ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

CHU MINH ĐỨC

PHÁT HIỆN TRANH CHẤP TRONG MẠNG NỘI BỘ KHÔNG DÂY

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội – 2019

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

CHU MINH ĐỨC

PHÁT HIỆN TRANH CHẤP TRONG MẠNG NỘI BỘ KHÔNG DÂY

Ngành : Công nghệ thông tin.
Chuyên ngành : Mạng máy tính và truyền thông dữ liệu.
Mã số : 8480102.01

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN
NGƯỜI HƯỚNG DẪN KHOA HỌC: PGS.TS. NGUYỄN ĐÌNH VIỆT

Hà Nội – 2019

LỜI CAM ĐOAN

Tôi xin cam đoan luận văn "**Phát Hiện Tranh Chấp Trong Mạng Nội Bộ Không Dây**" là do tôi thực hiện dưới sự hướng dẫn của PGS. TS. Nguyễn Đình Việt. Nội dung được trình bày thông qua kiến thức tổng hợp cùng với sự tham khảo của các tài liệu trong và ngoài nước, được ghi chú đầy đủ vào trong tài liệu tham khảo, có xuất xứ rõ ràng.

Tác giả luận văn

Chu Minh Đức

LỜI CÁM ƠN

Trước tiên, tôi xin chân thành cám ơn sự giảng dạy nhiệt tình, tâm huyết của tập thể giảng viên trường Đại Học Công Nghệ - Đại Học Quốc Gia Hà Nội, cám ơn đội ngũ nhân viên cán bộ trường đã tạo điều kiện thuận lợi trong quá trình học tập nghiên cứu của tôi tại trường. Đặc biệt là thầy giáo PGS. TS. Nguyễn Đình Việt, người rất nhiệt tình tận tâm chỉ bảo tôi từ lúc bỡ ngỡ nghiên cứu đến khi hoàn thành khóa học cùng với các góp ý quý báu trong quá trình thực hiện đề tài.

Tiếp đến tôi xin cám ơn gia đình bạn bè luôn quan tâm động viên tạo điều kiện cho tôi trong suốt khóa học.

Do thời gian và điều kiện có hạn nên bản khóa luận này không tránh khỏi những thiếu xót, tôi rất mong muốn nhận được sự ý kiến góp ý của các thầy cô cùng các bạn quan tâm tới lĩnh vực này.

Tác giả luận văn

Chu Minh Đức

MỤC LỤC

LỜI CAM ĐOAN ..... **i**
LỜI CÁM ƠN ..... **ii**
MỤC LỤC ..... **iii**
DANH MỤC CÁC KÍ HIỆU VÀ CHỮ VIẾT TẮT ..... **v**
DANH MỤC CÁC BẢNG ..... **vii**
DANH MỤC HÌNH VẼ ..... **viii**
LỜI MỞ ĐẦU ..... **x**
CHƯƠNG 1 – GIỚI THIỆU **1**
1.1 Mạng LAN không dây – WLAN 1
1.1.1 Sự ra đời và ứng dụng 1
1.1.2 So sánh ưu nhược điểm so với mạng LAN có dây 2
1.1.3 Các thành phần của kiến trúc IEEE 802.11 3
1.2 Giao thức cho mạng WLAN – CSMA/CA 4
1.2.1 Giao thức CSMA/CD cho mạng có dây 4
1.2.1.1 Giao thức CSMA 4
1.2.1.2 Giao thức CSMA/CD 4
1.2.2 Các lý do không thể áp dụng giao thức CSMA/CD cho mạng WLAN 5
1.2.2.1 Hiện tượng trạm ẩn (Hidden Terminal problem) 5
1.2.2.2 Hiện tượng trạm lộ (Exposed Terminal problem) 6
1.2.3 Giao thức cho mạng WLAN – CSMA/CA 7
1.3 Giao thức MAC cho mạng WLAN theo chuẩn 802.11 8
1.3.1 Giao thức CSMA/CA có bổ sung việc sử dụng gói tin ACK 11
1.3.2 Cơ chế điều khiển truy cập môi trường truyền DCF 12
1.3.2.1 Cảm nhận sóng mang 12
1.3.2.2 Các phương thức truyền trong DCF 13
1.3.3 Cơ chế điều khiển truy cập môi trường truyền PCF 15
1.3.4 Giao thức MAC theo chuẩn 802.11 (CSMA/CA,+ACK, +RTS/CTS) 16
1.4 Các kiểu tấn công mạng WLAN theo chuẩn 802.11 16

1.5 Các mục tiêu nghiên cứu chính của luận văn. 17
CHƯƠNG 2 – PHÂN TÍCH PHƯƠNG PHÁP TẤN CÔNG GÂY NGHẼN 18
2.1 Jammer và mô hình tấn công jamming 18
2.2 Sử dụng mô hình chuỗi Markov cho cơ chế DCF 19
2.3 Xây dựng biểu thức tính thông lượng cho cơ chế DCF 25
2.4 Phân tích sự tiêu hao năng lượng của nút mạng tấn công kiểu Jamming 28
2.5 Phân tích ảnh hưởng lên thông lượng 30
CHƯƠNG 3 – PHÂN TÍCH KẾ HOẠCH CHỐNG TẤN CÔNG KIỂU GÂY NGHẼN 31
3.1 Phát hiện sự nghẽn mạng (Dectection of Jamming) 31
3.2 Sửa cơ chế DCF để chống tấn công kiểu Jamming 33
CHƯƠNG 4 - MÔ PHỎNG VÀ ĐÁNH GIÁ KẾT QUẢ 39
4.1 Công cụ mô phỏng NS2 39
4.1.1 Giới thiệu và lịch sử phát triển bộ công cụ NS2 39
4.1.2 Cấu trúc bộ công cụ mô phỏng NS2 39
4.1.3 Đặc điểm của bộ mô phỏng NS2 41
4.2 Đề xuất mô hình phát hiện tắc nghẽn 42
4.3 Thực hiện mô phỏng 42
4.3.1 Kịch bản mô phỏng 42
4.3.2 Kết quả và đánh giá mô phỏng. 45
4.4 Kết luận về các kết quả nhận được từ mô phỏng. 52
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN TIẾP THEO 53
TÀI LIỆU THAM KHẢO 54
PHỤ LỤC 55

DANH MỤC CÁC KÍ HIỆU VÀ CHỮ VIẾT TẮT

Từ viết tắt	Từ tiếng Anh
ACK	Acknowledgement
AP	Access Point
BSS	Basic Service Set
CCA	Clear Channel Assessment
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CTS	Clear To Send
DCF	Distributed Coordination Function
DIFS	DCF Inter-Frame Space
DS	Distributed System
DSSS	Direct Sequence Spread Spectrum
EIFS	Extended Inter-Frame Space
ESS	Extended Services Set
FCS	Frame Check Sequence
FHSS	Frequency Hopping Spread Spectrum
GLRT	Generalized Likelihood Ratio Test
HCF	Hybrid Coordination Function
IBSS	Independent Basic Service Set
IEEE	Institute of Electrical and Electronics Engineers
LAN	Local Area Network
LLC	Logical Link Control
MAC	Medium Access Control
MPDU	MAC Protocol Data Units
MSDU	MAC Services Data Units
NAV	Network Allocation Vector
NFC	Near Field Communication

PCF	Point Coordination Function
PDR	Packet Delivery Ratio
PHY	Physical
PLCP	Physical Layer Convergence Procedure
PMD	Physical Medium Dependent
PSR	Packet Send Ratio
RFID	Radio Frequency Identification
ROC	Receiver Operating Characteristic
RSSI	Received Signal Strength Indicator
RTS	Request To Send
SIFS	Sort Inter-Frame Space
SSID	Service Set Identifier
WECA	Wireless Ethernet Compatibility Alliance
WEP	Wired Equivalent Privacy
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
WPA2	Wi-Fi Protected Access II

DANH MỤC CÁC BẢNG

Bảng 2-1 Các tham số thực nghiệm..... 21

Bảng 4- 1 Các thông số mô phỏng..... 45

DANH MỤC HÌNH VẼ

Hình 1-1 Extended Serivce Set 3
Hình 1-2 Hiện tượng hidden terminal 6
Hình 1-3 Hiện tượng exposed terminal 6
Hình 1-4 Giao thức truy cập CSMA/CA 7
Hình 1-5 Chuẩn 802.11 WLAN trên lớp PHY và lớp con MAC 8
Hình 1-6 Cấu trúc khung tin MAC 9
Hình 1-7 Giao thức CSMA/CA + ACK 11
Hình 1-8 Truy cập kênh truyền DCF cơ bản 14
Hình 1-9 Truy cập kênh truyền DCF với RTS/CTS 15
Hình 1-10 Chức năng cộng tác điểm PCF 15
Hình 2-1 Mô hình hóa DCF theo chuỗi Markov 21
Hình 2-2 Xác suất gói tin lỗi và xác suất tắc nghẽn 25
Hình 2-3 Ts và Tc 26
Hình 2-4 Thông lượng thực nghiệm và lý thuyết 27
Hình 2-5 Tương quan năng lượng sử dụng với xác suất gói tin lỗi của jammer 29
Hình 2-6 Lựa chọn xác suất tắc nghẽn với hạn chế năng lượng 30
Hình 3-1 ROC của bộ dò 33
Hình 3-2 DCF chỉnh sửa 34
Hình 3-3 Sử dụng năng lượng của jammer theo DCF, M-DCF với 3,10,20,50 trạm 36
Hình 3-4 Thông lượng và xác suất tắc nghẽn 37
Hình 3-5 Xác suất tắc nghẽn và số trạm 37
Hình 3-6 Xác suất truyền với xác suất gói tin lỗi 38
Hình 4-1 C++ và OTcl trong NS-2 40
Hình 4-2 Cấu trúc thư mục NS2 42
Hình 4-3 Sơ đồ mô phỏng 43

Hình 4-4 Năng lượng nút nguồn và jammer với interval 0.2 46
Hình 4-5 Năng lượng nút nguồn và jammer với interval 0.04 48
Hình 4-6 Năng lượng nút nguồn và jammer với interval 0.008 49
Hình 4-7 Năng lượng nút nguồn và jammer với interval 0.0016 51

LỜI MỞ ĐẦU

Như chúng ta đã biết sự phát triển của công nghệ thông tin trong thời gian gần đây ngày càng mạnh mẽ, mang đến cho người dùng những ứng dụng tuyệt vời, những trải nghiệm mà trước đây tưởng như không bao giờ thành hiện thực. Từ những mạng cơ bản phát triển lên đến mạng không dây 1 - 2 Mb đến nay mạng không dây đã được phát triển và thương mại hóa đến phiên bản 802.11ac tốc độ mạnh mẽ gấp hàng nghìn lần mạng không dây ban đầu.

Tuy phát triển mạnh mẽ là vậy nhưng qua trải nghiệm thực tế trong làm việc cũng như trong học tập tôi nhận thấy cũng có những hệ thống mạng không dây nội bộ hoạt động không hiệu quả. Tín hiệu chập chờn, hoặc tín hiệu tốt nhưng thông lượng lại gần như bằng không. Điều đó chứng minh trong hệ thống mạng có phát sinh tắc nghẽn. Với mong muốn có thể phát hiện nhanh chóng và chính xác tắc nghẽn để xử lý cũng như mang lại hiệu quả trong quá trình truyền tải dữ liệu trong hệ thống mạng nội bộ không dây, tôi thực hiện luận văn này nhằm mục đích nghiên cứu tìm ra được phương pháp phát hiện, tránh tắc nghẽn phát sinh trong hệ thống sớm và cố gắng nghiên cứu các giải pháp mang tính khả thi cao để phòng ngừa tình trạng tắc nghẽn tái phát.

CHƯƠNG 1 – GIỚI THIỆU

Ngày nay với sự phát triển mạnh của các thiết bị di động cũng như những ứng dụng công nghệ thông tin và đặc biệt là mạng xã hội và các ứng dụng tương tác thời gian thực, do đó yêu cầu kết nối mạng không dây là rất mạnh mẽ. Ta dễ dàng bắt gặp mạng LAN không dây – WLAN (WLAN), thường được gọi là mạng Wi-Fi ở bất cứ đâu từ hộ gia đình, quán cafe, quán ăn, công sở, bến xe thậm chí là quán nước... Điều gì khiến mạng WLAN trở nên phổ biến như vậy. Ưu và khuyết điểm của nó là gì? Chương này sẽ giúp chúng ta trả lời các câu hỏi trên.

1.1 Mạng LAN không dây – WLAN

1.1.1 Sự ra đời và ứng dụng

Mạng không dây WLAN là một hệ thống các thiết bị nối mạng không thông qua hệ thống cáp kết nối mà thông qua các kênh truyền không dây, sử dụng sóng điện từ. Mạng WLAN hoạt động dựa trên chuẩn IEEE 802.11 hay được gọi đơn giản là mạng Wi-Fi.

Các mốc thời gian quan trọng của mạng không dây:

- 1985: Ủy ban liên lạc liên bang Mỹ FCC (là cơ quan quản lý viễn thông) cho phép sử dụng 3 băng tần không dây không cần xin phép của chính phủ. Ba dải sóng này còn được gọi là các "băng tần rác" (900 MHz, 2,4 GHz, 5,8 GHz), được phân bổ cho các thiết bị sử dụng vào các mục đích ngoài liên lạc, chẳng hạn như lò nướng vi sóng sử dụng các sóng vô tuyến radio để đun nóng thức ăn.
- 1988: Công ty NCR vì muốn sử dụng dải tần "rác" để liên thông các máy rút tiền qua hệ thống không dây đã liên hệ với tổ chức IEEE, một tiểu ban mới - 802.11 được thiết lập nhằm xác định một tiêu chuẩn cho công nghệ không dây.
- 1997: Tiểu ban này đã phê chuẩn một bộ tiêu chí cơ bản, cho phép mức truyền dữ liệu 2Mbps, sử dụng một trong 2 công nghệ dải tần rộng là nhảy tần (frequency hopping), tránh nhiễu bằng cách chuyển đổi liên tục giữa các tần số radio; hoặc Direct Sequence Spread Spectrum (phát tín hiệu trên một dải gồm nhiều tần số - DSSS).
- 1999: Chuẩn IEEE 802.11b hoạt động trên băng tần 2.4Ghz được phê duyệt.
- 1999: Do bộ tiêu chuẩn 802.11 quá dài và phức tạp và vấn đề tương thích còn nhiều khó khăn, 6 công ty bao gồm Intersil, 3Com, Nokia,

Aironet (về sau được Cisco sáp nhập), Symbol và Lucent liên kết với nhau để tạo ra Liên minh tương thích Ethernet không dây WECA (Wireless Ethernet Compatibility Alliance). Để cho dễ gọi các thiết bị đạt chuẩn tương thích không dây về sau được gọi là Wi-Fi (Wireless Fidelity).

- 2000: Chuẩn IEEE 802.11a hoạt động trên băng tần 5.8Ghz được phê duyệt vào tháng 1.

Các chuẩn mạng được sử dụng rộng rãi đến nay [4]:

- 1997: 802.11 Wi-Fi thế hệ đầu tiên, có thể tốc độ 1 Mb/s và 2Mb/s, sử dụng băng tần 2.4Ghz của sóng radio hoặc hồng ngoại.
- 1999: 802.11b Wi-Fi thế hệ thứ hai, tốc độ lên đến 11Mb/s trên băng tần 2.4Ghz. Ra mắt đồng thời là 802.11a Wi-Fi thế hệ thứ ba, tốc độ đạt đến 54Mb/s hoạt động trên băng tần 5Ghz.
- 2003: 802.11g Wi-Fi thế hệ thứ tư, tốc độ truyền tải 54Mb/s trên băng tần 5Ghz.
- 2009: 802.11n Wi-Fi thế hệ thứ năm tốc độ tối đa 500Mb/s có thể hoạt động trên cả hai băng tần 2.4Ghz và 5Ghz. Nếu router hỗ trợ có thể phát sóng song song.
- 201x: 802.11ac tốc độ tối đa hiện là 1730Mb/s (sẽ còn tiếp tục tăng) chỉ chạy trên băng tần 5Ghz.

1.1.2 So sánh ưu nhược điểm so với mạng LAN có dây

Ưu điểm của mạng không dây với mạng có dây:

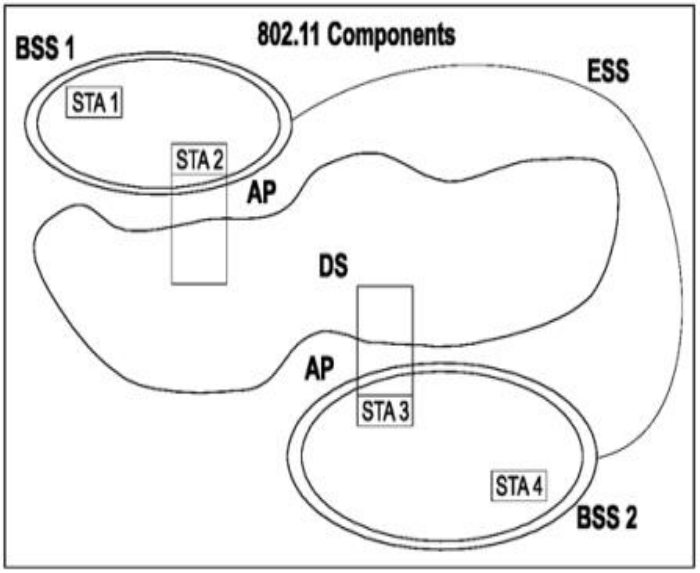
- Ưu điểm đầu tiên cũng là đặc điểm nổi bật của mạng WLAN kết nối không ràng buộc dây nối vật lý.
- WLAN có tính mềm dẻo cao, khả năng thiết lập nhanh và dễ dàng, có thể đáp ứng tức thì việc gia tăng số lượng người dùng.

Nhược điểm của mạng không dây với mạng có dây:

- Mạng không dây có phạm vi hoạt động nhỏ.
- Dễ bị ảnh hưởng bởi môi trường, của các thiết bị phát sóng khác.
- Tốc độ truyền dữ liệu vẫn thấp hơn mạng có dây.
- Mạng không dây dễ bị tổn thương về tính bảo mật.

1.1.3 Các thành phần của kiến trúc IEEE 802.11

- Kiến trúc 802.11 bao gồm 1 số thành phần tương tác với nhau nhằm giúp sự di chuyển của các trạm khác không dây là trong suốt không ảnh hưởng tới các lớp trên, một số các thành phần được đề cập đến trong luận văn là bacsic service set – BSS, independent basic service set – IBSS, extended service set – ESS, cụ thể như sau [3, tr184-187]:
- BSS: đây là khối cơ bản của kiến trúc 802.11, bao gồm 1 trạm phân phối thường là AP với các máy khách sử dụng chung môi trường truyền . Mỗi một BSS có một tên riêng gọi là SSID được quảng bá cho các máy khách để nhận biết được BSS và tham gia nếu có nhu cầu.
- IBSS: đây là thánh phần cơ bản nhất trong kiến trúc 802.11 LAN. Các BSS không có kết nối với nhau thì được gọi là các IBSS. Trong IBSS các máy khách giao tiếp với nhau trực tiếp không qua trạm phân phối AP. Đây là mô hình mạng ad-hoc sử dụng.
- ESS: đây là thành phần mở rộng từ các BSS kết nối với nhau thông qua hệ thống phân tán.



Hình 1-1 Extended Serivce Set

1.2 Giao thức cho mạng WLAN – CSMA/CA

1.2.1 Giao thức CSMA/CD cho mạng có dây

1.2.1.1 Giao thức CSMA

CSMA (Carrier-Sense Multiple Access) là một hệ thống quản lý đa truy cập được sử dụng trong mạng không dây đầu tiên ALOHA và rất phổ biến hiện nay với nhiều phiên bản cải tiến. CMSA có nhiều phiên bản giao thức, cụ thể như sau [1]:

- 1-persistent: thiết bị sẽ cảm nhận kênh truyền trước khi truyền. Nếu đường truyền rỗi thì truyền. Nếu đường truyền bận thì sẽ tiếp tục nghe đường truyền, đến khi thấy rỗi là truyền ngay (xác suất truyền khi rỗi là bằng 1).
- P-persistent:Tương tự như 1-persistent, nhưng có điểm khác là khi thấy kênh truyền rỗi, thiết bị sẽ truyền với xác suất $p < 1$.
- Non-persistent: thiết bị sẽ cảm nhận kênh truyền trước khi truyền. Nếu đường truyền rỗi thì truyền; Nếu đường truyền bận thì sẽ đợi sau một thời gian ngẫu nhiên được quy định bởi thuật toán back-off và sẽ lặp lại việc nghe kênh truyền.

1.2.1.2 Giao thức CSMA/CD

Đây là giao thức được sử dụng trong mạng LAN. Giao thức CSMA/CD được cải tiến từ giao thức CSMA bằng việc thêm vào tính năng phát hiện xung đột trong khi đang truyền, viết tắt là CD (Colission Detection). Khi các trạm có nhu cầu truyền cùng nhận thấy kênh truyền rảnh thì có thể đồng thời bắt đầu truyền, do đó có thể xảy ra xung đột, nếu không phát hiện được xung đột ngay khi xảy ra, các trạm vẫn phát tiếp gói tin cho đến hết và toàn bộ các gói tin đó sẽ phải phát lại, như vậy đường truyền bị sử dụng lãng phí một cách vô ích. Với CSMA/CD, trạm vừa truyền vừa tiếp tục theo dõi đường truyền, khi phát hiện xung đột trạm sẽ ngừng phát ngay và gửi quảng bá (broadcast) một gói tin báo hiệu (gọi là jamming signal) cho các máy trên mạng biết là có xung đột, để các trạm khác không gửi gói tin để tránh gia tăng tắc nghẽn đường truyền, và sẽ tiến hành gửi lại gói tin sau.

Khi gói tin bị hỏng do đụng độ thì trạm sẽ truyền lại đến khi truyền thành công hoặc khi đạt đến số lần truyền lại tối đa thì hủy bỏ truyền tin. Việc truyền lại được quyết định bởi thuật toán “truncated binary exponential backoff”, thuật toán này cho biết khoảng thời gian cần chờ để gửi lại gói tin

khi gặp đụng độ. Thời gian chờ được tính là r ngẫu nhiên có giá trị $0 < r < 2^k - 1$ được chọn sau mỗi lần đụng độ, k = min(n,10). Khi n đạt tới giá trị 16 tức là nỗ lực truyền lại 16 lần vẫn không được thì hủy việc truyền gói tin và bắt đầu một quá trình truyền mới [5].

Ưu điểm của CSMA/CD là đơn giản, mềm dẻo, hiệu quả truyền thông tin cao khi lưu lượng thông tin của mạng thấp. Điểm bất lợi của CSMA/CD là hiệu suất của mạng sẽ giảm nhanh chóng khi tải đưa vào mạng tăng lên cao.

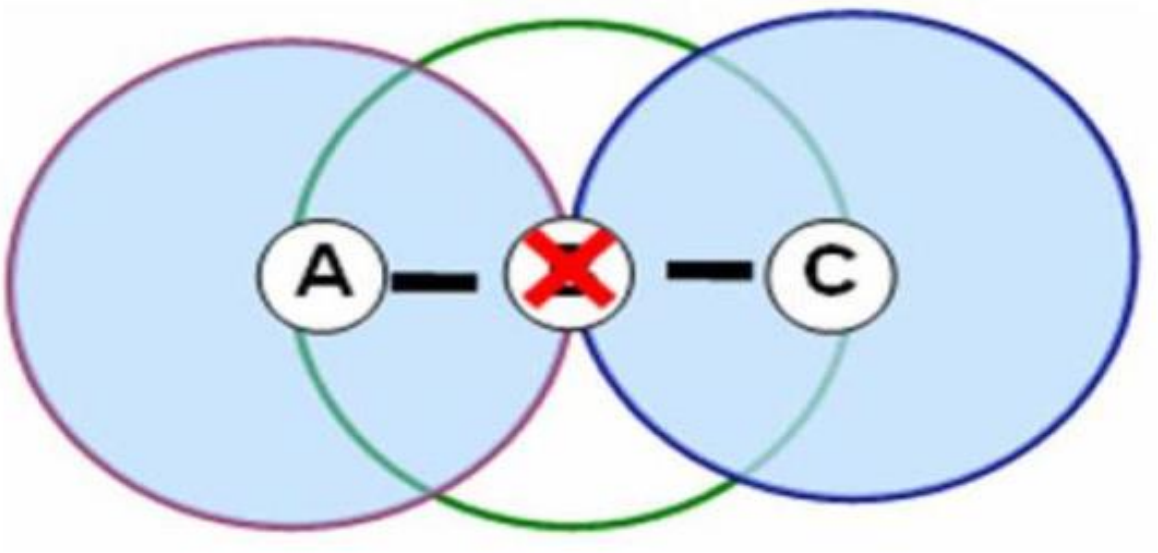
1.2.2 Các lý do không thể áp dụng giao thức CSMA/CD cho mạng WLAN

Khác với mạng có dây, việc phát hiện đụng độ là điều không khả thi vì trong mạng không dây trạm gửi chỉ có thể truyền và nhận gói tin trên môi trường truyền, nhưng không thể cảm nhận được gói tin truyền trong môi trường như thế nào, độ nhiễu của môi trường cũng ảnh hưởng đến quá trình lắng nghe. Bên cạnh đó muốn phát hiện đụng độ thì cần một bộ thu một bộ phát làm tăng giá thành của thiết bị lên quá cao.

Cũng chính vì môi trường truyền dẫn không dây nên có một số vấn đề CSMA/CD không khả dụng như là hiện tượng trạm ẩn, hiện tượng trạm lộ [1].

1.2.2.1 Hiện tượng trạm ẩn (Hidden Terminal problem)

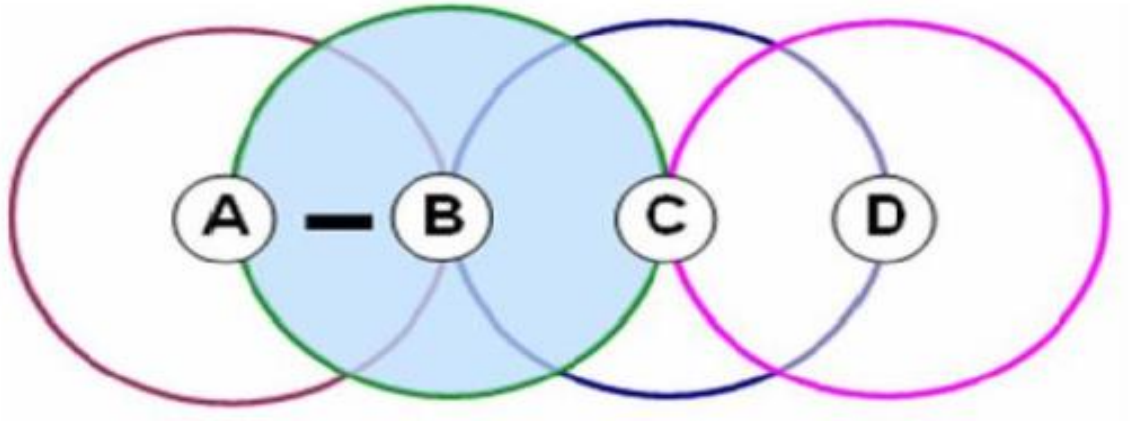
Đối với mạng không dây, đôi khi có những vị trí mà nút mạng tại đó không thể liên lạc trực tiếp với các nút khác trong mạng. Ta có thể quan sát theo hình 1-2, trạm B có thể liên với cả trạm A và C, nhưng trạm A và C không thể liên lạc trực tiếp với nhau (điều này có thể do khoảng cách giữa chúng quá xa so với khoảng cách từ nút B đến C hoặc đến A, do đó A nằm ngoài vùng thu/phát của C và ngược lại). Như vậy nút A và C là ẩn đối với nhau. Khi A, C có gói tin cần truyền cho B, cả 2 nút cảm nhận đều thấy đường truyền rỗi. Do đó A và C cùng truyền gói tin đến B, lúc này đụng độ sẽ xảy ra tại B, việc truyền là thất bại mà do A và C không cảm nhận được thấy nhau. Vấn đề này được gọi là hiện tượng trạm ẩn.



Hình 1-2 Hiện tượng hidden terminal

1.2.2.2 Hiện tượng trạm lộ (Exposed Terminal problem)

Hình 1-3 mô tả hiện tượng trạm lộ, đây là hiện tượng một nút tưởng đường truyền bận và dừng truyền tin mà thực tế thì không phải. Ví dụ trạm B đang gửi dữ liệu tới trạm A và trạm C muốn gửi dữ liệu cho trạm D. Nếu sử dụng giao thức CSMA/CD thì C phải đợi đến khi A và B hoàn thành truyền tin (để gửi gói tin cho D) nhưng trạm A và D không nằm trong vùng phủ sóng của nhau nên việc C đợi là không cần thiết, thực tế thì đường truyền giữa C và D là rỗi nên C có thể thực hiện truyền gói tin cho D mà không cần phải chở đợi. Như vậy nút C là exposed với nút B.

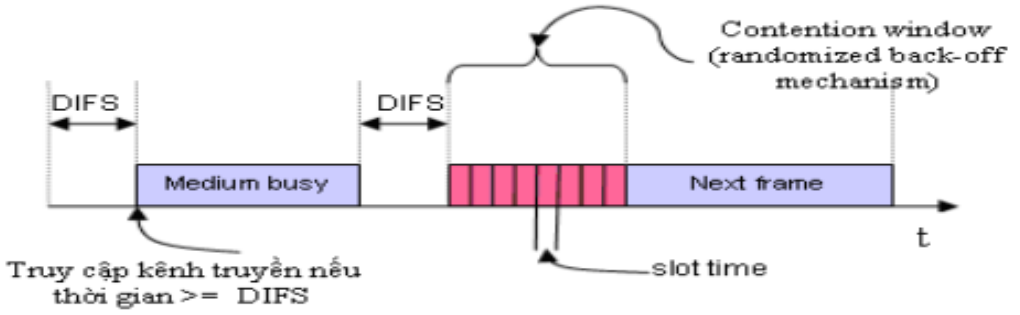


Hình 1-3 Hiện tượng exposed terminal

1.2.3 Giao thức cho mạng WLAN – CSMA/CA

Cũng như mạng có dây, mạng WLAN sử dụng môi trường truyền chung cho nên cũng cần có cơ chế ngăn chặn đụng độ xảy ra. CSMA/CA (Carrier-sense multiple access with collision avoidance) là một phương thức quản lý đa truy cập với phương thức tránh đụng độ sử dụng trong mạng không dây dựa trên cơ chế CSMA. Có nghĩa là trạm sẽ chỉ truyền khi cảm nhận môi trường truyền là rỗi, và khi truyền sẽ truyền toàn bộ dữ liệu. Cơ chế hoạt động của giao thức CSMA/CA như sau [1]:

- Khi một trạm muốn truy cập môi trường truyền, trạm đó sẽ nghe xem môi trường truyền có bận hay không (đây là cơ chế CS).
- Nếu môi trường truyền rỗi thì trạm đó đợi một khoảng thời gian ít nhất là DIFS để truy cập môi trường truyền (đây là cơ chế MA).
- Nếu môi trường truyền bận, trạm muốn truyền đó sẽ đợi một khoảng thời gian DIFS cộng với thời gian back-off được chọn ngẫu nhiên trong cửa sổ tranh chấp (Contention Window). Sau mỗi khoảng thời gian DIFS, nếu môi trường truyền rỗi, thời gian back-off này được giảm đi 1, ngược lại được giữ nguyên cho khoảng thời gian DIFS tiếp theo. Tuy nhiên, nếu một trạm bất kỳ khác đã truy cập môi trường truyền trước khi thời gian back-off của trạm này giảm đến 0 thì bộ đếm back-off sẽ tạm dừng cho đến lần truy cập tiếp theo (đây là cơ chế CA). Mặc dù vậy khi thời gian back-off kết thúc, trạm truyền bắt đầu truyền gói tin, tại giai đoạn này khả năng đụng độ có thể tái xảy ra. Nhưng nhìn chung cơ chế back-off giúp giảm thiểu xác suất xảy ra đụng độ.



DIFS: DCF IFS – Distributed Coordination Function Inter Frame Spacing
Contention Window: cửa sổ tranh chấp

Hình 1-4 Giao thức truy cập CSMA/CA

Giao thức CSMA/CA còn sử dụng gói tin ACK để giúp phát hiện đụng độ. Việc sử dụng ACK khá đơn giản, khi một thiết bị không dây gửi gói tin, trạm nhận sẽ đáp lại bằng ACK nếu như gói tin đó được nhận đúng và đầy đủ. Nếu trạm gửi không nhận được ACK thì nó xem như là đã có xung đột xảy ra và truyền lại gói tin.

CSMA/CA giảm nguy cơ đụng độ xảy ra tuy nhiên không thể loại bỏ hoàn toàn các vấn đề tiềm ẩn. Một cải tiến được áp dụng là bổ sung việc sử dụng cặp gói tin điều khiển RTS/CTS.

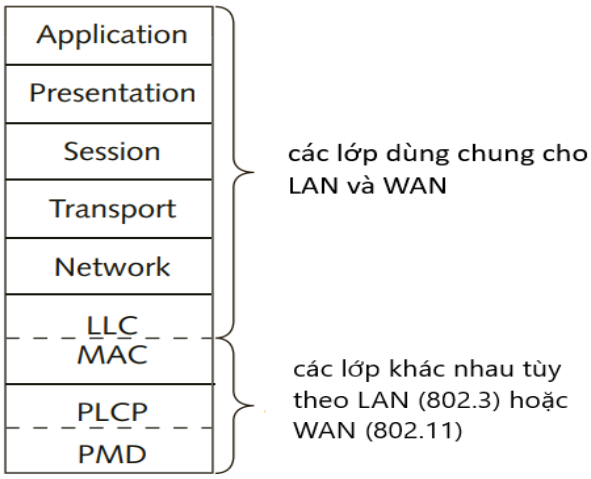
Hai vấn đề ở trên sẽ được làm rõ ở phần sau của luận văn.

1.3 Giao thức MAC cho mạng WLAN theo chuẩn 802.11

Các mạng WLAN theo tiêu chuẩn IEEE 802.11, đây là một tập hợp các đặc tả quản lý lớp điều khiển truy cập môi trường truyền (MAC) và lớp vật lý (PHY) cho việc triển khai mạng không dây cục bộ WLAN trên các dải tần 900MHz, 2.4GHz, 3.5 GHz, 5GHz và 60 GHz [3].

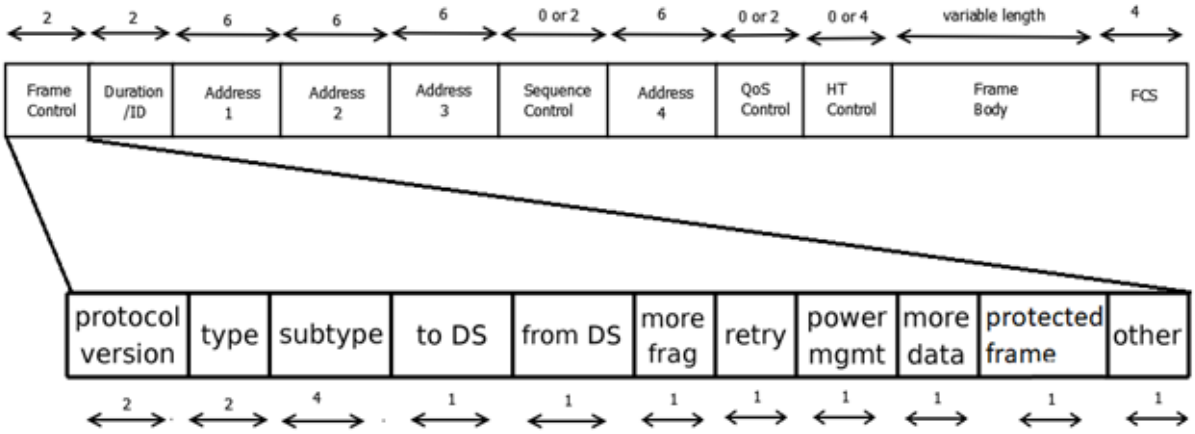
Các đặc tả của chuẩn 802.11 tập trung vào 2 lớp thấp nhất trong mô hình OSI là lớp vật lý (PHY) và liên kết dữ liệu Data Link. Mục tiêu chính của chuẩn 802.11 là phát triển lớp con MAC và lớp PHY cho các thiết bị di động.

Lớp LLC là lớp con trong lớp Data Link được định nghĩa trong chuẩn 802.2, LLC có trách nhiệm chính trong việc cung cấp giao tiếp giữa lớp MAC và các lớp cao hơn. LLC thực hiện nhiều chức năng trong việc hỗ trợ cho nhiều lớp ở tầng cao hơn. Và hơn thế nữa lớp con LLC còn có chức năng kiểm soát luồng và kiểm soát lỗi.



Hình 1-5 Chuẩn 802.11 WLAN trên lớp PHY và lớp con MAC

Lớp con MAC nhận dữ liệu từ lớp con LLC và có trách nhiệm thực hiện các chức năng liên quan đến việc truyền gói tin vào môi trường truyền. Cấu trúc của một frame MAC được mô tả theo hình 1-6 [3].



Hình 1-6 Cấu trúc khung tin MAC

- Frame control: trường frame control chứa một số trường con bao gồm:
    - Protocol version: trường này dài 2 bits và xác định phiên bản của MAC. Hiện tại chỉ có 1 tiêu chuẩn và nó được gán giá trị là 0.
    - Type: trường này dài 2 bits phân loại khung thuộc về quản lý, điều khiển hay là dữ liệu (management, control, data).
    - Subtype: trường con subtype dài 4 bits, giá trị trường này phụ thuộc vào giá trị của trường "Type" (management, control, data).
    - To DS from DS: 2 trường này dành cho frame thuộc hệ phân tán có các cặp giá trị khác nhau tùy thuộc vào kiến trúc mạng. Nếu giá trị "To DS" = 0 và "from DS" = 0 có nghĩa là khung data truyền giữa các trạm trong cùng 1 IBSS không qua AP. Nếu giá trị "To DS" = 1 và "from DS" = 0 có nghĩa là khung data truyền có thông qua AP. Nếu giá trị "To DS" = 0 và "from DS" = 1 tức là khung data truyền giữa các BSS chung AP và 3 trường address được sử dụng. Nếu giá trị "To DS" = 0 và "from DS" = 1 có nghĩa là khung data truyền giữa các AP khác nhau trong ESS và lúc này cả 4 trường address được sử dụng [3, tr641].
    - More frag: trường này có chiều dài 1bit, nếu frame bị phân mảnh thì tất cả các frame là mảnh của frame ban đầu bị phân mảnh có giá trị trường More frag là 1, trừ frame cuối.

    - Retry: trường con này dài 1 bit, nếu frame này cần được gửi lại thì giá trị này được gán là 1 (ngược lại là 0).
    - Power mgmt: trường này có chiều dài 1 bit được dùng để quy định chế độ năng lượng của máy trạm. Nếu thiết bị gửi gói tin đi đang ở trong trạng thái tiết kiệm năng lượng (powesave) thì giá trị được gán là 1 (và ngược lại là 0).
    - More data: khi thiết bị nhận ở chế độ powersave thì AP có thể lưu trữ tạm một số frame gửi cho nó. Bit này được đặt là 1 báo hiệu là AP có 1 vài frame cho thiết bị đang ở chế độ sleeping.
    - Protected Frame: có giá trị là 1 khi cơ chế mã hóa được dùng để mã hóa frame. Các cơ chế mã hóa có thể là WEP (Wired Equivalent Privacy), WPA (Wi-Fi Protected Access), hoặc WPA2 (Wi-Fi Protected Access II).
    - Other: được đặt là 1 nêu thứ tự frame được đặt ưu tiên tức là các frame bắt buộc phải được gửi theo thứ tự.
- Duration/ID: trường này có chiều dài 16 bits miêu tả thời gian truyền frame và nhận gói tin ACK. Việc này dùng để thiết lập NAV (network allocation vector) cho các thiết bị lân cận. Trường này có thể nhận 1 trong 3 dạng: Duration, Contention-Free Period (CFP), and Association ID (AID).
- Address: frame 802.11 có thể ghi nhận 4 địa chỉ MAC.
    - Address 1: địa chỉ của thiết bị nhận.
    - Address 2: địa chỉ của thiết bị gửi.
    - Address 3: dùng cho thiết bị nhận lọc gói tin.
    - Address 4: phần lớn trường hợp không sử dụng chỉ sử dụng khi frame truyền giữa các AP trong EES, hoặc giữa các nút trung gian trong mạng hỗn hợp.
- Sequence Control: trường này dùng để loại bỏ gói tin trùng lặp.
- QoS Control: trường này là trường lựa chọn, chỉ xuất hiện với gói tin của ứng dụng có yêu cầu QoS.
- HT Control: được bổ sung vào từ phiên bản 802.11 liên quan đến QoS.
- Frame Body: trường này chứa dữ liệu cần truyền, có độ dài thay đổi tùy vào loại khung và các trường subtypes.
- FCS: trường này dùng để kiểm tra tính toàn vẹn của gói tin ở bên nhận.

Trong mạng không dây các thiết bị truyền tín hiệu cho nhau thông qua sóng điện từ, chia sẻ môi trường truyền. Để đảm báo tín hiệu truyền thông suốt và sử dụng hiệu quả môi trường truyền cần có giao thức quản lý. Đó là

giao thức điều khiển truy cập môi trường truyền - MAC được thực hiện qua các chức năng cộng tác (coordination function). Các chức năng cộng tác này quyết định khi nào thì thiết bị có thể truyền qua sóng không dây.

Trong mạng WLAN có một số phương thức điều khiển môi trường truy cập chính là:

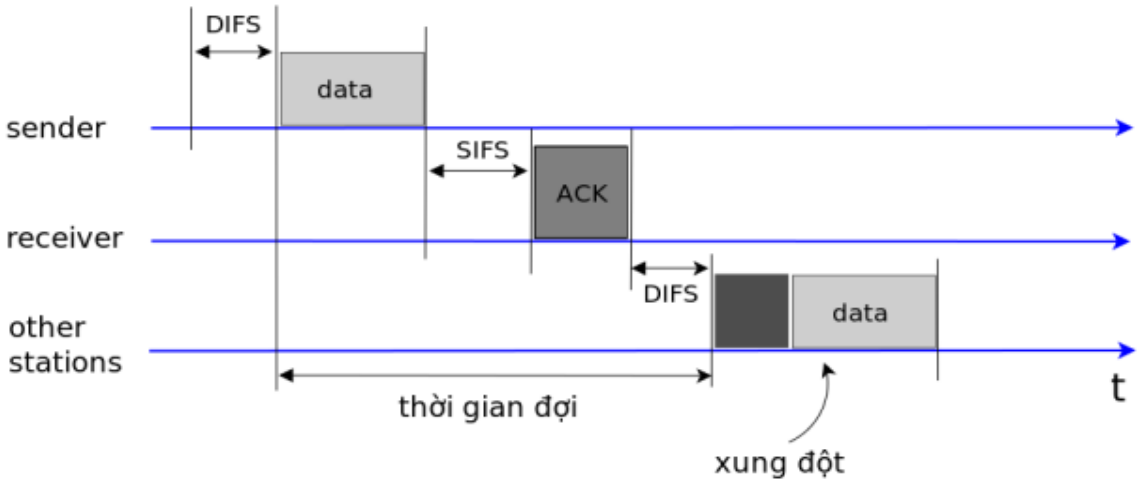
- Chức năng cộng tác phân tán DCF.
- Chức năng cộng tác phân tán DCF sử dụng 2 gói tin RTS/CTS.
- Chức năng cộng tác điểm PCF.

DCF là thành phần chính trong chuẩn 802.11 còn PCF là thành phần bổ sung nằm phía trên DCF hỗ trợ cho các lưu lượng thời gian thực. Sau này để hỗ trợ cho việc tăng chất lượng dịch vụ QoS thì một chức năng lai được thêm vào đó là HCF được giới thiệu trong chuẩn 802.11e. Trong luận văn này không đi chi tiết vào HCF.

1.3.1 Giao thức CSMA/CA có bổ sung việc sử dụng gói tin ACK

Giao thức CSMA/CA có sử dụng ACK được cải tiến từ giao thức CSMA/CA và thêm thông báo biên nhận ACK. Quá trình thực hiện của giao thức này như sau:

- Trạm có nhu cầu gửi nghe đường truyền, nếu đường truyền rỗi, nó phải chờ một khoảng thời gian ít nhất là DIFS rồi mới truyền.
- Bên nhận sau khi nhận được gói dữ liệu sẽ gửi gói biên nhận ACK sau khoảng thời gian SIFS (Short Inter-Frame Space)
- Nếu ACK bị mất (bên gửi chờ sau khoảng thời gian DIFS mà không nhận được ACK), việc truyền lại sẽ được tiến hành.



Hình 1-7 Giao thức CSMA/CA + ACK

Cơ chế báo nhận ACK được thêm vào giao thức CSMA/CA sẽ đảm bảo bên gửi biết được gói tin được đến đích mà không có lỗi, nếu có lỗi cũng có thể gửi lại một cách nhanh chóng. Các trạm muốn phát đều phải nghe đường truyền để phát gói tin vào các khe thời gian không giao nhau nên không thể xảy ra xung đột.

1.3.2 Cơ chế điều khiển truy cập môi trường truyền DCF

Lớp con DCF là giao thức truy cập môi trường cơ bản giữa các lớp PHY tương thích thông qua việc sử dụng CSMA/CA và thời gian chờ ngẫu nhiên (thuật toán random backoff) nếu môi trường truyền là bận. Giao thức CSMA/CA được thiết kế để giảm xác suất đụng độ giữa các trạm khi tham gia truyền tin trong cùng một môi trường.

Để thuật toán hoạt động hiệu quả và công bằng thuật toán DCF bao gồm một tập các độ trễ theo độ ưu tiên thuật toán. Độ trễ chúng ta cần quan tâm trước tiên là khoảng trống liên khung IFS. Thời gian cụ thể của IFS phụ thuộc vào từng loại frame. Các khung thời gian IFS được định nghĩa trong chuẩn 802.11 bao gồm [3, tr1307-1310]:

- SIFS (Short Inter-Frame Space): là khoảng thời gian giữa 2 frame ngắn nhất được sử dụng khi gửi gói tin ACK và CTS, khi nhận được 2 gói tin này thì môi trường đã sẵn sàng để truyền nên không cần phải chờ đợi lâu, hoặc được sử dụng khi trả lời thăm dò (polls) trong sơ đồ PCF.
- PIFS (Point coordination function Inter-Frame Space): là khoảng thời gian chờ giữa 2 khung, có giá trị lớn hơn SIFS và nhỏ hơn DIFS. PIFS được sử dụng trong sơ đồ PCF khi phát các gói tin thăm dò.
- DIFS (Distributed coordination function Inter-Frame Space): là khoảng thời gian giữa 2 khung tin (data frame). Được sử dụng là độ trễ tối thiểu cho các khung không đồng bộ tranh quyền truy cập.
- EIFS (Extended IFS): khi nhận 1 frame lỗi thì trạm phải đợi một khoảng EFIS thay vì DIFS thông thường trước khi truyền tiếp khung.

Các thời gian liên khung trên được xếp theo độ ưu tiên như sau SIFS < PIFS < DIFS < EIFS.

1.3.2.1 Cảm nhận sóng mang

Quá trình điều khiển truy cập theo DCF bao gồm 2 giai đoạn: cảm nhận sóng mang và truyền tránh đụng độ.

Trước khi khởi tạo quá trình truyền dữ liệu thì thiết bị cần cảm nhận sóng mang trên kênh truyền. Chuẩn 802.11 định nghĩa 2 phương thức cảm nhận sóng: cảm nhận sóng mang vật lý – PCS (Physical Carrier Sensing) và cảm nhận sóng mang ảo – VCS (Virtual Carrier Sensing).

Phương thức cảm nhận sóng mang vật lý – PCS (Physical Carrier Sensing) quá trình cảm nhận được diễn ra ở mức vật lý sử dụng chức năng kiểm tra kênh truyền rỗi (clear channel assessment function - CCA). CCA có thể được dùng cho cả hai phương thức phát hiện tín hiệu mạch lạc (coherent) và không mạch lạc (non-coherent).

- Phát hiện tín hiệu mạch lạc liên quan đến phát hiện trường preamble theo đó nốt cảm nhận sẽ đồng bộ với trường preamble của khung tin. Với phương thức này thì hàm CCA sẽ chạy liên tục để có thể phát hiện preamble trên kênh truyền.
- Phát hiện tín hiệu không mạch lạc thì bộ báo cường độ tín hiệu nhận được (the received signal strength indicator - RSSI) sẽ được so sánh với một số ngưỡng được quy định sẵn. Tín hiệu có thể được phát hiện ở giữa khung nên do đó tiết kiệm năng lượng hơn

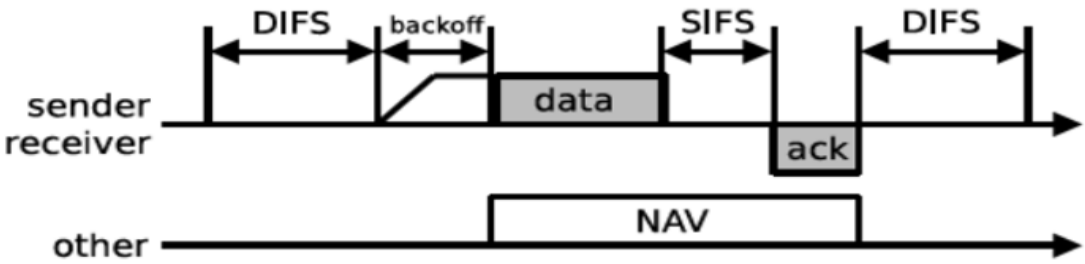
Phương thức cảm nhận sóng mang ảo – VCS (Virtual Carrier Sensing) thì sử dụng một bộ đếm gọi là vector phân bổ mạng – NAV để đặt trước kênh truyền. Mọi thiết bị đều theo dõi NAV và cảm nhận kênh truyền bằng giá trị của NAV. Nếu giá trị của NAV khác 0 có nghĩa là có thiết bị đang truyền trên kênh.

Thiết bị cảm nhận kênh trước khi truyền theo một khoảng thời gian được chỉ định bởi thời gian liên khung DIFS. Nếu kênh truyền rảnh trong thời gian này thì thiết bị truyền frame, ngược lại thiết bị tạm hoãn truyền và tiếp tục cảm nhận kênh truyền. Một khi kênh truyền rỗi trở lại thiết bị sẽ cảm nhận khoảng thời gian DIFS và sau đó vào giai đoạn chờ hết khoảng thời gian back-off. Bộ đếm back-off sẽ dừng lại nếu đường truyền bận và tiếp tục đếm khi đường truyền rỗi. Thiết bị sẽ truyền frame khi bộ đếm về 0.

1.3.2.2 () Các phương thức truyền trong DCF

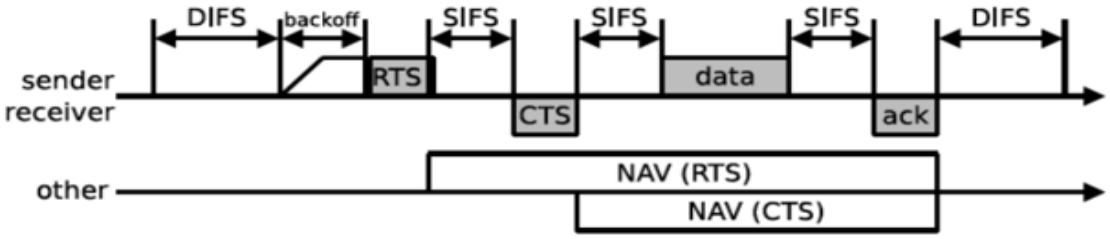
DCF hỗ trợ 2 phương thức truyền, cơ bản và có sử dụng RTS/CRS:

- Với phương thức truyền cơ bản thì thiết bị sẽ cảm nhận kênh truyền, nếu kênh truyền trong trạng thái rỗi thì thiết bị sẽ chờ thêm một khoảng thời gian backoff rồi sẽ truyền dữ liệu. Các thiết bị xung quanh phát hiện có frame trên kênh truyền sẽ thiết lập giá trị NAV phụ thuộc vào giá trị trên header của frame. Nếu frame truyền thành công thì thiết bị nhận sau khi chờ 1 khoảng thời gian SIFS sẽ trả lời một ACK trước khi NAV hết hạn. Tuy nhiên khả năng đụng độ vẫn có thể xảy ra mặc dù xác suất ít khi 2 nút cùng cảm nhận và thấy đường truyền rỗi và sau đó cùng truyền tin, lúc này đụng độ sẽ xảy ra. Quá trình truyền cơ bản được mô tả theo hình 1-8.



Hình 1-8 Truy cập kênh truyền DCF cơ bản

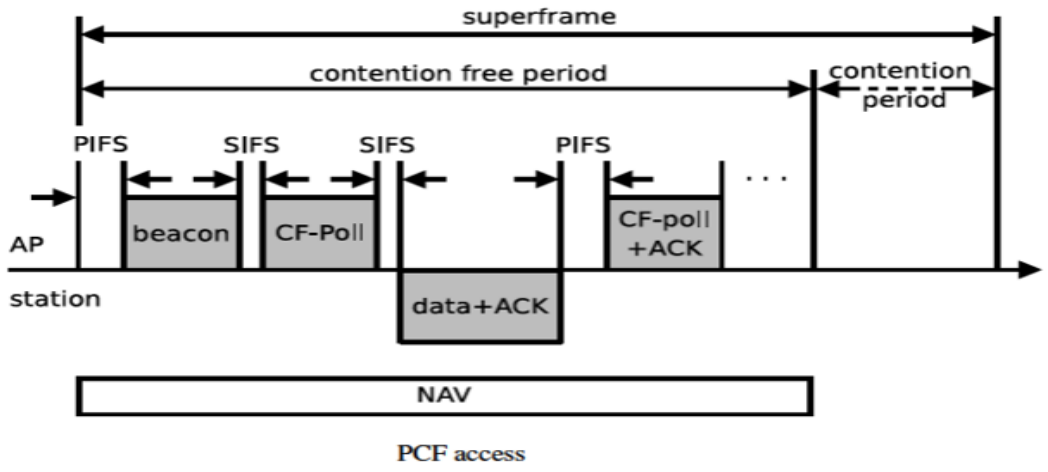
- Phương thức truyền tránh đụng độ RTS/CTS thì thiết bị sẽ đặt trước độ ưu tiên kênh truyền để truyền dữ liệu. Khi nghe kênh truyền và thấy rỗi trong khoảng thời gian SIFS bên gửi sẽ gửi 1 frame RTS. Khi bên nhận nhận được frame RTS nó sẽ chờ một khoảng thời gian SIFS và sau đó sẽ gửi frame CTS. Như vậy các gói tin RTS và CTS sẽ không thể đụng độ với các gói tin dữ liệu, chỉ được truyền sau khi môi trường truyền rỗi ít nhất là DIFS, có giá trị lớn hơn SIFS. Các thiết bị xung quanh khi cảm nhận có sự tồn tại của 1 trong 2 frame RTS/CTS sẽ thiết lập giá trị NAV theo trường duration trong frame RTS/CTS. Sau khi quá trình trao đổi RTS/CTS hoàn thành thì dữ liệu và ACK được truyền giống như phương thức cơ bản. Mặc dù ít xảy ra nhưng khả năng đụng độ khi sử dụng gói tin RTS/CTS vẫn có thể phát sinh, khả năng đụng độ phát sinh vào giai đoạn phát gói tin RTS. Do có gói tin CTS xác nhận nút nhận sẵn sàng nhận với trạm có gói tin RTS tương ứng, do đó xác suất đụng độ thấy hơn so với DCF cơ bản. Quá trình được minh họa tại hình 1-9.



Hình 1-9 Truy cập kênh truyền DCF với RTS/CTS

1.3.3 Cơ chế điều khiển truy cập môi trường truyền PCF

Bên cạnh điều khiển truy cập DCF còn có một phương thức truy cập kênh truyền khác là bình bầu (polling) - PCF. Chức năng cộng tác phân tán DCF chỉ hỗ trợ dịch vụ truyền khung đáng tin cậy, không đảm bảo khoảng thời gian một thiết bị phải chờ cho đến khi truyền được gói tin dữ liệu. PCF cung cấp khả năng truy cập môi trường truyền cho các dich vụ giới hạn thời gian. Sử dụng PCF đòi hỏi phải có một điểm truy cập - AP (Access point) để kiểm soát truy cập môi trường truyền và chỉ huy các trạm trong mạng không dây. Với PCF thì AP sẽ hỏi tuần tự các trạm xem có gì muốn truyền hay không. Nếu có dữ liệu truyền thì máy trạm sẽ trả lời với AP, nếu không có dữ liệu truyền thì máy trạm sẽ gửi một khung dữ liệu null và chuyển poll qua máy kế tiếp. Nếu qua lượt thì cho dù có dữ liệu cũng phải đợi tới lượt kết tiếp. Vì vậy kỹ thuật này không được sử dụng trong mạng Ad-hoc. Trong kỹ thuật truy cập PCF, thời gian được chia thành các khoảng, được gọi là super frame. Mỗi super frame gồm các khoảng tranh chấp (contention period) và các khoảng không tranh chấp CF (contention-free period).



Hình 1-10 Chức năng cộng tác điểm PCF

1.3.4 Giao thức MAC theo chuẩn 802.11 (CSMA/CA,+ACK, +RTS/CTS)

Như đã đề cập ở phần trên, chuẩn 802.11 quy định bắt buộc DCF sử dụng gói tin ACK cho quá trình tránh đụng độ. Tuy nhiên trên thực tế để tăng cường khả năng tránh đụng độ cũng như vấn đề trạm ẩn thì cơ chế RTS/CTS được khuyến nghị sử dụng thêm nhưng không phải là giải pháp tuyệt đối cho vấn đề trạm ẩn. Mặc dù vậy chuẩn 802.11 không phải lúc nào cũng sử dụng RTS/CTS vì chi phí cho gói tin RTS/CTS là khá cao, do đó chuẩn 802.11 quy định một ngưỡng cho phép sử dụng RTS/CTS, khi nào kích thước gói tin vượt ngưỡng này thì RTS/CTS mới được kích hoạt, nếu không thì frame dữ liệu sẽ được gửi luôn.

Cơ chế 802.11 RTS/CTS có thể giải quyết vấn đề trạm lộ, tuy nhiên chỉ khi nút nhận và nút trạm đồng bộ hóa tức là kích cỡ gói tin và tốc độ truyền giống nhau thì mới xử lý được, ngược lại vấn đề trạm lộ có thể xảy ra.

1.4 Các kiểu tấn công mạng WLAN theo chuẩn 802.11

Khi mạng WLAN hoạt động không ổn định, ngoài các nguyên nhân gây tắc nghẽn do tính tự nhiên phát sinh xung đột của hệ thống, thì còn có nguyên nhân khác tác động trực tiếp làm thay đổi hoạt động mạng gây ra tắc nghẽn cục bộ hoặc toàn phần hệ thống mạng. Đó chính là tác nhân tấn công do người dùng có chủ đích.

Các cuộc tấn công có nhiều phương thức khác nhau. Tấn công mạng có thể xảy ra ở những lớp mạng khác nhau.

- Ở lớp vật lý (PHY) tấn công can thiệp tỉ lệ SNR và làm cản trở truyền thông thậm chí là không thể truyền thông.
- Ở lớp truy cập môi trường truyền (MAC), tấn công làm tăng cửa sổ tranh chấp, giảm cơ hội truyền của các trạm.
- Ở lớp Network và Transport, kẻ tấn công chèn vào các gói tin lỗi, phá hủy các gói tin định tuyến, buộc giá trị của cửa sổ gửi (cửa sổ tắc nghẽn - congestion windows) ở giá trị thấp làm cho máy trạm tưởng bị tắc nghẽn.

Một số kiểu tấn công đáng chú ý là tấn công từ chối dịch vụ, tấn công dựa trên phân tích lưu lượng, tấn công dữ liệu riêng tư, tấn công vật lý.

Trong luận văn này tôi tập trung vào vấn đề tắc nghẽn ở lớp truy cập môi trường MAC, đồng thời tập trung nghiên cứu các phương thức cải tiến DCF

để giảm mức độ ảnh hưởng của mạng khi bị tấn công và cân bằng lại hệ thống giúp trở về trạng thái ổn định ban đầu.

1.5 Các mục tiêu nghiên cứu chính của luận văn.

- Nghiên cứu, nắm vững kiến thức căn bản và chuyên sâu về mạng cục bộ không dây. Nắm được cấu trúc cơ bản cũng như kiến trúc của mạng cục bộ không dây WLAN.
- Nghiên cứu các nguyên nhân gây tắc nghẽn (jamming) cũng như các phương pháp phát hiện tắc nghẽn và giải pháp khắc mục tắc nghẽn ở tầng vật lý (anti-jamming).
- Sử dụng bộ mô phỏng NS2 thực hiện mô phỏng đánh giá hiệu quả của các giải pháp khắc phục tắc nghẽn ở tầng vật lý.

CHƯƠNG 2 – PHÂN TÍCH PHƯƠNG PHÁP TẤN CÔNG GÂY NGHẼN

Để hiểu được hiệu quả của việc gây tắc nghẽn trên mạng WLAN 802.11 chúng ta cần hiểu cách thức hoạt động của các thiết bị gây nghẽn khi đụng độ xảy ra.

Như đã đề cập ở chương trước để giảm đụng độ gây tắc nghẽn mạng, chuẩn 802.11 cung cấp chức năng cộng tác phân tán DCF, trong đó có cơ chế trì hoãn theo hàm mũ (exponential backoff) được sử dụng, giúp trạm gửi giảm tần suất phát lại gói tin dựa trên tần suất đụng độ.

Để phân tích, đánh giá các phương pháp tấn công gây tắc nghẽn chúng ta cần theo dõi trạng thái của hệ thống mạng tại nhiều thời điểm khác nhau, để phát hiện kịp thời khi tắc nghẽn xảy ra.

Mô hình chuỗi Markov rất phù hợp để áp dụng vào việc phân tích, đánh giá hệ thống. Trong chương 2, 3 có sử dụng thông số thực nghiệm theo tài liệu tham khảo [10]. Trước tiên chúng ta sẽ tìm hiểu về jamming và kẻ tấn công jammer cũng như các phương thức jamming.

2.1 Jammer và mô hình tấn công jamming

Chúng ta định nghĩa jammer là một thực thể cố gắng can thiệp vào quá trình gửi và nhận ở tầng vật lý của truyền thông không dây. Để can thiệp vào quá trình truyền thông không dây, jammer có thể đạt được bằng các ngăn không cho thiết bị phát truyền thông tin hoặc cản trở thiết bị nhận thông tin hợp lệ, có ý nghĩa.

- Với mục đích đánh giá mức độ ngăn cản hoặc làm khó khăn với thiết bị truyền thì chúng ta có thể xác định hành vi jamming qua đơn vị tính là tỉ lệ gửi gói tin - PSR (**Packet Send Ratio**). Ví dụ một thiết bị A muốn gửi n thông điệp nhưng chỉ có m thông điệp có thể gửi đi, như vậy tỉ lệ PSR = $\frac{m}{n}$.
- Với mục đích đánh giá mức độ cản trợ hoặc gây khó khăn với nút nhận, chúng ta có thể xác định hành vi jamming qua đơn vị tính là tỉ lệ gói tin đến đích thành công – PDR (**Packet Delivery Ratio**). PDR được tính bằng tỉ lệ giữa gói tin không bị lỗi, nghĩa là giá trị trường CRC là hợp lệ, với tổng số gói tin nhận được.

Có rất nhiều chiến lược tấn công gây tắc nghẽn (jamming) có thể áp dụng để tấn công; theo các tác giả [2,6] có thể phân ra làm 4 loại jamming như sau:

- **Constant jamming**: đây là kiểu tấn công mà kẻ tấn công (jammer) liên tục phát ra tín hiệu, đẩy vào môi trường tín hiệu ngẫu nhiên không tuân theo quy định của giao thức MAC một cách liên tục và duy trì với tần suất cao.
- **Deceptive jamming**: thay vì sử dụng các bit ngẫu nhiên (các gói tin không theo chuẩn giao thức MAC được phát sinh ngẫu nhiên theo thuật toán random), kẻ tấn công (deceptive jammer) liên tục gửi các gói bình thường vào môi trường không có khoảng cách giữa các gói làm các trạm liên lạc bị lừa là có gói tin hợp pháp cần nhận nên luôn duy trì trạng thái nhận. Điều này làm cản trở các người gửi thật sự muốn gửi thông tin đến người nhận.
- **Random jamming**: tương tự như deceptive jamming, nhưng thay vì gửi liên tục, kẻ tấn công (random jammer) sẽ có một khoảng thời gian "thức" để thực hiện jamming, sau đó sẽ tắt sóng chuyển sang chế độ "ngủ" một khoảng thời gian ngẫu nhiên hoặc được thiết lập. Sau một khoảng thời gian "ngủ" jammer lại tiếp tục jamming, quá trình được lặp đi lặp lại. Thường thì chế độ nay hay được áp dụng vào những nút jammer có giới hạn mức năng lượng.
- **Reactive jamming**: khác hẳn với 3 loại jamming ở trên. Thay vì giữ môi trường truyền thông luôn bận thì kẻ tấn công (reactive jammer) chỉ tấn công khi cảm nhận có sự truyền thông trên môi trường truyền. Ví dụ như tấn công gói tin CTS khi thấy gói RTS trên môi trường, hay là tấn công gói ACK không cho gói ACK về với bên gửi. Ngoài ra còn có các phương pháp: gây nghẽn bằng cách làm hỏng gói tin biên nhận (ACK corruption jamming), gây nghẽn bằng cách làm hỏng gói dữ liệu (DATA corruption jamming), gây nghẽn trong khoảng thời gian chờ DIFS (DIFS wait jamming) [7].

Trong luận văn này tôi tập trung nghiên cứu vào reactive jamming hay còn gọi là intelligent jamming vì cơ chế jamming khá hiệu quả và khó phát hiện.

2.2 Sử dụng mô hình chuỗi Markov cho cơ chế DCF

Chức năng cộng tác phân tán DCF cơ bản có thể được mô hình hóa theo chuỗi Markov. Ta chọn chuỗi Markov để mô hình hóa là do nó có thể mô

hình hóa các trạng thái của thực thể giao thức một cách rõ ràng, ngắn gọn, dễ hiểu. Ta cũng có thể đưa vào mô hình này các trạng thái, sự chuyển trạng thái, bao gồm sự kiện (event) gây ra sự chuyển trạng thái và các phản ứng (action) của thực thể giao thức.

Khi một trạm muốn truyền gói tin thì trước tiên cần phải cảm nhận môi trường truyền ít nhất một khoảng thời gian DIFS = 50 µs [3]. Sau khoảng thời gian DIFS này thì 1 bộ đếm ngược backoff có giá trị ngẫu nhiên nằm trong khoảng (0, $W_0 - 1$) được khởi động đếm ngược về 0, với W0 là giá trị của cửa sổ tranh chấp trong trạng thái backoff đầu tiên. Bộ đếm backoff sẽ dừng khi thấy môi trường truyền bận và tiếp tục đếm ngược nếu như môi trường sẵn sàng cho phép truyền (môi trường truyền rỗi).

Ký hiệu $b_{i,k}$ biểu diễn trạng thái của trạm khi phát lại một gói tin lần thứ i, có giá trị backoff time là k.Trong thời gian đếm ngược này, cứ mỗi khoảng thời gian bằng giá trị aSlotTime = 20 µs trạm xét trạng thái backoff $b_{i,k}$ ($k \neq 0$) cho tới khi đạt tới trạng thái $b_{i,0}$. Khi bộ đếm backoff đạt tới giá trị = 0 (k=0) thì trạm gửi gói tin đi. Nếu trạm đích nhận được gói tin chính xác toàn vẹn, trạm sẽ đợi một khoảng thời gian SIFS = 10 µs rồi gửi lại gói tin ACK.

Gói tin được coi là bị lỗi nếu như trạm truyền không nhận được gói tin ACK trả về. Khi gói tin bị lỗi trạm gửi tăng giá trị cửa sổ tranh chấp lên W_1 và bộ đếm backoff chọn giá trị ngẫu nhiên trong khoảng (0, W_1-1) và tiếp tục lại quá trình cảm nhận môi trường rồi truyền gói tin lại như ban đầu, giá trị cửa sổ tranh chấp tăng dần khi lỗi xảy ra liên tục với giá trị $Wi = 2^i W_0$. Như vậy cửa sổ tranh chấp tăng theo hàm mũ cơ số 2 của số lần lỗi do có sự đụng độ các gói tin.

Khi trạm gửi đạt tới trạng thái backoff cuối, tức là khi gói tin truyền thành công hoặc trạm đạt ngưỡng truyền lại tối đa – gói tin sẽ bị loại bỏ (drop). Khi gói tin truyền thành công thì DCF trở lại trạng thái zero backoff.

Các tham số được các tác giả của [10] sử dụng trong thực nghiệm được mô tả theo bảng 2-1:

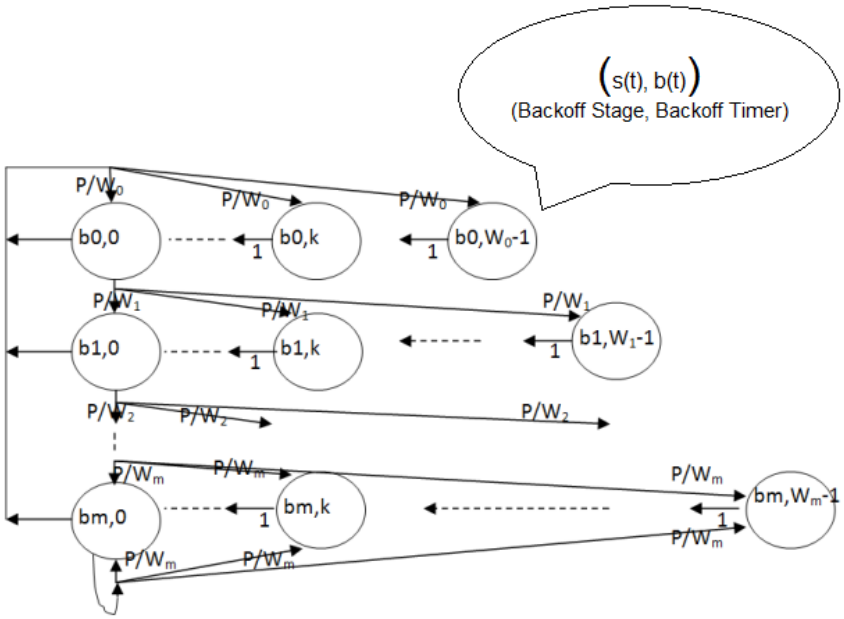
Các tham số	**Giá trị tham số**
aSlotTime	20 µs
SIFS	10 µs

DIFS	SIFS + 2x slot time = 50 µs
MAC frame size	512 bytes
Station transmission rate	11 Mbps
PHY header duration	192 µs
Jammer packet length	128 bytes
Jammer transmission rate	1 Mbps

Bảng 2-1 Các tham số thực nghiệm

Đây là cách mà chúng ta mô hình hóa DCF theo Bianchi Model [9], chuỗi Markov được mô tả theo hình 2-1. Trên hình, tên trạng thái cụ thể được ghi trong hình tròn; nhãn ghi cạnh các mũi tên chỉ sự chuyển trạng thái theo kiểu: even/action. Event là sự kiện gây ra sự chuyển trạng thái, action là các hành động của thực thể giao thức khi xảy ra sự kiện.

Giả sử mỗi gói tin có thể được phép truyền lại vô hạn lần; như vậy, một trạm gửi chỉ trở về trạng thái zero backoff khi gói tin truyền thành công.



Hình 2-1 Mô hình hóa DCF theo chuỗi Markov

Ký hiệu b_{ik} biểu diễn xác suất một trạm ở trạng thái $b_{i,k}$ và ký hiệu P là xác suất một gói tin bị lỗi, với m là trạng thái backoff tối đa. Hai đại lượng này có mối tương quan như sau $\mathbf{b_{ik} = P[b_{i-1,k}]}$; Nghĩa là trạm chuyển sang trạng thái i với backoff-time bằng k (ký hiệu là $b_{i,k}$) từ trạng thái $b_{i-1,k}$ và xảy ra lỗi gói tin.

Ta có phân phối trạng thái của chuỗi Markov với $b_{i,k}$ là giới hạn theo thời gian của trạng thái backoff của nút $b_{ik} = \lim_{t\to\infty} P\{s(t) = i, b(t) = k\}, i \in (0, m), k \in (0, W_i - 1)$.

Dựa vào hình 2-1 tác giả [10] xây dựng các biểu thức:

$$b_{10} = Pb_{00} \qquad (2.1)$$

Tôi (tác giả luận văn) có thể giải thích chi tiết ý nghĩa của phương trình (2.1) như sau:

b_{00} là xác suất thực thể giao thức MAC ở trạng thái truyền gói tin lần đầu (lần 0), chưa backoff lần nào; Theo định nghĩa cách ghi ký hiệu $b_{i,k}$ bên trên thì i=0, k=0.

b_{10} là xác suất thực thể giao thức MAC ở trạng thái truyền lại gói tin lần thứ 1 (đếm từ 0), chưa backoff lần nào; Theo định nghĩa cách ghi ký hiệu $b_{i,k}$ bên trên thì i=1, k=0.

Còn P là xác suất gói tin bị lỗi, như đã định nghĩa ở trên.

Như vậy, có thể dễ dàng suy ra rằng: khi truyền gói tin lần đầu (lần 0), ngay sau khoảng thời gian chờ DIFS đầu tiên, và nếu gói tin truyền bị lỗi (vế phải của (2.1) là biểu diễn toán học của phát biểu này), thì thực thể giao thức MAC phải truyền lại gói tin lần thứ 1 với cửa sổ backoff được giữ nguyên không thay đổi, vẫn bằng 0 (vế trái của (2.1) là biểu diễn toán học của phát biểu này).

Với các lập luận tương tự, có thể suy ra:

$$b_{20} = Pb_{10} = P^2b_{00} \qquad (2.2)$$

$$b_{(m-1)0} = P^{m-1}b_{00} \qquad (2.3)$$

$$b_{(m-1)0} = (1-p)b_{m0} \qquad (2.4)$$

Việc giải thích chi tiết ý nghĩa của phương trình (2.2), (2.3) và (2.4) tương tự như việc giải thích (2.1), vì vậy tôi không trình bày lại.

Với lập luận rằng nếu trạm đang ở trạng thái $b_{(m-1)0}$ mà gói tin không bị lỗi (xác suất là 1-P) thì trạm vẫn ở trạng thái $b_{(m-1)0}$. Chúng ta sẽ có:

$$(b_{(m-1)0})(1-P) = b_{(m-1)0}, \text{ suy ra } b_{m0} = Pb_{(m-1)0} + Pb_{m0};$$

Qua một số phép biến đổi (thay thế từ 2.3 và 2.4), có thể suy ra

$$b_{m0} = \frac{P^{m-1}}{1-P} b_{00} \qquad (2.5)$$

Với giả thiết giá trị của cửa sổ tranh chấp (khi thực hiện backoff) là một số nguyên được chọn ngẫu nhiên trong miền (0, W_i) và giảm đi 1 sau mỗi đơn vị thời gian slottime, có thể dễ dàng suy ra rằng:

$$b_{ik} = \frac{W_i - k}{W_i} b_{i0} \qquad (2.6)$$

Có thể giải thích chi tiết hơn việc suy ra (2.6) như sau:

Khi thực thể giao thức MAC truyền lại một gói tin đến lần thứ i, sau k slottime, cửa sổ tranh chấp ban đầu có giá trị là W_i giảm k đơn vị slottime xuống còn W_i-k. Nhưng vì w_i được chọn ngẫu nhiên trong miền (0, W_i), nên tính trung bình, việc truyền lại gói tin lần thứ i phải backoff (trì hoãn sau khoảng thời gian DIFS) một khoảng thời gian là $\frac{W_i - k}{W_i}$, tính theo đơn vị slottime. Lập luận này được biểu diễn dưới dạng toán học bằng phương trình (2.6).

Từ 2.5 và 2.6 tất cả giá trị của $b_{i,k}$ được biểu diễn là hàm của b_{00}.Với điều kiện bão hòa tức là các trạm đều luôn luôn có gói tin trong hàng đợi cần gửi, thì tổng của các trạng thái backoff là 1, theo [10] ta có biểu thức cho b_{00} như sau:

$$\sum_{i=0}^{i=m} \sum_{k=0}^{k=W_i} b_{ik} = 1 \qquad (2.7)$$

Theo tôi, có thể giải thích việc đưa ra phương trình (2.7) một các dễ dàng, dựa trên lập luận như sau:

Tổng Sigma thứ nhất (tổng theo k) là tổng xác suất thực thể giao thức MAC phát lại gói tin đến lần thứ i.

Tổng Sigma thứ hai (tổng theo i) là tổng xác suất thực thể giao thức MAC phát lại gói tin thành công. Điều này là dễ hiểu, bởi vì các tính toán đã được thực hiện với giả thiết rằng: số lần truyền (phát) lại gói tin là vô hạn, nghĩa là truyền đi truyền lại cho đến khi thành công.

Thay (2.6 vào (2.7) ta có

$$\sum_{i=0}^{i=m} b_{i0} \frac{W_i+1}{2} = 1 = \frac{b_{00}}{2}\left[W\left(\sum_{i=0}^{m-1}(2P)^i + \frac{(2P)^m}{1-p}\right) + \frac{1}{1-P}\right] \quad (2.8)$$

Từ đây suy ra được $b_{00} = \frac{2}{\sum_{i=0}^{m-1} P^i(W_i+1) + \frac{P^m}{1-P}(W_m+1)}$ (2.9)

Gọi τ là xác suất truyền của một nút ta có thể suy ra từ b_{00}

$$\tau = \sum_{i=0}^{m} b_{i0} = \frac{b_{00}}{1-P} \quad (2.10)$$

và p_c là xác suất một gói tin bị đụng độ với một nút khác đang truyền theo Bianchi model [9].

$$p_c = 1 - (1 - \tau)^{N-1} \quad (2.11)$$

Trong đó N là tổng số nút trong vùng phủ sóng của nút ta xét.

Khi có sự xuất hiện của jammer, gói tin truyền thất bại có thể do 2 nguyên nhân: do jammer hoặc do đụng độ truyền tin thông thường. Như vậy P là xác suất gói tin truyền không thành công được xác định bởi xác suất của jammer Q cùng với xác suất đụng độ p_c. Giả định jammer không xác định được đụng độ chỉ quyết định tấn công bằng cách tăng mức năng lượng trong môi trường truyền.

$$P = Q + p_c - Q.p_c$$

$$= p_c + (1 - p_c)Q \quad (2.12)$$

Gọi P_f,P_s là xác suất điều kiện của gói tin truyền thất bại và thành công

$$P_f = P \text{ và}$$

$$P_s = 1 - P_f = 1 - P \quad (2.13)$$

Gọi S là xác suất gói tin truyền thành công (Successful) và F là xác suất gói tin truyền thất bại (Fail), thì

$$S = N\tau(1 - \tau)^{N-1}(1-Q) \quad (2.14)$$

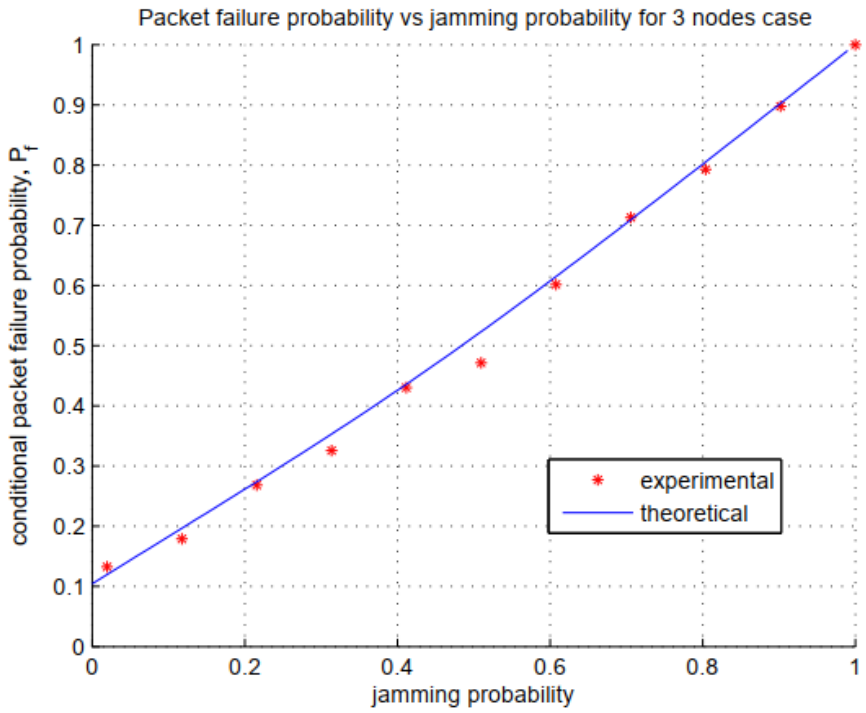
$$F = N\tau(1 - \tau)^{N-1}Q + [1 - (1 - \tau)^N - N\tau(1 - \tau)^{N-1}] \quad (2.15)$$

Với các tham số đã đưa ở bảng 2-1 tác giả [10] thực nghiệm với 5 trạm. Một trạm đóng vai trò jammer, 3 trạm dùng để truyền gói tin với 500 gói tin

và để mô tả tình trạng bão hòa (1 lượng lớn gói tin trong hàng đợi), trạm thứ 5 đóng vai trò trạm nhận gói tin từ 3 trạm phát. Tổng gói tin lỗi và tổng gói tin truyền đi được ghi lại. Xác suất điều kiện gói tin bị lỗi P_f thực nghiệm được tính như sau:

$$P_f = \frac{\#gói\ lỗi}{\#\ tổng\ các\ gói\ tin} \quad (2.16)$$

Dựa trên hình 2-2 theo tài liệu tham khảo [10] ta có thể quan sát thấy các số liệu tính toán và số liệu thực tế có kết quả tương đương chênh lệch không nhiều.



Hình 2-2 Xác suất gói tin lỗi và xác suất tắc nghẽn

2.3 Xây dựng biểu thức tính thông lượng cho cơ chế DCF

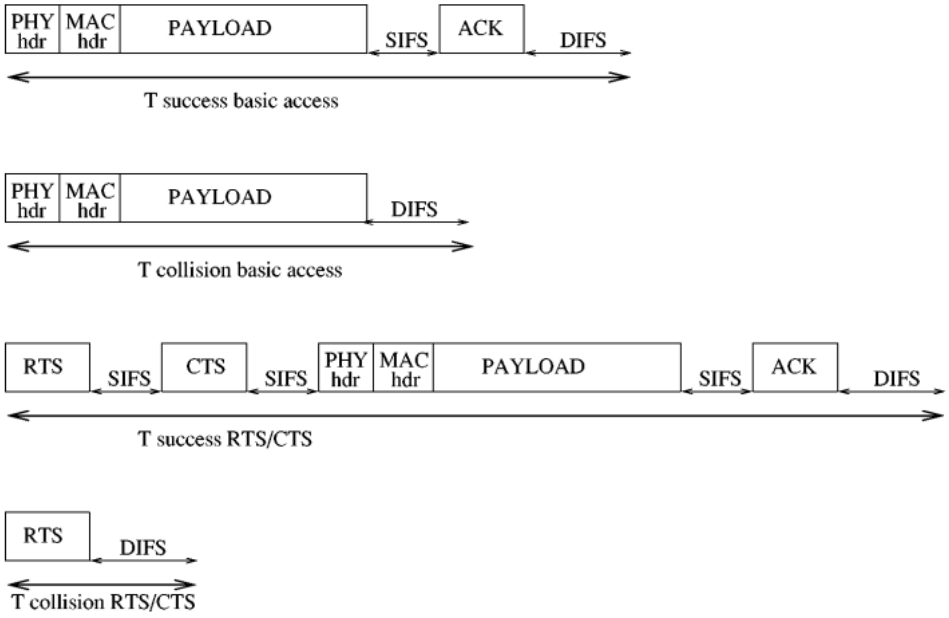
Để đánh giá độ tắc nghẽn của một hệ thống mạng có nhiều thông số. Một trong các thông số quan trọng chính là thông lượng. Để tính thông lượng chúng ta xem xét tại một slot time và tính xác suất truyền thành công S, xác suất truyền hỏng F hoặc xác suất môi trường truyền dẫn sẵn sàng, bằng 1-S-F.

Nếu bắt đầu 1 slot time, gói tin truyền thành công thì bộ đếm backoff sẽ dừng trong 1 khoảng T_s, trong khoảng thời gian T_s này thì có E_p slot được dùng để truyền data. Tương tự nếu bắt đầu 1 slot time bằng một gói tin bị đụng độ thì bộ đếm backoff sẽ dừng trong 1 khoảng T_c. Nếu slot time không có hoạt động gì thì mọi việc lại tiếp tục như vậy ở slot time tiếp theo.

Thông lượng của hệ thống được xây dựng như sau:

$$\Psi = \frac{SE_p}{(1-S-F)+FT_c+ST_s} \quad (2.17)$$

Ep: chiều dài trung bình gói tính bằng slot time.



Hình 2-3 Ts và Tc

T_c, T_s: thời gian trung bình tính bằng slot time của việc truyền gói bị lỗi hoặc gói truyền thành công, chúng được tính dựa vào hình 2-3 [9]:

T_s = Thời gian truyền gói thành công + SISF + ACK +DIFS

T_c = thời gian truyền gói thất bại + DIFS

Tiếp tục phần thực nghiệm ở mục 2.1, thông lượng thực nghiệm được tính theo công thức [10]:

$$\Psi = \frac{\#số\ gói\ thành\ công\ x\ E_p}{Tổng\ thời\ gian} \quad (2.18)$$

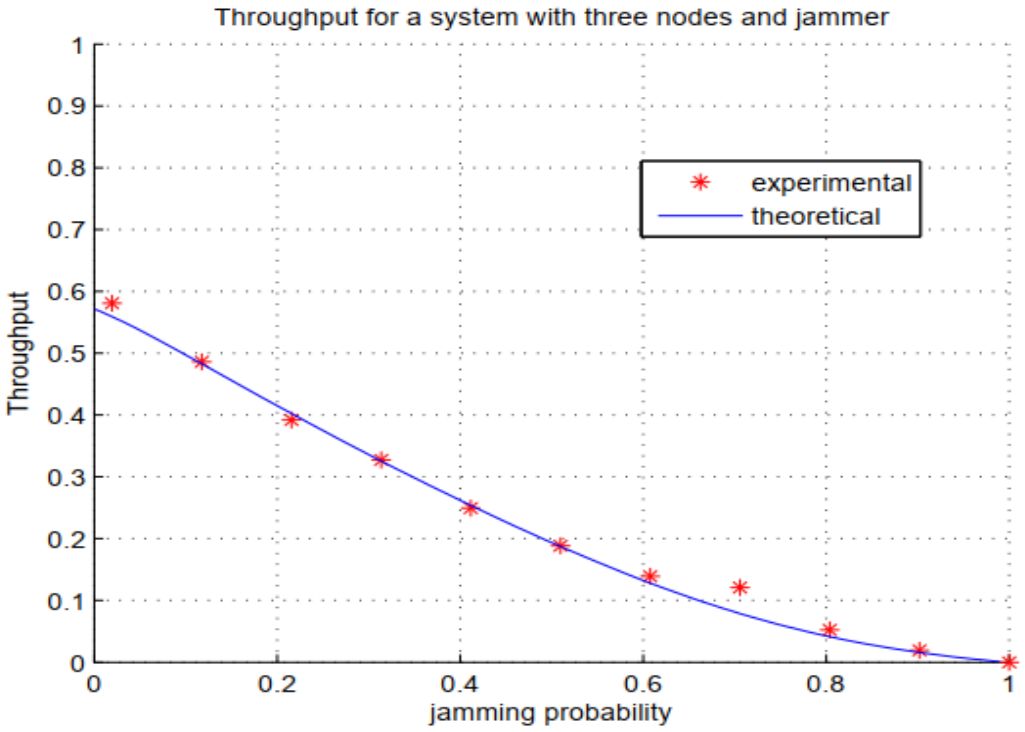
Với 3 trạm truyền dữ liệu với kích cỡ frame là 512 bytes, 500 gói tin được gửi từ mỗi trạm (mục đích tạo điều kiện bão hòa của hệ thống). Hai mốc thời gian được dùng để tính thông lượng mỗi trạm là lúc bắt đầu và kết thúc. Thông lượng thực tế được tác giả [10] tính toán như sau:

$$\text{Ep} = \frac{(MAC\ frame\ size\ /11M + 192\mu s)}{slot\ time} \quad (2.19)$$

$$T_s = \frac{(E_p + DIFS + ACK + SIFS)}{slot\ time} \quad (2.20)$$

$$\text{T}_c = \frac{\frac{Q}{P_f}\ thời\ gian\ gói\ tin\ jammer + \frac{P_c}{P_f}\ thời\ gian\ MAC\ frame}{slot\ time} \quad (2.21)$$

Theo quan sát từ hình 2- 4 [10], kết quả thử nghiệm giống như mong đợi ta thấy khi xác suất tắc nghẽn Q là 1 thì thông lượng tiến về zero. Xác suất gói tin thất bại P_f về 1 thì xác suất tắc nghẽn Q cũng tăng. So sánh kết quả thực tế và lý thuyết ta thấy kết quả khá tương đồng.



Hình 2-4 Thông lượng thực nghiệm và lý thuyết

2.4 Phân tích sự tiêu hao năng lượng của nút mạng tấn công kiểu Jamming

Một yếu tố gắn liền với tấn công kiểu jamming chúng ta cần lưu tâm đó chính là vấn đề mức năng lượng trung bình mà jammer sử dụng tỉ lệ với xác suất tắc nghẽn mà nó gây ra. Ta cũng biết một số mạng không dây cảm nhận tắc nghẽn bằng mức năng lượng trung bình trên kênh truyền. Một số cơ chế xây dựng biểu đồ mức năng lượng của mạng bình thường, khi thấy mức năng lượng vượt mức trung bình có thể cho là tắc nghẽn đã xảy ra. Do đó phân tích sự tiêu hao năng lượng của nút tấn công jamming cũng là một vấn đề cần quan tâm.

Giả sử thiết bị jammer được xây dựng với khả năng phát hiện được việc truyền và đụng độ. Mức năng lượng trung bình mà thiết bị jammer sử dụng tỉ lệ thuận với xác suất xảy ra jamming ký hiêu là C, C được ràng buộc theo điều kiện $C <= C_0$, với C_0 là một hằng số cụ thể được khảo sát dưới đây [10].

- Giả sử jammer không thể phát hiện đụng độ, xác suất tắc nghẽn. Mức năng lượng trung bình C được tính như sau:

$$C = Pr[\text{ít nhất 1 nút truyền trên mạng}]Q$$

$$= (1 - (1 - \tau)^{N-1})Q \quad (2.22)$$

  Như đã được giải thích về (2.10) và (2.11), trong (2.22) τ là xác suất truyền của một nút, N là tổng số nút trong vùng phủ sóng của nút ta xét. Xin nhắc lại tại đây, Q là xác xuất tắc nghẽn (jamming probability), nghĩa là xác suất xảy ra sự kiện khi jammer truyền gói tin thì đồng thời có một nút mạng cũng truyền, do đó gói tin nút mạng truyền đi bị hỏng – việc truyền bị thất bại.

- Giả sử jammer có thể phát hiện đụng độ, thì xác suất có điều kiện Q là xác suất thiết bị jammer đẩy gói tin gây tắc nghẽn vào mạng đúng lúc có một nút truyền tải.

$$C = Pr[\text{chỉ có 1 nút truyền trên mạng}]Q$$

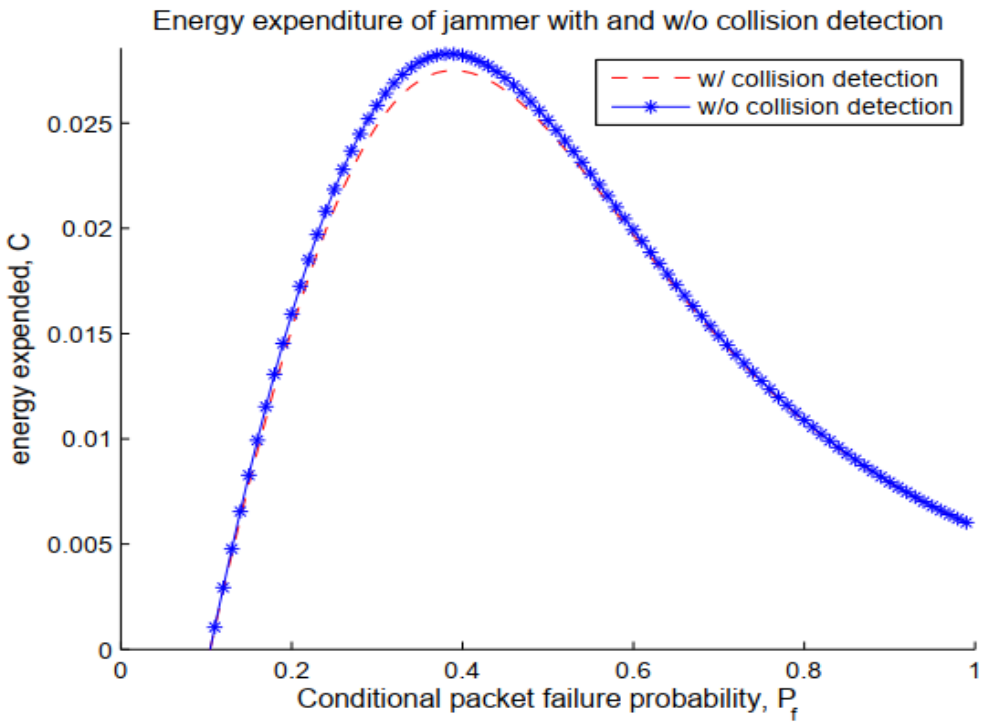
$$= N\tau(1 - \tau)^{N-1}Q \quad (2.23)$$

Theo kết quả từ hình 2-5 theo tài liệu tham khảo [10] ta thấy mức năng lượng tiêu hao của jammer tăng nhanh từ thấp đến cao theo xác suất tắc nghẽn, rồi giảm dần. Có nghĩa là thiết bị jammer ép các nút mạng sử dụng cơ

chế DCF sử dụng cửa sổ tranh chấp có giá trị tối đa, dẫn đến làm giảm xác suất truyền, đồng thời điều đó dẫn đến việc độ trễ giữa các lần truyền lại cũng gia tăng. Khi số lần truyền bị giảm dẫn đến số lần can thiệp (gây nhiễu, jamming) của thiết bị jammer cũng giảm, cho nên mức năng lượng tiêu thụ của nó (jammer) cũng ít đi.

Khi so sánh xác suất gói tin bị lỗi (P_f) với mức năng lượng sử dụng của jammer (C), chúng ta có thể thấy năng lượng khi có phát hiện đụng độ nhỏ hơn so với trường hợp không phát hiện đụng độ.

Như đã trình bày ở trên ta thấy mức năng lượng lúc bắt đầu jamming của jammer tăng lên nhanh, có nghĩa là xác suất tắc nghẽn tiến lên 1, thông lượng của hệ thống giảm dần về 0. Mức năng lượng của jammer trong giai đoạn này tỉ lệ nghịch với thông lượng. Tuy nhiên khi thông lượng về 0, tắc nghẽn ở mức cao, cửa sổ tranh chấp tăng dẫn đến ít lưu lượng được phát sinh nhằm giải quyết tắc nghẽn, dẫn đến năng lượng tiêu thụ của jammer giảm dần. Trong giai đoạn này có thể thấy năng lượng tiêu thụ của jammer tỉ lệ thuận với thông lượng.



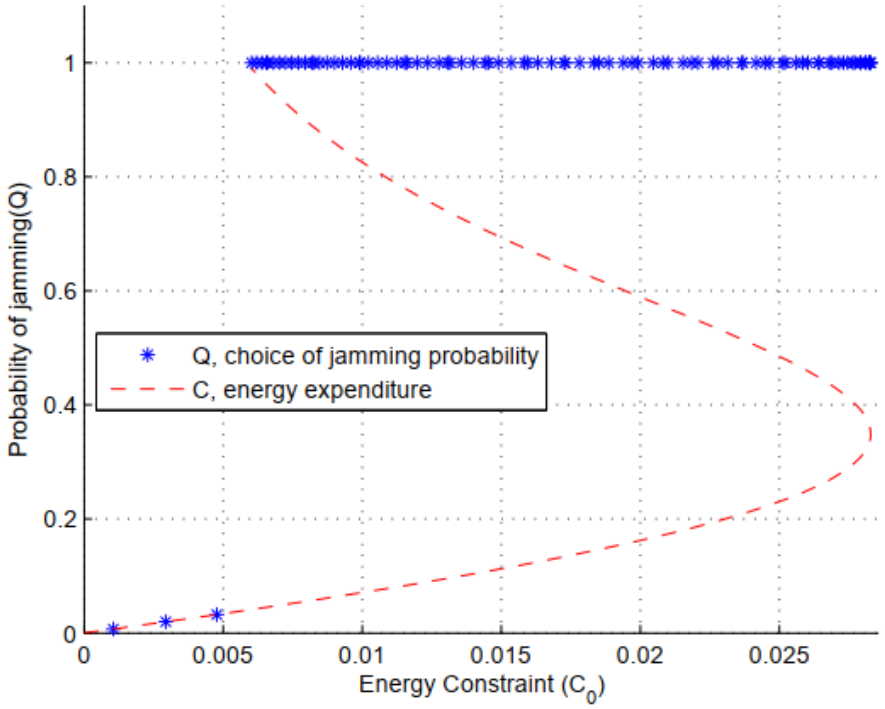
Hình 2-5 Tương quan năng lượng sử dụng với xác suất gói tin lỗi của jammer

2.5 Phân tích ảnh hưởng lên thông lượng

Quay lại hình 2-4, ta có thể quan sát được thông lượng là một hàm đơn điệu giảm của xác suất tắc nghẽn (Q), khi Q tiến về 1 thì thông lượng giảm dần tiến về 0. Tại hình 2-5 thì năng lượng tiêu hao tăng rồi giảm với xác xuất gói tin lỗi do xác xuất tắc nghẽn gây ra.

Theo hình 2-6, tác giả [10] đưa ra các nhận xét. Nếu $C_0 > C(1)$, với C(1) là giá trị của C khi Q = 1, Q luôn chọn là 1 vì xác suất tắc nghẽn càng cao thì mức năng lượng tiêu thụ và thông lượng càng thấp. Khi chọn xác suất tắc nghẽn là 1, thì jammer ép DCF sử dụng cửa sổ tranh chấp tối đa và làm tăng thời gian chờ giữa các lần truyền tin. Điều này dẫn đến giảm thời số lượng truyền của jammer trong một khoảng thời gian nhất định và do đó giảm mức năng lượng tiêu thụ của jammer.

Chỉ khi $C_0 < C(1)$, jammer mới chọn 1 xác xuất nhỏ ở phía tăng đường cong. Như vậy DCF chuẩn đem lại lợi ích nhất định về mặt tiêu thụ năng lượng cho jammer.



Hình 2-6 Lựa chọn xác suất tắc nghẽn với hạn chế năng lượng

CHƯƠNG 3 – PHÂN TÍCH KẾ HOẠCH CHỐNG TẤN CÔNG KIỂU GÂY NGHẼN

Mạng không dây WLAN 802.11 được xây dựng dựa trên mô hình chia sẻ môi trường chung, điều này dẫn đến rất dễ bị tấn công bởi phương thức gây tắc nghẽn jamming. Những cuộc tấn công này có thể dễ dàng xảy ra bằng cách can thiệp tín hiệu radio không theo chuẩn giao thức MAC. Những cuộc tấn công này có thể gây ảnh hưởng nghiêm trọng vào hệ thống mạng cục bộ không dây, do đó các cơ chế đối phó là cần thiết.

Trong quá trình nghiên cứu có thể thấy độ mạnh tín hiệu hoặc thời gian cảm nhận sóng mang không thể khẳng định hay phát hiện sự tồn tại của thiết bị jammer hay không. Ngoài ra xung đột xảy ra bởi jamming hay do xung đột thông thường cũng rất khó để phân biệt. Trong chương này tác giả [10] sửa cơ chế của DCF để chống tấn công kiểu Jamming.

3.1 Phát hiện sự nghẽn mạng (Dectection of Jamming)

Phát hiện sự nghẽn mạng đóng vai trò rất quan trọng vì đây là bước đầu tiên để khắc phục hệ thống cũng như duy trì tính ổn định của hệ thống mạng.

Để phát hiện jamming thì cần theo dõi các trạng thái của hệ thống. Giả định các trạm phát biết được tổng số các trạm đang hoạt động trên hệ thống. Sau khi truyền một số gói tin, mỗi trạm sẽ kiểm tra số lần ở trạng thái truyền tin $b_{0,0}$,$b_{1,0...}$ và cố gắng phát hiện sự hiện diện của jamming bằng cách đạt được trạng thái π. Trong đó π là một vecto có các phần tử là tập hợp số lần vào trạng thái backoff: $nb_{i,0}$.

$$\pi = n[b_{0,0},b_{1,0},...b_{i,0}]$$

Vấn đề của việc phát hiện jamming khi có đụng độ thông thường chính là kiểm tra giả thuyết tổng hợp, cách thức phổ biến được dùng để giải quyết vấn đề này là kiểm tra tỉ lệ khả dĩ phổ quát (GLRT - generalized likelihood ratio test) [8]. Theo tác giả [10] có 2 giả thuyết H_0 và H_1 được đưa ra:

H_0 = không có jamming thì Q = 0

H_1 = có jamming thì $0 < Q \leq 1$

Xác suất trạng thái có thể đạt được ở cả 2 giả thuyết H_0 và H_1 khi có jamming và không có jamming, được xác định theo biểu thức (3-1). Mức ngưỡng η được chọn trong GLRT phải đáp ứng được yêu cầu xác suất báo động giả P_{FA}. P_{FA} là xác suất máy dò xác nhận H_1 đúng khi H_0 đúng. Chúng

ta không biết xác suất của jammer, GLRT ước tính xác suất chuyển P bằng cách tối đa hóa xác suất có điều kiện khi ở trạng thái π. Xác suất chuyển P là xác suất có điều kiện của gói tin lỗi. Nó bao gồm xác suất đụng độ thường xuyên p_c và xác suất tắc nghẽn Q.

Ta giả định ta biết được số trạm trên hệ thống, biết được xác suất đụng độ các gói tin trong điều kiện bão hòa khi không có sự tham gia của jammer là p_c. Vậy luật để xác định là [10]:

$$\max_P f(\pi|H_1;P) \gtrless_{absent}^{present} \eta f\ (\pi|H_0;p_c) \qquad (3\text{-}1)$$

Gọi tổng số lượng gói tin đã được truyền đi là K, số gói tin truyền thành công là K_s và thất bại là K_f và được tính theo công thức $K = K_s + K_f$.

Số lượng gói tin thành công K_s tính theo trạng thái là $K_s = nb_{0,0}$. Tương tự số lượng gói tin thất bại K_f tính theo trạng thái là $K_f = \sum_{i=1}^{m} nb_{i,0}$ [10]. Do đó xác suất của trạng thái π được tính như sau:

$$\Pr[\pi] \propto P^{K_f}(1 - P)^{K_s}$$

Giá trị P ở công thức trên đạt giá trị cực đại khi $P_{max} = \frac{K_f}{K_s + K_f}$ (3.2)

Khi đó GLRT giảm xuống còn

$$\frac{f(\pi|H_1;P_{max})}{f(\pi|H_0;P_c)} \gtrless \eta$$

$$\Lambda(K_f) = \frac{P_{max}^{K_f}(1-P_{max})^{K_s}}{P_c^{K_f}(1-P_c)^{K_s}} \gtrless \eta \quad (3.3)$$

Để tính xác suất báo động giả, cần phải tìm sự phân phối của Λ khi H_0 là đúng. Ví dụ như là xác suất mà LRT vượt quá ngưỡng η khi H_0 là đúng

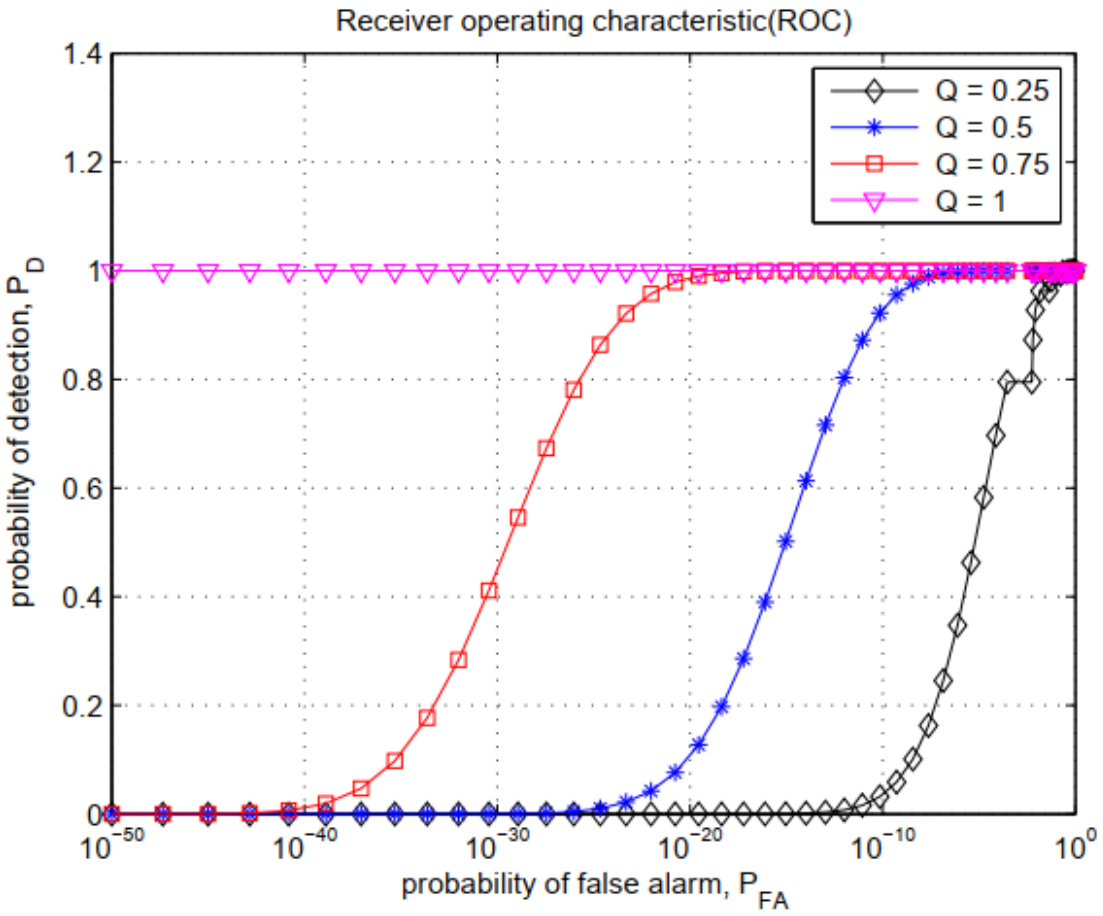
$$\Pr[K_f = \iota | H_0] = KC_\iota p_c^\iota (1 - p_c)^{K-\iota}$$

$$\Pr[\Lambda = \lambda | H_0] = \Pr[K_f = \Lambda^{-1}(\lambda) | H_0]$$

$$P_{FA} = \sum_{\lambda > \eta} \Pr[\Lambda = \lambda | H_0] \qquad (3.4)$$

Hình 3-1 thể hiện đặc tính hoạt động bộ thu ROC (Receiver Operating Characteristic) của máy dò. Trục tung là giá trị rất nhỏ của P_{FA}. Bộ dò phát

hiện jamming rất tốt. Với Q = 0.25 và P_D = 0.9, P_{FA} có giá trị xấp xỉ 0.01. Từ ROC này ngưỡng η được chọn ra. Ví dụ giá trị nhỏ nhất Q cần để phát hiện là 0.25, ngưỡng η được chọn từ đường cong tương ứng bởi P_D hoặc P_{FA}.[10]



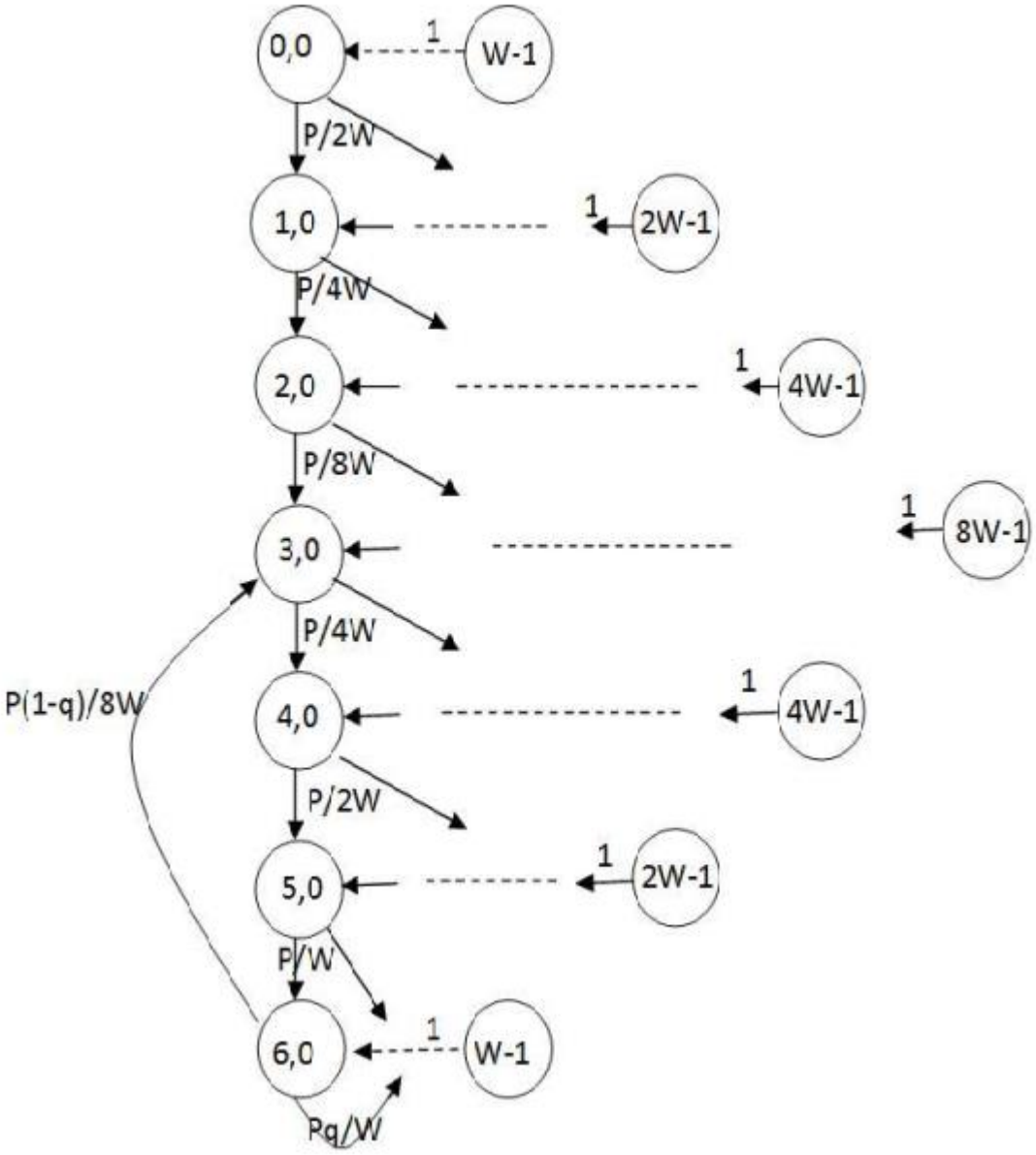
Hình 3-1 ROC của bộ dò

3.2 Sửa cơ chế DCF để chống tấn công kiểu Jamming

Cấu trúc chuẩn của DCF khi trường hợp đụng độ xảy ra sẽ giảm xác suất truyền gói tin của trạm xuống. Cơ chế này là một điểm để jammer khai thác. Như đã trình bày ở chương 2, khi xác suất tắc nghẽn cao thì jammer có thể tiết kiệm được năng lượng khi tấn công.

Hình 3-2 cho ta thấy cách chỉnh sửa DCF [10]. Thay vì giảm xác suất truyền khi xác suất tắc nghẽn cao như ở chương 2, mục đích của chỉnh sửa DCF là tăng xác suất truyền ngay cả khi tắc nghẽn cao, yêu cầu jammer gửi nhiều gói tin hơn và đốt năng lượng của jammer với tốc độ nhanh hơn. Khi

các thiết bị jammer bị hạn chế hoạt động ở công suất cao, bản thân các thiết bị jammer sẽ bị tê liệt trước và mạng WLAN có thể trở về trạng thái bình thường một cách nhanh chóng.



Hình 3-2 DCF chỉnh sửa

Bằng cách tính toán qua các phương trình tương tự như ở mục 2.2, theo tài liệu tham khảo kết quả được trình bày như sau [10]:

$$b_{0,0} = \frac{2}{\sum_{i=0}^{2}(2^i W+1)P^i + \frac{p^3}{1-\Delta}(\sum_{i=0}^{2}(\frac{8}{2^i}W+1)P + \frac{p^3}{1-P_q}(W+1))}$$

$$\Delta = \frac{P^4(1-q)}{q-Pq}$$

$$b_{1,0} = Pb_{0,0}$$

$$b_{3,0} = \frac{P^3}{1-\Delta} b_{0,0}$$

$$b_{5,0} = P^2 b_{3,0}$$

$$b_{6,0} = \frac{P^3}{1-Pq} b_{3,0}$$

Xác suất truyền của trạm (τ) và xác suất đụng độ (p_c) của trạm được tính

$$\tau = \sum_{i=0}^{6} b_{i,0}$$

$$p_c = 1 - (1 - \tau)^{N-1}$$

Chi phí điện năng trung bình của jammer được tính cho cả DCF chuẩn và DCF sửa đổi theo mục 2.4. Các thực nghiệm tham khảo của tác giả [10] lần lượt ở 3, 10, 20, 50 máy trạm để đánh giá mức năng lượng sử dụng khi có tắc nghẽn so sánh giữa DCF chuẩn và DCF sửa đổi ta gọi là M-DCF. Tác giả [10] có những nhận định như sau.

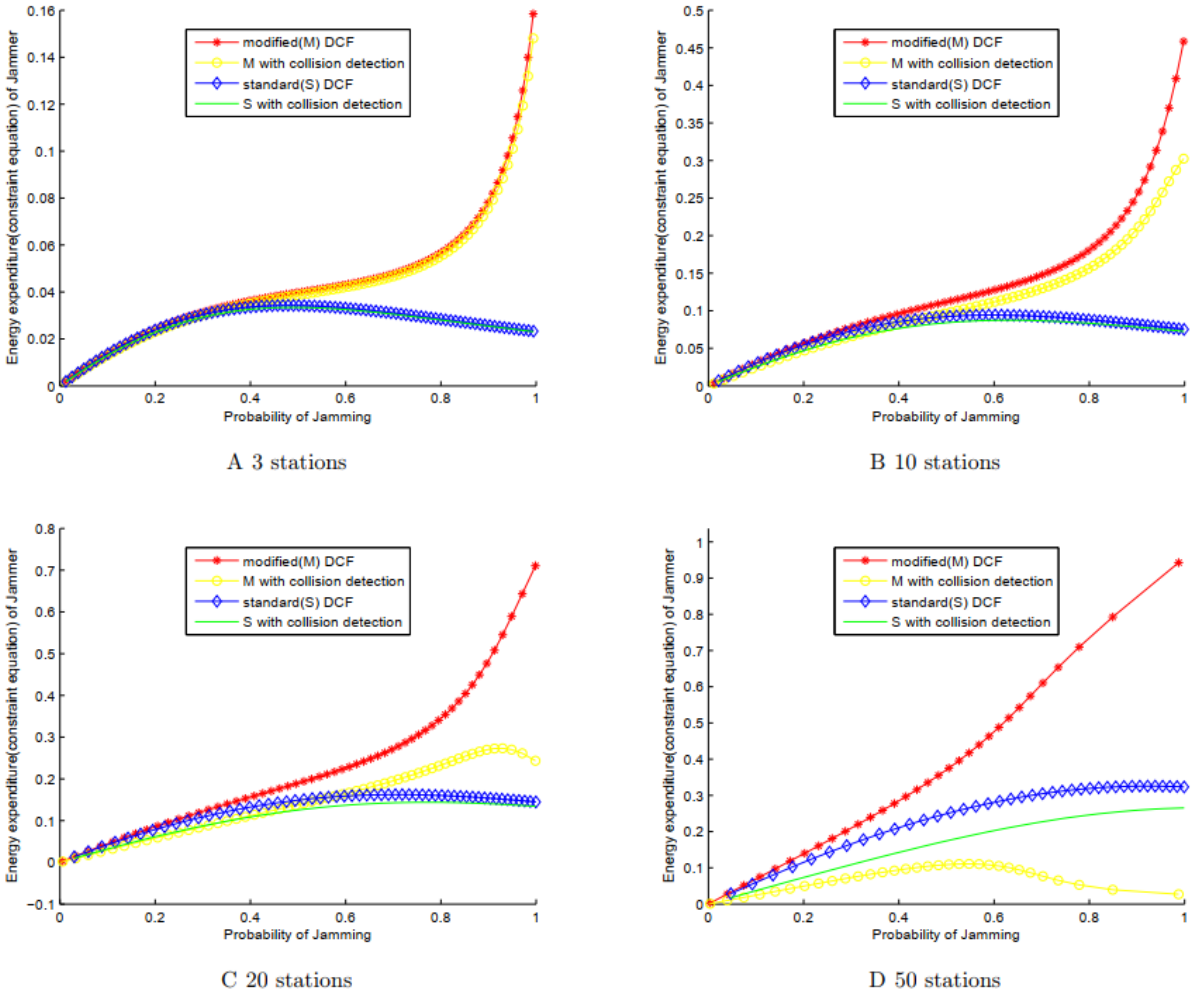
Hình vẽ 3-3 thể hiện sự tương quan giữa chi phí năng lượng và xác suất tắc nghẽn Q = 1. Hình vẽ 3-3A cho thấy tắc nghẽn có và không có đụng độ sử dùng gần như cùng một mức năng lượng trung bình và tăng lên giá trị rất cao khi sừ dụng với DCF sửa đổi. Tuy nhiên tại hình vẽ 3-3D jammer với phát hiện đụng độ tiêu tốn năng lượng ít hơn khi sử dụng M-DCF, điều này xảy ra do các đụng độ chiếm hầu hết đường truyền và với số lượng trạm lớn điện năng tiêu tốn của M-DCF có phát hiện đụng độ là ít. Tuy nhiên với M-DCF không phát hiện đụng độ hoặc số lượng đụng độ là rất nhỏ thì điện năng tiêu thụ của jammer là rất lớn và M-DCF tỏ ra khá hiệu quả

Hình vẽ 3-5 cho thấy sự tương quan giữa xác suất tắc nghẽn tác động tới thông lượng như thế nào giữa chuẩn DCF và M-DCF. Hình vẽ 3-4A thể hiện thông lượng ở biên được cải thiện đáng kể khi sử dụng M-DCF. Tuy nhiên

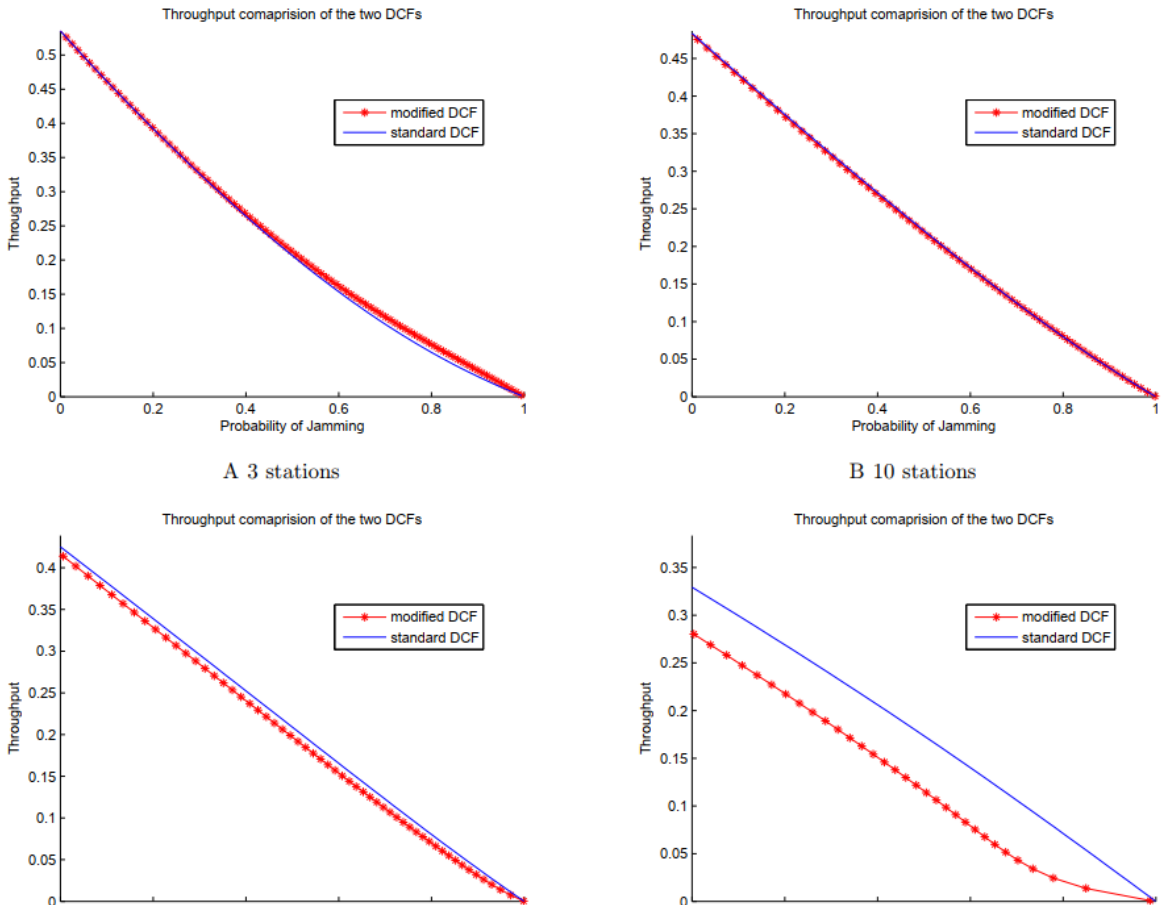
khi số lượng trạm tăng lên 50 thì M-DCF lại thể hiện thông lượng bị suy giảm. Điều này là do sự tăng cao của xác suất đụng độ p_c khi M-DCF được sử dụng. Mục tiêu chính là loại bỏ jammer càng sớm càng tốt bằng cách ép jammer hoạt động công suất cao.

Hình vẽ 3-4 so sánh mức tăng xác suất đụng độ với số trạm tăng lên trong cả 2 trường hợp DCF và M-DCF. Có thể dễ dàng quan sát xác suất đụng độ trong trường hợp M-DCF đạt tới 1 nhanh hơn DCF

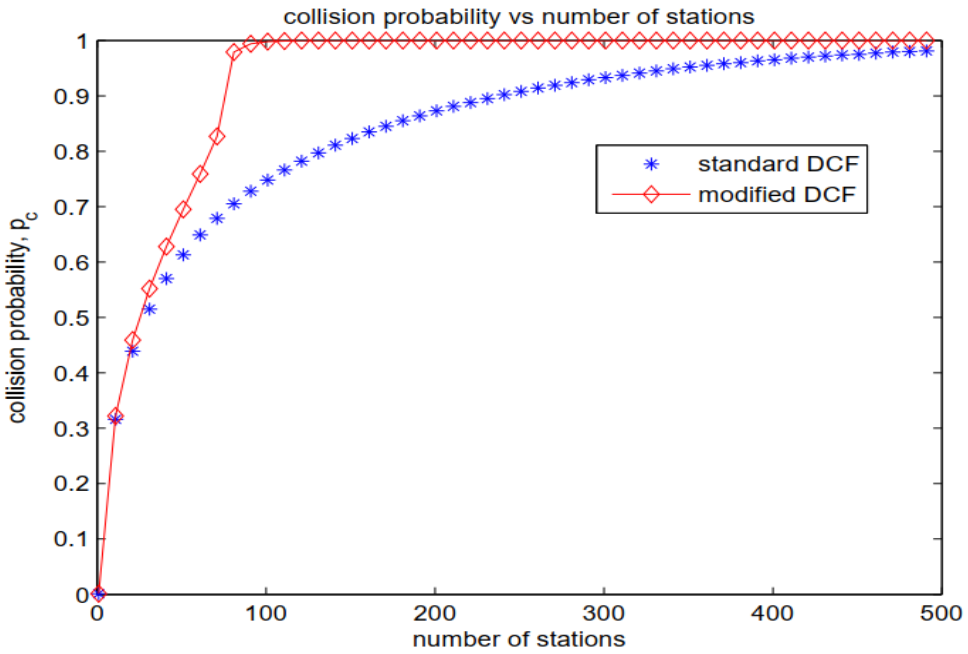
Với hình vẽ 3-6 tác giả cho ta thấy xác suất đụng độ giảm rồi tăng với xác suất gói tin lỗi P, điều này xảy ra là do đặc tính tự nhiên của M-DCF, cửa sổ backoff tăng rồi giảm.



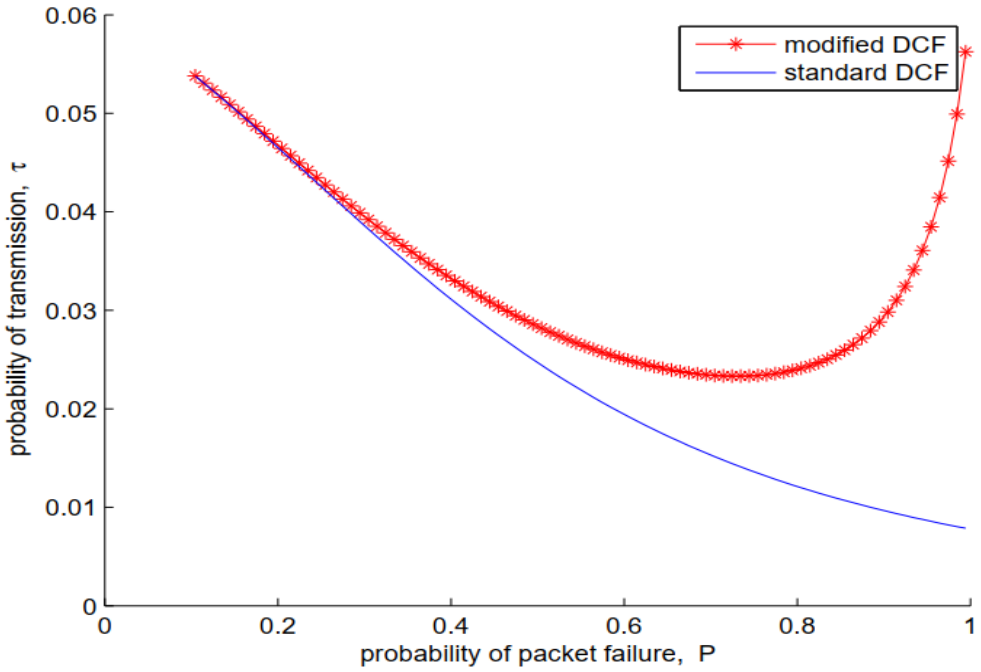
Hình 3-3 Sử dụng năng lượng của jammer theo DCF, M-DCF với 3,10,20,50 trạm



Hình 3-4 Thông lượng và xác suất tắc nghẽn



Hình 3-5 Xác suất tắc nghẽn và số trạm



Hình 3-6 Xác suất truyền với xác suất gói tin lỗi

CHƯƠNG 4 - MÔ PHỎNG VÀ ĐÁNH GIÁ KẾT QUẢ

4.1 Công cụ mô phỏng NS2

4.1.1 Giới thiệu và lịch sử phát triển bộ công cụ NS2

NS (phiên bản) là phần mềm mô phỏng mạng điều khiển sự kiện riêng rẽ hướng đối tượng, được phát triển tại UC Berkely, viết bằng ngôn ngữ C++ và OTcl. NS rất hữu ích cho việc mô phỏng mạng diện rộng (WAN) và mạng local (LAN). Bốn lợi ích lớn nhất của NS-2 phải kể đến đầu tiên là:

- Khả năng kiểm tra tính ổn định của các giao thức mạng đang tồn tại.
- Khả năng đánh giá các giao thức mạng mới trước khi đưa vào sử dụng.
- Khả năng thực thi những mô hình mạng lớn mà gần như ta không thể thực thi được trong thực tế.
- Khả năng mô phỏng nhiều loại mạng khác nhau

NS-2 (Network Simulator version 2.xx) là một trong các bộ mô phỏng mạng mã nguồn mở, tự do, được sử dụng rộng rãi phổ biến nhất trong các trường đại học (có giảng dạy về mạng) trên thế giới. NS-2 đặc biệt mạnh trong việc mô phỏng để nghiên cứu về các giao thức mạng, từ tầng MAC cho đến tầng Transport. Sử dụng NS-2 có thể mô phỏng các mạng có dây (Wired Networks), không dây (Wireless Networks), mạng di động không dây đặc biệt - MANET (Mobile Adhoc Networks), hỗn hợp, v.v. Các sự kiện xảy ra trong mạng mô phỏng và một số tham số cần nghiên cứu thường được kết xuất ra "tệp vết" (Trace file) hoặc một số tệp dạng văn bản. Để nhận được các kết quả mong muốn cần phải xử lý các tệp kết quả này thông qua các hàm hoặc các công cụ lập trình và biểu diễn kết quả thu được, thường là dưới dạng đồ thị.

Hiện tại phiên bản mới nhất là NS-3, là một phần mềm mã nguồn mở được phát triển gần đây. Đây là một phần mềm độc lập không tương thích ngược với NS2. NS-3 cung cấp các tính năng giả lập mạng vượt trội hơn so với NS-2 như: IPv6 mở rộng, các cải tiến wifi, new test framework… Phiên bản mới nhất là NS-3.29 được công bố ngày 04/09/2018. Chi tiết có thể tham khảo thêm tại trang chủ của trình mô phỏng ns3 [11].

4.1.2 Cấu trúc bộ công cụ mô phỏng NS2

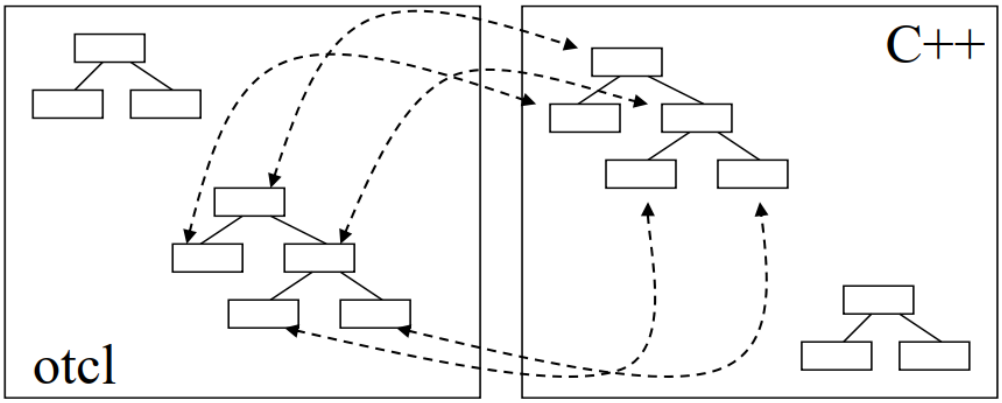
NS2 được viết bằng ngôn ngữ C++ và OTcl. Trong đó, phần cốt lõi của NS2 được viết bằng C++ phục vụ việc mô phỏng chi tiết các giao thức, có thể thao tác một cách hiệu quả đến các byte, các tiêu đề gói và các thuật toán

triển khai mà có thể chạy trên một tập dữ liệu lớn. Các thành phần được viết bằng ngôn ngữ C++ có tốc độ thực thi nhanh nhưng việc thay đổi hoạt động chậm và phức tạp nên phù hợp với việc triển khai các giao thức chi tiết. Các thành phần được viết bằng ngôn ngữ OTcl lại có khả năng thay đổi cấu hình nhanh nhưng tốc độ thực thi chậm hơn, hỗ trợ thực hiện các thao tác đòi hỏi sự thay đổi linh hoạt hơn như các tham số cấu hình mạng hay nghiên cứu một số hoạt cảnh, giúp triển khai ý tưởng của việc mô phỏng.

Để giảm thời gian xử lý gói và sự kiện (không phải thời gian mô phỏng), bộ danh sách sự kiện và các đối tượng thành phần mạng cơ bản trong đường dữ liệu (data path) được viết và biên dịch bằng C ++.

Các đối tượng biên dịch được cung cấp cho trình thông dịch OTcl thông qua một liên kết OTcl và tạo ra một đối tượng OTcl phù hợp cho từng đối tượng C ++ và tạo nên các hàm điều khiển và các biến có cấu hình bởi đối tượng C ++, các đối tượng được biên dịch đóng vai trò là các hàm thành viên, biến thành viên tương ứng của đối tượng OTcl. Như vậy việc điều khiển các đối tượng C++ được trao cho OTcl. Và cũng có nghĩa là thông qua các đối tượng liên kết OTcl cũng có thể thêm các hàm và biến thành viên. Hình 4-1 cho thấy sự phân cấp và liên kết đối tượng trong C++ và OTcl.

Có thể nói rằng, ngôn ngữ C++ là ngôn ngữ lập trình hệ thống đảm bảo tốc độ, hiệu quả hoạt động cho bộ mô phỏng NS2 còn ngôn ngữ kịch bản OTcl đảm bảo khả năng hoạt động đơn giản, linh hoạt cho NS2 và đối với cả người dùng.



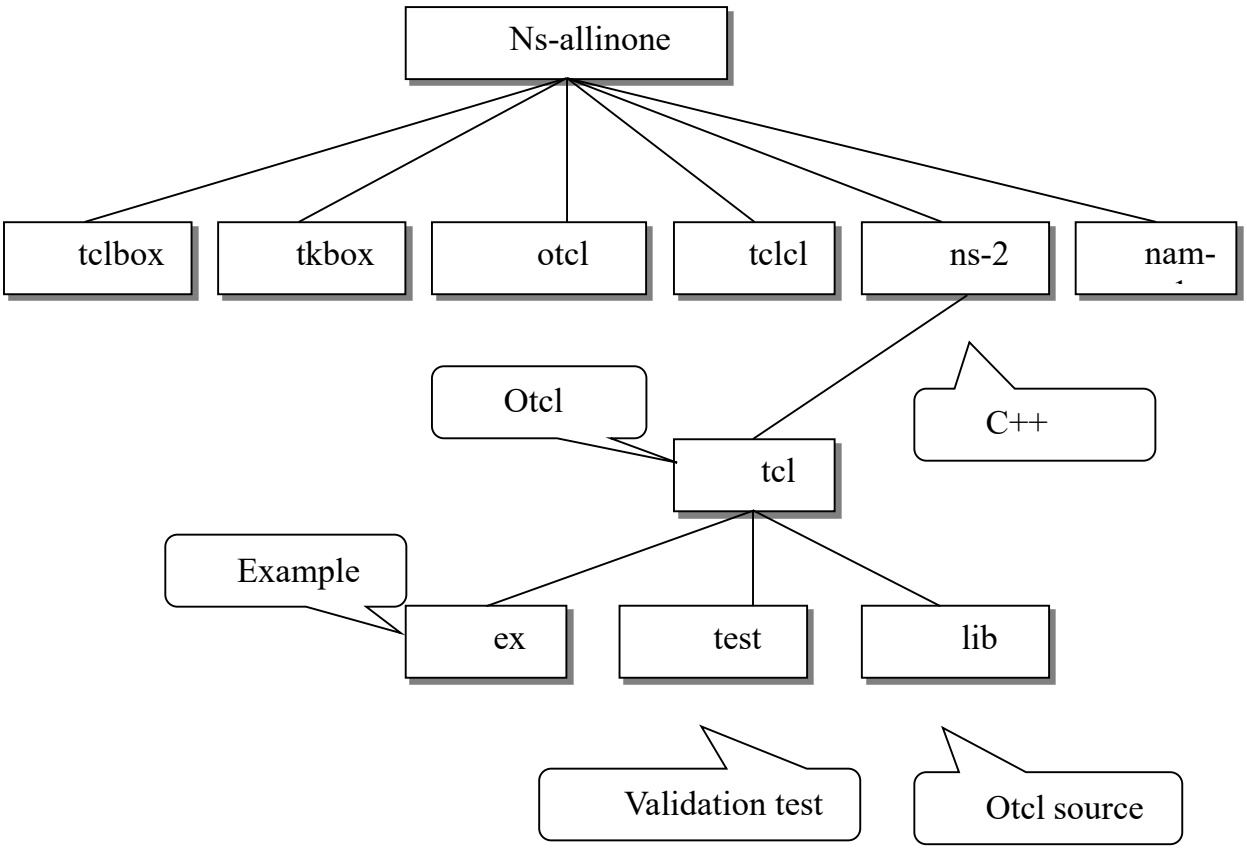
Hình 4-1 C++ và OTcl trong NS-2

4.1.3 Đặc điểm của bộ mô phỏng NS2

NS2 thường được sử dụng để thực hiện các nhiệm vụ mô phỏng thiết kế giao thức và luồng dữ liệu; so sánh các giao thức và hỗ trợ các thiết kế mới. Những khả năng nổi bật của NS2 gồm:

- Khả năng trừu tượng hóa: Giúp nghiên cứu giao thức mạng ở nhiều mức khác nhau, từ hoạt động đơn lẻ của một giao thức đến kết hợp của nhiều luồng dữ liệu và tương tác của nhiều giao thức. Điều này giúp cho người nghiên cứu có khả năng dễ dàng so sánh, đánh giá hoạt động của các giao thức cũng như tác động của các thay đổi, điều chỉnh.
- Khả năng tương tác với mạng thực: Cho phép chương trình mô phỏng đang chạy tương tác với các nút mạng thực đang hoạt động.
- Khả năng tạo ngữ cảnh: Cho phép người nghiên cứu tạo ra các mẫu lưu lượng, các hiện trạng mạng phức tạp, các sự kiện động như lỗi liên kết một cách dễ dàng. Điều này giúp cho việc nghiên cứu, kiểm chứng các giao thức mạng trong các mô hình khác nhau được đúng đắn hơn.
- Khả năng trực quan: Thông qua công cụ hiển thị trực quan NAM cho phép quan sát dễ dàng hoạt động của mạng và hiểu các hành vi phức tạp của mô phỏng mạng.
- Khả năng mở rộng: NS2 cho phép mở rộng các chức năng mới một cách dễ dàng như thay đổi các tham số của mạng, thay đổi hoạt động của các giao thức cũng như hoạt động của các nút. Cho phép người nghiên cứu có thể tự cài đặt, chỉnh sửa cải tiến hoạt động của các giao thức, các chức năng của nút và của các thành phần mạng.

Hình 4-2 cho ta thấy rõ hơn về cấu trúc của bộ mô phỏng NS-2. NS-2 cung cấp rất nhiều ví dụ mẫu tham khảo tại thư mục con (folder) Ns-allinone\ns-2\tcl\ex.



Hình 4-2 Cấu trúc thư mục NS2

4.2 Đề xuất mô hình phát hiện tắc nghẽn

Để kiểm tra hiệu quả của việc phát hiện tắc nghẽn ở tầng vật lý (jamming) tôi đề xuất mô hình mô phỏng như sau:

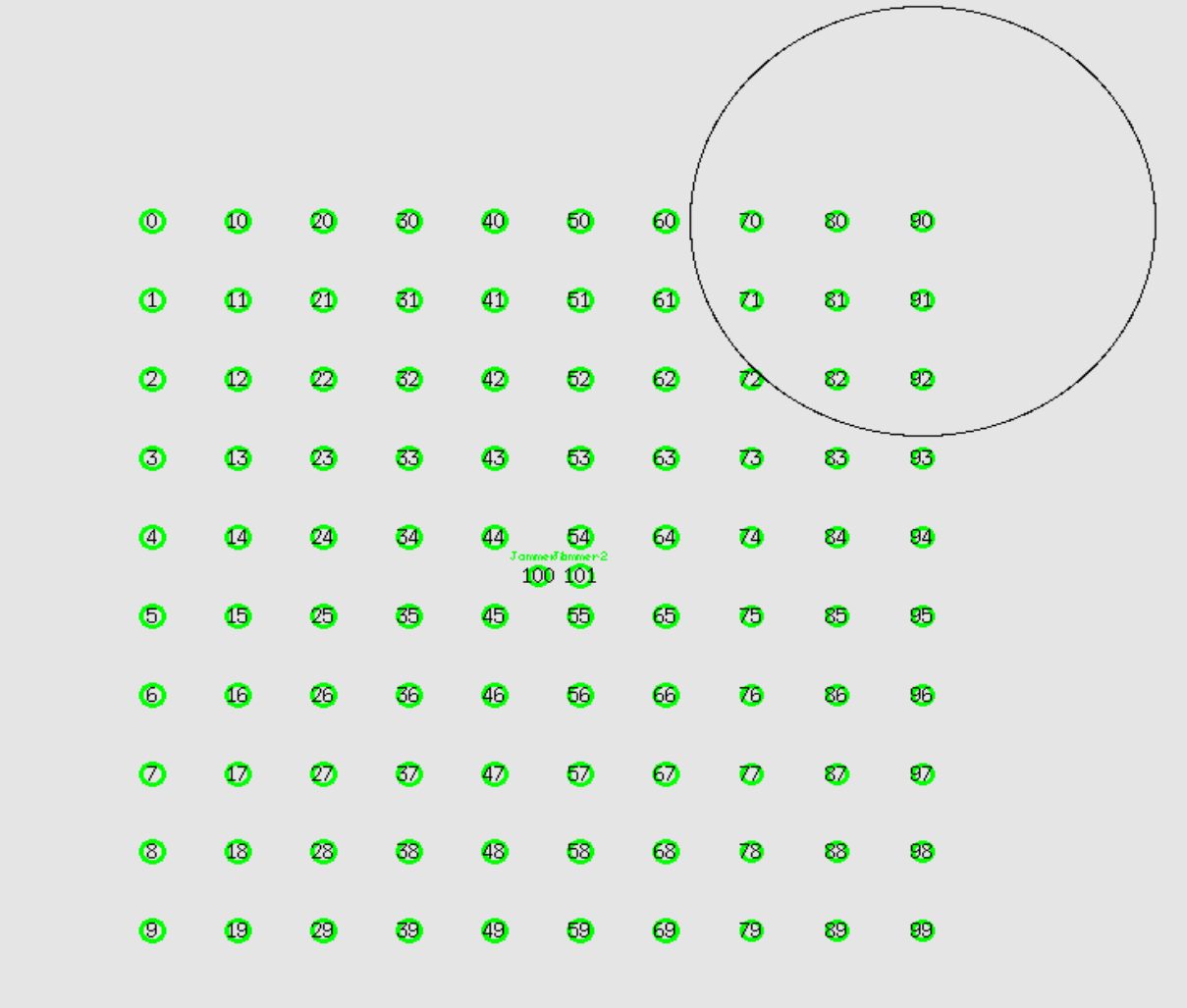
Mạng có cấu trúc tĩnh bao gồm 100 nút bình thường và 1 nút jammer, ngoài ra còn một nút chỉ đóng vai trò nhận gói tin gửi từ jammer.

4.3 Thực hiện mô phỏng

4.3.1 Kịch bản mô phỏng

Mô phỏng được tiến hành trên mô hình mạng có cấu trúc tĩnh tức là các nút không di chuyển, gồm 100 nút được phân bố đều trên diện tích mô phỏng 500x700m^2 sắp xếp thành lưới hình vuông, khoảng cách giữa các nút là 40m. Các nút có phạm vi phát sóng mặc định theo chuẩn IEEE 802.11.

Các nút gây nhiễu (jammer) bao gồm 02 nút 100 và 101 được đặt chính tâm ma trận, nút 101 nằm bên phải nút 100. Mục đích của việc đặt các nút jammer ở tâm vùng mô phỏng là để vùng phủ sóng (radio range – ví dụ trong hình 4-3 là vùng phủ sóng của nút 90) của chúng có thể bao trùm các nút chắc chắn thuộc đường truyền (path) luồng cbr thông thường. Khi không tấn



Hình 4-3 Sơ đồ mô phỏng

công các nút jammer được tắt đi để không ảnh hưởng đến hoạt động của mạng và phù hợp với nghiên cứu của luận văn là các nút tấn công chỉ hoạt động khi cảm nhận thấy có dữ liệu truyền trên môi trường. Băng thông đường truyền được đặt theo giá trị mặc định là 10Mbps.

Lưu lượng trong mạng được phân thành 2 loại khác nhau:

- Lưu lượng thông thường (không phải do jammer sinh ra) có tốc độ không đổi – kiểu cbr (constant bit rate): nút nguồn là nút 0 truyền dữ liệu đến nút đích 99, luồng dữ liệu cbr đi theo đường chéo từ góc trên

bên trái (top-left) đến góc dưới bên phải (bottom-right) của ma trận các nút. Thời gian truyền bắt đầu từ giây thứ 40 và dừng lại ở giây 190.

- Lưu lượng do jammer sinh ra, cũng thuộc kiểu cbr: sau khi nút nguồn số 0 truyền một khoảng thời gian là 10s, nút jammer 100 sẽ truyền dữ liệu cho nút 101 là nguyên nhân làm tắc nghẽn mạng. Thời gian kết thúc cùng lúc với nút 0.

Kịch bản mô phỏng được thực hiện 4 lần với các thông số tốc độ truyền của jammer khác nhau cho thấy mức năng lượng của jammer tiêu hao khác nhau như thế nào khi bị ép hoạt động với cường độ cao dẫn đến suy kiệt năng lượng và bị loại khỏi hệ thống.

Kết quả mô phỏng được đánh giá qua mức năng lượng sử dụng theo thời gian của nút nguồn và nút jammer.

Các file được sử dụng trong quá trình mô phỏng là:

- jamming-attack-vs0.tcl: Kịch bản mô phỏng, do tôi tự xây dựng, có thừa kế một số đoạn code trong các thí dụ thuộc thư viện các thí dụ mẫu của NS2.
- Column: file dùng để trích xuất thông tin vẽ biểu đồ (do một tác giả thuộc cộng đồng sử dụng NS2 đóng góp từ thời kỳ NS2 mới ra đời và được sử dụng rất rộng rãi. Chi tiết có trong phụ lục của luận văn).

Bảng 4-1 là một vài thông số được dùng trong mô phỏng.

Thông số	Giá trị
Số lượng nút của hệ thống	100
Số lượng nút tấn công	2
Giao thức định tuyến	DSDV
Giao thức truyền thông (tầng MAC)	802.11
Băng thông đường truyền	10 Mbps
Kích thước gói tin cbr	500 bytes
Năng lượng khởi tạo cho nút	50 Joule
Năng lượng truyền gói tin (tiêu hao năng lượng của nút trong trạng thái truyền)	0.9 Joule
Năng lượng nhận gói tin	0.5 Joule
Năng lượng khi không hoạt động	0.05 Joule
Năng lượng cảm nhận môi trường	0.0175
Thời gian chạy mô phỏng	200 s

Bảng 4-1 Các thông số mô phỏng

4.3.2 Kết quả và đánh giá mô phỏng.

Kịch bản mô phỏng thứ nhất: Nguồn cbr thường và nguồn cbr của jammer có cùng interval=0.2s.

Kết xuất năng lượng node nguồn (src) ra file “energy-of-srcnodes0-jammers-pktinterval0.2.txt”, như sau:

cat jamming-attack.tr |grep ^s|grep "AGT"|grep "cbr"|grep "0:0"|grep "99:0"|grep "_0_"|perl column 1 13 > energy-of-srcnodes0-jammers-pktinterval0.2.txt.

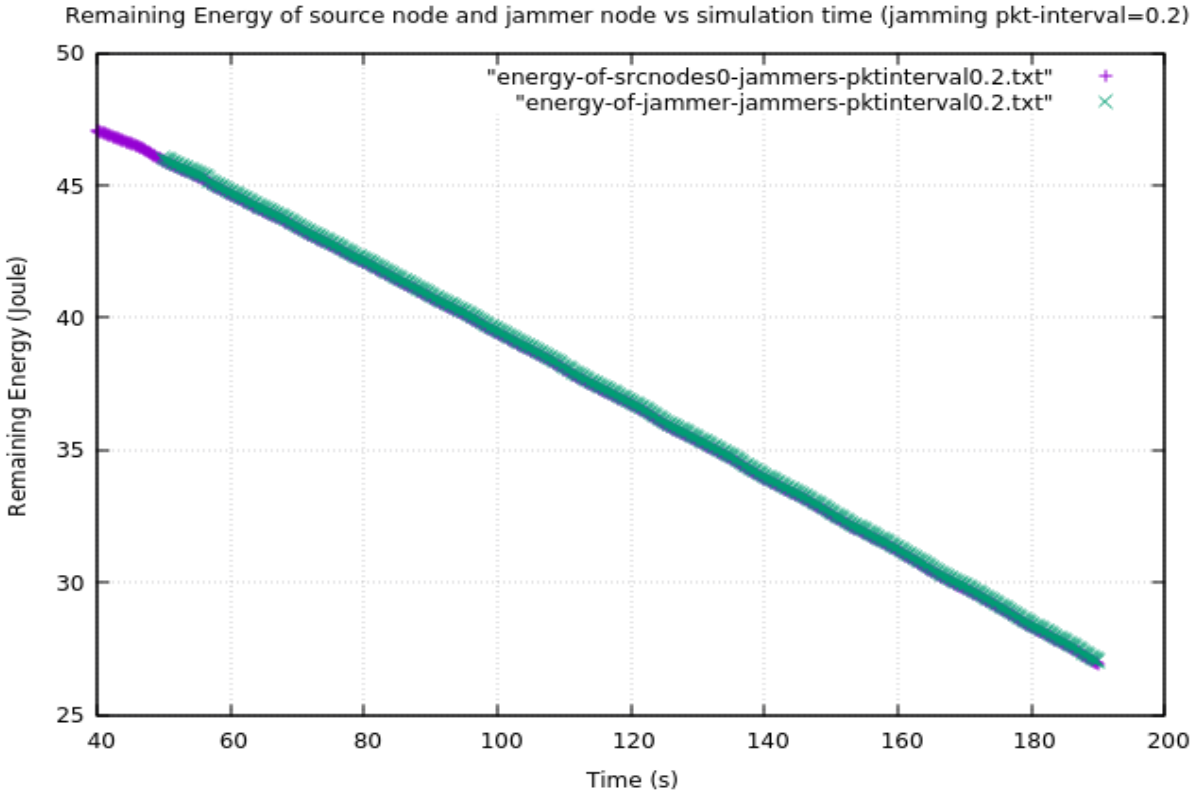
Kết xuất năng lượng node jammer ra file “energy-of-jammer-jammers-pktinterval0.2.txt”, như sau:

cat jamming-attack.tr |grep ^s|grep "AGT"|grep "cbr"|grep "100:0"|grep "101:0"|grep "_100_"|perl column 1 13 > energy-of-jammer-jammers-pktinterval0.2.txt.

Dùng gnuplot vẽ đồ thị so sánh sự giảm năng lượng của nút nguồn (nút 0) và jammer như sau:

- *set title "Remaining Energy of source node and jammer node vs simulation time (jamming pkt-interval=0.2)"*
- *set xlabel "Time (s)*
- *set xlabel "Time (s)"*
- *plot "energy-of-srcnodes0-jammers-pktinterval0.2.txt", "energy-of-jammer-jammers-pktinterval0.2.txt"*

Tôi nhận được đồ thị kết quả theo hình 4-4:



Hình 4-4 Năng lượng nút nguồn và jammer với interval 0.2

Nhận xét kết quả được biểu diễn bằng đồ thị - hình 4-4:

Vì:

- Nguồn sinh lưu lượng cbr của nút nguồn có các tham số: packetSize_=500 bytes; interval_=0.2. Từ đó suy ra tốc độ dữ liệu đưa vào mạng - rate = 500bytes/0.2sec=20000bps. Đây là tốc độ khá thấp so với banwidth của đường truyền (giá trị ngầm định tôi sử dụng là 10 Mbps).
- Nguồn sinh lưu lượng của nút jammer cũng có các tham số giống như vậy.

Kết luận:

- Nguồn năng lượng của nút truyền (nút bị tấn công kiểu jamming) và jammer giảm tuyến tính theo thời gian và có tốc độ giảm như nhau.

Kịch bản mô phỏng thứ hai: tương tự như trên, nhưng nguồn sinh lưu lượng cbr của jammer có packet interval=0.04.

- Kết xuất năng lượng node nguồn (src) ra file “energy-of-srcnodes0-jammers-pktinterval0.04.txt”:

cat jamming-attack.tr |grep ^s|grep "AGT"|grep "cbr"|grep "0:0"|grep "99:0"|grep "_0_"|perl column 1 13 > energy-of-srcnodes0-jammers-pktinterval0.04.txt.

- Kết xuất năng lượng node jammer ra file “energy-of-jammer-jammers-pktinterval0.04.txt”:

cat jamming-attack.tr |grep ^s|grep "AGT"|grep "cbr"|grep "100:0"|grep "101:0"|grep "_100_"|perl column 1 13 > energy-of-jammer-jammers-pktinterval0.04.txt.

- Dùng gnuplot vẽ đồ thị so sánh sự giảm năng lượng của nút nguồn (nút 0) và jammer như sau: (các bước thực hiện tương tự như vẽ hình ở kịch bản mô phỏng thứ nhất).

Nhận xét kết quả được biểu diễn bằng hình 4-5:

Vì:

- Nguồn sinh lưu lượng cbr của nút nguồn có các tham số giống như ở kịch bản 1.

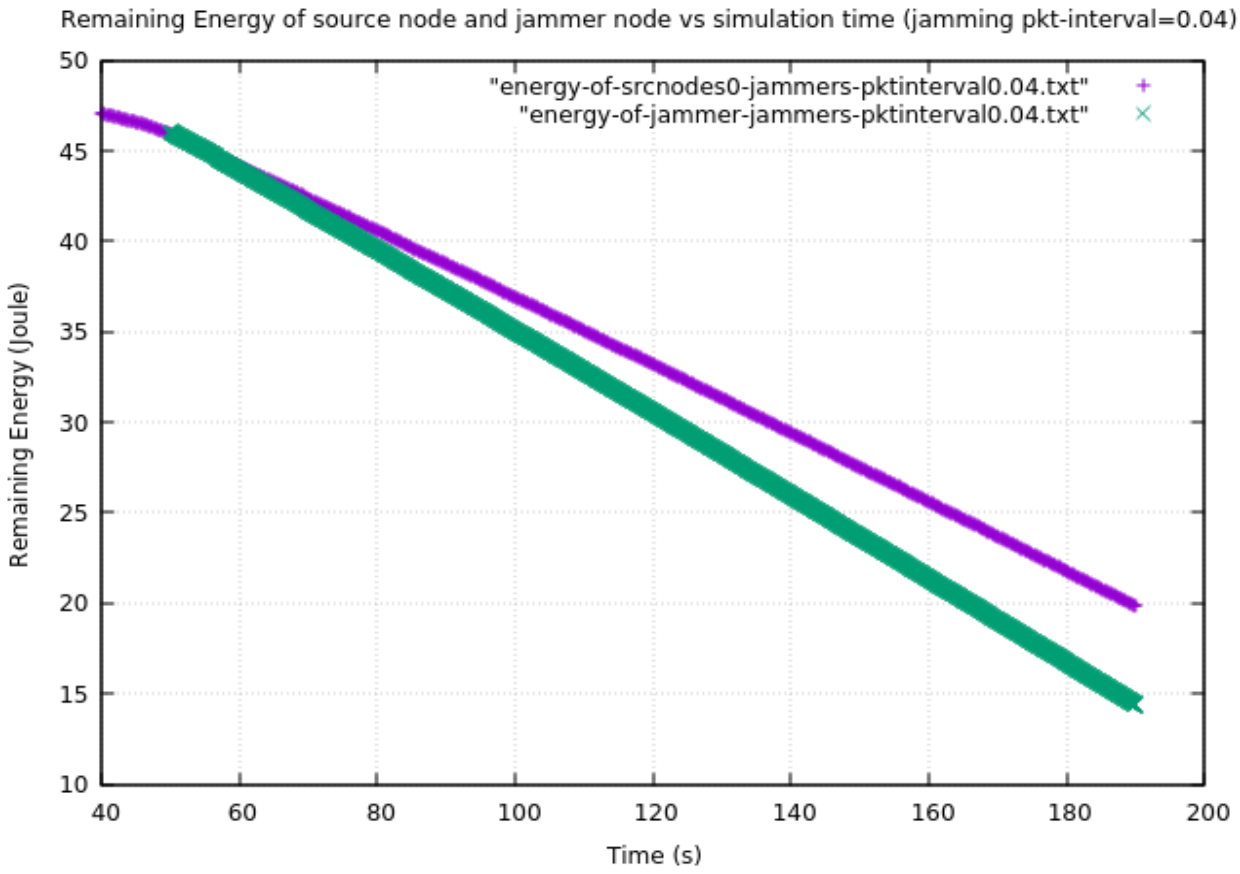
- Nguồn sinh lưu lượng của nút jammer cũng có các tham số packetSize_=500 bytes như trong kịch bản thứ nhất; Nhưng interval_=0.04, tức là nhỏ hơn 5 lần so với ở kịch bản 1. Từ đó suy ra tốc độ dữ liệu đưa vào mạng sẽ lớn hơn 5 lần, rate =5*20000bps = 100000bps. Đây là tốc độ khá thấp so với banwidth của đường truyền (giá trị ngầm định tôi sử dụng là 10mbps).

Kết luận:

- Nguồn năng lượng của nút tấn công (jammer) và nút truyền (nút bị tấn công kiểu jamming) đều giảm tuyến tính theo thời gian, tuy nhiên tốc độ giảm năng lượng của nút jammer nhanh hơn.
- Từ khi nút jammer bắt đầu tấn công (time=50s) đến khi ngừng tấn công (time=190s), năng lượng của nó giảm từ giá trị ban đầu là 50J xuống còn dưới 14J.

Kịch bản mô phỏng thứ ba: tương tự như trên, nhưng nguồn sinh lưu lượng cbr của jammer có packet interval=0.008.

- Kết xuất năng lượng node nguồn (src) ra file “energy-of-srcnodes0-jammers-pktinterval0.008.txt”:



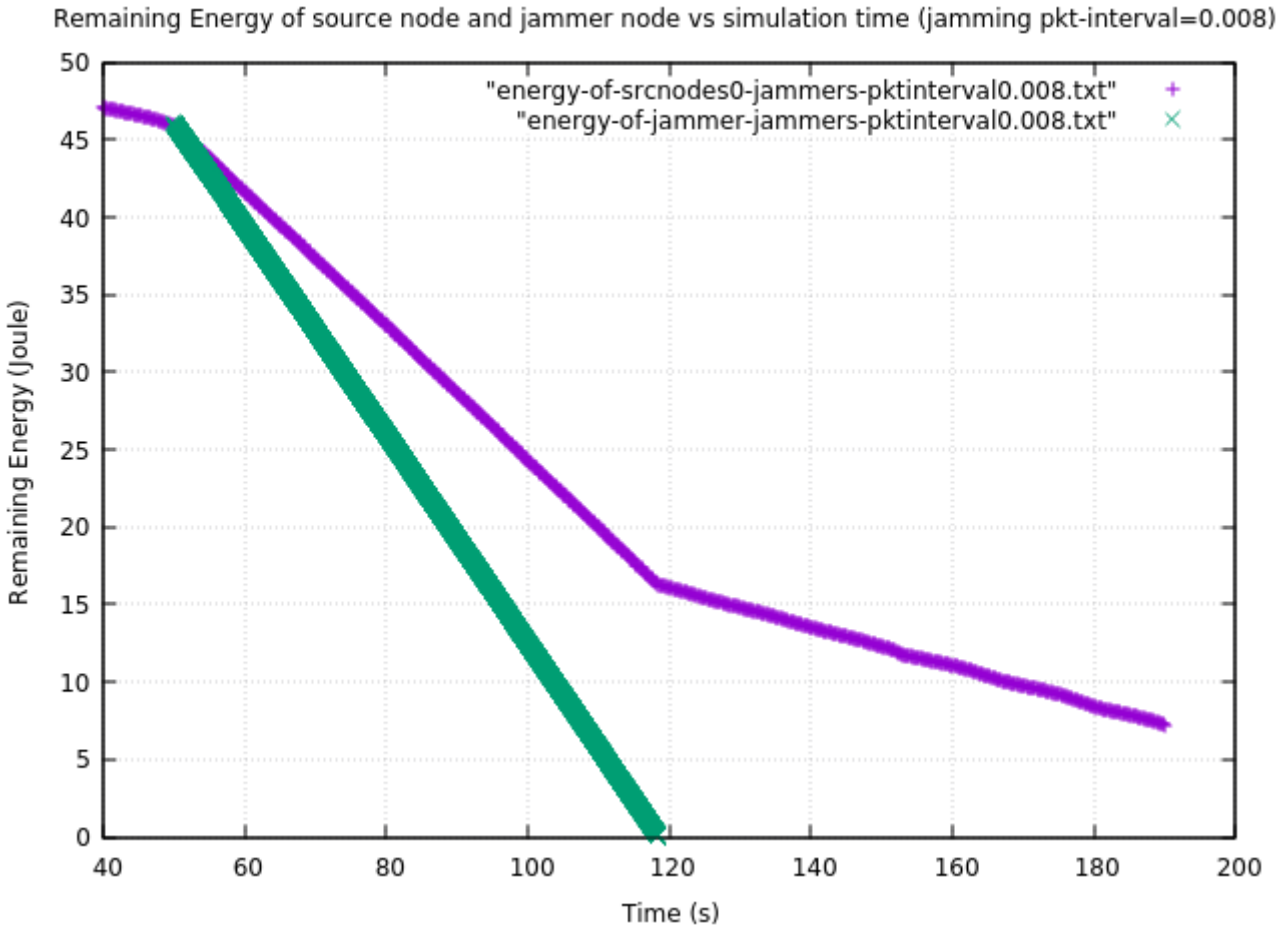
Hình 4-5 Năng lượng nút nguồn và jammer với interval 0.04

cat jamming-attack.tr |grep ^s|grep "AGT"|grep "cbr"|grep "0:0"|grep "99:0"|grep "_0_"|perl column 1 13 > energy-of-srcnodes0-jammers-pktinterval0.008.txt.

- Kết xuất năng lượng node jammer ra file “energy-of-jammer-jammers-pktinterval0.008.txt”:

cat jamming-attack.tr |grep ^s|grep "AGT"|grep "cbr"|grep "100:0"|grep "101:0"|grep "_100_"|perl column 1 13 > energy-of-jammer-jammers-pktinterval0.008.txt.

- Dùng gnuplot vẽ đồ thị so sánh sự giảm năng lượng của nút nguồn (nút 0) và jammer như sau: (các bước thực hiện tương tự như kịch bản đầu tiên).



Hình 4-6 Năng lượng nút nguồn và jammer với interval 0.008

Nhận xét kết quả được biểu diễn bằng đồ thị hình 4-6:

Vì:

- Nguồn sinh lưu lượng cbr của nút nguồn có các tham số giống như ở kịch bản 1 và 2.
- Nguồn sinh lưu lượng của nút jammer cũng có các tham số packetSize_=500 bytes như trong kịch bản thứ nhất; Nhưng interval_=0.008, tức là nhỏ hơn 25 lần so với ở kịch bản 1. Từ đó suy ra tốc độ dữ liệu đưa vào mạng sẽ lớn hơn 25 lần, rate =25*20000bps = 500000bps. Đây là tốc vẫn còn khá thấp so với banwidth của đường truyền (giá trị ngầm định tôi sử dụng là 10mbps).

Kết luận:

- Nguồn năng lượng của nút tấn công (jammer) và nút truyền (nút bị tấn công kiểu jamming) đều giảm tuyến tính theo thời gian, tuy nhiên tốc độ giảm năng lượng của nút jammer nhanh hơn.
- Từ khi nút jammer bắt đầu tấn công (time=50s) đến khoảng 120s thì nó bị hết năng lượng. Sau thời điểm này, nút bị tấn công vẫn giảm năng lượng một cách tuyến tính, nhưng với tốc độ giảm thấp hơn. Lý do là vì trong thời gian bị tấn công, có nhiều gói tin bị hỏng, dẫn đến việc phải truyền lại (ở tầng MAC).
- Kẻ tấn công (nút jammer) bị hết năng lượng khá nhanh (trong trường hợp này là 70s), mặc dù tổng lưu lượng đưa vào mạng mới bằng khoảng 5% (20000bps+500000bps =520000bps, xấp xỉ bằng 0.5mbps) băng thông 10 mbps của kênh truyền.

Kịch bản mô phỏng thứ tư: tương tự như trên, nhưng nguồn sinh lưu lượng cbr của jammer có packet interval=0.0016

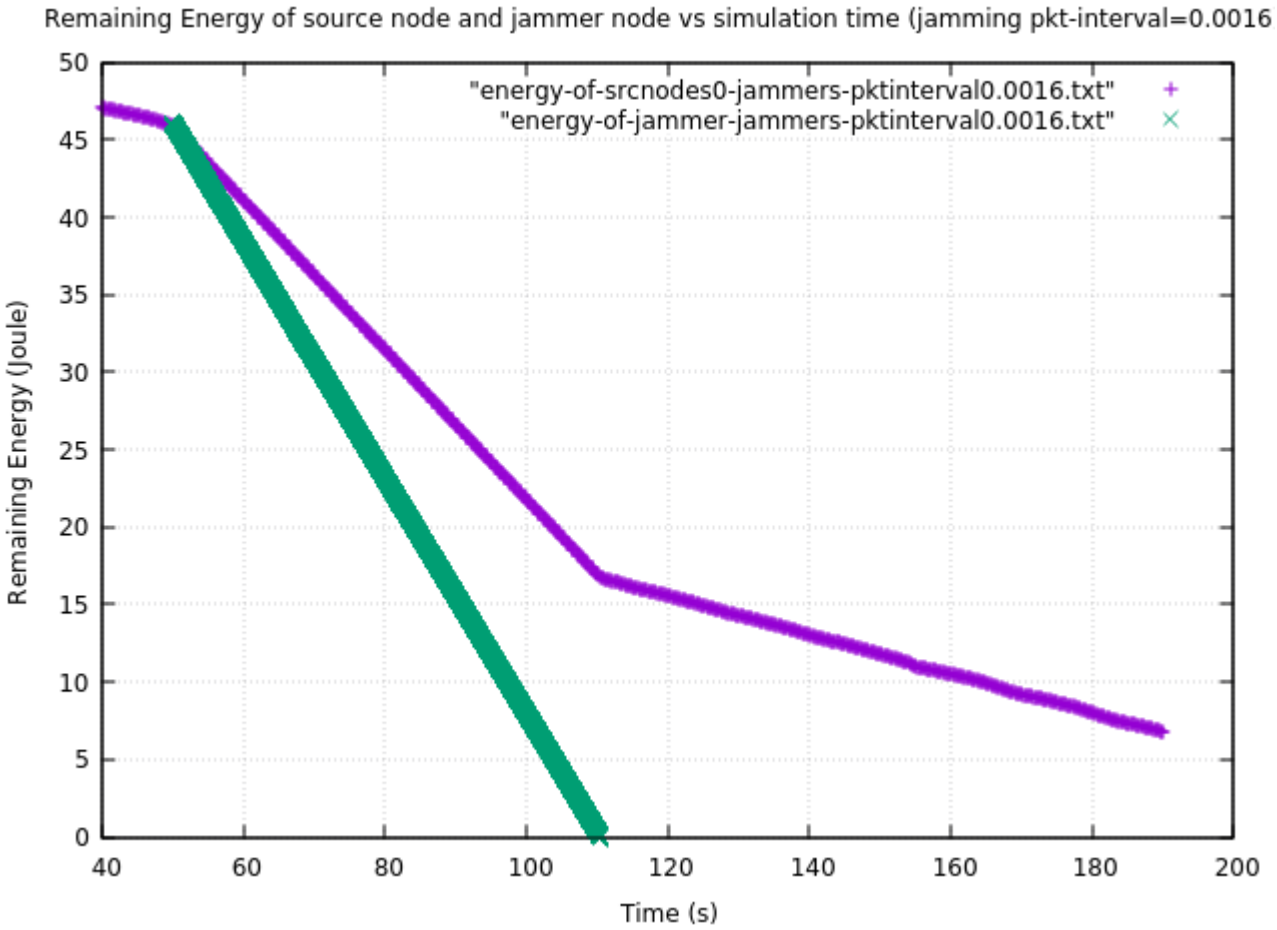
- Kết xuất năng lượng node nguồn (src) ra file “energy-of-srcnodes0-jammers-pktinterval0.0016.txt”:

cat jamming-attack.tr |grep ^s|grep "AGT"|grep "cbr"|grep "0:0"|grep "99:0"|grep "_0_"|perl column 1 13 > energy-of-srcnodes0-jammers-pktinterval0.0016.txt.

- Kết xuất năng lượng node jammer ra file “energy-of-jammer-jammers-pktinterval0.2.txt”:

cat jamming-attack.tr |grep ^s|grep "AGT"|grep "cbr"|grep "100:0"|grep "101:0"|grep "_100_"|perl column 1 13 > energy-of-jammer-jammers-pktinterval0.0016.txt.

- Dùng gnuplot vẽ đồ thị so sánh sự giảm năng lượng của nút nguồn (nút 0) và jammer như sau: (các bước thực hiện tương tự như kịch bản mô phỏng đầu tiên).



Hình 4-7 Năng lượng nút nguồn và jammer với interval 0.0016

Nhận xét kết quả được biểu diễn bằng đồ thị như hình 4-7:

Vì:

- Nguồn sinh lưu lượng cbr của nút nguồn có các tham số giống như ở kịch bản 1.
- Nguồn sinh lưu lượng của nút jammer cũng có các tham số packetSize_=500 bytes như trong kịch bản thứ nhất; Nhưng interval_=0.0016, tức là nhỏ hơn 125 lần so với ở kịch bản 1. Từ đó

zsuy ra tốc độ dữ liệu đưa vào mạng sẽ lớn hơn 125 lần, rate =125*20000bps = 2500000bps. Đây là tốc vẫn còn thấp hơn so với banwidth của đường truyền (giá trị ngầm định tôi sử dụng là 10mbps).

Kết luận:

- Nguồn năng lượng của nút tấn công (jammer) và nút truyền (nút bị tấn công kiểu jamming) đều giảm tuyến tính theo thời gian, tuy nhiên tốc độ giảm năng lượng của nút jammer nhanh hơn.
- Từ khi nút jammer bắt đầu tấn công (time=50s) đến khoảng 110s thì nó bị hết năng lượng. Sau thời điểm này, nút bị tấn công vẫn giảm năng lượng một cách tuyến tính, nhưng với tốc độ giảm thấp hơn. Lý do là vì trong thời gian bị tấn công, có nhiều gói tin bị hỏng, dẫn đến việc phải truyền lại (ở tầng MAC).
- Kẻ tấn công (nút jammer) bị hết năng lượng khá nhanh (trong trường hợp này là 60s), mặc dù tổng lưu lượng đưa vào mạng mới bằng khoảng 25.2% (20000bps+2500000bps =2520000bps, xấp xỉ bằng 2.52mbps) băng thông 10 mbps của kênh truyền.

4.4 Kết luận về các kết quả nhận được từ mô phỏng.

Cuộc tấn công gây tắc nghẽn mạng như theo kịch bản mô phỏng cho ta thấy với nghiên cứu chỉnh sửa cách thức phản ứng của DCF làm cho nút jammer hoạt động với công suất cao có thể loại bỏ nút jammer khá sớm ra khỏi hệ thống. Điều này giúp cho hệ thống sớm hồi phục lại trạng thái cân bằng.

Kết quả mô phỏng đã chứng minh hiệu quả của biện pháp khắc phục tắc nghẽn. Tuy nhiên theo tôi đánh giá nhược điểm của phương pháp này là khi áp dụng lên toàn bộ hệ thống thì hiệu suất mạng có khả năng giảm rõ rệt vì phải theo dõi trạng thái của toàn bộ các nút trong hệ thống mạng. Đặc biệt khi số lượng các nút trong mạng là lớn có thể gây ra ngưng trệ hệ thống trước khi nút jammer tấn công.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN TIẾP THEO

Vấn đề tắc nghẽn trong mạng nội bộ không dây WLAN tuy không phải là vấn đề mới, nổi cộm. Tuy nhiên việc phát hiện và phòng tránh tắc nghẽn đóng một vai trò không thể thiếu trong việc đảm bảo sự hoạt động ổn định của hệ thống mạng cũng như gia tăng tính trải nghiệm của người dùng trong hệ thống. Trong luận văn này đã nêu ra được khả năng tắc nghẽn không chỉ do tính tự nhiên của hệ thống mà còn có thể được gây ra bởi các tác nhân khác. Dựa trên nguyên tắc của hàm cộng tác phân tán DCF, luận văn nghiên cứu rõ thêm cách cải tiến DCF để xử lý tắc nghẽn.

Theo như tài liệu tham khảo [10] tác giả cũng có gợi ý khả năng là jammer có thể theo dõi trạng thái backoff của toàn bộ các trạm trong hệ thống, đây cũng là một ý tưởng tham khảo để có hướng đi tiếp theo giải quyết vấn đề hiệu quả hơn.

Do thời gian nghiên cứu có hạn cho nên những vấn đề được nêu ra trong luận văn còn nhiều hạn chế về độ chuyên sâu, cũng như chưa thực hiện được nhiều đánh giá thực nghiệm mong muốn. Tuy nhiên những kiến thức được nêu ra trong luận văn giúp tôi hiểu rõ hơn về cách thức hệ thống phản ứng với tắc nghẽn xảy ra. Trong thời gian tới nếu điều kiện cho phép, tôi sẽ tiếp tục nghiên cứu vấn đề sâu hơn và mong muốn có thể đưa ra được những phương pháp hoặc phần mềm phát hiện tranh chấp đơn giản với hiệu quả cao.

TÀI LIỆU THAM KHẢO

Tài liệu tiếng Việt

[1] Nguyễn Đình Việt (2008), *Bài giảng "Truyền số liệu và mạng máy tính"*, Chuyên ngành Mạng và Truyền thông máy tính, Khoa Công Nghệ Thông Tin, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội.

[2] Lê Quang Dũng (2014), *Chống tấn công gây nghẽn mạng cảm biến không dây*, Luận văn Thạc sĩ, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội.

Tài liệu tiếng Anh

[3] The IEEE standards association updated June 2016, *IEEE 802.11-2016 standard*, https://standards.ieee.org/standard/802_11-2016.html.

[4] William Stallings (2005), *Wireless Communications And Networks, 2e*, Pearson Education, Inc.

[5] Alan Holt, Chi-Yu Huang (2010), *802.11 Wireless Networks Security and Analysis*, Springer.

[6] W. Xu, W. Trappe, Y. Zhang, and T. Wood (2005), "The feasibility of launching and detecting jamming attacks in wireless networks", in Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing.

[7] D. J. Theunte and M. Acharya (2006), "Intelligent jamming in wireless networks with applications to 802.11b and other networks", in Proceedings of the 25th IEEE Communication Society Military Communications Conference (MILCOM 2006), vol. 7.

[8] S. M. Kay(1998), *Fundamentals of Statistical Signal Processing, Detection Theory, Volume II*, Prentice Hall PTR, pp 126-133.

[9] G. Bianchi(2000), "Performance analysis of the ieee 802.11 distributed coordination function",IEEE Journal On Selected Areas In Communications, vol. 18, pp 535-547.

[10] Ravieteja Chinta(2009), *Jamming And Anti-Jamming In IEEE 802.11 Wireless Lans*, Master thesis, University of Florida.

[11] NS Nam orgnaization. [Online]. https://www.nsnam.org/releases/ns-3-29/documentation/.

PHỤ LỤC

1. File kịch bản mô phỏng: jamming-attack-vs0.tcl

```
# Filename= jamming-attack.tcl
# Created by MS student Chu Minh Duc, June 2019
# under suggestion of Assoc.Prof. Nguyen Dinh Viet
# ================================================
set val(chan)  Channel/WirelessChannel
set val(prop)  Propagation/TwoRayGround
set val(ant)  Antenna/OmniAntenna
set val(ll)   LL
set val(ifq)  Queue/DropTail/PriQueue
set val(ifqlen) 50
set val(netif) Phy/WirelessPhy
set val(mac)  Mac/802_11
set val(rp)   DSDV
set val(nn)   102
set val(x)   700
set val(y)   500
set val(rlen)      10              ;#gioi han vuong cho ma tran
set val(stop)  200        ;# thoi gian dung mo phong
set cbr_start      40.0            ;# in seconds (old value 10.5)
set cbr_stop  190                  ;# in seconds
set cbr_pkt_size 0.0               ;# bytes
set cbr_pkt_interval 0.0        ;# second

set ns [new Simulator]
```

```
set tracefd    [open jamming-attack.tr w]
set namtrace   [open jamming-attack.nam w]

$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
set topo    [new Topography]
$topo load_flatgrid $val(x) $val(y)

set god_ [create-god $val(nn)]
set chan_1_ [new $val(chan)]

# Thiet lap luu luong CBR giua Scr, Dest nodes ----------------------
# (sent from node 0 to node 99 --------------------------------------
proc cbrtraffic { src dst starttime stoptime cbrpktsize cbrpktinterval} {
  global ns node_
  set udp_($src) [new Agent/UDP]
  eval $ns attach-agent \$node_($src) \$udp_($src)
  set null_($dst) [new Agent/Null]
  eval $ns attach-agent \$node_($dst) \$null_($dst)
  set cbr_($src) [new Application/Traffic/CBR]
  eval \$cbr_($src) set packetSize_ $cbrpktsize
  eval \$cbr_($src) set interval_ $cbrpktinterval
# eval \$cbr_($src) set rate_ 250kbps ;#interval=0.015625s
  eval \$cbr_($src) set random_ 0
  eval \$cbr_($src) attach-agent \$udp_($src)
  eval $ns connect \$udp_($src) \$null_($dst)
```

```
  $ns at $starttime "$cbr_($src) start"
  $ns at $stoptime "$cbr_($src) stop"
}
;# ---------------------------------------------------------------------
$ns node-config         -adhocRouting $val(rp) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \
                -phyType $val(netif) \
                -topoInstance $topo \
                                -channel $chan_1_ \
                -agentTrace ON \
                -routerTrace ON \
                -macTrace OFF \
                -movementTrace OFF\
        -energyModel "EnergyModel" \
                        -initialEnergy 50 \
                        -txPower 0.9 \
                        -rxPower 0.5 \
                        -idlePower 0.05 \
                        -sensePower 0.0175
;# Create nn nodes ----------------------------------------------------
for {set i 0} {$i < $val(nn) } { incr i } {
```

```
      set node_($i) [$ns node]
   }
;# ----------------------------------------------------------------------
;# Arrange nodes in a matrix rlen x rlen
set nodespace 40.0       ;# khoang cach giua cac node
set ttn      0                    ;# set tong node tam
for {set i 0} {$i < $val(rlen) } {incr i} {
    for {set j 0} {$j < $val(rlen) } {incr j} {
            set a $ttn
            $node_($a) set X_ [expr 0.0 + [ expr $i * $nodespace]]
            $node_($a) set Y_ [expr 0.0 - [ expr $j * $nodespace]]
            $node_($i) set Z_ 0.0
            #puts "gia tri cua a $a"
            incr ttn;
            #puts "gia tri cua ttn $ttn"
    }
    }
;# ----------------------------------------------------------------------
# set vi tri cua jammers next to node 0
    $node_(100) set X_ [expr 0.0 + [ expr 4.5 * $nodespace]]
    $node_(100) set Y_ [expr 0.0 - [ expr 4.5 * $nodespace]]
    $node_(100) set Z_ 0.0
    $node_(100) label "Jammer1"
    puts "Jammer 1 (send node)"

    $node_(101) set X_ [expr 0.0 + [ expr 5.0 * $nodespace]]
```

```
    $node_(101) set Y_ [expr 0.0 - [ expr 4.5 * $nodespace]]
    $node_(101) set Z_ 0.0
    $node_(101) label "Jammer2"
    puts "Jammer 2 (receive node)"
;# ---------------------------------------------------------------------
# 2 last value parameters of cbrtraffic are cbr packet size, and cbr pkt interval
cbrtraffic 0 99 $cbr_start $cbr_stop 500 0.2
cbrtraffic 100 101 [expr $cbr_start + 10.0] $cbr_stop 500 0.2 ;# this is
    jamming traffic
;# ---------------------------------------------------------------------
#setting initial position hien thi node len nam voi gia tri set la size cua node
for {set i 0} {$i < $val(nn)} { incr i } {
$ns initial_node_pos $node_($i) 10
}
;# ---------------------------------------------------------------------
#Tell nodes when the simulation ends
for {set i 0} {$i < $val(nn) } {incr i} {
  $ns at $val(stop) "$node_($i) reset";
}
;# ---------------------------------------------------------------------
#$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at [expr $val(stop) + 0.2] "puts \"end simulation\" ; $ns halt"
puts "DEBUG"
;# ---------------------------------------------------------------------
proc stop {} {
```

```
  global ns tracefd namtrace
  $ns flush-trace
  close $tracefd
  close $namtrace
}
;# ---------------------------------------------------------------------
puts "\nStarting Simulation..."
$ns run
```

2. File dùng để trích xuất thông tin vẽ biểu đồ: column.

```
    #!/usr/bin/perl
    # Mark Claypool
    # Last significantly modified: April 27, 1994
    # This program prints out fields of an indicated column.
    # The columns are numbered 1, 2, 3 ...

    &ParseCommandLine;
    $line = <STDIN>;
    while ($line) {
      $line =~ s/^\s+//;      # remove initial white-space
      $line =~ s/\s+/ /g;      # turn double-space into single space
      @word = split('\s+',$line);  # columns will then be $1, $2, $3 ...
      $i =0;                          #
      while ($i <= $#col) {
        print "@word[@col[$i]]\t";
    $i += 1;
```

```
  }
  print "\n";
  $line = <STDIN>;
 }
 exit;

 ##################################################################
#############
 # ParseCommandLine
 # check for the right number of command line arguments
 # print usage message and quit if there is an error
 # global variables are @col
 sub ParseCommandLine
 {
  while ($#ARGV >= 0) {
 $arg = shift(@ARGV);
 if ($arg =~ /^(\d+)/) {
  push(@col, $1);
 } else {
  &usage;
 }
  }
  if ($#col < 0) {
 &usage;
  }
```

```
}

################################################################
###############
# usage
# print a usage maessage and quit
sub usage
{
  print STDERR "column: print fields from an indicated column\n";
  print STDERR "Usage: column <flags>, where flags are:\n";
  print STDERR "    {# [#...]}\tcolumn(s) to print, numbered 0,1,2...\n";
exit;
}
```