**LỜI NÓI ĐẦU**

Con người cần giao tiếp. Bằng cách này hay cách khác, chúng ta luôn trao đổi các thông tin với nhau. Trong vài thập kỷ qua, công nghệ mới đã được phát minh để hỗ trợ quá trình trao đổi thông tin này một cách hiệu quả và sáng tạo. Trong số đó có thể kể đến việc tạo ra mạng điện thoại, các kênh sóng truyền hình và radio, sự ra đời của máy tính và sự xuất hiện của các kế nối không dây. Ban đầu, những công nghệ này tồn tại và hoạt động độc lập, phục vụ cho mục đích riêng của mỗi lãnh vực. Mãi cho đến gần đây, những kỳ quan công nghệ này dường như hội tụ, và chúng ta đã hình thành quan niệm phổ biến rằng mạng truyền thông máy tính là kết quả của sự hội tụ này.

Chương này trình bày tổng quan về các mạng truyền thông máy tính, và các khái niệm cơ bản về mô phỏng mạng. Phần 1.1 giới thiệu mạng máy tính cùng với mô hình tham chiếu được sử dụng để mô tả kiến trúc truyền thông của mạng máy tính. Trình bày ngắn gọn về thiết kế và lập mô hình một hệ thống phức tạp như mạng máy tính được đưa ra trong Mục 1.2. Trong Phần 1.3, những điều cơ bản của mạng máy tính mô phỏng. Phần 1.4 trình bày một trong những loại phổ biến nhất mô phỏng thời gian mô phỏng mạng. Mô phỏng ví dụ được nêu trong mục 1.5. Cuối cùng, mục 1.6 là tóm tắt chương.

* 1. **Mạng máy tính và kiến trúc phân tầng**

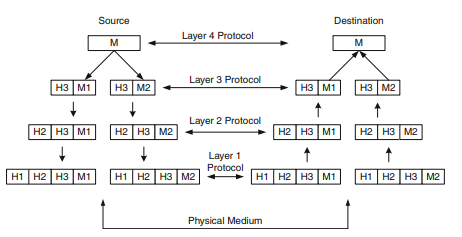
Một mạng máy tính thường được định nghĩa là một tập hợp các máy tính kết nối với nhau để thu thập, xử lý, và phân phối thông tin. Máy tính được hiểu như là một khái niệm rộng ở đây bao gồm các thiết bị như máy trạm, máy chủ, router, modem, trạm cơ sở, các điểm mở rộng không dây, v.v… Những máy tính được nối với nhau bằng các liên kết truyền thông như cáp đồng, cáp quang cáp quang và vi sóng / sóng vệ tinh / sóng vô tuyến. Một mạng máy tính có thể được xây thiết kế thành một tổ và có thể kết nối tới nhiều mạng. Internet là một ví dụ tốt về mạng máy tính. Trong thực tế, nó là một mạng của các mạng, trong đó, hàng chục ngàn mạng kết nối hàng triệu máy tính trên toàn thế giới.

* + 1. **Kiến trúc phân tầng**

Mạng máy tính là một hệ thống phức tạp. Để tạo điều kiện thiết kế và triển khai linh hoạt của một hệ thống như vậy, kiến trúc phân tầng ra đời. Sử dụng kiến trúc phân tầng, các chức năng của một mạng máy tính có thể được tổ chức như một chồng các tầng. Có mối quan hệ ngang hàng (hoặc liên kết ảo) giữa các tầng tương ứng trong hai nút giao tiếp. Tuy nhiên, thực tế luồng dữ liệu diễn ra theo chiều dọc - từ tầng cao nhất đến tầng thấp nhất trong một nút, và sau đó thông qua các liên kết vật lý để đi đến tầng thấp nhất ở nút kia, sau đó đi ngược lên trên để tới tầng cao nhất của nút đó. Mỗi tầng đại diện cho một phần riêng biệt và cụ thể của hệ thống, và cung cấp một số **dịch vụ** nhất định cho tầng trên. Có thể truy cập (bởi tầng phía trên) thông qua cái gọi là **giao diện**, các dịch vụ này thường xác định *cái gì* được thực hiện trong hoạt động mạng hay các hàm nguyên thủy (primitive), nhưng không xác định l*àm thế nào* để thực hiện những điều đó. Cách một dịch vụ được triển khai được gọi là một **giao thức**. Ví dụ, bộ phát tại nguồn có thể sử dụng giao thức cụ thể (ví dụ: lược đồ mã hóa dữ liệu) ở tầng vật lý để truyền các bit thông tin tới đích.

Đích có thể giải mã thông tin nhận được dựa trên các quy tắc của giao thức. Sự ưu việt của kiến trúc phân tầng này là tính độc lập giữa các tầng. Sự thay đổi trong giao thức của một tầng nhất định, miễn là các giao diện không đổi thì không ảnh hưởng đến phần còn lại của hệ thống . Ở đây, các từ dịch vụ, giao thức và giao diện được in đậm để nhấn mạnh rằng đó là sự tương tác giữa các thành phần này tạo nên kiến trúc phân tầng.

*Hình 1.1. Dòng dữ liệu trong kiến trúc phân tầng*

Hình 1.1 cho thấy một cái nhìn tổng thể về kiến trúc phân tầng được sử dụng để liên lạc giữa hai máy chủ lưu trữ: máy chủ lưu trữ nguồn và máy chủ lưu trữ đích. Trong hình này, các thành phần của mỗi máy chủ chia thành bốn tầng. 1 Mỗi tầng ở máy chủ này lại có một liên kết ảo với một tầng ở máy chủ kia, và như vậy chúng được gọi là các tầng *ngang hàng*.2 Mặc dù không được kết nối trực tiếp với nhau, những tầng này có mối liên kết ảo với nhau bằng giao thức được biểu thị bằng mũi tên. Như đã được đề cập, thực tế thông tin liên lạc cần được đưa xuống dần theo từng tầng như kiến trúc phân tầng đã đề cập ở trên.

Giả sử một tiến trình đang chạy trên Tầng 4 của nguồn tạo ra dữ liệu hoặc tin nhắn để gửi tới đích. Quá trình trao đổi bắt đầu bằng cách truyền một gói tin M xuống Tầng 3, nơi dữ liệu được phân đoạn thành hai khối (M1 và M2) và thông tin điều khiển được gọi là *header* (H3) đại diện cho Tầng 3 được nối thêm vào M1 và M2. Thông tin kiểm soát có thể là số thứ tự, kích cỡ gói và thông tin kiểm tra lỗi. Những thông tin này chỉ có thể hiểu và được sử dụng bởi các tầng ngang hàng của máy chủ đích để khôi phục dữ liệu (M). Dữ liệu kết quả (ví dụ: H3 + M1) được đẩy xuống tầng liền dưới, nơi một số thông tin điều khiển lại được thêm vào tin nhắn theo giao thức riêng. Quá trình này tiếp tục cho đến khi gói tin đạt đến tầng thấp nhất, ở đây việc truyền tin thực sự được thực hiện qua phương tiện vật lý. Lưu ý rằng, dọc theo quá trình này có thể cần phải phân đoạn thêm dữ liệu từ các tầng trên vào các phân đoạn nhỏ hơn cho các mục đích khác nhau. Khi gói tin đã tới đích, quá trình đảo ngược diễn ra. Trong quá trình gói tin được đẩy dần từ dưới lên, header của nó được tách dần theo từng tầng. Nếu cần thiết, một số gói tin được đặt lại với nhau trước khi được chuyển tới tầng trên. Quá trình tiếp tục cho đến khi gói tin gốc (M) được phục hồi ở tầng 4.

**1.1.2 Mô hình tham chiếu OSI và TCP/IP**

Mô hình OSI (Open Systems Interconnection) là mô hình tham chiếu đầu tiên được phát triển bởi ISO (Tổ chức tiêu chuẩn hóa quốc tế) để cung cấp một tiêu chuẩn mẫu mô tả chồng giao thức trong mạng máy tính. Nó bao gồm bảy tầng nơi mỗi tầng được thiết kế để thực hiện các chức năng riêng biệt. Đó là: tầng vật lý, tầng liên kết dữ liệu, tầng mạng, tầng giao vận, tầng phiên, tầng trình diễn và tầng ứng dụng. Mô hình OSI chỉ xác định mỗi tầng cần làm gì; nó không chỉ định các dịch vụ và giao thức chính xác được sử dụng trong mỗi tầng. Mặc dù không được triển khai trong các hệ thống hiện tại, triết lý mô hình OSI (kiến trúc phân tầng) đặt nền tảng cho sự phát triển trong máy tính mạng.

Mô hình tham chiếu TCP (Giao thức điều khiển truyền dẫn) / IP (Giao thức Internet), dựa trên hai giao thức chính, cụ thể là, TCP và IP, được sử dụng trong Internet hiện tại. Các giao thức này đã thể hiện sự mạnh mẽ, và kết quả là chúng được sử dụng và triển khai rộng rãi trong các mạng máy tính hiện nay. Nó được phát triển cho ARPANET, một nghiên cứu mạng được Bộ Quốc phòng Hoa Kỳ tài trợ, được coi là cha để của tất cả các mạng máy tính. Trong mô hình TCP/IP, giao thức ngăn xếp bao gồm năm tầng - vật lý, liên kết dữ liệu, mạng, giao vận và ứng dụng - mỗi tầng trong số đó chịu trách nhiệm cho một số dịch vụ nhất định sẽ được trình bày bên dưới. Lưu ý rằng tầng ứng dụng trong mô hình TCP/IP có thể là được coi là sự kết hợp giữa các tầng phiên, tầng trình diễn và tầng ứng dụng của mô hình OSI.

**Tầng ứng dụng**

Tầng ứng dụng nằm trên cùng của mô hình và sử dụng các dịch vụ từ tầng giao vận (Trình bày dưới đây). Tầng này hỗ trợ một số giao thức mức cao hơn chẳng hạn như HTTP (Giao thức truyền dẫn siêu văn bản) cho các ứng dụng World Wide Web, SMTP (Giao thức truyền thư đơn giản) cho thư điện tử, TELNET cho thiết bị đầu cuối ảo từ xa, DNS (Dịch vụ tên miền) để ánh xạ tên miền và địa chỉ mạng của máy chủ và FTP (Giao thức truyền file) để truyền tệp.

**Tầng giao vận**

Mục tiêu của tầng giao vận là truyền tải thông điệp từ tầng ứng dụng của máy chủ lưu trữ nguồn đến máy chủ lưu trữ đích. Để thực hiện mục tiêu này, hai giao thức nổi tiếng, cụ thể là TCP và UDP (User Datagram) , được định nghĩa trong tầng này. Trong khi TCP có trách nhiệm tạo kết nối và định hướng một cách tin cậy giữa các máy chủ, UDP hỗ trợ kết nối truyền nhận không tin cậy. TCP lý tưởng cho các ứng dụng cần độ chính xác cao hơn là tốc độ truyền, UDP thì ngược lại.

Thông thường, kiểm soát thông tin liên quan đến kiểm soát luồng và kiểm soát lỗi cần phải được nhúng vào các gói tin. Ngoài ra, trước khi thêm bất kỳ header nào, phân mảnh thường được thực hiện để phá vỡ một gói tin dài thành các phân đoạn. Vì lý do này, các đơn vị dữ liệu giao thức trong tầng này thường được gọi là các *phân đoạn*.

**Tầng mạng**

Tầng này cung cấp các dịch vụ định tuyến cho tầng giao vận. Tầng mạng được thiết kế để phân phối các đơn vị dữ liệu, thường được gọi là các *gói dữ liệu*, dọc theo các đường dẫn đi từ nguồn tới đích. Để tạo điều kiện định tuyến, tiêu đề chứa thông tin địa chỉ nguồn và đích được thêm vào đơn vị dữ liệu giao thức truyền tải nhằm tạo ra các đơn vị dữ liệu của của tầng mạng.

**Tầng liên kết**

Các gói tin thường được định tuyến qua một số liên kết truyền thông và các nút trước khi chúng tới được nút đích. Để định tuyến thành công các gói này, bằng cách này hay cách khác đến được đích, một cơ chế cần được đặt ra để nối từng nút này tới nút khác qua mỗi liên kết truyền thông. Giao thức tầng liên kết chịu trách nhiệm phân phối dữ liệu trên một liên kết truyền thông.

Giao thức tầng liên kết có ba trách nhiệm chính. Đầu tiên, kiểm soát tốc độ truyền trong liên kết truyền thông. Thứ hai, kiểm soát lỗi đảm bảo tính toàn vẹn của dữ liệu. Thứ ba, ghép/phân tách kênh kết hợp nhiều luồng dữ liệu vào và trích xuất các luồng dữ liệu từ một liên kết truyền thông. Lựa chọn giao thức tầng liên kết có thể thay đổi từ “máy chủ tới máy chủ” hay “mạng sang mạng”. Ví dụ về các giao thức / công nghệ tầng liên kết được sử dụng rộng rãi bao gồm Ethernet, Giao thức điểm-điểm (PPP), IEEE 802.11 (như WiFi) và Chế độ truyền không đồng bộ (ATM).

**Tầng vật lý**

Tầng vật lý đảm nhận việc truyền các bit dữ liệu qua một giao tiếp liên kết. Mục tiêu chính của nó là đảm bảo rằng các thông số truyền (ví dụ: công suất truyền tải, sơ đồ điều chế) được thiết lập phù hợp để đạt được hiệu suất truyền yêu cầu (ví dụ: để giữ được tỉ lệ lỗi ở mức cho phép).

Cuối cùng, chúng tôi chỉ ra rằng năm tầng được Trình bày ở trên là phổ biến đối với OSI. Như đã được đề cập, mô hình OSI còn bao gồm hai tầng nằm trên tầng giao vận, cụ thể là tầng phiên và tầng trình diễn. Tầng phiên chỉ đơn giản cho phép người dùng trên các máy tính khác nhau tạo các phiên giao tiếp với nhau. Tầng trình diễn về cơ bản quản lý sự trình diễn dữ liệu khác nhau hiện có trên mạng. Ví dụ, một hệ thống mạng quản lý việc thu thập dữ liệu với các định dạng khác nhau, từ cá máy tính khác nhau và chuyển đổi định dạng của các dữ liệu này thành định dạng phổ biến.

**1.2 Mô hình hệ thống**

Mô hình hóa hệ thống đề cập đến hành động đại diện cho một hệ thống thực tế một cách đơn giản. Mô hình hóa hệ thống rất quan trọng trong thiết kế hệ thống và phát triển, vì nó đưa ra ý tưởng về cách hệ thống sẽ thực hiện nếu thực sự được triển khai.

Với mô hình hóa, các thông số của hệ thống có thể được thay đổi, kiểm tra và phân tích. Quan trọng hơn, mô hình hóa, nếu được xử lý đúng cách, có thể tiết kiệm chi phí phát triển hệ thống.

Để mô hình hóa một hệ thống, một số giả định đơn giản hóa thường được yêu cầu. Điều quan trọng cần lưu ý là quá nhiều giả định sẽ đơn giản hóa mô hình hóa nhưng có thể dẫn đến sự biểu diễn không chính xác của hệ thống.

Theo truyền thống, có hai phương pháp mô hình hóa: phương pháp tiếp cận phân tích và phương pháp tiếp cận mô phỏng.

**1.2.1. Phương pháp tiếp cận phân tích**

Khái niệm chung của mô hình phân tích tiếp cận, đầu tiên là đưa ra cách để mô tả một hệ thống toán học với sự giúp đỡ của việc áp dụng các công cụ phân tích toán học như là hàng đợi và lý thuyết về xác suất. Và sau đó áp dụng các phương pháp số học để đạt được cái nhìn sâu sắc từ việc phát triển mô hình toán học.

Khi hệ thống đơn giản và tương đối nhỏ, mô hình phân tích sẽ thích hợp hơn (mô phỏng). Trong trường hợp này, mô hình có xu hướng toán học có thể xử lý được. Các giải pháp số về mô hình này có hiệu lực đòi hỏi những nỗ lực tính toán nhẹ.

Nếu được sử dụng đúng cách, mô hình phân tích có thể tiết kiệm chi phí và có thể cung cấp một cái nhìn trừu tượng về các thành phần tương tác với nhau trong hệ thống. Tuy nhiên, nếu nhiều giả định đơn giản hóa trên hệ thống là được thực hiện trong quá trình lập mô hình, các mô hình phân tích có thể không đưa ra một biểu diễn chính xác về hệ thống thực.

**1.2.2 Phương pháp tiếp cận mô phỏng**

Mô phỏng được sử dụng rộng rãi trong mô hình hóa hệ thống cho các ứng dụng khác nhau, từ nghiên cứu kỹ thuật, phân tích kinh doanh, lập kế hoạch sản xuất và thử nghiệm khoa học sinh học, trên đây chỉ mới là một số ví dụ mà thôi.

So với mô hình phân tích, mô phỏng thường đòi hỏi ít trừu tượng hơn trong mô hình (ví dụ, ít giả định đơn giản hơn) vì hầu như mọi chi tiết có thể có của các thông số kỹ thuật của hệ thống có thể được đưa vào mô hình mô phỏng để mô tả tốt nhất hệ thống thực tế.

Khi hệ thống khá lớn và phức tạp, một công thức toán học đơn giản có thể không khả thi. Trong trường hợp này, phương pháp mô phỏng thường được ưu tiên hơn cho phương pháp phân tích.

Nói chung với mô hình phân tích, mô hình mô phỏng có thể bỏ qua một số chi tiết, vì quá nhiều chi tiết có thể dẫn đến kết quả là không thể quản lý một mô phỏng và việc cố gắng tính toán là không đáng kể.

Điều quan trọng là phải xem xét cẩn thận một biện pháp đang được xem xét và không bao gồm chi tiết không liên quan vào mô phỏng.

Trong phần tiếp theo, chúng tôi mô tả các khái niệm cơ bản về mô phỏng chi tiết hơn với sự nhấn mạnh đặc biệt về mô phỏng mạng máy tính.

**1.3 Khái niệm cơ bản về mô phỏng mạng máy tính**

Một mô phỏng cho dù ít hay nhiều là một sự kết hợp của nghệ thuật và khoa học. Đó được gọi là chuyên môn trong lập trình máy tính và các công cụ toán học được áp dụng chiếm một phần quan trọng trong khoa học, kỹ năng phân tích và mô hình khái niệm.

Công thức thường sẽ đại diện cho phần nghệ thuật. Chúng ta có một danh sách dài các bước để thực hiện một quá trình mô phỏng, như được đưa ra, dường như đã phản ánh được yêu cầu phổ biến này.Về cơ bản, tất cả các bước này có thể được đưa vào ba nhiệm vụ chính, mỗi nhiệm vụ mang các mức độ quan trọng khác nhau.  
 Theo Shannon, 40% thời gian và nỗ lực được dành cho việc xác định vấn đề, thiết kế mô hình tương ứng và tạo ra một tập hợp các thí nghiệm được thực hiện trên mô hình mô phỏng. Hơn nữa, nó đã được chỉ ra rằng 1/20 % nên được sử dụng để lập trình các yếu tố khái niệm thu được trong bước đầu tiên. Cuối cùng là phần còn lại, 40 % nên được sử dụng trong việc xác minh / xác nhận mô hình mô phỏng, thử nghiệm với đầu vào được thiết kế (và bản thân các thử nghiệm có thể tự động điều chỉnh được ), phân tích kết quả. Có một lưu ý đó là công thức này không hoàn toàn chính xác. Bất kỳ mô phỏng thực tế nào cũng có thể cần nhiều thời gian và nỗ lực hơn, tùy thuộc vào bối cảnh và quan trọng là ở người lập mô hình.  
 Mô phỏng có thể được coi là một quá trình xuyên suốt dòng chảy của các thực thể mạng (ví dụ: nút, gói tin). Khi các thực thể này di chuyển qua hệ thống, chúng tương tác với các thực thể khác, tham gia các hoạt động nhất định, kích hoạt sự kiện, gây ra một số thay đổi đối với trạng thái của hệ thống và rời khỏi quá trình. Đôi khi, chúng tranh đấu hoặc chờ một số loại tài nguyên. Điều này có nghĩa là phải có một logic các trình tự được thực hiện để làm cho tất cả các hành động này xảy ra theo một cách dễ hiểu và quản lý được. Một chuỗi này đóng một vai trò quan trọng trong giám sát mô phỏng và đôi khi được sử dụng để mô tả các loại mô phỏng (xem Phần 1.4).

**1.3.1 Mô phỏng: Định nghĩa chính thức**

Theo Shannon, mô phỏng là “quá trình thiết kế mô hình của một hệ thống thực sự và tiến hành các thí nghiệm với mô hình này với mục đích hiểu hành vi của hệ thống và / hoặc đánh giá các chiến lược khác nhau cho hoạt động của hệ thống. ”Với tính chất động của mạng máy tính, do đó chúng tôi thực sự phải đối phó với một mô hình động của một hệ thống vô cùng năng động.

**1.3.2 Các yếu tố mô phỏng**  
 Theo Ingalls, các thành phần cấu trúc của một mô phỏng bao gồm những điều sau đây:

*Thực thể*

Các thực thể là các đối tượng tương tác với nhau trong một chương trình mô phỏng gây ra một số thay đổi đối với trạng thái của hệ thống. Trong bối cảnh máy tính mạng, các thực thể có thể bao gồm các nút máy tính, gói dữ liệu, luồng gói hoặc các đối tượng phi vật lý như đồng hồ mô phỏng. Để phân biệt các thực thể, các thuộc tính duy nhất được gán cho từng thuộc tính. Ví dụ, một gói tin thực thể có thể có các thuộc tính như độ dài gói, số thứ tự, mức độ ưu tiên và tiêu đề.

*Tài nguyên*

Tài nguyên là một phần của các hệ thống phức tạp này. Nhìn chung, nguồn cung hạn chế nguồn lực phải được chia sẻ giữa một nhóm các thực thể nhất định. Đây thường là trường hợp cho mạng máy tính, nơi băng thông, thời gian phát sóng, số lượng máy chủ, ví dụ, đại diện cho các tài nguyên mạng phải được chia sẻ giữa các thực thể mạng.

*Các hoạt động và sự kiện*

Theo thời gian, các thực thể tham gia vào một số hoạt động. Sự hấp dẫn này tạo  
các sự kiện và kích hoạt các thay đổi trong trạng thái hệ thống. Các ví dụ phổ biến về hoạt động bao gồm độ trễ và xếp hàng. Khi máy tính cần gửi gói nhưng máy tính đó đang ở chế độ bận thì nó sẽ chờ cho đến khi máy tính được thoát khỏi chế độ đó. Trong trường hợp này, gói được gửi qua không trung nhưng máy tính đang bận, gói tin được nói đến ở trên sẽ tham gia vào hoạt động chờ.

*Lập lịch biểu*

Trình lên lịch sẽ duy trì danh sách các sự kiện và thời gian thực hiện của chúng. Trong một mô phỏng, nó sẽ chạy một đồng hồ mô phỏng tạo ra các sự kiện và thực hiện chúng.

*Biến toàn cục*

Trong mô phỏng, một biến toàn cục có thể truy cập được bởi bất kỳ hàm hoặc thực thể nào trong hệ thống, và về cơ bản sẽ là theo dõi một số giá trị chung của mô phỏng. Trong ngữ cảnh của mạng máy tính, các biến như vậy có thể đại diện, ví dụ: độ dài của hàng đợi gói trong một mạng máy chủ đơn lẻ, tổng thời gian phát sóng bận của mạng không dây, hoặc tổng số gói tin được truyền đi.

*Trình tạo số ngẫu nhiên*

Một bộ tạo số ngẫu nhiên (RNG) được yêu cầu để giới thiệu tính ngẫu nhiên trong một mô hình mô phỏng. Số ngẫu nhiên được tạo bằng cách chọn tuần tự số từ một chuỗi xác định số ngẫu nhiên psudo, nhưng số được chọn từ chuỗi này sẽ là ngẫu nhiên. Trong hầu hết trường hợp, một chuỗi ngẫu nhiên psudo được xác định trước và được sử dụng bởi tất cả RNG.

Trong nhiều trường hợp, cần có một vài kết quả thống kê nhất định. Một yêu cầu RNG để bắt đầu chọn số từ vị trí khác (ví dụ: hạt giống) trong (giống nhau) trình tự psudo được xác định trước. Nếu không, kết quả cho mọi lần chạy sẽ giống nhau. Trong thực tế, một RNG được khởi tạo với một hạt giống.

Một hạt giống xác định vị trí bắt đầu trong một chuỗi ngẫu nhiên psudo, trong đó một RNG bắt đầu chọn số. Mô phỏng khác nhau được khởi tạo hạt giống khác nhau do đó tạo ra kết quả khác nhau (nhưng thống kê giống hệt nhau).

Trong một mô phỏng mạng máy tính, ví dụ, một quá trình đến gói,  
quá trình chờ đợi và quy trình dịch vụ thường được mô hình hóa như là các quá trình ngẫu nhiên.

Một quá trình ngẫu nhiên được thể hiện bằng các chuỗi các biến ngẫu nhiên. Những quá trình ngẫu nhiên thường được thực hiện với sự hỗ trợ của một RNG. Đối với một thực nghiệm toàn diện về việc thực hiện quy trình ngẫu nhiên (ví dụ: những người có các hàm phân phối đồng đều, hàm mũ, Gaussian, Poisson, nhị thức).

*Thống kê thu thập*

Trách nhiệm chính của người thu thập số liệu là thu thập dữ liệu từ các  
mô phỏng để các suy luận có ý nghĩa và có thể được rút ra từ các dữ liệu đó.

**1.4 Mô phỏng phụ thuộc thời gian**

Một loại mô phỏng chính là mô phỏng phụ thuộc vào thời gian cái mà hành động theo thời gian. Loại mô phỏng này duy trì một đồng hô mô phỏng cái mà theo dõi thời gian mô phỏng hiện tại. Trong hầu hết các trường hợp, việc mô phỏng chạy cho đến khi đồng hồ đạt đến mức được xác định từ trước.

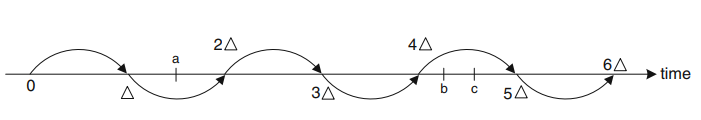
Mô phỏng phụ thuộc vào thời gian có thể phân chia thành mô phỏng theo thời gian mô phỏng và theo sự kiện mô phỏng. Một mô phỏng theo thời gian gây ra và thực hiện các sự kiện cho mọi khoảng thời gian được cố định. Hay nói cách khác, mô phỏng tăng lên từ khoảng thời gian này sang khoảng thời gian khác và thực hiện các sự kiên ( nếu có ) cho tới khi đạt đến giới hạn nhất định. Một mô phỏng theo sự kiện, mặt khác, gây ra các sự kiện ở một thời điểm tùy ý. Mô phỏng chuyển từ sự kiện này sang sự kiện khác và thực hiện sự kiện lại một lần nữa ( nếu có ) cho đến khi mô phỏng kết thúc.

Một số chú ý quan trọng cho mô phỏng phụ thuộc vào thời gian: Mô phỏng phải tiến hành theo thứ tự thời gian. Đương nhiên lưu ý này khá đơn giản cho một mô phỏng theo thời gian. Xác định hai điểm quan trọng cho một mô phỏng thực hiện theo hướng sự kiện: Đầu tiên, mọi sự kiện mới được lên lịch vào danh sách sự kiện phải được gắn thẻ với một dấu thời gian bằng hoặc lớn hơn thời gian của sự kiện hiện tại. Hay nói cách khác không có sự kiện lỗi thời nào được lên lịch. Thứ hai, sự kiện tiếp theo mà mô phỏng luôn thực hiện là sự kiện với dấu thời gian nhỏ nhất trong danh sách sự kiện. Nó sẽ không bao giờ nhảy qua thứ tự thời gian sự kiện được sắp xếp hoặc quay trở lại sự kiện trước đây.

**1.4.1. Mô phỏng theo thời gian**

Trong mô phỏng theo thời gian, đồng hồ mô phỏng được nâng cao tính chính xác bằng một khoảng thời gian cố định của đơn vị thời gian. Sau mỗi sự gia tăng của đồng hồ, mô phỏng tìm kiếm sự kiện có thể xảy ra trong suốt khoảng thời gian cố định này. Nếu có, các sự kiện như vậy được xử lý như là chúng đã xảy ra vào cuối của khoảng thời gian này.

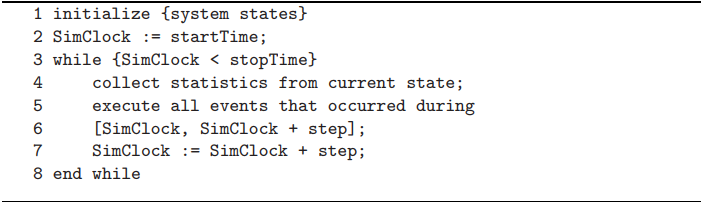
Hình 1.2 chỉ ra ý tưởng cơ bản đằng sau sự gia tăng trong một thời gian mô phỏng. Mũi tên cong đại điện cho những sự tăng lên như vậy và a,b, và c đánh dấu sự xuất hiện của các sự kiện cụ thể. Trong khoảng thời gian đầu tiên, không có sự kiện nào xảy ra, trong khi khoảng thời gian thứ hai chứa sự kiện a, cái mà không được dùng cho đến khi kết thúc khoảng thời gian. Một bất lợi của mô phỏng theo thời gian được minh họa ở khoảng thứ năm, nơi các sự kiện b và c được coi là xảy ra chính xác vào cuối khoảng thời gian. Điều này đòi hỏi một thủ tục xác định sự kiện nào sẽ được xử lý đầu tiên. Một giải pháp xung quanh tình huống này là thu hẹp khoảng thời gian mô phỏng sao cho mỗi khoảng chỉ chứa một sự kiện. Điều này, tuy nhiên, đặt đáng kể gánh nặng giả định trên giả lập. Mô phỏng theo thời gian thì không được đề xuất cho các mô hình hệ thống cái mà sự kiện của nó có xu hướng xảy ra khắp một khoảng thời gian ngẫn nhiên.



*Hình 1.2. Đồng hồ sự gia tăng trong môt mô phỏng theo thời gian*

Ví dụ 1.1. Chương trình 1.1 cho thấy các mã giả mô phỏng theo thời gian. Dòng 1và 2 khởi tạo các biến trạng thái hệ thống và đồng hồ mô phỏng, tương ứng. Dòng 3 chỉ định tiêu chuẩn dừng. Ở đây, các dòng 4-7 được chạy lâu vì đồng hồ mô phỏng (tức là, simClock) nhỏ hơn ngưỡng được xác định trước(tức là, thời gian dừng). Những dòng này thu thập số liệu thống kê, thực hiện các sự kiện và nâng cấp mô phỏng cho thời gian sự kiện hiện tại.

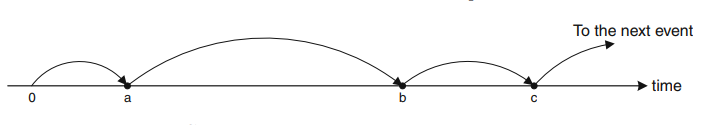
Chương trình 1.1 Sơ qua về vòng lặp xử lý sự kiện trong một mô phỏng theo thời gian.



**1.4.2 Mô phỏng theo hướng sự kiện**

Như tên cho thấy, mô phỏng theo hướng sự kiện được bắt đầu và được điều hành bởi một bộ các sự kiện. Một danh sách tất cả các sự kiện đã lên lịch thường được duy trì và cập nhật trong suốt quá trình mô phỏng.

Về mặt kỹ thuật, vòng lặp chính trong chương trình mô phỏng thực sự có trình tự thông qua danh sách này và xử lý một sự kiện khác cho đến khi danh sách trống hoặc tiêu chí dừng được thỏa mãn.



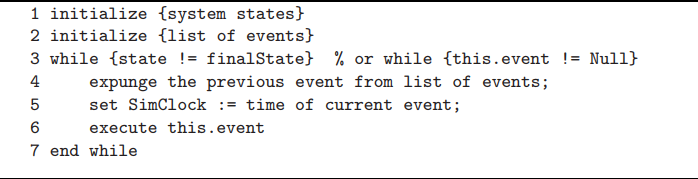
*Hình 1.3. Đồng hồ sự gia tăng trong mô phỏng theo hướng sự kiện.*

Cơ chế xử lý sự kiện được thể hiện bằng đồ thị trong hình 1.3,trong đó các sự kiện a, b và c được thực hiện theo thứ tự. Khoảng cách thời gian giữa hai sự kiện không cố định. Mô phỏng trước từ một sự kiện khác, như trái ngược với một khoảng thời gian khác trong một mô phỏng theo thời gian. Ngoại trừ cơ chế gia tăng thời gian, mô phỏng theo hướng sự kiện khá giống với cơ chế theo thời gian.

Trong mô phỏng theo hướng sự kiện, tất cả các sự kiện trong toàn bộ mô phỏng có thể không được tạo lúc khởi tạo. Khi mô phỏng gia tăng, một sự kiện có thể gây ra một hoặc nhiều sự kiện. Sự kiện mới thường được chèn vào chuỗi (tức là danh sách) các sự kiện được sắp xếp theo thứ tự thời gian. Mô phỏng theo hướng sự kiện bỏ qua các khoảng không hoạt động bằng cách thúc đẩy đồng hồ mô phỏng từ một thời gian sự kiện khác. Quá trình này diễn ra và tiếp tục cho đến khi tất cả các sự kiện được thực hiện hoặc cho đến khi hệ thống đạt đến trạng thái cụ thể (ví dụ: thời gian mô phỏng đạt đến giá trị được xác định trước). Bằng cách này, chúng chắc chắn cần một cách để thu thập một số thống kê hoặc trạng thái của hệ thống cho mục đích phân tích. Quá trình này thu thập thông tin có thể diễn ra ngay sau mỗi lần thực hiện sự kiện.Ngoài ra, nó có thể được thực hiện bằng cách sử dụng một thực thể chuyên ngành thu thập số liệu thống kê trong quá trình mô phỏng.

Ví dụ 1.2. Chương trình 1.2 cho thấy sơ qua của một chương trình mô phỏng theo sự kiện. Dòng 1 và 2 khởi tạo biến trạng thái hệ thống và danh sách của các sự kiện, tương ứng. Dòng 3 chỉ định tiêu chuẩn dừng. Các dòng 4–6 là được thực hiện giống như Dòng 3 trả về true. Tại đây, sự kiện được thực hiện trước bị xóa khỏi danh sách, đồng hồ mô phỏng được đặt thành thời gian được lập lịch của sự kiện hiện tại và sự kiện hiện tại được thực hiện. Trong vòng lặp như vậy,biến trạng thái hệ thống có thể được sửa đổi để nắm bắt những thay đổi xảy ra trong hệ thống theo sự kiện được thực hiện.

Chương trình 1.2 Sơ qua vòng lặp xử lý sự kiện trong một mô phỏng theo sự kiện



**1.5. Ví dụ về giả lập: Hệ thống xếp hàng đơn kênh**

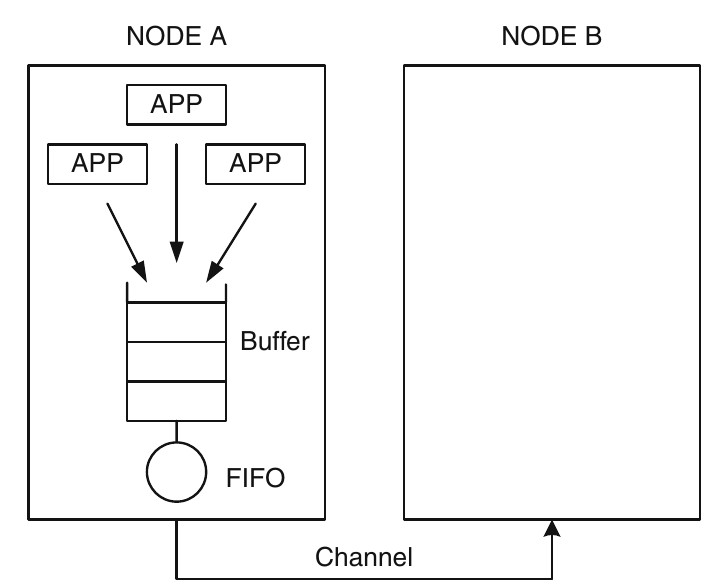
Phần này trình bày một mô phỏng của một hệ thống xếp hàng một kênh, làm ví dụ. Xem xét liên kết truyền thông có dây điểm-điểm có trong Hình 1.4. Để đơn giản, chúng tôi chỉ xem xét giao tiếp một chiều từ nút A đến nút B. Cụ thể, chúng tôi quan tâm đến hệ thống xếp hàng gói trong trạm tại nút A, trong đó một gói được lấy từ hàng đợi và được truyền đi (hoặc phục vụ) mỗi lần - thời gian truyền phụ thuộc vào băng thông hoặc dung lượng của liên kết.

Hơn nữa, chúng tôi giả định rằng các gói có thời gian liên kết đến sau một số phân bố xác suất, là không giới hạn và được tạo ngẫu nhiên từ một tập hợp các ứng dụng. Vì gói có thể có độ dài ngẫu nhiên và điều kiện của kênh có thể thay đổi, thời gian dịch vụ của mỗi gói cũng là ngẫu nhiên và sau một số phân bố xác suất. Trong trường hợp của chúng ta, nó được xác định là thời gian trôi qua từ thời điểm một gói tin được truyền tới thời điểm nó được nhận bởi nút B.

Tiếp theo, quy tắc xếp hàng được sử dụng tại nút A là First-In-First-Out (FIFO), tức là các gói được enqueued và truyền đi (được phục vụ) theo thứ tự đến của chúng. Để đơn giản, cơ chế xếp hàng tại nút B bị bỏ qua. Về mặt quảng cáo, để hệ thống ổn định, chúng tôi giả định rằng tỷ lệ đến thấp hơn tỷ lệ dịch vụ. Nếu không, hàng đợi sẽ tích tụ mà không bị ràng buộc.

***Thực thể***

1. Các thực thể chính trong mô phỏng này bao gồm:
2. • Máy chủ (tính khả dụng trung bình) với các thuộc tính nghỉvà bận,



*Hình 1.4. Hình ảnh thể hiện một hệ thống xếp hàng một kênh.*

• Các gói có thời gian đến và thuộc tính thời gian phục vụ, và

• Hàng đợi có thuộc tính trống và không trống.

***Nguồn***

Rõ ràng, tài nguyên duy nhất trong ví dụ này là thời gian truyền trong kênh.

***Biến trạng thái hệ thống và sự kiện***

* Hai biến trạng thái hệ thống:

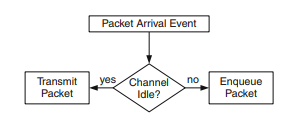
(i) num\_system là số lượng gói trong hệ thống, tức là số gói đang được phân phối và những gói đang chờ trong hàng đợi.

(ii) channel\_free là trạng thái của kênh (máy chủ) không hoạt động hoặc bận.

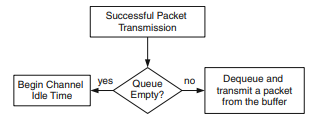
* Hai sự kiện:

(i) pkt\_arrival tương ứng với một sự kiện đến. Sự kiện này xảy ra khi một gói đến hàng đợi. Như thể hiện trong hình 1.5, một khi đã được kích hoạt, gói tin có thể đi trực tiếp đến dịch vụ hoặc đợi trong hàng đợi, tùy thuộc vào việc kênh có bận hay không.

(ii) pkt\_complete tương ứng với một sự kiện truyền gói thành công. Sự kiện này chỉ ra rằng một gói đã được nhận thành công bởi nút B. Khi hoàn thành, nút A bắt đầu truyền (phục vụ) một gói khác đang chờ trong hàng đợi. Nếu không có gói nào được gửi, kênh sẽ không hoạt động. Sơ đồ nợ của quá trình như vậy được thể hiện trong hình 1.6.



**Fig. 1.5.** Sự kiện packet tới



**Fig. 1.6.** Truyền packet thành công(dịch vụ thành công).

Hai yếu tố quan trọng khác trong mô phỏng theo hướng sự kiện là đồng hồ mô phỏng và danh sách sự kiện. Một đồng hồ mô phỏng duy trì thời gian mô phỏng hiện tại, như các tiến bộ mô phỏng. Danh sách sự kiện là một chuỗi các sự kiện được lên lịch (ví dụ: gói tin đến và truyền gói thành công) kết nối theo thứ tự thời gian. Một lần nữa, mô phỏng thực hiện một sự kiện sau khi một sự kiện khác xuống danh sách sự kiện và cập nhật đồng hồ mô phỏng dựa trên thời gian được xác định trong sự kiện đã thực hiện.

Thời gian chờ đợi trung bình là thời gian trung bình mà gói tin chi tiêu trong hàng đợi. Trong mô phỏng, chúng tôi xác định một biến toàn cầu theo dõi tổng thời gian tất cả các gói tin được truyền trong hàng đợi. Vào cuối mô phỏng, chúng tôi chia giá trị này cho tổng số gói được truyền để có được thời gian chờ đợi trung bình.

*Các biện pháp đo hiệu suất mô phỏng*

Ở đây, chúng tôi xem xét ba biện pháp hiệu suất sau đây là:

* Thời gian chờ đợi trung bình là thời gian trung bình mà gói tin chi tiêu trong hàng đợi. Trong mô phỏng, chúng tôi xác định một biến toàn cầu theo dõi tổng thời gian tất cả các gói tin được truyền trong hàng đợi. Vào cuối mô phỏng, chúng tôi chia giá trị này cho tổng số gói được truyền để có được thời gian chờ đợi trung bình.
* Độ trễ truyền gói tin trung bình là thời gian trung bình mà gói tin chi tiêu (từ khi nó đến lúc khởi hành) trong hệ thống. Đây là tổng thời gian của tất cả các gói dành cho hệ thống chia cho tổng số gói tin được truyền đi.
* Sử dụng máy chủ trung bình là thời gian phần trăm máy chủ bận. Trong quá trình mô phỏng, chúng tôi đo thời gian máy chủ bận. Vào cuối mô phỏng, chúng tôi chia thời gian bận này cho tổng thời gian mô phỏng và thu được mức sử dụng máy chủ trung bình.

Điều quan trọng cần lưu ý là tất cả các biện pháp trên là giá trị trung bình được thực hiện theo thời gian, ngụ ý rằng mô phỏng càng dài thì số liệu thống kê càng chính xác.

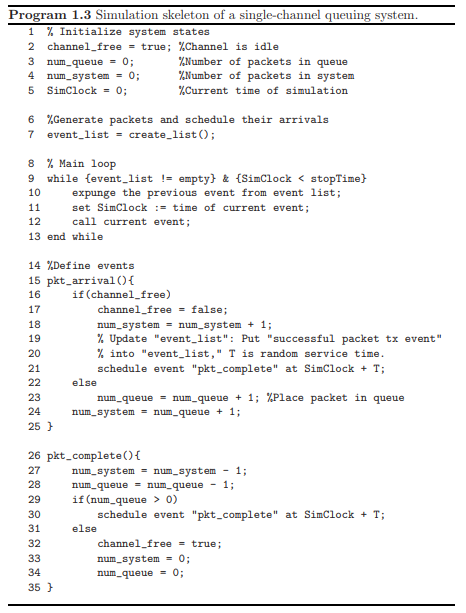
Program 1.3 cho thấy một bộ khung của chương trình mô phỏng có thể được sử dụng để thực hiện hệ thống xếp hàng một kênh được mô tả ở trên.

Chương trình bắt đầu với việc khởi tạo các biến trạng thái hệ thống như được định nghĩa ở trên. Ngoài ra, chúng tôi xác định num\_queue (Dòng 3) và num\_system (Dòng 4) để lưu trữ số gói đang chờ và số lượng tất cả các gói hiện có trong hệ thống (tức là cả hàng đợi và kênh), tương ứng. Biến SimClock cũng được khởi tạo bằng 0 ở đầu mô phỏng. Tiếp theo, Dòng 7 tạo ra một danh sách sự kiện bằng cách gọi thủ tục create\_list(). Chúng tôi giả định rằng chức năng này tự động tạo các gói và kết hợp từng gói với thời gian đến và dịch vụ ngẫu nhiên. Hơn nữa, chúng tôi giả định rằng event\_list ở đây được triển khai bằng cách sử dụng một số cấu trúc dữ liệu thích hợp thường cho biết loại sự kiện (đến hoặc hoàn thành) và dấu thời gian được liên kết (tức là thời gian đến và thời gian dịch vụ). Ban đầu, chỉ có các sự kiện đến được đưa vào event\_list.

Bây giờ chúng tôi xác định một vòng lặp chính liên tục kiểm tra xem mô phỏng có nên được chấm dứt hay không. Các tiêu chí dừng trong Dòng 9 là (1) danh sách sự kiện bị cạn kiệt và (2) đồng hồ mô phỏng đã đạt đến ngưỡng được xác định trước.



**Program 1.3** Mô phỏngkhái quát hệ thống xếp hàng một kênh.



Nếu không, các dòng 10-12 tiếp tục thực hiện sự kiện tiếp theo bằng cách gọi thủ tục pkt\_arrival () trong dòng 15-25 hoặc thủ tục pkt\_complete () trong dòng 26-35.

Các thủ tục pkt\_arrival () (dòng 15-25) kiểm tra xem các kênh (máy chủ)

là nhàn rỗi khi một gói đến. Nếu không hoạt động, kênh được đặt thành bận, và một sự kiện truyền gói thành công được chèn vào event\_list để thực hiện trong tương lai. Dấu thời gian được liên kết với sự kiện này bằng thời gian đồng hồ hiện tại (SimClock) cộng với thời gian dịch vụ được tạo ngẫu nhiên của gói (T). Nếu kênh bận, mặt khác, gói đơn giản được đặt trong hàng đợi có bộ đếm (num\_queue) được tăng lên bởi một đơn vị. Số lượng gói trong hệ thống cũng được cập nhật tương ứng.

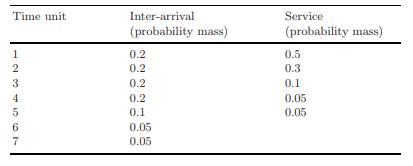
Khi SimClock tiến tới một sự kiện truyền gói thành công, thủ tục pkt\_complete () được thực thi (các dòng 26-35). Ở đây, số lượng gói trong hệ thống (num\_system) được cập nhật. Bộ đếm hàng đợi num\_queue bị giảm một đơn vị. Sau khi truyền gói thành công, nó cũng cần thiết để kiểm tra xem hàng đợi có trống không. Nếu không, gói đầu-of-the-line sẽ được phục vụ. Điều này được thực hiện bằng cách cho gói tin vào kênh và lên lịch để hoàn thành truyền vào thời gian SimClock + T. Tuy nhiên, nếu hàng đợi trống, kênh được đặt ở chế độ chờ và số lượng gói trong hàng đợi và hệ thống được đặt thành 0 .

Giả sử thời gian đến và thời gian phục vụ tuân thủ các hàm khối lượng xác suất được nêu trong Bảng 1.1. Bảng 1.2 cho thấy kết quả mô phỏng cho 10 gói. Thời gian đến và thời gian dịch vụ của mỗi gói được hiển thị trong cột đầu tiên và thứ hai, tương ứng. Cột thứ ba và thứ tư chỉ định thời gian mà gói tin đến và bắt đầu được phân phát. Cột ﬁ fth đại diện cho thời gian chờ gói tin, thời gian gói tin trong hàng đợi. Nó được tính là thời gian di chuyển giữa khi dịch vụ bắt đầu và khi gói tin đến. Cuối cùng, cột thứ sáu thể hiện độ trễ truyền gói tin, thời gian mà gói tin dành cho cả hàng đợi và kênh. Nó được tính như là tổng kết của thời gian chờ đợi và thời gian phục vụ.

Dựa trên kết quả trong Bảng 1.2, chúng tôi tính toán thời gian chờ trung bình và độ trễ truyền gói trung bình bằng cách tính trung bình các cột thứ sáu và thứ bảy (tức là thêm tất cả các giá trị và chia kết quả cho 10). Thời gian chờ trung bình và độ trễ truyền gói tin trung bình là 1.0 và

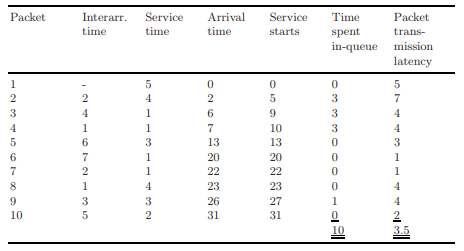
3,5 đơn vị thời gian, tương ứng.

**Table 1.1.** Xác suất ước lượng chức năng của thời gian đến và thời gian đáp ứng.



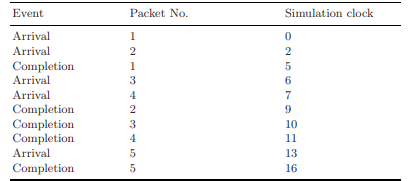


**Table 1.2.** Mô phỏng hệ thống xếp hàng một kênh.



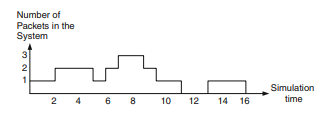
**Table 1.3. S**ự gia tăng của số lượng các gói tin trong hàng đợi theo thời gian.





Dựa trên thông tin trong Bảng 1.2, chúng tôi cũng trình bày trong Bảng 1.3 cách mỗi sự kiện cho các gói dữ liệu đầu tiên xảy ra theo thứ tự thời gian đối với Đồng hồ mô phỏng (SimClock).

Hình 1.7 mô tả sự tiến hóa của số lượng gói tin trong hàng đợi theo thời gian, cũng được thể hiện trong Bảng 1.3. Như thể hiện trong hình 1.7, ở các trường hợp khác nhau, số lượng gói trong các hệ thống di chuyển. Khi gói tin đầu tiên đang được truyền đi, một gói khác sẽ xuất hiện trong hàng đợi vào thời điểm 2. Số lượng gói tin trong hệ thống sẽ trở thành 2. Đó là, nó bao gồm gói đã được phục vụ cộng với gói vừa đến. Trong hình 1.7, sự kiện này gây ra một bước nhảy trong đồ thị tại thời điểm 2. Tại thời điểm 5, khi gói đầu tiên được nhận thành công tại nút B, gói tiếp theo trong hàng đợi được truyền đi. Do đó, biểu đồ sẽ giảm xuống



*Hình 1.7. Số gói tin trong hệ thống theo thời gian*

cấp 1, chỉ ra rằng chỉ có một gói trong hệ thống. Động lực này

tiếp tục cho đến khi tất cả các gói được truyền đi. Dựa trên hình 1.7, giá trị trung bình

sử dụng máy chủ có thể được tính từ tỷ lệ thời gian máy chủ

đang được sử dụng và thời gian mô phỏng là 14/16 = 0,875 trong trường hợp này.

**1.6 Tổng kết chương**

Mạng máy tính là một hệ thống phức tạp đòi hỏi phải có một cách học và thực hành cẩn thận trong thiết kế và thực hiện. Mô phỏng, được coi là một trong những công cụ phân tích hiệu suất mạnh nhất, thường được sử dụng trong việc thực hiện một điều trị như vậy để bổ sung cho các công cụ phân tích

.

Chương này tập trung chủ yếu vào mô phỏng phụ thuộc vào thời gian, tiến bộ trong một miền thời gian. Mô phỏng phụ thuộc vào thời gian có thể được chia thành hai loại mèo. Mô phỏng theo thời gian thúc đẩy mô phỏng theo thời gian tương tác, trong khi mô phỏng theo hướng sự kiện diễn ra từ sự kiện này đến sự kiện khác. NS2 là một công cụ mô phỏng theo hướng sự kiện. Thiết kế mô hình mô phỏng theo hướng sự kiện sử dụng NS2 là chủ đề của phần còn lại của cuốn sách.