

BỘ GIÁO DỤC VÀ ĐÀO TẠO

BỘ CÔNG THƯƠNG

VIỆN NGHIÊN CỨU ĐIỆN TỬ, TIN HỌC TỰ ĐỘNG HÓA

----*----**

VŨ TÁT THÀNH

**MÔ HÌNH TỰ THÍCH NGHI – GIAO THỨC HỌ TCP CHO CÁC ỨNG
DỤNG ĐA PHƯƠNG TIỆN TRONG MẠNG KHÔNG DÂY**

Chuyên ngành: KỸ THUẬT ĐIỆN TỬ

Mã số: 62 52 02 03

LUẬN ÁN TIẾN SỸ KỸ THUẬT

Hà nội 2014

LỜI CAM ĐOAN

Tôi xin cam đoan rằng các kết quả khoa học được trình bày trong luận án này là thành quả nghiên cứu của bản thân tôi trong suốt thời gian làm nghiên cứu sinh và chưa từng xuất hiện trong công bố của các tác giả khác. Các kết quả đạt được là chính xác và trung thực.

Tác giả luận án

LỜI CẢM ƠN

Trước hết, tôi xin bày tỏ lời cảm ơn sâu sắc đến PGS. TSKH. Nguyễn Hồng Vũ, TS. Ngô Văn Sỹ đã hướng dẫn trực tiếp nghiên cứu cũng như đã hỗ trợ về mọi mặt để tôi hoàn thành bản luận án này. Tôi xin cảm ơn GS. TSKH Dietrich Reschke (Trường ĐH Kỹ thuật Ilmenau, CHLB Đức) đã truyền cho tôi cảm hứng nghiên cứu khoa học.

Qua đây, Tôi xin gửi lời cảm ơn GS. TSKH Nguyễn Xuân Quỳnh, TS. Phạm Thế Truyền và Viện Nghiên cứu Điện Tử Tin Học Tự động hóa, những người luôn cho tôi sự tư vấn, hỗ trợ kịp thời và tạo điều kiện thuận lợi cho tôi ngay từ những ngày đầu học tập làm nghiên cứu sinh.

Cuối cùng, tôi dành lời yêu thương nhất đến gia đình tôi: bố mẹ, em trai và đặc biệt là vợ cùng hai con. Sự động viên, giúp đỡ và hy sinh, nhẫn nại của gia đình là động lực mạnh mẽ giúp tôi vượt qua khó khăn để hoàn thành luận án này.

Xin chân thành cảm ơn!

Hà nội, ngày 18 tháng 11 năm 2014
Tác giả luận án

Vũ Tất Thành

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ VIẾT TẮT	6
DANH MỤC HÌNH VẼ	8
MỞ ĐẦU	9
CHƯƠNG 1. ĐIỀU KHIỂN LƯU LƯỢNG VÀ TẮC NGHẼN	12
1.1 Mạng máy tính và ứng dụng	12
1.2 Mạng Internet và mô hình TCP/IP	13
1.2.1 Tầng Internet	14
1.2.2 Tầng giao vận	14
1.2.3 Tầng Ứng dụng	15
1.2.4 Tầng Host-to-Network	15
1.3 Điều khiển tắc nghẽn trong TCP truyền thống	15
1.3.1 Cơ bản về điều khiển lưu lượng và điều khiển tắc nghẽn	16
1.3.1.1 Khái niệm	16
1.3.1.2 Các tầng có thể thực hiện điều khiển luồng	16
1.3.1.3 Điều khiển luồng theo cơ chế cửa sổ	17
1.3.1.4 Biên nhận từng gói số liệu	18
1.3.1.5 Biên nhận ở cuối cửa sổ	19
1.3.1.6 Điều khiển tắc nghẽn	21
1.3.1.7 Điều khiển luồng trong giao thức TCP	24
1.3.1.8 Tính thời gian khứ hồi và thời gian hết hạn gói tin	28
1.3.1.9 Rút lui theo hàm mũ	28
1.3.1.10 Tránh tắc nghẽn	29
1.4 Điều khiển tắc nghẽn trong mạng có kết nối phức tạp	30
1.4.1 TCP và ứng dụng đa phương tiện	30
1.4.2 Các vấn đề ảnh hưởng QoS trên mạng không dây	31
1.4.3 Hiệu suất của giao thức TCP trong mạng có đường truyền không dây	32
1.4.4 Ảnh hưởng của đặc tính lỗi đường truyền không dây	32
1.4.5 Ảnh hưởng của sự gián đoạn kết nối thường xuyên	33
1.5 Đặt vấn đề nghiên cứu	36
1.6 Kết luận chương I	41
CHƯƠNG 2. CÁC GIẢI PHÁP ĐIỀU KHIỂN TẮC NGHẼN TRONG MẠNG CÓ KẾT NỐI PHỨC TẠP	43
2.1 Cấu trúc mạng có kết nối phức tạp – mạng có kết nối không dây	43
2.2 Các kỹ thuật nhằm cải thiện hiệu năng TCP cho mạng không dây	44
2.2.1 Che giấu phần mạng hay làm mất gói số liệu do lỗi đường truyền	45
2.2.1.1 Các giải pháp ở tầng liên kết dữ liệu	45
2.2.1.2 Các giải pháp ở tầng giao vận	47
2.2.1.3 Các giải pháp liên tầng	50
2.2.2 Thông báo rõ ràng về nguyên nhân mất gói số liệu	52
2.3 Xác định nhu cầu băng thông và trạng thái đường truyền	53
2.4 Kết luận chương II	62

CHƯƠNG 3. MÔ HÌNH ĐIỀU KHIỂN TRUYỀN THÔNG TỰ THÍCH	
NGHI CHO HỌ GIAO THỨC TCP TRONG MẠNG KHÔNG DÂY.....	64
3.1 Mô hình điều khiển thích nghi mạng không dây	64
3.2 Thuật toán tính RTT	71
3.2.1 Phân tích công thức RTT theo hàm thống kê.....	72
3.2.2 Đề xuất phương pháp tính RTT phù hợp môi trường không dây.....	74
3.3 Giao thức tự thích nghi họ TCP cho môi trường không dây WRCAP	79
3.4 Kết luận chương III.....	87
KẾT LUẬN.....	89
DANH SÁCH CÁC CÔNG TRÌNH ĐÃ CÔNG BỐ.....	91
TÀI LIỆU THAM KHẢO.....	92

DANH MỤC CÁC THUẬT NGỮ VIẾT TẮT

ARQ	Automatic Repeat reQuest	Yêu cầu gửi nhắc lại tự động
ACK	Acknowledgement	Xác nhận nhận được gói tin
ADW	Adaptive Delayed Acknowledgment Strategy	Chiến lược trì hoãn xác nhận thích nghi
AIMD	Additive Increase, Multiplicative Decrease	Chính sách tăng theo cấp số cộng, giảm theo cấp số nhân
ARPANET		
AP	Access Point	Thiết bị /Điểm truy cập
BS	Base Station	Trạm gốc
BER	Bit Error Rate	Tỉ lệ lỗi bit
BS	Base Station	Trạm gốc
BSD	Berkeley Software Distribution	Hệ điều hành Linux của Berkeley
CA	Congestion Avoidance	Thuật toán tránh tắc nghẽn
CBR	Constant Bit Rate	Tốc độ bit không đổi
CWND	Congestion window	Cửa sổ tắc nghẽn
DNS	Domain Name System	Hệ thống tên miền
ECN	Explicit Congestion Notification	Thông báo tường minh tắc nghẽn
ECIMD	Exponential Combination Increase, Multiplicative Decrease	Chính sách tăng theo hàm mũ kết hợp, giảm theo cấp số nhân
ELN	Explicit Loss Notification	Thông báo tường minh mất gói
FEC	Forward Error Correcting	Sửa lỗi hướng đi
FH	Fixed Host	Trạm cố định
FTP	File Transfer Protocol	Giao thức truyền file
HDLC	High-Level Data Link Control	Giao thức kết nối dữ liệu bậc cao
I/O	Input / Output	Vào / Ra
IP	Internet Protocol	Giao thức Internet
I-TCP	Indirect TCP	TCP gián tiếp
LAN	Local Area Network	Mạng cục bộ
MTU	Maximum Transmission Unit	Đơn vị truyền thông tối đa
MH	Mobile Host	Trạm di động;
NRT	Non-congestion Retransmission Timeout	Hết thời gian truyền lại khi không tắc nghẽn
OSI	Open Systems Interconnection	Mô hình liên kết các hệ thống mở
QoS	Quanlity of Service	Chất lượng dịch vụ
RED	Random Early Detection	Phát hiện/ Hủy bỏ sớm ngẫu nhiên
RTO	Round trip Time Out	Thời gian hết giờ
RTT	Round Trip Time	Thời gian khứ hồi
SATNET	Satellite Network	Mạng vệ tinh
SACK	Selective ACK	Xác nhận có lựa chọn
SMTP	Simple Mail Transfer Protocol	Giao thức trao đổi thư đơn giản

SRTT	Sample RTT	Giá trị thời gian khứ hồi mẫu
SS	Slow Start	Thuật toán khởi động chậm
TCP	Transmission Control Protocol	Giao thức Điều khiển truyền thông
TELNET	Network Virtual Terminal Protocol	Giao thức Trạm ảo mạng
TFRC	TCP Friendly Rate Control	Giao thức họ TCP, điều khiển tốc độ
UDP	User Datagram Protocol	Giao thức tầng giao vận không xác nhận
WRCAP	Wireless ready control adaptive protocol	Giao thức điều khiển thích nghi cho môi trường không dây

DANH MỤC HÌNH VẼ

Hình 1.1 Mô hình tham chiếu TCP/IP và mô hình tham chiếu OSI	13
Hình 1.2 Các giao thức và các mạng trong mô hìnhTCP/IP ban đầu	15
Hình 1.3 Điều khiển lưu lượng bằng cửa sổ, biên nhận từng gói số liệu	18
Hình 1.4 Điều khiển lưu lượng bằng cửa sổ, biên nhận ở cuối cửa sổ. Kích thước cửa sổ nhận và gửi ban đầu bằng 3	19
Hình 1.5 Điều khiển lưu lượng bằng cửa sổ, biên nhận ở cuối cửa sổ.	20
Hình 1.6 Sự xuất hiện tắc nghẽn khi quá tải	21
Hình 1.7 Sự phân lớp các giao thức	25
Hình 1.8 Sự tăng của cửa sổ trong cơ chế khởi động chậm	27
Hình 1.9 Mạng để nghiên cứu thực nghiệm về liên mạng di động	34
Hình 1.10 Ảnh hưởng của tỉ lệ lỗi bit (BER) cao và Sự chuyển cuộc gọi đến hiệu suất của TCP	35
Hình 1.11 Mô hình mạng điều khiển tắc nghẽn và lưu lượng giữa hai đầu cuối	37
Hình 1.12 Kiến trúc nguyên lý điều khiển thích nghi	38
Hình 1.13 Mô hình điều khiển tự thích nghi	40
Hình 2.1 Ví dụ mô hình mạng có kết nối phức tạp	44
Hình 2.2 Nghiên cứu 2 nút mạng tổng quát	54
Hình 2.3 Mô hình xác định băng thông khả dụng	56
Hình 2.4 So sánh giá trị $R(t)$ theo công thức (2.7) và (2.11).	60
Hình 2.5 Biến đổi của $R(t)$ theo giá trị Tqs .	61
Hình 2.6 Biến đổi của $R(t)$ theo trọng số dung sai X	61
Hình 3. 1 Cơ chế điều khiển thích nghi	65
Hình 3. 2 Biến đổi của Thông lượng theo p	70
Hình 3. 3 Biến đổi của Thông lượng theo RTT	70
Hình 3. 4 Các tình huống lấy mẫu RTT bị sai	71
Hình 3.5 Mô phỏng quan hệ WR theo giá trị α , N	75
Hình 3.6 Lưu đồ thuật toán EWMA RTT	76
Hình 3.7 Giá trị RTT đo bằng phương pháp EWMA RTT	78
Hình 3.8 Nguyên lý hoạt động của WRCAP	80
Hình 3.9 Theo dõi QoS: Di động - Cố định	82
Hình 3.10 Theo dõi QoS: Cố định - Di động	83
Hình 3.11 Mô hình các trạng thái của giao thức tự thích nghi WRCAP	84
Hình 3.12 Thông lượng nguồn TCP và WRCAP	87

MỞ ĐẦU

Xu hướng hội tụ về công nghệ mạng IP và tính chất đa dạng của các mạng máy tính, viễn thông trong tương lai sẽ khiến việc đảm bảo chất lượng dịch vụ cho các kết nối thông qua các mạng này trở thành một thử thách lớn, đặc biệt khi mô hình mạng hiện nay là hỗn hợp, bao gồm nhiều loại kênh truyền khác nhau.

Xu hướng sử dụng ngày càng phổ biến các ứng dụng đa phương tiện, chứa nhiều loại dữ liệu dung lượng lớn như thoại, ảnh, video... cũng yêu cầu băng thông cấp phát cho mỗi ứng dụng này càng cao.

Mặc dù băng thông của các mạng không dây thế hệ mới này đã được cải thiện, và công nghệ mạng có dây thế hệ mới cũng không ngừng phát triển, bởi vậy mạng không dây vẫn sẽ là nơi thắt nút của mạng hỗn hợp, gồm các thiết bị trong mạng có dây, như Internet, đến thiết bị của mạng không dây (ví dụ như một handy phone). Trong kết nối mạng có dây vào mạng không dây, điểm kết nối thường xảy ra tắc nghẽn.

Nguyên nhân gây nên việc mất các gói tin trong mạng không dây khác xa các giả thiết về nguyên nhân gây mất các gói tin khi thiết kế các giao thức truyền thông truyền thống như TCP/IP.

Vì vậy nhu cầu cần thiết phải xây dựng một mô hình tự thích nghi, thông qua việc đo băng thông tức thời, phát hiện chất lượng đường truyền, điều chỉnh tốc độ truyền tin để đảm bảo chất lượng của các ứng dụng đa phương tiện, trong mạng không dây là rất cần thiết. Đây chính là nội dung của công trình nghiên cứu này.

Mục tiêu của luận án là đề xuất phương pháp xác định nhanh chóng trạng thái kênh truyền, điều chỉnh phương pháp tính thời gian khứ hồi gói tin, từ đó xây dựng một mô hình thích nghi với sự thay đổi tham số của môi trường mạng, đặc biệt có thể biến thiên với phần mạng không dây. Luận án áp dụng kết quả nghiên cứu lý thuyết vào xây dựng thử nghiệm một giao thức truyền thông thuộc họ giao thức TCP, cải thiện thông lượng, hội tụ nhanh, thích ứng tốt với lỗi mất gói tin cho ứng dụng đa

phương tiện trong mạng hỗn hợp, có sử dụng các thiết bị di động như các đầu cuối để thu phát tín hiệu với các trạm gốc.

Bố cục của luận án gồm 3 chương.

Chương I giới thiệu mô hình tham chiếu TCP/IP, là chuẩn để các mạng máy tính khác nhau có thể truyền thông tin với nhau, và phân tích đặc điểm thiết kế của giao thức truyền thông TCP. Với giả định ban đầu nguyên nhân gây mất gói tin trên mạng là do tắc nghẽn, nên TCP đã ứng dụng lý thuyết kiểm soát lưu lượng, chống tắc nghẽn vào thiết kế của mình. Chương I trình bày vấn đề điều khiển tắc nghẽn trong mạng có kết nối phức tạp, không đồng nhất, bao gồm đoạn mạng không dây, từ các ứng dụng đa phương tiện, là các ứng dụng phổ biến hiện nay, gồm nhiều loại dữ liệu khác nhau, nhiều loại dịch vụ và ứng dụng với nhu cầu hết sức đa dạng, các thiết bị đầu cuối cũng rất khác nhau về nhiều mặt. TCP được phân tích không đạt hiệu năng cao khi hoạt động trong môi trường mạng như vậy. Chương I đặt mục tiêu nghiên cứu của luận án là xây dựng một mô hình điều khiển thông minh, nhằm phản ứng linh hoạt hơn với các biến cố của đường truyền.

Chương II tổng hợp các đề xuất trong việc khắc phục điểm yếu của giao thức truyền thông họ TCP cho đến nay, bao gồm các giải pháp che giấu phần mạng hay làm mất gói số liệu do lỗi đường truyền, sao cho bên gửi chỉ phát hiện được những sự mất gói số liệu do tắc nghẽn, loại thứ hai bao gồm các giải pháp cải tiến TCP bằng các cơ chế thông báo rõ ràng về nguyên nhân mất gói số liệu, giúp cho TCP có thể phân biệt được các kiểu mất gói số liệu khác nhau. Cả hai nhóm giải pháp đều cố tránh thay đổi cách hoạt động của TCP, do đó không đóng góp nhiều trong việc cải thiện hiệu năng TCP trong phần mạng không dây. Trong chương này, luận án xây dựng công thức tính nhanh chóng nhu cầu băng thông của các kết nối và băng thông khả dụng của đường truyền, trạng thái đường truyền từ mỗi nút mạng, dựa trên tốc độ đến gói tin và kích thước bộ đệm, từ đó cho phép giao thức truyền thông có khả năng nhận biết và điều khiển tắc nghẽn nhanh hơn.

Chương III đề xuất mô hình điều khiển thích nghi, là mô hình điều khiển tổng quát cho các giao thức họ TCP, đảm bảo hiệu suất truyền thông đồng thời sự công bằng giữa các luồng tin. Trong mô hình này cơ chế ECIMD được đề xuất thay thế cho AIMD của TCP, và được phân tích trong tình huống điều chỉnh kích thước cửa sổ truyền, với các giá trị mới của các hệ số điều khiển, đảm bảo hiệu năng và khả năng đáp ứng nhanh với môi trường mạng, đồng thời đảm bảo công bằng giữa các luồng tin do đó hạn chế tắc nghẽn. Nghiên cứu cho thấy trong tình huống việc điều khiển giá trị cửa sổ khi có lỗi đơn cho thấy cơ chế này mang lại thông lượng tốt hơn so với AIMD.

Chương này cũng đề xuất phương pháp tính giá trị thời gian khứ hồi gói tin, dựa trên phân tích tổng trọng số của N mẫu gần nhất. Điều này đặc biệt quan trọng với môi trường không dây hay biến đổi, nên chỉ cần quan tâm đến sự ảnh hưởng của một số giá trị gần nhất.

Mô hình đề xuất đã được áp dụng để xây dựng một giao thức họ TCP là WRCAP và thử nghiệm mô phỏng trong môi trường NS đạt hiệu suất cao hơn, có khả năng phát hiện, phân biệt và phòng lỗi hiệu quả hơn so với các kết quả nghiên cứu đang sử dụng hiện nay trong các giao thức họ TCP, khi chạy trên môi trường hỗn hợp, trong mô hình có trạm gốc và trạm di động.

CHƯƠNG 1. ĐIỀU KHIỂN LƯU LƯỢNG VÀ TẮC NGHẼN

1.1 Mạng máy tính và ứng dụng

Năm 1967, Robert L. G. đã đề xuất một mạng máy tính thí nghiệm, sau đó trở thành mạng ARPANET (Department of Defense Advanced Research Projects Agency Network) của Bộ Quốc phòng Mỹ. Ngay từ năm 1967 người ta đã nhận ra rằng các mạng này sẽ là những tài nguyên tính toán quý giá, đem lại nhiều lợi ích cho Bộ Quốc phòng Mỹ cũng như cho cộng đồng khoa học, nếu chúng cung cấp các dịch vụ truyền thông, cho phép truy cập từ xa tới tất cả tài nguyên của hệ thống [6].

Năm 1969, Bộ Quốc phòng Mỹ triển khai thực hiện mạng ARPANET với hãng BBN (Bolt, Beranek, and Newman). Đến tháng 9 năm 1969, mạng máy tính chuyển mạch gói đầu tiên trên thế giới ra đời, kết nối mạng của các trường đại học, các trung tâm nghiên cứu của chính phủ và của các hãng công nghiệp trên khắp nước Mỹ.

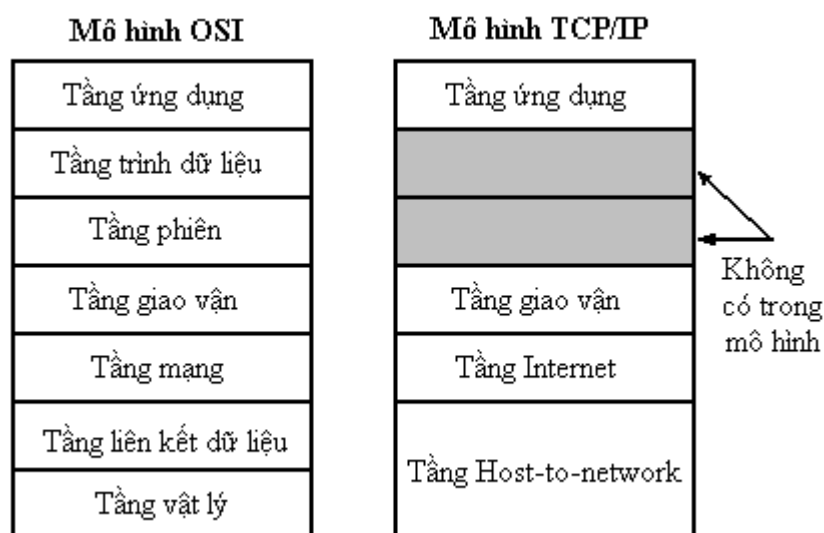
Trong quá trình nghiên cứu và triển khai mạng ARPANET, nhiều tư tưởng và phương pháp mới lần đầu tiên được đề xuất và đưa vào thực hiện, trong đó có giao thức, mạng rôi (mesh network), điều khiển lưu lượng (flow control), đặc tính chịu lỗi. Với các đặc tính này, mạng vẫn có khả năng hoạt động được khi có một số nút hoặc đường truyền bị hỏng mà không cần sự can thiệp của người điều hành. Ngoài ra, những người nghiên cứu và thực hiện mạng ARPANET cũng đã sử dụng một cách phổ biến các mô hình giải tích và mô hình mô phỏng để dự đoán và đánh giá hiệu suất mạng.

Mạng ARPANET là tiền thân của mạng Internet. Khi các mạng vệ tinh và vô tuyến ra đời thì các giao thức đang được sử dụng không đáp ứng được yêu cầu liên mạng, do đó cần phải có các mô hình kiến trúc mới, có khả năng liên kết nhiều mạng với nhau một cách trong suốt. Kiến trúc mới này được gọi là mô hình tham chiếu TCP/IP [6] [27].

1.2 Mạng Internet và mô hình TCP/IP

Các mạng máy tính hiện đại được thiết kế bằng cách phân chia cấu trúc ở mức độ cao nhằm làm giảm độ phức tạp của việc thiết kế; mạng được chia thành các tầng (layer), hay còn gọi là mức hoặc lớp, mỗi tầng được xây dựng dựa trên tầng bên dưới nó. Trong các mạng khác nhau, số tầng, tên, nội dung và chức năng của các tầng có thể khác nhau. Tuy nhiên, mỗi tầng trên sử dụng các dịch vụ do các tầng bên dưới cung cấp và cung cấp những dịch vụ nhất định cho các tầng cao hơn, sao cho các tầng này khi sử dụng các dịch vụ đó không cần phải quan tâm tới các thao tác chi tiết mà các dịch vụ phải thực hiện.

Để các mạng máy tính khác nhau có thể truyền thông tin với nhau, chúng cần phải tuân theo các chuẩn. Người ta đã xây dựng nên các chuẩn như vậy và chúng còn được gọi là mô hình tham chiếu, làm cơ sở chung cho các nhà thiết kế dựa vào khi thiết kế mạng. Sau đây mô hình tham chiếu TCP/IP sẽ được trình bày và so sánh với mô hình tham chiếu OSI truyền thống trong truyền thông.



Hình 1.1 Mô hình tham chiếu TCP/IP và mô hình tham chiếu OSI

1.2.1 Tầng Internet

Tầng Internet còn được gọi là tầng IP, có chức năng tương đương tầng mạng trong mô hình OSI. Nhiệm vụ của tầng Internet là định tuyến gói số liệu, điều khiển lưu lượng và điều khiển tắc nghẽn. Mỗi gói số liệu có thể đi tới điểm đến theo các con đường khác nhau; tại điểm đến, thứ tự nhận các gói số liệu có thể khác với thứ tự lúc chúng được phát đi từ nguồn, do đó tầng giao vận sẽ phải giải quyết vấn đề này. Tầng Internet định nghĩa một khuôn dạng gói số liệu và giao thức chính là giao thức IP.

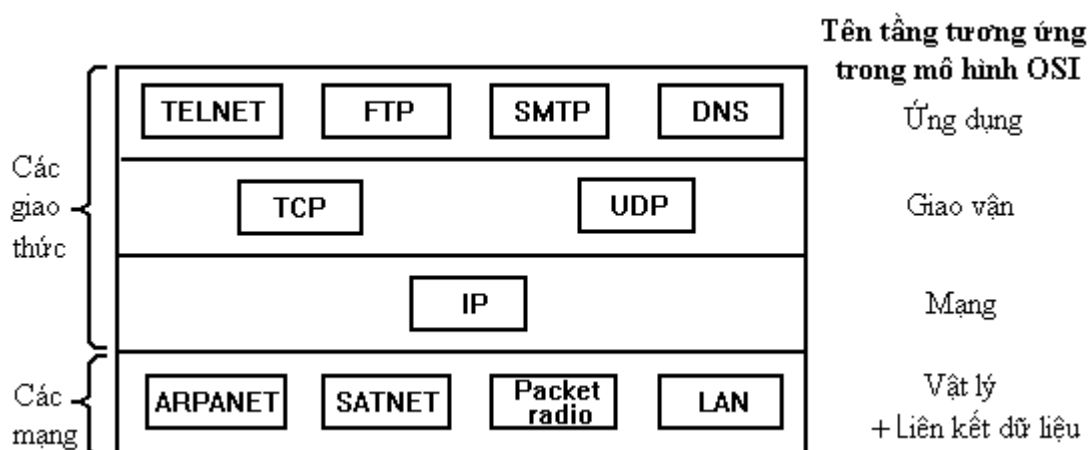
1.2.2 Tầng giao vận

Tầng này phải được thiết kế sao cho các thực thể ngang hàng ở máy nguồn và máy điểm đến có thể truyền thông với nhau, tương tự như trong mô hình OSI. Tại tầng này, người ta định nghĩa hai giao thức kiểu đầu cuối - đầu cuối là TCP (Transmission Control Protocol) và UDP (User Datagram Protocol).

TCP là giao thức hướng kết nối, được đảm bảo, nó vận chuyển dòng byte sinh ra từ máy (nguồn) tới một máy tùy ý khác (điểm đến) trong liên mạng mà không có lỗi. TCP phân mảnh dòng byte từ các tầng trên đi xuống thành các gói số liệu riêng biệt rồi chuyển từng gói số liệu này xuống cho tầng Internet. Tại máy điểm đến, tiến trình TCP nhận và thực hiện lắp ráp các gói số liệu nhận được lại thành dòng byte rồi chuyển lên tầng trên. TCP cũng có các chức năng điều khiển lưu lượng và điều khiển tắc nghẽn, để đảm bảo bên gửi tốc độ cao không "làm lụt" bên nhận có tốc độ thấp và để tránh tắc nghẽn mạng. Vấn đề điều khiển lưu lượng trong giao thức TCP sẽ được nghiên cứu tại mục 1.3 của chương này.

UDP là giao thức không hướng kết nối, không đảm bảo (không có sự biên nhận cho gói số liệu UDP), dành cho các ứng dụng không muốn sử dụng các chức năng điều khiển lưu lượng và phân phát các gói số liệu theo đúng thứ tự như TCP mà muốn tự cung cấp các chức năng này. UDP được sử dụng rộng rãi trong các ứng dụng kiểu dùng một lần, khách/chủ và các ứng dụng mà trong đó việc phân phát tin

nhANH chóng quan trọng hơn việc phân phối tin chính xác. Mối quan hệ giữa IP, TCP và UDP được thể hiện trên hình 1.2.



Hình 1.2 Các giao thức và các mạng trong mô hình TCP/IP ban đầu

1.2.3 Tầng Ứng dụng

Tầng ứng dụng gồm các giao thức bậc cao, như các giao thức TELNET (virtual terminal protocol), FTP (File Transfer Protocol) và SMTP (Simple Mail Transfer Protocol) v.v.

1.2.4 Tầng Host-to-Network

Bên dưới tầng Internet là một khoảng trống lớn, mô hình tham chiếu TCP/IP thực tế hầu như không nói gì về tầng này, ngoài việc chỉ ra rằng máy tính (host) phải nối với mạng bằng cách sử dụng một số giao thức để có thể gửi các gói số liệu IP đi trên mạng. Tầng này không được định nghĩa và nó khác nhau trên các máy tính khác nhau cũng như trên các mạng khác nhau [6].

1.3 Điều khiển tắc nghẽn trong TCP truyền thống

Nói chung, nếu không có một cơ chế điều khiển nào đối với các lưu lượng đến mạng thì tắc nghẽn là điều chắc chắn sẽ xảy ra. Chính vì vậy, các quy tắc hay các cơ chế điều khiển lưu lượng, cần được triển khai thực hiện đúng đắn, nếu không, mạng sẽ bị tắc nghẽn, đó là trạng thái khi lưu lượng đến mạng tăng lên, thông lượng vận

chuyển của mạng lại giảm đi. Thậm chí thông lượng vận chuyển của mạng có thể bằng không ở trạng thái chết tắc [26],[36],[45],[46].

1.3.1 Cơ bản về điều khiển lưu lượng và điều khiển tắc nghẽn

1.3.1.1 Khái niệm

Điều khiển luồng liên quan đến việc vận chuyển giữa một người gửi nào đó và một người nhận. Nhiệm vụ của nó là đảm bảo rằng bên gửi có tốc độ nhanh không thể tiếp tục truyền dữ liệu nhanh hơn mức mà bên nhận có thể tiếp thu được. Điều khiển luồng luôn luôn liên quan đến một sự phản hồi trực tiếp từ phía người nhận đến người gửi để báo cho bên gửi về khả năng nhận số liệu thực của bên nhận.

Điều khiển tắc nghẽn thực hiện nhiệm vụ đảm bảo cho mạng có khả năng vận chuyển lưu lượng đưa vào.

Điều khiển luồng và điều khiển tắc nghẽn là hai khái niệm khác nhau, nhưng liên quan chặt chẽ với nhau. Điều khiển luồng là để tránh tắc nghẽn, còn điều khiển tắc nghẽn là để đề phòng tắc nghẽn trước khi nó xuất hiện và giải quyết tắc nghẽn khi nó có dấu hiệu xảy ra. Trong thực tế triển khai thực hiện các thuật toán điều khiển luồng và điều khiển tắc nghẽn, nhiều khi cả hai thuật toán này cùng được cài đặt trong một giao thức, thể hiện ra như là một thuật toán duy nhất, thí dụ trong giao thức TCP [24] [30].

1.3.1.2 Các tầng có thể thực hiện điều khiển luồng

Có thể thực hiện điều khiển lưu lượng ở một vài tầng trong mạng, thí dụ:

Điều khiển luồng ở tầng giao vận: thường được gọi là điều khiển lưu lượng đầu cuối - đầu cuối: nhằm tránh cho bộ đệm của quá trình nhận tại điểm đến khỏi bị tràn.

Điều khiển luồng trên từng chặng: nhằm tránh cho từng đường truyền khỏi bị tắc nghẽn. Tuy nhiên, việc điều khiển luồng trên từng chặng sẽ có ảnh hưởng đến các chặng khác, do đó nó cũng có tác dụng tránh tắc nghẽn cho các đường truyền có

hiều chặng. Trong mô hình tham chiếu OSI, điều khiển lưu lượng theo từng chặng được thực hiện ở tầng liên kết dữ liệu và tầng mạng.

1.3.1.3 Điều khiển luồng theo cơ chế cửa sổ

Điều khiển lưu lượng bằng cửa sổ trượt là một trong các cơ chế điều khiển lưu lượng được sử dụng rộng rãi nhất, có thể áp dụng tại một hay nhiều tầng của mạng.

Cơ chế điều khiển luồng bằng cửa sổ cho phép bên gửi phát đi liên tiếp một số gói số liệu nhất định rồi dừng lại chờ thông báo biên nhận, trước khi tiếp tục phát. Bên nhận điều khiển lưu lượng bằng cách kim lại hay gửi ngay biên nhận, hoặc một gói số liệu có chứa thông tin điều khiển, dùng để báo cho bên gửi biết về việc đã nhận một hay một số gói số liệu như thế nào. Tại mọi thời điểm, bên gửi phải ghi nhớ một danh sách chứa số thứ tự liên tiếp các gói số liệu mà nó được phép gửi đi, nằm trong cửa sổ gửi. Bên nhận cũng duy trì một danh sách gọi là cửa sổ nhận, tương ứng với các gói số liệu mà nó được phép nhận.

Người ta đã đề xuất và sử dụng một số phương thức quản lý cửa sổ khác nhau. Thí dụ như, biên nhận riêng rẽ cho mỗi gói số liệu nhận được, biên nhận ở cuối cửa sổ sau khi đã nhận được gói số liệu cuối cùng trong cửa sổ, hay biên nhận ở đầu cửa sổ v.v.

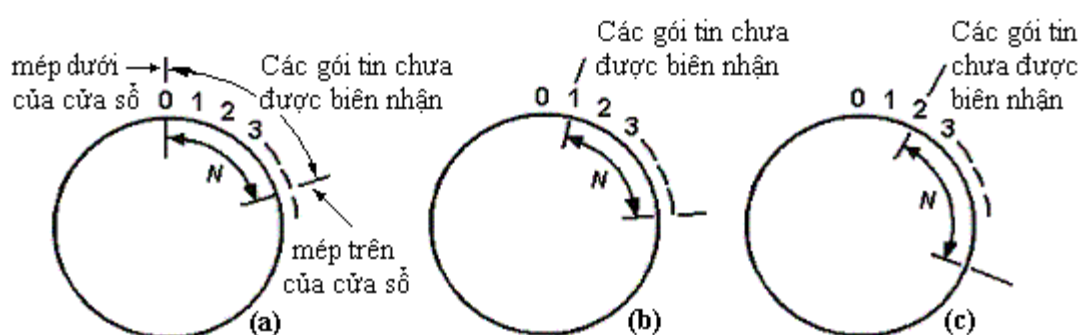
Để việc phân tích sự điều khiển luồng theo cơ chế cửa sổ được thuận lợi, trong luận án sử dụng một số ký hiệu cho các tham số nêu trong bảng sau:

Bảng 1.1 Định nghĩa các tham số

T_t	Thời gian truyền (transmit) một gói số liệu.
T_p	Thời gian truyền tín hiệu cho phép.
τ	Thời gian trễ lan truyền.
W	Kích thước cửa sổ, đơn vị là gói số liệu.
D	Thời gian từ khi bắt đầu truyền gói số liệu đầu tiên trong cửa sổ, cho đến khi nhận được tín hiệu cho phép truyền tiếp.

1.3.1.4 Biên nhận từng gói số liệu

Theo cách quản lý này, mỗi khi nhận được một gói số liệu, bên nhận sẽ gửi một biên nhận cho bên gửi. Cửa sổ gửi tương ứng với các gói số liệu đã gửi đi nhưng chưa được biên nhận (hình 1.3 a). Khi có một gói số liệu mới từ tầng trên chuyển xuống để gửi đi, nó sẽ được gán số thứ tự lớn nhất tiếp theo, do đó mép trên của cửa sổ gửi sẽ tăng thêm 1. Nếu cửa sổ tăng tới cực đại thì tiến trình truyền ở tầng trên bị chặn lại, không thể truyền các gói số liệu xuống nữa, cho đến khi có chỗ trống trong vùng nhớ đệm. Mỗi gói số liệu sau khi tới điểm đến sẽ được bên nhận biên nhận một cách riêng rẽ. Khi biên nhận về đến bên gửi, mép dưới của cửa sổ gửi sẽ được tăng thêm 1, làm cho danh sách các gói số liệu đã truyền nhưng còn chờ biên nhận giảm đi một phần tử, đồng thời vùng nhớ tương ứng với phần tử đó cũng được giải phóng để cấp phát cho một gói số liệu mới (hình 1.3 b, c). Bằng cách này, cửa sổ gửi luôn ghi nhớ được danh sách các gói số liệu còn chưa được biên nhận. Vì các gói số liệu trong cửa sổ gửi có thể bị hỏng hoặc mất trên đường truyền, nên bên gửi phải giữ lại bản sao của chúng trong bộ nhớ đệm để phát lại nếu sau một khoảng thời gian nhất định vẫn không nhận được biên nhận.



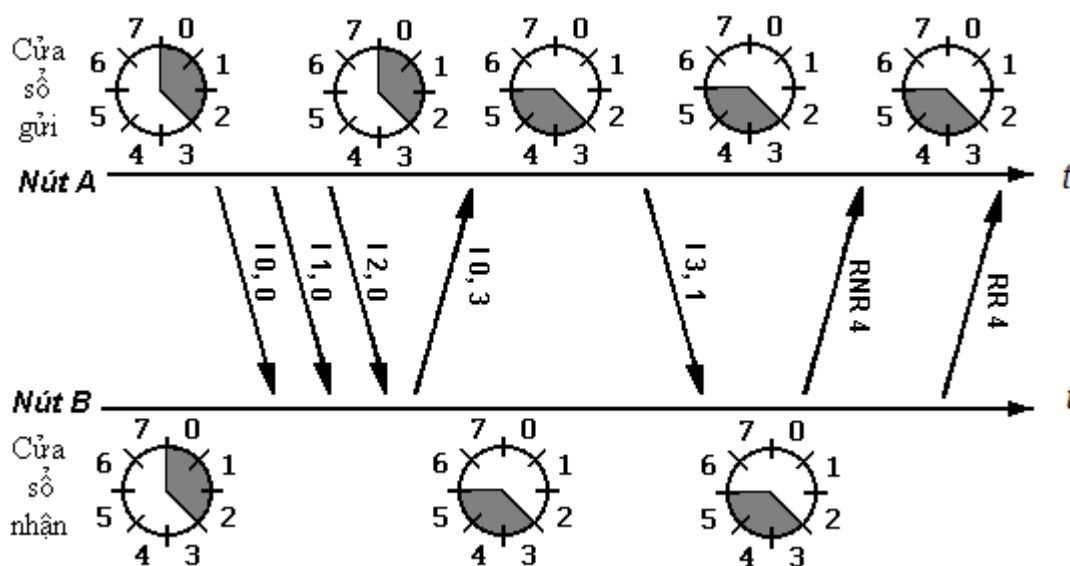
Hình 1.3 Điều khiển lưu lượng bằng cửa sổ, biên nhận từng gói số liệu. a. Trạng thái ban đầu b. Gói số 0 được biên nhận c. Gói số 1 được biên nhận

Cửa sổ bên nhận tương ứng với các gói số liệu mà nó được phép nhận, các gói số liệu nằm ngoài cửa sổ nhận nếu đến sẽ bị loại bỏ. Khi nhận được một gói số liệu có số thứ tự bằng mép dưới của cửa sổ, nó sẽ được truyền cho tầng trên, bên nhận tạo ra một biên nhận để gửi tới người gửi và tăng cửa sổ lên một ô. Nếu kích thước của

sổ nhận bằng 1, có nghĩa là nó chỉ chấp nhận các gói số liệu đến theo đúng thứ tự. Nếu khác 1 thì không phải như vậy, trong trường hợp này, bên nhận sẽ giữ gói số liệu đến không đúng thứ tự trong bộ đệm, chờ nhận đủ các gói số liệu trong cửa sổ rồi mới chuyển các gói số liệu lên tầng trên theo thứ tự mà bên gửi đã gửi đi. Khoảng thời gian chờ này luôn được giới hạn.

1.3.1.5 Biên nhận ở cuối cửa sổ

Đây là cách đơn giản nhất, bên nhận sẽ phát ra một biên nhận sau khi nhận được tất cả các gói số liệu trong cửa sổ nhận. Hình 1.4 minh họa cho phương pháp này, trong đó ví dụ nút A truyền thông với nút B, sử dụng một giao thức tựa như giao thức HDLC, kích thước cửa sổ gửi và cửa sổ nhận ban đầu bằng 3. Các gói số liệu đi trên mạng được biểu diễn bằng các mũi tên, kiểu của gói số liệu được ghi bên cạnh mũi tên, ý nghĩa như sau [21]:



Hình 1.4 Điều khiển lưu lượng bằng cửa sổ, biên nhận ở cuối cửa sổ. Kích thước cửa sổ nhận và gửi ban đầu bằng 3

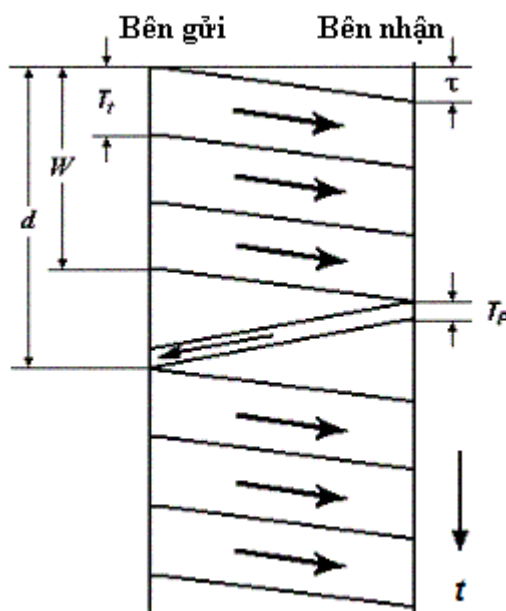
- I n.m: là gói số liệu, với trường số thứ tự gói số liệu $N(S) = n$, trường biên nhận $N(R) = m$. Bên B cho gói số liệu của nó “cồng” (“piggyback”) biên nhận tới bên A, việc này giúp nâng cao hiệu quả sử dụng đường truyền. Tất nhiên, bên A cũng có thể biên nhận các gói số liệu mà B gửi cho nó bằng cách trên.

- RNR 4: là gói số liệu điều khiển, B báo cho A rằng, lúc này nó không thể nhận tiếp các gói số liệu của A, đồng thời biên nhận cho các gói số liệu có số thứ tự nhỏ hơn hoặc bằng 3. Khi nhận được tín hiệu này, A sẽ phải ngừng gửi, chờ cho đến khi nhận được tín hiệu cho phép gửi tiếp của B.
- RR 4: là gói số liệu điều khiển, B báo cho A rằng, lúc này nó sẵn sàng nhận tiếp các gói số liệu của A, bắt đầu từ gói số 4.

Giả sử rằng bên gửi luôn có sẵn dữ liệu để gửi, còn bên nhận cũng gửi biên nhận ngay sau khi nhận được gói số liệu cuối cùng trong cửa sổ nhận. Chúng ta có thể tính được thông lượng trung bình lớn nhất có thể đạt được dựa trên hình 1.5:

$$R_e = \frac{WT_t}{d} R \quad (1.1)$$

với $d = WT_t + T_p + 2\tau$ và R là dung lượng đường truyền giữa A và B.



Hình 1.5 Điều khiển lưu lượng bằng cửa sổ, biên nhận ở cuối cửa sổ.

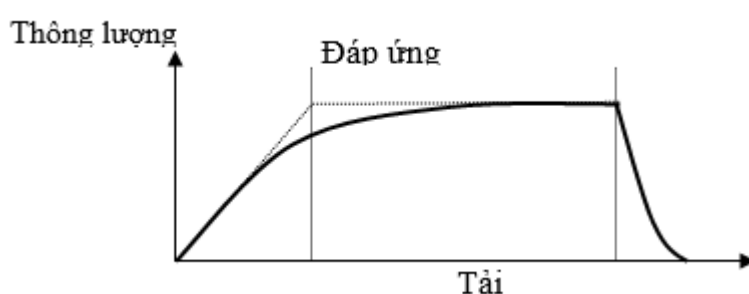
Kích thước cửa sổ $W=3$.

So với cách biên nhận từng gói số liệu, rõ ràng là số lượng biên nhận giảm đi, từ một biên nhận/một gói số liệu, xuống còn một biên nhận/một cửa sổ. Thí dụ, nếu kích thước cửa sổ $W=8$, thì số biên nhận giảm đi 8 lần.

1.3.1.6 Điều khiển tắc nghẽn

Tắc nghẽn

Trong mạng máy tính, tắc nghẽn xảy ra khi số lượng gói số liệu đến nút mạng vượt quá khả năng xử lý của nó hoặc vượt quá khả năng vận tải của các đường truyền ra, điều đó dẫn đến việc thông lượng của mạng bị giảm đi khi lưu lượng đến mạng tăng lên. Hiện tượng tắc nghẽn có thể xảy ra ở một hoặc một số nút mạng, hay trên toàn mạng và được miêu tả trên hình 1.6.



Hình 1.6 Sự xuất hiện tắc nghẽn khi quá tải

Khi số lượng gói số liệu đến mạng còn tương đối nhỏ, nằm trong khả năng vận tải của nó, chúng sẽ được phân phát đi hết, số lượng gói số liệu được chuyển đi tỉ lệ thuận với số lượng gói số liệu đến mạng. Do luôn có một tỉ lệ gói số liệu phải phát lại do bị lỗi trong quá trình vận chuyển, lưu lượng mà mạng thực sự phải vận chuyển nhìn chung lớn hơn lưu lượng đi qua mạng (thông lượng).

Khi lưu lượng đến cao quá một mức nào đó, các nút mạng không còn đủ khả năng chứa và chuyển tiếp các gói số liệu, do đó các nút mạng bắt đầu phải loại bỏ các gói số liệu. Bên gửi sẽ phát lại các gói số liệu không được biên nhận sau một khoảng thời gian nhất định, gọi là thời gian hết giờ. Nếu lưu lượng đến mạng tiếp tục tăng lên nữa, tỉ lệ gói số liệu phát lại trên tổng số gói số liệu trong mạng có thể tăng đến 100%, nghĩa là không có gói số liệu nào được phân phát đi cả, thông lượng của mạng giảm xuống bằng không, mạng bị nghẹt hoàn toàn.

Một số yếu tố có thể dẫn đến tắc nghẽn, ngay cả khi lưu lượng đi vào mạng thấp hơn khả năng vận tải của mạng. Chẳng hạn, khi số lượng gói số liệu đến trên hai hoặc ba lối vào của một nút mạng đều cần đi ra trên cùng một đường truyền để đến điểm đến, chúng sẽ phải xếp hàng đợi được truyền đi. Nếu tình trạng trên kéo dài, hàng đợi sẽ dài dần ra, tới một giới hạn nào đó hàng đợi sẽ đầy, không còn chỗ cho các gói số liệu mới đến, chúng bị loại bỏ và sẽ được phát lại, việc này làm tăng tỉ lệ gói số liệu phát lại trong mạng. Biện pháp khắc phục bằng cách tăng kích thước hàng đợi (bộ nhớ) tại các nút mạng trong một chừng mực nào đó là có ích, tuy nhiên, người ta đã chứng minh được rằng, tăng kích thước hàng đợi quá một giới hạn nào đó sẽ không mang lại lợi ích gì, thậm chí còn có thể làm cho vấn đề tắc nghẽn tồi tệ hơn. Đó là vì các gói số liệu sẽ bị hết giờ ngay trong quá trình xếp hàng, bản sao của chúng đã được bên gửi phát lại rồi, làm tăng số lượng gói số liệu phát lại trong mạng [1] [27] [30].

Tốc độ xử lý chậm của các nút mạng cũng là một nguyên nhân quan trọng gây nên tắc nghẽn, bởi vì chúng có thể sẽ làm hàng đợi bị tràn ngay cả khi lưu lượng gói số liệu đến nút mạng nhỏ hơn năng lực vận tải của đường truyền đi ra. Tương tự như vậy, các đường truyền dung lượng thấp cũng có thể gây ra tắc nghẽn. Việc tăng dung lượng đường truyền nhưng không nâng cấp bộ xử lý tại nút mạng, hoặc chỉ nâng cấp từng phần của mạng đôi khi cũng cải thiện được tình hình đôi chút, nhưng thường chỉ làm cái “cổ chai”, nơi xảy ra tắc nghẽn, dời đi chỗ khác mà thôi. Giải quyết vấn đề tắc nghẽn nói chung, cần đến các giải pháp đồng bộ.

Tắc nghẽn có khuynh hướng tự làm cho nó trầm trọng thêm. Nếu một nút mạng nào đó bị tràn bộ đệm, gói số liệu đến sẽ bị loại bỏ, trong khi đó nút mạng bên trên, phía người gửi, vẫn phải giữ bản sao của gói số liệu đã gửi trong hàng đợi, cho đến khi hết giờ để phát lại. Việc phải giữ bản sao gói số liệu trong hàng đợi để chờ biên nhận, cộng thêm việc có thể phải phát lại gói số liệu một số lần có thể làm cho hàng đợi tại chính nút trên cũng có thể bị tràn. Sự tắc nghẽn lan truyền ngược trở lại phía nguồn phát sinh ra gói số liệu.

Các giải pháp điều khiển tắc nghẽn

Vấn đề điều khiển tắc nghẽn có thể được giải quyết theo quan điểm của lý thuyết điều khiển. Theo cách tiếp cận này, có thể chia các giải pháp thành hai nhóm: các giải pháp vòng mở (open loop) và các giải pháp vòng đóng (closed loop). Theo các giải pháp vòng mở, tắc nghẽn sẽ được giải quyết bằng việc thiết kế tốt. Thực chất đó là việc đảm bảo sao cho tắc nghẽn không xảy ra. Một hệ thống như vậy phải có khả năng quyết định khi nào thì nhận thêm các lưu lượng mới vào, khi nào thì loại bỏ các gói số liệu và loại các gói số liệu nào. Các quyết định này phải theo lịch trình và phải có ở từng nút mạng. Bản chất của điều khiển vòng mở là theo từng chặng.

Trái lại, các giải pháp vòng đóng là điều khiển toàn tuyến, từ đầu cuối đến đầu cuối, dựa trên khái niệm về vòng phản hồi (feedback loop). Chúng gồm có ba phần, hay ba bước như sau:

Bước một: theo dõi hệ thống để phát hiện dấu hiệu tắc nghẽn xảy ra khi nào và ở đâu. Việc phát hiện tắc nghẽn có thể dựa trên một số độ đo khác nhau. Các độ đo thường được sử dụng là tỉ lệ gói số liệu bị loại bỏ do thiếu bộ đệm, chiều dài trung bình của hàng đợi, số gói số liệu phải phát lại do bị hết giờ, thời gian trễ trung bình của gói số liệu khi đi qua mạng v.v. Sự tăng lên của các số đo này nói lên rằng tắc nghẽn đang tăng lên trong mạng.

Bước hai: thông báo trạng thái tắc nghẽn. Nơi phát hiện ra tắc nghẽn cần phải chuyển thông tin về sự tắc nghẽn đến những nơi có thể phản ứng lại. Một cách thực hiện rất đơn giản là nút mạng phát hiện ra tắc nghẽn sẽ gửi gói số liệu đến các nguồn sinh lưu lượng trên mạng, báo tin về sự cố. Tất nhiên, việc này sẽ làm tăng thêm lưu lượng đưa vào mạng đúng lúc lẽ ra phải giảm đi. Người ta cũng đã đề xuất và thực hiện một số cách khác nữa. Chẳng hạn, nút mạng phát hiện ra tắc nghẽn sẽ đánh dấu vào một bit hay một trường định trước của mọi gói số liệu trước khi gói số liệu được nút mạng chuyển tiếp đi, nhằm loan báo cho các nút mạng khác về trạng thái tắc nghẽn. Có thể nêu ra một cách thực hiện khác nữa, đó là làm cho các nút mạng đều đặn gửi đi các gói số liệu thăm dò để biết tình trạng của mạng.

Bước ba: điều chỉnh lại hệ thống để sửa chữa sự cố. Các cơ chế thực hiện phản hồi đều nhằm mục đích là để các máy tính trên mạng có những phản ứng phù hợp nhằm làm giảm tắc nghẽn. Nếu phản ứng xảy ra quá nhanh, lưu lượng trong hệ thống sẽ thẳng giảm mạnh và không hội tụ. Nếu phản ứng quá chậm, việc điều khiển tắc nghẽn có thể không có ý nghĩa thực tế gì nữa. Chính vì vậy, để cơ chế phản hồi có hiệu quả, cần phải sử dụng một số cách tính trung bình [23] [24].

1.3.1.7 Điều khiển luồng trong giao thức TCP

Giao thức TCP ban đầu

Giao thức TCP được xây dựng dựa trên các khái niệm được Cerf và Kahn đưa ra đầu tiên [8]. Đó là giao thức hướng kết nối, kiểu đầu cuối - đầu cuối, tin cậy, được thiết kế phù hợp với kiến trúc phân lớp các giao thức. Giao thức TCP nằm trên giao thức IP, nó hỗ trợ các ứng dụng liên mạng. Giao thức TCP có khả năng gửi và nhận liên tiếp các đơn vị dữ liệu chiều dài có thể thay đổi, được gọi là phân đoạn – segment, mỗi phân đoạn được đóng gói trong một “phong bì” và tạo nên một gói số liệu IP. TCP cung cấp sự truyền thông tin cậy giữa hai tiến trình chạy trên hai máy tính ở các mạng khác nhau nhưng kết nối với nhau. Khi thiết kế TCP người ta giả thiết rằng TCP sẽ nhận được các dịch vụ vận chuyển gói số liệu không tin cậy (không có sự biên nhận cho các gói số liệu) do các giao thức ở các tầng bên dưới cung cấp. Về nguyên tắc giao thức TCP phải có khả năng hoạt động bên trên một miền rất rộng các hệ thống truyền thông, từ các mạng có các đường truyền cố định, tới các mạng chuyển mạch gói và các mạng chuyển mạch cứng.

Trên hình 1.7 là một chồng các giao thức, trong đó giao thức TCP nằm trên giao thức IP.

Các giao thức bậc cao hơn
TCP
IP
Mạng truyền thông

Hình 1.7 Sự phân lớp các giao thức

Kết nối: Để đảm bảo việc vận chuyển tin cậy và thực hiện được cơ chế điều khiển lưu lượng, TCP phải khởi tạo và duy trì một số thông tin trạng thái cho mỗi dòng dữ liệu. Sự kết hợp các thông tin này, bao gồm socket, số thứ tự gói số liệu và kích thước cửa sổ, được gọi là kết nối. Mỗi một kết nối được xác định rõ ràng, duy nhất bởi một cặp socket thuộc hai đầu kết nối đó.

Sự tin cậy: TCP phải khôi phục lại được gói số liệu bị hỏng, bị mất, bị lặp hoặc bị phân phát sai thứ tự do hệ thống truyền thông gây ra. Điều này có thể đạt được bằng cách gán số thứ tự cho mỗi byte được truyền đi và phải có sự biên nhận đã nhận đúng (ACK) từ bên nhận của kết nối TCP. Nếu bên gửi không nhận được biên nhận trong khoảng thời gian hết hạn, nó sẽ phát lại gói số liệu. Tại bên nhận, nó sẽ kiểm tra số thứ tự các gói số liệu nhận được và loại bỏ các gói số liệu không đúng thứ tự hoặc trùng lặp. Các gói số liệu không đúng thứ tự sẽ phải được phát lại, theo thiết kế nguyên thủy của giao thức này.

Điều khiển lưu lượng bằng cửa sổ: TCP cung cấp phương tiện cho bên nhận để nó điều khiển lưu lượng mà bên gửi phát đi. Để thực hiện việc điều khiển này, bên nhận sẽ gửi một “cửa sổ” trong mỗi biên nhận, chỉ ra một miền các số thứ tự mà nó sẽ chấp nhận, tiếp sau số thứ tự của gói số liệu mà nó đã nhận thành công. Đó là số lượng byte mà bên gửi được phép gửi, trước khi nó nhận được một sự cho phép mới. Người ta giả thiết rằng con số này liên quan đến không gian nhớ đệm dữ liệu dành cho kết nối này, mà bên nhận hiện đang có để sử dụng. Việc chỉ ra một cửa sổ lớn sẽ khuyến khích việc phát, nhưng nếu dữ liệu đến nhiều hơn mức có thể nhận,

chúng sẽ bị loại bỏ và sẽ dẫn đến việc phải phát lại quá nhiều. Việc chỉ ra một cửa sổ nhỏ sẽ hạn chế việc phát dữ liệu, có thể làm thời gian trễ của mỗi gói số liệu gửi đi tăng thêm một lượng bằng khoảng thời gian khứ hồi.

Bên gửi của kết nối TCP phải có khả năng nhận và gửi được ít nhất một byte dữ liệu mới từ người sử dụng, ngay cả khi cửa sổ gửi đã bằng không. Ngoài ra, bên nhận cũng phải đều đặn gửi lại cho bên gửi một gói số liệu (“rỗng”) khi cửa sổ nhận của nó đã bằng không. Khoảng thời gian gửi lại đều đặn đó được gọi ý lấy bằng hai phút [38]. Các khả năng trên đảm bảo mỗi bên của kết nối TCP có thể báo cho bên kia của kết nối rằng, nó vẫn “còn sống”, đồng thời có thể báo cho bên kia biết rằng nó chưa sẵn sàng nhận thêm dữ liệu. Khi cửa sổ gửi bằng không, nếu có một gói số liệu đến, nó phải có khả năng gửi biên nhận, cho bên gửi biết số thứ tự gói số liệu tiếp theo mà nó đang chờ và giá trị cửa sổ nhận hiện thời.

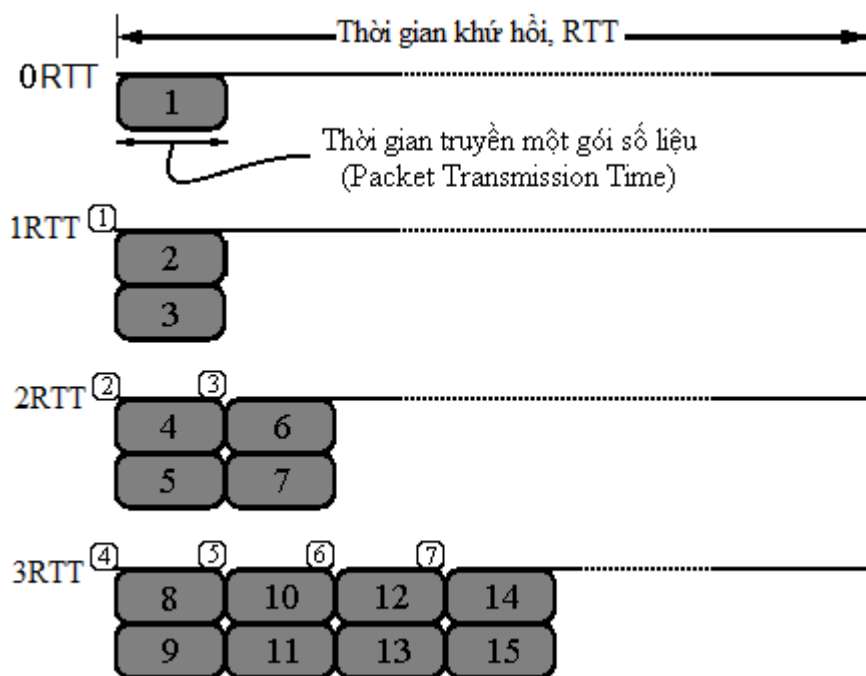
Khởi động chậm

Thuật toán khởi động chậm (SS, Slow Start) khắc phục nguyên nhân gây tắc nghẽn mạng do gửi nhiều gói tin hơn khả năng đáp ứng của mạng, bằng cách tăng dần lượng dữ liệu được vận chuyển cho tới khi đạt tới cân bằng. Thuật toán tóm tắt như sau:

- Bổ sung thêm tham số cửa sổ tắc nghẽn **cwnd** (congestion **w**indow) vào tập trạng thái của mỗi kết nối.
- Khi bắt đầu phát hoặc bắt đầu lại việc phát sau khi có gói số liệu bị mất, đặt cwnd bằng một gói số liệu.
- Mỗi khi nhận được một biên nhận mới, tăng cwnd lên một gói số liệu.
- Khi gửi, gửi số lượng gói số liệu là số nhỏ hơn của kích thước cửa sổ mà hai bên đã thoả thuận và cwnd.

Cửa sổ theo cơ chế khởi động chậm tăng lên theo hàm mũ, nó đạt tới kích thước W sau thời gian bằng $RTT \cdot \log_2 W$, trong đó RTT là thời gian khứ hồi và W là kích thước cửa sổ tính bằng đơn vị gói số liệu. Điều này có nghĩa là cửa sổ mở đủ nhanh để ảnh hưởng không đáng kể đến hiệu suất, ngay cả trên các đường truyền có

tích số giữa dải thông và độ trễ lớn. Theo thuật toán này, người gửi sẽ truyền dữ liệu với tốc độ khi cao nhất là gấp đôi giá trị cực đại có thể của đường truyền. Điều đó cũng có nghĩa là việc mất gói số liệu sẽ là điều không thể tránh khỏi.



Hình 1.8 Sự tăng của cửa sổ trong cơ chế khởi động chậm

Hình 1.8 minh họa sự tăng của cửa sổ trong cơ chế khởi động chậm. Trục thời gian được cắt thành các đoạn có chiều dài bằng khoảng thời gian khứ hồi RTT , các đoạn này được chồng lên nhau theo chiều đứng, hướng trên-dưới ứng với chiều tăng của thời gian. Các gói số liệu được biểu diễn bằng các hình chữ nhật màu xám, bên trong là số thứ tự của gói số liệu. Các hình vuông nhỏ, không tô màu, có đánh số, biểu diễn cho các gói số liệu biên nhận tương ứng. Trên hình vẽ có thể thấy rõ, mỗi khi có một biên nhận trở về, hai gói số liệu sẽ được phát ra: một gói tương ứng với biên nhận (vì mỗi biên nhận cho biết rằng đã có một gói số liệu rời khỏi mạng, do đó cần gửi đi một gói thế chỗ cho nó), còn gói thứ hai là do biên nhận đã làm tăng cửa sổ lên một đơn vị gói số liệu. Hai gói số liệu này được vẽ chồng lên nhau, thể hiện rằng chúng cần được phát đi đồng thời, tuy nhiên, trong thực tế chỉ có thể phát đi lần

lượt từng gói số liệu, do đó trong khi một gói số liệu đang được phát đi, gói số liệu còn lại trong cửa sổ sẽ phải nằm chờ trong hàng đợi.

1.3.1.8 Tính thời gian khứ hồi và thời gian hết hạn gói tin

Tính thời gian khứ hồi theo đặc tả cho giao thức TCP, RFC-793 [38] tính ước lượng thời gian khứ hồi như sau:

$$RTT = (\alpha \cdot RTT_{\text{cũ}}) + ((1 - \alpha) \cdot RTT_{\text{mới}}) \quad (1.2)$$

Trong đó RTT là ước lượng thời gian khứ hồi trung bình, $RTT_{\text{mới}}$ là số đo thời gian khứ hồi từ gói số liệu đã được biên nhận mới nhất và α là hằng số khuếch đại của bộ lọc, giá trị mà người ta gợi ý nên sử dụng là $\alpha=0.9$. Sau khi ước lượng về RTT đã được cập nhật, thì khoảng thời gian hết giờ đối với việc phát lại gói số liệu tiếp theo RTO (**R**ound trip **T**ime **O**ut) được tính theo công thức

$$RTO(k+1) = \beta \cdot RTT(k+1) \quad (1.3)$$

Hằng số β được sử dụng nhằm chống lại sự thăng giáng quá nhanh của thời gian khứ hồi, Trong TCP [38] người ta sử dụng $\beta = 2$.

Nhược điểm của thuật toán trên khi đánh giá tham số lưu lượng trên mạng thông qua giá trị thời gian khứ hồi gói tin là phương pháp này trở thành một mô hình tuyến tính, xử lý một dữ liệu là tham số môi trường không có nhiều đặc điểm tuyến tính. Do vậy thuật toán trên sẽ khó có đáp ứng phù hợp trong các trạng thái không mong muốn, là điều thường xảy ra trong môi trường di động.

1.3.1.9 Rút lui theo hàm mũ.

Đây là cách giải quyết đúng đắn, bởi vì cửa sổ gửi theo cơ chế khởi động chậm tăng lên theo hàm mũ, cho nên cũng cần phải rút lui theo cách này cho đủ nhanh khi đã có dấu hiệu của tắc nghẽn. TCP sẽ đặt lại đồng hồ phát bằng khoảng thời gian rút lui và khoảng đó sẽ được tăng gấp đôi cứ mỗi lần bị hết giờ liên tiếp. Cơ chế rút lui này được giải thích tỉ mỉ trong [20], [30].

1.3.1.10 Tránh tắc nghẽn

Một chiến lược tránh tắc nghẽn (CA, Congestion Avoidance) như đề xuất trong [20],[27],[29] bao gồm: *thứ nhất: mạng phải có khả năng gửi tín hiệu đến cho các thực thể ở đầu cuối của các kết nối (endpoint), báo cho chúng biết là tắc nghẽn đang xảy ra hoặc sắp xảy ra; thứ hai: các endpoint phải có chính sách giảm lưu lượng đưa vào mạng nếu nhận được các tín hiệu báo và tăng thêm lưu lượng đưa vào mạng nếu không nhận được tín hiệu báo này.*

Chính sách của TCP đối với tắc nghẽn:

Đó chính là chính sách tăng theo cấp số cộng, giảm theo cấp số nhân (AIMD), như đã được triển khai thực hiện trong phiên bản TCP trên BSD [18][34]; chính sách đó được giải thích như sau:

- Mỗi khi xảy ra sự kiện hết giờ, đặt giá trị cửa sổ tắc nghẽn $cwnd$ bằng một phần hai giá trị cửa sổ hiện thời. Đó là sự giảm theo cấp số nhân.
- Mỗi khi nhận được một biên nhận cho gói số liệu mới, tăng $cwnd$ thêm một lượng bằng $1/cwnd$, đây là sự tăng theo cấp số cộng. (Trong giao thức TCP, kích thước cửa sổ và kích thước gói số liệu được tính bằng byte, vì thế sự tăng nói trên được chuyển thành $maxseg * maxseg / cwnd$, trong đó $maxseg$ là kích thước gói số liệu cực đại và $cwnd$ là cửa sổ tắc nghẽn, được tính bằng bytes).
- Khi gửi, sẽ gửi đi số gói số liệu bằng số bé hơn trong hai số: kích thước cửa sổ mà bên nhận đã đề nghị và $cwnd$.

Bởi vì mức độ tắc nghẽn tăng lên theo hàm mũ, do khi có tắc nghẽn, các gói tin không đến được bên nhận nhanh chóng, chưa kể bị vứt bỏ khỏi các hàng đợi tại các nút mạng, bên gửi không nhận được biên nhận sau khoảng thời gian chờ đợi, bên gửi sẽ tiếp tục gửi lại các gói tin này, làm cho hiện tượng tắc nghẽn thêm trầm trọng. Nếu tắc nghẽn được phát hiện sớm, thì chỉ cần một vài điều chỉnh nhỏ đối với cửa sổ của người gửi cũng có thể giải quyết được vấn đề; Tuy nhiên, phát hiện tắc nghẽn sớm một cách tin cậy là khó do bản chất luôn thăng giáng mạnh của lưu lượng.

1.4 Điều khiển tắc nghẽn trong mạng có kết nối phức tạp

1.4.1 TCP và ứng dụng đa phương tiện

Ngày nay các ứng dụng phổ biến gồm nhiều loại dữ liệu khác nhau, hay còn gọi là các ứng dụng đa phương tiện, và thường được sử dụng trong môi trường mạng không đồng nhất, nghĩa là một môi trường mạng có sử dụng nhiều chủng loại công nghệ rất khác nhau, có dây hoặc không dây, nhiều kỹ thuật truy nhập và kết nối khác nhau, nhiều loại dịch vụ và ứng dụng với nhu cầu hết sức đa dạng, các thiết bị đầu cuối cũng rất khác nhau về nhiều mặt. Trong môi trường mạng như vậy, một đặc điểm nổi bật có thể nhận thấy là chất lượng dịch vụ do các phần mạng khác nhau cung cấp không hoàn toàn giống nhau, nghĩa là có sự biến đổi về chất lượng dịch vụ của từng phần mạng.

Các ứng dụng đa phương tiện thường có nhu cầu khắt khe về chất lượng dịch vụ. Các ứng dụng đa phương tiện bao gồm các thành phần âm thanh, hình ảnh hay dữ liệu. Các thành phần đa phương tiện như vậy có các yêu cầu rất khác nhau đối với mạng về băng thông, độ trễ cho phép, biến động trễ (jitter) và tổn hao gói ... Ví dụ một luồng video rất nhạy cảm với giới hạn băng thông, còn luồng thoại có thể không tổng hợp được do biến động trễ quá mức. Ngoài ra các dịch vụ như âm thanh và hình ảnh còn có các yêu cầu đảm bảo chất lượng dịch vụ ở một mức độ nhất định. Hỗ trợ chất lượng dịch vụ cho các ứng dụng đa phương tiện là một vấn đề nan giải.

Chất lượng dịch vụ (QoS) thể hiện khả năng của mạng và hệ thống đầu cuối hỗ trợ các yêu cầu ứng dụng của người sử dụng và đảm bảo đáp ứng được yêu cầu đối với các tham số đặc tính và các tham số cảm nhận được của dịch vụ cung cấp cho người sử dụng. Trong mô hình tham chiếu chức năng của mạng, khái niệm chất lượng dịch vụ (QoS) phải được xem xét và quản lý ở mức dịch vụ bất kỳ của hệ thống [4],[5],[47],[49]. Do đó cần phải đặt ra khái niệm QoS ở các mức khác nhau: QoS mức người sử dụng, QoS mức hệ thống (hệ điều hành và các thiết bị I/O) và QoS mức mạng.

Đáp ứng yêu cầu về chất lượng dịch vụ cho các ứng dụng đa phương tiện chưa được thiết kế hiện tại của TCP hỗ trợ một cách thích hợp. TCP được xây dựng trên kiến trúc IP, dựa trên mô hình cung cấp dịch vụ kiểu cố gắng tốt nhất (best effort) và ứng dụng chính của chúng là truyền số liệu (telnet, ftp, e-mail ...) [4]. Một ví dụ cụ thể là các dịch vụ Video Online [2], [3], [6] và [7] đã được phát triển trên nền mạng IP, nhưng các yêu cầu về chất lượng dịch vụ vẫn chưa hoàn toàn được đảm bảo. Mạng chưa có sự đảm bảo các giới hạn về trễ truyền dẫn đầu cuối đến đầu cuối.

Một đặc điểm dễ nhận thấy là các ứng dụng đa phương tiện yêu cầu phải đảm bảo chất lượng dịch vụ thông qua yêu cầu phải đảm bảo một số tham số chất lượng ở mức người sử dụng như chất lượng cảm nhận, độ tin cậy, độ an toàn ... ở một giới hạn nhất định.

1.4.2 Các vấn đề ảnh hưởng QoS trên mạng không dây

Các cơ chế điều khiển việc đáp ứng của các giao thức truyền thông hiện tại dựa trên một số giả thuyết về môi trường của mạng có dây, do vậy cũng tồn tại tính chính xác khi triển khai cho mạng hỗn hợp nói chung và mạng không dây nói riêng. Ví dụ mô hình điều khiển tắc nghẽn của giao thức TCP sẽ yêu cầu thiết bị gửi giảm một nửa tốc độ truyền, mỗi khi có một gói tin bị mất. Mô hình này phản ứng như vậy dựa trên giả thiết rằng, việc mất gói xảy ra khi có tắc nghẽn đầu đó trên đường truyền, và các gói tin trong các bộ đệm bị tràn và bị bỏ đi. Rõ ràng giả thiết đó không hề đúng với môi trường mạng hỗn hợp và môi trường mạng không dây, là nơi các gói tin bị mất có thể do rất nhiều lý do, xuất phát từ đặc điểm của môi trường truyền dẫn[75].

Các cơ chế điều khiển được sử dụng trong các giao thức truyền thông hiện tại, thường sử dụng giá trị cố định một số tham số cho các thuật toán của mình. Do vậy bản chất các giao thức này, đều là một trường hợp cụ thể của một mô hình điều khiển, song chưa có mô hình nào được tổng quát hóa, để từ đó có thể đưa ra cơ chế điều khiển tốt nhất, nhằm đảm bảo chất lượng dịch vụ.

1.4.3 Hiệu suất của giao thức TCP trong mạng có đường truyền không dây

Giao thức TCP hoạt động rất tốt trong các mạng kiểu truyền thống (mạng có dây, các máy tính mạng cố định). Tuy nhiên, các liên mạng đã có sự phát triển bùng nổ và chúng có thể bao gồm cả các mạng không dây. Trong mục này sẽ trình bày tóm tắt các đặc tính cơ bản của đường truyền không dây và ảnh hưởng nặng nề của nó đến hiệu suất của giao thức TCP, để thấy rõ sự cần thiết phải cải thiện hiệu suất của TCP trong các mạng hỗn hợp như vậy.

1.4.4 Ảnh hưởng của đặc tính lỗi đường truyền không dây

Tỉ lệ lỗi bit, BER của đường truyền không dây thường cao hơn rất nhiều so với BER của các đường truyền có dây, khi cùng tốc độ truyền tin. Trong đường truyền không dây, BER nằm trong khoảng từ 10^{-2} đến 10^{-6} ; trong đường truyền dùng cáp đồng, BER nằm trong khoảng từ 10^{-5} đến 10^{-9} ; còn trong cáp quang, BER nằm trong khoảng từ 10^{-9} đến 10^{-12} , thậm chí còn thấp hơn thế nữa. Tỉ lệ lỗi bit cao trên các đường truyền không dây gây ra do sự kết hợp của nhiều yếu tố: hiện tượng fading, do địa hình, do các yếu tố môi trường và do can nhiễu của các tín hiệu khác được truyền trong không trung. Các lỗi đường truyền này thường gây ra sự bùng nổ số gói số liệu bị mất và dẫn đến sự gián đoạn của kết nối. Việc mất gói số liệu gây ra sự phát lại hoặc hết giờ và điều đó lại kích hoạt cơ chế khởi động chậm, làm giảm mạnh tốc độ truyền, dẫn đến làm giảm trầm trọng thông lượng.

Với đặc tính tỉ lệ lỗi bit cao của đường truyền không dây, người ta phải chọn giá trị đơn vị truyền thông tối đa, MTU (Maximum Transmission Unit) nhỏ hơn nhiều so với đơn vị dữ liệu trong các mạng có dây. Kích thước nhỏ của đơn vị dữ liệu truyền này cũng dẫn đến việc phải dùng các gói số liệu nhỏ trong các phần mạng có đường truyền có dây, mặc dù trên các đường truyền có dây có thể dùng các gói số liệu kích thước lớn hơn. Hệ quả là các chi phí cho xử lý gói số liệu (đóng gói dữ liệu, tách dữ liệu...) ở các nút trên đường truyền tăng lên và làm giảm thông lượng.

Các kết quả nghiên cứu ảnh hưởng lỗi đường truyền, cũng như thời gian tạm ngừng kết nối do sự chuyển cuộc gọi lên hiệu suất của giao thức TCP, được kết hợp trình bày ở phần dưới đây.

1.4.5 Ảnh hưởng của sự gián đoạn kết nối thường xuyên

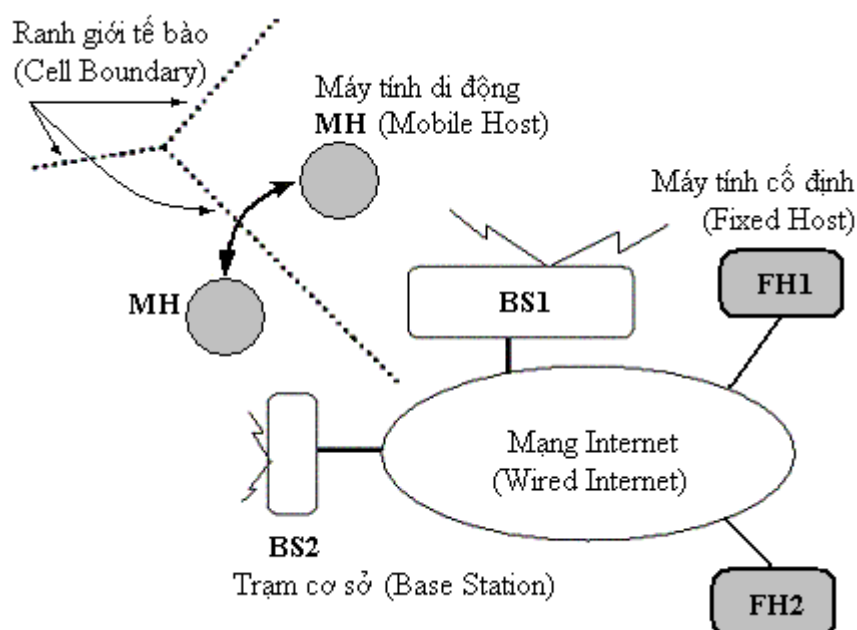
Có rất nhiều tình huống trong đó việc chuyển cuộc gọi mềm từ điểm truy cập này đến điểm truy cập khác không thể thực hiện được. Thí dụ, trong một mạng sử dụng giao thức IP di động, khi người dùng di động đi từ vùng do một đại lý khách vắng lai, một trạm gốc-BS (Base Station) nào đó quản lý sang một vùng do một BS khác quản lý, các gói số liệu gửi cho nó có thể vẫn đang đi trên đường đi đến BS cũ khi nó đã đến vùng BS mới quản lý rồi. Khi đó có thể xảy ra tình huống BS cũ không thể chuyển tiếp các gói số liệu này tới BS mới do việc chuyển cuộc gọi chưa được thực hiện, hoặc thậm chí BS cũ không còn đủ bộ đệm để chứa tạm các gói số liệu này nếu kết nối bị đứt đoạn quá lâu (việc chuyển cuộc gọi có thể xảy ra trong khoảng vài chục tới vài trăm mili giây trong hầu hết các hệ thống truyền thông di động [17]). Bản thân sự di động cũng có thể làm mất các gói số liệu khi người dùng di động ra khỏi vùng thu/phát tin cậy của các trạm gốc, nhất là trong các mạng di động mà các tế bào ít hoặc không gối lên nhau.

Tác động đồng thời của tỉ lệ lỗi bit cao và sự kết nối hay bị đứt đoạn của các đường truyền không dây lên hiệu suất của giao thức TCP đã được nhiều người nghiên cứu, trong đó phải kể đến nghiên cứu của R. Yavatkar và N. Bhagwat trong [19]. Một mạng để làm thí nghiệm được xây dựng, như được mô tả trên Hình 1.9. Mạng này gồm hai phần; phần thứ nhất chính là mạng Internet trong đó có các máy tính cố định(FH); phần thứ hai là mạng không dây với các trạm làm việc di động, MH. Trạm gốc(BS), còn được gọi là trạm hỗ trợ di động nối hai mạng này với nhau, giống như các gateway cho phép hai mạng có dây và không dây có thể trao đổi các gói số liệu với nhau.

Trong thí nghiệm, người ta đã truyền một tệp kích thước 100KB giữa một máy tính di động và một máy tính cố định, qua một kết nối TCP thông thường. Thời gian

truyền trung bình đo được, được trình bày trong bảng 1.1, các con số ghi trong ngoặc vuông là các giới hạn trên và dưới của miền giá trị đo được. Trong bảng 1.2 chuyển đổi các kết quả từ thời gian truyền trung bình ở bảng 1.1 thành tốc độ truyền trung bình. Cách chuyển đổi như sau:

$$\text{Tốc độ truyền trung bình (Kbps)} = 100\text{KB} * 8 / \text{Thời gian truyền trung bình(s)}.$$



Hình 1.9 Mạng để nghiên cứu thực nghiệm về liên mạng di động;
MH: Trạm di động; FH: Trạm cố định; BS: Trạm gốc.

Bảng 1.1 Thời gian truyền trung bình (s)

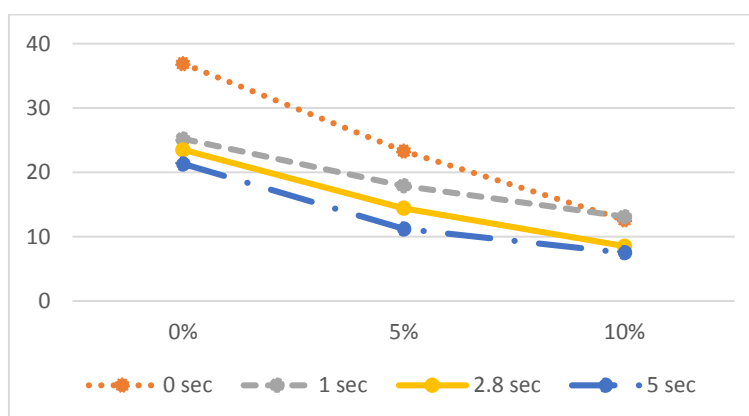
Thời gian dừng do chuyển cuộc gọi	Tỉ lệ gói số liệu bị mất (%)		
	0 %	5 %	10 %
0 s	21.7 [19.3, 24.1]	34.4 [30.6, 38.2]	63.3 [53.0, 73.6]
1 s	31.7 [27.6, 35.9]	44.6 [40.9, 48.3]	56.6 [50.0, 62.7]
2.8 s	32.6 [29.2, 36.0]	52.1 [45.6, 58.6]	88.7 [77.6, 99.7]
5 s	36.7 [34.0, 39.3]	69.8 [60.1, 79.6]	99.9 [86.6, 113.1]

Từ kết quả được trình bày trên hình 1.10, chúng ta có thể nhận thấy rõ:

- Ngay cả khi đường truyền không gây lỗi (tỉ lệ mất gói số liệu bằng 0%), việc dừng kết nối do chuyển cuộc gọi, cũng làm giảm tốc độ truyền rất nhiều. Thí dụ, thời gian dừng kết nối 1s làm giảm tốc độ truyền từ 36.9Kbit/s xuống còn 25.5Kbit/s (69%). Thời gian dừng kết nối càng dài, tốc độ truyền càng giảm.
- Ngay cả khi không có sự tạm dừng kết nối do chuyển cuộc gọi (đường trên cùng của đồ thị, hình 1.10), tỉ lệ mất gói số liệu tăng lên làm tốc độ truyền giảm đi rất mạnh. Cụ thể là: tỉ lệ mất gói số liệu tăng từ 0% lên 5% làm tốc độ truyền giảm xuống còn 63% (23.3/36.9); nếu tỉ lệ mất gói số liệu tăng lên đến 10%, tốc độ truyền giảm xuống chỉ còn 34% (12.6/36.9).

Bảng 2.2 Tốc độ truyền trung bình (Kbit/s)

Thời gian dừng do chuyển cuộc gọi	Tỉ lệ gói số liệu bị mất (%)		
	0 %	5 %	10 %
0 sec	36.9	23.3	12.6
1 sec	25.2	17.9	13.1
2.8 sec	23.5	14.4	8.5
5 sec	21.3	11.2	7.5



Hình 1.10 Ảnh hưởng của tỉ lệ lỗi bit (BER) cao và thời gian chuyển cuộc gọi đến tốc độ truyền của TCP (Kbit/s)

1.5 Đặt vấn đề nghiên cứu

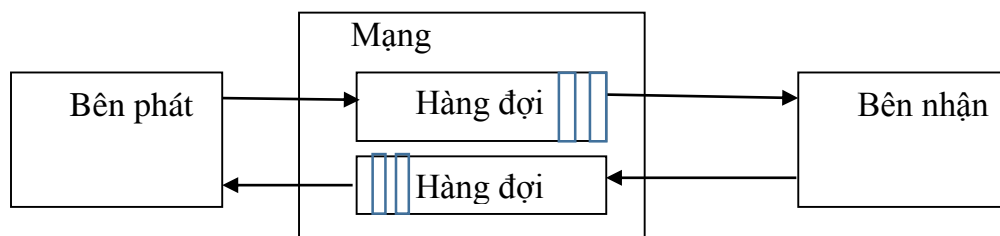
Các cơ chế điều khiển được sử dụng trong các giao thức truyền thông hiện tại, như ở trên đã trình bày, thường sử dụng giá trị cố định một số tham số cho các thuật toán của mình. Do vậy bản chất các giao thức này, đều là một trường hợp cụ thể của một mô hình điều khiển, xong chưa có mô hình nào được tổng quát hóa, để từ đó có thể đưa ra cơ chế điều khiển tốt nhất, nhằm đảm bảo chất lượng dịch vụ.

Mục tiêu nghiên cứu của luận án là xây dựng một mô hình điều khiển thông minh, nhằm phản ứng linh hoạt hơn với các biến cố của đường truyền, thông qua việc lưu giữ các trạng thái trước đó của môi trường và của chính hệ thống. Việc ứng dụng mô hình trên vào các công việc cụ thể để tính toán các thông số của các giao thức truyền thông, như tốc độ truyền, thời gian lưu thông của các gói tin, quản lý bộ nhớ đệm cũng sẽ được đề cập cụ thể. Mô hình được đề xuất cũng có thể dùng để mô tả các cơ chế hiện tại sử dụng trong các giao thức truyền thông hiện nay, như trường hợp suy biến của một số tham số nào đó.

Minh họa cho nhận định này, ta thấy điều khiển kết nối sẽ xác định một loạt thông số để quyết định một kết nối mới được chấp nhận hay bị từ chối. Quyết định này dựa trên việc xác định các nguồn lực cần thiết và kiểm tra khả năng đáp ứng chất lượng đối với các dịch vụ đã được cam kết về chất lượng. Trong một ví dụ khác, như đối với mô hình điều khiển phân luồng tại các điểm mạng, thời gian đến của các gói tin được theo dõi, độ trễ của mỗi gói tin được tính toán, thời gian được gán lên mỗi gói tin và dựa trên đây, bộ phận phân luồng sẽ quyết định, gói nào trong hàng chờ nào sẽ được ưu tiên phục vụ trong thời gian truyền dẫn tiếp theo. Các ví dụ khác có thể minh họa bằng cơ chế điều khiển tắc nghẽn. Mô hình mạng tương ứng được thể hiện minh họa trong Hình 1.11

Mạng máy tính thường được mô hình hóa với hai hàng đợi, một hàng đợi cho các gói tin được truyền từ bên gửi tới bên nhận, một cho các gói tin phản hồi ACK từ phía nhận. Thường sẽ có hai tham số cần xác định, đó là RTT và tỷ lệ mất gói tin. Phía nhận sẽ đo RTT cũng như tỷ lệ mất gói tin và thông báo lại cho bên gửi thông

qua các gói tin ACK. Khi các hàng đợi bị đầy, các gói tin sẽ bị bỏ bớt. Các hàm quyết định hoạt động tại phía gửi. Có hai quyết định thường thấy là tăng hoặc giảm tốc độ gửi tin.



Hình 1.11. Mô hình mạng điều khiển tắc nghẽn và lưu lượng giữa hai đầu cuối

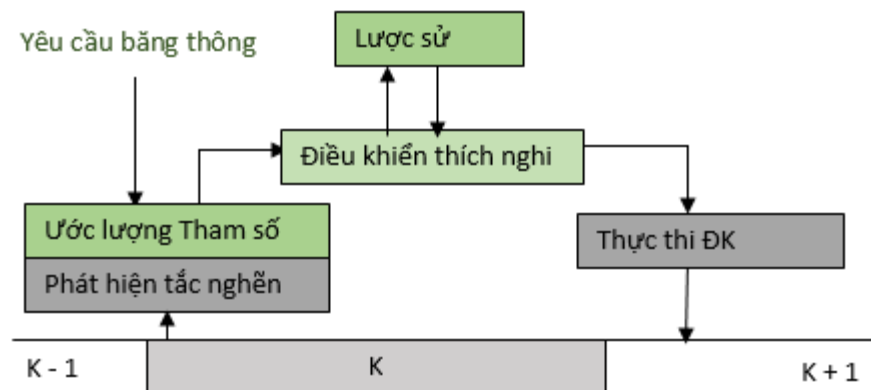
Với các ví dụ trên, chúng ta thấy các cơ chế điều khiển đảm bảo QoS có ba quá trình: đo đạc, xử lý và ra quyết định.

Luận án đề xuất mô hình hệ thống điều khiển thích nghi, ở đó các thông số trạng thái tại trạng thái K sẽ được tính toán, dựa trên những hiểu biết mà hệ thống đã học được ở các trạng thái trước đó. Những tính toán này sẽ ảnh hưởng đến trạng thái hệ thống tại thời điểm tiếp theo. Mô hình thích nghi có thể được điều chỉnh thông qua các tham số đầu vào khác, tùy thuộc các tham số mà chúng ta quan tâm.

Mô hình thích nghi sẽ có các thành phần nhằm đo đạc các tham số trạng thái tại thời điểm xem xét và có các chức năng sau:

Lược sử

Là nơi lưu giữ thông số của N trạng thái quá khứ còn có ảnh hưởng đến hiện tại. Đối với môi trường không dây, giá trị các tham số môi trường biến thiên nhanh và nếu hệ thống điều khiển phản ứng ngay tức khắc với mỗi thay đổi theo cách thông thường, có thể sẽ thiếu chính xác. Ví dụ khi thiết bị di động đi qua một vật chắn, băng thông có thể bị giảm đi đáng kể trong khoảng thời gian tương ứng với thời gian đi qua vật chắn. Nhưng sau khi thoát ra khỏi vật chắn đó, tốc độ gửi tin không nên giảm đi mà cần tính toán dựa trên tốc độ trước khi thiết bị đi vào vùng bị cản sóng. Mô hình cần cho phép để điều chỉnh giá trị N để phù hợp các môi trường khác nhau.



Hình 1.12 Kiến trúc nguyên lý điều khiển thích nghi

Ước lượng Tham số

Là chức năng xử lý thông tin về các tham số môi trường, tham khảo các giá trị đã lưu trong ký ức của hệ thống để tiệm cận nhanh nhất đến giá trị các trạng thái tức thời của hệ thống. Mô hình thích nghi sẽ dựa trên thông tin thu được từ trạng thái của môi trường và các giá trị thông số đầu vào mong muốn của người sử dụng để điều chỉnh mức độ chính xác của quá trình tính toán.

Trong một hệ thống, có thể triển khai nhiều bộ phận thu nhận thông số đầu vào. Ví dụ để phát hiện băng thông khả dụng, phát hiện tắc nghẽn, phát hiện gói đến chậm, hoặc để phát hiện lỗi mất gói do đặc tính môi trường.

Đối với việc tính toán nhu cầu sử dụng băng thông, sẽ tính số lượng bits được gửi đến nơi trong một khoảng thời gian, vì vậy cần phải đo được thời gian một gói tin bắt đầu và kết thúc gửi, cũng như thời gian cần thiết để gói tin đó đi đến đích.

Tính toán tỷ lệ lỗi gói tin xảy ra trên mạng không dây sẽ phân biệt các lỗi mất gói do tắc nghẽn trên phần mạng có dây dẫn. Nhờ đó phản ứng của hệ thống sẽ phù hợp hơn.

Thu nhận thông số độ dài của hàng đợi sẽ giúp đầu đối diện dự đoán là kết nối sẽ còn tiếp tục gửi bao nhiêu gói tin. Đầu đối diện sẽ nhanh chóng nhận biết tình trạng mất gói tin nhanh chóng hơn.

Điều khiển thích nghi:

Thành phần này sẽ ra quyết định tăng hay giảm giá trị tham số đang theo dõi trên hệ thống. Quyết định này đưa ra nhằm giúp các tham số của hệ thống nằm trong ranh giới cho phép, ví dụ tốc độ luồng tin $[R_{\min}, R_{\max}]$. Nhờ đó chất lượng dịch vụ được đảm bảo. Thành phần này cũng giúp cơ chế phản ứng không bị thái quá làm tổn nguồn lực và ảnh hưởng đến các cơ chế điều khiển khác đang hoạt động chung trong cùng môi trường chia sẻ. Việc ra quyết định sẽ phụ thuộc vào các giá trị đầu vào đo được từ kênh truyền, đồng thời tham khảo các giá trị được lưu trong *Lược sử*.

Qui trình điều khiển hiện nay sử dụng phương pháp mô hình chất lưu

$$\frac{dx(t)}{dt} = f(x(t), e(t))$$
, trong đó $x(t)$ là trạng thái của môi trường. $e(t)$ là kết quả tính toán một thông số nào đấy, hoặc sử dụng phương pháp mô hình rời rạc $x_{(k+1)} = x_{(k)} + f(x)$

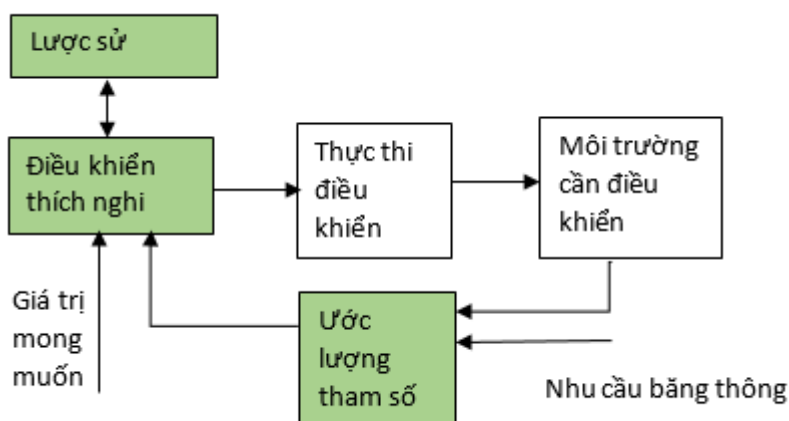
Với việc đưa vào thông tin mà mô hình điều khiển thu thập được từ N trạng thái trước đó, mô hình rời rạc sẽ có dạng: $x_{(k+1)} = x_{(k)} + f[x_{(k-1)}, \dots, x_{(k-N+1)}]$

Thực thi điều khiển

Đây là bộ phận thi hành các quyết định đã được bộ phận **điều khiển thích nghi** đưa ra, trong một giới hạn thời gian nào đó. Việc thực thi này sẽ thể hiện là các phản ứng của hệ thống đối với môi trường.

Có thể nói các đặc tính của mô hình thích nghi mà chúng ta nêu ra ở đây là mô hình tổng quát cho các cơ chế điều khiển đảm bảo QoS trước đây. Trong đó các thông số đầu vào được thu thập từ việc xác định các giá trị thực tế của hệ thống. Sau khi so sánh các giá trị này với các giá trị mong muốn, bộ phận điều khiển sẽ dùng các thuật toán để quyết định xem hệ thống cần phải làm gì.

Mô hình điều khiển thích nghi có thể được thể hiện như Hình 1.13, ở đó các thông số đầu vào và các điều chỉnh đối với đầu ra phụ thuộc vào mỗi loại thông số môi trường cụ thể. Ví dụ, với mục tiêu kiểm soát tỷ lệ lỗi gói, các thông số đầu vào là tỷ lệ lỗi gói thực tế và mong muốn, thông số đầu ra là giới hạn kích cỡ các hàng đợi cần điều chỉnh, nhằm không để xảy ra việc vứt bỏ các gói tin khi bị tràn bộ đệm.



Hình 1.13 Mô hình điều khiển tự thích nghi

Để quản lý bộ nhớ đệm, ta sẽ có $Z_k = Z_{k-1} + u(t)$, với $u(t)$ là hàm của các biến. Với các phương pháp quản lý bộ đệm hiện nay như RED ta sẽ có các phương trình tương ứng:

$$Z_k = Z_{k-1} + \omega (q_t - Z_{k-1})$$

$$Z_k = Z_{k-1} + u(\Delta)$$

Z_k là trạng thái thứ k , Z_{k-1} là trạng thái thứ $k-1$, q_t là kích thước hàng đợi, ω là trọng số, $u(\Delta)$ là bước thay đổi.

Cơ chế điều chỉnh áp dụng kết quả nghiên cứu phương pháp quản lý vùng đệm theo RED trong [1] [61] để hạn chế khả năng tràn bộ đệm.

Các biến trạng thái có thể được tính mỗi khi có một gói tin được nhận hoặc khoảng thời gian giữa các gói tin. Quyết định đưa ra cho việc xử lý gói tin đang đến dựa trên việc so sánh biến trạng thái với các giá trị giới hạn.

Ví dụ: Với cơ chế quản lý bộ nhớ đệm RED duy trì 2 giá trị giới hạn q_{\min} , q_{\max} . Khi một gói tin đến, các phép toán sau được thực hiện

$$p_b = p_{\max} \cdot \frac{Z_k - q_{\min}}{q_{\max} - q_{\min}} \quad \text{và} \quad p_a = \frac{p_b}{1 - n \cdot p_b}$$

Với p_a là xác suất bị đánh dấu / bị loại bỏ, n là số đếm các gói tin kể từ lần đánh dấu / loại bỏ gần nhất. Và quyết định đưa ra với mỗi kích thước bộ đệm q_{ave}

- Nếu $q_{ave} < q_{\min} \rightarrow$ Không làm gì
- Nếu $q_{\min} \leq q_{ave} \leq q_{\max} \rightarrow$ đánh dấu/ loại bỏ theo xác suất p_a .
- Nếu $q_{ave} \geq q_{\max} \rightarrow$ đánh dấu/ loại bỏ mọi gói tin.

1.6 Kết luận chương I

Trong chương I đã giới thiệu mô hình tham chiếu TCP/IP, là chuẩn để các mạng máy tính khác nhau có thể truyền thông tin với nhau, và phân tích đặc điểm thiết kế của giao thức truyền thông TCP, một giao thức hướng kết nối, đảm bảo vận chuyển dòng byte sinh ra từ máy (nguồn) tới một máy tùy ý khác (điểm đến) trong liên mạng không có lỗi. Với giả định ban đầu nguyên nhân gây mất gói tin trên mạng là do tắc nghẽn, nên TCP đã ứng dụng lý thuyết kiểm soát lưu lượng, chống tắc nghẽn vào thiết kế của mình.

Chương I đặt mục tiêu nghiên cứu của luận án là xây dựng một mô hình điều khiển thông minh, nhằm phản ứng linh hoạt hơn với các biến cố của đường truyền, thông qua việc lưu giữ các trạng thái trước đó của môi trường và của chính hệ thống. Ứng dụng mô hình để tính toán chính xác hơn các thông số của các giao thức truyền thông, như tốc độ truyền, thời gian lưu thông của các gói tin, quản lý bộ nhớ đệm sẽ giúp giao thức truyền thông phản ứng thích hợp hơn với môi trường.

Trong chương tiếp theo sẽ trình bày và phân tích nhu cầu đảm bảo chất lượng dịch vụ (QoS) cho các ứng dụng gồm nhiều loại dữ liệu, các ứng dụng đa phương tiện, trong môi trường mạng có kết nối phức tạp, có dây hoặc không dây, với nhiều

kỹ thuật truy nhập và kết nối khác nhau, phục vụ nhiều loại dịch vụ và ứng dụng với nhu cầu hết sức đa dạng, đồng thời phân tích hạn chế của giao thức TCP, là đạt hiệu năng thấp khi môi trường không dây xảy ra nhiều lỗi mất gói tin. Từ đó luận án tổng hợp các phương án, đề xuất hiện có, nhằm mục đích cải tiến giao thức TCP để đạt hiệu suất cao hơn trong môi trường không dây và mạng hỗn hợp.

CHƯƠNG 2. CÁC GIẢI PHÁP ĐIỀU KHIỂN TẮC NGHẼN TRONG MẠNG CÓ KẾT NỐI PHỨC TẠP

Trong mạng có đường truyền không dây, sự mất gói số liệu chủ yếu là do lỗi đường truyền hoặc do sự chuyển cuộc gọi gây trễ quá dài, chứ không phải là do tắc nghẽn mạng. Trong chương này, sẽ nghiên cứu, xem xét nhu cầu đảm bảo chất lượng dịch vụ và hiệu năng của TCP trong môi trường không dây, và trình bày các hướng tiếp cận chính nhằm cải thiện hiệu năng của TCP trong môi trường không dây.

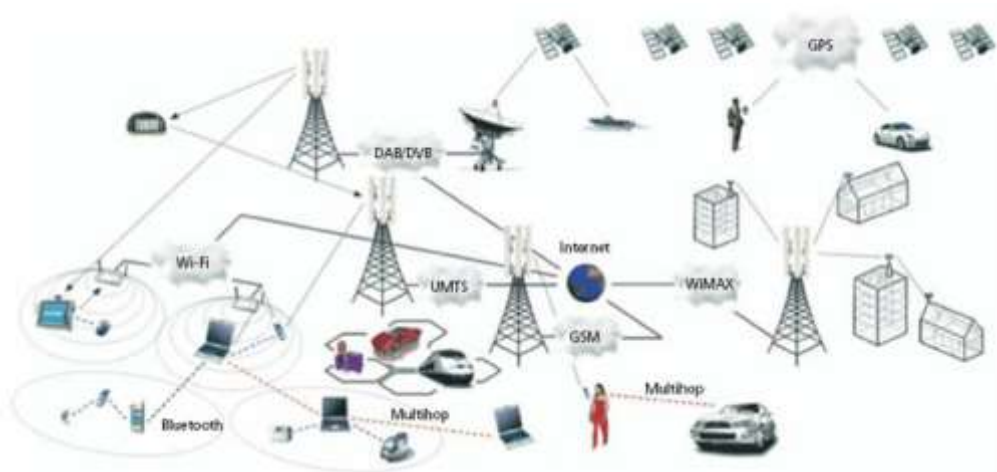
2.1 Cấu trúc mạng có kết nối phức tạp – mạng có kết nối không dây

Trong thực tế, việc đáp ứng khả năng kết nối của các loại thiết bị khác nhau hiện nay của con người, tạo nên một hệ thống mạng lưới phức hợp, bao gồm các máy tính và thiết bị khác sử dụng nhiều hệ điều hành và giao thức truyền thông khác nhau. Ví dụ về mạng phức hợp như mạng nội bộ kết nối các máy tính chạy hệ điều hành Windows và Linux cùng máy tính chạy hệ điều hành Apple Machintosh trên nền tảng kết nối cục bộ và không dây.

Sự phức hợp cũng được thể hiện trong các mạng không dây sử dụng các công nghệ truy cập khác nhau. Ví dụ, mạng không dây cung cấp dịch vụ thông qua công nghệ không dây cho mạng nội bộ và có thể duy trì dịch vụ qua các tế bào của mạng viễn thông.

Xu hướng các thiết bị hội tụ về công nghệ IP và tính chất đa dạng của các mạng máy tính, viễn thông đang làm công việc đảm bảo chất lượng dịch vụ cho các kết nối, cùng việc điều khiển chống tắc nghẽn trong mạng có kết nối phức tạp trở thành một thử thách lớn, và vẫn đang cần tiếp tục nghiên cứu [62].

Hình 2.1 minh họa cho nhu cầu và các tình huống sử dụng trong thực tế, trong một mạng có kết nối phức tạp, bao gồm nhiều loại thiết bị, truyền dẫn nhiều loại dữ liệu, sử dụng nhiều công nghệ truy cập khác nhau.



Hình 2.1 Ví dụ mô hình mạng có kết nối phức tạp

2.2 Các kỹ thuật nhằm cải thiện hiệu năng TCP cho mạng không dây.

Luận án phân loại các phương pháp tiếp cận thành hai tập hợp các giải pháp. Tập hợp thứ nhất là các giải pháp che giấu phần mạng hay làm mất gói số liệu do lỗi đường truyền, sao cho bên gửi chỉ phát hiện được những sự mất gói số liệu do tắc nghẽn. Tập hợp thứ hai các giải pháp cải tiến TCP bằng các cơ chế thông báo rõ ràng về nguyên nhân mất gói số liệu, giúp cho TCP có thể phân biệt được các kiểu mất gói số liệu khác nhau. Một số đề xuất yêu cầu thay đổi thiết kế mô hình truyền thống, song mức độ hiệu quả chưa được chứng minh rõ ràng, do việc thay thế ngay lập tức toàn bộ hệ thống thiết bị, mạng viễn thông yêu cầu đầu tư rất lớn và không khả thi.

Với mỗi tập hợp, các giải pháp lại có thể được phân loại thành các giải pháp đơn tầng, và giải pháp liên tầng. Các giải pháp liên tầng dựa trên sự trao đổi tương tác giữa hai tầng trong kiến trúc mô hình tham chiếu OSI. Các giải pháp liên tầng được nghiên cứu, thường cần sự tương tác giữa TCP và một tầng khác, hoặc là tầng mạng, hoặc là tầng liên kết dữ liệu. Những đề xuất giải pháp liên tầng thường có động lực dựa trên thực tế là cung cấp thông tin của các tầng dưới cho tầng trên, có thể cải thiện hiệu năng tầng trên tốt hơn. Những đề xuất giải pháp đơn tầng tuân thủ đặc điểm độc lập giữa các tầng trong mô hình tham chiếu OSI. Do vậy, tùy thuộc vào loại tầng OSI mà mỗi đề xuất can thiệp, các đề xuất đơn tầng lại có thể chia ra thành các giải pháp tầng giao vận, tầng mạng hay giải pháp tầng liên kết dữ liệu.

2.2.1 Che giấu phần mạng hay làm mất gói số liệu do lỗi đường truyền

Phương pháp này che giấu sự mất gói số liệu không phải do tắc nghẽn, không cho bên gửi của kết nối TCP phát hiện ra. Theo cách tiếp cận này, vì vấn đề xảy ra có tính cục bộ, cho nên cần giải quyết nó theo cách cục bộ, tầng giao vận không cần phải nhận thấy các đặc điểm của các đường truyền riêng lẻ. Các giao thức theo cách tiếp cận này cố gắng làm cho các đường truyền gây nhiều lỗi thể hiện ra như thể các đường truyền có chất lượng cao, nhưng có dải thông sử dụng được nhỏ hơn. Kết quả là bên gửi của kết nối TCP hầu như chỉ nhận thấy được sự mất gói số liệu do tắc nghẽn mạng.

2.2.1.1 Các giải pháp ở tầng liên kết dữ liệu

Các kỹ thuật điều khiển lỗi phổ biến nhất ở tầng liên kết dữ liệu đều dựa trên hai chức năng sau [16][38]:

- Phát hiện lỗi / Khắc phục lỗi: người ta bổ sung thêm một lượng thông tin dư thừa vào dữ liệu, theo một số thuật toán nhất định, sao cho bên nhận có thể phát hiện hoặc sửa lại cho đúng dữ liệu mà nó nhận được. Kỹ thuật phát hiện lỗi yêu cầu bổ sung ít thông tin dư thừa hơn kỹ thuật khắc phục lỗi.
- Yêu cầu phát lại tự động (Automatic Repeat reQuest - ARQ): khi phát hiện ra một lỗi trong dãy dữ liệu đến, bên nhận yêu cầu bên gửi gửi lại [28],[37],[39],[40],[41].

Bằng cách kết hợp kỹ thuật phát hiện lỗi và kỹ thuật yêu cầu phát lại tự động trong giao thức tầng liên kết dữ liệu, người ta có thể biến tầng liên kết dữ liệu không tin cậy thành một tầng tin cậy. Cơ chế ARQ cơ bản chỉ có khả năng phát hiện lỗi và phát lại. Trong các cơ chế phát lại có sử dụng ARQ, có ba cơ chế phổ biến nhất là:

- Dừng và chờ (Stop-and-wait ARQ),
- Phát lại từ gói số liệu thứ N (Go-back-N) và
- Phát lại có chọn lọc (Selective-repeat) [42],[43].

Trong các cơ chế đó, cơ chế phát lại có chọn lọc cho hiệu quả cao nhất, nhưng cũng là cơ chế phức tạp nhất khi áp dụng.

ARQ cũng có thể được áp dụng riêng trong một đoạn mạng gặp nhiều lỗi, như đoạn mạng không dây, giữa thiết bị di động và trạm gốc, để che lỗi mất gói trước các giao thức ở tầng giao vận, đồng thời cải thiện thời gian phản ứng khi lỗi xảy ra. [28] đề xuất phát lại trong nội bộ kênh truyền không dây và nghiên cứu ảnh hưởng lên tham số RTT của giao thức TCP, là một giải pháp như vậy.

Việc sử dụng ARQ có hiệu quả khi sự mất mát gói số liệu là không thường xuyên và thời gian trễ truyền là không đáng kể. Cơ chế này chỉ làm tốn một dải thông phụ thêm khi có gói số liệu phải phát lại. Tuy nhiên, ARQ có thể gây trở ngại cho các cơ chế của TCP, trong khi tầng liên kết dữ liệu đang cố gắng phát lại, bên gửi của kết nối TCP có thể bị hết giờ, dẫn đến kích hoạt giai đoạn khởi động chậm, SS. Giải pháp mà người ta đã đề xuất để giải quyết vấn đề này là sử dụng giao thức ARQ có nhận biết TCP (TCP-aware ARQ protocol); tầng liên kết dữ liệu sẽ chặn các biên nhận lặp, không cho chúng đến được bên gửi. Nếu như tầng liên kết dữ liệu không thể gửi lại các gói số liệu thì bên gửi sẽ bị hết giờ và chính nó sẽ phát lại gói số liệu. Giải pháp này chỉ có thể áp dụng được khi đường truyền gây nhiều lỗi là chằng cuối cùng để đến điểm đến. Ngược lại, nếu sau đường truyền gây nhiều lỗi lại có các đường truyền và bộ định tuyến nữa, những sự mất mát do tắc nghẽn mạng sẽ bị che đi, cơ chế điều khiển tắc nghẽn mất tác dụng, điều này cần phải tránh.

Để giảm ảnh hưởng xấu của tỉ lệ lỗi bit cao của đường truyền không dây đến việc vận chuyển dữ liệu, các thuật toán mã hoá sửa lỗi ở phía trước - FEC (Forward Error Correcting) đã được nghiên cứu và sử dụng rộng rãi. Thuật toán FEC chỉ ra cách bổ sung thông tin dư thừa vào dữ liệu gửi, để cho bên nhận có thể khôi phục lại phần bị hỏng của gói số liệu. Nhược điểm của thuật toán này là thông tin dư thừa bổ sung thêm vào gói số liệu gửi đi sẽ không cần sử dụng đến trong những lúc trạng thái đường truyền là tốt, gây ra sự lãng phí dải thông. Ngoài ra, việc tính toán đối với thông tin bổ sung này cũng cần đến thời gian xử lý của CPU và bộ nhớ. Tuy nhiên, trong những tình huống kênh truyền rất dài lại có tỉ lệ lỗi cao, như đường truyền vệ tinh chẳng hạn, ưu điểm của FEC rất xứng với cái giá mà chúng ta phải trả. Kỹ thuật FEC còn có một ưu điểm nổi bật nữa là nó không cản trở các cơ chế của giao thức

TCP. Với các ưu và nhược điểm trên, FEC thường được sử dụng trong truyền thông vệ tinh và truyền thông vũ trụ.

Để kết hợp ưu điểm của ARQ và FEC, [44] đề xuất phương án lai cải tiến kênh truyền, kết hợp kỹ thuật mã sửa sai và ARQ nhằm giảm tải tính toán, đồng thời đáp ứng với tình huống mất gói tin khi không phục hồi được.

Tóm lại, ưu điểm chính của việc khắc phục lỗi ở tầng liên kết dữ liệu là nó thích hợp một cách tự nhiên với cấu trúc phân lớp của các giao thức mạng. Các giao thức tầng liên kết dữ liệu hoạt động độc lập với các giao thức của các tầng bên trên và nó không phải duy trì bất cứ một thông tin trạng thái kết nối nào.

Điều chủ yếu mà chúng ta phải quan tâm, chính là khả năng nó có ảnh hưởng có hại đối với các giao thức tầng giao vận, thí dụ giao thức TCP.

2.2.1.2 Các giải pháp ở tầng giao vận

Các giải pháp này cố gắng nâng cao chất lượng đường truyền bằng cách phát lại các gói số liệu ở mức giao thức TCP chứ không phải ở tầng liên kết dữ liệu. Người ta sẽ tạo ra một tiến trình TCP đặc biệt trên bộ định tuyến ở đầu vào của đường truyền gây nhiều lỗi gọi là agent, tiến trình này sẽ giữ bản sao của mọi gói số liệu từ bộ định tuyến đi vào đường truyền này. Nếu nhận được biên nhận của một gói số liệu, tiến trình đó sẽ loại bỏ bản sao của gói số liệu đó khỏi bộ nhớ đệm của nó, còn nếu biết rằng gói số liệu bị mất, tiến trình sẽ thay mặt cho bên gửi phát lại gói số liệu. Kỹ thuật này đã được người ta đề xuất sử dụng cho các mạng không dây trên mặt đất, trong đó thời gian trễ truyền không quan trọng đến mức phải dùng đến kỹ thuật FEC. Agent TCP được đặt trong các trạm gốc, chúng nằm ở đường vào mạng không dây. Cho đến nay có một số cơ chế sử dụng agent, điển hình là TCP gián tiếp (I-TCP - Indirect TCP)[2], [3], [4], [9], [10], [36], Snoop TCP [15], [22], Split TCP[63], TCP-ADW [72]...và một số đề xuất điều chỉnh các tham số TCP cho phù hợp như điều chỉnh kích thước động cho bộ đệm [53], [55],[71] điều chỉnh cơ chế điều khiển TCP[52], [56], [62], [67], [73], cải thiện cơ chế AIMD [54], điều khiển kích thước

gói tin [60], điều khiển bằng hàm lưu lượng [57] hay bổ sung cơ chế che lỗi đường truyền và khôi phục liên kết (một dạng snoop ở tầng liên kết) [17].

TCP gián tiếp (Indirect TCP)[2], [3], [4], [9], [10], [36]: theo cơ chế này, kết nối TCP từ người gửi sẽ kết thúc tại đầu vào đường truyền hay gây lỗi, nơi đặt tiến trình TCP, tiến trình sẽ biên nhận các gói số liệu mà nó nhận được và chịu trách nhiệm gửi nó đến điểm đến. Trên đường truyền không dây, nơi có tỉ lệ lỗi bit cao và thất thường, một kết nối TCP được tinh chỉnh cho phù hợp với đặc điểm của đường truyền này (thí dụ SACK TCP [27]) sẽ thiết lập. Ngoài kết nối TCP, tại đây, cũng có thể sử dụng một giao thức giao vận khác.

Nhược điểm của I-TCP là không tuân thủ nguyên tắc điều khiển lưu lượng “đầu cuối - đầu cuối” trong giao thức TCP, nút trung gian (trong đó có tiến trình TCP) gửi biên nhận thay cho người nhận, do đó biên nhận có thể đến người gửi trước khi gói số liệu thực sự đến người nhận. Ngoài ra, I-TCP cũng gây ra các khó khăn cho các trạm gốc, vì chúng phải chuyển cho nhau một lượng lớn thông tin trạng thái khi xảy ra việc chuyển cuộc gọi.

Snoop-TCP [15],[22]: cơ chế thực hiện tiến trình TCP thứ hai này tôn trọng ý nghĩa “end-to-end”. Tiến trình nằm giữa hai mạng không chia đôi kết nối TCP, nó chỉ giữ bản sao các gói số liệu chứ không tự sinh ra các biên nhận. Các biên nhận không phải là lặp mà bên nhận gửi lại sẽ được tiến trình chuyển tiếp tới cho bên gửi, còn các biên nhận lặp sẽ bị chặn lại. Khi nhận được biên nhận lặp thứ ba, hoặc khi tiến trình đã đợi quá một khoảng thời gian hết giờ cục bộ, gói số liệu tương ứng sẽ được tiến trình phát lại. Thời gian hết giờ cục bộ này phải được xác định phù hợp với đường truyền không dây chỉ có một chặng, nó đương nhiên là nhỏ hơn thời gian hết giờ mà bên người gửi (nguồn) sử dụng. Giải pháp này cũng có nhược điểm tương tự giải pháp phát lại ở tầng liên kết dữ liệu, đó là việc nó có thể cản trở các cơ chế kiểu đầu cuối - đầu cuối. Về thực chất giải pháp Snoop-TCP cũng giống giải pháp phát lại ở tầng liên kết dữ liệu, chúng che giấu mọi sự mất gói số liệu gây ra bởi

đường truyền. Cả hai giải pháp này đều đòi hỏi không có sự mất gói số liệu do tắc nghẽn trên đường truyền giữa Snooper agent và điểm đến.

Split TCP and UDP là cơ chế được Jorge Navarro-Ortiz đề cập trong [63], trong đó các thiết bị quản lý truy cập tại trạm gốc chịu trách nhiệm trong việc tắt bật các cơ chế kiểm soát tắc nghẽn trên đường truyền không dây. Cơ chế này tận dụng chức năng quản lý phân luồng HCCA (Hybrid Coordination function Controlled Channel Access) của chuẩn IEEE 802.11e. Với cải tiến này, TCP được cho là giảm tải, do giảm số lượng các gói phản hồi. Song việc ứng dụng của giải pháp này, như thiết kế, chỉ giới hạn với chức năng HCCA và 802.11e.

Lin and Long trong [66] đề xuất sử dụng tham số trung bình biến thiên của thời gian các gói tin tới đích để xác định nguyên nhân mất gói tin, qua đó điều khiển tốc độ truyền tin tương ứng. [67][68][69] cũng là các đề xuất khác nhằm điều chỉnh hành vi của TFRC[65][66][67], dựa trên tham số tỷ lệ biến thiên thời gian tới của gói tin, dựa trên đặc tính thiết bị AP (Access Point) nằm ở vị trí trạm gốc, hoặc tách rời thời gian khứ hồi gói tin khi tính đến thời gian xử lý để sinh ra gói tin phản hồi.

TCP-ADW là đề xuất của A M Al-Jubari và đồng sự [72], cố gắng che giấu số lượng các gói tin phản hồi ACK bị chậm liên tục, để tránh việc TCP kích hoạt quá trình gửi lại các gói tin, mà theo [72] là không cần thiết. Mục tiêu của ADW là giảm số lượng các gói tin ACK về giá trị tối ưu nhất, đảm bảo sự ổn định của giao thức TCP và giảm tải cho kênh truyền. ADW tính toán giá trị các gói tin tối ưu này, dựa trên nhiều tham số như giá trị cửa sổ truyền tin, khoảng cách giữa hai gói tin liên tiếp, độ dài đường truyền và tỷ lệ mất gói tin. Với sự điều chỉnh này, bên thu có thể thích ứng với nhiều kiểu trễ tương ứng với kênh truyền không dây. đương nhiên thời gian trễ của các gói tin ACK cần được kiểm soát để nhỏ hơn giá trị RTO, nếu không bên gửi sẽ kích hoạt quá trình truyền lại gói tin, mà nó cho là đã bị thất lạc.

Trong thiết kế của ADW, nếu một gói tin dữ liệu cập bến, trước thời điểm nó được mong đợi tại bên nhận, không gói tin ACK nào được gửi đi. ADW cập nhật cho bên gửi khi cần thiết và tránh cho bên gửi bị mất các gói ACK khi xảy ra lỗi mất gói

tin. Cơ chế trì hoãn ACK giúp ADW hoạt động hiệu quả, do sử dụng băng thông hiệu quả, đặc biệt khi kênh truyền trong trạng thái không ổn định và khoảng cách truyền lớn. Sử dụng giá trị cửa sổ truyền linh động, ADW ở góc độ nào đó, có thể phát hiện nguồn lực còn lại của đường truyền, dù không có thông tin từ các tầng phía dưới. Một vấn đề quan trọng ADW gặp phải là bên nhận phải được thông báo về kích cỡ cửa sổ truyền của bên gửi, thông qua một trường trong phần đầu của gói tin. Tuy nhiên trong thực tế các phiên làm việc của TCP thường là một chiều, các gói dữ liệu được gửi từ bên gửi đến bên nhận, nên trường thông tin về kích cỡ cửa sổ truyền trong phần đầu của gói tin dữ liệu từ phía bên nhận thường bị bỏ phí.

P. Kumar trong [73] nêu ra ý tưởng xây dựng mô hình kiểm soát tắc nghẽn dựa trên tham số tốc độ truyền, kích cỡ vùng nhớ đệm, để tính toán số lỗi mất gói tin riêng do nguyên nhân tắc nghẽn. P. Kumar cũng chỉ ra rằng, các giải pháp hiện nay còn nhiều vấn đề trong việc phân biệt lỗi, để giúp hiệu năng của TCP trong môi trường không dây tận dụng nguồn lực của mạng.

2.2.1.3 Các giải pháp liên tầng

TCP NRT là một trong những đề xuất mới nhất, cải tiến giao thức TCP cho truyền thông không dây của Sreekumari, Lee [70], nhằm phục hồi một cách hiệu quả cách gói tin theo giá trị RTO, thông qua việc phân biệt các lỗi mất gói tin ngẫu nhiên trong khi tính toán giá trị RTO, nhờ đó nâng cao hiệu suất của TCP. Đề xuất NRT cũng là một dạng che dấu những mất mát trên đường truyền, đồng thời có sử dụng việc thông báo về lỗi tắc nghẽn.

NRT phân biệt các lỗi mất gói ngẫu nhiên do đường truyền và mất gói do tắc nghẽn, và không đưa thông số các mất mát gói tin do đường truyền vào việc tính toán giá trị RTO. Nhờ đó NRT phản ứng ít nhạy cảm hơn TCP cho các lỗi mất gói tin dạng này, và nâng cao hiệu năng của TCP.

NRT cần phải sửa đổi ECN, như các khuyến cáo trong chuẩn RFC 3168 và được triển khai trong nhiều bộ định tuyến trên Internet, song ECN yêu cầu cơ chế quản lý hàng đợi chủ động để có thể hoạt động tốt. Phần lớn các cơ chế quản lý hàng

đội có thể đạt được hiệu năng cao với việc kích hoạt ECN, thậm chí khi tải của đường truyền cao hơn 85% [69]. ECN sử dụng cơ chế quản lý hàng đợi RED, với ý tưởng thiết kế chủ đạo là thông báo cho phía người gửi về việc phòng ngừa tắc nghẽn bằng cách loại bỏ các gói tin đến theo một xác suất nào đấy, dù bộ đệm vẫn chưa đầy. Các nghiên cứu gần đây [69] chỉ ra rằng, các bộ định tuyến sử dụng cơ chế quản lý hàng đợi RED sẽ có hiệu năng không tốt, nếu không có ECN để đánh dấu gói tin

Cải tiến TFRC là đề xuất của Bin Zhou trong [65], nhằm cải thiện hoạt động của TFRC, một giao thức họ TCP tuân theo công thức thông lượng Reno, tức là thiết kế cho mạng có dây. TFRC bộc lộ điểm yếu trong môi trường mạng hỗn hợp, khi có sự tham gia của môi trường không dây. Với TFRC cải tiến, công thức thông lượng được điều chỉnh trực tiếp từ mô hình TCP không dây, do đó trong tình huống lỗi gói tin trong môi trường không dây được cho là 10%, hiệu quả của đề xuất có thể tăng 300% so với TFRC nguyên bản. Vấn đề phân biệt lỗi đường truyền thuộc loại nào, do tắc nghẽn hay do môi trường vẫn chưa được giải quyết.

Các đề xuất [48],[50] sử dụng tín hiệu liên tầng để thích nghi kênh truyền cho các dữ liệu luồng video thời gian thực được đảm bảo chất lượng dịch vụ, song giới hạn nghiên cứu là cho mạng wifi.

Đề xuất liên tầng của **Taleb** trong [64] liên quan đến điều chỉnh hoạt động của 5 tầng vật lý, kết nối dữ liệu, mạng, tầng giao vận và tầng ứng dụng. Cơ chế nhận biết trạng thái của mỗi tầng có một thế mạnh, do đó để nhận biết nguồn lực hiện có của hệ thống, tầng giao vận gửi các gói tin thăm dò và tham số giữa các tầng được trao đổi với nhau, giúp khả năng đánh giá hiện trạng của mạng chính xác và nhanh chóng. Tuy nhiên đề xuất này phá vỡ kiến trúc mạng phân lớp hiện nay, và yêu cầu thay đổi rất lớn từ tất cả các nhà sản xuất thiết bị. Việc mô hình phân lớp hiện nay được chấp nhận rộng rãi và triển khai bởi tất cả các nhà sản xuất, do nó cho phép các tầng hoạt động độc lập và không phụ thuộc tầng sâu phía dưới, mà chỉ giao tiếp với các tầng sát cạnh. Tầng ứng dụng do vậy không cần quan tâm thiết bị phần cứng hay trình điều khiển nào đang phục vụ nó. Việc phá vỡ cấu trúc này có thể giải quyết

phần nào bài toán nhận biết thông số môi trường, song sẽ tạo ra hàng loạt vấn đề tương thích giữa các thiết bị mạng.

2.2.2 Thông báo rõ ràng về nguyên nhân mất gói số liệu

Việc cải thiện chất lượng các đường truyền Internet không che giấu được tất cả các mất mát không phải do tắc nghẽn. Nguyên nhân thứ nhất là, mặc dù các mất mát gói số liệu kiểu này tập trung trên các đường truyền không dây, nhưng chúng cũng có thể xuất hiện ở mọi nơi trong mạng và có lẽ không thể che giấu hết các lỗi này. Thứ hai là, các cơ chế kiểu cục bộ chẳng hạn như FEC cũng không thể khôi phục được các gói số liệu bị mất.

Lớp kỹ thuật thứ hai được trình bày dưới đây nhằm cải thiện hiệu suất TCP bằng một số cơ chế làm cho bên gửi nhận thấy được sự tồn tại của các chặng không dây và những sự mất mát gói số liệu không phải do tắc nghẽn mạng. Như vậy, sẽ tránh cho bên gửi khỏi việc kích hoạt các thuật toán điều khiển tắc nghẽn khi xảy ra mất mát gói số liệu không do tắc nghẽn.

Thông báo rõ việc mất gói số liệu – ELN (Explicit Loss Notification): Cho đến nay, người ta đã đề xuất hai cách tiếp cận, cách thứ nhất là thông báo rõ ràng nơi xảy ra việc mất gói số liệu không phải do tắc nghẽn bằng tín hiệu ELN. Bên gửi sẽ phản ứng lại bằng cách phát lại gói số liệu bị mất mà không giảm kích thước cửa sổ của nó. Người ta cũng đề xuất việc sử dụng một tín hiệu giống hệt như vậy để dừng ngay việc điều khiển tắc nghẽn ở bên gửi khi xuất hiện sự đứt kết nối do việc chuyển cuộc gọi trong mạng tế bào. Cái khó của việc thực hiện giải pháp này là nếu có gói số liệu bị hỏng, nó sẽ bị loại ngay tại tầng liên kết dữ liệu trước khi được chuyển lên cho tầng giao vận, vì thế khó có thể nhận được thông tin ELN.

Thông báo rõ về tắc nghẽn - ECN (Explicit Congestion Notification): Cách tiếp cận thứ hai là cải tiến sự điều khiển lưu lượng của giao thức TCP chứ không phải là việc khôi phục lại sau khi có sự mất gói số liệu không phải do tắc nghẽn. Các giải pháp được người ta đề xuất nhằm mục đích tách việc phát hiện tắc nghẽn khỏi vấn đề mất mát gói số liệu. Bằng cách bổ sung một số cơ chế vào trong mạng hoặc bổ sung

vào bên nguồn, tắc nghẽn sẽ bị phát hiện và thông lượng sẽ được giảm trước khi xảy ra tràn các bộ đệm trong mạng. Một trong các giải pháp như vậy được thực hiện trong phiên bản Vegas TCP [24] [25].

Với ECN, các bộ định tuyến sẽ gửi một tín hiệu chỉ rõ sự tắc nghẽn trong mạng cho bên gửi của kết nối TCP, chứ không loại bỏ các gói số liệu. Nếu tất cả mọi người gửi, người nhận và bộ định tuyến đều phục tùng cơ chế này (Vegas hoặc ECN), sự mất gói số liệu do tắc nghẽn sẽ được giảm đi đáng kể. Do đó, các sự mất mát còn lại có thể được xem chủ yếu gây ra bởi các vấn đề khác, chứ không phải là tắc nghẽn; bên gửi chỉ cần phát gói số liệu mà không phải giảm kích thước cửa sổ. Sự biến mất của các mất mát do tắc nghẽn có thể dẫn tới việc xác định lại thuật toán điều khiển tắc nghẽn của giao thức TCP, để cho nó phản ứng đỡ quá mạnh mẽ đối với việc mất gói số liệu.

Các mạng ngày nay không có hành trạng lý tưởng như vậy. Bởi vì không có sự phản hồi từ mạng, nên việc phát hiện tắc nghẽn sớm, trước khi nó xảy ra, tại bên gửi, sẽ không thực hiện được; dẫn đến sự mất gói số liệu do tắc nghẽn là không thể tránh được. Nếu chúng ta muốn thực hiện cơ chế điều khiển tắc nghẽn tại nguồn, dựa trên các thông tin rõ ràng nhận được từ mạng, như trong ECN, thì điều chắc chắn là sẽ không dễ dàng, vì trong mạng, sẽ luôn có một số bộ định tuyến không tuân theo mệnh lệnh cung cấp cho bên nguồn các thông tin nó yêu cầu. Điều này là một thực tế, bởi vì trên Internet, hiện có hàng chục, thậm chí hàng trăm nghìn bộ định tuyến, thuộc nhiều chủ sở hữu khác nhau, không ai có thể buộc các chủ sở hữu này thay đổi cơ chế hoạt động của các bộ định tuyến của họ.

2.3 Xác định nhu cầu băng thông và trạng thái đường truyền

Xác định chính xác được nhu cầu băng thông của các kết nối và băng thông khả dụng của đường truyền, giúp việc phân bổ băng thông khả dụng một cách hợp lý, hoặc giúp các bên gửi và nhận nhận biết sớm khả năng nhu cầu của mình không thể được đáp ứng và có quyết định phù hợp, có thể được coi là kỹ thuật cơ bản để điều khiển tắc nghẽn. Do đó, TCP cũng có cơ chế để xác định băng thông, nhưng dựa trên

các gói tin biên nhận thành công, nên TCP cần nhiều thời gian để tính toán. Đề xuất ước lượng băng thông khả dụng vẫn tiếp tục được công bố như [58],[59] cho thấy tìm ra phương pháp ước lượng chính xác hơn nhu cầu băng thông vẫn là nhu cầu cần thiết.

Luận án đề xuất phương pháp xác định nhu cầu băng thông và trạng thái đường truyền nhanh chóng từ mỗi nút mạng, dựa trên tốc độ đến gói tin và kích thước bộ đệm, từ đó đảm bảo khả năng điều khiển tắc nghẽn nhanh hơn.

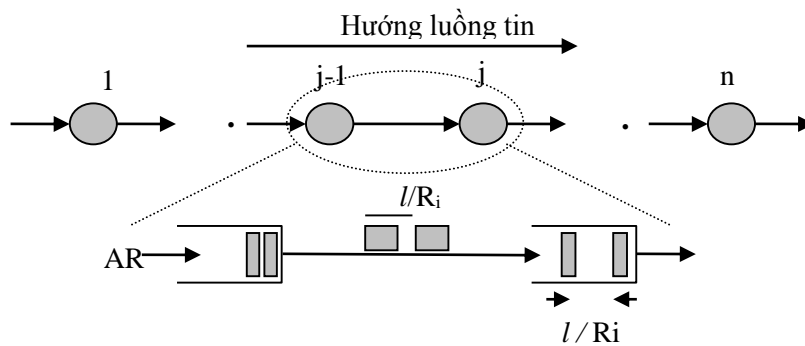
Ta xét một đường truyền dẫn phải qua nhiều nút mạng, có hình thức phổ biến như hình 2.2 phía dưới đây, và không mất tính tổng quát khi ta xét trên đoạn giữa nút $j-1$ và j có các gói tin có cùng kích cỡ.

Ta có link j nằm giữa nút j và $j-1$, các gói tin gửi đến nút $j-1$ với tốc độ là AR (bps), là tổng của các luồng tới nút $j-1$.

$$AR = \sum \lambda_i$$

Trong đó λ_i là tốc độ tới của ứng dụng thứ i tại nút $j-1$ (Hình 3.5a) Mỗi ứng dụng có mức độ ưu tiên tương ứng với trọng số γ_i

$$\lambda_i = \gamma_i * AR \text{ và } \sum \gamma_i = 1$$



Hình 2.2 Nghiên cứu 2 nút mạng tổng quát

Khoảng cách thời gian đến giữa hai gói tin liên tục tại nút $j-1$ là l/AR . Một cách tổng quát, hai gói tin này có thể thuộc hai ứng dụng khác nhau.

Ta gọi R_j là tổng băng thông được sử dụng trên kết nối thứ j . C là băng thông danh nghĩa, thường được biết trước. Nếu ước lượng được R_j , ta sẽ suy ra băng thông khả dụng của kết nối j là hiệu số giữa băng thông danh nghĩa và băng thông đã sử dụng.

Thời gian truyền gói tin có kích thước l sẽ là $\tau = l/R_j$ (giả thiết kích thước gói bằng nhau). Do thời gian trễ là như nhau với các gói tin có cùng kích cỡ, nên tốc độ các gói tin được gửi đi từ nút $j-1$ sẽ là tốc độ gói tin được nhận tại nút j và thời gian giữa hai gói tin liên tục sẽ là l/R_j , với giả thiết bộ đệm tại $j-1$ luôn có gói để gửi đi.

Thời gian cần để gửi gói tin thứ k có kích thước l tại nút j được tính theo công thức

$$T_j^k = \frac{l}{R_{j-1}} + tf_{j-1} + tp_{j-1}$$

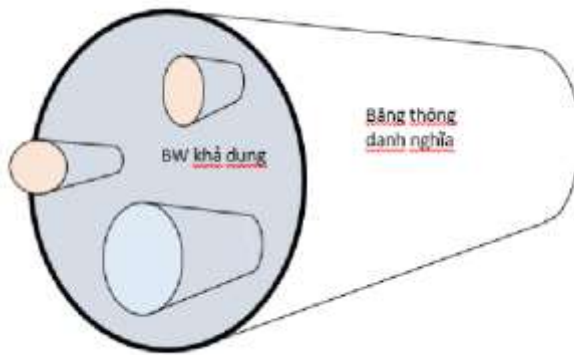
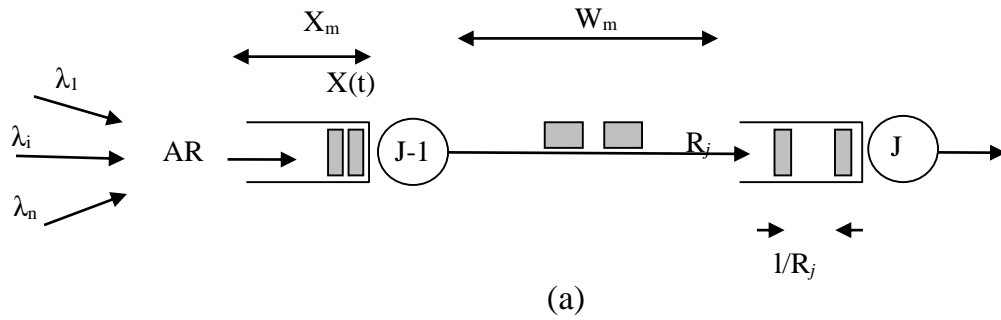
Thời gian cần để gửi gói tin thứ $k+1$ có kích thước L tại nút j được tính theo công thức

$$T_j^{k+1} = \frac{l}{R_{j-1}} + tf_{j-1} + tp_{j-1} + \frac{l}{R_j}$$

Do đó ta có thời gian giữa hai gói tin liên tục là

$$\Delta = T_j^{k+1} - T_j^k = \frac{l}{R_j}$$

Khi các gói tin đến nút $j-1$, chúng sẽ được lưu trong bộ đệm để chờ đến lượt để truyền phát đến nút j . Ta ký hiệu $Q(t)$ là kích thước thực của bộ đệm tại thời điểm t , X_m là kích thước tối đa của bộ đệm. AR là tốc độ gói đến nút $j-1$ và lưu tại bộ đệm, R_j là tốc độ gói ra khỏi bộ đệm, và cũng là tốc độ gói đến nút j , tức là R . W là số lượng gói tin đang được vận chuyển trên đường truyền từ $j-1$ đến j .



Hình 2.3 Mô hình xác định băng thông khả dụng

Từ lý thuyết hàng đợi [32][33], ta coi hệ kết hợp giữa đường truyền j-1,j và bộ đệm tại nút j-1 là một hàng đợi lớn. Khi đó ta có:

$$Q(t) + W(t) = Q(t-) + W(-) + A(t) + D(t) \quad (2.1)$$

Với $A(t)$ và $D(t)$ là số gói tin đến nút j-1 và j tại thời điểm t. $Q(t)$ là kích thước bộ đệm tại j-1 tại thời điểm t, $W(t)$ là số gói đang trên đường được vận chuyển từ nút j-1 đến j, tại thời điểm t.

$A(t_1, t_2)$ và $D(t_1, t_2)$ ký hiệu cho lưu lượng đến và đi khỏi nút j-1 trong khoảng thời gian $[t_1, t_2]$, ta có

$$Q(t) + W(t) = \max_{0 \leq t_1 \leq t_2} [A(t_1, t_2) - D(t_1, t_2)] \quad (2.2)$$

Áp dụng mô hình chất lỏng [74], trạng thái của hệ này có thể được biểu diễn bằng phương trình vi phân sau:

$$\frac{\partial Q(t)}{\partial t} + \frac{\partial W(t)}{\partial t} = AR(t) - R(t) \quad (2.3)$$

Giả thiết, chọn giá trị cho W trong tình huống xấu nhất, ta có $W = R \times T_{j-1,j}$ với $T_{j-1,j}$ là thời gian để gói tin đi được từ j-1 tới j. Ngoài ra, để hệ thống ổn định, không bị tràn bộ đệm, R cần biến đổi sao cho phù hợp với độ lớn hàng đợi trong các khoảng thời gian điều khiển rất ngắn, do đó có thể sử dụng ký hiệu $R(t)$ thay cho R và thiết kế $R(t)$ tỷ lệ với $Q(t)$, ta có

$$R(t) \sim Q(t)$$

Ta chọn tham số α , sao cho:

$$R(t) = \alpha \cdot Q(t) \Rightarrow Q(t) = R(t) / \alpha \quad (2.4)$$

Phương trình vi phân (3.3) ở trên trở thành:

$$\frac{\partial R(t)}{\partial t} \left(\frac{1 + \alpha \cdot T_{j-1,j}}{\alpha} \right) = AR(t) - R(t) \quad (2.5)$$

Để đơn giản hóa cách giải phương trình trên, ta giả thiết AR là tổng nhu cầu về băng thông của các ứng dụng và có giá trị không đổi trong khoảng thời gian quan sát $T_{qs} = [t_1, t_2]$. Giả thiết $T_{qs} = \sigma \cdot T_{j-1,j}$ Nghiệm của (2.5) có dạng:

$$R(t) = AR \left[1 - e^{-\frac{\sigma \cdot \alpha \cdot t}{\sigma + \alpha \cdot T_{qs}}} \right] + R_0 \cdot e^{-\frac{\sigma \cdot \alpha \cdot t}{\sigma + \alpha \cdot T_{qs}}}, 0 \leq t \leq T_{qs}$$

Chọn $\sigma = 1$ để đơn giản hóa nghiệm trên, ta có

$$R(t) = AR \left[1 - e^{-\frac{\alpha \cdot t}{1 + \alpha \cdot T_{qs}}} \right] + R_0 \cdot e^{-\frac{\alpha \cdot t}{1 + \alpha \cdot T_{qs}}}, 0 \leq t \leq T_{qs} \quad (2.6)$$

Nghĩa là ước lượng được giá trị $R(t)$ là tổng băng thông sử dụng cho tất cả các luồng trên liên kết j-1, j, dựa vào giá trị ban đầu đo được trước đó R_0 .

Gọi L là tổng số bit các gói tin tới nút $j-1$ trong khoảng thời gian quan sát T_{qs} , khi đó có thể tính AR là giá trị trung bình (tốc độ trung bình của tất cả các luồng tin đi vào nút $j-1$).

$$AR = \frac{L}{T_{qs}}$$

Ta viết lại (2.6) như sau:

$$R(t) = \frac{L}{T_{qs}} \left[1 - e^{-\frac{\alpha t}{1 + \alpha T_{qs}}} \right] + R_0 \cdot e^{-\frac{\alpha t}{1 + \alpha T_{qs}}}, 0 \leq t \leq T_{qs} \quad (2.7)$$

Nếu coi các luồng tin đến là các tiến trình ngẫu nhiên, độc lập nhau thì có thể ước lượng được bằng thông sử dụng cho mỗi luồng tin i trên liên kết $j-1, j$ theo hệ phương trình sau:

$$\begin{cases} R_1(t) = \frac{L_1}{T_{qs}} \left[1 - e^{-\frac{\alpha t}{1 + \alpha T_{qs}}} \right] + R_1(0) \cdot e^{-\frac{\alpha t}{1 + \alpha T_{qs}}} \\ R_2(t) = \frac{L_2}{T_{qs}} \left[1 - e^{-\frac{\alpha t}{1 + \alpha T_{qs}}} \right] + R_2(0) \cdot e^{-\frac{\alpha t}{1 + \alpha T_{qs}}} \\ \vdots \\ R_i(t) = \frac{L_i}{T_{qs}} \left[1 - e^{-\frac{\alpha t}{1 + \alpha T_{qs}}} \right] + R_i(0) \cdot e^{-\frac{\alpha t}{1 + \alpha T_{qs}}} \\ \vdots \\ \vdots \end{cases} \quad (2.8)$$

Do nhu cầu về băng thông của các luồng phụ thuộc trọng số γ , như đề cập ở trên. Hệ phương trình băng thông của các luồng này, được đơn giản hóa như sau:

$$\begin{cases} R_1(t) = \frac{L_1}{T_{qs}} \left[1 - e^{-\frac{\alpha t}{1 + \alpha T_{qs}}} \right] + R_0 \cdot \gamma_1 \\ R_2(t) = \frac{L_2}{T_{qs}} \left[1 - e^{-\frac{\alpha t}{1 + \alpha T_{qs}}} \right] + R_0 \cdot \gamma_2 \\ \vdots \\ R_i(t) = \frac{L_i}{T_{qs}} \left[1 - e^{-\frac{\alpha t}{1 + \alpha T_{qs}}} \right] + R_0 \cdot \gamma_i \\ \vdots \\ \vdots \end{cases} \quad (2.9)$$

Nhận xét:

Giá trị băng thông sử dụng ước lượng trên liên kết j-1, j gồm hai thành phần:

- Thành phần phụ thuộc vào giá trị băng thông sử dụng trước đó (R_0), nghĩa là thời gian bắt đầu quan sát.
- Thành phần tức thời, phụ thuộc vào số lượng bit của các luồng tin đến và thời gian quan sát T_{qs} ,

Rõ ràng là, giá trị ước lượng sẽ chính xác hơn và tiệm cận đến giá trị đúng, khi thành phần 1 rất lớn, thành phần 2 nhỏ (độ dung sai).

Phương trình (2.7) chính là dạng tổng quát để xác định giá trị băng thông mà ở đó, giá trị mới được tính theo một phần giá trị cũ cộng thêm một phần giá trị vừa đo được.

Thông thường trong các giao thức họ TCP, thành phần 1 có thể chiếm tỷ lệ lớn, thành phần 2 chiếm tỷ lệ nhỏ, để tăng được độ chính xác và ổn định của ước lượng. Đó là cơ sở để có thể xác định hệ số α và T_{qs}

Ví dụ, chọn tỷ lệ của phần dung sai là X:

$$1 - e^{-\frac{\alpha \cdot t}{1 + \alpha \cdot T_{qs}}} = X \Rightarrow \alpha = \frac{\ln\left(\frac{1}{1-X}\right)}{t - T_{qs} \cdot \ln\left(\frac{1}{1-X}\right)} \quad (2.10)$$

Như vậy, ta có giá trị của α phụ thuộc hàm giá trị $f(T)$ của thời gian quan sát

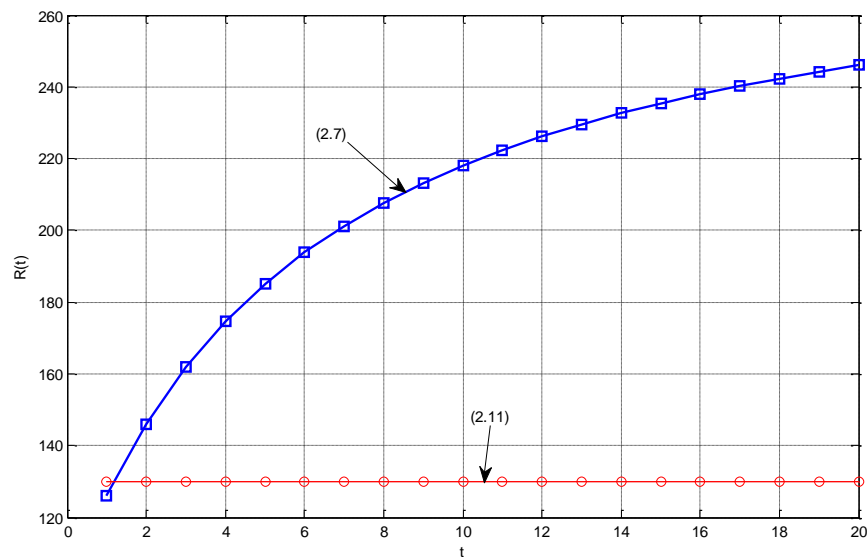
Để phù hợp với họ các giao thức TCP, thành phần $X=1/8 \approx 0,1 \Rightarrow \alpha \approx \frac{1}{10t-T}$, công thức (2.7) có thể biến đổi thành:

$$R_{uoc\ luong} = \frac{L}{T_{qs}} \cdot 0,1 + R_0 \cdot 0,9 \quad (2.11)$$

Phương trình (2.7) (2.11) sẽ được đưa vào mô hình tính toán để đối sánh.

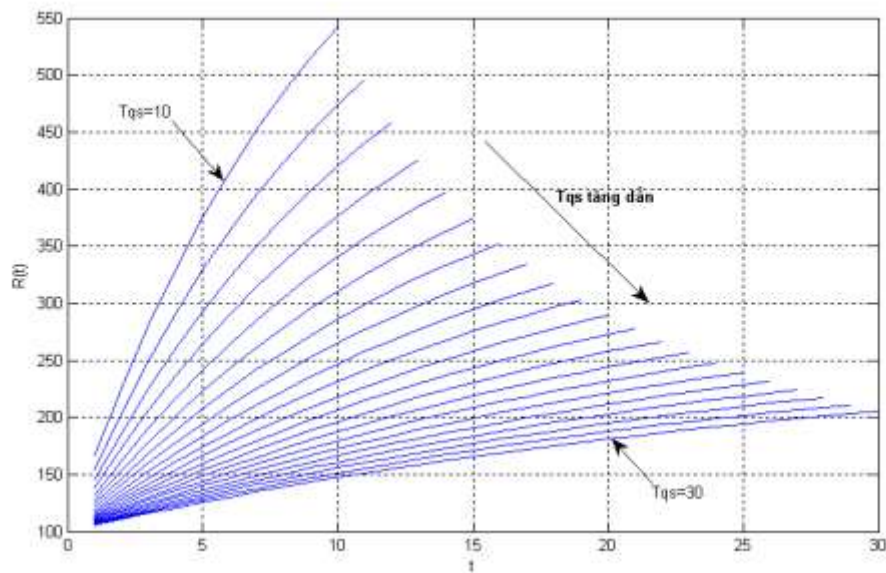
Tính toán mô phỏng

- Nghiên cứu biến đổi của băng thông ước lượng $R(t)$, theo hai công thức (2.7) và (2.11). Ý nghĩa vật lý của α trong (2.7) là tỷ lệ giữa giá trị hàng đợi tại nút mạng $j-1$ và băng thông sử dụng trên đoạn mạng $j-1,j$ do vậy $\alpha > 0$. $R(t)$ được tính toán theo (2.7) với $R_0=100$; $T_{qs}=20$; $L=8000$; $\alpha =10$ là đường cong có đánh dấu bằng các nút vuông. $R(t)$ được tính theo công thức (2.11) sẽ cho đường được đánh dấu bởi các nút tròn (Hình 2.4)

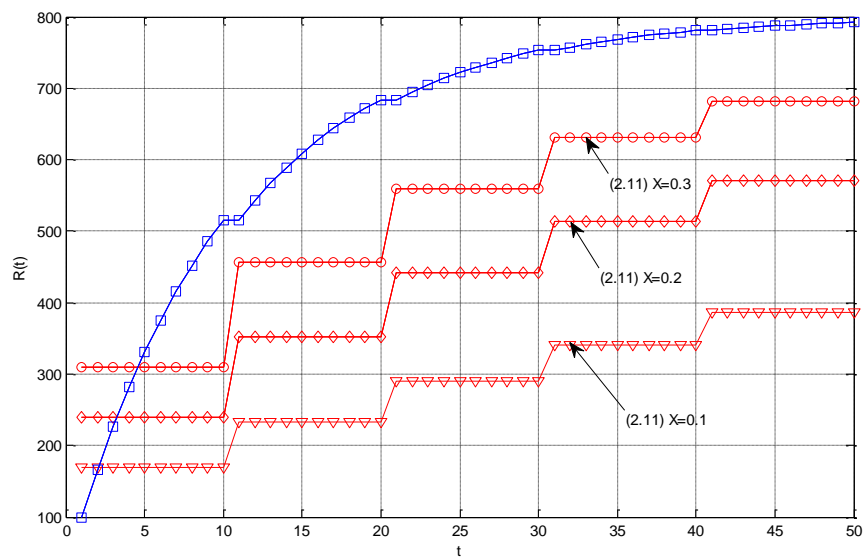


Hình 2.4 So sánh giá trị $R(t)$ theo công thức (2.7) và (2.11).

- Nghiên cứu biến đổi của băng thông ước lượng $R(t)$ trong công thức (2.7), theo sự biến đổi của thời gian quan sát T_{qs} , với các giá trị $L=8000; R_0=100$; $\alpha = 100$, $T_{qs} = 10..30$ ta có đồ thị như trong Hình 2.5. Với T_{qs} nhỏ, $R(t)$ có giá trị băng thông lớn hơn, do đó đường $R(t)$ dốc hơn. Như vậy với T_{qs} nhỏ, phép ước lượng băng thông nhạy cảm hơn, nhanh chóng



Hình 2.5 Biến đổi của $R(t)$ theo giá trị T_{qs} .



Hình 2.6 Biến đổi của $R(t)$ theo trọng số dung sai X

- Nghiên cứu biến đổi của băng thông ước lượng $R(t)$ theo công thức (2.7) cho các đoạn T_{qs} liên tiếp, với các giá trị $L=8000; R_0=100; \alpha = 100, T_{qs} = 10$ và so sánh với trường hợp trọng số của phần dung sai X cố định, như trường hợp được dùng trong các giao thức TCP, lần lượt bằng 0,1;0,2;0,3 như đồ thị tương ứng (Hình 2.6). Đường đồ thị được đánh dấu vuông là giá

trị của $R(t)$ theo công thức (2.7), ta có thể thấy nó xuất phát thấp hơn, nhưng sau đó lại tiệm cận nhanh hơn đến gần giá trị của băng thông, sao một số khoảng thời gian quan sát.

Như vậy, có thể nói, công thức (2.7) cho phép tính băng thông sử dụng nhanh hơn, so với phương pháp làm mịn trước đây của các giao thức TCP, nhờ việc sử dụng trọng số của phần dung sai là một hàm mũ, thay vì là một hằng số. Đồng thời ta tính được băng thông theo giá trị Tqs , là thời gian gói tin đi từ nút mạng $j-1$, đến j , tức là bằng một nửa thời gian RTT thông thường. Công thức (2.7) cho phép tính toán băng thông tại bên nhận, khi có đủ giá trị của các tham số, nhờ đó các giao thức sử dụng công thức này sẽ phản ứng nhanh hơn.

2.4 Kết luận chương II

Ta đã phân tích nhu cầu đảm bảo chất lượng dịch vụ trên mạng hỗn hợp, không đồng nhất, bao gồm đoạn mạng không dây, từ các ứng dụng đa phương tiện, là các ứng dụng phổ biến hiện nay, gồm nhiều loại dữ liệu khác nhau, nhiều loại dịch vụ và ứng dụng với nhu cầu hết sức đa dạng, các thiết bị đầu cuối cũng rất khác nhau về nhiều mặt. TCP được phân tích không đạt hiệu năng cao khi hoạt động trong môi trường mạng như vậy.

Cũng trong chương II, xem xét các nghiên cứu và đề xuất trong việc khắc phục điểm yếu của giao thức truyền thông họ TCP cho đến nay, ta có thể tổng hợp và chia làm hai loại chính. Loại thứ nhất bao gồm các giải pháp che giấu phần mạng hay làm mất gói số liệu do lỗi đường truyền, sao cho bên gửi chỉ phát hiện được những sự mất gói số liệu do tắc nghẽn, loại thứ hai bao gồm các giải pháp cải tiến TCP bằng các cơ chế thông báo rõ ràng về nguyên nhân mất gói số liệu, giúp cho TCP có thể phân biệt được các kiểu mất gói số liệu khác nhau. Nhóm giải pháp loại một bao gồm các đề xuất mới nhất [57,68,69], trong đó cơ chế kiểm soát tắc nghẽn trên đường truyền không dây bị tắt đi, nhằm giải quyết vấn đề khó khăn khi cơ chế chống tắc nghẽn của TCP hoạt động không tốt trong môi trường này. Nhóm giải pháp loại hai tránh thay

đổi cách hoạt động của TCP, do đó không đóng góp nhiều trong việc cải thiện hiệu năng TCP trong phần mạng không dây.

Từ những kết quả khảo sát cho thấy việc cải tiến hoạt động nhằm làm cho giao thức truyền thông đạt hiệu quả hơn vẫn luôn thu hút sự quan tâm của nhiều nhà nghiên cứu, vì hiện chưa có đề xuất nào có thể giải quyết tối ưu vấn đề phức tạp và thiếu ổn định của môi trường không dây trong mạng hỗn hợp.

Xác định chính xác nhu cầu băng thông của các kết nối và băng thông khả dụng của đường truyền, có thể được coi là kỹ thuật cơ bản để điều khiển tắc nghẽn. Do đó, luận án đề xuất phương pháp xác định nhu cầu băng thông và trạng thái đường truyền nhanh chóng từ mỗi nút mạng, dựa trên tốc độ đến gói tin và kích thước bộ đệm, từ đó đảm bảo khả năng điều khiển tắc nghẽn nhanh hơn, so với các giao thức TCP.

Trong chương tiếp theo, từ những cơ sở kết luận được chứng minh trong chương II, luận án đề xuất phương án cải tiến cơ chế quản lý tắc nghẽn của TCP, giúp giao thức thích nghi tốt hơn trong môi trường không dây, cải thiện thông lượng, đồng thời đảm bảo công bằng giữa các luồng dữ liệu.

CHƯƠNG 3. MÔ HÌNH ĐIỀU KHIỂN TRUYỀN THÔNG TỰ THÍCH NGHI CHO HỌ GIAO THỨC TCP TRONG MẠNG KHÔNG DÂY

Với việc họ các giao thức TCP được sử dụng rộng rãi hiện nay, những giao thức mới được đưa ra cần có phản ứng gần giống với cách phản ứng của TCP khi quản lý việc sử dụng băng thông, để đảm bảo mạng được ổn định và không xảy ra tắc nghẽn. Trong chương này, luận án đề xuất phương án cải tiến cơ chế AIMD của giao thức TCP bằng mô hình tự thích nghi, giúp giao thức thích nghi tốt hơn trong môi trường không dây, cải thiện thông lượng, đồng thời đảm bảo công bằng giữa các luồng dữ liệu.

3.1 Mô hình điều khiển thích nghi mạng không dây

Mô hình chống tắc nghẽn AIMD của TCP theo cơ chế cửa sổ như sau:

$W = W + 1$ nếu không mất gói tin

$W = W/2$ nếu mất gói tin

W là giá trị của cửa sổ truyền, nghĩa là số gói tin đang truyền trên mạng.

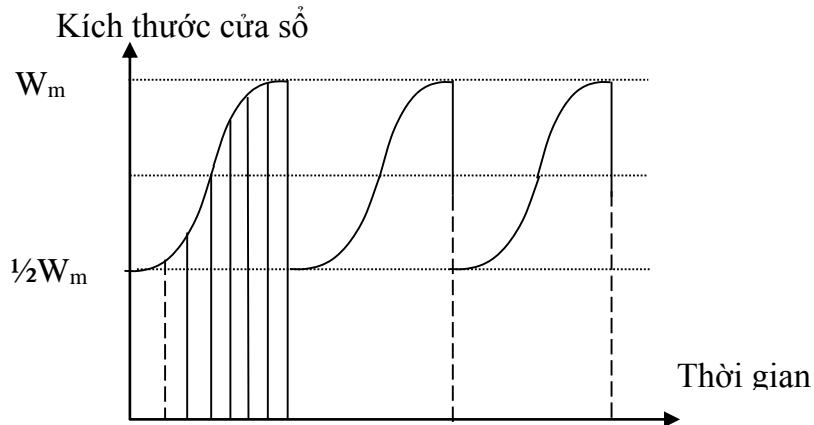
Cơ chế điều khiển thích nghi được đề xuất như sau:

$$W_{t+R} = W_t + a_1 \cdot e^{Kt} + a_2 \cdot e^{-Kt} \quad \text{nếu không mất gói tin} \quad (3.1)$$

$$W_{t+R} = W_t/2 \quad \text{nếu mất gói tin}$$

Trong đó a_1 , a_2 , là các tham số điều khiển; K là hệ số; W_t là cửa sổ tại thời điểm t ; W_{t+R} là cửa sổ điều chỉnh sau thời gian quay vòng, tức là sau khi có phản hồi từ phía nhận.

Cơ chế điều khiển thích nghi được gọi là ECIMD (Exponent Combinational Increase Multiplicative Decrease) được đề xuất thay thế cho cơ chế AIMD của các giao thức họ TCP.



Hình 3.1. Cơ chế điều khiển thích nghi

Sở cứ thực thể của cơ chế này như sau:

- Khi không mất gói tin, cơ chế điều khiển thích nghi cho phép tăng dần tốc độ truyền tin vào mạng. Với việc đưa ra thông tin gồm thành phần $a_1 \cdot e^K$ vào công thức (3.1), ECIMD đã thể hiện giai đoạn khởi động chậm, tăng dần theo hàm mũ. Việc tăng chậm góp phần tránh tăng đột ngột tải lưu lượng trong mạng, đặc biệt khi phải khởi động sau quá trình khôi phục lại tốc độ sau khi mất gói tin.
- Thành phần $a_2 \cdot e^{-Kt}$ trong công thức thể hiện giai đoạn tăng gần bão hòa, nghĩa là tốc độ tăng chậm dần để giúp quá trình thực hiện hạn chế tắc nghẽn và công bằng lưu lượng giữa các luồng tin.
- Hệ số K được xét ở đây là khác 0, vì nếu $K=0$ cơ chế ECIMD sẽ suy biến về dạng AIMD (tăng theo cấp số cộng, giảm theo cấp số nhân)

Hình 3.1 biểu diễn cơ chế điều khiển tăng giảm nói trên cho trường hợp lỗi mất gói đơn (chỉ giảm W một lần). $W_m/2$ là giá trị đặt lại của cửa sổ tương tự TCP mỗi khi mất gói, W_m là giá trị cửa sổ lớn nhất của cửa sổ truyền. Có thể hiểu tốc độ tối đa (hay băng thông sử dụng tối đa của các luồng tin) phụ thuộc số các ứng dụng, nhu cầu ứng dụng (như đã nêu ở 2.3). Luận án xét tình huống cụ thể này, để chứng

minh cơ chế ECIMD vẫn thân thiện với họ các giao thức TCP, do công thức thông lượng của ECIMD có thể suy biến về AIMD.

Xét khoảng thời gian $t_0 \rightarrow T$, thành phần $\mathbf{a}_1 \cdot \mathbf{e}^{Kt}$ là tổng các sóng hài $\sum_{t_0}^T a_1 e^{-Kt}$ và $\mathbf{a}_2 \cdot \mathbf{e}^{-Kt}$ là tổng các sóng hài $\sum_{t_0}^T a_2 e^{-Kt}$

Bởi tính liên tục của hàm mũ, ta có tích phân sau:

$$W_T = W_{t_0} + \int_{t_0}^T a_1 \cdot e^{Kt} dt + \int_{t_0}^T a_2 \cdot e^{-Kt} dt$$

Đơn giản hóa cách giải với $a_1 = a_2 = a$, ta thu được

$$W_T = W_{t_0} + \frac{a}{K} \cdot (e^{Kt} - 1) - \frac{a}{K} \cdot (e^{-Kt} - 1) \quad (3.2)$$

$$\frac{dW(t)}{dt} = a \cdot e^{Kt} + a \cdot e^{-Kt} \quad (3.3)$$

Do đó, ta có phương trình $W(t)$ trong đoạn $t_0 = 0$ đến T quan sát có dạng :

$$W(t) = \frac{a}{K} e^{Kt} - \frac{a}{K} e^{-Kt} + C$$

C là hằng số K là hệ số.

Ta chọn thời gian gốc là $t_0 = 0$, khi đó $W(t_0=0) = W_m/2$.

$$\frac{W_m}{2} = \frac{a}{K} e^{Kt} - \frac{a}{K} e^{-Kt} + C \Rightarrow \text{thay } t=0 \text{ ta được } C = \frac{W_m}{2}$$

$$\Rightarrow W(t) = \frac{a}{K} e^{Kt} - \frac{a}{K} e^{-Kt} + \frac{W_m}{2} \quad (3.4)$$

Để K phản ánh sự ảnh hưởng của kích thước cửa sổ cực đại W_m , ta giả sử không ràng buộc:

$$K = \frac{a}{W_m}$$

Đưa giá trị K vừa chọn vào (3.4) ta có :

$$W(t) = W_m \left(e^{K.t} - e^{-K.t} + \frac{1}{2} \right) \quad (3.5)$$

Tại $t=t_1$; ta có $W_{(t=t_1)} = W_m$

$$\Rightarrow W_m = W_m \left(e^{K.t_1} - e^{-K.t_1} + \frac{1}{2} \right)$$

$$\Rightarrow e^{K.t_1} - e^{-K.t_1} = \frac{1}{2} \quad (3.6)$$

$$\text{Đặt } e^{K.t_1} = x \Rightarrow e^{-K.t_1} = \frac{1}{x}$$

$$(3.6) \text{ biến đổi thành } x - \frac{1}{x} = \frac{1}{2} \Rightarrow 2x^2 - x - 2 = 0$$

Phương trình trên có 1 nghiệm dương $x = \frac{1 + \sqrt{17}}{4}$, tức là $e^{K.t_1} = \frac{1 + \sqrt{17}}{4}$

$$\text{Đặt } X = \frac{1 + \sqrt{17}}{4}$$

$$\Rightarrow t_1 = \frac{1}{K} \ln(X) \quad (3.7)$$

Trong khoảng (t_0, t_1) là thời gian giữa hai gói tin bị mất gần nhất, giả thiết là sau một gói tin bị lỗi, sẽ có $1/p$ gói được gửi thành công với xác suất lỗi p . Như vậy một gói tin sẽ bị mất tại thời điểm t_1 và cửa sổ sẽ giảm thành $W_m/2$.

Đặt N là số gói tin giữa 2 lần bị mất gói tin. N là diện tích hình gạch dọc (Hình 3.1)

$$\begin{aligned} N &= \frac{1}{RTT} \int_{t_0}^{t_1} W(t) dt \\ &= \frac{W_m}{RTT} \int_{t_0}^{t_1} \left(e^{K.t} - e^{-K.t} + \frac{1}{2} \right) dt \end{aligned}$$

$$= \frac{W_m}{RTT} \left(\frac{1}{K} e^{K.t} - \frac{1}{K} e^{-K.t} + \frac{1}{2} t \right) \Big|_{t_0}^{t_1}$$

Với $t_0=0$, ta có:

$$\begin{aligned} N &= \frac{W_m}{RTT} \left[\left(\frac{1}{K} e^{K.t_1} - \frac{1}{K} e^{-K.t_1} + \frac{1}{2} t_1 \right) - \left(\frac{1}{K} + \frac{1}{K} + 0 \right) \right] \\ N &= \frac{W_m}{RTT} \left(\frac{1}{K} e^{K.t_1} - \frac{1}{K} e^{-K.t_1} + \frac{1}{2} t_1 - \frac{2}{K} \right) \end{aligned} \quad (3.8)$$

Thay giá trị t_1 ở (3.7) vào (3.8):

$$\begin{aligned} N &= \frac{W_m}{RTT} \left(\frac{1}{K} .X + \frac{1}{K.X} + \frac{\ln(X)}{2.K} - \frac{2}{K} \right) \\ N &= \frac{W_m}{RTT.K} \left(X + \frac{1}{X} + \frac{\ln(X)}{2} - 2 \right) \\ \text{Đặt } \left(X + \frac{1}{X} + \frac{\ln(X)}{2} - 2 \right) &= Y \\ N &= \frac{W_m.Y}{RTT.K} \\ N &= \frac{W_m^2.Y}{RTT.a} \end{aligned} \quad (3.9)$$

Thay $p=1/N$ từ giả thiết vào phương trình trên, ta thu được

$$\begin{aligned} W_m^2 &= \frac{RTT.a}{p.Y} \\ W_m &= \frac{\sqrt{RTT.a}}{\sqrt{p.Y}} \end{aligned} \quad (3.10)$$

Trong khoảng thời gian (t_0, t_1) có gói tin N được gửi, do vậy thông lượng θ_{ECIMD} sẽ được tính bằng theo công thức:

$$\theta_{ECIMD} = \frac{N}{t_1 - t_0}$$

$$\begin{aligned}
&= \frac{W_m.Y}{RTT.K} : \left(\frac{1}{K} \ln(X) \right) \\
&= \frac{\sqrt{RTT.a}}{\sqrt{p.Y}} . Y . \frac{1}{RTT.\ln(X)}
\end{aligned}$$

$$\theta_{ECIMD} = \frac{\sqrt{a}}{\sqrt{RTT}.\sqrt{p}} \frac{\sqrt{Y}}{\ln(X)} \quad (3.11)$$

Nếu chọn giá trị tham số điều khiển sao cho

$$\sqrt{a} = \beta . \frac{\sqrt{3/2}}{\sqrt{Y.RTT}} \ln(X) \quad (3.12)$$

Công thức (3.11) trở thành

$$\theta_{ECIMD} = \frac{\sqrt{3/2}}{RTT.\sqrt{p}} . \beta \quad (3.13)$$

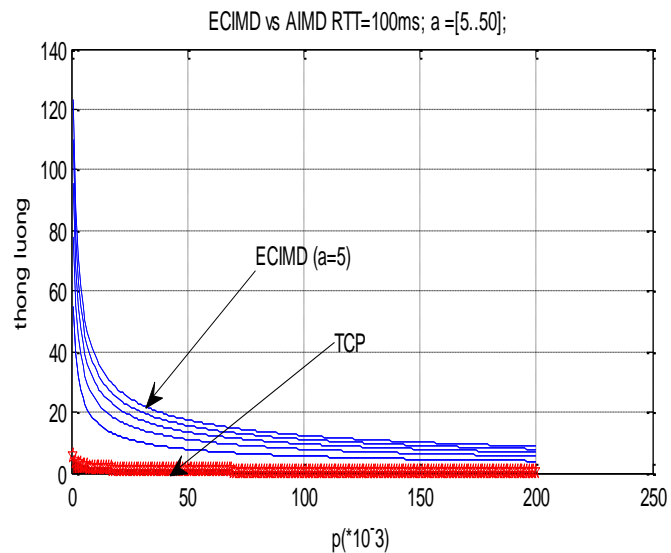
Công thức (3.13) với $\beta = 1$ chính là công thức tính thông lượng được sử dụng trong TCP

$$\theta_{TCP} = \frac{\sqrt{3/2}}{RTT.\sqrt{p}}$$

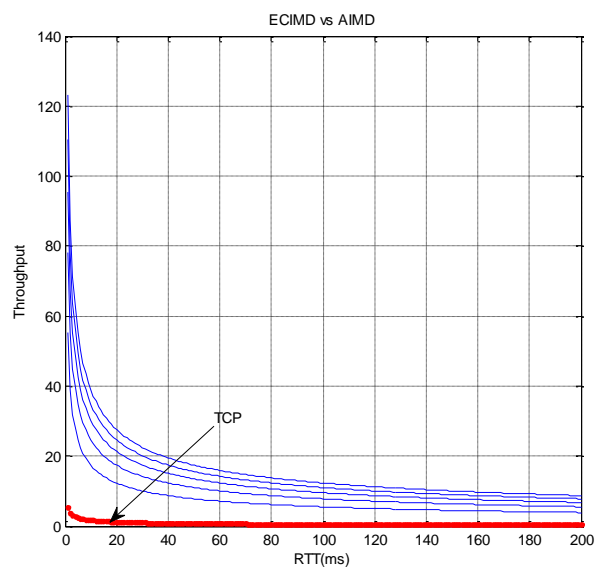
Do vậy có thể nói, *mô hình sử dụng công thức (3.13) tổng quát hơn, so với trường hợp của TCP, và ECIMD vẫn thân thiện với cơ chế AIMD của TCP.* Thông lượng của cơ chế ECIMD có mức độ nhạy cảm hơn với giá trị RTT tùy thuộc a . Do đó khi chọn tham số điều khiển phù hợp để $\beta > 1$, ECIMD đạt thông lượng lớn hơn TCP với cùng giá trị RTT, p của đường truyền.

Tính toán mô phỏng

- Nghiên cứu biến đổi của thông lượng theo công thức (3.11) theo giá trị xác suất mất gói tin $p=[0.001;1]$, và $RTT = 100$, ta có đồ thị như Hình 3.2. Đường TCP có mũi tên chỉ, minh họa cho thông lượng tương ứng của TCP. Giá trị p càng nhỏ, thông lượng càng lớn.



Hình 3. 2 . Biến đổi của Thông lượng theo p



Hình 3. 3 . Biến đổi của Thông lượng theo RTT

- Nghiên cứu biến đổi của thông lượng theo RTT với các tham số điều khiển khác nhau (H 3.3), ta có thể thấy thông lượng đạt được của cơ chế ECIMD cao hơn thông lượng của cơ chế tăng cửa sổ theo AIMD.

3.2 Thuật toán tính RTT

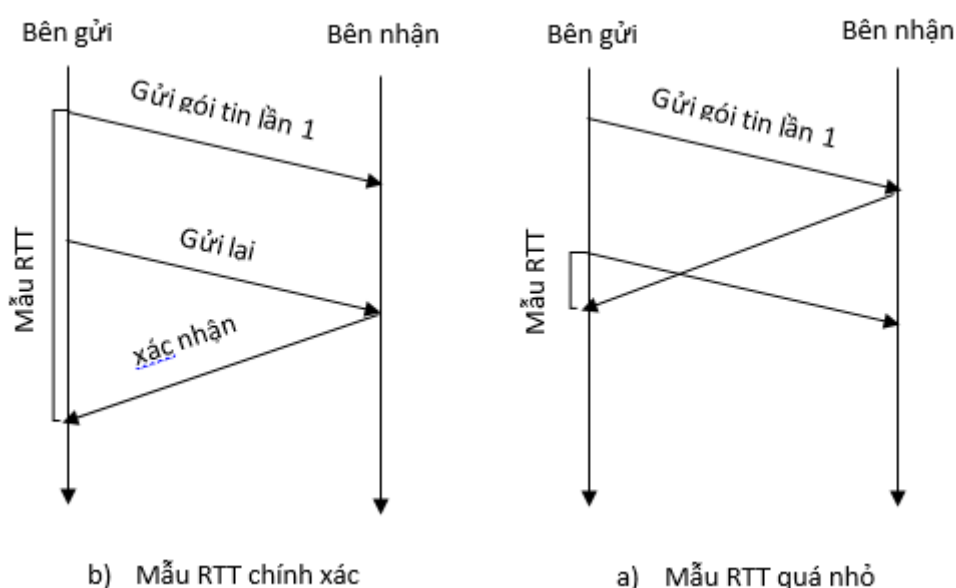
Trong TCP, với thuật toán nguyên gốc, để làm mịn sự biến đổi của giá trị RTT, RTT được áp dụng theo bộ lọc thông thấp:

$$RTT = (\alpha \cdot RTT_{cũ}) + ((1 - \alpha) \cdot SRTT_k) \quad (3.14)$$

$$\text{Hay } RTT_k = (\alpha \cdot RTT_{k-1}) + ((1 - \alpha) \cdot SRTT_k)$$

Với k là ký hiệu cho giá trị thống kê thứ k. $SRTT_k$ là giá trị thời gian khứ hồi đo được tương ứng.

Khi triển khai TCP[38], α được chọn bằng 7/8, để tận dụng đặc tính của các bộ vi xử lý, khi dùng phép toán dịch chuyển thành ghi, tối ưu hơn về tốc độ xử lý, so với việc thực hiện phép toán nhân hay chia.



Hình 3.4 . Các tình huống lấy mẫu RTT bị sai

Theo thuật toán sửa đổi của Karn/Partridge [18], RTT sẽ không được lấy mẫu khi gói tin bị truyền lại, để tránh tình huống lấy mẫu RTT sai trong tình huống như hình minh họa (Hình 3.4)

Theo thuật toán Jacobson/Karels [23], RTT được tính theo cách mới:

$$\text{Diff} = R - \text{RTT}$$

$$\text{RTT} = \text{RTT} + (d * \text{Diff}) \quad \text{với } d=4; \quad (3.15)$$

Tuy nhiên, các công thức tính RTT như trên đều có thể phân tích về dạng một hàm tính giá trị trung bình thống kê mà Luận án sẽ thực hiện ở phần tiếp theo, ở đó, ta sẽ thấy, với giá trị cố định của các hệ số α hay d , công thức tính RTT sẽ thể hiện quan hệ cứng nhắc với các tập hợp mẫu trong quá khứ.

Trên cơ sở phân tích công thức tính RTT hiện tại là thiếu chính xác cho môi trường không dây, do công thức (3.14) là công thức truy hồi, liên quan đến tất cả các giá trị mẫu trong quá khứ, dù chúng không còn liên quan đến trạng thái hiện tại do thiết bị di động có thể đã sang khu vực có sóng mạnh/yếu hơn, gần/xa trạm phát hơn nên thời gian gói tin cần để đến đích cũng thay đổi..., luận án đề xuất phương pháp mới, cho phép xác định một cách linh hoạt hệ số làm mịn tùy theo yêu cầu về số lượng các mẫu thống kê gần nhất, có tính thời sự nhất trong môi trường không dây.

3.2.1 Phân tích công thức RTT theo hàm thống kê

Do Công thức (3.15) có thể biến đổi về dạng (3.14), nên ta chỉ cần phân tích (3.14) làm đại diện:

$$RTT_k = (1 - \alpha) \left(R_k + \frac{\alpha}{1 - \alpha} (RTT_{k-1}) \right)$$

Tiếp tục khai triển RTT_{k-1} theo các giá trị trước đó:

$$RTT_k = (1 - \alpha) \left(R_k + \frac{\alpha}{(1 - \alpha)} ((1 - \alpha) \cdot R_{k-1} + \alpha \cdot RTT_{k-2}) \right)$$

$$RTT_k = (1 - \alpha) (R_k + \alpha \cdot R_{k-1} + \alpha^2 \cdot R_{k-2} + \dots + \alpha^k R_{k-k})$$

Áp dụng hằng đẳng thức:

$$1 - x^n = (1 - x)(1 + x^2 + x^3 + \dots + x^{n-1})$$

$$(1 + x^2 + x^3 + \dots + x^{n-1}) = \frac{1 - x^n}{1 - x} = \frac{1}{1 - x}$$

$$(n \rightarrow \infty; 1 > x > 0)$$

Với $x = \alpha$, ta có

$$RTT_k = \frac{(R_k + \alpha R_{k-1} + \alpha^2 R_{k-2} + \dots + \alpha^k R_{k-k})}{1 + \alpha + \alpha^2 + \dots + \alpha^k} \quad (3.16)$$

Công thức trên có mẫu số là tổng các hệ số của đa thức trên tử số, chính là hàm số thống kê EWMA, là hàm tính bình quân dịch chuyển có trọng số theo hàm mũ, trong đó các giá trị lấy mẫu thứ j là R_{k-j} có trọng số được tính theo hàm mũ bậc j của α .

Ta có thể tính tỷ số của tổng trọng số của N trạng thái gần nhất, so với trọng số toàn bộ các mẫu thống kê, theo công thức:

$$WR = \frac{(1 + \alpha + \alpha^2 + \dots + \alpha^N)}{(1 + \alpha + \alpha^2 + \dots + \alpha^\infty)}$$

$$WR = \frac{\left(\frac{1 - \alpha^{N+1}}{1 - \alpha}\right)}{\frac{1}{1 - \alpha}}$$

$$WR = 1 - \alpha^{N+1} \quad (3.17)$$

Với $\alpha = 7/8$ như trong TCP, với các giá trị khác nhau của N , ta tính được giá trị WR như bảng sau:

BẢNG 3.1. Quan hệ số lượng mẫu và tỷ lệ trên tổng tỷ trọng

N	5	10	15	20	25
WR	55%	77%	88%	94%	97%

Ta nhận thấy, $WR = 77\%$ khi $N = 10$, tức là 10 giá trị RTT đo được gần nhất, thời sự nhất, đóng góp 77% trên tổng số trọng số của tất cả các giá trị RTT. Khi N càng

nhỏ, tỷ lệ trọng số trên tổng trọng số của N mẫu gần nhất càng nhỏ. Ngược lại khi số lượng mẫu N lớn, WR càng lớn.

Do trong TCP giá trị α là cố định, công thức (3.14) truy hồi đến tất cả các giá trị mẫu, bởi vậy ta có thể thấy các giá trị mẫu đo được trong quá khứ, kể từ thời điểm bắt đầu phiên làm việc hoặc trước khi xảy ra hiện tượng đứt kết nối, vẫn góp phần vào giá trị của RTT. Điều này cho phép phát biểu, cách tính RTT của TCP là không phù hợp cho môi trường không dây, vốn có nhiều tham số môi trường luôn biến thiên, và không cần số lượng mẫu N lớn.

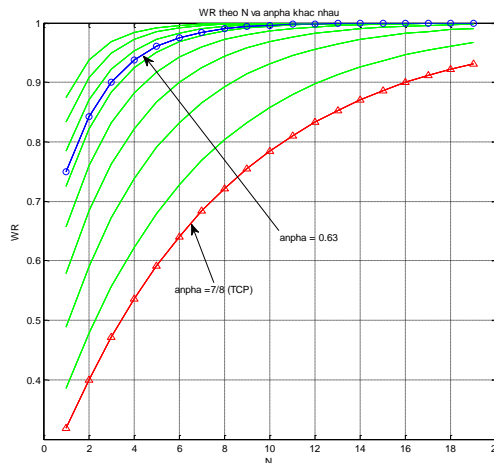
Từ kết quả trên, cho thấy rằng, thay đổi giá trị α , hoàn toàn có thể nhận được số lượng mẫu gần nhất ít hơn, song chiếm tỷ trọng lớn hơn, trong giá trị của tham số RTT. Điều này đồng nghĩa việc giao thức sẽ phản ứng nhanh hơn đối với môi trường, khi các giá trị mẫu xa trong quá khứ, không có ý nghĩa nhiều trong việc tính ra giá trị mới của các tham số.

3.2.2 Đề xuất phương pháp tính RTT phù hợp môi trường không dây

Với môi trường truyền thông không dây, các giá trị trong quá khứ không nên có vai trò lớn trong việc tính giá trị RTT, do đặc điểm môi trường này khiến các tham số thăng giáng liên tục. Do vậy, với công thức tính RTT như ở (3.14) với $\alpha = 7/8$ hay ~ 0.9 , giá trị mẫu SRTT mới chỉ đóng góp 10% tỷ trọng vào giá trị trung bình của RTT, và điều này rõ ràng là không hợp lý.

Nghiên cứu quan hệ của WR và giá trị α , N trong công thức (3.17) ta có đồ thị như minh họa (Hình 3.5), có thể thấy với $\alpha = 7/8$ của TCP, WR cần nhiều mẫu hơn, nhiều thời gian hơn để đạt được cùng một giá trị tổng trọng số.

Để RTT bắt kịp sự biến đổi nhanh của môi trường, ta cần nâng cao tỷ lệ trọng số của N trạng thái gần nhất. Ví dụ với $N = 4$, và mong muốn tỷ lệ trọng số $WR \geq 90\%$, ta tính được $\alpha \leq 0.63$. Trong khi đó với TCP truyền thống, cần số mẫu là 15. Tuy nhiên, α nhỏ quá cũng không cần thiết, vì lúc đó ý nghĩa của các giá trị mẫu trong quá khứ gần cũng không được tính đến.



Hình 3.5. Mô phỏng quan hệ WR theo giá trị α , N

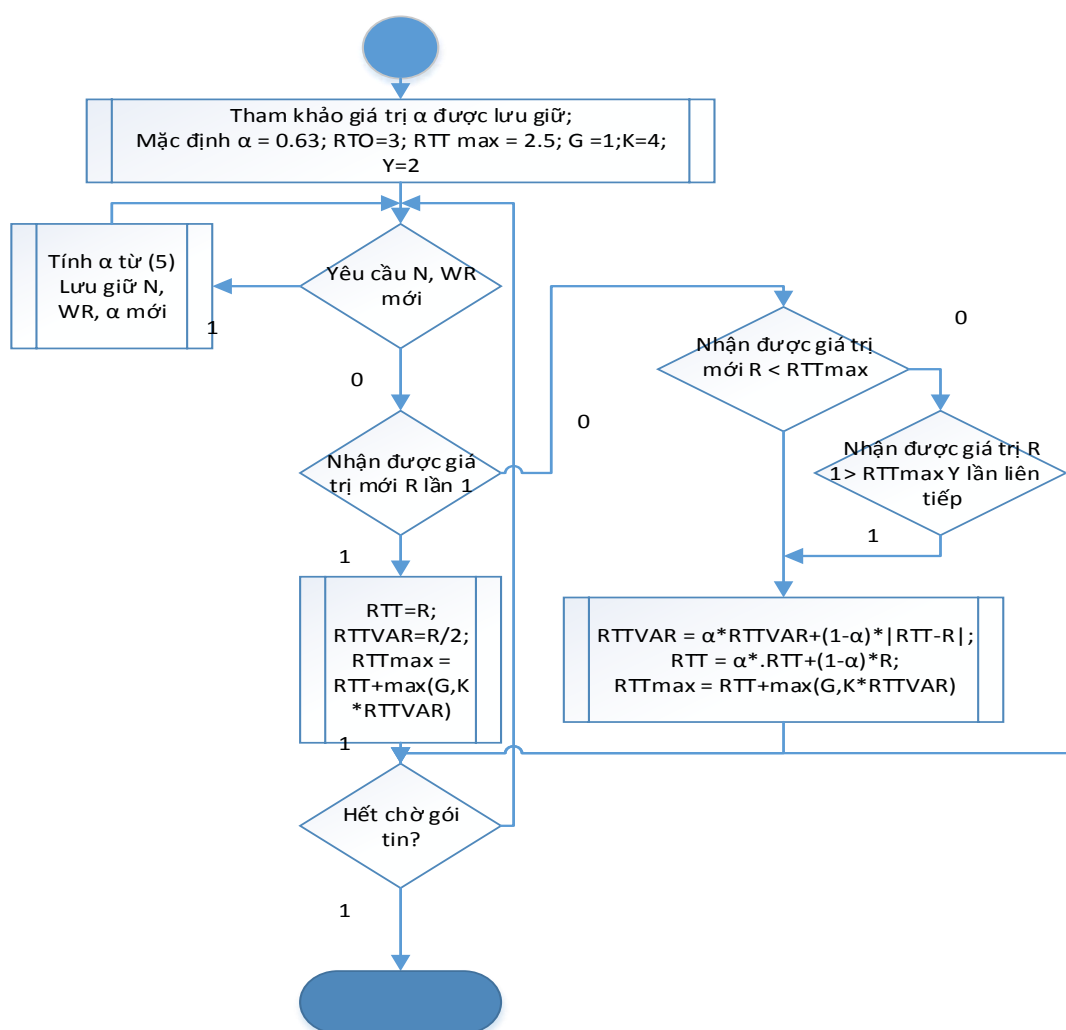
Với cơ sở trên, luận án đề xuất chọn $\alpha = 0.63$, là giá trị mặc định. Có thể tính toán α tăng giảm, theo yêu cầu của người sử dụng về mức độ đáp ứng nhanh với môi trường, thông qua giá trị tỷ lệ trọng số cần đạt được (WR) cho số lượng mẫu trong quá khứ cần quan tâm (N), có thể nhận và lưu giữ yêu cầu của người sử dụng trong bộ phận lược sử, để dùng cho lần khởi tạo tiếp theo.

Thuật toán thực hiện phép tính RTT cần xử lý những biến thiên thái quá của giá trị đo mới của RTT. Khi áp dụng công thức (3.14) để có giá trị α phù hợp, theo kỳ vọng về số lượng mẫu có tỷ trọng lớn theo công thức (3.17) cần tham chiếu giá trị quá khứ các yêu cầu của người sử dụng. Các giá trị mẫu SRTT của các gói tin phải gửi lại, sẽ không được tính, để loại bỏ các giá trị mẫu sai và quá nhỏ so với thực tế, do bên gửi có thể nhận được phản hồi cho gói tin đã gửi lần đầu, song lại lại nhằm là phản hồi cho gói tin vừa được gửi lại, như hình minh họa (Hình 3.4).

Tiếp đến, cần xác định giá trị nào của mẫu RTT được coi là quá cao. Nếu một mẫu RTT mới nằm trong khoảng được chấp nhận, thuật toán sẽ áp dụng công thức làm mịn với công thức (3.14). Những giá trị mẫu RTT ngoài khoảng chấp nhận, sẽ bị loại bỏ một số lần nhất định. Nếu vẫn nhận được các giá trị mẫu quá lớn, sau số lần định trước, giá trị này sẽ được tính vào RTT theo công thức (3.14). Kết quả là với

RTT được làm mịn theo cách này, giao thức truyền thông vẫn có thể hội tụ nhanh về trạng thái của mạng.

Luận án đề xuất thuật toán EWMA RTT, sử dụng cơ chế lưu trữ những yêu cầu của người sử dụng về số lượng N trạng thái trong lược sử, và có lưu đồ như hình minh họa (Hình 3.6). Ở trạng thái khởi tạo, các tham số được tham khảo ở lần hoạt động trước đó. Nếu chưa lưu giá trị nào, các giá trị mặc định được dùng. Các giá trị RTO, RTTmax lần lượt là thời gian hết hạn chờ đợi, giá trị cực đại của thời gian khứ hồi gói tin, lấy các giá trị khởi tạo tương ứng là 3s, 2.5s, tương tự trong thiết kế của TCP [5].



Hình 3.6 Lưu đồ thuật toán EWMA RTT

Khi có gói tin phản hồi, xác định được giá trị mẫu thời gian quay vòng R, tùy thuộc gói tin đây là gói tin đầu tiên phản hồi hay không, các giá trị RTT, RTTmax được tính toán phù hợp. Với giá trị R quá lớn, thuật toán sẽ bỏ qua 1 lần, trước khi coi đây là giá trị đáng tin cậy.

Với thuật toán EWMA RTT, giá trị RTT thu được sẽ phản ánh rõ hơn biến thiên của môi trường không dây. Đề xuất này cho phép giao thức điều khiển mềm dẻo, tùy theo yêu cầu và điều kiện môi trường và mong muốn của người sử dụng.

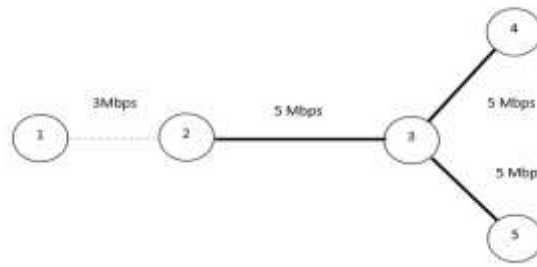
Tính toán mô phỏng

Để xây dựng mô phỏng để so sánh thuật toán RTT điều chỉnh và phương pháp của TCP, Luận án sử dụng môi trường mô phỏng NS cho một trạm di động sử dụng TCP nối với trạm gốc bằng đường truyền 3Mbps mô phỏng cho mạng không dây (Hình 3.7).

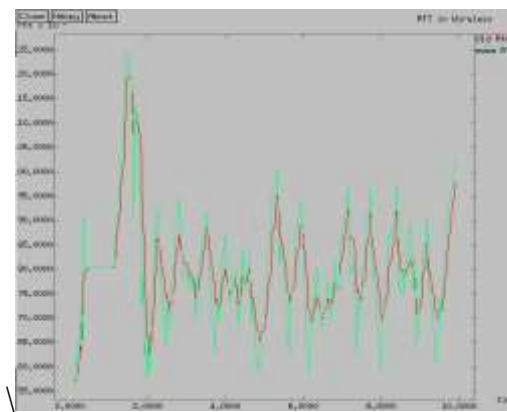
Các gói tin TCP với kích thước 1000 byte được gửi từ nút 1 là đến các đầu cuối nhận tin là nút số 4 và 5, thời gian khứ hồi gói tin là 15 ms. Đường truyền mô phỏng cho mạng có dây có băng thông là 5Mbps. Mô phỏng cho tình huống mất gói tin, đường truyền không dây bị gián đoạn từ thời điểm [1.4- 1.45] và từ [1.6-1.615].

EWMA được áp dụng tính RTT cho luồng dữ liệu từ nút 1 đến 4 với số lượng lược sử $N = 5$, $\alpha = 0.6$ để so sánh với giá trị RTT của luồng tin từ nút 1 đến 5.

Theo dõi khoảng thời gian từ khi gửi gói tin TCP thứ i được gửi và nhận được gói tin ACK tương ứng, ta có kết quả mẫu SRTT thứ i . Với trường hợp gói tin TCP bị mất và phải gửi lại, SRTT sẽ không được tham gia vào quá trình tính toán RTT. So sánh phương pháp tính RTT cổ điển của TCP và phương pháp EWMA RTT, ta nhận thấy giá trị RTT biến động nhanh hơn, bám sát hơn với đặc tính đường truyền. Do vậy, ứng dụng EWMA RTT vào môi trường không dây sẽ hiệu quả hơn.



a)



b)

Hình 3.7 Giá trị RTT đo bằng phương pháp EWMA RTT

Như vậy, có thể nói với việc trong mạng không dây, nguyên nhân mất gói tin có thể do nhiều lý do đặc biệt như nguồn thu phát bị che khuất, tán xạ, phản xạ, nhiễu loạn... việc tính toán chính xác các tham số truyền thông như thời gian trễ khứ hồi RTT là rất quan trọng, do sẽ giúp phát hiện có lỗi và giúp giao thức phản ứng phù hợp, chính xác hơn.

Sử dụng phương pháp hàm thống kê phân tích những điểm chưa phù hợp trong cách tính tham số RTT trong họ giao thức TCP, luận án đã tìm thấy mối liên hệ giữa trọng số của các giá trị mẫu thời gian trễ khứ hồi và hệ số làm mịn trong TCP, góp phần lý giải hiệu năng thấp của TCP trong môi trường không dây, do đã chọn hệ số làm mịn. Luận án đề xuất thuật toán điều khiển cho phép điều khiển mềm dẻo hơn với các hệ số làm mịn tùy theo yêu cầu của người sử dụng.

3.3 Giao thức tự thích nghi họ TCP cho môi trường không dây WRCAP

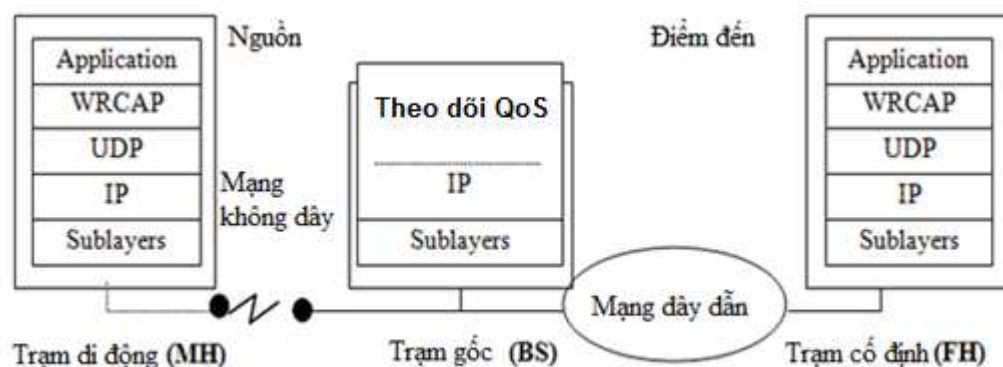
Luận án áp dụng các nghiên cứu về mô hình điều khiển thích nghi đã đề xuất vào triển khai giao thức WRCAP trên nền giao thức UDP trong môi trường mô phỏng NS-2 [31], phiên bản 2.1b9 để thực hiện các thí nghiệm mô phỏng. Kết quả cho thấy, ***so với TCP Reno, WRCAP cải thiện thông lượng của mạng không dây, đặc biệt với tình huống đường truyền không dây bị gián đoạn.***

WRCAP được gọi là họ TCP do đây là giao thức đầu cuối – đầu cuối hoạt động tương tự cách hoạt động để quản lý tắc nghẽn và ổn định trên mạng. Lý do sử dụng UDP làm nền tảng phát triển giao thức mới, vì UDP là giao thức đơn giản hơn trong việc cải tiến và thử nghiệm mà không cần mất quá nhiều công. Các chức năng của giao thức được xây dựng theo hướng dẫn thiết kế giao thức [35], để phù hợp với cơ chế điều khiển ECIMD.

Giao thức điều chỉnh tốc độ tự thích nghi WRCAP bao gồm bộ phận *Theo dõi QoS* được triển khai tại trạm gốc và các modul phần mềm là các thực thể được cài đặt tại trạm đầu cuối, trong kịch bản là nguồn phát và nguồn thu (Hình 3.8)

Modul phần mềm tại bên phát và bên thu được cải tiến trên nền giao thức gốc UDP. Bộ phận *Theo dõi QoS* được minh họa là nằm phía trên tầng IP, do cần sử dụng một số dịch vụ của của tầng IP. Tuy nhiên các modul này được triển khai các chức năng điều khiển QoS. Chúng ta sẽ mô tả các chức năng mở rộng trong phần tiếp theo.

Bộ phận *Theo dõi QoS* là cần thiết, do đường truyền không dây có nhiều lỗi, bởi vậy có yêu cầu ngặt nghèo đối với việc đảm bảo chất lượng dịch vụ, và cần được quan tâm đặc biệt. Khi triển khai *Theo dõi QoS* tại trạm gốc, sẽ giúp trạm di động nhận biết nhanh nhất các lỗi phát sinh trên đường truyền không dây, để đưa ra các hành động thích hợp khác. Ngoài ra *Theo dõi QoS* có một chức năng quan trọng nữa là đo băng thông khả dụng, triển khai theo công thức (2.7). Trong thực tế, có thể sửa đổi phần mềm điều khiển hoạt động của tầng IP tại trạm gốc để đưa thêm bộ phận *Theo dõi QoS* như đã nói. Những điều chỉnh này sẽ không làm ảnh hưởng đến hoạt động và ý nghĩa đầu cuối - đầu cuối của các ứng dụng.



Hình 3.8 Nguyên lý hoạt động của WRCAP

WRCAP là giao thức điều khiển tốc độ phù hợp trên cơ sở mô hình điều khiển thích nghi. Các hàm đánh giá, điều khiển và quyết định được thực hiện tại cả hai bên, bên gửi và bên nhận tin. WRCAP điều chỉnh tốc độ gửi tin của bên gửi dựa trên thông tin phản hồi từ bên nhận. WRCAP gửi các gói tin dữ liệu có các số tuần tự và tem thời gian, chỉ thời gian mà gói tin đó đã được gửi đi. Tại phía nhận, tỷ lệ lỗi và trễ sẽ được đánh giá. Các báo cáo sau đó được gửi lại cho bên gửi vào mỗi khoảng thời gian bằng thời gian khứ hồi gói tin. Nếu không có lỗi nào được phát hiện, bên gửi sẽ tăng tốc độ gửi tin một cách tuần tự theo ECIMD. Ngược lại, bên gửi sẽ giảm tốc độ gửi theo cấp số nhân.

Với WRCAP, quá trình gửi các gói tin được chia làm hai chiều riêng rẽ: khi truyền tin từ trạm di động đến trạm cố định và khi truyền tin từ trạm cố định đến trạm di động

Truyền tin từ trạm di động đến trạm cố định:

Bộ phận tại BS *Theo dõi QoS* sẽ đánh giá băng thông trên đoạn đường truyền không dây R1, đưa thông tin về băng thông này vào trong phần nhận gói tin và gửi gói tin đến bên nhận. Tại phía bên nhận, băng thông của toàn bộ kết nối sẽ được đo với giá trị R2. Do đó, tốc độ khởi tạo cho kết nối sẽ được chọn là $R_{init} = \min(R1, R2)$ và được gửi ngược trở lại bên gửi tin, thông qua gói tin ACK.

$$R_{est} = \frac{L}{T_{qs}} \left[1 - e^{-\frac{\alpha \cdot t}{1 + \alpha \cdot T_{qs}}} \right] + R_0 \cdot e^{-\frac{\alpha \cdot t}{1 + \alpha \cdot T_{qs}}}, 0 \leq t \leq T_{qs}$$

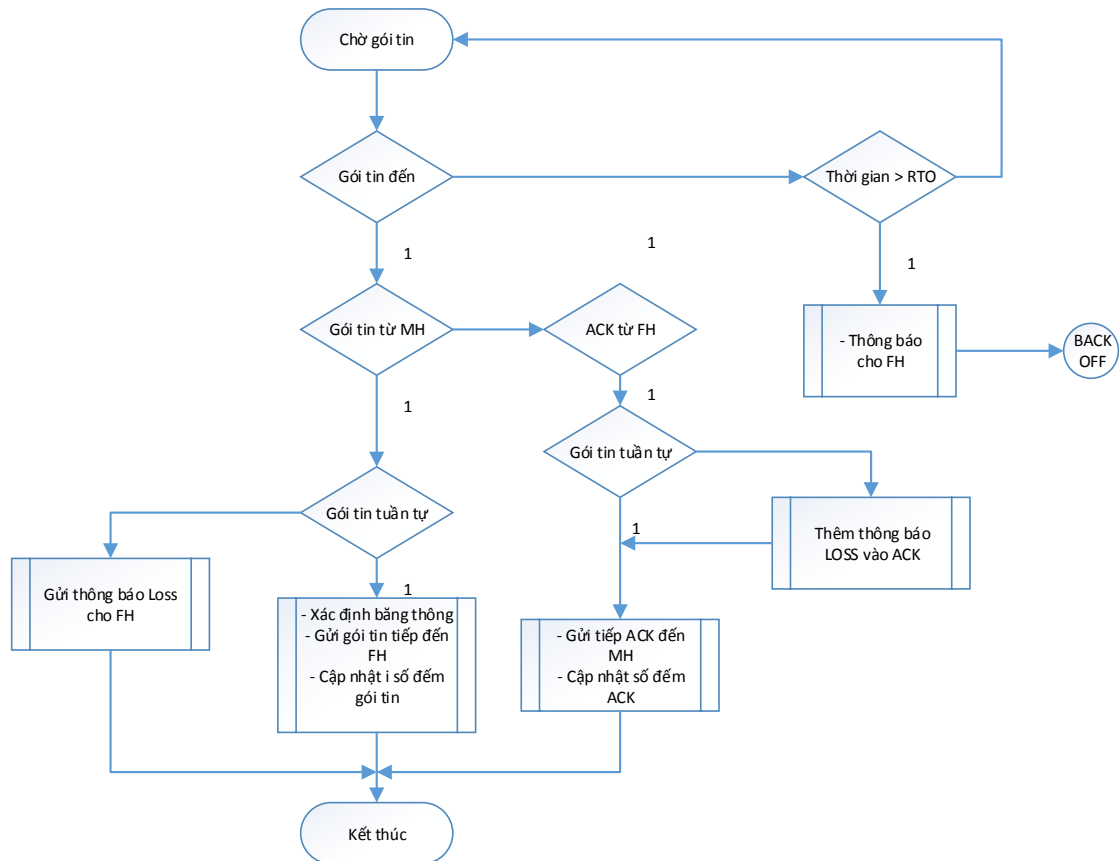
Tốc độ đo được R_{est} được cập nhật mỗi khi có một gói tin nhận được, bên gửi sẽ gửi các gói tin với số thứ tự, giúp bộ phận *Theo dõi QoS* có thể theo dõi để phát hiện sự ngắt quãng trong các số hiệu tuần tự và phát hiện ra lỗi của đường truyền không dây. Nếu thời gian quá dài mà không nhận được gói tin, thì cũng coi là đã xảy ra lỗi mất gói tin. Khi lỗi được phát hiện, *Theo dõi QoS* sẽ gửi thông báo bằng gói tin rỗng đến nguồn phát/ nhận WRCAP. Luận án nhấn mạnh đến việc gửi kèm thông báo trong phần nhãn các gói tin hiện cần gửi, do đó không cần thêm băng thông cho quá trình thông tin này. Mặt khác, nguồn phát WRCAP khởi động trạng thái backoff khi phát hiện lỗi không dây, tạm ngừng gửi tin và cố gắng liên lạc với *Theo dõi QoS* bằng cách thăm dò kênh không dây theo chu kỳ. Khi kênh không dây trở lại bình thường, một phản hồi tích cực sẽ được trả lời từ *Theo dõi QoS*.

Chức năng của *Theo dõi QoS* sẽ bao gồm khả năng quản lý hàng đợi một cách công bằng giữa các luồng thông tin. Không cần đặt trước nguồn lực của hệ thống hay của mạng. Trọng số của mỗi kết nối có thể được đặt sao cho tốc độ gửi tin của kết nối phải đảm bảo trong khoảng cho phép, để chất lượng của kết nối đó được đảm bảo. Ngoài ra, trong thời gian bị lỗi kênh không dây, kết nối sẽ được tính toán và đền bù sau khi kênh được kết nối trở lại.

Thuật toán triển khai trong bộ phận *Theo dõi QoS* trong quá trình này được mô tả như trong Hình 3.9.

Truyền tin từ trạm cố định đến trạm di động

Để nhận biết nhanh chóng trạng thái của kênh truyền không dây, Luận án đề xuất để bên nhận của giao thức WRCAP sẽ gửi phản hồi cho mọi gói tin nhận được. Không những vậy, kích cỡ của gói tin ACK nhỏ khi so với gói tin dữ liệu, do đó tải của phần thông tin điều khiển được duy trì ở mức khá nhỏ. Việc đo đặc bằng thông được thực hiện giống như trong trường hợp hướng gửi tin theo chiều ngược lại và bên gửi có thể căn cứ vào đó để khởi tạo giá trị gửi ban đầu.



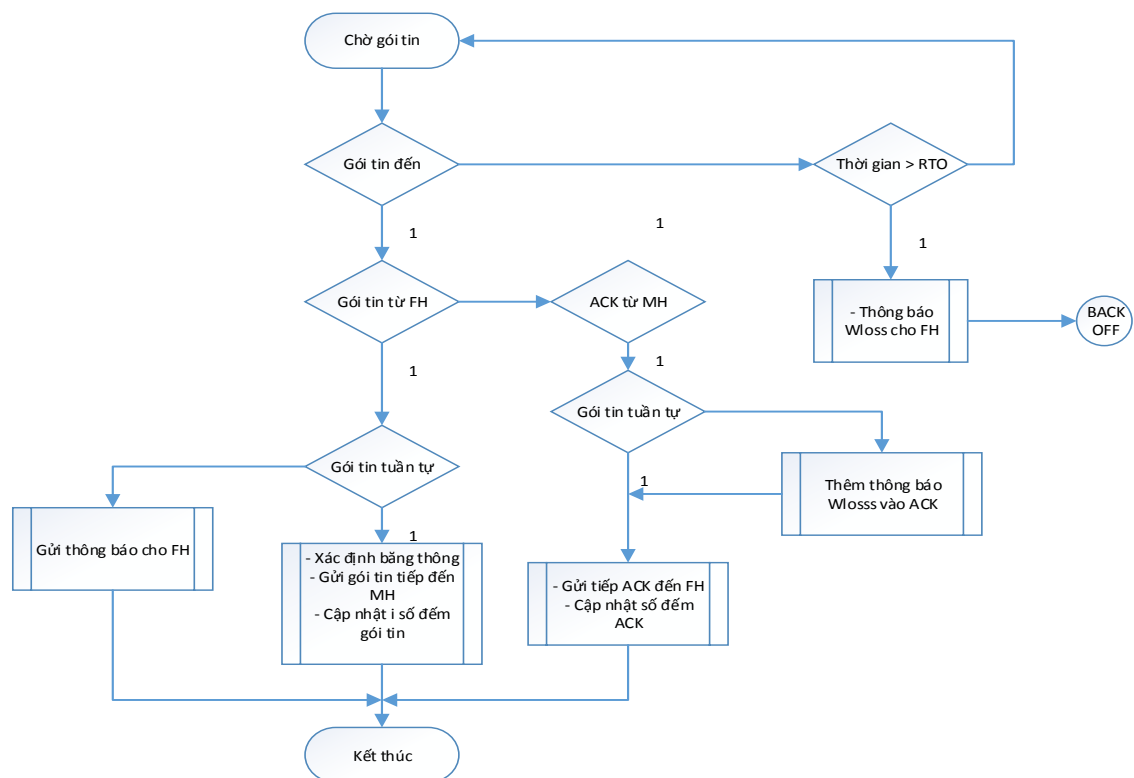
Hình 3.9 Theo dõi QoS: Di động - Cố định

Bằng việc theo dõi số tuần tự của các gói tin, bộ phận *Theo dõi QoS* phát hiện các lỗi do tắc nghẽn trong đoạn đường từ trạm cố định đi đến trạm gốc, nếu thấy xuất hiện sự gián đoạn trong các gói tin tuần tự. Trong trường hợp đó, bộ phận *Theo dõi QoS* gửi một thông điệp thông báo về mất mát cho bên gửi tại trạm cố định. Thông điệp này có thể được gắn kèm trong các gói tin dữ liệu, đang trên đường từ trạm di động đi đến trạm cố định. Sau khi gửi các gói tin cần gửi đến trạm di động, *Theo dõi QoS* sẽ chờ gói tin phản hồi từ phía trạm di động. Nếu quá thời gian chờ hoặc có xuất hiện gián đoạn trong số thứ tự của các gói tin phản hồi ACK được gửi từ trạm di động, trạng thái kênh không dây sẽ được cho là xấu. Bộ phận *Theo dõi QoS* sẽ gửi thông điệp thông báo trạng thái cho bên gửi tin tại trạm cố định, WRCAP tại trạm cố

định sẽ hoãn gửi tin và chuyển sang trạng thái backoff để chờ phản hồi tích cực từ bộ phận *Theo dõi QoS*.

Chức năng của bộ phận *Theo dõi QoS* có thể mở rộng để quản lý vùng nhớ đệm và chức năng uỷ thác (proxy) bậc cao tại trạm gốc. Chức năng proxy có thể làm thực thể trung gian giữa bên gửi và bên nhận, tự động duy trì danh sách các gói tin chưa nhận được phản hồi tích cực, tự động gửi lại các gói tin này khi có lỗi mất gói tin xảy ra. Tùy theo kết quả của việc đo băng thông của kênh không dây, các gói tin ít ưu tiên (ví dụ các gói tin video có mức ưu tiên thấp) có thể được chủ động loại bỏ. Thông tin chuyển mã của dữ liệu (ví dụ các mẫu, kích cỡ hình ảnh, độ phân giải, v.v.) sẽ được sử dụng hiệu quả, nhờ đó kênh không dây sẽ không bị quá tải khi vẫn duy trì các kết nối có chất lượng dịch vụ chấp nhận được.

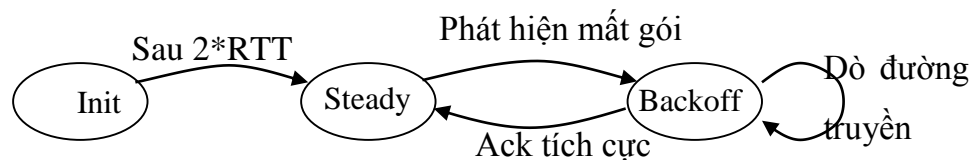
Thuật toán của bộ phận *Theo dõi QoS* trong quá trình truyền tin từ trạm cố định đến trạm di động có thể được thực hiện như Hình 3.10



Hình 3.10 Theo dõi QoS: Cố định - Di động

Các trạng thái của WRCAP

Như đề cập ở trên, nguồn phát WRCAP gửi các gói tin có đánh số hiệu tuần tự, còn nguồn thu WRCAP sẽ phản hồi mỗi gói tin nhận được bằng một gói tin ACK. Mỗi gói tin ACK chứa thông tin về số hiệu của gói tin đã nhận được. Như Hình 3.11 WRCAP sẽ hoạt động với ba trạng thái: Initial, steady và backoff.



Hình 3.11 Mô hình các trạng thái của giao thức tự thích nghi WRCAP

- **Trạng thái Initial (khởi tạo)**

Trạng thái này bắt đầu khi thiết lập một kết nối mới và kéo dài 2 chu kỳ RTT, là thời gian khứ hồi gói tin. Chất lượng của kết nối đa phương tiện cần nằm trong khoảng cho phép (Q_{min} ; Q_{max}). Q_{min} là chất lượng nhỏ nhất có thể chấp nhận được. Q_{max} là chất lượng mong muốn của kết nối. Vượt quá giá trị này là không cần thiết do phần kết quả thu được không đáng kể, song có thể gây thêm tải cho hệ thống. Tương ứng với khoảng chất lượng dịch vụ yêu cầu, mỗi kết nối sẽ cần được duy trì với một khoảng băng thông để truyền tin tương ứng, là $[R_{min}, R_{max}]$. Do đó, WRCAP cần khởi tạo và gửi tin với tốc độ ban đầu không được nhỏ hơn giá trị R_{min} .

Giá trị tốc độ gửi tin ban đầu sẽ được chọn theo tại bên nhận, do bên nhận đo được băng thông khả dụng nhanh hơn: $R_{init} = \min(R1, R2)$ với $R1$ là giá trị băng thông trong đoạn không dây, đo được bởi bộ phận *Theo dõi QoS* tại trạm gốc, $R2$ là tốc độ do bên thu đo được với công thức (2.7). Bên nhận WRCAP gửi lại thông tin về tốc độ được chọn cho bên gửi thông qua các gói tin ACK trong khoảng thời gian 2 lần chu kỳ gói tin khứ hồi RTT.

- **Trạng thái Steady**

Trong trạng thái này, WRCAP tại bên nhận đo băng thông đường truyền liên tục trong quá trình kết nối, theo công thức (). Hơn nữa, RTT được điều chỉnh theo EWMA. Các thông số này sẽ được gửi về WRCAP ở bên gửi thông qua gói tin ACK.

Khi phát hiện sự gián đoạn trong chuỗi các gói tin tuần tự, bên nhận WRCAP sẽ lấy thêm thông tin được gửi từ bộ phận *Theo dõi QoS* đã gắn vào phần mào đầu trong mỗi gói tin, để phân biệt, lỗi xảy ra là thuộc loại gì, vì lý do tắc nghẽn hay do mất gói tin trong phần đường truyền không dây. Các chỉ thị về mất gói tin được gửi thông qua gói tin ACK đến bên gửi của giao thức WRCAP.

Tại bên gửi, WRCAP sử dụng thông tin từ các gói tin ACK để đo băng thông khả dụng và biết được trạng thái của mạng, cũng như tính toán và cập nhật giá trị thời gian khứ hồi, để phục vụ các quyết định tiếp sau của nó.

Dựa trên giá trị RTT đã được cập nhật và thông tin trong gói ACK, WRCAP sẽ quyết định:

- Tăng tốc độ gửi với tốc độ R_{new} , cho mỗi khoảng thời gian RTT, nếu không có lỗi xảy ra

$$R_{new} = \min(R_{old} + \Delta R, R_{max}) \quad (3.15)$$

Trong đó ΔR là thành phần hàm mũ, giúp tốc độ gửi tiệm cận đến giá trị R_{max} nhanh.

- Giảm tốc độ, khi gặp lỗi do tắc nghẽn

$$R_{new} = \max(R_{old} / 2, R_{min}) \quad (3.16)$$

- Chuyển sang chế độ backoff, khi lỗi do mạng không dây được phát hiện.

Lỗi hết giờ (time out) xảy ra, khi trong khoảng thời gian RTO không nhận được gói tin ACK nào, cũng sẽ được coi là lỗi do mạng không dây. Điều này có thể dự đoán chính xác hơn, khi trong gói tin ACK có thông tin phần còn trống trong vùng nhớ đệm.

WRCAP sử dụng thuật toán để phục hồi nhanh chóng tốc độ gửi tin, ngay khi kết nối được phục hồi, sau khi bị mất gói tin do đường truyền internet. Hơn nữa, do cần đáp ứng với yêu cầu tốc độ gửi trong khoảng (R_{min} ; R_{max}) như trong công thức trên, tốc độ hội tụ đến giá trị thực của băng thông của WRCAP sẽ còn tốt hơn.

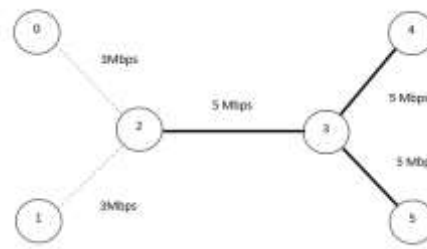
Trong tình huống băng thông được xác định là thấp hơn giá trị tốc độ gửi tối thiểu, chất lượng dịch vụ được coi là không đảm bảo. Với tình huống này, WRCAP phải thông báo cho ứng dụng, để ứng dụng điều chỉnh chất lượng dịch vụ mong muốn, hoặc điều chỉnh lượng dữ liệu cần gửi để phù hợp với đường truyền (tăng giảm mẫu, bỏ bớt các khung hình mở rộng, thay đổi codec...) Một khả năng nữa là sử dụng proxy, là phần chức năng có thể mở rộng của *Theo dõi QoS*, để tự động lưu các gói tin chưa nhận được phản hồi, để khi có điều kiện, có thể gửi lại nhanh chóng, như đã nhắc đến ở trên.

- **Trạng thái Backoff**

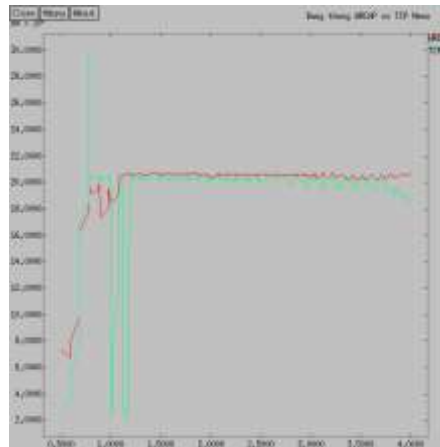
WRCAP sẽ ở trạng thái này khi phát hiện ra lỗi do mạng không dây gây ra. Các việc truyền tin sẽ tạm thời ngừng lại, WRCAP sẽ định kỳ gửi các gói tin nhỏ gọi là probe để nghe ngóng phản hồi. Việc sử dụng các gói tin probe có kích thước nhỏ sẽ hạn chế việc hoạt động một cách vô ích, tiết kiệm năng lượng cho thiết bị di động. WRCAP trở về trạng thái Steady, khi nhận được phản hồi tích cực từ gói tin Probe nào đó, từ bộ phận *Theo dõi QoS*.

- **Tính toán mô phỏng**

Để xây dựng mô phỏng cho giao thức WRCAP Luận án sử dụng môi trường mô phỏng NS cho một trạm di động sử dụng WRCAP và một trạm sử dụng TCP nối với trạm gốc bằng đường truyền 3Mbps mô phỏng cho mạng không dây (H.3.12a). Các gói tin được gửi với kích thước 1000 byte, thời gian xử lý gói tin là 15 ms. Đường truyền còn lại mô phỏng cho mạng có dây, băng thông là 5Mbps. Mô phỏng cho tình huống mất gói tin, đường truyền không dây bị gián đoạn từ thời điểm [1 - 1.05] và từ [1.2-1.215].



(a)



(b)

Hình 3.12 Thông lượng nguồn TCP và WRCAP

Như trong hình 3.12b cho thấy, WRCAP hoạt động tốt trong hoàn cảnh có lỗi do tắc nghẽn và lỗi do mạng không dây gây ra. Tại thời điểm 1.0 và 1.2 đường truyền gián đoạn WRCAP chuyển sang trạng thái backoff và tạm dừng quá trình truyền. Ngay sau khi đường truyền được khôi phục, tốc độ cũ ngay lập tức được hồi phục, mang lại tốc độ cao cho kết nối. Kết quả thể hiện rất rõ, với lỗi xảy ra trong đường truyền không dây, thông lượng WRCAP đáp ứng tốt hơn TCP.

3.4 Kết luận chương III

Trong chương III đã đề xuất mô hình điều khiển thích nghi sử dụng cơ chế ECIMD, thay thế cho AIMD của TCP. Nghiên cứu cơ chế ECIMD trong tình huống việc điều khiển giá trị cửa sổ khi có lỗi đơn cho thấy cơ chế này mang lại thông lượng tốt hơn so với AIMD.

Công thức tính thông lượng với ECIMD có thể suy biến về công thức thông lượng của TCP đã và đang được áp dụng hiện nay, cho thấy ECIMD là mô hình tổng quát của AIMD. Đây là cơ sở cho đề xuất chọn các giá trị mới cho các hệ số trong công thức làm mịn truyền thống của TCP. Kết quả mô phỏng cho thấy với một số giá trị mới của hệ số, thông lượng của kết nối đã tăng lên đáng kể.

Chương này cũng đã đề xuất phương pháp tính giá trị thời gian khứ hồi gói tin, dựa trên phân tích tổng trọng số của N mẫu gần nhất. Điều này đặc biệt quan trọng với môi trường không dây hay biến đổi, nên chỉ cần quan tâm đến sự ảnh hưởng của một số giá trị gần nhất.

Mô hình đề xuất đã được áp dụng để xây dựng một giao thức họ TCP là WRCAP và thử nghiệm mô phỏng trong môi trường NS đạt hiệu suất cao hơn, có khả năng phát hiện, phân biệt và phòng lỗi hiệu quả hơn so với các kết quả nghiên cứu đang sử dụng hiện nay trong các giao thức họ TCP, khi chạy trên môi trường hỗn hợp, trong mô hình có trạm gốc và trạm di động.

KẾT LUẬN

Mục tiêu chính của luận án là nghiên cứu và đề xuất một mô hình điều khiển và tự thích nghi với môi trường vào điều khiển luồng và điều khiển tắc nghẽn trong mạng hỗn hợp cố định – di động. Luận án áp dụng mô hình này vào xây dựng giao thức WRCAP là một giao thức họ TCP. Mô hình thích nghi có thể xác định và dự đoán trạng thái môi trường, từ đó đáp ứng tốt hơn, mang lại thông lượng tốt hơn. Với những kết quả tính toán mô phỏng mô hình điều khiển thích nghi, luận án đã chứng minh khả năng điều khiển và tính khả thi của mô hình được đề xuất.

Các kết quả chính đạt được trong luận án là :

1. So sánh các cơ chế kiểm soát lỗi đầu cuối - đầu cuối. Kết quả so sánh cho thấy: việc phục hồi nhanh chóng trên đường truyền không dây được sử dụng trong giải pháp WRCAP là thích hợp và mang lại thông lượng cao cho hệ thống.
2. Đề xuất phương pháp ước lượng giá trị tổng nhu cầu băng thông, băng thông từng luồng, băng thông khả dụng từ phía nhận, mà không cần chờ gói tin phản hồi, giúp quá trình đo đạc và dự đoán tham số này được thực hiện nhanh chóng, ảnh hưởng tốt tới quá trình hoạt động của giao thức.
3. Đề xuất cơ chế điều khiển thích nghi theo phương pháp ECIMD mới, thay cho cơ chế AIMD được dùng trong TCP, nhằm cải thiện tốc độ hội tụ của giao thức, đưa ra cách ứng xử phù hợp khi gặp lỗi và phục hồi sau lỗi mất gói tin. Cơ chế này đảm bảo công bằng với các luồng tin sử dụng giao thức họ TCP, giúp tránh tắc nghẽn trên mạng.
4. Đề xuất phương pháp tính giá trị trung bình thống kê của RTT mới, phù hợp với môi trường hay biến đổi, và chứng minh ưu điểm của phương pháp này với phương pháp tính tổng trọng số của tập mẫu gần nhất.

5. Thực hiện mô phỏng để chứng minh ưu điểm và tính khả thi của mô hình điều khiển thích nghi đã đề xuất.

Những kết quả thu được nói trên trong luận án có thể cho phép kết luận rằng mô hình điều khiển thích nghi và ứng dụng của nó là giao thức WRCAP, hoàn toàn có thể triển khai, áp dụng với mô hình mạng có kết nối phức tạp, với các thiết bị di động là một trong hai nguồn thu / phát, sử dụng kết nối không dây. Trong mô hình mạng có kết nối phức tạp này, đoạn mạng giữa thiết bị di động (máy tính xách tay, smartphone, máy tính bảng..) và trạm gốc (BS hoặc AP) thường thiếu ổn định.

DANH SÁCH CÁC CÔNG TRÌNH ĐÃ CÔNG BỐ

- [1] Vu Tat Thanh, “Một mô hình mới dự đoán xác suất lỗi gói tin trong môi trường không dây”, Báo Cáo Khoa Học Kỷ niệm 25 năm thành lập Viện VIELINA, 2010.
- [2] Nguyen Hong Vu, Vu Tat Thanh, “Về một phương pháp mới xác định băng thông và trạng thái đường truyền”, Hội thảo Điện tử - Truyền thông – An toàn thông tin, 2012.
- [3] Vu Tat Thanh, Nguyen Hong Vu - “A new method to estimate the current RTT calculation algorithm”, REV 2013 Hanoi – 17 Dec 2013.

TÀI LIỆU THAM KHẢO

- [1] Hoang Dang Hai , “*Quality of Service Control in the Mobile Wireless Environment*”, Peter Lang Publishing, Jan 2003.
- [2] S. Floyd, M.Handley, J.Padhya, “A Comparison of Equation-Based and AIMD Congestion Control”, <http://www.aciri.org/tfrc/aimd.ps>, May, 2000.
- [3] V. Jaconson, “Congestion Avoidance and Control”, *ACM SIGCOMM’88*, 1988
- [4] S.Floyd, K. Fall. “Promoting the Use of End-to-End Congestion Control in the Internet”, *IEEE/ACM Transactions on Networking*, Vol.7, No.4, pp.458-472, Aug 1999.
- [5] W. Steven, “TCP/IP Illustration, Volume I: The Protocols”, Addison Wesley 1997.
- [6] R.Morris, “Scalable TCP Congestion Control”, in *Proc. Of IEEE INFOCOM* 2000.
- [7] Vu Tat Thanh, D. Reschke, W. Horn - Dynamic Packet Size Mechanism (DPSM) for Multimedia in Wireless networks – *MIK, Germany*, 2002.
- [8] A. Bakre and B. R. Badrinath (1995), “Handoff and system support for indirect TCP/IP”, *Proc. 2nd Usenix Symp. Mobile and Location-Independent Computing*, Apr. 1995.
- [9] A. Bakre and B. R. Badrinath (1995), “I-TCP: Indirect TCP for Mobile hosts”, *IEEE 1995*, 1063-6927/95.
- [10] A. Bakre and B. R. Badrinath (1997), “Implementation and Performance Evaluation of Indirect TCP”, *IEEE Transaction on Computer*, 46(3), March 1997.
- [11] Amit Bhargava, James F. Kurose, D. Towsley and G. Vanleemput , “Performance Comparison of Error Control Schemes in High-Speed Computer Communication Networks”, *IEEE Journal on Selected Area in Communications*, Dec. 1988
- [12] Andrew S. Tannenbaum, Computer Networks, Prentice Hall, New Jersey, 1996.

- [13] Bikram S. Bakshi, P. Krishna, N. H. Vaidya, D. K. Pradhan, "Improving performance of TCP over wireless networks", *Texas A&M University Technical Report*, TR-96-014, May 1996.
- [14] Cerf V., R. Kahn, "A Protocol for Packet Network Intercommunication", *IEEE Transactions on Communications*, 22(5), 637-648, May 1974.
- [15] H. Balakrishnan, S. Seshan, and R. H. Katz (1995), "Improving reliable transport and handoff performance in cellular wireless networks", *ACM Wireless Networks*, 1, Dec. 1995.
- [16] H. Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan, and Randy H. Katz (1997), "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", *IEEE/ACM TRANSACTIONS ON NETWORKING*, Dec. 1997.
- [17] H. Chaskar, T. V. Lakshman, U. Madhow (1996), "On the Design of Interfaces for TCP/IP over Wireless", IEEE. 1996.
- [18] Kahn, Phil; Craig Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols", *ACM SIGCOMM '87*. pp. 2–7, 2009.
- [19] R. Yavatkar and N. Bhagwat (1994), "Improving end-to-end performance of TCP over mobile internetworks", *Mobile 94, Workshop Mobile Computing Syst. Appl.*, Dec. 1994.
- [20] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control", RFC 2581, 1999.
- [21] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.
- [22] J. Rendon, F. Casadevall, D. Serarols, and J. L. Faner (2001), "Analysis of SNOOP TCP protocol in GPRS system", *IEEE 2001*, 0-7803-6728-6/01.
- [23] Van Jacobson, Michael J. Karels. "Congestion Avoidance and Control", *Proceedings of the Sigcomm '88 Symposium*, vol.18 (4), pp.314–329. Stanford, CA, August 1988.
- [24] Jacobson V. (1990), "Modified TCP Congestion Avoidance Algorithm," *Technical report*, 30 Apr. 1990, URL <ftp://ftp.ee.lbl.gov/email/vanj.90apr30.txt>.

- [25] Jain R., Ranmakrishnan, K. Chiu, D. M, "Congestion avoidance in computer networks with a connectionless network layer". *Tech. Rep. DEC-TR-506, Digital Equipment Corporation*, June 1. 1997.
- [26] John D. Spragins, Joseph L. Hammond, Krzysztof Pawlikowski, *Telecommunications: Protocols and Design*, Addition-Wesley, New York, 1991.
- [27] K. Fall and S. Floyd (1996), "Simulation-based Comparisons of Tahoe, Reno, and Sack TCP," *Computer Communication Review*, 1996.
- [28] K. Ratnam and I. Matta (1998), "Effect of Local Retransmission at Wireless Access Points on the Round Trip Time Estimation of TCP", *Proc. IEEE 31st Annual Simulation Symposium '98*, Boston, MA, April 1998.
- [29] Lawrence S. Brakmo, Sean W. O'Malley, Larry L. Peterson (1994), "TCP Vegas: New Techniques for Congestion Detection and Avoidance", *SIGCOMM 94*, 8/94, London, England.
- [30] Lawrence S. Brakmo, Larry L. Peterson (1995), "TCP Vegas: End to End Congestion Avoidance on a Global Internet", *IEEE Journal on Selected Area in Communications*, 13(8).1995.
- [31] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Ya Xu, H. Yu , "Advances in Networks Simulation", *The VINT Project*. May 2000.
- [32] Leonard Kleinrock (1976), *Queueing systems, Volume II: Computer Applications*, John Wiley&Sons, New York, 1976.
- [33] Leonard Kleinrock (1993), "On the Modeling and Analysis of Computer Networks", *Proceeding of the IEEE*, 81(8), 1993.
- [34] Matthew Mathis, Jamshid Mahdavi, Sally Floyd, and Allyn Romanow (1996), "TCP Selective Acknowledgement option (and related changes) for FreeBSD". (Internet draft, work in progress), 1996.
- [35] Mischa Schwartz (1987), *Telecommunication Networks: Protocols, Modeling and Analysis*, Addition-Wesley, Massachusetts.
- [36] P. Schweitzer and S. Lam (Nov. 1976), "Buffer overflow in a store and forward network node", *IBM J. Res. Develop*, 542-550. 1976.

- [37] L. Badia, M. Rossi, M. Zorzi, "Some Results on the Statistics of Delay Terms in SR ARQ on Markov Channels," *Proc. IEEE ISWCS, Siena, Italy*, Sep. 2005.
- [38] J. Postel, "Transmission Control Protocol", RFC 793, September 1981
- [39] D.-L. Lu, J.-F. Chang, "Performance of ARQ Protocols in Nonindependent Channel Errors," *IEEE Trans. Commun.*, vol. 41, no. 5, pp. 721–730, 1993.
- [40] M. Rossi, L. Badia, M. Zorzi, "SR-ARQ delay statistics on N-State Markov channels with finite round trip delay," *IEEE Trans. on Wirel. Commun.*, Jun. 2006.
- [41] L. Badia, "On the impact of correlated arrivals and errors on ARQ delay terms," *IEEE Trans. on Commun.*, Feb. 2009.
- [42] W. Luo, K. Balachandran, S. Nanda, and K. Chang, "Delay analysis of selective-repeat ARQ with applications to link adaptation in wireless packet data systems," *IEEE Trans. Wireless Commun.*, vol. 4, no. 3, pp. 1017–1029, May 2005.
- [43] L. Badia, M. Levorato, M. Zorzi, "Markov analysis of selective repeat type II hybrid ARQ using block codes," *IEEE Trans. Commun.*, Sep. 2008.
- [44] L. Badia, M. Levorato, M. Zorzi, "A channel representation method for the study of hybrid retransmission-based error control," *IEEE Trans. Commun.*, 2009.
- [45] Michele Garetto and Don Towsley, "An Efficient Technique to Analyze the Impact of Bursty TCP Traffic in Wide-Area Networks", *Performance Evaluation*, 65(2), pp. 181-202, February 2008
- [46] Yu-feng Li, Han Qiu, Ju-Long Lan, and Bin-Qiang Wang, "Study on TCP Flow-Competing Congestion and Buffer Requirement of the Congested Links", *Journal of Software*, 19(6), pp.1499-1507, June 2008
- [47] Joseph Kee-yin, Karl R. P. H, Wai Wong, "Quality of Service for MPEG Video in Human Perspective", <http://www.comp.hkbu.edu.hk/tech-report/tr03002f.pdf>, 2002.
- [48] Ivaylo Haratcherev, Jacco Taal, Koen Langendoen, Reginald Lagendijk, Henk Sips, "Fast 802.11 link adaptation for real-time video streaming by cross-layer signaling", *In Proc. International Symposium on Circuits and Systems, Kobe*, 2005.

- [49] Adam Petcherm, "QoS in Wireless Data Networks", http://www.cs.wustl.edu/~jain/cse574-06/wireless_qos.htm, 2006.
- [50] Qingwen Liu, Xin Wang, GB Giannakis, "A cross-layer scheduling algorithm with QoS support in wireless networks", *Vehicular Technology, IEEE Transactions on*, Vol. 55, No. 3. pp. 839-847 (2006).
- [51] T. He, B.M. Blum, J.A. Stankovic and T. Abdelzaher, "AIDA: Adaptive Application-Independent Data Aggregation in Wireless Sensor Networks", *ACM Transactions on Embedded Computing Systems*, Vol. 3, No. 2, Pages 426–457, May 2004.
- [52] D. H. Hoang (2005), "Quality of Service Control in the Overlay Networks Using a TCP-like Protocol", *Asian Information Committee AIC33 International Conference*, paper No 132, Jakarta, Indonesia, 21-25th November 2005,
- [53] Yu Gu, Don Towsley, Chris V. Hollot, and Honggang Zhang, "Congestion Control for Small Buffer High Speed Networks", *Proceedings of IEEE INFOCOM 2007*, pp. 1037-1045, May 2007.
- [54] Eugenius Kaszkurewicz and Amit Bhaya, "An Adaptive AIMD-Type Scheme for Congestion Control Derived Using Virtual Equilibria", *Proceedings of the 46th IEEE Conference on Decision and Control (CDC 2007)*, pp. 5696-5703, December 2007
- [55] Yueping Zhang and Dmitri Loguinov, "ABS: Adaptive Buffer Sizing for Heterogeneous Networks", *Proceedings of the 16th International Workshop on Quality of Service (IWQoS 2008)*, pp. 90-99, June 2008
- [56] T. Abdelzaher, Y. Diao, J.L. Hellerstein, C. Lu and X. Zhu, Introduction to Control Theory and its Application to Computing Systems, Book Chapter, *Performance Modeling and Engineering*, Springer, 2008. Note: Reading for the tutorial at SIGMETRICS'08
- [57] X. Wang, D. Jia, C. Lu and X. Koutsoukos, DEUCON: Decentralized End-to-End Utilization Control for Distributed Real-Time Systems, *IEEE Transactions on Parallel and Distributed Systems*, 18(7):996-1009, July 2007.
- [58] Albert Cabellos-Aparicio, A Novel Available Bandwidth Estimation and Tracking Algorithm, *E2EMON conference*, 2008

- [59] A. Khiruthika Devi, S. Rajarajeswari, Dr. J. Sutha “Enhancing the Performance of Wireless Network through Efficient Bandwidth Estimation Technique”, *International Journal of Advanced Research in Computer Science & Technology IJARCSST*, 2014.
 - [60] Vu Tat Thanh , “A Transport Protocol With Optimised Packet Size” ; *IWK conference, Germany*, 2003.
 - [61] Vu Tat Thanh, “Effect of RED and different packet sizes on Multimedia performance over wireless networks” ; *MIK Conference, Germany*, 2003
 - [62] Atul M. G.; Bhargavi H. G.; Uditnarayan K. “Evolution of Congestion Control Mechanisms for TCP and Non TCP Protocols”. *Matrix Academic International Online Journal Of Engineering And Technology (MAIOJET)*, p13-20, 2013.
 - [63] Jorge Navarro-Ortiz , “Removing redundant TCP functionalities in wired-cum-wireless networks with IEEE 802.11e HCCA support”, *International Journal of Communication Systems*, April, 2013.
- <http://onlinelibrary.wiley.com/doi/10.1002/dac.2546/abstract>
- [64] Taleb, T. ; Grad. Sch. of Inf. Sci., Tohoku Univ., Sendai ; Kashibuchi, K. ; Leonardi, A. ; Palazzo, S. , “A Cross-Layer Approach for an Efficient Delivery of TCP/RTP-Based Multimedia Applications in Heterogeneous Wireless Networks”, *IEEE Transactions on Vehicular Technology (Volume:57 , Issue: 6)*, Nov.2008.
 - [65] Bin Zhou ; Cheng Peng Fu ; Li , “TFRC Veno: An Enhancement of TCP Friendly Rate Control over Wired/Wireless Networks”, *Network Protocols, ICNP 2007. IEEE International Conference*, 2007.
 - [66] Lin, Y.H. and Z.H. Long, “Improved algorithm of TFRC aiming at controlling real-time transmission in wireless network”, *Comp. Eng. Design*, 31: 1898-1900, 2010.

- [67] Zhou, B., C.P. Fu, C.T. Lau, C.H. Foh., "An enhancement of TFRC over wireless networks", *Proceedings of the IEEE Wireless Communications and Networking Conference, Hong Kong*, pp: 3019-3024, March 11-15, 2007.
- [68] Li, Q., D. Chen, Y.C. Liu, L.N. Zheng," Jitter ratio based TFRC scheme in wireless-wired hybrid network", *Proceedings of the International Conference on Digital Telecommunications, Cap Esterel, Cote d'Azur, France*, pp: 38-38, August 29-31, 2006.
- [69] Zhi-Jiang Xu, Shui-Qin Chu, Yuan Wu and Li-Min Meng, "An Improved TFRC Scheme for Wired/Wireless Hybrid Networks", *Information Technology Journal*, 12: 168-173, 2012.
- [70] Sreekumari and Lee , "TCP NRT: a new TCP algorithm for differentiating non-congestion retransmission timeouts over multihop wireless networks", *EURASIP Journal on Wireless Communications and Networking*, 2013.
- [71] L Long, J Aikat, K Jeffay, FD Smith, "The effects of active queue management and explicit congestion notification on web performance", *Networking, IEEE/ACM Transactions* 15, 1217–1230, 2007.
- [72] A M Al-Jubari, M Othman, B Mohd Ali, and N A Abdul Ham, "An Adaptive Delayed Acknowledgment Strategy to Improve TCP Performance in Multi-hop Wireless Networks", *Wireless Pers Commun, Springer Science+Business Media*, 2012.
- [73] P.Praveen Kumar, Dr.K. Rameswaraiah," Improving The Performance Of Congestion Control In Wireless Networks", *International Journal of Engineering Research and Applications (IJERA)* ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 3, May-Jun 2013
- [74] Ahn, S.; Ramaswami, V. "Fluid Flow Models and Queues—A Connection by Stochastic Coupling". *Stochastic Models*, Vol. 19, Issue 3, p325-348, 2003.

- [75] Ashish C., Mauli J. , Shanu M. “A Survey on versions of TCP over WiMAX”.
International Journal of Application or Innovation in Engineering & Management (IJAIEEM), Volume 3, Issue 5, May 2014.