The Corresponding Options of TCP Variants for Fairness Problem in AD HOC Networks

Alem Čolaković, Tarik Čaršimamović

Abstract : ***The ad hoc network is a continuously self-configuring and decentralized network where nodes communicate with each other without the fixed network infrastructure or centralized administration. TCP (Transmission Control Protocol) is a connection-oriented transport protocol that provides a reliable exchange of data streams. Implementation of TCP in wireless networks has many challenges such as the* issues *of the efficiency and TCP* fairness *problem. The fairness means that network nodes (users or applications) are receiving a fair share of overall resources. In this paper, we study the problem of maintaining the fairness for TCP connections in ad hoc networks. Our research has been made to present the TCP fairness problem in MANET (ad hoc mobile networks) while considering the sending and receiving of traffic. Achieving fairness in these networks is a challenge due to specific characteristics of an ad hoc environment and it is necessary to adapt TCP for ad hoc networks. The primary goal of this paper is to present fairness in ad hoc networks using combinations of different TCP variants and routing protocols. We evaluated the results of our research by using the proper simulation method.***

Index Terms: Ad hoc, MANET, VANET, TCP, fairness.

I. INTRODUCTION

The *ad hoc* network is a continuously self-configuring network that enables communication between nodes without fixed network infrastructure and central administration. MANET (*mobile ad-hoc network*) is an *ad hoc* network with mobile nodes that perform the exchange of control information. TCP (*Transmission Control Protocol*) is a transport protocol that was not originally developed for wireless networks and it exhibits serious network performance degradation in these networks. *Various TCP variants have been* proposed such as Tahoe, Reno, New-Reno and Vegas that make some improvements and extensions of standard TCP but there is no universal TCP variant that works well in all network scenarios including different network sizes, traffic loads, node mobility patterns, etc.

According to paper [1] the biggest challenge in MANET is the design of TCP variant which should give the best performance in all network scenarios. This has been an area of active research recently, and progress has been reported in several directions with different types of challenges that are posed to TCP design among such networks [2].

One of the challenges is to obtain proportional share of the network resources for network nodes and this problem is known as TCP fairness problem.

Previous studies of TCP fairness are mostly based on simulation results focusing on throughput and only a small number of researches are based on analytical studies. Mechanisms to improve TCP fairness in ad hoc networks can be grouped into two groups: layered design and cross-layer design [3]. Layered design involves some changes on MAC or transport layer, while cross-layer design involves interactions among TCP, routing and MAC protocols and include some information to diagnose the reasons of fairness and performance degradation. It was found that the main reason lies in the unfairness of MAC (*Media Access and Control*) protocol, while the TCP timeout mechanism makes the unfairness more severe [4]. The existing TCP variants cannot yield a good performance in a highly mobile environment due to the fact that TCP is not able to differentiate the cause of the packet drops [5]. Also, there is a significant influence of routing protocol to performance of *ad hoc* networks because each node must be able to forward data to other nodes. The efficient routing algorithms can provide remarkable benefits in *ad hoc* networks, including higher throughput, lower average end-to-end delay, less number of dropped data packets and generally an ameliorated network performance [6].

TCP fairness problem is particularly pronounced in wireless networks and despite some improvements there is no fair treatment of all traffic flows. According to described facts, TCP variant and routing protocol election in ad hoc networks presents a great challenge. Our focus is on the corresponding options of TCP variants for TCP fairness problem in ad hoc networks. The rest of the paper is organized as follows: Section (2) presents TCP protocol challenges in ad hoc networks and the focus is on fairness performance. Section (3) presents existing fairness metrics and describes metric used in this paper. Section (4) presents mechanisms to improve the fairness in ad hoc networks according to related works. Section (5) presents a simulation model and we present the results of our case study. The concluding remarks and future research ideas are given in Section (6).

II. TCP PROTOCOL FOR AD HOC NETWORKS

TCP performances in ad hoc networks depends on various factors including the buffer size, topology, static or mobile configuration of network (SANET or MANET), presence of hidden stations, channel capacity, implementation details of the MAC layer, etc. TCP provides reliable data transfer with full duplex connection and flow control mechanisms which include the avoidance of networks congestion. At each arrival of a packet to the destination, an ACK is sent back to the source with the

Manuscript Received on April 2015.

M. Sc. Alem Čolaković, System Engineer, Faculty of Traffic and Communications, University of Sarajevo, Bosnia and Herzegovina.

Dr. Tarik Čaršimamović, Assistant to Exeucitve Director for Business Development, BH Telecom JSC Sarajevo, Bosnia and Herzegovina.

information of the next sequence number that is expected (Fig.1).

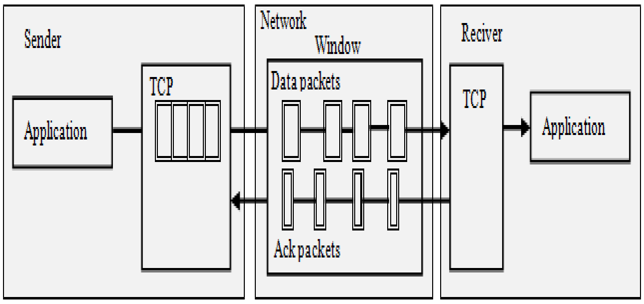

Fig. 1 Flow control based on congestion window

Congestion in a network occurs whenever the demands exceed the maximum capacity of communication link, especially when multiple traffic streams try to access a shared media simultaneously [7]. Congestion leads to packet losses while TCP triggering mechanism reduces the speed of sending packets and thus reduces the transmission performance. TCP protocol assumes that all packets losses are due to congestion, but wireless networks might suffer from losses that are related to other causes. A possible erroneous conclusion that all losses are inducted by congestion leads to retransmission that causes unnecessary degradation of network performances. One of the major TCP problems in ad hoc networks lies in performing congestion control when the losses are not induced by network congestion. Also a challenge is to choose the appropriate TCP congestion window size. The small congestion window does help TCP connections to improve fairness but it is less efficient for throughput.

There are different variants of the TCP protocol that improve congestion control and efficient management of network resources. The improvements are mainly distinguished by fast retransmission algorithms. Basic TCP algorithms are: *Slow Start*, *Congestion Avoidance*, *Fast Retransmission* and *Fast Recovery* and these algorithms are valid for TCP Tahoe, while other TCP variants (Reno, New-Reno, Vegas, etc.) include some extensions. Some of the algorithms are proposed to distinguish packet losses due interference, collision or links failure induced loss. The challenge is to find a TCP variant which should give the best fairness performance in all network scenarios.

Slow Start algorithm forces the TCP sender to decrease the sending rate and then increasing it while finding the available bandwidth between nodes. When packet loss occurs TCP invokes the *Congestion Avoidance* mechanism. On the source side there is *cwnd* which represents a measure of the capacity of network, and *rcvwnd*, which represents a measure of the available capacity on the destination side. The maximum number of unacknowledged segments is expressed as *min* {*cwnd, rcvwnd*}. When packet loss occurs the window size is reduced and *Slow Start* stage is performed. During *Slow Start* stage TCP Tahoe increases window size exponentially i.e. for every acknowledgement received it sends two packets as shown in Fig. 2. After the first slow start period is over the congestion window reduces to its half size. The moment when the window size outperforms the product *throughput x delay*, the packet losses occurs on the transmission link and TCP begins with the *Congestion Avoidance*. During *Congestion Avoidance* TCP increases the window size by one packet per RTT (*Round Trip Time*). Small congestion window usually causes better fairness for TCP flows but it causes decreasing throughput. One of the tasks is to find optimal congestion window size or to find mechanism to improve fairness when using large congestion window. The window size is reduced by half of its actual size for every packet loss that has been detected and this algorithm is known as AIMD *(Additive-Increase/Multiplicative-Decrease). This algorithm is used by Tahoe and* packet loss probability *is used to adjust window size* in order to achieve congestion control.

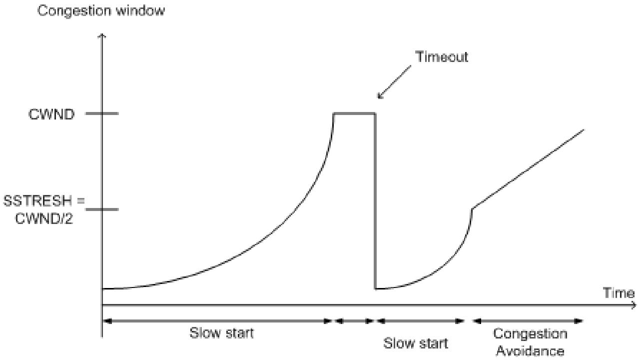

Fig. 2 Slow start and congestion avoidance

If congestion continues the transmitter responds with complete suspension of sending packages. This control limits the number of packets sent by the sender before the arrival of *acknowledgement packets* from receiver. All that leads to unfair usage of network resources.

Reno TCP includes some *improvements* to the *Tahoe* TCP so the packets losses are detected earlier and the pipeline is not emptied every time a packet loss occurs. The Reno TCP uses *Fast Retransmit* that allows continuing congestion avoidance instead of starting the *Slow Start* algorithm after receiving repeated confirmation (observing three *acknowledgments*). By not waiting for the RTO (*Retransmission Timeout*) interval to send the appropriate segment Reno increases the utilization of network resources because there is no waiting for the expiration of the timer. Whenever 3 duplicate ACKs were received it takes as a sign that the segment was lost and retransmitting begins without waiting for timeout. TCP Reno does not reduce the congestion window to 1 after a packet loss occurs while it starts *Fast Recovery* that allows higher throughput under congestion environment. *TCP Reno* performs very well when the packet losses are small because it detects a single packet loss. But when there are multiple packet losses in one window then Reno doesn’t perform too well and its performance is almost the same as Tahoe under conditions of high packet loss [1]. *TCP New-Reno* brings a slight change in the retransmission timer and it is more efficient when multiple packet losses occur. The issue of this TCP version is limitation of detecting and resending only one packet loss per RTT.

TCP Vegas extends Reno retransmission mechanisms as recording RTT of each packet. *Bandwidth Estimation Scheme* is used according to the difference between the expected flow rates (estimated throughput) and the actual flow rates (measured throughput).Threshold is defined as the difference between the expected and actual bandwidth and it is used to adjust the *Window Size*. When measured throughput is smaller than the expected it indicates that the available bandwidth is not fully utilized and TCP Vegas can take the action e.g. increases or decrease the *Window Size*. *TCP Vegas* has problems when packets do not follow the same route because RTT recorded from the previous route is no longer accurate.

TCP unfairness in ad hoc networks is a result of the *shared* usage of *wireless* channel among multiple neighboring nodes and because of location dependency, e.g. some flows experience more packet loss and thus tends to reduce their congestion window more frequently than others. This indicates that not every problem and packet loss should be interpreted as congestion because packets losses in *ad hoc* networks may be occurred due to other reasons such as routing algorithm failures, handover failures, weak wireless connection that leads to link brakes, etc. Short-lived connections end very quickly and it will not affect other flows as much as long-lived TCP flows. When some node experiences a “bad phase” of the communication channel, it slows down transmission rate by shrinking its congestion window, translating the lost packets as an indication of congestion. The other node, at the absence of packet drops, continues increasing its window size and injecting more packets into the network. Eventually, some nodes can occupy the bandwidth that should be used by other nodes. At that time, the node that experienced the error is unable to utilize its fair share of the network resources. After the “*bad phase*” for the first node is over, the protocol will try to use more bandwidth but the network will be occupied by the other nodes that lead to TCP’s deficiency in term of fairness level. Due to these facts there is a need for solutions that will enable differentiation of packet losses incurred due to congestion and losses incurred as a result of other factors. Motivated by the above discussion, we carry out a simulation based study to examine the effects of the various TCP variants at the fairness characteristics of a network.

III. FAIRNESS METRICS

There are many interpretations of fairness and its meaning and that leads to several mathematical and conceptual definitions. We consider resource consumption in an ad hoc network and if each node is consuming proportional resources it is considered as fair. There are many factors that influence TCP fairness in ad hoc networks such as buffer size, topology, static or mobile configuration of network (SANET or MANET), presence of hidden stations, channel capacity, implementation details of the MAC layer, routing protocol, etc. Inability to discover the reason of packet loss, hidden nodes and other problems may cause that some nodes occupy network resources more than others and that leads to poor fairness. One of the challenges is to find the best combination of available mechanisms to enhance fairness performance in ad hoc networks and to do that an adequate fairness metric is needed.

There are several metrics that are widely used to quantitatively demonstrate fairness such as Raj Jain's proportional fairness index, Standard deviation, min-max ratio, channel saturation index, channel occupation index, etc. Generally, those metrics were not able to satisfy excepted needs because of the assumption that network nodes operate under homogenous conditions. Most of the studies focusing on fairness in IEEE 802.11 networks (including both simulation and measurement work) implicitly or explicitly rely on these assumptions. [8] There are only few metrics that consider several parameters such as DA-index (*demand-aware fairness metric*) that include channel demands of nodes while calculating fairness index.

In our case study we do not attempt to examine efficiency of existing metrics or to propose a new fairness metric. We use Raj Jain's fairness index that is widely accepted by the researchers and our goal is to examine which is the best *variant* among *TCP* protocols according to fairness performance. Raj Jain's equation of fairness is formally defined as follows:

$$I(x_1, x_2, \dots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \cdot \sum_{i=1}^{n} x_i^2} \quad [9]$$

Indicator n is number of users and x_i are users’ allocation for *i-th* connection. The result ranges from *1/n* (the worst case) to *1* (the best case – 100% fairness system), and it is maximum when all users receive the same allocation of network resources. One of the advantages of this metric is that it is agnostic of the nature of allocated resources (x), i.e. it can be *packets*, *bytes*, *dropped traffic*, *power consumption*, *throughput*, etc. To determine TCP fairness in wireless network some authors use average throughput such as in [10], channel occupancy time [11], or other indicators such as power usage, response time, etc.

IV. RELATED WORKS

Numerous solutions and mechanisms for solving TCP fairness problems in *ad hoc* network environments have been proposed, but there is no guarantee for a fully fair distribution of network resources. The IEEE 802.11 standards enable wireless *ad hoc* networking by using DCF (*Distributed Coordination Function*) for multiple accesses to the shared radio channel and adopting CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) algorithm for that environment. Unfortunately, the interaction between TCP dynamics, driven by the AIMD (*Additive Increase Multiplicative Decrease*) paradigm, and DCF channel access rules, which are based on the CSMA/CA algorithm, leads to an inefficient spatial channel usage [12]. The IEEE 802.11 DCF protocol can lead to severe unfairness, i.e. some nodes occupies the whole channel capacity while others are starved [4]. Mechanisms to improve TCP fairness in *ad hoc* networks can be grouped into two groups: layered design and cross-layer design.

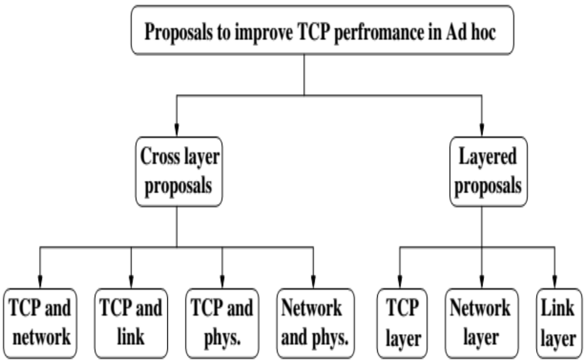

Fig. 3 Classification of proposals to improve TCP performance in Ad hoc networks [3]

Some TCP fairness improvements assume to perform local retransmission without starting a TCP congestion control mechanisms. Using local retransmission at the link layer, instead of an end-to-end retransmission, significantly reduces the probability of packet loss due to problems of the wireless link. Several Link Layer improvements have been proposed such as AIRMAIL (*Asymmetric Reliable Mobile Access in Link-layer*), fixed RTO is sender-based technique, TCP Snoop that not rely on feedback from the network.

TCP Snoop is implemented on a link layer of the TCP/IP protocol stack in order to reduce the degradation of network performance using the PEP (*Performance Enhancing Proxy*). The idea of this layered proposal is to hide any losses incurred on the link layer from the sender. The Snoop protocol runs on a Snoop agent that is implemented in a base station or wireless devices as presented in paper [13]. A packet loss is detected by the arrival of a small number of duplicate acknowledgments from the receiver or by a local timeout. The TCP sender does not know and thus there is no start of a congestion control mechanism. By acting on the transport layer, TCP is not familiar with packet losses due wireless link failures and it performs a local retransmission of lost segments without running the TCP congestion control algorithm. The Snoop module supervises all segments that pass the TCP connection in both directions. Snoop stores the TCP segments received from higher layers that have not been acknowledged by the recipient. For storing segments a snoop cache is used and this *store and forward* procedures helps to perform local retransmission of lost segments without informing the sender. The size of the memory of stored segments is proportional to the size of the sliding window on the receiver side. Snoop cache memory releases when receiving an ACK.

Transport Layer approaches are mostly based on TCP congestion control mechanism. TCP variants have various mechanisms that influence on fairness performances as it has been presented in paper [14] for Reno and Vegas. There are also some improvements based on implementing other transport protocols instead of TCP such as MCTCP (*Mobile Control Transport Protocol*), SCTP (*Stream Control Transmission Protocol*), ATP (*Autonomous Transport Protocol)*, TCP DOOR (*TCP Detection of Out-of-Order and Response)*, etc. For example, TCP DOOR is a Layered proposal based on end-to-end approach. This improvement is based on idea to implement some mechanism at the ends of the network that significantly reduce the complexity of the network and the idea of error notification remains the same. End-to-end protocols use selective acknowledgments (SACKs) to allow sender recovery when multiple packet losses occur. SACKs were added as an option to TCP by RFC 1072 [15] and it has been suggested to add as an option to TCP in RFC 2018 [16]. According to this idea it attempts differentiate congestion packet losses from other losses by using ELN (*Explicit Loss Notification*). ELN is based on the fact that the sender is notified about the true cause of data loss. The sender gets a notification message about the real reason of data loss and according to that information it prevents wrong interpreting of congestion. ELN and ECN (*Explicit Congestion Notification*) can provide information to the user about the reason of packet losses and some improvements based on this idea has been presented in paper [17]. In Mobile TCP the retransmission mechanism is triggered but there is no reduction of window size if losses are detected and this avoids reducing of network performance.

Link Layer improvements are mainly based on error correction using FEC (*Forward Error Correction*) and retransmission by using ARQ (*Automatic Repeat Request*). One of most popular approach is adding timer to the MAC layer. The Timer is responsible to add delay to the TCP packets before sending, so the traffic of other nodes gets the chance to access the media. In paper [18] an approach to differentiate TCP and UDP flows with the help of IEEE 802.11e standard has been presented. Major problem of MAC layer solutions is implementation difficulty because all wireless nodes require having consistent MAC protocol (any modification of the MAC protocol requires update at all nodes). There are also some back off-based improvements that are based on modification of the *back off policy* of MAC protocol.

There are several routing protocols as AODV (*Ad Hoc On-Demand Distance Vector*), DSDV (*Destination-Sequenced Distance Vector*) and DSR (*Dynamic Source Routing*) that are widely used in ad hoc networks. Choosing routing algorithms such as DSR, AODV and DSDV in ad hoc networks with dynamic change of topology have a direct impact on TCP fairness in the ad hoc network with mobile nodes and it is shown in paper [14]. AODV and DSR immediately look for an alternative route after change of topology while DSDV is looking for new routes in a periodic manner. Although AODV and DSR are both on-demand reactive routing protocols and they share similar on-demand behavior, the differences in the protocol mechanisms can lead to significant fairness issues. Experimental results obtained in [6] showed which routing protocol is better when throughput, end-to-end delay and packet loss was observed. Although numerous routing protocols have been proposed for mobile ad hoc networks, there is no universal scheme that works well in scenarios with different network sizes, traffic loads and node mobility patterns, so mobile ad hoc routing protocol election presents a great challenge [19].

In paper [20] the extended original RED (*Randomly*

Early Detection) solution called NRED (*Neighborhood RED*) has been proposed to improve TCP fairness in ad hoc networks. An algorithm of the NRED scheme and guidelines for setting configurable parameters has been given. It uses original RED algorithm to calculate drop probability where each neighbor node computes its local drop probability based on its channel bandwidth usage. This calculation is used to drop packets accordingly. NRED is implemented on a network layer and there is no modification to MAC layer. Calculated drop probability propagates to neighboring nodes that cooperative packet drops should be performed. Simulation studies confirm that the NRED scheme can improve TCP unfairness substantially in ad hoc networks [20] but its implementation decreases the aggregated throughput of TCP flows.

Some solutions are based on scheduling (pacing) new packets according to the transmission interval formed from scaled RTT and congestion window. This idea is based on an improved channel reuse while specific scale *parameter x* has been given. Solution proposed in [21] includes adding extra adaptive delay in scheduling to „*penalize aggressive nodes*“ and simulation results show that proposed scheme eliminates the extreme unfairness.

The several cross-layer solutions have been proposed such as TCP-F and ATCP. TCP-F (*TCP Feedback*) is an example of cross layer proposal that allows the TCP senders to distinguish losses due to routes failure and network congestion. Routing agent of a node detects route failure and sends a RFN (*Route Failure Notification*) packet to the source. Source goes into a snooze state until it is notified of the restoration of the route through RRN (*Route Re-establishment Notification*) message. ELFN (*Explicit Link Failure Notification technique*) is also example of cross layer solution with similar idea based on interaction between TCP and the routing protocol. ATCP (*Ad Hoc TCP*) is a cross layer proposal without any changes in TCP protocol but adjusts the feedback. It is implemented in the layer between TCP and IP and monitors the status of the network. The "*Destination Unreachable*" and ECN (*Explicit Congestion Notification*) messages by the ICMP (*Internet Control Message Protocol*) are used and ATCP puts the sender into the appropriate state. ATC (*Adaptive Transmission Control*) algorithm has been proposed in paper [23] to improve the short-term fairness without unduly degrading the throughput. Implementation of a thin layer between TCP and IP that improves end-to-end TCP throughput without modification of standard TCP has been proposed in paper [24].

Some fairness improvements include *distributed scheme with adaptive pause*. Using this *scheme* each node monitors the channel usage while dynamically determines whether it should temporary stop a time interval in order to avoid channel capture. Simulation results obtained in paper [22] show that this scheme can improve the TCP fairness. It is very important that this scheme is simple and requires less overhead. In paper [12] a cross-layer algorithm has been proposed to dynamically limit the number of in flight segments in a TCP connection. By exploiting proper interactions between the MAC and the transport layer the fairness performances could be improved.

TCP variants are based on various mechanisms that lead to differences in network performances such as fairness. Although several improvements have been proposed, neither solution does not solve problem of TCP fairness in ad hoc networks. There is no universal TCP variant that works well in all network scenarios and it is a challenge to find an appropriate combination of TCP variants and other protocols to get the best network performances while operating under different networks conditions.

V. SIMULATION AND ANALYSIS

A. Simulation cases

TCP variants are based on various mechanisms that lead to differences in fairness performance. We have used Raj Jain's metric to calculate fairness indexes in ad hoc networks and to compare results obtained by different TCP variants. We simulated different scenarios while using various routing protocols such as DSDV, AODV and DSR. DCF (*Distributed coordination function*) is the fundamental MAC technique of the IEEE 802.11 based standard. DCF employs a CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) access method to avoid collisions by transmitting only when the channel is sensed to be "*idle*". If some nodes occupy the channel then the others should wait for the „*idle*“ state and that causes unfairness in channel usage. Therefore, we examine fairness considering the number of sent and received traffic of each node. The simulation has been implemented using Network Simulator version 2 (NS-2).

Simulation was conducted in two different scenarios while considering fairness performances of TCP variants including *Tahoe, Reno, New Reno and Vegas*. Our simulation scenarios include *NxN* grid network shown in Fig. 4. They can present a special type of an ad hoc network called VANET (*Vehicular Ad Hoc Network*). It consists of a high mobility of cars referred as 9 nodes (node 0… node 8) that provide a way to exchange information between cars without depending on fixed infrastructure.

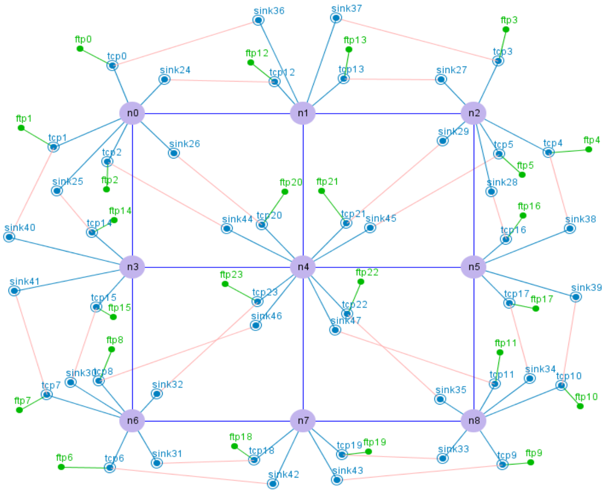

Fig. 4 Simulation topology

Each node communicates with its immediate neighbors and FTP connections are used as TCP traffic. In our simulation scenarios all traffic flows start and finish at the same time while we observe TCP fairness according to sent and received traffic. Other simulation parameters are given in Table 1. and these parameters are the same for both simulation scenarios.

Table 1 Simulation parameters

Parameter	**Value**
TCP variants	Tahoe, Reno, New-Reno, Vegas
Flow type	FTP
MAC Layer	802.11
Routing protocol	DSDV, DSR, AODV
Topology area	1000m x 500m
Simulation time	60 sec.
Channel bandwidth	11 Mbps
Basic rate	1 Mbps
Buffer management	Drop Tail
Queue limit	50 packets
Packet size	1024 Bytes

B. Simulation results

In the first simulation scenario all nodes are mobile with the same movement speed 15 m/s. Movement direction and distance from other nodes are the same in every moment and each node communicates with its immediate neighbors. Nodes have different sending and receiving demands and other parameters are the same for all simulation scenarios as described in previous chapter. This scenario cannot represent the case of real world networks because of homogenous conditions. Fig. 5.1 and Fig. 5.2 presents the results of the first simulation scenario while observing sent and received traffic. According to sent traffic the value of the fairness index for each protocol combination shows that the Tahoe, Reno and New Reno flows yield the best performance when DSDV routing protocol was used. Vegas obtained better fairness results than other TCP variants while AODV routing protocol was used and only Vegas has fairness index bigger than 0.8 while using different routing protocols. According to received traffic only Vegas has the fairness index 0.8 for AODV routing protocol while other combinations have lower fairness indexes.

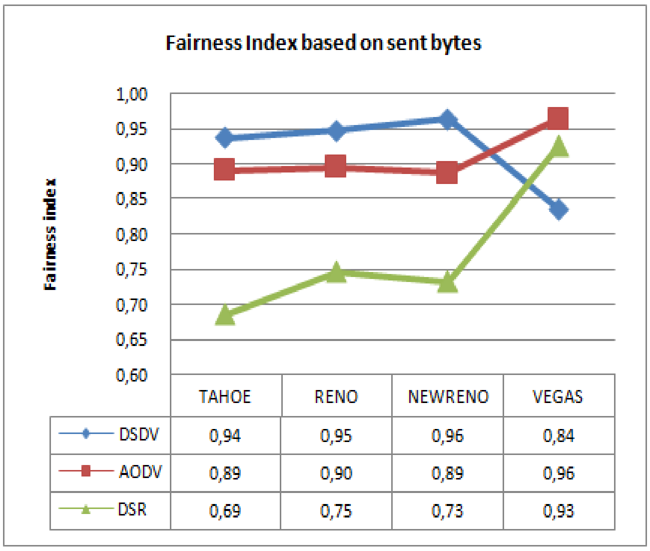

Fig. 5.1 Fairness Index for Tahoe, Reno, Newreno and Vegas for scenario 1.

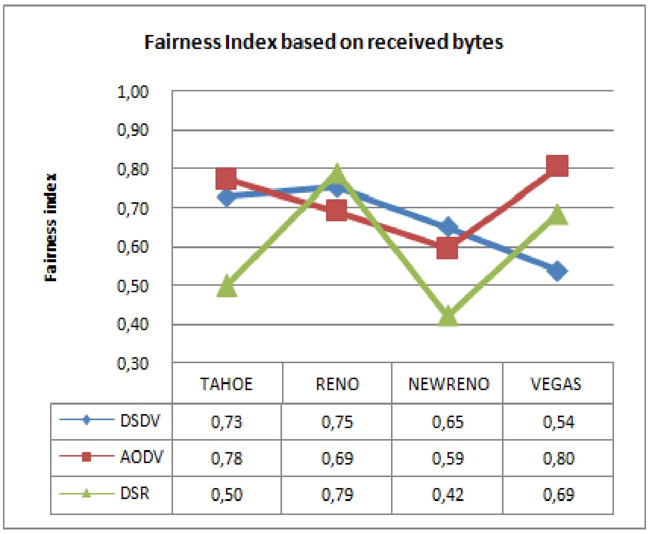

Fig. 5.2 Fairness Index for Tahoe, Reno, Newreno and Vegas for scenario 1.

Fig. 6.1 and Fig. 6.2 presents the results of second simulation scenario while observing sent and received traffic. Second simulation scenario includes observing nodes with different sending and receiving demands while movement direction and speed of nodes are different. These conditions have significant effect on the energy consumption of wireless nodes while there is dynamic topology change. This scenario can represent realistic VANET (*Vehicular Ad Hoc Network*) network. In both cases the worst fairness index is when using DSR routing protocol for every TCP variant. All TCP variants have similar fairness indexes for DSDV and AODV routing protocols while observing sent traffic. If we observe received traffic all TCP variants have the best fairness performances while AODV routing protocol was used and in that case fairness indexes are bigger than 0.8.

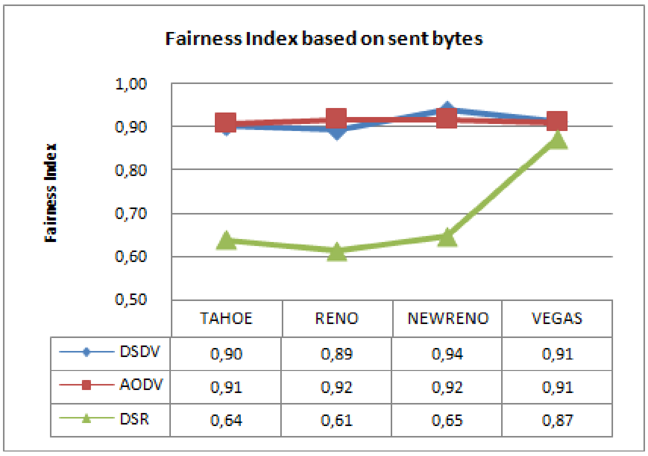

Fig. 6.1 Fairness Index for Tahoe, Reno, New Reno and Vegas for scenario 2.

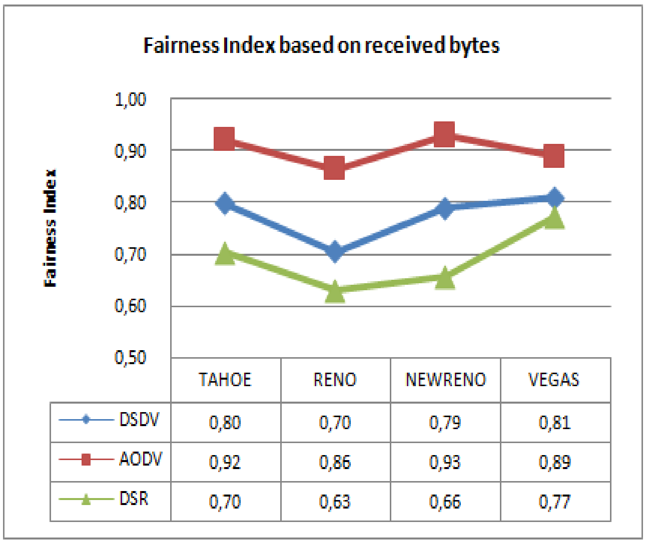

Fig. 6.2 Fairness Index for Tahoe, Reno, New Reno and Vegas for scenario 2.

It would be useful to find and compare the fairness index according to other parameters such as throughput usage, dropped traffic, etc. Fig. 7.1 presents TCP fairness indexes while dropped traffic was observed for scenario 1.

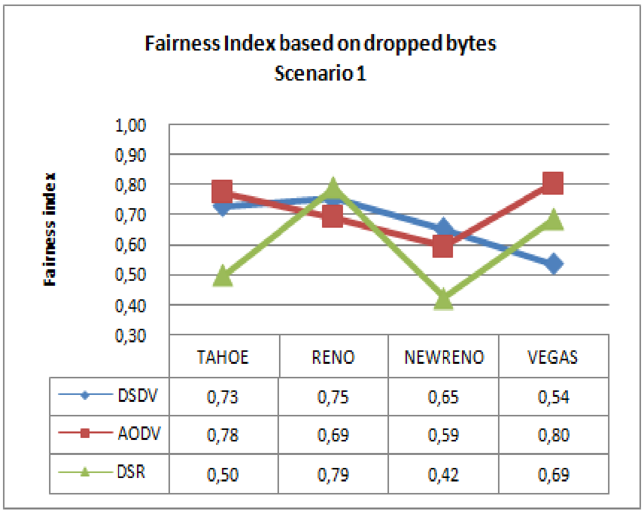

Fig. 7.1 TCP fairness Index based on dropped bytes for scenarios 1

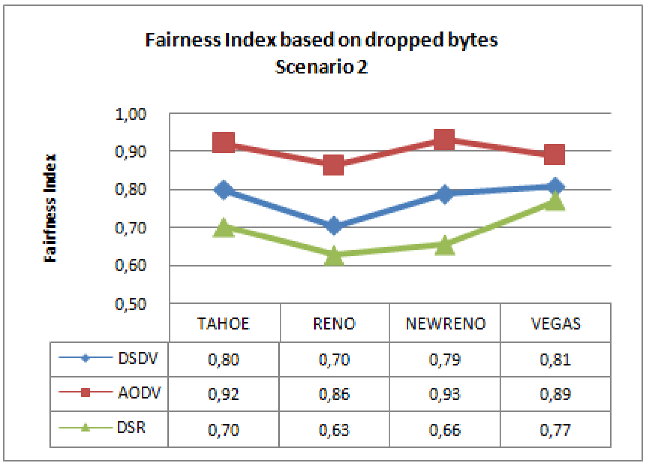

Fig. 7.2 TCP fairness Index based on dropped bytes for scenarios 2

Fig. 8 presents differences between fairness indexes obtained while observing sent and received traffic for scenario 1. We can see that there are significant differences unfairness results while observing sent and received traffic. In this simulation scenario only TCP VEGAS obtain farness index 0.8 or larger in both cases when AODV was used as routing protocol.

Fig. 9 presents differences between fairness indexes obtained while observing sent and received traffic. The worst TCP fairness performances are while using DSR routing protocol in both cases. For most cases the best TCP fairness performances are while using AODV as routing protocol and in this case all values of fairness indexes are larger than 0.8. Similar results were obtained while using TAHOE as transport protocol and DSDV as routing protocol.

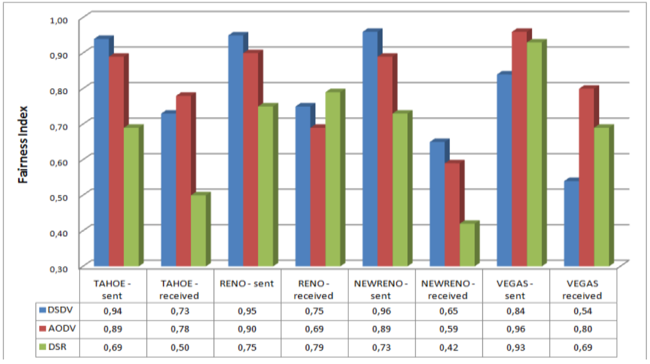

Fig. 8 Comparison of TCP fairness index for scenario 1.

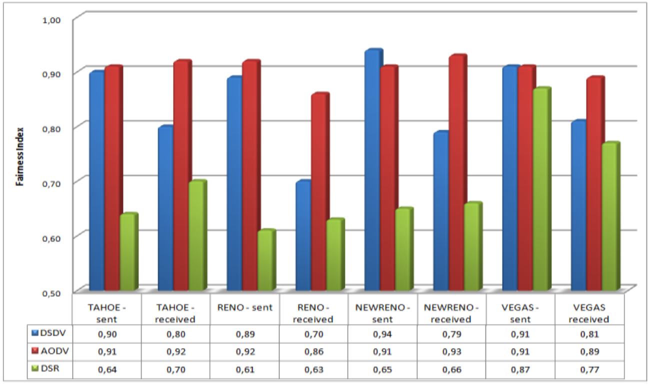

Fig. 9 Comparison of TCP fairness index for scenario 2.

In order to compare results of both simulated scenarios, Fig. 10, 11, 12 presents fairness indexes for both scenarios and cases of sent and received traffic. According to the results there are differences between fairness results obtained while observing sent and received traffic. We can see differences in TCP fairness performances in different scenarios that mean a significant effect on TCP fairness performances have dynamic topology change.

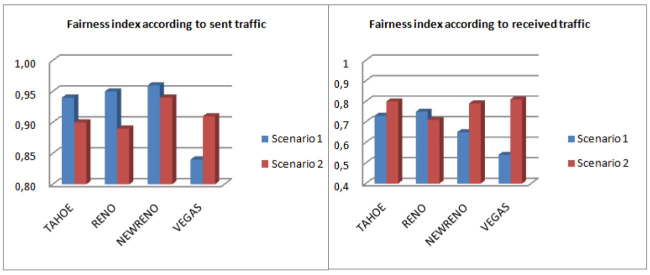

Fig. 10 TCP fairness index while using DSDV routing protocol

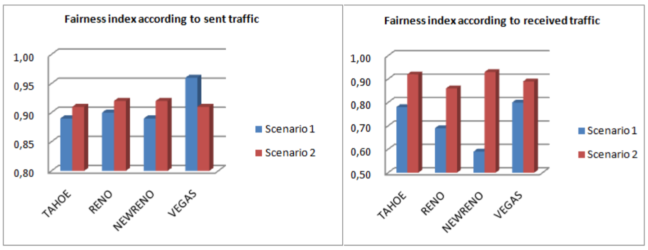

Fig. 11 TCP fairness index while using AODV routing protocol

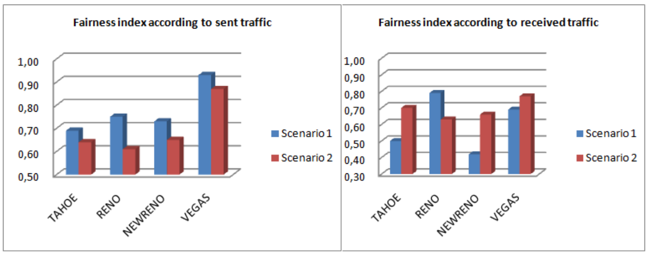

Fig. 12 TCP fairness index while using DSR routing protocol

Based on simulation results we can see that in ad hoc networks DSR outperforms AODV and DSDV routing protocols while observing TCP fairness performances. The most of TCP variants have poor fairness indexes while using DSR as routing protocol. In most cases the best TCP fairness performances are while using AODV as routing protocol. TCP variants showed *differences* in fairness performances, so according to our simulation results in both scenarios and all cases we can see that VEGAS have the best fairness performances. This means that *there* is significant fairness influences of TCP mechanisms that are involved in different TCP variants. According to these facts, election of TCP variant and other protocols such as routing protocol is a challenge especially when different scenarios are observed.

VI. CONCLUSION

In this paper we indicate some TCP fairness problems and we examined TCP fairness in ad hoc networks while using different TCP variants. We also indicated a problem of finding adequate *parameters-aware* fairness metric to include different network performances while calculating fairness indexes. It is a challenge to create a metric that reflects the fairness according to different parameters such as sent and received traffic, throughput consumption, dropped traffic, traffic demands, etc.

Our simulation results shows that traffic demands, topology change, energy consumption and routing protocol play a major role in TCP fairness results. According to simulation results there is no universal TCP variant that works well in different network scenarios with various numbers of nodes, traffic demands, node mobility patterns, etc. It is a challenge to find the best combination of TCP variant and other protocols, such as routing protocol, to get the best network performances while operating under different networks conditions. TCP variants are based on various mechanisms that lead to differences in network performances and there is a possibility of improving these mechanisms with significant influence on network performances including fairness. TCP protocol assumes that packets losses are due to congestion since bit error rates are very low in wired networks but this is not case in wireless networks because there are several types of losses that are not related to network congestion. The major problem of TCP fairness in ad hoc networks lies in performing congestion control in case of losses that are not induced by network congestion. According to that a congestion control has become one of the key issue in ad hoc networks that has a significant influence on network performances including fairness.

Our future works will be focused on study and improving solutions for congestion notification and control while investigating algorithms for estimating queue size and other indicators that influence on fairness gain. Also, we will be focused to examine differences between existing fairness metrics in real network environment and try to find appropriate metric with appliance in non-homogenous networks conditions. Our future works will also include study of TCP fairness when different TCP variants compete with each other for network resources and competition between TCP and other flows such as multimedia streaming.

REFERENCES

1. Saylee Gharge, Ajinkya Valanjoo: "Review of different TCP variants in Ad hoc networks", VESIT, International Technological Conference-2014 (I-TechCON), Jan. 03 – 04, 2014.
2. Hatim Mohamad Tahir, Abas Md Said, Alex W.M. Tan: „Improving TCP fairness Flows in Wireless Ad Hoc Networks", Information Systems International Conference (ISICO), 2-4. December 2013.
3. Ahmad Al Hanbali, Eitan Altman, Philippe Nain: "A Survey of TCP over Ad hoc Networks", Communications Surveys & Tutorials, IEEE (Volume:7 , Issue: 3), 2006.
4. Linfang Dong, Shang Liu: "Research on TCP Fairness Improvement over Wireless Ad Hoc Networks", Distributed Computing and Applications to Business Engineering and Science (DCABES), 2010 Ninth International Symposium on 10-12 August 2010.
5. Ms.M.Santhiya ,Mr.M.Sunil karthick, Ms.M.keerthika: "Performance of various TCP in Vehicular Ad Hoc network based on timer management", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Issue 12, December, 2013.

6. Vijendra Rai: “Simulation of Ad-hoc Networks Using DSDV, AODV And DSR Protocols And Their Performance Comparison” Proceedings of the 4th National Conference; INDIACom-2010 Computing For Nation Development, February 25 – 26, 2010.
7. Yi Lu, Yuhui Zhong, Bharat Bhargava: „Packet Loss in Mobile Ad Hoc Networks“, Center for Education and Research in Information Assurance and Security, Department of Computer Sciences, Purdue University, Wes Lafayette, USA.
8. Dimitry Kuptsov, Boris Nechaev, Andrey Lukyanenko: “A novel Demand-Aware Fairness Metric for IEEE 802.11 Wireless Networks“, Combra, Portugal, SAC'13 March 18-22, 2013.
9. Jain, R.; Chiu, D.M.; Hawe, W.: "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems", DEC Research Report TR-30, 1984.
10. Saar Pilosof, Ramachandran Ramjee, Danny Raz, Yuval Shavitt, Prasun Sinha: „Understanding TCP fairness over Wireless LAN“, IEEE INFOCOM 2003.
11. Tan G., Guttag J.: „Time-based fairness Improves Performance in Multi-rate WLANs”. In Proc. of USENIX'04, Boston, MA, 2004.
12. G. Boggia, P. Camarda, L. A. Grieco, T. Mastrocristino, and G. Tesoriere: „A Cross-layer Approach to Enhance TCP Fairness in Wireless Ad-hoc Networks“, Wireless Communication Systems, 2005. 2nd International Symposium, Siena, September 2005.
13. Chi Ho Ng, Jack Chow, Ljiljana Trajkovic: „Performance Ecaluation of TCP over WLAN 802.11 with Snoop Performance Enhancing Proxy“, In Proceedings of OPNETWORK, School of Engineering Science Simon Fraser University, Vancouver Canada V5A 1S, 2002.
14. Awedeh R.: „Compatibility of TCP Reno and TCP Vegas in wireless ad hoc networks“, IET Communications, vol. 1, issue 6, 2007.
15. V. Jacobson, R. Braden: “TCP Extensions for Long-Delay Paths” Network Working Group, Request for Comments: 1072, 1988.
16. M. Mathis, J. Mahdavi, S. Floyd, A. Romanow: “TCP Selective Acknowledgment Options” Network Working Group, Request for Comments: 2018, 1996.
17. K. Ramakrishnan, S. Floyd, D. Black: "The Addition of Explicit Congestion Notification (ECN) to IP", Network Working Group, Request for Comments: 3168, September 2001.
18. Forouzan Pirmohammadi, Mahmood Fathy, Hossein Ghaffarian: „TCP and UDP Fairness in Vehicular Ad hoc Networks“, International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 6, June 2012.
19. Sabina Baraković, Jasmina Baraković: “Comparative Performance Evaluation of Mobile Ad HocRouting Protocols” MIPRO, Opatija, Croatia, 2010.
20. Kaixin Xu, Mario Gerla, Lantao Qi, Yantai Shu: „Enhancing TCP Fairness in Ad Hoc Wireless Networks Using Neighborhood RED“, MobiCom '03, USA, 2003.
21. Yahia M: „Enhanced congestion control for TCP fairness in ad hoc wireless networks“, Industrial Electronics and Applications (ICIEA), 7th IEEE Conference on 18-20 July 2012.
22. Linfang Dong, Yantai Shu, Sanadidi M., Gerla M.: "A Method for Improving the TCP Fairness in Wireless Ad Hoc Networks", Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on 12-14 Oct. 2008.
23. Zhifei Li, Sukumar Nandi, Anil K. Gupta: "Achieving MAC fairness in wireless ad-hoc networks using adaptive transmission contro", Computers and Communications, 2004. Proceedings. ISCC 2004. Ninth International Symposium on July 2004.
24. Jian Liu, Singh Suresh: "ATCP: TCP for mobile ad hoc networks", Selected Areas in Communications, IEEE Journal on (Volume:19 , Issue: 7), July 2001.

AUTHOR PROFILE

M.Sc. Alem Čolaković works as System Engineer at Faculty of Traffic and Communications, University of Sarajevo, Bosnia and Herzegovina. Alem is PhD Candidate at Department of Communication Technologies.

Dr. Tarik Čaršimamović is Assistant to Exeucitve director for Business Development at BH Telecom JSC Sarajevo, Bosnia and Herzegovina and he is deployed as a professor at Faculty of Traffic and Communications and Faculty of Electrical Engineering in Sarajevo.