JOURNAL OF SCIENCE OF HNUE DOI: 10.18173/2354-1075.2015-0059
Educational Sci., 2015, Vol. 60, No. 7A, pp. 121-130
This paper is available online at http://stdb.hnue.edu.vn

GIẢI PHÁP CHỐNG TẤN CÔNG BLACKHOLE TRONG MẠNG MANET

[1]Nguyễn Phúc Hải, [2]Nguyễn Thị Quỳnh Hoa và [2]Nguyễn Thế Lộc
[1]*Phân viện khu vực Tây Nguyên, Học viện Hành chính Quốc gia*
[2]*Khoa Công nghệ Thông tin, Trường Đại học Sư phạm Hà Nội*

Tóm tắt. MANET (Mobile Ad-hoc Network) là mạng kết nối các máy trạm di chuyển một cách ngẫu nhiên, không phụ thuộc vào cơ sở hạ tầng mạng cố định nên chi phí hoạt động thấp, triển khai nhanh và có tính di động cao. Vấn đề đặt ra cho mạng MANET là lỗ hổng trong giao thức định tuyến khiến cho mạng trở thành mục tiêu của một số dạng tấn công đặc thù. Bài báo này nghiên cứu ảnh hưởng của kiểu tấn công lỗ đen (Blackhole Attack) đối với mạng MANET thông qua việc đánh giá các thông số như tỉ lệ mất gói tin trong mạng. Bài báo cũng đề xuất một giải pháp phòng chống tấn công lỗ đen trên giao thức AODV và trình bày những kết quả thực nghiệm để kiểm chứng.
Từ khóa: Mạng adhoc, giao thức AODV, tấn công lỗ đen, mạng MANET, phòng chống tấn công lỗ đen.

1. Mở đầu

Mạng không dây di động (MANET) là mạng mà trong đó bao gồm các máy trạm tự trị, tự quản lý mà không có bất kỳ cơ sở hạ tầng nào. Mỗi trạm (sau đây gọi là nút) đồng thời đóng vai trò của một router có khả năng tìm kiếm, duy trì và định tuyến các gói dữ liệu cho các nút nằm trong vùng phủ sóng của nó. Tất cả các nút đều ngang hàng và không có nút nào đóng vai trò máy chủ trung tâm. Các nút có thể gia nhập hay rời bỏ mạng bất kể khi nào do đó tạo ra sự thay đổi topology một cách liên tục. MANET phù hợp cho việc sử dụng trong tình huống mà mạng có dây hoặc mạng không dây dựa trên cơ sở hạ tầng không thể truy cập, quá tải, hư hỏng hoặc bị phá hủy như trong các trường hợp khẩn cấp hoặc các nhiệm vụ cứu hộ, cứu trợ thiên tai và chiến thuật trên chiến trường, hoặc thông thường như các hội nghị trực tuyến, trong nghiên cứu mạng cảm biến. Bên cạnh những đặc tính tối ưu của mạng MANET thì vấn đề bảo mật cũng đóng vai trò quan trọng. Giao thức định tuyến không có cơ chế xác định độ chính xác của gói tin nên các nút độc hại trong mạng có thể tấn công bằng cách làm sai lệch hướng gửi dữ liệu của các nút trong mạng và lấy cắp các gói tin.

Ngày nhận bài: 15/7/2015 Ngày nhận đăng: 01/11/2015
Liên hệ: Nguyễn Phúc Hải, e-mail: hainp@napa.vn/ hoantq@hnue.edu.vn/ locnt@hnue.edu.vn

2. Nội dung nghiên cứu

2.1. Giao thức AODV

Giao thức định tuyến AODV (Ad hoc On-demand Distance Vector routing) định tuyến theo nhiều bước, nó tìm đường đi chi khi có nhu cầu truyền dữ liệu, sử dụng thông điệp yêu cầu tuyến RREQ và thông điệp trả lời tuyến RREP [3].

2.1.1. Quá trình tìm đường đi

Quá trình tìm đường được khởi tạo khi một nút cần truyền thông tin với nút khác mà liên kết giữa chúng không có sẵn trong bảng định tuyến của nút nguồn. Mỗi nút duy trì hai bộ đếm riêng biệt: một số thứ tự nút và một ID quảng bá. Nút nguồn bắt đầu tìm đường bằng việc quảng bá một gói tin yêu cầu tuyến RREQ (Route ReQuest) tới các nút lân cận của nó [1]. Gói tin RREQ chứa một số trường sau:

<source_addr source sequence# Broadcast ID dest_addr dest_sequence# hopcnt>

<địa chỉ nguồn, thứ tự nguồn#, ID quảng bá, địa chỉ đích, thứ tự đích#, hop count>

Cặp < *source_addr, broadcast_id* > xác định duy nhất một RREQ, *broadcast_id* được tăng lên mỗi khi nguồn phát ra một gói tin RREQ mới. Mỗi nút lân cận sẽ gửi lại một bản tin trả lời RREP (Route Reply) nếu thỏa mãn yêu cầu của gói RREQ hoặc sẽ quảng bá gói tin RREQ sau khi tăng giá trị của trường *hopcnt* (hop count: chiều dài tuyến). Một nút có thể nhận nhiều bản sao của cùng một gói tin RREQ từ các nút khác nhau. Khi một nút trung gian nhận được một RREQ, nếu trước đó nó đã nhận được một RREQ với cùng *broadcast_id* và địa chỉ đích thì nó sẽ loại bỏ gói tin RREQ đến sau. Nếu một nút không thỏa mãn RREQ, nó giữ lại các thông tin cần thiết để thiết lập đường chuyển tiếp đảo chiều cho gói tin RREP [1]. Các thông tin này bao gồm: địa chỉ IP đích, địa chỉ IP nguồn, ID quảng bá, thời gian sống cho đường đảo chiều, số thứ tự của nút nguồn. Các thành phần trên được mô hình hóa trong công cụ NS2 bởi cấu trúc dữ liệu sau:

```
struct hdr_aodv_request {
u_int8_t rq_type; // Packet Type
u_int8_t reserved[1];
u_int8_t rq_hop_count; // Hop Count
u_int32_t rq_bcast_id; // Broadcast ID
nsaddr_t rq_dst; // Destination IP Address
u_int32_t rq_dst_seqno; // Destination Sequence Number- DSN
nsaddr_t rq_src; // Source IP Address
u_int32_t rq_src_seqno; // Source Sequence Number
double rq_timestamp; // when REQUEST sent;
```

2.1.2. Thiết lập tuyến đường chuyển tiếp

Khi một nút trung gian nhận được bản tin RREQ, nếu trong bảng định tuyến nó có tuyến đường đến đích mong muốn, nó sẽ kiểm tra tính khả dụng của tuyến đường đó bằng cách so sánh Destination Sequence Number (DSN-số thứ tự đích) tương ứng với tuyến đường có trong bảng định tuyến của nó với DSN trong gói tin RREQ mà nó nhận được. Nếu DSN của RREQ lớn hơn DSN được lưu giữ bởi nút trung gian thì nút đó không được sử dụng thông tin về tuyến đường đến

đích trong bảng định tuyến để trả lời cho gói tin RREQ. Thay vào đó nó sẽ phải tiếp tục quảng bá gói tin RREQ. Nút trung gian chỉ có thể phản hồi khi nó biết một tuyến đường đến đích với DSN lớn hơn hoặc bằng DSN chứa trong RREQ. Khi nút trung gian có một tuyến đến đích khả dụng và gói tin RREQ chưa được xử lý trước đó, thì nó sẽ trả lời bằng một gói tin RREP theo đường truyền đơn hướng ngược lại nút mà nó đã nhận RREQ trước đó [1]. Một RREP chứa các thông tin sau:

<source_addr, dest_addr, dest_sequence#, hop_cnt, lifetime>

<địa chỉ nguồn, địa chỉ đích, số thứ tự đích# hop count, thời gian sống>

Khi RREQ được quảng bá đến một nút có thể cung cấp tuyến đường đến đích mong muốn, một đường đảo chiều tới nút nguồn của RREQ sẽ được thiết lập. Mỗi nút dọc theo tuyến đảo chiều mà RREP quay lại nguồn sẽ đóng vai trò là một nút chuyển tiếp, các nút này sẽ cập nhật lại toàn bộ thông tin về tuyến đường giữa nguồn và đích đồng thời ghi lại DSN gần nhất của nút đích. Tuyến đường mà theo đó RREP quay lại nguồn được gọi là đường chuyển tiếp. Các nút không nằm trên tuyến chuyển tiếp sẽ xóa các tuyến ngược chiều mà chúng đã thiết lập trước đó khi khoảng thời gian ACTIVE_ROUTE_TIMEOUT kết thúc (3000 mili giây) [1]. Các nút trung gian sẽ truyền gói tin RREP đầu tiên về nguồn ngay khi nó nhận được. Nếu nó nhận nhiều gói tin RREP, việc cập nhật thông tin định tuyến và truyền bản tin RREP sau chỉ được thực hiện khi RREP sau có DSN lớn hơn DSN trong RREP trước hoặc bằng nhưng tuyến đường mới có số hop count nhỏ hơn. Việc tăng số lượng RREP gửi về nguồn đảm bảo thông tin được cập nhật nhanh và nhiều nhất. Nút nguồn thực hiện truyền dữ liệu sớm nhất có thể khi nó hoàn thành quá trình cập nhật thông tin từ RREP đầu tiên mà nó nhận được và có thể tiếp tục cập nhật thông tin định tuyến nếu nó nhận được một tuyến đường tốt hơn [1]. Các thành phần trên được mô hình hóa trong công cụ NS2 bởi cấu trúc dữ liệu sau:

```
struct hdr_aodv_reply {
u_int8_t rp_type; // Packet Type
u_int8_t reserved[1];
u_int8_t rp_hop_count; // Hop Count
nsaddr_t rp_dst; // Destination IP Address
u_int32_t rp_dst_seqno; // DSN
nsaddr_t rp_src; // Source IP Address
double rp_lifetime;  // Lifetime
double rp_timestamp; // when corresponding REQ sent;
sendReply(rq→rq_src, // IP Destination
1, // Hop Count
index, // Dest IP Address
seqno, // Dest Sequence Num
MY_ROUTE_TIMEOUT, // Lifetime
rq→rq_timestamp); // timestamp
```

2.2. Tấn công lỗ đen và phòng chống tấn công lỗ đen

2.2.1. Tấn công lỗ đen

Giao thức AODV có thể bị tấn công làm sai lệch thông tin đường đi để chuyển hướng đường đi và bắt đầu các cuộc tấn công khác. Sự sai lệch của bất cứ trường nào trong gói tin định tuyến có thể khiến giao thức AODV gặp sự cố. Các trường dễ bị phá hoại trong thông điệp định tuyến của AODV bao gồm *Destination Sequence Number (DSN), HopCount, Broadcast ID*... Trong giao thức AODV các nút nguồn khi nhận các gói tin định tuyến RREP không có cơ chế xác định độ chính xác của gói tin. Nút nguồn chỉ kiểm tra độ mới (số DSN) và độ dài đường đi (HopCount) tới nút đích. Từ đó dẫn đến việc một nút trong mạng có thể gửi một gói tin RREP với thông tin sai lệch đến nút nguồn làm cho nút nguồn gửi dữ liệu đến nút khác nút đích.

Để thực hiện một cuộc tấn công lỗ đen trong giao thức AODV, nút lỗ đen (tức là nút khởi xướng cuộc tấn công) chờ gói tin RREQ gửi từ các nút láng giềng của nó. Khi nhận được gói RREQ, nó lập tức gửi trả gói tin RREP trong đó thiết lập giá trị DSN cao nhất (số thứ tự cao nhất của giao thức AODV là 4294967295, số nguyên 32 bit lớn nhất) và giá trị Hop count nhỏ nhất (bằng 1) trước khi các nút hợp lệ khác kịp gửi thông điệp trả lời tuyến. Sau đó mọi dữ liệu từ nút nguồn sẽ bị chuyển đến nút lỗ đen và bị hủy tại đó thay vì được chuyển tới nút đích [4], nút lỗ đen sẽ hút mọi dữ liệu tương tự như lỗ đen trong vũ trụ, vì vậy mà kiểu tấn công này có tên gọi là tấn công lỗ đen.

2.2.2. Phòng chống tấn công lỗ đen

Để phòng chống tấn công lỗ đen, một số tác giả đã đề xuất giải pháp (sau đây gọi là *Giải pháp đối chứng*) bỏ qua gói tin RREP đầu tiên mà nút nguồn nhận được [2, 4, 5]. Giải pháp này dựa trên đặc điểm hoạt động của nút lỗ đen là thông điệp RREP mà nó gửi tới nút nguồn thường tới đích đầu tiên do nút lỗ đen không mất thời gian tra cứu bảng định tuyến như những nút khác. Giải pháp này có nhược điểm là trong trường hợp nút đích nằm gần nút nguồn hơn nút lỗ đen thì gói tin RREP đầu tiên mà nút nguồn nhận được là gói tin tốt nhưng vẫn bị loại bỏ một cách oan uổng đồng thời gói tin RREP của nút lỗ đen tới sau nên lọt lưới, dẫn đến hậu quả là nút lỗ đen sẽ hút hết luồng dữ liệu. Để khắc phục nhược điểm trên, bài báo này đề xuất 2 giải pháp phòng chống tấn công lỗ đen như sau:

- *Giải pháp thứ nhất*: Bỏ qua những gói tin RREP có DSN lớn hơn một ngưỡng định sẵn. Khi tấn công, nút lỗ đen thường gửi gói tin RREP với DSN bằng giá trị cao nhất (4294967295) nên chúng tôi đề xuất loại bỏ những gói tin có số DSN lớn hơn một giá trị ngưỡng (trong thực nghiệm chúng tôi chọn giá trị ngưỡng là 4294960000). Khi nút nguồn nhận được một gói tin RREP, nếu DSN của gói tin đó lớn hơn ngưỡng định sẵn thì gói tin sẽ bị nút nguồn loại bỏ. Tuy nhiên khi tấn công, nút lỗ đen có thể sử dụng số DSN có giá trị nhỏ hơn ngưỡng định sẵn. Khi đó phương pháp phòng chống trở nên không có tác dụng. Hay khi nút tấn công thay đổi DSN một cách thường xuyên, khi đó chúng ta sẽ khó để xác định được giá trị ngưỡng.

- *Giải pháp thứ hai*: Bỏ qua những gói tin RREP có DSN quá lớn so với giá trị DSN lớn nhất mà nút nguồn theo dõi được. Gọi giá trị DSN lớn nhất mà nút nguồn biết (cho tới thời điểm đó) là x, vậy nút nguồn sẽ bỏ qua mọi gói tin RREP có DSN > x + $\triangle$ trong đó độ lệch $\triangle$ là một giá trị xác định trước, trong thực nghiệm chúng tôi chọn $\triangle$ = 600.

Cả hai phương pháp đề xuất đều kém hiệu quả trong trường hợp nút nguồn không có số DSN của nút đích trong bảng định tuyến (khi đó được khởi tạo bằng 0), khi nhận được gói tin

RREP đầu tiên, nút nguồn không tiến hành so sánh mà ngay lập tức thêm đường đi vào bảng định tuyến. Nếu nút Blackhole nằm gần nút nguồn hơn nút đích hay nút trung gian có đường đi tốt đến đích trong trường hợp này thì nút nguồn sẽ bị tấn công vì gói RREP đầu tiên do nút blackhole gửi đến. Nếu nút blackhole nằm xa hơn nút đích hoặc nút trung gian thì gói tin RREP đầu tiên sẽ được thêm vào bảng định tuyến của nút nguồn là gói tin của các nút trung gian hoặc nút đích. Sau đó từ gói tin thứ 2 trở đi mới bắt đầu tiến hành xét xem DSN > x + △ hay không khi đó nút blackhole không thể tấn công thay đổi được đường đi (cập nhật lại tuyến đường của bảng định tuyến). Một nút chỉ tăng số SN lên 1 đơn vị khi phát một gói RREQ hoặc một gói RREP[3]. Chúng tôi đặt △ = 600 trong trường hợp này vì khi thực nghiệm số lượng kết nối là 18 với số nút là 20. Trong thực tế thực nghiệm nhận thấy một nút trung bình phát sinh tổng cộng khoảng 600 gói tin RREQ và RREP.

2.3. Thực nghiệm

2.3.1. Chỉnh sửa nhân NS-2 để tấn công lỗ đen

Công cụ NS2 cho phép thực hiện các kịch bản mô phỏng mạng MANET với nhiều giao thức định tuyến trong đó có giao thức AODV, tuy nhiên để thực hiện một cuộc tấn công lỗ đen trong giao thức AODV thì phải chỉnh sửa lại phần nhân (kernel) của NS2. Để mô phỏng cuộc tấn công chúng tôi sửa nhân của NS2 để:

- Nút lỗ đen luôn luôn là nút gửi đường đi đầu tiên và tốt nhất đến nút nguồn.
- Nút lỗ đen xóa đi các gói tin mà nó thu hút được.

Đầu tiên chúng tôi sửa lại giao thức AODV và tạo ra giao thức mới BlackholeAODV trong đó các nút gửi cho nhau các gói tin theo khuôn dạng giống như giao thức AODV. Chúng tôi thay đổi tất cả các lớp, chức năng, cấu trúc, biến và hằng số của tất cả các file trong thư mục Blackholeaodv trừ struct AODV thuộc aodv_packet.h. Sau đó để thêm giao thức định tuyến mới BlackholeAODV vào NS2 phải sửa đổi 2 tệp. Tệp đầu tiên được sửa đổi là " tcl lib ns-lib.tcl", tệp thứ hai được sửa đổi là "makefile" trong thư mục gốc của "ns-2.35". Sau khi sửa đổi lại 2 tệp trên, chúng tôi biên dịch NS-2 một lần nữa để tạo ra các tệp thực thi [4]. Sau các bước trên ta đã có một giao thức mới với tên là BlackholeAODV hoạt động tương tự như giao thức AODV. Để nút lỗ đen xóa các gói tin thu hút được thì cần sửa lại tập tin Blackholeaodv.cc của giao thức BlackholeAODV. Để tấn công khi nhận được gói tin RREQ, nút lỗ đen ngay lập tức gửi gói tin RREP như thể nó có đường tốt nhất và mới nhất tới đích. Trong Blackholeaodv được chỉnh sửa như sau:

sendReply(rq→rq_src, *// IP Destination*

1, *// Hop Count*

index, *// Dest IP Address*

4294967295, *// Highest Dest Sequence Num*

MY_ROUTE_TIMEOUT, *// Lifetime*

rq→rq_timestamp); *// timestamp*

2.3.2. Chỉnh sửa nhân NS-2 để phòng chống tấn công lỗ đen

- Giải pháp đối chứng

Chúng tôi sao chép lại giao thức AODV và sửa tên thư mục thành idsAODV cũng như toàn bộ tập tin trong thư mục giống như đã làm để tạo ra giao thức BlackholeAODV như trên. Sau khi

tạo ra được giao thức idsAODV để phục vụ quá trình thực hiện giải pháp đối chứng, chúng tôi tiến hành sửa đổi tập tin idsaodv.cc để giao thức bỏ qua tập tin RREP đầu tiên [4].

```
idsAODV::recvReply(Packet *p) {
idsBroadcastRREP * r = rrep_lookup(rp→rp_dst);
if(ih→daddr() == index) {
if (r == NULL) {
count = 0;
rrep_insert(rp→rp_dst);
} else {r→count ++;
count = r→count;}
UPDATE ROUTE TABLE
} else {Forward(p) ;}}
```

- Giải pháp đề xuất

Khi nút nguồn gửi dữ liệu mà trong bảng định tuyến không có đường đi đến nút đích nút nguồn sẽ gửi một một gói tin RREQ mà trong trong đó số DSN được gán bằng 0 [5]. Trong trường hợp đó khi nhận được gói tin RREP đầu tiên nút nguồn ngay lập tức thêm vào bảng định tuyến và gửi dữ liệu đi.

```
rt = rtable.rt_lookup(rp→rp_dst);
if(rt == 0) {
 rt = rtable.rt_add(rp→rp_dst);
 }
```

- Giải pháp thứ nhất

Nếu trong quá trình gửi dữ liệu đường đi thay đổi do nhiều nguyên nhân. Khi đó tuyến đường sẽ được cập nhật lại. Khi cập nhật nút nguồn sẽ tiến hành so sánh số DSN trong gói tin RREP. Nếu "số DSN trong gói RREP lớn hơn số DSN trong bảng định tuyến và số DSN đó phải nhỏ hơn ngưỡng định sẵn là 4294960000" hoặc "số DSN trong gói RREP bằng số DSN trong bảng định tuyến và hopcount trong gói RREP bằng hopcount trong bảng định tuyến và số DSN đó phải nhỏ hơn ngưỡng định sẵn" thì mới tiến hành cập nhật bảng định tuyến. Đoạn mã dưới đây thể hiện sự so sánh trên.

```
if ( ((rt→rt_seqno < rp→rp_dst_seqno) && (rp→rp_dst_seqno < 4294960000))
|| ((rt→rt_seqno == rp→rp_dst_seqno) && (rt→rt_hops > rp→rp_hop_count) &&
(rp→rp_dst_seqno < 4294960000)) ) {
```

Sau khi tất cả thay đổi kết thúc, tiến hành biên dịch lại tất cả các tệp trong phần mềm NS2 và tiến hành thực nghiệm.

- Giải pháp thứ hai

Nếu trong quá trình gửi dữ liệu đường đi thay đổi do nhiều nguyên nhân. Khi đó tuyến đường sẽ được cập nhật lại. Khi cập nhật nút nguồn sẽ tiến hành so sánh số DSN trong gói tin RREP. Nếu "số DSN trong gói RREP lớn hơn số DSN trong bảng định tuyến và số DSN đó phải nhỏ hơn số DSN trong bảng định truyến + 600 (số nguyên 32 bit)" hoặc "số DSN trong gói RREP bằng số DSN trong bảng định tuyến và hopcount trong gói RREP bằng hopcount trong bảng định

tuyến và số DSN đó phải nhỏ hơn hơn số DSN trong bảng định truyến + 600 (số nguyên 32 bit)" thì mới tiến hành cập nhật bảng định tuyến. Trong luận văn của chúng tôi chọn độ lệch là 600 vì khi tiến hành thống kê tổng các gói tin RREP và RREQ phát sinh trong quá trình thực nghiệm kịch bản của luận văn trung bình một nút nhận và gửi khoảng 600 gói tin RREP và RREQ. Đoạn mã dưới đây thể hiện sự so sánh nêu trên.

if (((rt→rt_seqno < rp→rp_dst_seqno) && ((rp→rp_dst_seqno - rt→rt_seqno) < 600)) || ((rt→rt_seqno == rp→rp_dst_seqno) && (rt→rt_hops > rp→rp_hop_count) && ((rp→rp_dst_seqno - rt→rt_seqno) < 600))) {

Sau khi tất cả thay đổi kết thúc, tiến hành biên dịch lại tất cả các tệp trong phần mềm NS2 và tiến hành thực nghiệm.

2.3.3. Tạo kịch bản TCL random waypoint

Trong mô phỏng mạng Ad-hoc có hai loại mô hình di chuyển được sử dụng cho mạng không dây là mô hình Vết (trace-based model) và mô hình Tổng hợp (synthetic model) [4]. Mô hình Vết cung cấp cho ta thông tin chính xác, đặc biệt là khi nó có liên quan tới nhiều bên tham gia và có thời gian đủ dài. Tuy nhiên mô hình Tổng hợp là phù hợp hơn đối với mạng di động ngày nay, nó thể hiện hành vi của các nút di động bằng cách thống kê. Mỗi nút sẽ được biểu diễn thông qua một giải thuật nhằm ngẫu nhiên hóa quá trình di chuyển. Hai mô hình tiêu biểu mô phỏng các mạng Ad-hoc là: Random Waypoint và Random Walk. Tại mô hình này, ban đầu mỗi nút có một vị trí ngẫu nhiên trong khu vực mô phỏng và ở tại đó một khoảng thời gian tạm dừng. Khi hết quãng thời gian tạm dừng, nút chọn cho mình một đích ngẫu nhiên trong khu vực mô phỏng và chuyển động với tốc độ phân bố đồng đều giữa $[speed_min, speed_max]$. Khi tới vị trí mới nút dừng một khoảng thời gian trong khoảng $[P_min, P_max]$ và sau đó tiếp tục lại quá trình. Trong phần thực nghiệm này chúng tôi tạo 10 kịch bản có các thông số như trong bảng dưới đây, số nút lỗ đen trong mỗi kịch bản là 2.

Bảng 1. Cấu hình mạng thực nghiệm

Thông số	Giá trị
Cấu hình chung	
Khu vực địa lý	0,75Km x 0,75Km
Tổng số nút	20 nút
Vùng thu phát sóng	250m
Thời gian chạy thử	600 giây
Cấu hình di chuyển	
Tốc độ di chuyển nhanh nhất	20 m/s = 72 km/h
Tốc độ di chuyển chậm nhất	0 m/s = Đứng yên
Cấu hình truyền dữ liệu	
Dạng truyền thông	UDP
Số nút nguồn phát	random
Số nút nhận	random
Số tuyến kết nối	18
Kích thước gói tin	512 bytes

Bảng 2. Tỉ lệ phát gói tin thành công của giao thức AODV

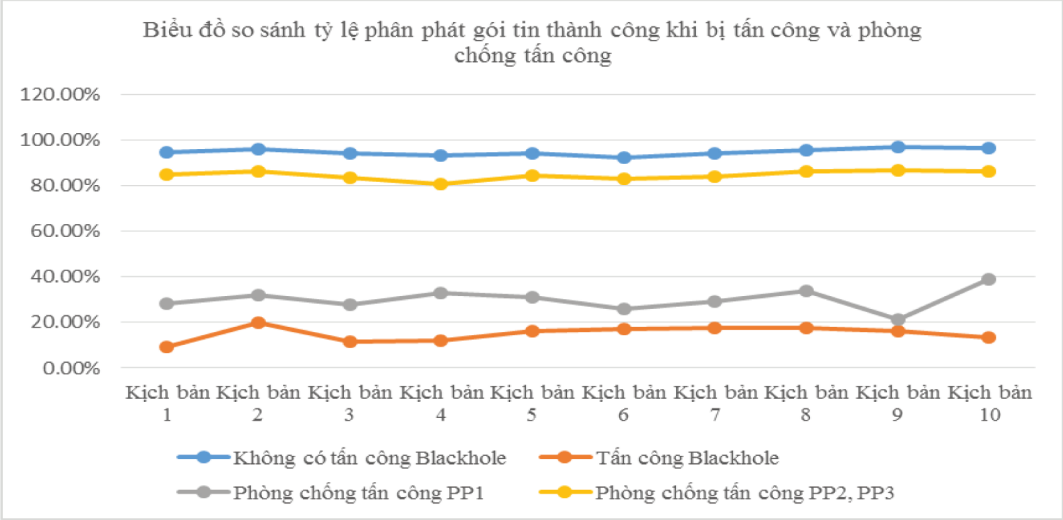
STT	Tên kịch bản	Không có Blackhole		
		Tổng số gói tin gửi đi từ các nút nguồn	*Tổng số gói tin nhận được*	*Tỉ lệ phân phát gói tin*
1	Kịch bản 1	8669	8216	94,77%
2	Kịch bản 2	8723	8371	95,96%
3	Kịch bản 3	8740	8228	94,14%
4	Kịch bản 4	8758	8173	93,32%
5	Kịch bản 5	8675	8190	94,41%
6	Kịch bản 6	8736	8077	92,46%
7	Kịch bản 7	8715	8197	94,06%
8	Kịch bản 8	8729	8336	95,5%
9	Kịch bản 9	8708	8433	96,84%
10	Kịch bản 10	8717	8417	96,56%

Bảng 3. Tỉ lệ phát gói tin thành công khi xảy ra tấn công Blackhole

STT	Tên kịch bản	Tấn công Blackhole		
		Tổng số gói tin gửi đi từ các nút nguồn	*Tổng số gói tin nhận được*	*Tỉ lệ phân phát gói tin*
1	Kịch bản 1	8717	814	9,34%
2	Kịch bản 2	8722	1727	19,8%
3	Kịch bản 3	8673	1025	11,82%
4	Kịch bản 4	8705	1040	11,95%
5	Kịch bản 5	8727	1410	16,15%
6	Kịch bản 6	8730	1496	17,14%
7	Kịch bản 7	8671	1514	17,46%
8	Kịch bản 8	8734	1534	17,56%
9	Kịch bản 9	8712	1429	16,4%
10	Kịch bản 10	8765	1187	13,54%

Bảng 4. Tỉ lệ phát gói tin thành công khi áp dụng ba giải pháp phòng chống

STT	Tên kịch bản	Giải pháp đối chứng			Giải pháp 1 và 2
		Tổng số gói tin gửi đi từ các nút nguồn	*Tổng số gói tin nhận được*	*Tỉ lệ phân phát gói tin*	
1	Kịch bản 1	8692	2443	28,12%	85,19%
2	Kịch bản 2	8679	2791	32,16%	86,43%
3	Kịch bản 3	8731	2445	28%	83,71%
4	Kịch bản 4	8683	2855	32,88%	80,7%
5	Kịch bản 5	8701	2713	31,18%	84,35%
6	Kịch bản 6	8730	2255	25,83%	82,93%
7	Kịch bản 7	8740	2539	29,05%	83,82%
8	Kịch bản 8	8719	2954	33,88%	86,26%
9	Kịch bản 9	8761	1862	21,25%	86,83%
10	Kịch bản 10	8711	3417	39,23%	86,14%



Hình 1. Biểu đồ so sánh tỉ lệ phân phát gói tin

3. Kết luận

Bài báo đã trình bày tổng quan về kiểu tấn công Blackhole và phương pháp cài đặt mô phỏng tấn công Blackhole. Bài báo cũng đề xuất hai phương pháp phòng chống tấn công Blackhole trong giao thức AODV và tiến hành các thực nghiệm bằng công cụ NS-2.35 để kiểm chứng hiệu quả.

Từ kết quả thực nghiệm thu được chúng ta nhận thấy khi chỉ có giao thức AODV thì tỉ lệ phát gói tin thành công từ 92,46% cho tới 96,84% trung bình 94,8%, mất mát dữ liệu là 5,2%. Khi bị tấn công Blackhole thì tỉ lệ phát gói tin thành công giảm xuống trung bình chỉ còn 15,12%, tức là cuộc tấn công đã làm tăng tỉ lệ mất gói tin lên đến 84,88%. Khi sử dụng giải pháp đối chứng (giao thức idsAODV) được đề xuất bởi những nghiên cứu trước đây thì tỉ lệ phát gói tin thành công tăng từ 4,85% lên tới 35,69% tức là tỉ lệ mất gói tin giảm từ 84,88% xuống còn 69,84%. Khi sử dụng giải pháp đề xuất 2 và 3 thì tỉ lệ phát gói tin thành công trung bình là 84,64%, tỉ lệ mất gói tin giảm còn 15,3%, tốt hơn nhiều so với giải pháp đối chứng.

Lời cảm ơn: Những kết quả của bài báo này được hoàn thành trong khuôn khổ của đề tài nghiên cứu "Một số giải pháp nâng cao chất lượng dịch vụ trong mạng sử dụng công nghệ MPLS", mã số: SPHN-12-202, chủ nhiệm đề tài: Nguyễn Thị Quỳnh Hoa.

TÀI LIỆU THAM KHẢO

[1] Royer, E., Chai Keong Toh, April 1999, *A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks*. IEEE Personal Communications, volume:6, issue: 2, page: 46 - 55, ISSN: 1070-9916, 1999.

[2] D. Djenouri, L. Kheladi, N. Badache, 2005. *A Survey of Security Issues in Mobile Ad hoc and Sensor Networks*. IEEE Communications Surveys and Tutorials, volume:6, issue: 2, page: 2 - 28, ISSN: 1553-877X.

[3] Semih Dokurer, 2006 "*Simulation of black hole attack in wireless ad-hoc networks*". A master's thesis in Computer Engineering Atilim University.

[4] C.E.Perkins & E.M. Royer, 2000. *The Adhoc On Demand Distance Vector Protocol in Adhoc networking*. pp.173-219, Addision-Westly.

[5] C.E Perkins and E.M. Royer, 2010. *Ad-hoc On-Demand Distance Vector Routing*. Proceeding of the International Journal of Distributed and Parallel Systems (IJDPS), Vol.1, No.2.

ABSTRACT

A novel way to prevent a blackhole attack

MANET (Mobile Ad-hoc Network) is a network connecting workstations to move at random, regardless of the network infrastructure fixed, therefore offering low operating costs, rapid deployment and high mobility. However, it was thought that MANET networks could become the target of some specific types of attacks. In this paper we look at the influence of the black hole attack (Blackhole Attack) on MANET networks by evaluating packet loss rate in the network. The author also proposes a way to prevent attacks on AODV black hole protocol and presents experimental results to verify his conclusion.

Keywords: Adhoc network, AODV routing protocol, blackhole attack, MANET network, blackhole detection and prevention.