

TÌM HIỂU GIẢI PHÁP KẾT HỢP CỦA TCP-RENO VÀ VEGAS VỚI GIAO THỨC ĐỊNH TUYẾN ZRP TRÊN MẠNG MANET

Võ Thanh Tú*

Lê Thị Bích Phượng**

Tóm tắt

Sự kết hợp giữa giao thức điều khiển truyền với định tuyến lai trong mạng MANET đóng một vai trò quan trọng trong việc điều khiển truyền dữ liệu từ đầu cuối đến đầu cuối. Đóng góp chính của bài báo này là tìm ra hiệu quả các kịch bản khác nhau trên giao thức định tuyến lai ZRP và các giao thức điều khiển truyền cải tiến (Reno, Vegas) trên mạng MANET. Thông qua thực nghiệm mô phỏng bằng công cụ mô phỏng mạng NS2 (phiên bản 2.34) dựa trên các tham số: tỷ lệ phát gói tin thành công, tỷ lệ rơi gói tin, thông lượng trung bình và độ trễ trung bình, bài báo tiến hành phân tích, xử lý dữ liệu và đánh giá hiệu năng.

Từ khóa: Mạng MANET, TCP-Reno, TCP-Vegas, giao thức định tuyến ZRP.

1. Giới thiệu

Mạng tùy biến không dây di động (MANET) là tập hợp tất cả các điểm di động, cũng là tập hợp các thiết bị định tuyến di động được kết nối bởi các liên kết không dây, các kết nối này tạo thành một cấu trúc liên kết ngẫu nhiên. Nhờ vào lợi thế không dây và tự do di chuyển mà mạng MANET phù hợp với các tình huống khẩn cấp như thiên tai, thảm họa do con người gây ra, các xung đột quân sự, các tình huống y tế khẩn cấp.

Một số nghiên cứu sự kết hợp giữa các giao thức truyền tin và giao thức định tuyến theo nhóm chủ ứng và phản ứng trên mạng MANET cũng đã có những kết quả nhất định. Sự kết hợp của giao thức truyền tin TCP-Vegas với giao thức định tuyến AODV (giao thức phản ứng) trên mạng MANET đã nâng cao được hiệu suất truyền tin với tỷ lệ phát gói tin thành công cao hơn so với khi TCP-Vegas kết hợp với DSDV (giao thức chủ ứng)[1]; hiệu suất truyền tin của TCP-Reno được đánh giá cao hơn khi

kết hợp với OLSR (giao thức chủ ứng)[5]. Tuy nhiên, do tính chất năng động của môi trường mạng MANET, các nút di chuyển vừa theo nhóm, vừa rời rạc thì những sự kết hợp đó có hiệu quả không cao[4]. Vì vậy, khi có sự kết hợp giữa giao thức truyền tin và giao thức định tuyến lai trên mạng MANET phù hợp sẽ góp phần cải thiện hiệu năng sử dụng và giảm chi phí truyền thông. Chính vì vậy cũng đã có nhiều nghiên cứu nhằm cải tiến và phát triển các giao thức định tuyến theo nhóm bằng nghi, thích nghi, định tuyến lai (DSDV, AODV, ZRP)[8], giao thức truyền tin (Tahoe, Reno, New-Reno, Vegas)[1][3], và nghiên cứu sự kết hợp giữa các giao thức này với nhau để góp phần nâng cao hiệu năng trên MANET. Trong bài báo này sẽ tập trung nghiên cứu ảnh hưởng của sự kết hợp giữa giao thức truyền tin TCP-Reno, TCP-Vegas với giao thức định tuyến vùng thuộc nhóm định tuyến lai ZRP, đồng thời thông qua mô phỏng để so sánh, đánh giá hiệu năng trên các kịch bản khác nhau và đưa ra phương án lựa chọn tối ưu góp phần nâng cao hiệu năng trên MANET.

2. Sự kết hợp giữa giao thức TCP với

* PGS TS, Trường Đại học Khoa học Huế

** ThS, Văn phòng Tỉnh Ủy

giao thức định tuyến

2.1. Giao thức TCP-Reno

TCP-Reno là một giao thức giao vận cải tiến của TCP được áp dụng rộng rãi trên Internet ngày nay, nơi mà lưu lượng ngày càng tăng nên việc nghiên cứu kết hợp với các giao thức định tuyến trên mạng diện rộng để điều khiển tắc nghẽn là rất có ý nghĩa. Để điều khiển truyền thông, TCP-Reno sử dụng nhiều cơ chế điều khiển tắc nghẽn riêng biệt như: cơ chế bắt đầu chậm, cơ chế tránh tắc nghẽn, cơ chế phát lại nhanh và cơ chế phục hồi nhanh.

Thuật toán TCP-Reno[2] như sau:

Ban đầu khi một kết nối được thiết lập thì TCP-Reno đặt ngưỡng của cửa sổ phát có độ lớn tối đa và kích thước của cwnd bằng 1 segment.

Tại thời điểm (t+1) thì chỉ xảy ra ở 1 trong các trường hợp sau:

* **Trường hợp 1:** Trạm phát nhận được segment hồi đáp ACK (Truyền nhận thành công):

Nếu $w \leq ssth$ thì:

Sử dụng cơ chế bắt đầu chậm:

Kích thước cửa sổ được cập nhật:

$$w(t+1) = w(t) + 1$$

Ngoài ra (Cho trường hợp $w > ssth$):

Tăng kích thước w theo tuyến tính:

Kích thước cửa sổ được cập nhật:

$$w(t+1) = w(t) + 1w(t)$$

* **Trường hợp 2:** Trạm phát nhận được 3 segment ACK hồi đáp trùng lặp số hiệu:

Sử dụng cơ chế phát lại nhanh và phục hồi nhanh:

Đặt lại ngưỡng: $ssth = w(t)/2$

Kích thước cửa sổ được cập nhật:

$$w(t+1) = ssth$$

* **Trường hợp 3:** Khi phát hiện có segment bị quá thời gian chờ

Đặt lại ngưỡng: $ssth = w(t)/2$

Kích thước cửa sổ được cập nhật:

$$w(t+1) = 1$$

Sử dụng cơ chế bắt đầu chậm.

Trong đó, w: kích thước cửa sổ phát;
ssth: ngưỡng bắt đầu chậm

Như vậy TCP-Reno điều khiển cửa sổ phát bằng cách nhận thông tin từ segment hồi đáp ACK và segment bị hết thời gian chờ (time out). Sự cải tiến của TCP-Reno ở đây là sau khi sử dụng cơ chế phát lại nhanh nó sử dụng cơ chế phục hồi nhanh, TCP-Reno đã tận dụng được đường truyền và cải thiện đáng kể về thông lượng.

2.2. Giao thức TCP-Vegas

TCP-Vegas là một giao thức cải tiến tránh tắc nghẽn, nó được phát triển và kế thừa từ giao thức TCP-Reno. Thay vì tăng tỷ lệ gửi cho đến khi xảy ra mất gói tin thì TCP-Vegas cố gắng để ngăn thiệt hại đó bằng cách giảm tỷ lệ gửi khi nó nhận được tình huống chuẩn bị tắc nghẽn ngay cả khi chưa có dấu hiệu mất các segment[2]. Chính vì vậy TCP-Vegas có thể được xem như là một giao thức TCP cải tiến “chủ ứng” và nó trái ngược hoàn toàn với TCP-Reno là “phản ứng” vì TCP-Reno chỉ xử lý đáp ứng khi xảy ra việc mất các segment. TCP-Vegas sử dụng các cơ chế trong quá trình điều khiển truyền thông như: cơ chế cửa sổ trượt (slide windows), cơ chế bắt đầu chậm (slow start), cơ chế tránh tắc nghẽn, cơ chế phát lại nhanh, phục hồi nhanh và cơ chế điều khiển truyền thông của nó.

Các hành động điều khiển tắc nghẽn chủ ứng của TCP-Vegas dựa trên phép đo RTT (Round Trip Time, thời gian đo gói tin đi trọn một vòng). Trên mỗi RTT, TCP-Vegas tính thông lượng đo thực tế và so sánh nó với thông lượng dự kiến.

- Thông lượng dự kiến được tính toán như sau: $Expected = cwnd/BaseRTT$,

Trong đó, $cwnd$: kích thước cửa sổ tắc nghẽn hiện tại; $BaseRTT$: phép đo RTT quan sát.

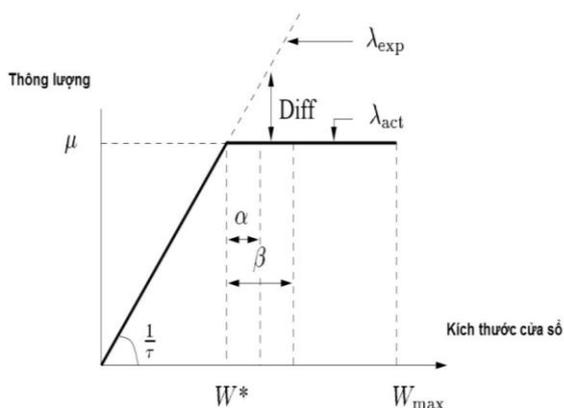
- Thông lượng thực tế được tính như sau: $Actual = RTT_{len}/RTT$,

Trong đó, RTT chính là RTT trung bình của các phân đoạn hồi đáp ACK trong RTT cuối cùng; RTT_{len} là số byte được truyền trong RTT cuối cùng

Sự khác nhau giữa hai phép đo được tính toán trong các phân đoạn RTT cơ sở, giá trị Δ được tính theo công thức sau:

$$\Delta = (cwnd/BaseRTT - RTT_{len}/RTT)/BaseRTT$$

TCP-Vegas thiết lập các giá trị α , β và γ , việc lựa chọn các giá trị này phù hợp cũng sẽ ảnh hưởng lớn đến việc điều khiển truyền thông. Nút nguồn so sánh Δ với γ trong giai đoạn bắt đầu chậm và so sánh với các giá trị α , β trong giai đoạn tránh tắc nghẽn để điều chỉnh kích thước cửa sổ gửi sau mỗi vòng round trip time (RTT) và tính các giá trị, được thể hiện qua cơ chế (Hình 1)



Hình 1. Cơ chế điều khiển cửa sổ truyền tin của TCP-Vegas

Trong cơ chế bắt đầu chậm, khi nhận một ACK mới và Δ nhỏ hơn γ , nút nguồn tăng $cwnd$ thêm 1, ngược lại $cwnd$ giảm một lượng bằng $p\%$, $ssthresh$ được đặt lại bằng $cwnd$ và chuyển sang cơ chế tránh tắc nghẽn. Như vậy, cơ chế bắt đầu chậm được

khởi động tại thời điểm ban đầu hoặc sau sự kiện timeouts và kết thúc khi $cwnd$ lớn hơn ngưỡng $ssthresh$. Kích thước cửa sổ trong cơ chế bắt đầu chậm được mô tả như sau:

$$cwnd = \begin{cases} cwnd + 1 & \text{if } \Delta < \gamma \\ cwnd * (1 - p) & \text{if } \Delta \geq \gamma \end{cases}$$

Trong cơ chế tránh tắc nghẽn, khi nhận một ACK mới, nút nguồn tăng kích thước cửa sổ thêm $1/cwnd$ nếu Δ nhỏ hơn α , giảm một lượng $1/cwnd$ nếu Δ lớn hơn β , và không đổi khi Δ nằm trong khoảng α và β . Kích thước cửa sổ trong cơ chế tránh tắc nghẽn được mô tả như sau:

$$cwnd = \begin{cases} cwnd + \frac{1}{cwnd} & \text{if } \Delta \leq \alpha \\ cwnd & \text{if } \alpha < \Delta < \beta \\ cwnd - \frac{1}{cwnd} & \text{if } \Delta \geq \beta \end{cases}$$

Trong đó α , β và γ thường được thiết lập lần lượt bằng 1, 3 và 1. Trong giai đoạn tránh tắc nghẽn còn có hai cơ chế phát lại nhanh và phục hồi nhanh cho phép giải quyết nhanh chóng tình trạng tắc nghẽn mạng. TCP-Vegas nhanh chóng phát lại các gói tin bị mất khi nhận ba ACKs lập mà không quan tâm đến thời gian hết hạn của gói tin.

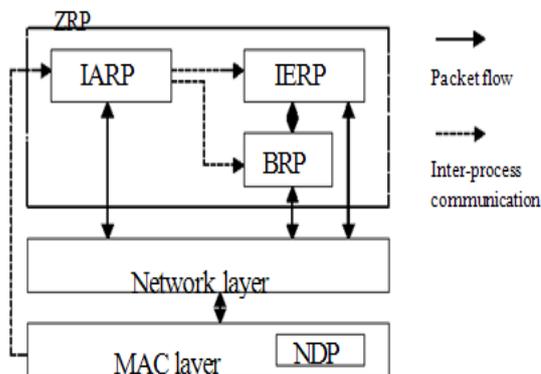
2.3. Giao thức định tuyến vùng (ZRP)

ZRP[7] là giao thức định tuyến lai, nó kết hợp tính năng giữa chủ ứng và phản ứng để mở rộng việc gia tăng kích thước và số lượng nút mạng. ZRP được định nghĩa là một vùng xung quanh các nút, trong vùng sử dụng định tuyến chủ ứng và ngoài vùng sử dụng định tuyến phản ứng. Hiệu năng của giao thức này có thể được tối ưu hóa khi điều chỉnh tham số bán kính vùng phù hợp, việc điều chỉnh này phụ thuộc vào các yếu tố như: tốc độ di chuyển, mật độ nút...

Cấu trúc của ZRP

Để phát hiện các nút láng giềng mới và liên kết thất bại, ZRP dựa vào giao thức khám phá láng giềng (NDP) được cung cấp bởi lớp MAC. NDP truyền thông điệp HELLO cảnh báo đều đặn. Khi tiếp nhận một tín hiệu thì bảng láng giềng được cập nhật. Các láng giềng không nhận được tín hiệu trong một thời gian nhất định nó sẽ được loại bỏ khỏi bảng. Nếu tầng MAC chưa có BDP thì tính năng này sẽ phải được cung cấp bởi IARP.

Mối quan hệ giữa các thành phần được minh họa trong (Hình 2): cập nhật tuyến được khởi động bởi NDP nó thông báo cho IARP khi bảng láng giềng được cập nhật. IARP sử dụng bảng định tuyến của IERP để đáp ứng các truy vấn tuyến. IERP chuyển tiếp truy vấn cho BRP. BRP sử dụng bảng định tuyến của IARP để hướng dẫn truy vấn tuyến đi từ nguồn truy vấn.



Hình 2. Cấu trúc của ZRP

2.4. Sự kết hợp giữa các giao thức

ZRP có lợi thế là một giao thức định tuyến lai, được kết hợp các ưu điểm của giao thức định tuyến chủ ứng (proactive) và giao thức định tuyến phản ứng (reactive). TCP-Vegas có thể được xem như là một giao thức TCP “chủ ứng” khi cố gắng điều chỉnh kích thước cửa sổ phù hợp ngay cả khi chưa có dấu hiệu mất các segment. TCP-Reno có thể được gọi là giao thức TCP “phản ứng” vì chỉ xử lý đáp ứng

khi có các segment tồn thất xảy ra.

2.4.1. TCP-Reno kết hợp với ZRP

ZRP kết hợp hai phương án định tuyến khác nhau hoàn toàn trong cùng một giao thức. Trong vùng định tuyến, thành phần chủ ứng IARP duy trì bảng định tuyến up-to-date, có nhiệm vụ duy trì thông tin định tuyến cho nhiều nút nằm trong vùng định tuyến của nút, bất kỳ một nút trong vùng đều luôn duy trì trong bảng định tuyến của nó thông tin định tuyến đến tất cả các nút khác trong vùng. Thông tin định tuyến được phát broadcast theo một khoảng thời gian quy định để giúp cho bảng định tuyến luôn cập nhật những thông tin mới nhất. Với việc thường xuyên duy trì thông tin định tuyến của IARP đã giúp cho giao thức này hoàn toàn phù hợp với TCP-Reno, bởi TCP-Reno liên tục tăng cửa sổ phát cho đến khi nhận thông tin từ segment hồi đáp ACK và segment bị hết thời gian chờ (time out), TCP-Reno đã tận dụng được đường truyền nên cải thiện thông lượng một cách đáng kể.

TCP-Reno có hiệu năng sử dụng tốt hơn so với các giao thức khác khi kết hợp với giao thức định tuyến OLSR trên MANET (giao thức định tuyến chủ ứng)[5]. Tuy nhiên, vấn đề quan trọng của kết hợp này là việc xác định bán kính vùng phù hợp, bán kính vùng tối ưu phụ thuộc vào một số yếu tố, bao gồm cả tốc độ nút, mật độ nút và chiều dài mạng. Khi thông số này thay đổi thì bán kính vùng phải được điều chỉnh để tối ưu hiệu năng.

2.4.2. TCP-Vegas kết hợp với ZRP

TCP-Vegas được lựa chọn là phù hợp với thành phần IERP của ZRP, thay vì tăng tỷ lệ gửi cho đến khi xảy ra mất gói tin như TCP-Reno thì TCP-Vegas cố gắng để ngăn thiệt hại đó bằng cách giảm tỷ lệ gửi khi nó nhận được tình huống chuẩn bị tắc nghẽn

ngay cả khi chưa có dấu hiệu mất các segment. Khi một nút yêu cầu tuyến, nếu trong vùng thì các tuyến có sẵn ngay lập tức nhưng đối với đích nằm bên ngoài vùng thì lúc này thành phần IERP (có thể được xem là giao thức định tuyến phản ứng của ZRP) được đề nghị tăng khám phá tuyến và dịch vụ bảo trì tuyến dựa vào kết nối cục bộ bởi IARP. Khi một nút yêu cầu một tuyến đến đích, nó phải khởi đầu một quá trình khám phá tuyến để tìm đường đi đến đích (Route discovery). Quá trình khám phá tuyến này hoàn tất khi có một tuyến đã sẵn sàng hoặc đã kiểm tra các tuyến khả thi [2].

Qua tài liệu nghiên cứu [1] đã đánh giá rằng với sự kết hợp của giao thức truyền tin TCP-Vegas với giao thức định tuyến AODV (giao thức phản ứng) trên mạng

MANET đã nâng cao được hiệu suất truyền tin với tỷ lệ phát gói tin thành công cao so với khi TCP-Vegas kết hợp với DSDV (giao thức chủ ứng). Chính nhờ ưu điểm của thành phần “phản ứng” IERP của giao thức định tuyến ZRP và tính năng “chủ ứng” của TCP-Vegas có thể hỗ trợ lẫn nhau, giúp tiết kiệm được tài nguyên mạng, cải thiện được băng thông.

3. Đánh giá kết quả bằng mô phỏng

Trong bài báo này thực hiện mô phỏng hai kịch bản đó là giao thức truyền TCP-Reno kết hợp với giao thức định tuyến ZRP, TCP-Vegas kết hợp với ZRP, sau đó so sánh, đánh giá và đưa ra phương án lựa chọn tối ưu. Các tham số (Bảng 1) thực hiện mô phỏng như sau:

Bảng 1. Các tham số thực hiện mô phỏng

STT	Tham số	Giá trị
1	Giao thức định tuyến	ZRP
2	Giao thức truyền tin	Reno, Vegas
3	Tầng MAC	802.11
4	Kích thước gói tin	512bytes
5	Phạm vi di chuyển của nút	1000mx1000m
6	Số nút	40
7	Thời gian mô phỏng	100s
8	Mô hình di động	Random Waypoint Mobility
9	Bán kính phát sóng của nút	250 ^m
10	Bán kính di chuyển của nút	500 ^m
11	Tốc độ di chuyển tối đa	40m/s
12	Bán kính vùng	2 node
13	Giá trị α, β, γ	lần lượt 1, 3 và 1
14	Phần mềm mô phỏng	NS-2 phiên bản 2.34

Kịch bản mô phỏng được thể hiện (Hình 3):

Từ các kết quả đạt được sau khi thực hiện mô phỏng, tôi sử dụng phần mềm Trace Graph để phân tích kết quả lưu vết qua các file lưu vết và thực hiện phân tích bởi các số liệu sau: Tỷ lệ phát gói tin thành

công; Thông lượng; Tỷ lệ rơi gói tin; Độ trễ trung bình.

3.1. Tỷ lệ phát gói tin thành công

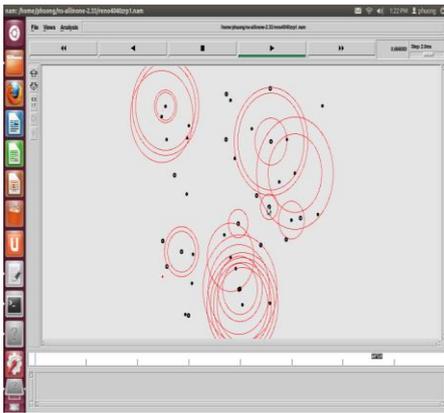
Tỷ lệ phát gói tin thành công (PDR) được tính theo công thức:

$$PDR = (\text{Tổng số gói tin nhận} / \text{Tổng số gói$$

tin gửi) x 100

Biểu đồ (Hình 4) cho thấy tỷ lệ phát gói tin của TCP-Reno đạt cao nhất có lúc lên đến 560 packets/TIL, cao hơn so với TCP-Vegas 540 packets/TIL và cũng có lúc thấp nhất là 190 packets/TIL, thấp hơn so với TCP-Vegas 200 packets/TIL.

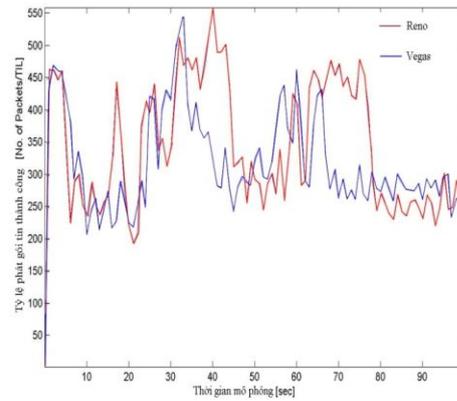
Các số liệu thể hiện TCP-Reno đã chứng tỏ ưu thế tăng kích thước cửa sổ truyền dữ



Hình 3. Kích bản mô phỏng

So sánh giữa hai biểu đồ, TCP-Reno kết hợp với ZRP có mức dao động lớn hơn (Hình 4), tổng số gói tin đạt được là 33941, với tỷ lệ đạt được là 90,70%, thấp hơn so với tỷ lệ của TCP-Vegas. Khi TCP-Vegas

liệu nên tỷ lệ phát gói tin thành công đôi lúc có giá trị đạt được khá cao, tuy nhiên khi các nút mạng liên tục thay đổi và việc định tuyến lại các tuyến đường cũng chính là điểm bất lợi của giao thức này khi tỷ lệ phát gói tin thành công lại thấp hơn so với TCP-Vegas.



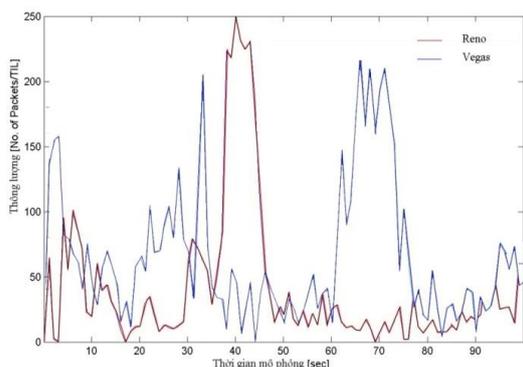
Hình 4. Tỷ lệ gói tin gửi thành công

kết hợp ZRP, biểu đồ được thể hiện có mức dao động nhỏ hơn, với 32428 số lượng gói tin gửi thành công nhưng có tỷ lệ hiệu suất lại đạt cao hơn đó là 91,65% (Bảng 2).

Bảng 2. Tỷ lệ phát gói tin thành công

TT	Giao thức sử dụng	Tổng số gói tin	Số gói tin gửi	Tỷ lệ%
1	TCP-Reno	37420	33941	90,70%
2	TCP-Vegas	35381	32428	91,65%

3.2. Thông lượng trung bình



Hình 5. Thông lượng đạt được của các giao thức

Thông lượng trung bình là số lượng gói tin gửi thành công/giây đến các nút đích trong suốt thời gian mô phỏng.

Qua biểu đồ (Hình 5) của cả hai kịch bản mô phỏng ta thấy thông lượng của cả hai giao thức đều có mức giao động lớn.

Việc tăng kích thước cửa sổ liên tục của giao thức TCP-Reno đã ảnh hưởng rất lớn đến việc chiếm giữ đường truyền gây ra thường xuyên tắc nghẽn mạng, có ít nhất 3 lần thông lượng trở về ở mức 0 tại thời điểm thứ 3s, 17s, 69s (xảy ra hiện tượng tắc nghẽn mạng).

Thông lượng trung bình của TCP-Vegas là 196 packets/s, (Bảng 3), chứng tỏ TCP-Vegas khó có cơ hội tăng kích thước cửa sổ nhờ vào các tham số alpha, beta phù hợp. Biểu đồ (Hình 5) cũng thể hiện hiện tượng tắc nghẽn mạng chỉ xảy ra một lần tại thời điểm 44s.

Bảng 3. Thông lượng đạt được của các giao thức

TT	Giao thức sử dụng	Thông lượng
1	TCP-Reno	225
2	TCP-Vegas	196

1	TCP-Reno	225
2	TCP-Vegas	196

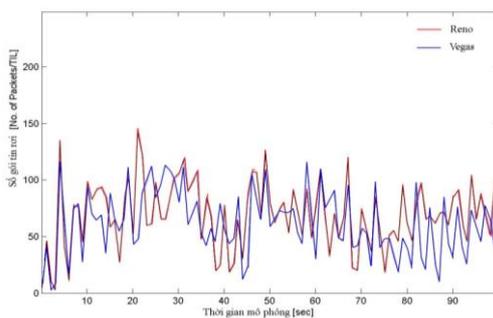
3.3. Tỷ lệ rơi gói tin

Tỷ lệ rơi gói tin (PLR) được tính theo công thức:

$$PLR = \frac{\text{(tổng số gói tin rơi)}}{\text{(tổng số gói tin gửi)}} \times 100$$

Biểu đồ (Hình 6) thể hiện tỷ lệ gói tin rơi của các giao thức TCP-Reno và Vegas khi kết hợp với giao thức định tuyến ZRP.

So sánh các kết quả cho thấy tỷ lệ gói tin rơi của TCP-Reno nhiều hơn so với TCP-Vegas thể hiện ở số liệu 18,63% so với 17,98% (bảng 4). Số liệu trên chứng tỏ rằng TCP-Vegas có thể ngăn ngừa các gói tin bị mất trong quá trình truyền thông nhờ vào việc theo dõi các RTT để điều chỉnh kích thước cửa sổ tăng giảm phù hợp. Trái lại, TCP-Reno lại liên tục tăng kích thước cửa sổ của nó cho đến khi phát hiện dấu hiệu tắc nghẽn mạng, chính điều này đã làm cho tỷ lệ gói tin rơi cao hơn so với TCP-Vegas.



Hình 6. Tỷ lệ gói tin rơi

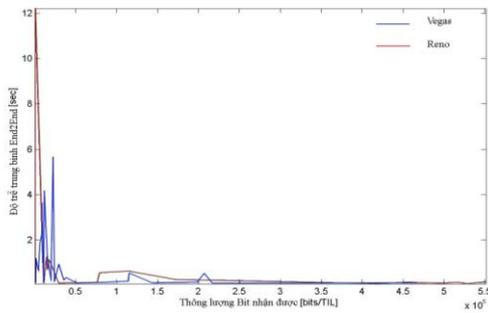
Bảng 4. Tỷ lệ gói tin rơi

TT	Giao thức sử dụng	Số gói tin chung	Số gói tin rơi	Tỷ lệ
1	TCP-Reno	37420	6974	18,63%
2	TCP-Vegas	35381	6365	17,98%

3.4. Độ trễ trung bình End-to-End

Độ trễ trung bình End-to-End là thời gian trung bình cần thiết để một gói tin

được truyền thành công từ nút nguồn đến nút đích.



Hình 7. Độ trễ trung bình End-to-End

Độ trễ trung bình chịu ảnh hưởng lớn bởi độ dài của tuyến đường truyền tin cũng như các bộ đệm tại các nút, nếu gói tin đến bộ đệm nhiều thì thời gian chờ tại các bộ đệm sẽ tăng lên. Với cơ chế của mình TCP-Vegas cố gắng duy trì một lượng nhỏ các gói tin trong hàng đợi thì TCP-Reno đã chiếm một lượng lớn các gói tin trong bộ đệm.

Ở biểu đồ (hình 7) độ trễ của TCP-Reno tăng rất cao, có lúc lên đến 12s, chứng tỏ việc tăng liên tục kích thước cửa sổ đã làm cho băng thông trên đường truyền tăng cao, kéo theo độ trễ trung bình của TCP-Reno cao: 0.1757s. Ngược lại TCP-Vegas có độ trễ trung bình rất thấp: 0.0914s (bảng 5) nhờ vào việc điều khiển kích thước cửa sổ tăng giảm phù hợp.

Bảng 5. Độ trễ End-to-End của các giao thức

TT	Giao thức sử dụng	Độ trễ trung bình
1	TCP-Reno	0.1757
2	TCP-Vegas	0.0914

4. Kết luận

Như vậy, bài báo này đã tìm hiểu giải pháp kết hợp của TCP-Reno và TCP-Vegas với giao thức định tuyến ZRP, qua hai kịch bản mô phỏng cho thấy TCP-Vegas kết hợp với giao thức định tuyến ZRP có hiệu suất tốt hơn nhờ vào cơ chế ước lượng băng thông và các giá trị α , β , γ để đo lường tắc nghẽn mạng và điều chỉnh kích thước cửa sổ phù hợp, thể hiện qua các số liệu như: có tỷ lệ phát gói tin thành công, tỷ lệ rơi gói tin thấp và độ trễ trung bình thấp hơn so với TCP-Reno. Tuy nhiên TCP-Vegas có một nhược điểm là việc tận dụng thông lượng trên đường truyền chưa được tối ưu, vẫn còn ở mức thấp hơn so với TCP-Reno.

Tương lai chúng tôi sẽ tiếp tục tìm hiểu và mô phỏng việc kết hợp thêm giữa các giao thức truyền tin cải tiến khác như TCP-Vegas W, TCP-Vegas A... với giao thức định tuyến ZRP nhằm tìm ra kết hợp tối ưu với mục đích nâng cao hiệu năng sử dụng trong MANET □

TÀI LIỆU THAM KHẢO

- [1] Mạc Quốc Bảo (2014), “Nghiên cứu một số giải pháp nâng cao hiệu năng của TCP-Reno và Vegas kết hợp giao thức định tuyến AODV trên mạng MANET”, Luận văn Thạc sĩ Công nghệ thông tin, Trường Đại học Quy Nhơn.
- [2] Võ Thanh Tú (2012), “Mạng và truyền dữ liệu nâng cao”, Nxb Đại học Huế.
- [3] Alaa Seddik-Ghaleb, Yacine Ghamri-Doudane, Sidi-Mohammed Senouci (2006) “Effect of Ad Hoc Routing Protocols on TCP Performance within MANET”, Sensor and Ad Hoc Communications and Networks, IEEE, pp.866 – 873.
- [4] Avni Khatkar, Yudhvir Singh (2012), “Performance Evaluation of Hybrid Routing Protocols in Mobile Adhoc Networks”, Advanced Computing & Communication Technologies, pp. 542 – 545.
- [5] Dongkyun Kim, Juan-Carlos Cano and P. Manzoni (2006), “A comparison of

- the performance of TCP-Reno and TCP-Vegas over MANET*”, Wireless Communication Systems, IEEE, pp. 495 - 499.
- [6] Erlend Larsen (2012), “*TCP in MANET – challenges and Solutions*”, Norwegian Defence Research Establishment (FFI).
- [7] Jitendranath Mungara, M.N. SreeRangaRaju (2011), “*Optimized ZRP for MANET and its Applications*”, International Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, No. 3, pp. 84-94.
- [8] Savita Gandhi SMIEEE1, Nirbhay Chaubey MIEEE, Naren Tada, Srushti Trivedi (2012), “*Scenario-based Performance Comparison of Reactive, Proactive & Hybrid Protocols in MANET*”, Computer Communication and Informatics (ICCCI), IEEE, pp. 1-5.

Abstract

Exploring the solution of TCP-Reno and TCP-Vegas with ZRP over MANET

The combination of both hybrid routing protocol and transmission control protocol (TCP) plays an important role in end-to-end data packet transmission. The main contribution of this article is to find the effect of different scenarios on hybrid routing protocols and the TCP variants (Reno, Vegas) over MANET. According to the simulation results by simulation tool NS2 (version 2.34), Packet delivery, Drop ratio, Average throughput, Average end to end delay, the article conducts data processing, analysing and performance evaluation.

Keywords: *MANET, TCP-Reno, TCP-Vegas, ZRP routing protocol*