**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

A red and yellow sign

Description automatically generated with low confidence

**BÁO CÁO BÀI TẬP LỚN MẠNG THẾ HỆ SAU**

**Đề tài:** **Phân tích bài báo “Network Slicing for mMTC and URLLC Using Software-Defined Networking with P4 Switches”**

**Giảng viên:** PGS.TS. Trương Thị Diệu Linh

TS. Đinh Thị Hà Ly

**Nhóm 8**

|  |  |
| --- | --- |
| **Họ và tên** | **MSSV** |
| Đỗ Trọng Đạt | 20190164 |
| Đỗ Ngọc Tuân | 20194398 |
| Nguyễn Tiến Nam | 20194337 |

**Mục lục**

[CHƯƠNG 1. GIỚI THIỆU 2](#_Toc142169508)

[CHƯƠNG 2. KIẾN TRÚC VÀ QUY TRÌNH HOẠT ĐỘNG CỦA SDNPS 6](#_Toc142169509)

[CHƯƠNG 3. ĐỊNH DẠNG GÓI DỮ LIỆU SDNPS 8](#_Toc142169510)

[CHƯƠNG 4. MÃ GIẢ CỦA SDNPS 11](#_Toc142169511)

[CHƯƠNG 5. ĐÁNH GIÁ HIỆU SUẤT 13](#_Toc142169512)

[CHƯƠNG 6. TRIỂN KHAI CHƯƠNG TRÌNH 17](#_Toc142169513)

# 

# **CHƯƠNG 1. GIỚI THIỆU**

mMTC và URLLC là hai dịch vụ chính trong mạng không dây di động thế hệ thứ năm (5G). Các mạng này đã được phát triển với yêu cầu chất lượng dịch vụ cao: khả năng mở rộng cho mMTC, độ tin cậy và độ trễ thấp cho URLLC. Network slicing đóng một vai trò quan trọng trong việc hỗ trợ các yêu cầu riêng biệt của các dịch vụ này.

Software-defined networking (SDN), một công nghệ đầy hứa hẹn để phần mềm hóa mạng, phân tách vật lý network control plane khỏi data plane bằng cách điều khiển tập trung các switches bằng SDN controller.

Paper này đề xuất một framework SDN với P4 switches và xác định định dạng gói chứa dữ liệu đo mạng từ xa trong băng tần để hỗ trợ đồng thời mMTC và URLLC trong mạng 5G. Phương pháp này vừa đáp ứng các yêu cầu của mMTC và URLLC, vừa giảm tải cho bộ điều khiển SDN.

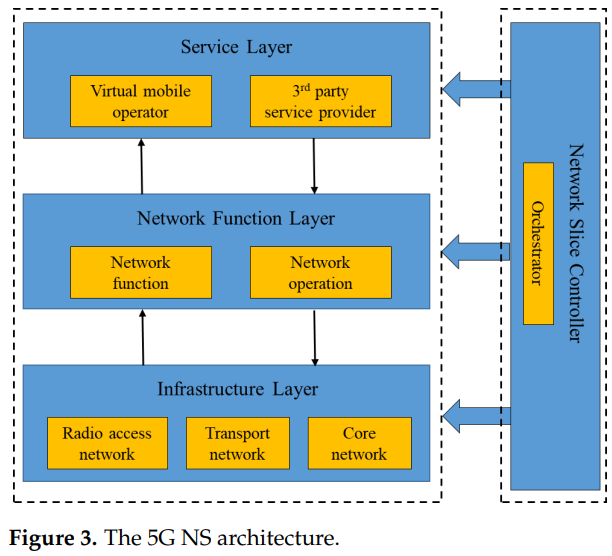
P4 (programming protocol– independent packet processor) là một công nghệ lập trình switch tiên tiến cung cấp chức năng chuyển tiếp trạng thái nâng cao và hiển thị trạng thái liên tục trên SDN data plane.

Network slicing (NS) là một công nghệ cho phép tích hợp nhiều dịch vụ khác nhau trên một cơ sở hạ tầng mạng chung, được kỳ vọng sẽ được sử dụng rộng rãi trong các mạng trong tương lai.

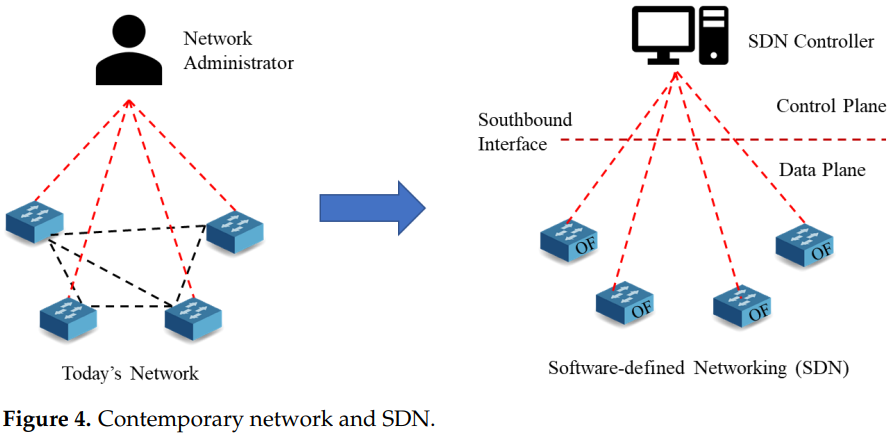
Trong NS, ảo hóa được sử dụng để phân đoạn một mạng 5G vật lý thành nhiều mạng logic end-to-end (E2E) phục vụ các trường hợp sử dụng khác nhau.

Bằng cách sử dụng network slices, các công ty có thể yêu cầu các nhà cung cấp dịch vụ internet cung cấp một mạng 5G tùy chỉnh.

Hình dưới đây mô tả kiến trúc NS 5G dưới dạng các khối. Khối trái đại diện cho việc thực hiện phân đoạn thực tế, và khối phải đại diện cho quản lý và cấu hình phân đoạn. Khối thực hiện bên trái là một kiến trúc đa tầng bao gồm lớp dịch vụ, lớp chức năng mạng và lớp cơ sở hạ tầng. Khối điều khiển bên phải là một đơn vị mạng được xử lý trung tâm, gọi là bộ điều khiển phân đoạn mạng, theo dõi và quản lý các chức năng giữa hai trong số ba lớp trong khối trái để hiệu quả điều phối nhiều phân đoạn mạng đồng thời.



Sự khác biệt lớn nhất giữa các hoạt động mạng đương đại và SDN là sự tạo ra một mặt phẳng điều khiển ảo có thể thực hiện các quyết định quản lý thông minh cho các chức năng mạng, khắc phục khoảng cách giữa việc cung cấp dịch vụ và quản lý mạng. Với SDN, điều khiển mạng có thể được đạt được trực tiếp và có thể lập trình thông qua một giao diện tiêu chuẩn hóa. Ví dụ, giao thức OpenFlow thường được sử dụng định nghĩa một cơ chế giao tiếp giữa bộ điều khiển trong mặt phẳng điều khiển và các thành phần chuyển tiếp trong mặt phẳng dữ liệu.



Bài báo này thiết kế một SDN framework với các switch P4 và định nghĩa một packet format là INT (in-band network telemetry) để đồng thời hỗ trợ mMTC và URLLC trong 5G.

Giả sử tất cả các switch trong data plane đều hỗ trợ P4 và chia các luật forwarding của các switch P4 thành hai phần: một cho các dịch vụ mMTC và một cho các dịch vụ URLLC. Trong SDNPS, INT cho phép các gói dữ liệu mang thông tin trạng thái mạng sao cho các switch P4 trên forwarding path có thể cập nhật thông tin trạng thái mạng, giảm tải điều khiển trên bộ điều khiển SDN.

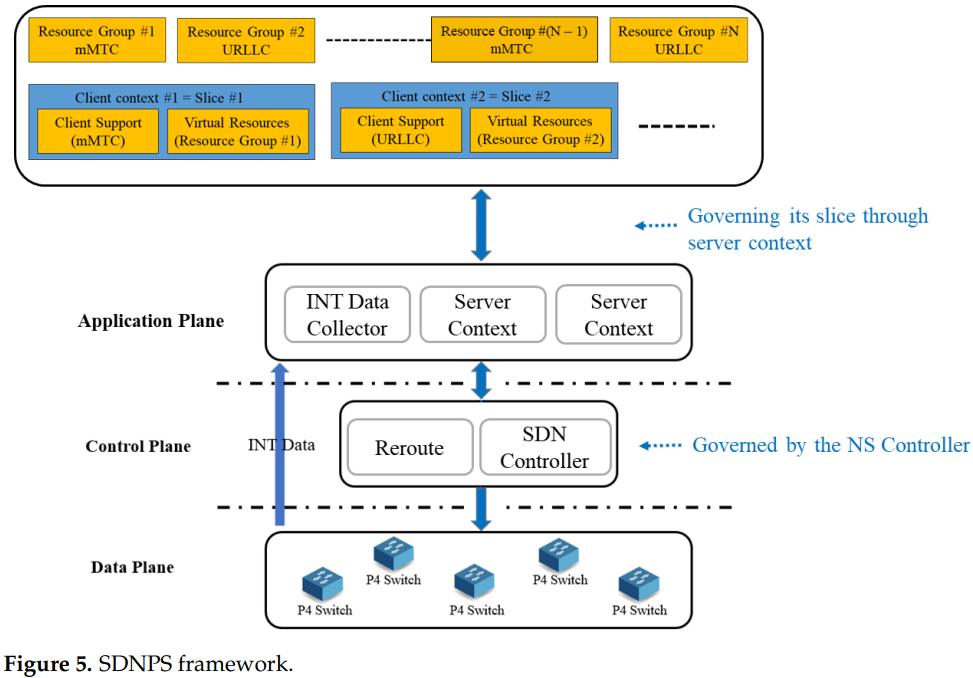
# **CHƯƠNG 2. KIẾN TRÚC VÀ QUY TRÌNH HOẠT ĐỘNG CỦA SDNPS**

Các đóng góp chính của SDNPS được đề xuất là thiết kế một khung SDN với các switch P4 và định nghĩa định dạng gói tin chứa dữ liệu INT (in-band network telemetry) để đồng thời hỗ trợ lưu lượng IoT khổng lồ và URLLC trong hạt nhân của các mảnh mạng 5G. Do đó, các yêu cầu dịch vụ cực kỳ khác biệt của mMTC và URLLC có thể được đáp ứng và tải điều khiển trên bộ điều khiển SDN có thể được giảm bớt. Giả định rằng tất cả các switch trong mặt dữ liệu đều hỗ trợ P4 và chia các luật chuyển tiếp của các switch P4 thành hai phần: một cho các dịch vụ mMTC và một cho các dịch vụ URLLC. Trong SDNPS, INT cho phép các gói dữ liệu mang thông tin trạng thái mạng sao cho các switch P4 trên đường chuyển tiếp có thể cập nhật thông tin trạng thái được bao gồm trước khi chuyển tiếp các gói tin, giảm tải điều khiển trên bộ điều khiển SDN.

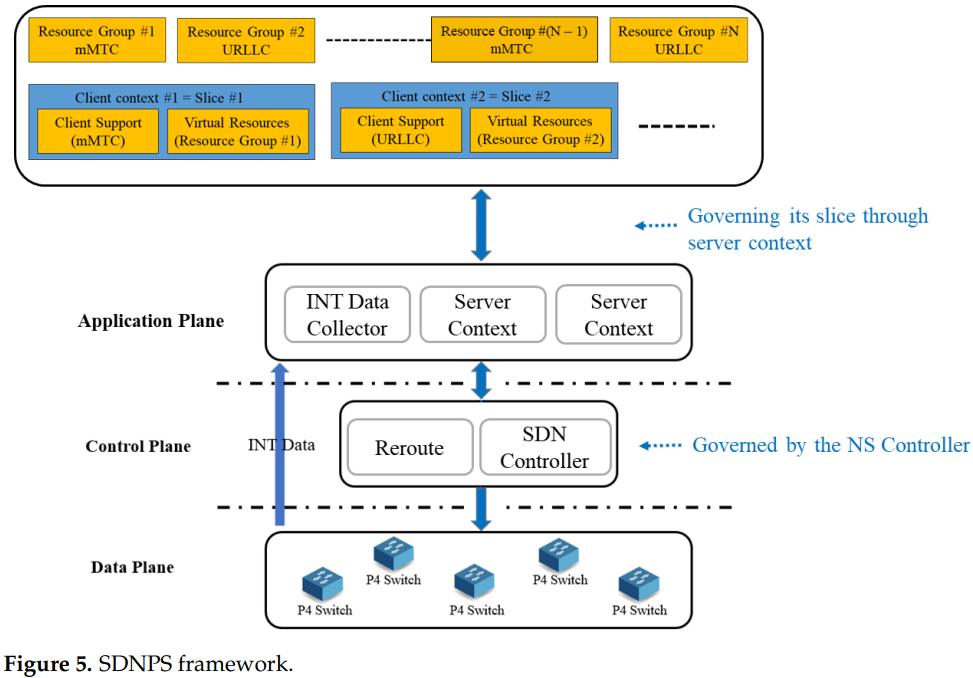
**Kiến trúc chi tiết**

Các client services được chia thành hai loại - mMTC và URLLC - được gán cho các mảnh network slice khác nhau. Mỗi mảnh mạng là độc lập logic với nhau về tài nguyên mạng, bao gồm các thiết bị, truy cập, truyền tải và hạt nhân mạng.

SDN bao gồm ba tầng: ứng dụng, điều khiển và dữ liệu.



Với INT, mỗi switch P4 trên một đường chuyển tiếp trong mặt dữ liệu có thể ghi lại trực tiếp trạng thái mạng, bao gồm sự chiếm dụng hàng đợi, thông lượng liên kết và độ trễ xử lý từ phần tiêu đề INT được thêm vào bởi switch hiện tại trước khi chuyển tiếp. Khi một gói tin dữ liệu mang dữ liệu INT đến switch tiếp theo, dữ liệu INT được nhân bản và gửi lên tới mô-đun thu thập dữ liệu INT trong tầng ứng dụng để giám sát trạng thái của mỗi switch trên một đường chuyển tiếp. Bộ điều khiển NS chịu trách nhiệm thay đổi các đường chuyển tiếp và thông báo cho mô-đun đổi hướng về cấu hình lại đường chuyển tiếp. Các switch P4 được kích hoạt để cập nhật các luật chuyển tiếp khi cần thiết.



# **CHƯƠNG 3. ĐỊNH DẠNG GÓI DỮ LIỆU SDNPS**

Đặc điểm INT là thu thập và báo cáo trạng thái mạng trong data plane mà không can thiệp vào control plane. Trong INT, thông tin trạng thái mạng (ví dụ như sự chiếm dụng hàng đợi của switch, tốc độ đầu ra gói tin và độ trễ xử lý) được đóng gói vào một gói tin dữ liệu hoặc gói điều khiển được xác định trước bởi mỗi switch P4 trên một đường truyền dữ liệu. Gói tin mang thông tin trạng thái mạng được trích xuất và gửi lên tầng ứng dụng để phân tích thêm trước khi được chuyển tiếp đến đích. Do đó, tải trao đổi thông tin hai chiều giữa control plane và data plane được giảm đáng kể và thông tin trạng thái mạng chính xác mới nhất trong mặt dữ liệu có thể được thu thập và xử lý càng nhanh càng tốt.

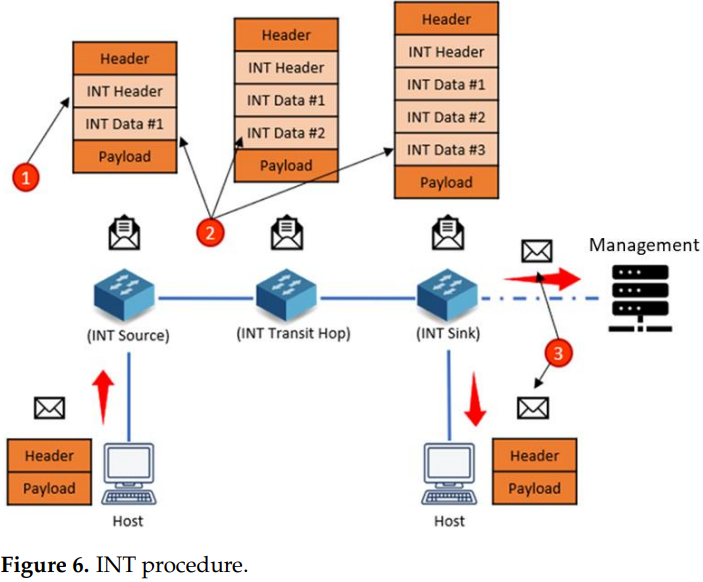
**Quy trình INT**

Quy trình INT được chia thành ba bước:

1. Khi gói dữ liệu từ máy chủ nguồn đi vào mặt phẳng dữ liệu, INT source có thể thêm INT header để báo cho các switch P4 tiếp theo biết thông tin (được gọi là ‘INT data’) mà chúng nên ghi

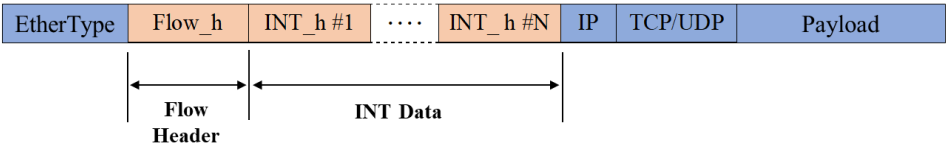
2. INT transit hop có thể ghi dữ liệu INT của chính nó vào trường được chỉ định theo nội dung được đưa ra bởi phần INT header

3. Khi một gói tin dữ liệu mang INT data đến INT sink, tất cả các dữ liệu INT đã được ghi lại trên một đường chuyển tiếp được trích xuất và báo cáo cho mô-đun thu thập dữ liệu. Sau đó, gói tin dữ liệu ban đầu được chuyển tiếp đến host đích

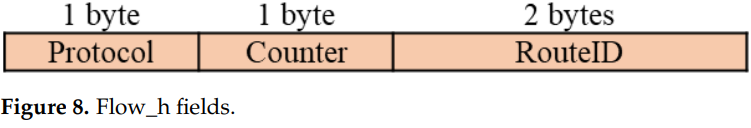


Trường EtherType là 0×0801 xác định đó có phải là switch P4.

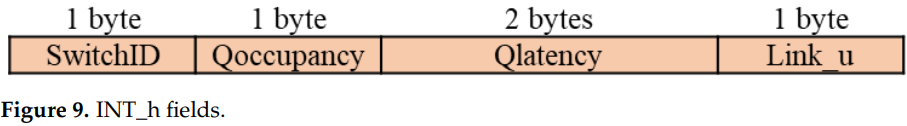
Flow\_h (Flow Header) dùng để thông báo cho các switch P4 về việc liệu packet có chứa INT data hay không vì INT data cho mỗi bộ chuyển mạch P4 chỉ được gửi định kỳ chứ không phải với mọi gói dữ liệu.



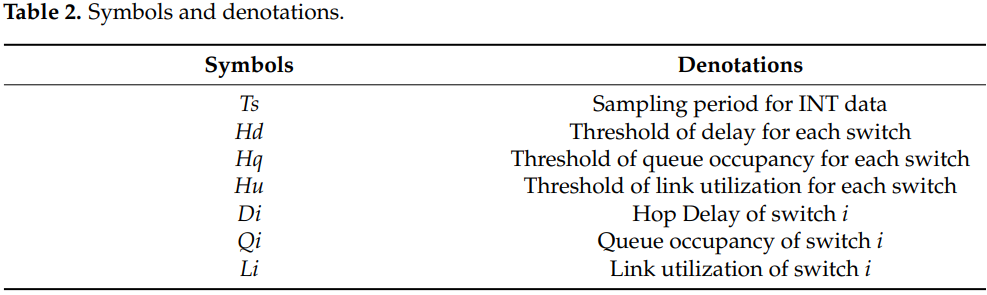
Định dạng của Flow\_h là 4 byte. Trường Protocol được đặt thành 0×00 và 0×01 xác định xem dữ liệu sau Flow Header tương ứng là IP header hay INT data. Trường Counter ghi lại số switch P4 mà một gói đi qua. Mỗi RouteID là duy nhất và biểu thị một forwarding path; các gói chứa INT data bao gồm các dữ liệu này từ mỗi P4 switch dọc theo đường đi được chỉ định bởi RouteID.



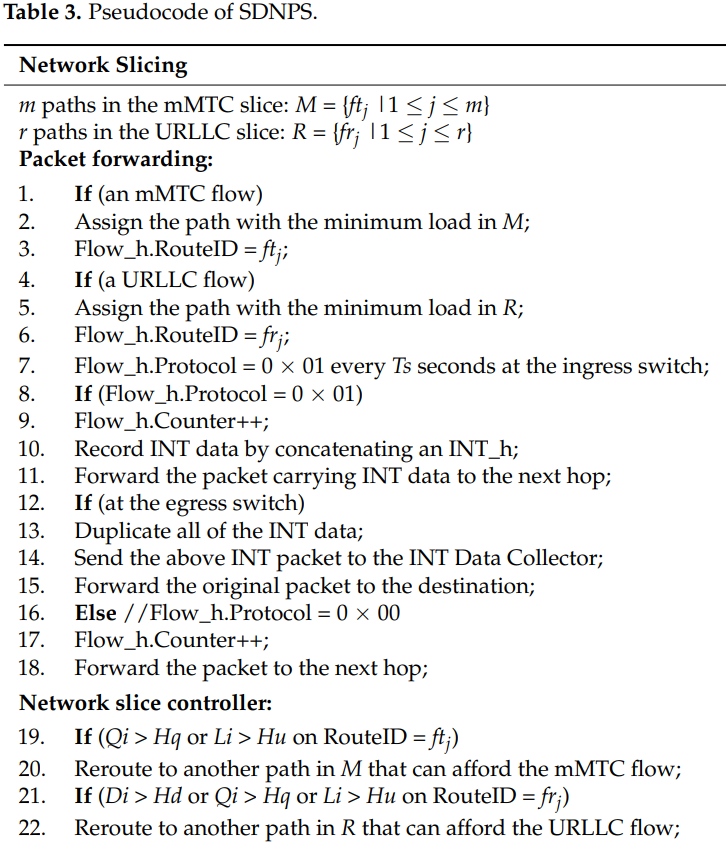
Định dạng của INT data 5 byte (ký hiệu là INT\_h) cho một switch P4, bao gồm SwitchID (1 byte), Qoccupancy (1 byte), Qlatency (2 byte) và Link\_utilization (1 byte). Trường SwitchID được sử dụng để ghi lại ID của switch, trường Qlatency được sử dụng để ghi lại thời gian xử lý, trường Qoccupancy được sử dụng để ghi số packet trong queue của switch và trường Link\_u được sử dụng để ghi lại mức sử dụng băng thông của link.



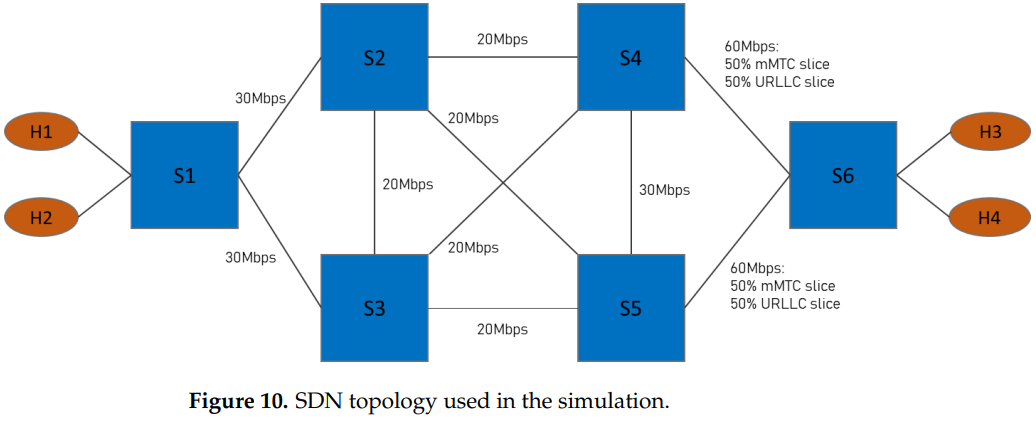
# **CHƯƠNG 4. MÃ GIẢ CỦA SDNPS**



Ký hiệu được sử dụng bởi SDNPS như được tóm tắt trong Bảng 2, Ts đại diện cho khoảng thời gian lấy mẫu cho dữ liệu INT. Hd, Hq và Hu là ngưỡng độ trễ, chiếm dụng hàng đợi và sử dụng liên kết, nếu vượt qua ngưỡng nào đó tại bất kỳ switch nào sẽ gây ra thay đổi đường chuyển tiếp trong cùng một lát cắt vì một điểm nghẽn tại bất kỳ switch nào trên đường chuyển tiếp dẫn đến suy giảm hiệu suất nghiêm trọng cho đường chuyển tiếp đó. Di, Qi và Li đại diện cho độ trễ xử lý, chiếm dụng hàng đợi và sử dụng liên kết của switch i

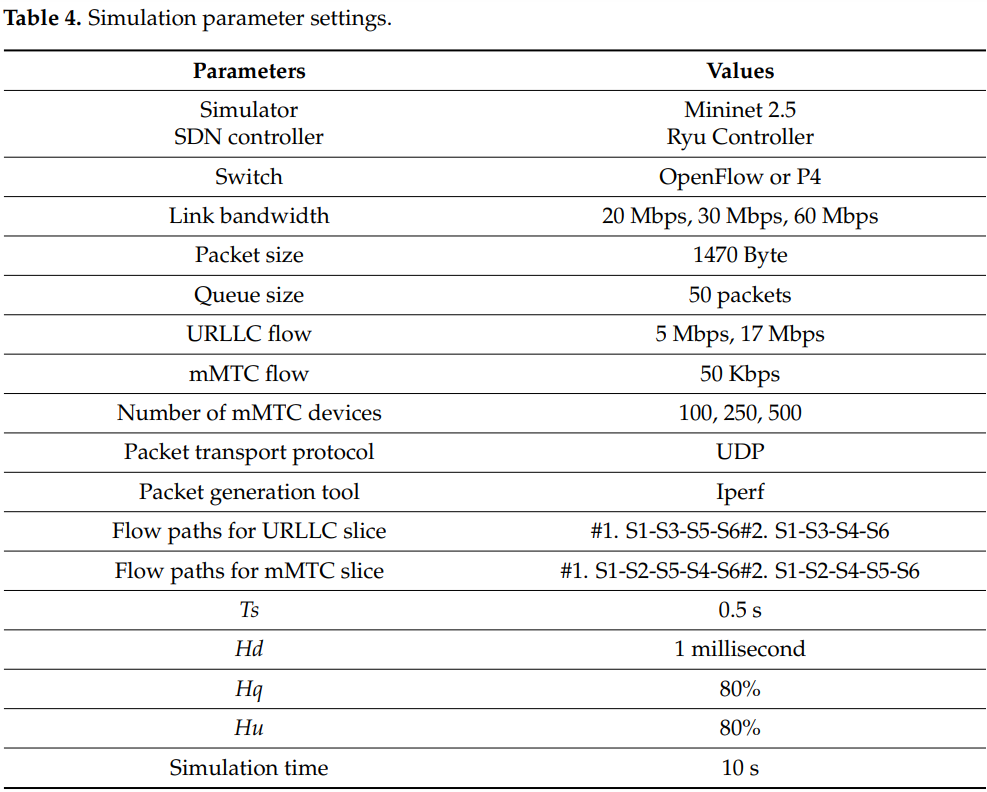


**CHƯƠNG 5. ĐÁNH GIÁ HIỆU SUẤT**

Băng thông của mỗi liên kết giữa hai switch là 20, 30 hoặc 60 Mbps. Kích thước hàng đợi của mỗi switch là 50 gói tin. Công cụ Iperf được sử dụng để tạo ra các luồng UDP ở tốc độ dữ liệu ổn định là 50 Kbps cho mỗi thiết bị loại máy, và số lượng thiết bị loại máy thay đổi từ 100 đến 500. Hai luồng URLLC-UDP được truyền với tốc độ là 5 và 17 Mbps. Các máy nguồn và đích của các luồng mMTC là H1 và H3; các máy nguồn và đích của các luồng URLLC lần lượt là H2 và H4. 

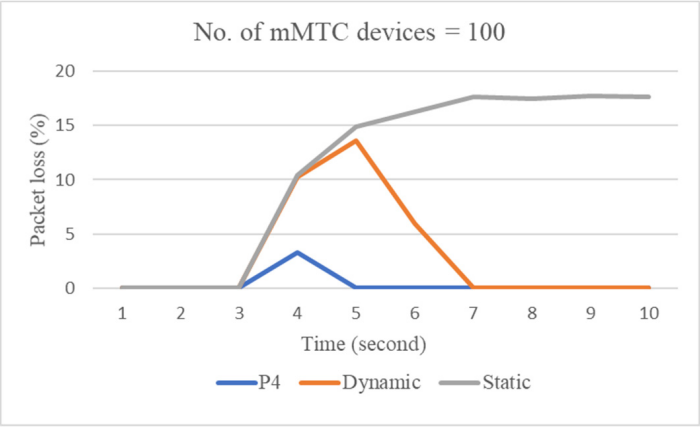
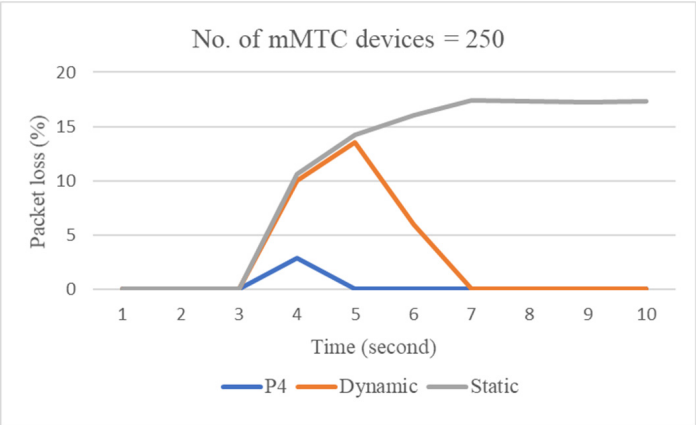
**Cấu trúc mạng SDN được sử dụng để mô phỏng**

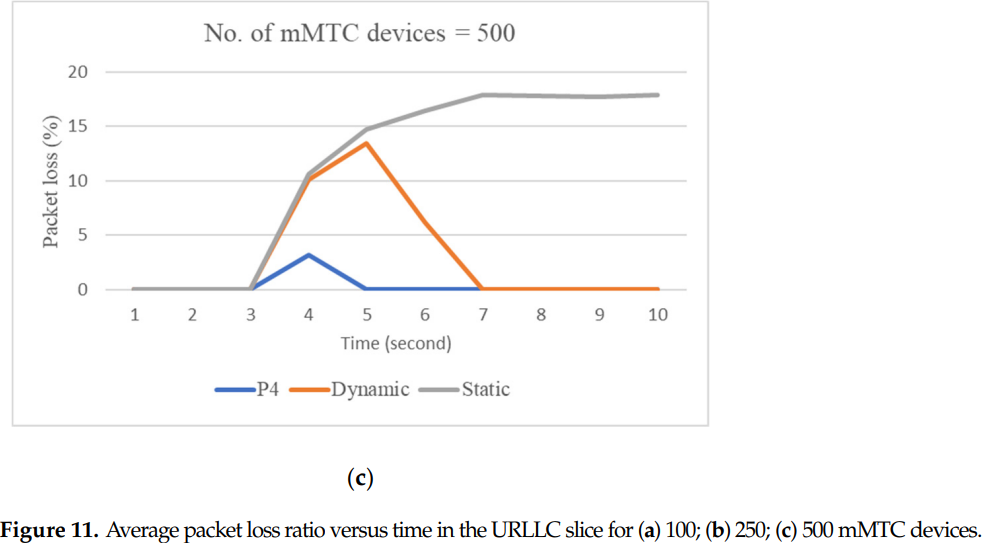
URLLC slice được chỉ định hai tuyến đường chuyển tiếp S1-S3-S5-S6 và S1-S3-S4-S6 với số bước nhảy ít nhất. Để đạt được mật độ kết nối cao thay vì độ trễ, lát cắt mMTC được chỉ định hai đường chuyển tiếp S1-S2-S5-S4-S6 và S1-S2-S4-S5-S6. Ban đầu, luồng dữ liệu đầu tiên trong bất kỳ lát cắt nào luôn được sử dụng, và luồng dữ liệu thứ hai được sử dụng nếu đường chuyển tiếp đầu tiên có suy giảm chất lượng dịch vụ. Khoảng thời gian lấy mẫu cho dữ liệu INT là 0,5 giây. Thời gian mô phỏng là 10 giây. Trong suốt thời gian mô phỏng, một luồng URLLC với tốc độ 5 Mbps được tạo ra ban đầu, và một luồng URLLC khác với tốc độ 17 Mbps được tạo ra sau 3 giây; 100, 250 hoặc 500 luồng mMTC được tạo ra đồng thời từ đầu đến cuối quá trình mô phỏng.



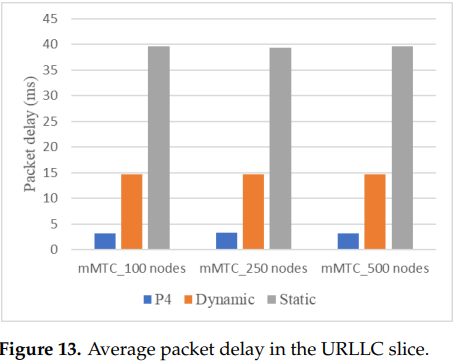
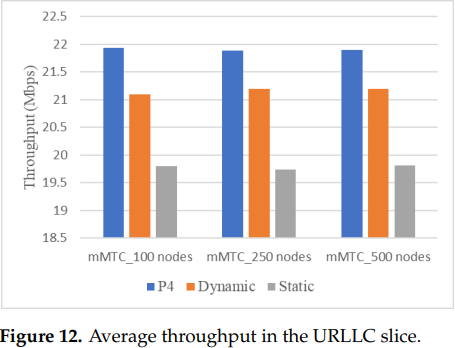
**Kết quả mô phỏng với URLLC Slice**

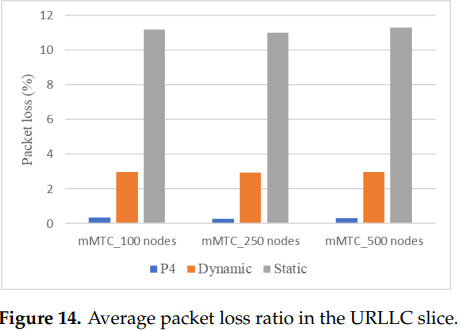
Tỷ lệ mất gói trung bình với SDNPS (đường cong màu xanh) luôn giảm xuống gần 0 nhanh hơn sau khi qua ba giây so với SDN truyền thống với định tuyến tĩnh (đường cong màu xám) hoặc định tuyến động (đường cong màu cam). Kết quả này xảy ra vì với SDNPS, dữ liệu INT cho mỗi switch được mang bởi một gói tin dữ liệu mỗi 0,5 giây, được trích xuất tại switch đích và gửi đến bộ điều khiển lát cắt mạng, cho phép phản hồi nhanh chóng cho tắc nghẽn mạng. Do đó, bộ điều khiển NS nhanh chóng quyết định chuyển hướng luồng URLLC 17 Mbps sang đường chuyển tiếp thứ hai (tức là S1-S3-S4-S6) trong lát cắt URLLC.



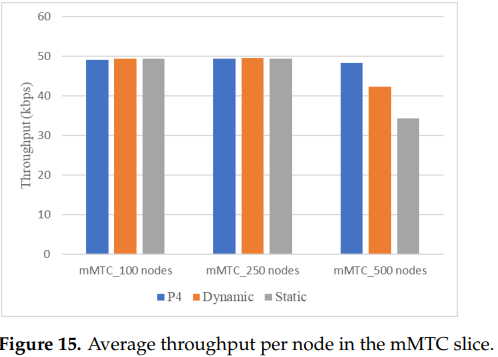
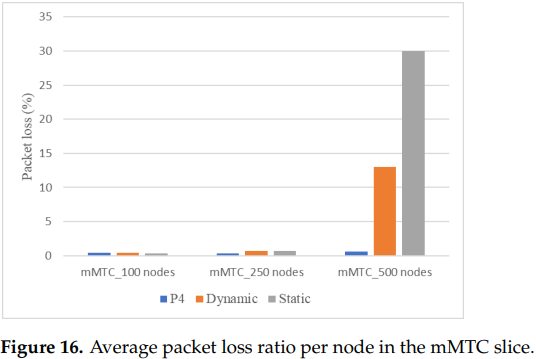


Với SDN Dynamic, cập nhật quy tắc chuyển tiếp cho các switch yêu cầu trao đổi thông tin đáng kể giữa bộ điều khiển SDN với các switch OpenFlow; do đó, định tuyến lại đòi hỏi nhiều thời gian hơn. Với SDN Static, tỷ lệ mất gói trung bình tăng liên tục sau giây thứ 3 vì các quy tắc chuyển tiếp không thay đổi trong định tuyến tĩnh trừ khi đường dẫn hiện tại bị lỗi.





**Kết quả mô phỏng với mMTC Slice**

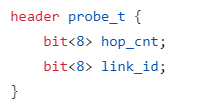
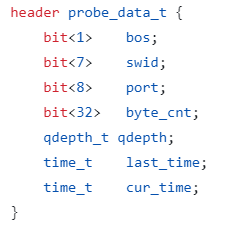
**** ****

**CHƯƠNG 6. TRIỂN KHAI CHƯƠNG TRÌNH**

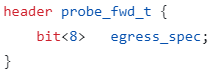
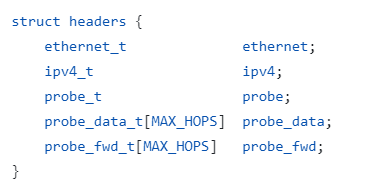
* Ngôn ngữ lập trình switch: P4
* Ngôn ngữ lập trình controller: python
* SDN API: P4runtime
* Đã hoàn thành việc triển khai gói tin INT data
* Chưa hoàn thành việc gửi gói tin từ switch lên controller

**Triển khai gói tin INT data**

Flow header INT data

Port forwarding data

Link ID: URLLC là 0 và 2, mMTC là 1 và 3

Threshold:

Link: 5mbps

Độ dài queue: 5

