TCP NEW VEGAS: A NEW APPROACH TO IMPROVE TCP VEGAS PERFORMANCE BY DELAY REDUCTION AND INCREASED THROUGHPUT OVER MANET'S 9 h 16 31 10	GIAO THỨC TCP VEGAS MỚI: MỘT PHƯƠNG PHÁP MỚI ĐỂ CẢI THIỆN HIỆU SUẤT TCP VEGAS BẰNG CÁCH GIẢM TRỄ VÀ TĂNG LƯU LƯỢNG TIN TRÊN MANET'S
Abstract: With proliferation of wireless devices and increased popularity of wireless communication, researchers have shown immense interest in wireless	Tóm tắt: Với sự gia tăng của thiết bị không dây và tăng tính phổ biến của truyền thông không dây, các nhà nghiên cứu ngày càng quan tâm đến các mạng không dây. Sự di động của

networks. Mobility of nodes in wireless networks, leads to frequent link breakages. UDP supports unreliable communication over wireless networks. TCP protocol which is used to attain reliable communication over wired networks cannot be used directly over wireless networks due to its properties that were designed for wired networks. Therefore, to allow reliable communication over wireless networks, different variants of TCP protocol have been proposed. However, all the variants have their advantages and disadvantages. The proposed mechanism presents the study and comparison of TCP Vegas.The performance is measured with respect to the parameters throughput, delay and packet delivery ratio (PDR). Based on the disadvantages measures are proposed to overcome and the modified is analysed.
Keywords: TCP Vegas, NewTcpVegas, Congestion Window.

I. INTRODUCTION Transmission Control Protocol (TCP) is a connection oriented protocol designed for reliable and secure communication between the links. The reliability is achieved by the TCP by assigning sequence number for each packet sent from the source and also time the packet takes to reach the destination. Then TCP require an acknowledgement for each of the packet sent from the destination which ensures that the packet has reached the destination. By this mechanism the TCP is able to detect packet loss, packet damage,

các nút trong mạng không dây dẫn đến sự phá vỡ liên kết thường xuyên. UDP hỗ trợ giao tiếp không đáng tin cậy trên các mạng không dây. Giao thức TCP được dùng để đạt đến sự truyền thông đáng tin cậy trên các mạng có dây không thể dùng trực tiếp cho các mạng không dây do các đặc tính của nó được thiết kế cho mạng có dây. Do đó, để tạo sự giao tiếp đáng tin cậy trên các mạng không dây, người ta đã đề xuất các biến thể khác nhau của giao thức TCP. Tuy nhiên, tất cả những biến thể này đều có ưu và nhược điểm riêng. Cơ chế đề xuất trình bày về vấn đề nghiên cứu và so sánh TCP Vegas. Hiệu suất được đo theo các tham số như lưu lượng tin, thời gian trễ và tỷ lệ chuyển tiếp gói tin (PDR). Căn cứ vào những nhược điểm, chúng tôi đưa ra biện pháp khắc phục và phân tích những sửa đổi.

[redacted]

[redacted]

duplication of the packet and it can also come to know if the packets have arrived at the destination out of order so that the TCP is able to correct or resend the packets. TCP also uses a mechanism called as congestion window which indicates the sender the maximum number of packets that can be transmitted to the receiver at a time. This number also indicates weather to increase or decrease the value of the congestion window. Different versions of TCP have been proposed to improve the transportation of the packets. Some of them include TCP Tahoe, TCP Reno, TCP NewReno, TCP Sack and TCP Vegas.

II. TCP VEGAS It was proposed by Brakmo[1] as a new TCP version with new. congestion avoidance structure from that in Reno and hence achieves greater throughput than Reno [1].Vegas is innovative strategy of TCP that is included an improved retransmission approach (compared to other TCP versions) that is built on the estimation of RTT as well as new algorithm for detection the congestion within slow start and congestion avoidance. The congestion control algorithm of Vegas is not constantly increasing cwnd within congestion avoidance, but it attempts to determine early congestion via associating the restrained throughput to the expected. Many of research prove that TCP Vegas succeeds to providing higher throughput than other TCP versions, but this is true in homogeneous network but not in the heterogeneous because its incapable to achieve fair bandwidth [1].Vegas

[redacted]

[redacted]

are not depending exclusively on the losses of packet as a mark of congestion occurrence, but it discovers the congestion state before the losses happen. Vegas induced major changes in slow start, retransmission and congestion avoidance. When a duplicate ACK is received, Vegas checks if the recent time of the segment is larger than RTT, then it directly resends the segment and no need to wait three 3 duplicate ACKs. In addition, Vegas able to detecting a multiple losses in packets and its decreases the congestion window only if the retransmitted segment was transmitted after last decrement. Vegas are unlike other TCP variants in the behavior within congestion avoidance phase, because it's not using the segment losses triggers that is congestion happened, but it's determining the congestion via a decreasing in transmission rate by compared it to the predictable rate. In Vegas slow start mechanism, when the Many of research prove that TCP Vegas succeeds to providing higher throughput than other TCP versions, but this is true in homogeneous network but not in the heterogeneous because its incapable to achieve fair bandwidth [1].Vegas are not depending exclusively on the losses of packet as a mark of congestion occurrence, but it discovers the congestion state before the losses happen. Vegas induced major changes in slow start, retransmission and congestion avoidance. When a duplicate ACK is received, Vegas checks if the recent time of the segment is larger than

[redacted]

[redacted]

[redacted]

RTT, then it directly resends the segment and no need to wait three 3 duplicate ACKs. In addition, Vegas able to detecting a multiple losses in packets and its decreases the congestion window only if the retransmitted segment was transmitted after last decrement. Vegas are unlike other TCP variants in the behavior within congestion avoidance phase, because it’s not using the segment losses triggers that is congestion happened, but it’s determining the congestion via a decreasing in transmission rate by compared it to the predictable rate. In Vegas slow start mechanism, if the variance goes above a specific limit it leavs slow start and go in congestion avoidance phase [1].

III. DRAWBACKS OF TCP VEGAS TCP Vegas always uses the estimation of the propagation delay to adjust the window size. But it’s very important for the Vegas to have an accurate measure of the propagation delay because whenever a path failure occurs rerouting takes place and the rerouting of a path may change the delay and results in the degradation of the throughput.

The RTT time gets overestimated due to the rerouting. TCP Vegas makes use of the estimated propagation delay which is used to detect the congestion. Rerouting changes the propogation delay which in turn degrades the performance. When TCP Vegas competes with other TCP variant like TCP Reno in the network, then Vegas is always left behind due to the dominating and aggressive nature of the TCP

[redacted]

[redacted]

[redacted]

Reno and Vegas will never get its fair share of bandwidth due to its modified congestion avoidance mechanism.	[redacted]
IV. IMPROVEMENTS The main goal of our proposed solution is to improve the performance of the existing TCP Vegas with rerouting and proposing a new mechanism to detect the congestion by measuring the RTT and to slowdown the packet loss.	[redacted]
The proposed solution is to make use of the rerouting delay to estimate the RTT which improves the performance by detecting the packet loss.	[redacted]
The solution is that if change in the path occurs rerouting in the network is detected and RTT is computed for detecting the packet loss in Vegas. if RTT is higher than the threshold, then the congestion is detected and the congestion window size is altered which increases the performance in terms of throughput. Hence the congestion detection is done by estimation of the RTT using rerouting propagation delay.	[redacted]
V. SYSTEM ARCHITECTURE A. Steps 1. Network initialization takes place.	[redacted]
Fig: 1 System Architecture 2. 3.	[redacted]
4. Then the source and destination is identified.	[redacted]
5. The modified TCP Vegas is attached.	[redacted]
6. The data forward takes place through the FTP.	[redacted]

7. Calculate the neighbor node for the out range.	[redacted]
8. Then update the congestion window where the changes are taking place.	[redacted]
9. Then the results are compared with that of the existing TCP Vegas and analyzed.	[redacted]
VI. SIMULATION AND RESULTS	[redacted]
A. Simulation Parameters Simulation Parameters Routing Protocol AODV MAC Protocol MAC 802 11 Traffic Type FTP for TCP	[redacted]
Packet Interval 0.002	[redacted]
Packet Size 500 Bytes Data Rate 80 Mbps Simulation Time 100 seconds Terrain Size 1000m* 1000m Number of Nodes 50 B. Network Initialization and Identifying Source and Destination Fig: 2 Network Initialization and identifying source and destination Here in this step the network initialization takes place. The nodes will be initialized along with their parameters. Then the mobility for each node is set using the setdest command. Then the source and destination is found.	[redacted]
C. Finding the Range of each node Fig 3: Finding the range of each node Here in this step all the nodes after their initialization sends hello packets to their neighboring nodes in order to calculate their range. Hence in our case we have considered 82 nodes and each of then send the hello packet to the other neighboring node. D. Packet forwarding from Source to	[redacted]

Destination

Fig 4:Packet forwarding from source and destination Here in this step once the hello packet is sent to the neighboring nodes and the range is calculated then the source and destination is fixed at the initial stage and the source starts sending the packets to the destination via the intermediate nodes which will forward the packets to the required destination.

E. Comparison of TCP Vegas and Modified TCP Vegas

Fig 5: Throughput (Existing TCP Vegas v/s Modified TCP Vegas)

The above snapshot shows the packet delivery ratio, throughput and delay/latency of the existing TCP Vegas considering 80 nodes with 240sec (4min) simulation time. The awk file is run for each of the parameters and input to the awk file is the trace file of the existing TCP Vegas and the trace file of the modified TCP Vegas where we are obtaining in this desertion.

F. TCP Vegas v/s Modified TCP Vegas:

PDR and Latency:

The above snapshot shows the packet delivery ratio, throughput and delay/latency of the existing TCP Vegas considering 80 nodes with 240sec (4min) simulation time.

Generated Packets : 363
Received Packets : 343
Packet Delivery Ratio : 94.4964%
Average Delay : 6.6181914
nikki@ubuntu:~/APPV1.16S awk -f pdrnddelay.awk
vegaspd rnddelay.t r
vegasthroughput.t r
nikki@ubuntu:-/APPVl.16S awk -f

[redacted]

[redacted]

[redacted]

pdrnddelay.awk	[redacted]
Generated Packets : 235 Received Packets : 263 Packet Delivery Ratio : 86.383% Average Delay : 8.6158581_ nikkiftubuntu:-/APPV1.16$	[redacted]
Fig 6: PDR and delay (Modified TCP Vegas)	[redacted]
The above snapshot shows the packet delivery ratio, throughput and delay/latency of the existing TCP Vegas considering 80 nodes with 240sec (4min) simulation time. The awk file is run for each of the parameters and input to the awk file is the trace file of the existing TCP Vegas and the trace file of the modified TCP Vegas where we are obtaining in this desertion. From the above snapshots it can be seen that the performance of the modified TCP Vegas in terms of PDR, throughput and latency is higher when compared to the existing Vegas.	[redacted]
G. Comparison Graphs of TCP Vegas and Modified TCP Vegas: The graphs are plotted for the Vegas and the modified Vegas with respect to the parameters PDR, throughput and latency.	[redacted]
The PDR graph is plotted with PDR as x-axis and node mobility as y-axis. The throughput is plotted with throughput as x-axis and node mobility as y-axis. The latency is plotted with latency as x-axis and node mobility as y-axis. The above graph shows the PDR v/s Delay plotted for the modified TCP Vegas v/s existing TCP Vegas where the	[redacted]

PDR of the modified is more than the existing TCP Vegas at the same time. Similarly the same case for the delay and throughput graphs which shows improvements in their values when compared to existing TCP Vegas.

Fig 7: PDR (Y axis) v/s Time (X axis in sec)

Fig 8: Throughput (Y axis in bits) v/s Time (X axis in sec) From the graphs it is observed that the modified TCP Vegas shows better performance in terms of all the 3 parameters PDR, throughput and latency. The congestion window variance is also plotted for the existing and the modified TCP Vegas which shows that there is The congestion window variance is also plotted for the existing and the modified TCP Vegas which shows that there is decrease and then after sometime increases which is not the case in existing TCP Vegas where the congestion window size decreases and never increases. From the above all graphs we obtained effective increase in the throughput and packet delivery ratio and the delay remains almost same. Hence it can be concluded that our improvements provide with the efficient version of the existing TCP Vegas which is here referred as TCP NewVegas.

The parameters considered were PDR, throughput and delay along with these the congestion window graph is also plotted.

Fig 9: Congestion window (y axis) v/s Time (x axis in sec)

VII. CONCLUSION In this paper

[redacted]

[redacted]

[redacted]

[redacted]

an enhanced version of TCP Vegas is developed to enable reliable and fair communication with improvements in the performance. The performance of the proposed TCP Vegas is analyzed through simulation and compared with that of existing TCP Vegas for different scenarios with respect to parameters like packet delivery ratio, throughput and delay. Hence this dissertation worked towards achieving improved performance than the existing one. For this, study and comparison of different variants of TCP was carried out. Performance of some the variants of TCP like TCP Reno, TCP NewReno and TCP Vegas were critically analysed through simulations configured for different scenarios. Based on the above work inferences are drawn through it, suggestions to enhance functionality of existing TCP Vegas to achieve reliable, fair and improved performance were made. The solution: Comes up with another advantage of detecting congestion at earlier stage and thereby preventing the packet loss. Improves the performance of the existing TCP Vegas with rerouting and proposed a new mechanism to detect the congestion by measuring the RTT and to slowdown the packet loss.

VIII. FUTURE ENHANCEMENT

The modified version of TCP Vegas can be implemented in real time and evaluations can be carried out on real test bed to avoid the unrealistic assumptions in simulation environment.

[redacted]

[redacted]