. We see that Itcp works quite well in large network meshes with medium node density.

Hence the data transfer capability of iTCP is exploited in this regard. But in the recent years many more techniques have also evolved like Smesh which provide infrastructure WMN. In an WMN there are quite a number of gateways to serve all the nodes in the network. The problem of congestion again arises at these gateways as these gates act as destination and also sources. Hence, this field of network congestion requires quite a lot of work in the near future.

Reference:

59. Jacobson, V. (1988). Congestion avoidance and control. ACM

SIGCOMM Computer Communication Review, 18(4), 314–329.

34. Floyd, S., Henderson, T., & Gurtov, A. (1999). The newreno

modification to TCP’s fast recovery algorithm. RFC 2582.

100. Stevens, W. (1997). Rfc 2001: TCP slow start, congestion

avoidance, fast retransmit, and fast recovery algorithms. January

1997.

35. Floyd, S., Mahdavi, J., Podolsky, M., & Mathis, M. (2000). An

extension to the selective acknowledgement (SACK) option for

TCP. RFC Editor.

71. Mathis, M., & Mahdavi, J. (1996). Forward acknowledgement:

Refining TCP congestion control. In Proceedings of the conference

on applications, technologies, architectures, and protocols

for computer communications, pp. 281–291, August

28–30.

10. Brakmo, L. S., & Peterson, L. L. (1995). TCP Vegas: End to end

congestion avoidance on a global internet. IEEE Journal on

Selected Areas in Communications, 13(8), 1465–1480.

42. Ha, S., Rhee, I., & Cubic, L Xu. (2008). A new TCP-friendly

high-speed TCP variant. ACM SIGOPS Operating Systems

Review, 42(5), 64–74.

98. Song, K. T. J., Zhang, Q., & Sridharan, M. (2006). Compound

TCP: A scalable and TCP-friendly congestion control for highspeed

networks. In Proceedings of PFLDnet 2006.

1. Abdeljaouad, I., Rachidi, H., Fernandes, S., & Karmouch, A.

(2010). Performance analysis of modern TCP variants: A comparison

of cubic, compound and new reno. In 2010 25th Biennial

symposium on Communications (QBSC), pp. 80–83. IEEE, 2010.

67. Liu, S., Basar, T., & Srikant, R. (2006) TCP-illinois: A loss and

delay-based congestion control algorithm for high-speed networks.

In Proceedings of the international conference on performance

evaluation methodologies and tools (VALUETOOLS),

October 11–13 2006.

61. Jehan, M., Radhamani, G., & Kalakumari, T. (2011). Vegas:

Better performance than other TCP congestion control algorithms

on manets. International Journal of Computer Networks

(IJCN), 3(2), 151–158.

109. Wei, D. X. & Cao, P. (2006). NS-2 TCP-linux: An NS-2 TCP

implementation with congestion control algorithms from linux.

In Proceeding from the 2006 workshop on NS-2: The IP network

simulator, p. 9. ACM.