TCN 68 - 224: 2004

GIAO THỨC KẾT NỐI GIỮA MẠNG GSM GPRS VÀ MẠNG INTERNET (GIAO THỨC IP)

YÊU CẦU KỸ THUẬT

INTERCONNECTING PROTOCOL BETWEEN GSM GPRS NETWORK AND INTERNET (IP PROTOCOL)

TECHNICAL REQUIREMENTS

MỤC LỤC

CONTENTS

LỜI NÓI ĐẦU

Tiêu chuẩn Ngành TCN 68 - 224: 2004 **"Giao thức kết nối giữa mạng GSM GPRS và mạng Internet (Giao thức IP) – Yêu cầu kỹ thuật"** được xây dựng trên cơ sở chấp thuận nguyên vẹn các yêu cầu kỹ thuật của tài liệu IETF RFC 791 (1981) của Nhóm đặc trách về kỹ thuật Internet (IETF).

Tiêu chuẩn Ngành TCN 68 - 224: 2004 do Viện Khoa học Kỹ thuật Bưu điện (RIPT) biên soạn theo đề nghị của Vụ Khoa học - Công nghệ và được ban hành theo Quyết định số 33/2004/QĐ-BBCVT ngày 29/7/2004 của Bộ trưởng Bộ Bưu chính, Viễn thông.

Tiêu chuẩn Ngành TCN 68 - 224: 2004 được ban hành dưới dạng song ngữ (tiếng Việt và tiếng Anh). Trong trường hợp có tranh chấp về cách hiểu do biên dịch, bản tiếng Việt được áp dụng.

VỤ KHOA HỌC - CÔNG NGHỆ

GIAO THỨC KẾT NỐI GIỮA MẠNG GSM GPRS VÀ MẠNG INTERNET (GIAO THỨC IP) YÊU CẦU KỸ THUẬT

(Ban hành kèm theo Quyết định số 33/2004/QĐ-BBCVT ngày 29/7/2004 của Bộ trưởng Bộ Bưu chính, Viễn thông)

1. Mục tiêu và phạm vi

1.1. Mục tiêu

Tiêu chuẩn này quy định những yêu cầu kỹ thuật thiết yếu đối với giao thức kết nối giữa các mạng GSM GPRS và mạng Internet (giao thức internet – IP), nhằm đảm bảo khả năng kết nối, phối hợp hoạt động hiệu quả giữa các mạng GSM GPRS và mạng Internet, phục vụ công tác quản lý kết nối mạng của các doanh nghiệp.

Giao thức internet được thiết kế để dùng trong các hệ thống liên kết của các mạng truyền thông máy tính chuyển mạch gói. Một hệ thống như thế được gọi là một "catenet". Giao thức internet giúp cho việc truyền các khối dữ liệu, được gọi là các gói tin, từ các nguồn đến các đích, trong đó các nguồn và các đích là các máy chủ được nhận dạng theo các địa chỉ có độ dài cố định. Giao thức internet cũng cho phép phân đoạn và tái lắp ráp các gói tin dài, nếu cần thiết, để truyền qua các mạng "gói nhỏ".

1.2. Phạm vi

Giao thức internet được giới hạn cụ thể trong phạm vi cung cấp các chức năng cần thiết cho việc phân phát một gói các bit (một gói tin internet) từ một nguồn tới một đích trên một hệ thống liên kết các mạng. Không có các cơ chế làm tăng độ tin cậy của dữ liệu đầu cuối - đầu cuối, điều khiển luồng, sắp xếp theo trình tự, hoặc các dịch vụ khác thường thấy trong các giao thức máy chủ - máy chủ. Giao thức internet có thể sử dụng các dịch vụ của các mạng đang hỗ trợ nó để cung cấp nhiều loại dịch vụ và nhiều chất lượng dịch vụ khác nhau.

2. Tài liệu tham khảo

[1] ETSI TS 101 348 V7.3.0 (3/2001), "Digital cellular telecommunications system (phase 2+); General Packet Radio Service (GPRS); Interwoking between the Public Land Mobile Network (PLMN) supporting GPRS and Packet Data Networks (PDN) (3GPP TS 09.61 version 7.3.0 Release 1998)".

[2] IETF RFC 791 (1981): "Internet protocol" (STD5).

3. Thuật ngữ

1822

Báo cáo BBN 1822, "Đặc tả về tính liên kết của một máy chủ và một IMP". Đặc tả về giao diện giữa một máy chủ và ARPANET.

Mào đầu ARPANET

Thông tin điều khiển trong một bản tin của mạng ARPANET tại giao diện máy chủ - IMP.

Bản tin của mạng ARPANET

Đơn vị truyền giữa một máy chủ và một IMP trong mạng ARPANET. Kích cỡ tối đa là khoảng 1012 octet (8096 bit).

Gói ARPANET

Một đơn vị truyền được sử dụng bên trong mạng ARPANET giữa các IMP. Kích cỡ tối đa là khoảng 126 octet (1008 bit).

Đích

Địa chỉ đích, một trường của phần mào đầu internet.

DF

Bit không phân đoạn được mang trong trường các cờ.

Các cờ

Một trường của phần mào đầu internet mang nhiều loại cờ điều khiển.

Độ dịch đoạn

Một trường của phần mào đầu internet cho biết một đoạn ở chỗ nào trong gói tin internet.

GGP

Giao thức Cổng - Cổng, giao thức được sử dụng chủ yếu giữa các cổng để điều khiển việc định tuyến và các chức năng cổng khác.

Phần mào đầu

Thông tin điều khiển ở phần đầu của một bản tin, đoạn, gói tin, gói hoặc khối dữ liệu.

ICMP

Giao thức bản tin điều khiển internet, được thực thi trong mô-đun internet, ICMP được sử dụng từ các cổng tới các máy chủ và giữa các máy chủ để thông báo các lỗi và đưa ra các đề xuất định tuyến.

Nhận dạng

Một trường của phần mào đầu internet mang giá trị nhận dạng do bên gửi gán để trợ giúp việc lắp ráp các đoạn của một gói tin.

IHL

Trường Độ dài của phần mào đầu internet trong phần mào đầu internet cho biết độ dài của phần mào đầu internet được tính theo đơn vị từ 32 bit.

IMP

Bộ xử lý bản tin của giao diện, bộ chuyển gói của mạng ARPANET.

Địa chỉ Internet

Một địa chỉ đích hoặc nguồn 4 octet (32 bit) gồm có một trường mạng và một trường địa chỉ cục bộ.

Gói tin internet

Đơn vị dữ liệu được trao đổi giữa một cặp mô-đun internet (bao gồm cả phần mào đầu internet).

Đoạn internet

Một phần dữ liệu của một gói tin internet với một phần mào đầu internet.

Địa chỉ cục bộ

Địa chỉ của một máy chủ trong phạm vi một mạng. Việc ánh xạ trên thực tế một địa chỉ cục bộ internet lên các địa chỉ máy chủ trong một mạng là hoàn toàn phổ biến, kể cả các ánh xạ nhiều địa chỉ cục bộ vào một địa chỉ máy chủ.

MF

Cờ chỉ báo còn đoạn được mang trong trường các cờ của phần mào đầu internet.

Mô-đun

Một sự thực thi một giao thức hoặc các thủ tục khác, thường là bằng phần mềm.

Cờ chỉ báo còn đoạn

Một cờ, được mang trong trường các cờ của phần mào đầu internet, cho biết gói tin internet này có chứa phần cuối của một gói tin internet hay không.

NFB

Số các khối dữ liệu của một đoạn internet. Tức là, độ dài của một phần dữ liệu được đo theo đơn vị là 8 octet.

Octet

Một byte 8 bit.

Các tùy chọn

Trường Các tùy chọn của phần mào đầu internet có thể bao gồm vài tùy chọn, và mỗi tùy chọn có thể có chiều dài là một vài octet.

Đệm

Trường Đệm của phần mào đầu internet được sử dụng để đảm bảo rằng dữ liệu bắt đầu trên biên từ 32 bit. Đệm bằng 0.

Giao thức

Ký hiệu nhận dạng giao thức mức cao hơn kế tiếp, một trường của phần mào đầu internet.

Phần còn lại

Phần địa chỉ cục bộ của một địa chỉ internet.

Nguồn

Địa chỉ nguồn, một trường của phần mào đầu internet.

TCP

Giao thức điều khiển truyền tải: Một giao thức máy chủ - máy chủ cho sự truyền thông tin cậy trong các môi trường internet.

Đoạn TCP

Đơn vị dữ liệu được trao đổi giữa các mô-đun TCP (bao gồm cả phần mào đầu TCP).

TFTP

Giao thức chuyển tệp bình thường: Một giao thức chuyển tệp đơn giản dựa vào UDP.

Thời gian sống

Một trường của phần mào đầu internet cho biết giới hạn trên về thời gian mà gói tin internet này có thể tồn tại.

TOS

Loại dịch vụ.

Độ dài tổng

Trường Độ dài tổng của phần mào đầu internet cho biết độ dài của gói tin tính theo octet bao gồm cả dữ liệu và phần mào đầu internet.

TTL

Thời gian sống.

Loại dịch vụ

Một trường của phần mào đầu internet cho biết loại (hoặc chất lượng) của dịch vụ đối với gói tin internet này.

UDP

Giao thức gói tin người dùng: Một giao thức mức người dùng cho các ứng dụng hướng giao dịch.

Người dùng

Người dùng giao thức internet. Người dùng này có thể là một mô-đun giao thức mức cao hơn, một chương trình ứng dụng, hoặc một chương trình cổng.

Phiên bản

Trường Phiên bản cho biết khuôn dạng của phần mào đầu internet.

4. Yêu cầu kỹ thuật

4.1. Yêu cầu chung

4.1.1 Các giao diện

Giao thức internet được các giao thức máy chủ - máy chủ yêu cầu trong một môi trường internet. Giao thức này yêu cầu các giao thức mạng cục bộ truyền gói tin internet đến cổng kế tiếp hoặc đến máy chủ đích.

Ví dụ, một mô-đun TCP sẽ yêu cầu mô-đun internet lấy một đoạn TCP (bao gồm cả phần mào đầu TCP và dữ liệu người dùng) làm phần dữ liệu của một gói tin internet. Mô-đun TCP sẽ cung cấp các địa chỉ và các tham số khác trong phần mào đầu internet cho mô-đun internet thông qua các đối số của lệnh. Khi đó, mô-đun internet sẽ tạo ra một gói tin internet và yêu cầu giao diện mạng cục bộ truyền gói tin internet.

Ví dụ trong trường hợp ARPANET, mô-đun internet sẽ yêu cầu một mô-đun mạng cục bộ bổ sung bản ghi đầu nhóm 1822 vào gói tin internet nhằm tạo ra một bản tin của mạng ARPANET để truyền tới IMP. Địa chỉ của mạng ARPANET sẽ được suy ra từ địa chỉ internet theo giao diện mạng cục bộ và sẽ là địa chỉ của máy chủ nào đó trong mạng ARPANET, máy chủ đó có thể là một cổng đối với các mạng khác.

4.1.2 Hoạt động

Giao thức internet thực hiện hai chức năng cơ sở: lập địa chỉ và phân đoạn.

Các mô-đun internet sử dụng các địa chỉ được tải trong phần mào đầu internet để truyền các gói tin internet về các đích của chúng. Việc lựa chọn một đường truyền tải được gọi là định tuyến.

Mô-đun internet sử dụng các trường trong phần mào đầu internet để phân đoạn và tái lắp ráp các gói tin internet để truyền qua các mạng "gói nhỏ" khi cần thiết.

Mô hình hoạt động là một mô-đun internet lưu trú trong mỗi máy chủ tham gia vào quá trình truyền thông internet và trong mỗi cổng liên kết các mạng. Các mô-đun này dùng các qui tắc chung để diễn giải các trường địa chỉ và để phân đoạn và lắp ráp các gói tin internet. Ngoài ra, các mô-đun này (đặc biệt là các mô-đun internet lưu trú trong các cổng) có các thủ tục quyết định việc định tuyến và các chức năng khác.

Giao thức internet coi mỗi gói tin internet như một thực thể độc lập không liên quan với bất cứ gói tin internet nào khác. Không có các kết nối hoặc các kênh logic (ảo hoặc khác).

Giao thức internet sử dụng 4 cơ chế chủ yếu trong quá trình cung cấp dịch vụ của nó: Loại dịch vụ, thời gian sống, các tùy chọn, và kiểm tra tổng phần mào đầu.

Loại dịch vụ được sử dụng để biểu thị chất lượng dịch vụ mong muốn. Loại dịch vụ là một tập hợp trừu tượng hoặc tổng quát hóa của các tham số đặc trưng cho các lựa chọn dịch vụ được cung cấp trong các mạng cấu thành internet. Các cổng sử dụng chỉ dẫn về loại dịch vụ này để lựa chọn các tham số truyền thực cho một mạng cụ thể (mạng được sử dụng cho chặng kế tiếp, hoặc cổng kế tiếp khi định tuyến một gói tin internet).

Thời gian sống là một chỉ dẫn về một giới hạn trên của thời gian tồn tại của một gói tin internet. Thời gian sống được thiết lập bởi bên gửi gói tin và bị giảm dọc theo tuyến tại các điểm nó bị xử lý. Nếu thời gian sống bằng 0 trước khi gói tin internet đến đích, thì gói tin internet bị loại bỏ. Thời gian sống có thể được coi như một giới hạn của thời gian tự loại bỏ.

Các tùy chọn cung cấp các chức năng điều khiển cần thiết hoặc hữu ích trong một số tình huống nhưng lại không cần thiết trong phần lớn những quá trình truyền thông thông thường. Các tùy chọn bao gồm những dàn xếp về nhãn thời gian, bảo mật và định tuyến đặc biệt.

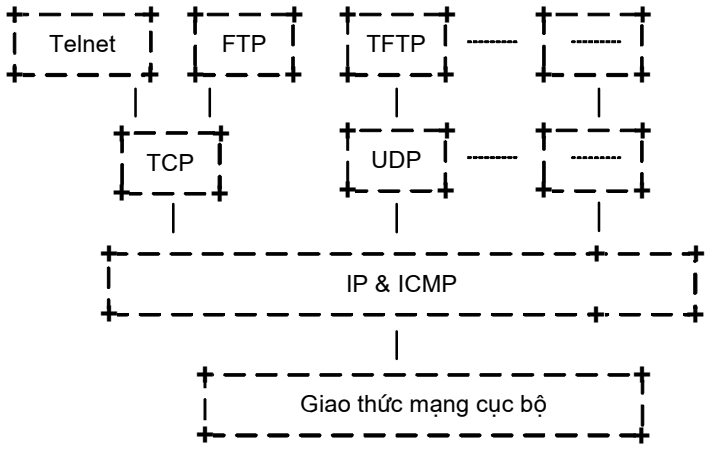
Kiểm tra tổng phần mào đầu cho phép kiểm tra thông tin sử dụng trong gói tin internet đang xử lý có được truyền đúng hay không. Dữ liệu này có thể chứa các lỗi. Nếu kiểm tra tổng phần mào đầu sai, gói tin internet lập tức bị loại bỏ bởi thực thể đã phát hiện ra lỗi.

Giao thức internet không cung cấp một phương tiện truyền thông tin cậy: không có các báo nhận đầu cuối - đầu cuối hay báo nhận theo chặng; không có kiểm soát lỗi cho dữ liệu, mà chỉ có kiểm tra tổng phần mào đầu; không có phát lại; không có điều khiển luồng.

Các lỗi được phát hiện có thể được thông báo qua Giao thức bản tin điều khiển Internet (ICMP), giao thức này được thực thi trong mô-đun giao thức internet.

4.1.3 Mối tương quan với các giao thức khác

Sơ đồ sau đây minh họa vị trí của giao thức Internet trong phân cấp của giao thức:



Hình 1: Mối tương quan của giao thức

Giao thức Internet một phía có giao diện với các giao thức máy chủ - máy chủ ở mức cao hơn và phía kia có giao diện với giao thức mạng cục bộ. Trong ngữ cảnh này, một "mạng cục bộ" có thể là một mạng nhỏ trong một tòa nhà hoặc một mạng lớn như mạng ARPANET.

4.1.4 Mô hình hoạt động

Mô hình hoạt động để truyền một gói tin từ một chương trình ứng dụng đến một chương trình ứng dụng khác được minh họa theo kịch bản sau đây:

Chúng ta giả định rằng việc truyền gói tin này sẽ phải qua một cổng trung gian.

Chương trình ứng dụng ở bên gửi chuẩn bị dữ liệu của nó và yêu cầu mô-đun internet cục bộ của nó gửi dữ liệu đó như một gói tin và truyền địa chỉ đích và các tham số khác thông qua các đối số của lệnh.

Mô-đun internet chuẩn bị một phần mào đầu của gói tin và gắn dữ liệu với nó. Mô-đun internet xác định một địa chỉ mạng cục bộ cho địa chỉ internet này, trong trường hợp này là địa chỉ của một cổng.

Mô-đun internet gửi gói tin này và địa chỉ mạng cục bộ đến giao diện mạng cục bộ.

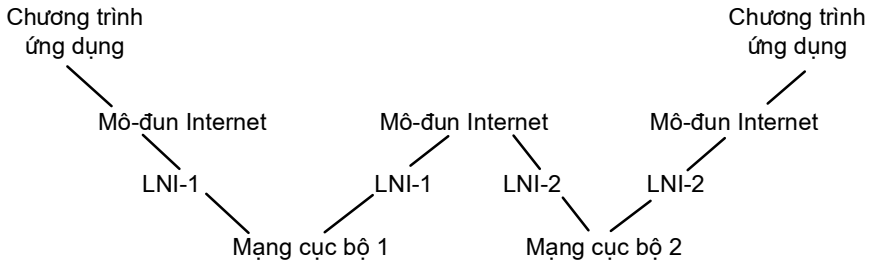
Giao diện mạng cục bộ tạo ra một phần mào đầu của mạng cục bộ và gắn gói tin với nó, sau đó gửi kết quả qua mạng cục bộ.

Gói tin tới một máy chủ cổng được bao bọc trong phần mào đầu của mạng cục bộ, giao diện của mạng cục bộ tước bỏ phần mào đầu này và chuyển giao gói tin cho mô-đun internet. Dựa vào địa chỉ internet, mô-đun internet xác định gói tin cần được chuyển tiếp đến máy chủ khác trong một mạng thứ hai. Mô-đun internet xác định một địa chỉ mạng cục bộ cho máy chủ đích. Nó yêu cầu giao diện mạng cục bộ với mạng thứ hai này để gửi gói tin đi.

Giao diện mạng cục bộ này tạo ra một phần mào đầu của mạng cục bộ và gắn gói tin vào rồi gửi kết quả đến máy chủ đích.

Tại máy chủ đích này, gói tin bị giao diện mạng cục bộ tước bỏ phần mào đầu mạng cục bộ và chuyển giao cho mô-đun internet.

Mô-đun internet xác định rằng gói tin dành cho một chương trình ứng dụng trong máy chủ này. Nó chuyển dữ liệu đến chương trình ứng dụng để đáp ứng một lệnh hệ thống, chuyển địa chỉ nguồn và các tham số khác như là các kết quả của lệnh này.



Hình 2: Đường truyền tải

4.1.5 Mô tả chức năng

Chức năng hay mục đích của Giao thức Internet là di chuyển các gói tin qua một tập hợp các mạng liên kết với nhau. Việc này được thực hiện bằng cách chuyển gói tin từ mô-đun internet này đến mô-đun internet khác cho đến khi tới đích. Các mô-đun internet lưu trú trong các máy chủ và các cổng trong hệ thống internet. Các gói tin được định tuyến từ một mô-đun internet đến mô-đun internet khác qua các mạng riêng biệt dựa vào sự diễn giải một địa chỉ internet. Do đó, một bộ phận quan trọng của giao thức internet là địa chỉ internet.

Trong việc định tuyến các bản tin từ mô-đun internet này đến mô-đun internet khác, các gói tin có thể cần đi ngang qua một mạng mà kích thước gói tối đa của

mạng này nhỏ hơn kích thước của gói tin. Để khắc phục khó khăn này, một cơ chế phân đoạn được cung cấp trong giao thức internet.

4.1.5.1 Lập địa chỉ

Có điểm khác biệt giữa tên, địa chỉ và tuyến. Tên cho biết đối tượng tìm kiếm. Địa chỉ cho biết vị trí. Tuyến cho biết làm thế nào để đến đó. Giao thức internet chủ yếu làm việc với các địa chỉ. Các giao thức ở mức cao hơn (tức là máy chủ-máy chủ hoặc ứng dụng) có nhiệm vụ thực hiện phép ánh xạ từ tên sang địa chỉ. Mô-đun internet ánh xạ các địa chỉ internet lên các địa chỉ mạng cục bộ. Các thủ tục ở mức thấp hơn (tức là mạng cục bộ hoặc các cổng) có nhiệm vụ thực hiện phép ánh xạ từ các địa chỉ mạng cục bộ lên các tuyến.

Các địa chỉ có độ dài cố định là 4 octet (32 bit). Một địa chỉ bắt đầu bằng phần mạng, tiếp theo là phần địa chỉ cục bộ (được gọi là "phần còn lại"). Có 3 khuôn dạng hay phân lớp địa chỉ internet: ở phân lớp địa chỉ A, bit bậc cao là 0, 7 bit kế tiếp chỉ thị phần mạng và 24 bit cuối cùng chỉ thị phần địa chỉ cục bộ; ở phân lớp địa chỉ B, 2 bit bậc cao là 1-0, 14 bit kế tiếp chỉ thị phần mạng và 16 bit cuối cùng chỉ thị phần địa chỉ cục bộ; ở phân lớp địa chỉ C, 3 bit bậc cao là 1-1-0, 21 bit kế tiếp chỉ thị phần mạng và 8 bit cuối cùng là chỉ thị phần địa chỉ cục bộ.

Phải thận trọng trong phép ánh xạ các địa chỉ internet vào các địa chỉ mạng cục bộ; một máy chủ vật lý đơn phải có khả năng phục vụ như thể nó là vài máy chủ khác biệt xét về khía cạnh sử dụng một vài địa chỉ internet khác biệt. Một số máy chủ cũng sẽ có vài giao diện vật lý (multi-homing).

Như vậy phải dự phòng cho một máy chủ có vài giao diện vật lý với mạng, mỗi giao diện vật lý có vài địa chỉ internet logic.

4.1.5.2 Phân đoạn

Việc phân đoạn một gói tin internet là cần thiết khi gói tin internet khởi phát trong một mạng cục bộ cho phép một kích cỡ gói lớn và phải đi ngang qua một mạng cục bộ hạn chế các gói vào ở một kích cỡ nhỏ hơn để đến đích của nó.

Một gói tin internet có thể được đánh dấu "không phân đoạn". Bất cứ gói tin internet nào được đánh dấu như vậy sẽ không bị phân đoạn internet trong bất cứ hoàn cảnh nào. Nếu gói tin internet được đánh dấu "không phân đoạn" không thể phân phát đến đích của nó được nếu không phân đoạn thì gói tin này sẽ bị loại bỏ.

Sự phân đoạn, truyền tải và tái lắp ráp qua một mạng cục bộ mà không thể nhìn thấy được đối với mô-đun của giao thức internet được gọi là phân đoạn intranet và có thể được sử dụng.

Thủ tục phân đoạn và tái lắp ráp internet phải có khả năng cắt một gói tin thành một số mảnh gần như tùy ý, các mảnh này có thể được tái lắp ráp sau này. Bên nhận

các đoạn sử dụng trường nhận dạng để đảm bảo rằng các đoạn của các gói tin khác nhau không bị trộn lẫn. Trường độ dịch đoạn cho bên nhận biết vị trí của một đoạn trong gói tin gốc. Độ dài đoạn và độ dịch đoạn xác định phần mà đoạn này chiếm trong gói tin gốc. Cờ chỉ báo còn đoạn cho biết (bằng cách thiết lập lại) đoạn cuối cùng. Các trường này cung cấp đủ thông tin để tái lắp ráp các gói tin.

Trường nhận dạng được sử dụng để phân biệt các đoạn của một gói tin với các đoạn của gói tin khác. Mô-đun giao thức khởi phát của một gói tin internet thiết lập trường nhận dạng ở một giá trị duy nhất đối với giao thức và cặp nguồn-đích đó trong thời gian gói tin tồn tại trong hệ thống internet. Mô-đun giao thức khởi phát của một gói tin đầy đủ thiết lập Cờ chỉ báo còn đoạn bằng 0 và độ dịch đoạn bằng 0.

Để phân đoạn một gói tin internet dài, một mô-đun giao thức internet (ví dụ, trong một cổng), tạo ra hai gói tin internet mới và sao chép nội dung của các trường phần mào đầu internet từ gói tin dài vào cả hai phần mào đầu internet mới. Dữ liệu của gói tin dài được chia thành hai phần trên một biên 8 octet (64 bit) (phần chia thứ hai có thể không phải là một bội số nguyên của 8 octet, nhưng phần chia đầu tiên thì bắt buộc). Gọi số khối 8 octet trong phần chia đầu tiên là NFB (Số khối của đoạn). Phần chia đầu tiên của dữ liệu được đặt vào gói tin internet mới đầu tiên, và trường độ dài tổng được thiết lập bằng độ dài của gói tin đầu tiên. Cờ chỉ báo còn đoạn được thiết lập bằng 1. Phần chia thứ hai của dữ liệu được đặt vào gói tin internet mới thứ hai và trường độ dài tổng được thiết lập bằng độ dài của gói tin thứ hai. Cờ chỉ báo còn đoạn mang cùng một giá trị như gói tin dài. Trường dịch đoạn của gói tin internet mới thứ hai được thiết lập bằng giá trị của trường đó trong gói tin dài cộng với NFB.

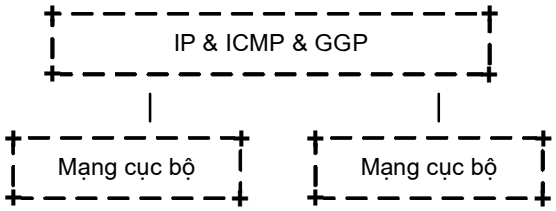
Thủ tục này có thể được suy rộng cho việc phân tách thành n phần, thay cho việc phân tách thành 2 phần như đã mô tả.

Để ghép các đoạn của một gói tin internet, một mô-đun giao thức internet (ví dụ ở một máy chủ đích) tổ hợp tất cả các gói tin internet có cùng một giá trị đối với 4 trường: nhận dạng, nguồn, đích và giao thức. Việc tổ hợp được thực hiện bằng cách đặt phần chia dữ liệu của mỗi đoạn vào vị trí tương đối được xác định bởi độ dịch đoạn trong phần mào đầu internet của đoạn đó. Đoạn đầu tiên sẽ có độ dịch đoạn bằng 0, và đoạn cuối cùng sẽ có Cờ chỉ báo còn đoạn được thiết lập bằng 0.

4.1.6 Các cổng

Các cổng thực thi giao thức internet để chuyển tiếp các gói tin giữa các mạng. Các cổng cũng thực thi Giao thức Cổng - Cổng để phối hợp việc định tuyến và thông tin điều khiển internet khác.

Trong một cổng, các giao thức mức cao hơn không cần được thực thi và các chức năng của GGP được bổ sung cho mô-đun IP.

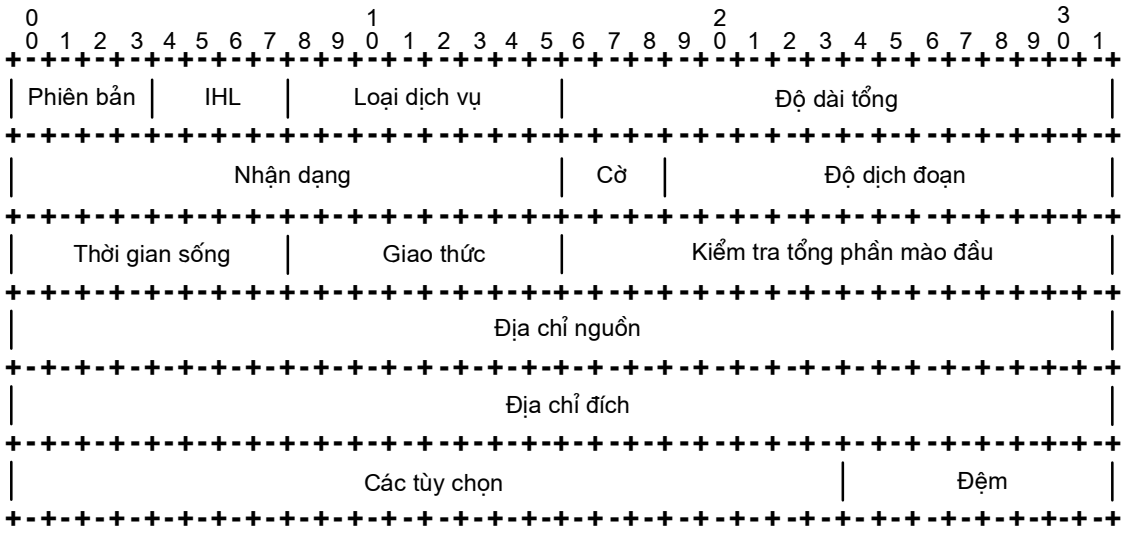


Hình 3: Các giao thức cổng

4.2. Yêu cầu kỹ thuật

4.2.1 Khuôn dạng của phần mào đầu internet

Tóm tắt về các nội dung của phần mào đầu internet như sau:



Hình 4: Ví dụ về phần mào đầu của gói tin internet

Chú ý rằng mỗi dấu phân thời biểu diễn một vị trí bit.

Phiên bản: 4 bit

Trường Phiên bản cho biết khuôn dạng của phần mào đầu internet. Tài liệu này mô tả phiên bản 4.

IHL: 4 bit

Trường độ dài phần mào đầu internet cho biết độ dài của phần mào đầu internet tính theo đơn vị là các từ 32 bit và do đó chỉ ra vị trí bắt đầu của phần dữ liệu. Chú ý rằng giá trị IHL tối thiểu cho một phần mào đầu đúng là 5.

Loại dịch vụ: 8 bit

Loại dịch vụ cho biết các tham số trừu tượng về chất lượng dịch vụ mong muốn. Các tham số này thường được sử dụng để hướng dẫn việc lựa chọn các tham

số dịch vụ thực tế khi truyền một gói tin qua một mạng cụ thể. Một vài mạng cung cấp thứ tự ưu tiên của dịch vụ, bằng cách nào đó sẽ xử lý lưu lượng có thứ tự ưu tiên cao là quan trọng hơn so với các lưu lượng khác (thông thường bằng cách chỉ chấp nhận lưu lượng có thứ tự ưu tiên trên một mức nào đó tại thời điểm có tải cao). Sự lựa chọn chủ yếu là một sự cân bằng ba chiều giữa độ trễ thấp, độ tin cậy cao và thông lượng cao.

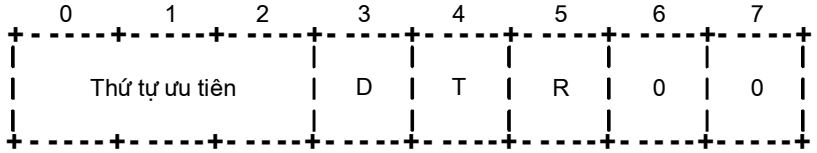
Các bit 0 - 2: Thứ tự ưu tiên

Bit 3: 0 = Độ trễ bình thường, 1 = Độ trễ thấp

Các bit 4: 0 = Thông lượng bình thường, 1 = Thông lượng cao

Các bit 5: 0 = Độ tin cậy bình thường, 1 = Độ tin cậy cao

Bit 6 - 7: Được dự trữ để dùng trong tương lai



Thứ tự ưu tiên

111 - Điều khiển mạng

110 - Điều khiển liên mạng

101 - CRITIC/ECP

100 - Ưu tiên hơn tin ngắn

011 - Tin ngắn

010 - Tức thời

001 - Ưu tiên

000 - Thường trình

Việc sử dụng các chỉ báo Độ trễ, Thông lượng và Độ tin cậy có thể làm tăng chi phí của dịch vụ (theo nghĩa nào đó). Trong nhiều mạng, chỉ tiêu tốt hơn đối với một trong các tham số này được kết hợp với chỉ tiêu kém hơn trên tham số khác. Ngoại trừ những trường hợp rất không bình thường, tối đa là hai trong ba chỉ báo này sẽ được thiết lập.

Loại dịch vụ được sử dụng để quy định cách xử lý gói tin trong thời gian truyền gói tin qua hệ thống internet.

Việc chỉ định thứ tự ưu tiên Điều khiển mạng dự kiến chỉ được sử dụng trong phạm vi một mạng. Việc sử dụng và kiểm soát việc chỉ định đó trên thực tế là tùy

theo từng mạng. Việc chỉ định Điều khiển liên mạng dự kiến chỉ được sử dụng cho những bộ khởi phát điều khiển cổng. Nếu việc sử dụng thực tế các chỉ định thứ tự ưu tiên này được xem xét bởi một mạng cụ thể, thì mạng đó có trách nhiệm kiểm soát việc truy nhập và sử dụng các chỉ định thứ tự ưu tiên đó.

Độ dài tổng: 16 bit

Độ dài tổng là độ dài của gói tin, được đo bằng octet, bao gồm phần mào đầu internet và dữ liệu. Trường này cho phép gói tin có độ dài tới 65535 octet. Các gói tin dài như vậy là không thực tế đối với hầu hết các máy chủ và các mạng. Tất cả các máy chủ phải sẵn sàng chấp nhận các gói tin có độ dài tới 576 octet (dù chúng đến toàn bộ hay theo các đoạn). Người ta khuyến nghị rằng các máy chủ chỉ gửi các gói tin lớn hơn 576 octet nếu chúng đảm bảo rằng đích sẵn sàng chấp nhận các gói tin lớn hơn.

Số 576 được chọn để cho phép một khối dữ liệu có kích cỡ hợp lý được truyền (ngoài thông tin về phần mào đầu đã được yêu cầu). Ví dụ, kích cỡ này cho phép một khối dữ liệu là 512 octet cộng với 64 octet của phần mào đầu để khớp với một gói tin. Phần mào đầu internet tối đa là 60 octet, và một phần mào đầu internet điển hình là 20 octet, cho phép một khoảng dự trữ cho các phần mào đầu của các giao thức mức cao hơn.

Nhận dạng: 16 bit

Một giá trị nhận dạng được gán bởi bên gửi để trợ giúp việc ghép các đoạn của một gói tin.

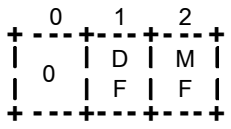
Cờ: 3 bit

Các cờ điều khiển khác nhau

Bit 0: được dự trữ, phải bằng 0

Bit 1: (DF) 0 = có thể phân đoạn, 1 = không phân đoạn

Bit 2: (MF) 0 = đoạn cuối cùng, 1 = còn đoạn nữa



Độ dịch đoạn: 13 bit

Trường này cho biết đoạn này ở chỗ nào trong gói tin.

Độ dịch đoạn được đo theo đơn vị 8 octet (64 bit). Đoạn đầu tiên có độ dịch đoạn bằng 0.

Thời gian sống: 8 bit

Trường này cho biết thời gian tối đa mà gói tin được phép ở lại trong hệ thống internet. Nếu trường này chứa giá trị 0, thì gói tin phải bị hủy. Trường này được sửa

đổi trong quá trình xử lý phần mào đầu internet. Thời gian được đo theo đơn vị giây, nhưng vì mỗi mô-đun xử lý một gói tin phải giảm TTL đi ít nhất 1 giây dù nó xử lý gói tin trong thời gian dưới 1 giây, TTL phải được coi như một giới hạn trên của thời gian một gói tin có thể tồn tại. Mục đích là khiến cho các gói tin không thể phân phát được phải bị loại bỏ và quy định giới hạn cho thời gian sống tối đa của gói tin.

Giao thức: 8 bit

Trường này cho biết giao thức mức kế tiếp nào được sử dụng trong phần dữ liệu của gói tin internet.

Kiểm tra tổng phần mào đầu: 16 bit

Chỉ kiểm tra tổng trên phần mào đầu. Vì một số trường phần mào đầu thay đổi (ví dụ thời gian sống), nên việc kiểm tra tổng phần mào đầu được tính toán lại và xác minh tại mỗi điểm mà phần mào đầu internet được xử lý.

Thuật toán kiểm tra tổng là:

Trường kiểm tra tổng là phần bù của trường 16 bit của tổng bù tất cả các từ 16 bit trong phần mào đầu. Cho mục đích tính toán tổng kiểm tra, giá trị của trường kiểm tra tổng phải bằng 0.

Đây là một thuật toán đơn giản để tính kiểm tra tổng và bằng chứng thực nghiệm cho thấy nó là thích hợp, nhưng nó là tạm thời và có thể được thay thế bằng một thủ tục CRC, phụ thuộc vào kinh nghiệm triển khai về sau.

Địa chỉ nguồn: 32 bit (xem mục 3.2)

Địa chỉ đích: 32 bit (xem mục 3.2)

Các tùy chọn: thay đổi

Các tùy chọn có thể xuất hiện hoặc không xuất hiện trong các gói tin. Chúng phải được mọi mô-đun IP (máy chủ và các cổng) thực thi. Việc truyền chúng trong bất cứ gói tin riêng biệt nào là tùy chọn (chứ không phải việc thực thi chúng).

Trong một số môi trường, tùy chọn bảo mật có thể được yêu cầu trong mọi gói tin.

Trường tùy chọn có độ dài thay đổi. Có thể không có tùy chọn nào hoặc có thể có nhiều tùy chọn. Có 2 trường hợp về khuôn dạng của một tùy chọn:

- Trường hợp 1: Một octet đơn kiểu-tùy chọn

- Trường hợp 2: Một octet kiểu-tùy chọn, một octet độ dài-tùy chọn, và các octet dữ liệu-tùy chọn hiện thời.

Octet độ dài - tùy chọn tính đến octet kiểu-tùy chọn và octet độ dài-tùy chọn cũng như các octet dữ liệu - tùy chọn.

Octet kiểu - tùy chọn được xem như có 3 trường:

1 bit cờ sao chép;

2 bit loại tùy chọn;

5 bit số tùy chọn.

Cờ sao chép cho biết tùy chọn này được sao chép vào tất cả các đoạn khi phân đoạn.

0 = không được sao chép

1 = được sao chép

Các loại tùy chọn là:

0 = điều khiển

1 = dự trữ để sử dụng trong tương lai

2 = gỡ rối và đo kiểm

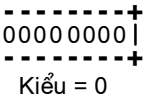
3 = dự trữ để sử dụng trong tương lai

Các tùy chọn internet sau đây được xác định:

LOẠI	SỐ	ĐỘ DÀI	MÔ TẢ
0	0	-	Kết thúc danh sách tùy chọn. Tùy chọn này chỉ chiếm 1 octet, nó không có octet độ dài.
0	1	-	Không hoạt động. Tùy chọn này chỉ chiếm 1 octet, nó không có octet độ dài.
0	2	11	Bảo mật. Được dùng để truyền Bảo mật, Phân chia ngăn, Nhóm người dùng (TCC), và các Mã hạn chế điều khiển tương thích với các yêu cầu của DOD.
0	3	thay đổi	Định tuyến nguồn không nghiêm ngặt. Được dùng để định tuyến gói tin internet dựa trên thông tin do nguồn cung cấp.
0	9	thay đổi	Định tuyến nguồn nghiêm ngặt. Được dùng để định tuyến gói tin internet dựa trên thông tin do nguồn cung cấp.
0	7	thay đổi	Tuyến ghi. Được dùng để dò lại tuyến mà một gói tin internet đã đi.
0	8	4	ID luồng. Được dùng để truyền ký hiệu nhận dạng luồng.
2	4	thay đổi	Nhãn thời gian internet.

Các định nghĩa về tùy chọn riêng

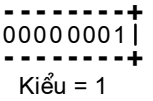
Kết thúc danh sách tùy chọn



Tùy chọn này cho biết kết thúc của danh sách tùy chọn. Kết thúc này có thể không trùng khớp với kết thúc của phần mào đầu internet căn cứ theo độ dài của phần mào đầu internet. Tùy chọn này được sử dụng ở phần cuối của tất cả các tùy chọn, chứ không phải ở phần cuối của mỗi tùy chọn và chỉ cần dùng nếu phần cuối của các tùy chọn không trùng khớp với phần cuối của phần mào đầu internet.

Có thể được sao chép, được sử dụng, hoặc bị xóa khi phân đoạn, hoặc vì bất cứ lý do nào khác.

Không hoạt động

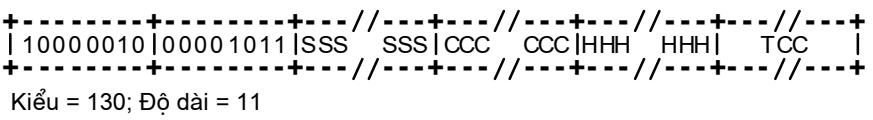


Tùy chọn này có thể được sử dụng giữa các tùy chọn, ví dụ để cân chỉnh phần đầu của một tùy chọn tiếp theo trên một biên 32 bit.

Có thể được sao chép, được sử dụng, hoặc bị xóa khi phân đoạn, hoặc vì bất cứ lý do nào khác.

Bảo mật

Tùy chọn này cung cấp một phương pháp cho các máy chủ gửi các tham số bảo mật, phân chia ngăn, hạn chế điều khiển và TCC (nhóm người dùng khép kín). Khuôn dạng của tùy chọn này như sau:



Bảo mật (trường S): 16 bit

Chỉ định một trong số 16 mức bảo mật (8 mức trong số 16 mức này được dự trữ để dùng trong tương lai).

00000000 00000000 - Chưa được xếp loại

11110001 00110101 - Mật

01111000 10011010 - EFTO

10111100 01001101 - MMMM

01011110 00100110 - PROG

10101111 00010011 - Cấm

11010111 10001000 - Bí mật

01101011 11000101 - Tối mật

00110101 11100010 - (Được dự trữ để dùng trong tương lai)

10011010 11110001 - (Được dự trữ để dùng trong tương lai)

01001101 01111000 - (Được dự trữ để dùng trong tương lai)

00100100 10111101 - (Được dự trữ để dùng trong tương lai)

00010011 01011110 - (Được dự trữ để dùng trong tương lai)

10001001 10101111 - (Được dự trữ để dùng trong tương lai)

11000100 11010110 - (Được dự trữ để dùng trong tương lai)

11100010 01101011 - (Được dự trữ để dùng trong tương lai)

Phân chia ngăn (trường C): 16 bit

Một giá trị toàn là 0 được sử dụng khi thông tin đã truyền không được phân chia ngăn. Các giá trị khác dành cho trường chia ngăn có thể được nhận từ Cơ quan Tình báo Quốc phòng.

Hạn chế điều khiển (trường H): 16 bit

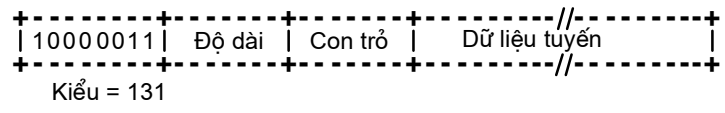
Các giá trị để đánh dấu kiểm soát và phát hành là các chữ ghép chữ-số và được xác định trong Sách hướng dẫn của Cơ quan Tình báo Quốc phòng DIAM 65-19, "Đánh dấu bảo mật chuẩn"

Mã điều khiển truyền (trường TCC): 24 bit

Cung cấp một phương tiện để chia tách lưu lượng và xác định các cộng đồng có quyền lợi được điều chỉnh trong số các thuê bao. Các giá trị TCC là các nhóm ba chữ cái kế tiếp nhau, và khả dụng từ Mã HQ DCA 530.

Phải được sao chép khi phân đoạn. Tùy chọn này xuất hiện tối đa một lần trong một gói tin.

Tuyến ghi và tuyến nguồn không nghiêm ngặt



Tùy chọn tuyến ghi và tuyến nguồn không nghiêm ngặt (LSRR) đưa ra một phương pháp để nguồn của một gói tin internet cung cấp thông tin định tuyến được sử dụng bởi các cổng để chuyển tiếp gói tin đến đích và để ghi lại thông tin về tuyến.

Tùy chọn này bắt đầu với mã kiểu tùy chọn. Octet thứ hai là độ dài tùy chọn bao gồm mã kiểu tùy chọn và octet độ dài, octet con trỏ và (độ dài –3) octet dữ liệu

tuyến. Octet thứ ba là con trỏ trỏ vào dữ liệu tuyến chỉ báo octet bắt đầu địa chỉ nguồn kế tiếp sẽ được xử lý. Con trỏ là tương đối đối với tùy chọn này và giá trị hợp lệ nhỏ nhất cho con trỏ là 4.

Một dữ liệu tuyến bao gồm một dãy các địa chỉ internet. Mỗi địa chỉ internet là 32 bit hoặc 4 octet. Nếu con trỏ lớn hơn độ dài, thì tuyến nguồn là trống (và tuyến ghi đầy) và việc định tuyến cần được dựa trên trường địa chỉ đích.

Nếu đã đến được địa chỉ trong trường địa chỉ đích và con trỏ không lớn hơn độ dài, thì địa chỉ kế tiếp trong tuyến nguồn thay thế địa chỉ trong trường địa chỉ đích, và địa chỉ tuyến ghi thay thế địa chỉ nguồn vừa mới dùng và con trỏ được tăng lên 4.

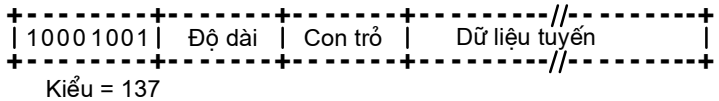
Địa chỉ tuyến ghi là địa chỉ internet của chính mô-đun internet như đã biết trong môi trường mà gói tin này đang được yêu cầu chuyển tiếp.

Thủ tục thay thế tuyến nguồn bằng tuyến ghi này (mặc dù nó ngược với trình tự mà nó phải theo để được sử dụng như một tuyến nguồn) có nghĩa là tùy chọn (và toàn bộ phần mào đầu IP) vẫn giữ nguyên một độ dài không đổi khi gói tin qua internet.

Tùy chọn này là một tuyến nguồn không nghiêm ngặt vì IP của cổng hoặc máy chủ được phép dùng bất cứ tuyến nào trong số các tuyến có các cổng trung gian khác để đến địa chỉ kế tiếp trong tuyến.

Phải được sao chép khi phân đoạn. Xuất hiện tối đa một lần trong một gói tin.

Tuyến ghi và nguồn nghiêm ngặt



Tùy chọn tuyến ghi và tuyến nguồn nghiêm ngặt (SSRR) đưa ra một phương pháp để nguồn của một gói tin internet cung cấp thông tin định tuyến được sử dụng bởi các cổng để chuyển tiếp gói tin đến đích và để ghi lại thông tin về tuyến.

Tùy chọn này bắt đầu với mã kiểu tùy chọn. Octet thứ hai là độ dài tùy chọn, bao gồm mã kiểu tùy chọn và octet độ dài, octet con trỏ và (độ dài -3) octet dữ liệu tuyến. Octet thứ ba là con trỏ trỏ vào dữ liệu tuyến chỉ báo octet bắt đầu địa chỉ nguồn kế tiếp sẽ được xử lý. Con trỏ là tương đối đối với tùy chọn này và giá trị hợp lệ nhỏ nhất cho con trỏ là 4.

Một dữ liệu tuyến bao gồm một dãy các địa chỉ internet. Mỗi địa chỉ internet là 32 bit hay 4 octet. Nếu con trỏ lớn hơn độ dài thì tuyến nguồn là trống (và tuyến ghi đầy) và việc định tuyến cần được dựa trên trường địa chỉ đích.

Nếu đã đến được địa chỉ trong trường địa chỉ đích và con trỏ không lớn hơn độ dài thì địa chỉ kế tiếp trong tuyến nguồn thay thế địa chỉ trong trường địa chỉ đích và địa chỉ tuyến ghi thay thế địa chỉ nguồn vừa mới dùng và con trỏ được tăng lên 4.

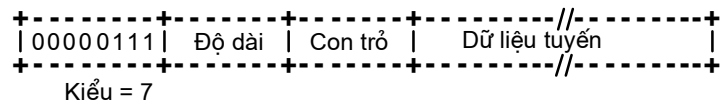
Địa chỉ tuyến ghi là địa chỉ internet của chính mô-đun internet như đã biết trong môi trường mà gói tin này đang được yêu cầu chuyển tiếp.

Thủ tục thay thế tuyến nguồn bằng tuyến ghi này (mặc dù nó ngược với trình tự mà nó phải theo để được sử dụng như một tuyến nguồn) có nghĩa là tùy chọn (và toàn bộ phần mào đầu IP) vẫn giữ nguyên một độ dài không đổi khi gói tin qua internet.

Tùy chọn này là một tuyến nguồn nghiêm ngặt vì IP của cổng hoặc máy chủ phải gửi gói tin trực tiếp đến địa chỉ kế tiếp trong tuyến nguồn chỉ qua mạng kết nối trực tiếp (được chỉ báo trong địa chỉ kế tiếp) để đến cổng hoặc máy chủ kế tiếp (được chỉ định trong tuyến).

Phải được sao chép khi phân đoạn. Xuất hiện tối đa một lần trong một gói tin.

Tuyến ghi



Tùy chọn tuyến ghi đưa ra một phương pháp để ghi lại tuyến của một gói tin internet.

Tùy chọn bắt đầu với mã kiểu tùy chọn. Octet thứ hai là độ dài tùy chọn, bao gồm mã kiểu tùy chọn và octet độ dài, octet con trỏ và (độ dài –3) octet dữ liệu tuyến. Octet thứ ba là con trỏ trỏ vào dữ liệu tuyến chỉ báo octet bắt đầu vùng kế tiếp để lưu giữ một địa chỉ tuyến. Con trỏ là tương đối đối với tùy chọn này và giá trị hợp lệ nhỏ nhất cho con trỏ là 4.

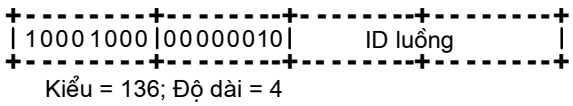
Một tuyến ghi bao gồm một dãy các địa chỉ internet. Mỗi địa chỉ internet có độ dài 32 bit hay 4 octet. Nếu con trỏ lớn hơn độ dài thì vùng dữ liệu của tuyến ghi là đầy. Máy chủ khởi phát phải thiết lập tùy chọn này với một vùng dữ liệu tuyến đủ lớn để chứa toàn bộ địa chỉ đã yêu cầu. Kích cỡ của tùy chọn không thay đổi khi điền thêm các địa chỉ vào. Các nội dung ban đầu của vùng dữ liệu tuyến phải bằng 0.

Khi một mô-đun internet định tuyến một gói tin, nó sẽ kiểm tra xem có tùy chọn tuyến ghi không. Nếu có tùy chọn tuyến ghi, mô-đun này chèn địa chỉ internet của chính nó, như đã biết trong môi trường mà gói tin này đang được yêu cầu chuyển tiếp vào tuyến ghi bắt đầu tại octet do con trỏ chỉ báo và tăng con trỏ lên 4.

Nếu vùng dữ liệu tuyến đã đầy (con trỏ vượt quá độ dài) thì gói tin được chuyển tiếp mà không cần chèn địa chỉ vào tuyến ghi. Nếu còn chỗ nào đó nhưng không đủ để chèn một địa chỉ đầy đủ, thì gói tin gốc được coi như bị lỗi và bị loại bỏ. Trong cả hai trường hợp, một bản tin về vấn đề tham số ICMP có thể được gửi đến máy chủ nguồn.

Không được sao chép khi phân đoạn, tùy chọn này chỉ xuất hiện trong đoạn đầu tiên và xuất hiện tối đa một lần trong một gói tin.

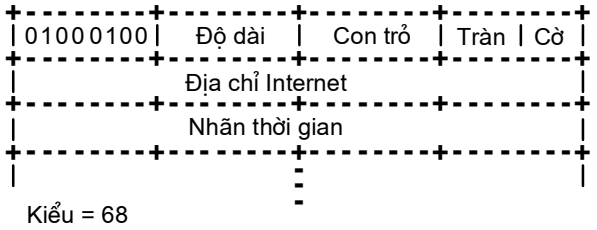
Từ nhận dạng luồng



Tùy chọn này đưa ra một phương pháp để từ nhận dạng luồng 16 bit SATNET được truyền qua các mạng không hỗ trợ khái niệm luồng.

Phải được sao chép khi phân đoạn. Xuất hiện tối đa một lần trong một gói tin.

Nhãn thời gian của internet



Độ dài tùy chọn là số các octet trong tùy chọn kể cả octet kiểu, octet độ dài, octet con trỏ, và các octet tràn/cờ (độ dài tối đa là 40).

Con trỏ là số các octet từ phần đầu của tùy chọn này đến phần cuối của các nhãn thời gian cộng với 1 (tức là nó trỏ vào octet bắt đầu khoảng trống cho nhãn thời gian kế tiếp). Giá trị hợp lệ nhỏ nhất là 5. Vùng nhãn thời gian đầy khi con trỏ lớn hơn độ dài.

Phần tràn (oflw) [4 bit] là số các mô-đun IP không thể đăng ký các nhãn thời gian do thiếu khoảng trống.

Các giá trị của cờ (flg) [4 bit] là:

0: Chỉ các nhãn thời gian, được lưu giữ trong các từ 32 bit liên tiếp,

1: Mỗi nhãn thời gian đứng sau địa chỉ internet của thực thể đăng ký,

3: Các trường địa chỉ internet được chỉ định trước. Một mô-đun IP chỉ đăng ký nhãn thời gian của nó nếu nó so khớp địa chỉ của chính nó với địa chỉ internet được chỉ định kế tiếp.

Nhãn thời gian là một nhãn thời gian 32 bit, căn phải tính theo miligiây từ nửa đêm theo giờ quốc tế. Nếu thời gian không theo dạng miligiây hoặc không thể được cung cấp từ nửa đêm theo giờ quốc tế thì bất cứ thời gian nào cũng có thể bị chèn vào như một nhãn thời gian miễn là bit bậc cao của trường nhãn thời gian được thiết lập bằng 1 để chỉ báo việc dùng một giá trị không chuẩn.

Máy chủ khởi phát phải thiết lập tùy chọn này với một vùng dữ liệu nhãn thời gian đủ lớn để chứa toàn bộ thông tin yêu cầu về nhãn thời gian. Kích cỡ của tùy

chọn không thay đổi khi điền thêm các nhãn thời gian vào. Những nội dung ban đầu của vùng dữ liệu của nhãn thời gian phải bằng 0 hoặc bằng các cặp địa chỉ internet / 0.

Nếu vùng dữ liệu nhãn thời gian đã đầy (con trỏ vượt quá độ dài) thì gói tin được chuyển tiếp mà không cần chèn nhãn thời gian, nhưng số đếm tràn được tăng lên 1.

Nếu còn chỗ nào đó nhưng không đủ để chèn một nhãn thời gian đầy đủ, hoặc bản thân số đếm tràn cũng tràn, thì gói tin gốc được coi như bị lỗi và bị loại bỏ. Trong cả hai trường hợp, một bản tin về vấn đề tham số của ICMP có thể được gửi đến máy chủ nguồn.

Tùy chọn nhãn thời gian không được sao chép trong khi phân đoạn. Nó được truyền trong đoạn đầu tiên và xuất hiện tối đa một lần trong một gói tin.

Đệm: thay đổi

Đệm của phần mào đầu internet được sử dụng để đảm bảo rằng các phần cuối của phần mào đầu internet ở trên một biên 32 bit. Đệm bằng 0.

4.2.2 Mô tả

Việc thực thi một giao thức phải thiết thực. Mỗi thực thi phải hoạt động liên kết được với những thực thi khác tạo bởi nhiều thực thể khác nhau. Trong khi mục đích của phần mô tả này là nhằm diễn đạt rõ ràng, đầy đủ về giao thức, thì vẫn có khả năng có những sự diễn giải khác. Về tổng quan, một thực thi phải thận trọng trong chế độ gửi và phải rộng rãi trong chế độ nhận của nó. Tức là, phải cẩn thận để gửi các gói tin có khuôn dạng chuẩn nhưng phải chấp nhận bất cứ gói tin nào mà có thể diễn giải được (ví dụ: không phản đối các lỗi kỹ thuật khi nghĩa vẫn còn rõ ràng).

Dịch vụ internet cơ sở là dịch vụ định hướng gói tin và cung cấp sự phân đoạn các gói tin tại các cổng, cùng với việc tái lắp ráp xảy ra tại mô-đun giao thức internet đích trong máy chủ đích. Đương nhiên, sự phân đoạn và tái lắp ráp các gói tin bên trong một mạng hoặc theo sự thỏa thuận riêng giữa các cổng của một mạng cũng được cho phép vì sự phân đoạn và tái lắp ráp này là trong suốt đối với các giao thức internet và các giao thức mức cao hơn. Kiểu phân đoạn và tái lắp ráp trong suốt này được gọi là sự phân đoạn "phụ thuộc vào mạng" (hoặc intranet) và không được đề cập thêm nữa.

Các địa chỉ internet phân biệt các nguồn và các đích với mức của máy chủ và cũng cung cấp một trường giao thức. Giả thiết là mỗi giao thức sẽ cung cấp mọi sự ghép kênh cần thiết bên trong một máy chủ.

4.2.2.1 Lập địa chỉ

Để linh hoạt trong việc gán địa chỉ cho các mạng và tính đến nhiều mạng có kích cỡ từ nhỏ đến trung bình, sự thể hiện của trường địa chỉ được mã hóa để xác định một số ít mạng với nhiều máy chủ một số lượng vừa phải mạng với một số lượng vừa phải máy chủ và một số lượng lớn mạng với ít máy chủ. Ngoài ra, có một mã thoát đến chế độ lập địa chỉ mở rộng.

Các khuôn dạng của địa chỉ

Các bit bậc cao	Khuôn dạng	Phân lớp
0	7 bit của mạng, 24 bit của máy chủ	A
10	14 bit của mạng, 16 bit của máy chủ	B
110	21 bit của mạng, 8 bit của máy chủ	C
111	Thoát đến chế độ lập địa chỉ mở rộng	

Một giá trị 0 trong trường mạng dành cho mạng này. Giá trị này chỉ được sử dụng trong các bản tin ICMP nhất định. Chế độ lập địa chỉ mở rộng không được xác định. Cả hai đặc trưng này được dự trữ để sử dụng trong tương lai.

Địa chỉ cục bộ, do mạng cục bộ gán, phải tính đến việc một máy chủ vật lý đơn thực hiện vai trò như vài máy chủ internet riêng biệt. Tức là, phải có một phép ánh xạ giữa các địa chỉ của máy chủ internet và các giao diện mạng/máy chủ mà cho phép vài địa chỉ internet tương ứng với một giao diện. Cũng phải tính đến một máy chủ có nhiều giao diện vật lý và coi các gói tin từ một vài giao diện trong số đó như thể chúng đều được lập địa chỉ đến cùng một máy chủ đơn.

4.2.2.2 Phân đoạn và tái lắp ráp

Sử dụng trường nhận dạng internet (ID) cùng với địa chỉ nguồn, địa chỉ đích và các trường giao thức để nhận dạng các đoạn của gói tin cho việc tái lắp ráp.

Bit MF (Cờ chỉ báo còn đoạn) được thiết lập nếu gói tin này không phải là đoạn cuối cùng. Trường Độ dịch đoạn cho biết vị trí đoạn so với phần đầu của gói tin gốc chưa phân đoạn. Các đoạn được tính theo đơn vị 8 octet. Chiến lược phân đoạn được lập ra để một gói tin không bị phân đoạn có mọi thông tin phân đoạn bằng 0 (MF = 0, độ dịch đoạn = 0). Nếu một gói tin internet bị phân đoạn, phần dữ liệu của nó phải bị ngắt trên các biên 8 octet.

Khuôn dạng này cho phép 2^{13} = 8192 đoạn, mỗi đoạn có 8 octet, do vậy khuôn dạng này cho phép tổng số 65536 octet. Chú ý rằng, điều này là phù hợp với trường độ dài tổng của gói tin (dĩ nhiên, phần mào đầu được tính theo độ dài tổng và không tính theo các đoạn).

Khi xảy ra phân đoạn, một số tùy chọn được sao chép, nhưng các tùy chọn khác vẫn giữ nguyên chỉ ở đoạn đầu tiên.

Mỗi mô-đun internet phải có khả năng chuyển tiếp một gói tin 68 octet mà không cần phân đoạn. Vì một phần mào đầu internet có thể tối đa là 60 octet và đoạn tối thiểu là 8 octet.

Mỗi đích internet phải có khả năng nhận một gói tin 576 octet hoặc dưới dạng một gói tin nguyên vẹn hoặc dưới dạng các đoạn cần được tái lắp ráp.

Việc phân đoạn có thể làm ảnh hưởng đến các trường sau:

(1) Trường các tùy chọn

(2) Cờ chỉ báo còn đoạn

(3) Độ dịch đoạn

(4) Trường độ dài phần mào đầu internet

(5) Trường độ dài tổng

(6) Kiểm tra tổng phần mào đầu

Nếu bit cờ không phân đoạn (DF) được thiết lập thì sự phân đoạn internet của gói tin này là KHÔNG được phép, mặc dù nó có thể bị loại bỏ. Điều này có thể được sử dụng để ngăn chặn sự phân đoạn trong các trường hợp máy chủ nhận không có đủ tài nguyên để tái lắp ráp các đoạn internet.

Một ví dụ về việc dùng đặc trưng không phân đoạn là tải tuyến xuống một máy chủ nhỏ. Một máy chủ nhỏ có thể có một chương trình nạp, chương trình này chấp nhận một gói tin, lưu trữ nó trong bộ nhớ và sau đó thực hiện nó.

Các thủ tục phân đoạn và tái lắp ráp đa số được mô tả một cách dễ dàng bằng các ví dụ. Các thủ tục sau đây là những ví dụ về thực thi.

Ký hiệu chung trong các chương trình giả như sau: "=<" nghĩa là "nhỏ hơn hoặc bằng", "#" nghĩa là "không bằng", "=" nghĩa là "bằng", "<-" nghĩa là "được thiết lập bằng". Cũng vậy, "x đến y" bao gồm x đến y và loại trừ y; ví dụ, "4 đến 7" sẽ bao gồm 4, 5, và 6 (nhưng không bao gồm 7).

4.2.2.2.1 Ví dụ về thủ tục phân đoạn

Gói tin có kích cỡ tối đa mà có thể được truyền qua mạng kế tiếp được gọi là đơn vị truyền tối đa (MTU).

Nếu độ dài tổng nhỏ hơn hoặc bằng đơn vị truyền tối đa thì đưa gói tin này tới bước kế tiếp trong quá trình xử lý gói tin, nếu không thì cắt gói tin thành hai đoạn, đoạn đầu tiên có kích cỡ tối đa và đoạn thứ hai là phần còn lại của gói tin. Đoạn đầu tiên được đưa đến bước kế tiếp trong quá trình xử lý gói tin, trong khi đoạn thứ hai được đưa đến thủ tục này trong trường hợp nó vẫn còn quá lớn.

Ghi chú:

FO - Độ dịch đoạn

IHL - Độ dài phần mào đầu internet

DF - Cờ chỉ báo không phân đoạn

MF - Cờ chỉ báo còn đoạn

TL - Độ dài tổng

OFO - Độ dịch của đoạn cũ

OIHL - Độ dài của phần mào đầu internet cũ

OMF - Cờ chỉ báo còn đoạn cũ

OTL - Độ dài tổng cũ

NFB - Số khối của đoạn

MTU - Đơn vị truyền tối đa

Thủ tục:

IF TL = < MTU THEN đưa gói tin này đến bước kế tiếp trong quá trình xử lý gói tin ELSE IF DF = 1 THEN loại bỏ gói tin ELSE

Để tạo ra đoạn đầu tiên:

(1) Sao chép phần mào đầu internet gốc

(2) OIHL <- IHL; OTL <- TL; OFO <- FO; OMF <- MF;

(3) NFB <- (MTU - IHL*4)/8;

(4) Gắn NFB * 8 octet dữ liệu đầu tiên;

(5) Hiệu chỉnh phần mào đầu:

MF <-1; TL<- (IHL*4) + (NFB*8);

Tính toán lại Checksum;

(6) Đưa đoạn này đến bước kế tiếp trong quá trình xử lý gói tin;

Để tạo ra đoạn thứ hai:

(7) Sao chép có lựa chọn phần mào đầu internet (một số tùy chọn không được sao chép, xem các định nghĩa về tùy chọn);

(8) Nối thêm dữ liệu còn lại;

(9) Hiệu chỉnh phần mào đầu:

IHL<- (((OIHL*4) - (độ dài của các tùy chọn không được sao chép)) + 3)/4;

TL<- OTL – NFB*8 - (OIHL - IHL)*4);

FO<- OFO + NFB; MF<- OMF; Tính toán lại Checksum;

(10) Đưa đoạn này đến phép thử phân đoạn; done.

Trong thủ tục trên, mỗi đoạn (trừ đoạn cuối) được thiết lập với kích thước tối đa cho phép. Một cách khác có thể tạo các gói tin có kích cỡ nhỏ hơn kích cỡ tối đa. Ví dụ, người ta có thể thực thi một thủ tục phân đoạn mà lặp lại việc chia đôi các gói tin lớn cho đến khi các đoạn thu được có kích cỡ nhỏ hơn kích cỡ của đơn vị truyền tối đa.

4.2.2.2.2 Ví dụ về thủ tục tái lắp ráp

Đối với mỗi gói tin, từ nhận dạng bộ đệm được tính toán như sự ghép nối của các trường nguồn, đích, giao thức, nhận dạng. Nếu đó là một gói tin nguyên vẹn (tức là cả trường còn nhiều đoạn nữa và trường độ dịch đoạn đều bằng 0) thì bất cứ tài nguyên tái lắp ráp nào gắn với từ nhận dạng đệm này cũng đều được giải phóng và gói tin được chuyển tiếp đến bước kế tiếp trong quá trình xử lý gói tin.

Nếu không có sẵn đoạn nào khác cùng với từ nhận dạng đệm này thì tài nguyên tái lắp ráp được phân bổ. Tài nguyên tái lắp ráp gồm có một bộ đệm dữ liệu, một bộ đệm phần mào đầu, một bảng bit của khối đoạn, một trường độ dài dữ liệu tổng và một bộ định thời. Dữ liệu từ đoạn được đặt vào bộ đệm dữ liệu căn cứ theo độ dài và độ dịch đoạn của nó và các bit được thiết lập trong bảng bit của khối đoạn tương ứng với các khối đoạn đã nhận được.

Nếu đó là đoạn đầu tiên (tức là độ dịch đoạn bằng 0) thì phần mào đầu này được đặt vào bộ đệm của phần mào đầu. Nếu đó là đoạn cuối cùng (tức là trường còn nhiều đoạn nữa bằng 0) thì độ dài của dữ liệu tổng được tính toán. Nếu đoạn này hoàn thành trọn vẹn gói tin (được thử nghiệm bằng cách kiểm tra các bit đã được thiết lập trong bảng khối đoạn) thì gói tin được gửi đến bước kế tiếp trong quá trình xử lý gói tin; nếu không thì bộ định thời được thiết lập bằng trị số cực đại của giá trị bộ định thời hiện thời và giá trị trường thời gian sống từ đoạn này và thủ tục tái lắp ráp sẽ thôi không điều khiển nữa.

Nếu bộ định thời chạy hết thời gian, mọi tài nguyên tái lắp ráp cho từ nhận dạng đệm này được giải phóng. Giá trị ban đầu của bộ định thời là biên thấp hơn trong khoảng thời gian chờ tái lắp ráp. Sở dĩ như vậy là vì thời gian chờ sẽ tăng lên nếu Thời gian sống trong đoạn đang tới lớn hơn nhiều giá trị của bộ định thời hiện tại nhưng thời gian chờ sẽ không giảm nếu Thời gian sống trong đoạn tới nhỏ hơn giá trị thực của bộ định thời. Trị số tối đa mà bộ định thời này có thể đạt tới là thời gian sống tối đa (xấp xỉ 4,25 phút). Khuyến nghị hiện tại là thiết lập giá trị ban đầu của bộ định thời bằng 15 giây. Giá trị này có thể thay đổi theo kinh nghiệm làm việc với giao thức này. Chú ý rằng việc lựa chọn giá trị của tham số này có liên quan đến dung lượng khả dụng của bộ đệm và tốc độ dữ liệu của môi trường truyền; tức là, tốc độ dữ liệu nhân với giá trị của bộ định thời bằng kích cỡ của bộ đệm (ví dụ, 10 kbit/s * 15 s = 150 kbit).

Ghi chú:

FO - Độ dịch đoạn

IHL - Độ dài phần mào đầu internet

MF - Cờ chỉ báo còn đoạn

TTL - Thời gian sống

TL - Độ dài tổng

NFB - Số khối của đoạn

TDL - Độ dài dữ liệu tổng

BUFID - Từ nhận dạng bộ đệm

RCVBT - Bảng bit nhận được của đoạn

TLB - Giới hạn dưới của bộ định thời

Thủ tục:

(1) BUFID <- nguồn | đích| giao thức | nhận dạng

(2) IF FO = 0 AND MF = 0

(3) THEN IF bộ đệm với BUFID được phân bổ

(4) THEN xóa sạch mọi sự tái lắp ráp đối với BUFID này;

(5) Đưa gói tin đến bước kế tiếp; DONE.

(6) ELSE IF không có bộ đệm nào cùng với BUFID được phân bổ

(7) THEN phân bổ tài nguyên tái lắp ráp với BUFID;

TIMER <- TLB; TDL<- 0;

(8) Đặt dữ liệu từ đoạn vào bộ đệm dữ liệu với

BUFID từ octet FO*8 đến

octet (TL – (IHL*4) + FO*8);

(9) Đặt các bit RCVBT từ FO

đến FO + ((TL – (IHL*4) + 7)/8);

(10) IF MF = 0 THEN TDL <- TL - (IHL*4) + (FO*8);

(11) IF FO = 0 THEN đặt phần mào đầu vào bộ đệm phần mào đầu

(12) IF TDL # 0

(13) AND tất cả các bit RCVBT từ 0 đến (TDL + 7)/8 được thiết lập

(14) THEN TL <- TDL + (IHL*4)

(15) Đưa gói tin đến bước kế tiếp;

(16) Giải phóng mọi tài nguyên tái lắp ráp

cho BUFID này; DONE.

(17) TIMER <- MAX (TIMER, TTL);

(18) Bỏ cho đến khi đoạn kế tiếp hoặc cho đến khi bộ định thời hết hạn;

(19) Bộ định thời hết hạn: xóa sạch mọi sự tái lắp ráp đối với BUFID này; DONE.

Trong trường hợp hai hoặc nhiều đoạn chứa cùng một dữ liệu hoặc giống hệt nhau hoặc do sự chồng chéo một phần thì thủ tục này sẽ sử dụng bản sao đã tới gần nhất trong bộ đệm dữ liệu và gói tin đã phân phát.

4.2.2.3 Nhận dạng

Việc lựa chọn từ nhận dạng cho một gói tin được thực hiện theo nhu cầu nhận dạng duy nhất các đoạn của một gói tin cụ thể. Mô-đun giao thức đang tái lắp ráp các đoạn xét thấy các đoạn thuộc về cùng một gói tin nếu chúng có cùng một nguồn, đích, giao thức và từ nhận dạng. Vì vậy, bên gửi phải chọn dùng từ nhận dạng là đơn nhất đối với cặp nguồn, đích và giao thức trong thời gian gói tin (hoặc bất cứ đoạn nào của nó) có thể vẫn còn tồn tại trong internet.

Như vậy một mô-đun giao thức gửi cần chứa một bảng các từ nhận dạng, một mục nhập cho mỗi đích mà nó đã liên lạc với trong thời gian sống tối đa của gói cuối cùng đối với internet.

Tuy nhiên, vì trường của từ nhận dạng cho phép 65536 giá trị khác nhau, nên một số máy chủ có thể đơn giản sử dụng các từ nhận dạng đơn nhất không phụ thuộc vào đích.

Việc lựa chọn từ nhận dạng là thích hợp với một số giao thức mức cao hơn. Ví dụ, các mô-đun giao thức TCP có thể phát lại một đoạn TCP giống hệt và xác suất thu đúng sẽ tăng nếu quá trình phát lại chứa cùng một từ nhận dạng như quá trình phát ban đầu vì các đoạn của một trong hai gói tin có thể được sử dụng để kết cấu một đoạn TCP đúng.

4.2.2.4 Loại dịch vụ

Loại dịch vụ (TOS) phục vụ việc lựa chọn chất lượng dịch vụ internet. Loại dịch vụ được chỉ định theo các tham số trừu tượng: thứ tự ưu tiên, độ trễ, thông lượng và độ tin cậy. Các tham số trừu tượng này cần được ánh xạ vào các tham số dịch vụ thực của các mạng cụ thể mà gói tin đi ngang qua.

Thứ tự ưu tiên. Thước đo độc lập về tầm quan trọng của gói tin này.

Độ trễ. Việc phân phát nhanh là quan trọng đối với các gói tin có chỉ dẫn này.

Thông lượng. Tốc độ dữ liệu cao là quan trọng đối với gói tin có chỉ dẫn này.

Độ tin cậy. Mức độ cố gắng cao hơn để đảm bảo cho việc phân phát là quan trọng đối với các gói tin có chỉ dẫn này.

Ví dụ, ARPANET có một bit ưu tiên, và một sự lựa chọn giữa các bản tin "chuẩn" (loại 0) và các bản tin "không bị kiểm soát" (loại 3), (sự lựa chọn giữa các bản tin đa gói và đơn gói cũng có thể được xem như một tham số dịch vụ). Các bản tin không bị kiểm soát có chiều hướng được phân phát kém tin cậy hơn và chịu độ trễ nhỏ hơn. Giả sử một gói tin internet cần được gửi qua ARPANET. Giả sử loại dịch vụ internet được cho như sau:

Thứ tự ưu tiên: 5

Độ trễ: 0

Thông lượng: 1

Độ tin cậy: 1

Trong trường hợp này, việc ánh xạ các tham số này lên các tham số khả dụng đối với ARPANET sẽ như thế nào đó để thiết lập bit ưu tiên của ARPANET lên trên (vì thứ tự ưu tiên của internet nằm ở nửa trên trong dải của nó) để lựa chọn các bản tin chuẩn vì các yêu cầu về thông lượng và độ tin cậy đã được chỉ báo, còn yêu cầu về độ trễ thì không.

4.2.2.5 Thời gian sống

Thời gian sống được thiết lập bởi bên gửi bằng thời gian tối đa mà gói tin được phép tồn tại trong hệ thống internet. Nếu gói tin tồn tại trong hệ thống internet lâu hơn thời gian sống thì gói tin phải bị loại bỏ.

Trường này phải bị giảm tại mỗi điểm mà phần mào đầu internet được xử lý, để phản ánh về thời gian đã dùng để xử lý gói tin. Dù không có sẵn thông tin nội bộ nào về thời gian đã dùng thực tế thì trường này cũng phải bị giảm đi 1. Thời gian được đo theo đơn vị là giây (giá trị 1 nghĩa là 1 giây). Như vậy, thời gian sống tối đa là 255 giây hoặc 4,25 phút. Vì mỗi mô-đun xử lý một gói tin phải giảm TTL đi ít nhất là 1 giây dù cho nó xử lý gói tin trong thời gian ít hơn 1 giây, TTL phải được hiểu chỉ như một giới hạn trên về thời gian một gói tin có thể tồn tại. Mục đích là khiến cho các gói tin không thể phân phát được phải bị loại bỏ và để giới hạn thời gian sống tối đa của gói tin.

Một số giao thức kết nối tin cậy ở mức cao hơn đã dựa trên các giả thiết rằng các gói tin sao lại cũ sẽ không đến nữa sau một khoảng thời gian nào đó trôi qua. TTL là một phương pháp dành cho các giao thức như vậy để đảm bảo rằng giả thiết của chúng được thỏa mãn.

4.2.2.6 Các tùy chọn

Các tùy chọn là tùy chọn trong mỗi gói tin nhưng chúng được yêu cầu trong khi thực thi. Tức là, việc một tùy chọn xuất hiện hoặc không xuất hiện là do lựa

chọn của bên gửi, nhưng mỗi mô-đun internet phải có khả năng phân tích mỗi tùy chọn. Có thể có một vài tùy chọn trong trường tùy chọn.

Các tùy chọn có thể không kết thúc trên một biên 32 bit. Phần mào đầu internet phải điền các octet của các số 0 vào. Octet đầu tiên trong số những octet này có thể được hiểu là tùy chọn kết-thúc-các-tùy-chọn, và phần còn lại được hiểu là đệm của phần mào đầu internet.

Mỗi mô-đun internet phải có khả năng tác động đến mỗi tùy chọn. Tùy chọn bảo mật được yêu cầu nếu cần chuyển lưu lượng mật, lưu lượng cấm, hoặc lưu lượng đã phân chia ngăn.

4.2.2.7 Kiểm tra tổng

Kiểm tra tổng phần mào đầu internet được tính toán lại nếu phần mào đầu internet bị thay đổi. (Ví dụ, sự giảm thời gian sống, những sự bổ sung hoặc thay đổi đối với Các tùy chọn internet, hoặc do sự phân đoạn). Kiểm tra tổng này ở mức internet nhằm bảo vệ các trường của phần mào đầu internet khỏi bị các lỗi trong quá trình truyền.

Có một số ứng dụng chấp nhận một vài lỗi bit dữ liệu trong khi không chấp nhận các trễ do quá trình phát lại. Nếu giao thức internet đòi hỏi sự chính xác của dữ liệu thì các ứng dụng như vậy không thể được hỗ trợ.

4.2.2.8 Các lỗi

Các lỗi của giao thức internet có thể được thông báo qua các bản tin ICMP.

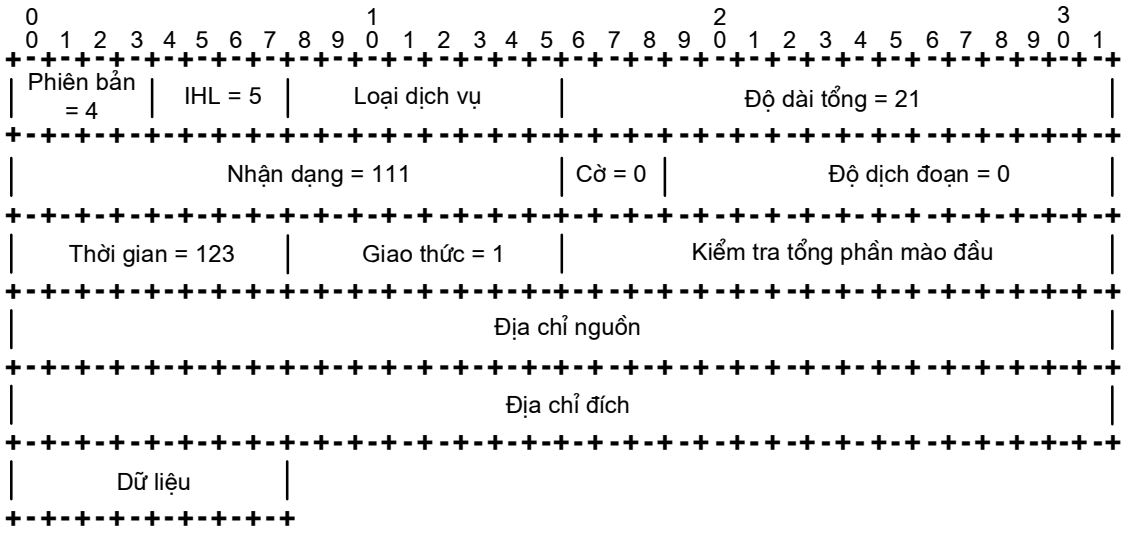
4.2.3 Các giao diện

Mô tả chức năng của các giao diện người dùng đối với IP là sự mô tả tưởng tượng, vì mỗi hệ điều hành sẽ có các phương tiện khác nhau. Những sự thực thi IP khác nhau có thể có các giao diện người dùng khác nhau. Tuy vậy, mọi IP phải cung cấp một tập hợp tối thiểu nào đó của các dịch vụ để bảo đảm rằng mọi sự thực thi của IP có thể hỗ trợ cùng một sự phân cấp của giao thức. Mục này chỉ rõ các giao diện chức năng được yêu cầu trong mọi sự thực thi của IP.

Giao thức internet, một phía giao diện với mạng cục bộ và phía kia giao diện với một chương trình ứng dụng hoặc với một giao thức mức cao hơn. Trong phần tiếp sau đây, giao thức mức cao hơn hoặc chương trình ứng dụng (hoặc ngay cả một chương trình cổng) cũng sẽ được gọi là "người dùng" vì nó sử dụng mô-đun internet. Vì giao thức internet là một giao thức gói tin, nên sẽ có bộ nhớ tối thiểu hoặc một trạng thái được duy trì giữa các quá trình truyền gói tin, và mỗi yêu cầu mô-đun giao thức internet bởi người dùng sẽ cung cấp toàn bộ thông tin cần thiết cho IP để thực hiện dịch vụ yêu cầu.

PHỤ LỤC A
CÁC VÍ DỤ VÀ KỊCH BẢN

***Ví dụ 1**:* Đây là một ví dụ về gói tin internet mang dữ liệu tối thiểu:



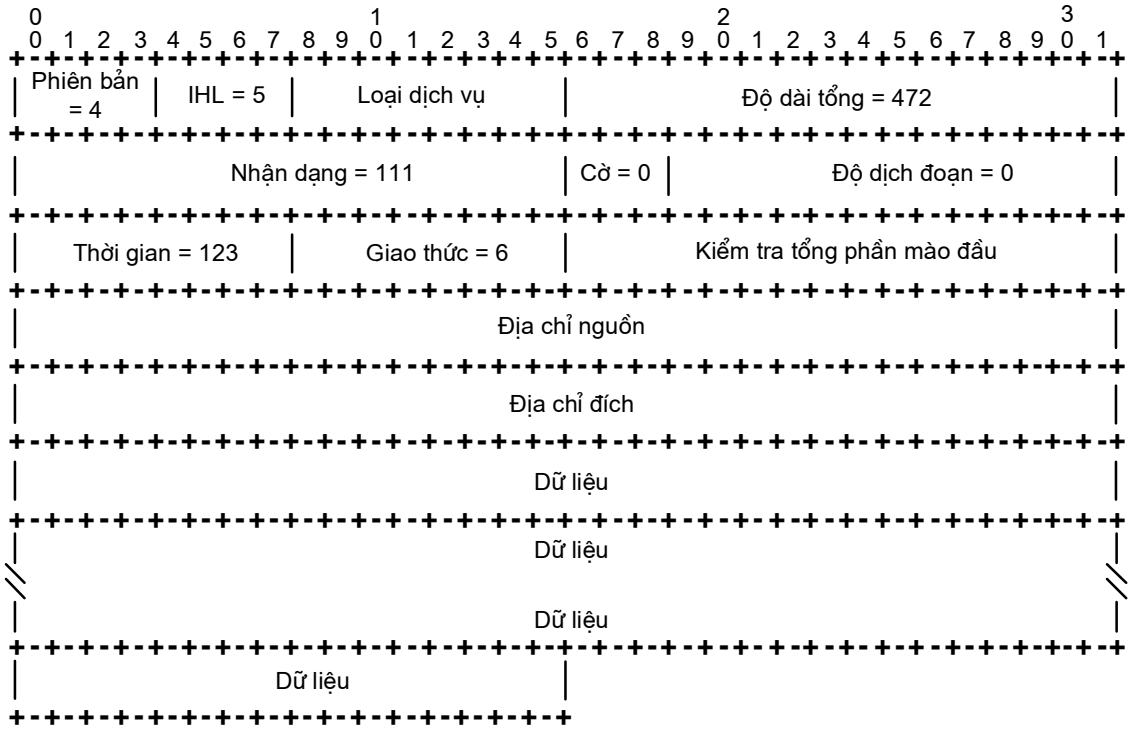
Hình 5: Ví dụ về gói tin internet

Chú ý rằng mỗi dấu phân thời biểu diễn một vị trí bit.

Đây là một gói tin internet trong phiên bản 4 của giao thức internet; phần mào đầu internet gồm có năm từ 32 bit và độ dài tổng của gói tin là 21 octet. Gói tin này là một gói tin đầy đủ (không phải một đoạn).

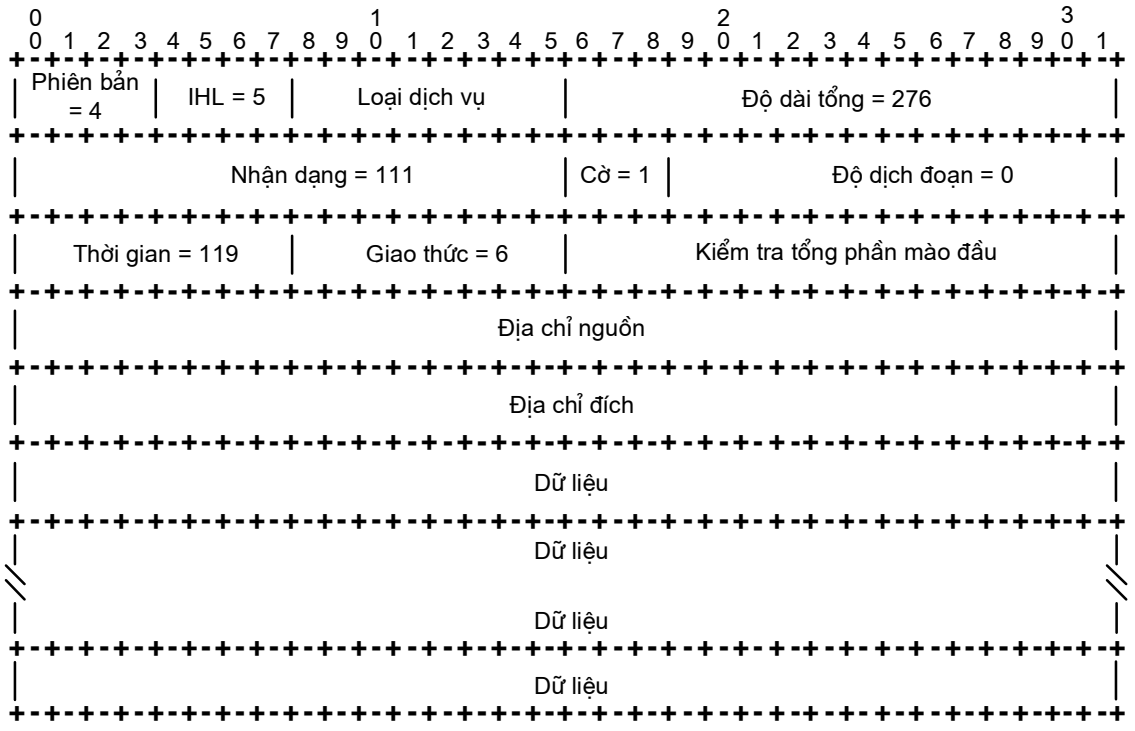
***Ví dụ 2**:*

Trong ví dụ này, trước tiên là một gói tin internet có kích cỡ vừa phải (452 octet dữ liệu), sau đó là hai đoạn internet (có thể là do sự phân đoạn của gói tin này) nếu kích cỡ tối đa được phép truyền là 280 octet).



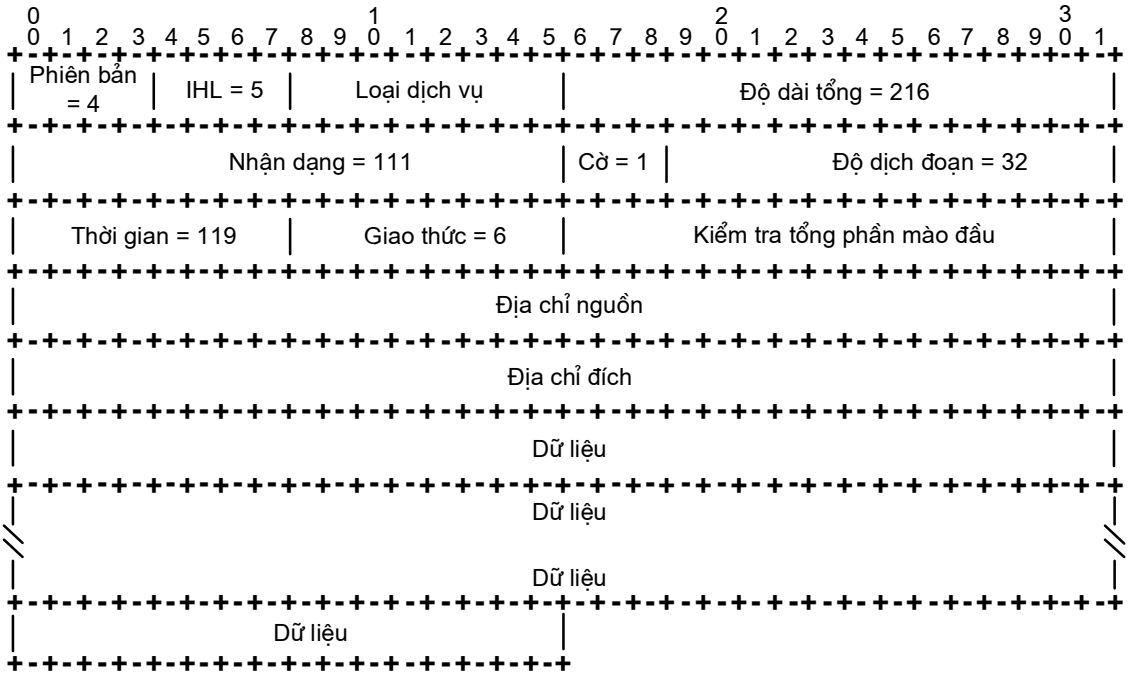
Hình 6: Ví dụ về gói tin internet

Lúc này đoạn đầu tiên là kết quả của việc phân tách gói tin sau 256 octet dữ liệu.



Hình 7: Ví dụ về đoạn internet

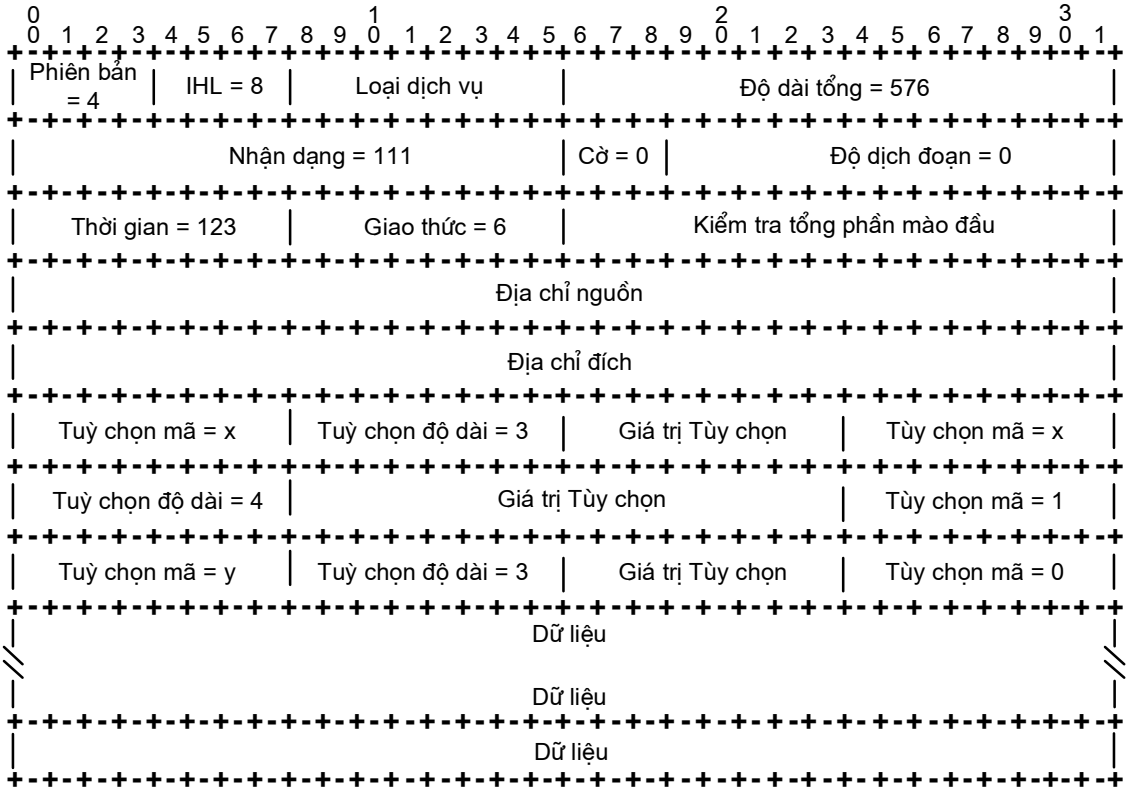
Và đoạn thứ hai



Hình 8: Ví dụ về đoạn internet

***Ví dụ 3**:*

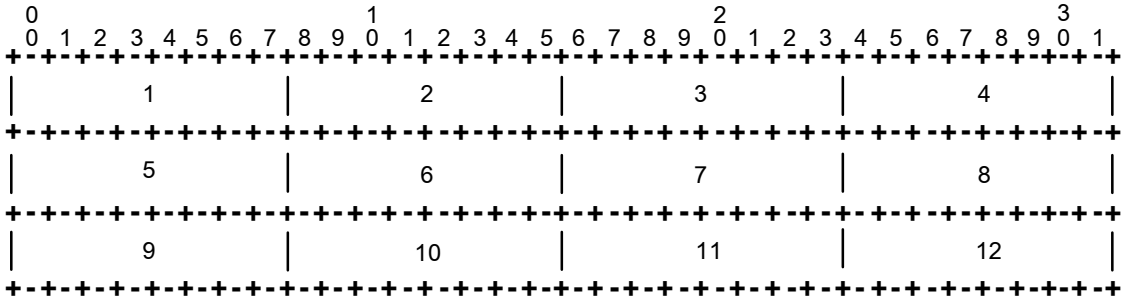
Đây là một ví dụ về gói tin chứa các tùy chọn:



Hình 9: Ví dụ về gói tin internet

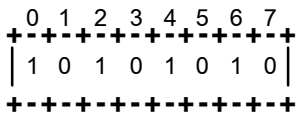
PHỤ LỤC B
THỨ TỰ TRUYỀN DỮ LIỆU

Thứ tự truyền phần mào đầu và dữ liệu đã mô tả trong tài liệu này được quyết định theo mức octet. Bất cứ khi nào một sơ đồ cho thấy một nhóm các octet, thì thứ tự truyền các octet đó là thứ tự bình thường mà chúng được đọc bằng tiếng Anh. Ví dụ, trong sơ đồ sau đây các octet được truyền theo thứ tự chúng được đánh số.



Hình 10: Thứ tự truyền các byte

Bất cứ khi nào một octet biểu diễn một đại lượng bằng số thì bit tận cùng bên trái trong sơ đồ là bit có trọng số cao nhất hoặc bit bậc cao. Tức là, bit được dán nhãn 0 là bit có trọng số cao nhất. Ví dụ, sơ đồ sau đây biểu diễn giá trị 170 (decimal).



Hình 11: Ý nghĩa của các bit

Cũng vậy, bất cứ khi nào một trường đa octet biểu diễn một đại lượng bằng số thì bit tận cùng bên trái của toàn bộ trường là bit có trọng số cao nhất. Khi một đại lượng đa octet được truyền thì octet có trọng số cao nhất được truyền trước tiên.

PHỤ LỤC C
VÍ DỤ VỀ GIAO DIỆN MỨC TRÊN

Hai lệnh ví dụ sau đây thỏa mãn các yêu cầu cho người dùng truyền thông mô-đun giao thức internet ("=>" có nghĩa là trả về):

SEND (src, dst, prot, TOS, TTL, BufPTR, len, Id, DF, opt => result)

Trong đó:

src = địa chỉ nguồn

dst = địa chỉ đích

prot = giao thức

TOS = loại dịch vụ

TTL = thời gian sống

BufPTR = con trỏ đệm

len = độ dài bộ đệm

Id = từ nhận dạng

DF = không phân đoạn

opt = dữ liệu tùy chọn

result = tín hiệu đáp ứng

OK = gói tin đã được gửi tốt

Error = lỗi trong các đối số hoặc lỗi của mạng cục bộ

Chú ý rằng thứ tự ưu tiên được tính đến trong TOS và tính bảo mật/phân chia ngăn được chấp nhận như một tùy chọn.

RECV (BufPTR, prot, => result, src, dst, TOS, len, opt)

Trong đó:

BufPTR = con trỏ đệm

prot = giao thức

result = tín hiệu đáp ứng

OK = gói tin đã được nhận tốt

Error = lỗi trong các đối số

len = độ dài bộ đệm

src = địa chỉ nguồn

dst = địa chỉ đích

TOS = loại dịch vụ

opt = dữ liệu tùy chọn

Khi người dùng gửi một gói tin, nó thực hiện một lệnh SEND cung cấp mọi đối số. Mô-đun giao thức internet, khi nhận được lệnh này, kiểm tra các đối số, chuẩn bị và gửi bản tin. Nếu các đối số là đúng và gói tin được mạng cục bộ chấp nhận thì lệnh phản hồi thành công. Nếu các đối số là sai, hoặc gói tin không được mạng cục bộ chấp nhận thì lệnh phản hồi không thành công. Khi các lệnh phản hồi không thành công, phải thông báo về nguyên nhân của vấn đề, nhưng chi tiết của các báo cáo như vậy là tuỳ thuộc vào những sự thực thi riêng.

Khi một gói tin từ mạng cục bộ đến mô-đun giao thức internet thì sẽ có hoặc không có một lệnh RECV đang treo từ người dùng đã lập địa chỉ. Trong trường hợp đầu tiên, lệnh đang treo được đáp ứng bằng cách chuyển thông tin từ gói tin đến người dùng. Trong trường hợp thứ hai, người dùng đã lập địa chỉ được thông báo về một gói tin đang treo. Nếu người dùng đã lập địa chỉ không tồn tại thì một bản tin báo lỗi của ICMP được phản hồi cho bên gửi và dữ liệu bị loại bỏ.

Thông báo về một người dùng có thể qua cơ chế ngắt giả hoặc một cơ chế tương tự, thích hợp với việc thực thi trong môi trường của hệ điều hành cụ thể.

Một lệnh RECV của người dùng sau đó có thể được một gói tin đang treo đáp ứng ngay lập tức, hoặc lệnh có thể treo cho đến khi một gói tin đến.

Địa chỉ nguồn được bao hàm lệnh SEND đề phòng trường hợp máy chủ gửi có vài địa chỉ (nhiều kết nối vật lý hoặc nhiều địa chỉ logic). Mô-đun internet phải kiểm tra xem địa chỉ nguồn có là một trong số các địa chỉ hợp lệ đối với máy chủ này hay không.

Một thực thi cũng có thể cho phép hoặc yêu cầu một lệnh đến mô-đun internet để cho biết sự quan tâm hoặc để đăng ký trước việc sử dụng độc quyền một loại gói tin (ví dụ tất cả gói tin đó có một giá trị nào đó trong trường giao thức).

Mục này mô tả đặc điểm chức năng một giao diện USER/IP. Ký hiệu được sử dụng là tương tự với đa số thủ tục của các lệnh chức năng trong các ngôn ngữ mức cao, nhưng cách sử dụng này không có ý định loại trừ các lệnh của dịch vụ kiểu bẫy (ví dụ SVCs, UUOs, EMTs), hoặc bất cứ dạng truyền thông liên chương trình nào khác.

FOREWORD

The technical standard TCN 68 - 224: 2004 **"Interconnecting Protocol between GSM GPRS network and Internet (IP Protocol) – Technical Requirements"** is based on the IETF RFC 791 (1981) of the Internet Engineering Task Force (IETF).

The Technical Standard TCN 68-224: 2004 is drafted by Research Institute of Posts and Telecommunications (RIPT) at proposal of the Department of Science & Technology of Ministry of Posts and Telematics. The technical standard is adopted by the Decision No 33/2004/QD-BBCVT dated 29/7/2004 of the Minister of Posts and Telematics.

The Technical Standard TCN 68-224: 2004 is issued in a bilingual document (Vietnamese version and English version). In cases of interpretation disputes, Vietnamese version is applied.

DEPARTMENT OF SCIENCE & TECHNOLOGY

INTERCONNECTING PROTOCOL BETWEEN GSM GPRS NETWORK AND INTERNET (IP PROTOCOL)

TECHNICAL REQUIREMENTS

(Issued together with the Decision No. 33/2004/QD-BBCVT dated 29/7/2004 of the Minister of Posts and Telematics)

1. Motivation and scope

1.1. Motivation

This technical standard specifies the essential specifications required for the interconnecting protocol between GSM GPRS networks and the Internet (internet protocol – IP), to ensure that the interconnecting and interworking capabilities between GSM GPRS networks and the Internet are effective, and to serve the network interconnection management of telecommunications operators.

The internet protocol is designed for use in interconnected systems of packet-switched computer communication networks. Such a system has been called a "catenet". The internet protocol provides for transmitting blocks of data called datagrams from sources to destinations, where sources and destinations are hosts identified by fixed length addresses. The internet protocol also provides for fragmentation and reassembly of long datagrams, if necessary, for transmission through "small packet" networks.

1.2. Scope

The internet protocol is specifically limited in scope to provide the functions necessary to deliver a package of bits (an internet datagram) from a source to a destination over an interconnected system of networks. There are no mechanisms to augment end-to-end data reliability, flow control, sequencing, or other services commonly found in host-to-host protocols. The internet protocol can capitalize on the services of its supporting networks to provide various types and qualities of service.

2. References

[1] ETSI TS 101 348 V7.3.0 (3/2001), "Digital cellular telecommunications system (phase 2+); General Packet Radio Service (GPRS); Interwoking between

the Public Land Mobile Network (PLMN) supporting GPRS and Packet Data Networks (PDN) (3GPP TS 09.61 version 7.3.0 Release 1998)".

[2] IETF RFC 791 (1981): "Internet protocol" (STD5).

3. Glossary

1822

BBN Report 1822, "The Specification of the Interconnection of a Host and an IMP". The specification of interface between a host and the ARPANET.

ARPANET leader

The control information on an ARPANET message at the host-IMP interface.

ARPANET message

The unit of transmission between a host and an IMP in the ARPANET. The maximum size is about 1012 octets (8096 bits).

ARPANET packet

A unit of transmission used internally in the ARPANET between IMPs. The maximum size is about 126 octets (1008 bits).

Destination

The destination address, an internet header field.

DF

The Don't Fragment bit carried in the flags field.

Flags

An internet header field carrying various control flags.

Fragment Offset

This internet header field indicates where in the internet datagram a fragment belongs.

GGP

Gateway to Gateway Protocol, the protocol used primarily between gateways to control routing and other gateway functions.

Header

Control information at the beginning of a message, segment, datagram, packet or block of data.

ICMP

Internet Control Message Protocol, implemented in the internet module, the ICMP is used from gateways to hosts and between hosts to report errors and make routing suggestions.

Identification

An internet header field carrying the identifying value assigned by the sender to aid in assembling the fragments of a datagram.

IHL

The internet header field Internet Header Length is the length of the internet header measured in 32 bit words.

IMP

The Interface Message Processor, the packet switch of the ARPANET.

Internet Address

A four octet (32 bit) source or destination address consisting of a Network field and a Local Address field.

Internet datagram

The unit of data exchanged between a pair of internet modules (includes the internet header).

Internet fragment

A portion of the data of an internet datagram with an internet header.

Local Address

The address of a host within a network. The actual mapping of an internet local address on to the host addresses in a network is quite general, allowing for many to one mappings.

MF

The More-Fragments Flag carried in the internet header flags field.

Module

An implementation, usually in software, of a protocol or other procedure.

More-fragments flag

A flag indicating whether or not this internet datagram contains the end of an internet datagram, carried in the internet header Flags field.

NFB

The Number of Fragment Blocks in a the data portion of an internet fragment. That is, the length of a portion of data measured in 8 octet units.

Octet

An eight bit byte.

Options

The internet header Options field may contain several options, and each option may be several octets in length.

Padding

The internet header Padding field is used to ensure that the data begins on 32 bit word boundary. The padding is zero.

Protocol

In this document, the next higher level protocol identifier, an internet header field.

Rest

The local address portion of an Internet Address.

Source

The source address, an internet header field.

TCP

Transmission Control Protocol: A host-to-host protocol for reliable communication in internet environments.

TCP Segment

The unit of data exchanged between TCP modules (including the TCP header).

TFTP

Trivial File Transfer Protocol: A simple file transfer protocol built on UDP.

Time to Live

An internet header field which indicates the upper bound on how long this internet datagram may exist.

TOS

Type of Service

Total Length

The internet header field Total Length is the length of the datagram in octets including internet header and data.

TTL

Time to Live

Type of Service

An internet header field which indicates the type (or quality) of service for this internet datagram.

UDP

User Datagram Protocol: A user level protocol for transaction oriented applications.

User

The user of the internet protocol. This may be a higher level protocol module, an application program, or a gateway program.

Version

The Version field indicates the format of the internet header.

4. Technical Requirements

4.1. General Requirements

4.1.1 Interfaces

This protocol is called on by host-to-host protocols in an internet environment. This protocol calls on local network protocols to carry the internet datagram to the next gateway or destination host.

For example, a TCP module would call on the internet module to take a TCP segment (including the TCP header and user data) as the data portion of an internet datagram. The TCP module would provide the addresses and other parameters in the internet header to the internet module as arguments of the call. The internet module would then create an internet datagram and call on the local network interface to transmit the internet datagram.

In the ARPANET case, for example, the internet module would call on a local net module which would add the 1822 leader to the internet datagram creating an ARPANET message to transmit to the IMP. The ARPANET address would be

derived from the internet address by the local network interface and would be the address of some host in the ARPANET, that host might be a gateway to other networks.

4.1.2 Operation

The internet protocol implements two basic functions: addressing and fragmentation.

The internet modules use the addresses carried in the internet header to transmit internet datagrams toward their destinations. The selection of a path for transmission is called routing.

The internet modules use fields in the internet header to fragment and reassemble internet datagrams when necessary for transmission through "small packet" networks.

The model of operation is that an internet module resides in each host engaged in internet communication and in each gateway that interconnects networks. These modules share common rules for interpreting address fields and for fragmenting and assembling internet datagrams. In addition, these modules (especially in gateways) have procedures for making routing decisions and other functions.

The internet protocol treats each internet datagram as an independent entity unrelated to any other internet datagram. There are no connections or logical circuits (virtual or otherwise).

The internet protocol uses four key mechanisms in providing its service: Type of Service, Time to Live, Options, and Header Checksum.

The Type of Service is used to indicate the quality of the service desired. The type of service is an abstract or generalized set of parameters which characterize the service choices provided in the networks that make up the internet. This type of service indication is to be used by gateways to select the actual transmission parameters for a particular network, the network to be used for the next hop, or the next gateway when routing an internet datagram.

The Time to Live is an indication of an upper bound on the lifetime of an internet datagram. It is set by the sender of the datagram and reduced at the points along the route where it is processed. If the time to live reaches zero before the internet datagram reaches its destination, the internet datagram is destroyed. The time to live can be thought of as a self destruct time limit.

The Options provide for control functions needed or useful in some situations but unnecessary for the most common communications. The options include provisions for timestamps, security, and special routing.

The Header Checksum provides a verification that the information used in processing internet datagram has been transmitted correctly. The data may contain errors. If the header checksum fails, the internet datagram is discarded at once by the entity which detects the error.

The internet protocol does not provide a reliable communication facility. There are no acknowledgments either end-to-end or hop-by-hop. There is no error control for data, only a header checksum. There are no retransmissions. There is no flow control.

Errors detected may be reported via the Internet Control Message Protocol (ICMP) which is implemented in the internet protocol module.

4.1.3 Relation to Other Protocols

The following diagram illustrates the place of the internet protocol in the protocol hierarchy:

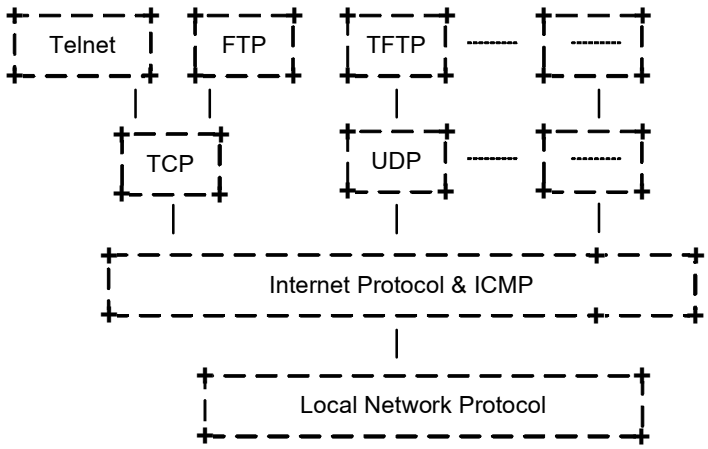


Figure 1: Protocol Relationships

Internet protocol interfaces on one side to the higher level host-to-host protocols and on the other side to the local network protocol. In this context a "local network" may be a small network in a building or a large network such as the ARPANET.

4.1.4 Model of Operation

The model of operation for transmitting a datagram from one application program to another is illustrated by the following scenario:

We suppose that this transmission will involve one intermediate gateway.

The sending application program prepares its data and calls on its local internet module to send that data as a datagram and passes the destination address and other parameters as arguments of the call.

The internet module prepares a datagram header and attaches the data to it. The internet module determines a local network address for this internet address, in this case it is the address of a gateway.

It sends this datagram and the local network address to the local network interface.

The local network interface creates a local network header, and attaches the datagram to it, then sends the result via the local network.

The datagram arrives at a gateway host wrapped in the local network header, the local network interface strips off this header, and turns the datagram over to the internet module. The internet module determines from the internet address that the datagram is to be forwarded to another host in a second network. The internet module determines a local net address for the destination host. It calls on the local network interface for that network to send the datagram.

This local network interface creates a local network header and attaches the datagram sending the result to the destination host.

At this destination host the datagram is stripped of the local net header by the local network interface and handed to the internet module.

The internet module determines that the datagram is for an application program in this host. It passes the data to the application program in response to a system call, passing the source address and other parameters as results of the call.

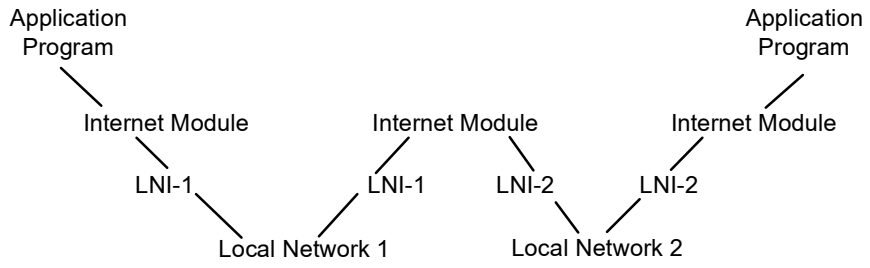


Figure 2: Transmission Path

4.1.5 Function Description

The function or purpose of Internet Protocol is to move datagrams through an interconnected set of networks. This is done by passing the datagrams from one

internet module to another until the destination is reached. The internet modules reside in hosts and gateways in the internet system. The datagrams are routed from one internet module to another through individual networks based on the interpretation of an internet address. Thus, one important mechanism of the internet protocol is the internet address.

In the routing of messages from one internet module to another, datagrams may need to traverse a network whose maximum packet size is smaller than the size of the datagram. To overcome this difficulty, a fragmentation mechanism is provided in the internet protocol.

4.1.5.1 Addressing

A distinction is made between names, addresses, and routes. A name indicates what we seek. An address indicates where it is. A route indicates how to get there. The internet protocol deals primarily with addresses. It is the task of higher level (i.e., host-to-host or application) protocols to make the mapping from names to addresses. The internet module maps internet addresses to local net addresses. It is the task of lower level (i.e., local net or gateways) procedures to make the mapping from local net addresses to routes.

Addresses are fixed length of four octets (32 bits). An address begins with a network number, followed by local address (called the "rest" field). There are three formats or classes of internet addresses: in class a, the high order bit is zero, the next 7 bits are the network, and the last 24 bits are the local address; in class b, the high order two bits are one-zero, the next 14 bits are the network and the last 16 bits are the local address; in class c, the high order three bits are one-one-zero, the next 21 bits are the network and the last 8 bits are the local address.

Care must be taken in mapping internet addresses to local net addresses; a single physical host must be able to act as if it were several distinct hosts to the extent of using several distinct internet addresses. Some hosts will also have several physical interfaces (multi-homing).

That is, provision must be made for a host to have several physical interfaces to the network with each having several logical internet addresses.

4.1.5.2 Fragmentation

Fragmentation of an internet datagram is necessary when it originates in a local net that allows a large packet size and must traverse a local net that limits packets to a smaller size to reach its destination.

An internet datagram can be marked "don't fragment." Any internet datagram so marked is not to be internet fragmented under any circumstances. If internet datagram marked don't fragment cannot be delivered to its destination without fragmenting it, it is to be discarded instead.

Fragmentation, transmission and reassembly across a local network which is invisible to the internet protocol module is called intranet fragmentation and may be used.

The internet fragmentation and reassembly procedure needs to be able to break a datagram into an almost arbitrary number of pieces that can be later reassembled. The receiver of the fragments uses the identification field to ensure that fragments of different datagrams are not mixed. The fragment offset field tells the receiver the position of a fragment in the original datagram. The fragment offset and length determine the portion of the original datagram covered by this fragment. The more-fragments flag indicates (by being reset) the last fragment. These fields provide sufficient information to reassemble datagrams.

The identification field is used to distinguish the fragments of one datagram from those of another. The originating protocol module of an internet datagram sets the identification field to a value that must be unique for that source-destination pair and protocol for the time the datagram will be active in the internet system. The originating protocol module of a complete datagram sets the more-fragments flag to zero and the fragment offset to zero.

To fragment a long internet datagram, an internet protocol module (for example, in a gateway), creates two new internet datagrams and copies the contents of the internet header fields from the long datagram into both new internet headers. The data of the long datagram is divided into two portions on a 8 octet (64 bit) boundary (the second portion might not be an integral multiple of 8 octets, but the first must be). Call the number of 8 octet blocks in the first portion NFB (for Number of Fragment Blocks). The first portion of the data is placed in the first new internet datagram, and the total length field is set to the length of the first datagram. The more-fragments flag is set to one. The second portion of the data is placed in the second new internet datagram, and the total length field is set to the length of the second datagram. The more-fragments flag carries the same value as the long datagram. The fragment offset field of the second new internet datagram is set to the value of that field in the long datagram plus NFB.

This procedure can be generalized for an n-way split, rather than the two-way split described.

To assemble the fragments of an internet datagram, an internet protocol module (for example at a destination host) combines internet datagrams that all

have the same value for the four fields: identification, source, destination, and protocol. The combination is done by placing the data portion of each fragment in the relative position indicated by the fragment offset in that fragment's internet header. The first fragment will have the fragment offset zero, and the last fragment will have the more-fragments flag reset to zero.

4.1.6 Gateways

Gateways implement internet protocol to forward datagrams between networks. Gateways also implement the Gateway to Gateway Protocol (GGP) to coordinate routing and other internet control information.

In a gateway the higher level protocols need not be implemented and the GGP functions are added to the IP module.

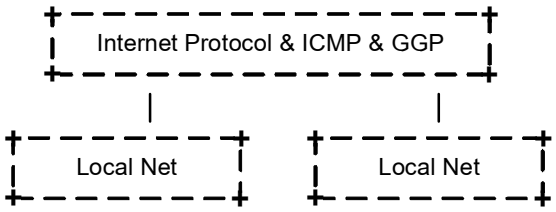


Figure 3: Gateway Protocols

4.2. Technical Requirements

4.2.1 Internet Header Format

A summary of the contents of the internet header follows:

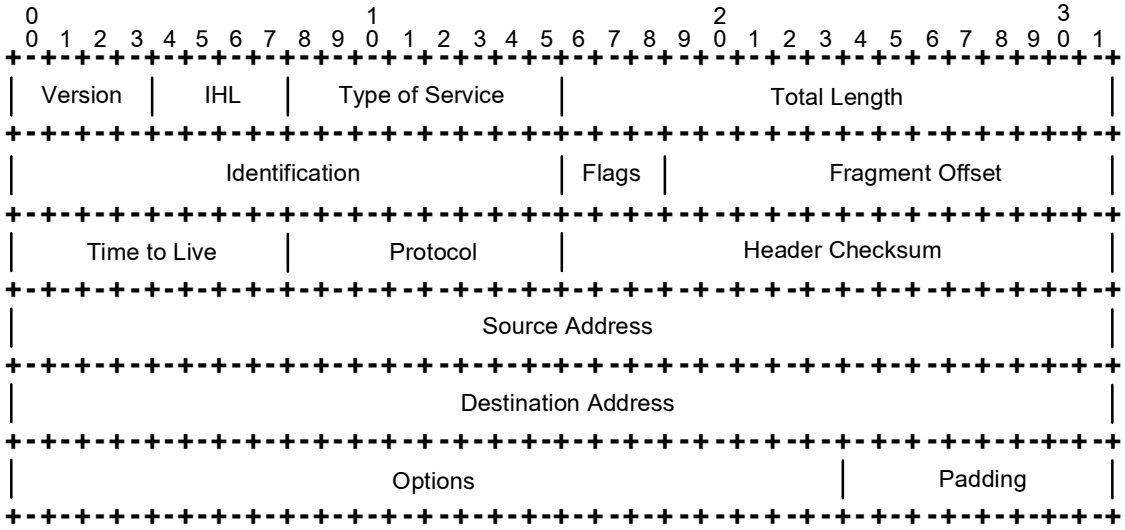


Figure 4: Example Internet Datagram Header

Note that each tick mark represents one bit position.

Version: 4 bits

The Version field indicates the format of the internet header. This document describes version 4.

***IHL*:** 4 bits

Internet Header Length is the length of the internet header in 32 bit words, and thus points to the beginning of the data. Note that the minimum value for a correct header is 5.

***Type of Service*:** 8 bits

The Type of Service provides an indication of the abstract parameters of the quality of service desired. These parameters are to be used to guide the selection of the actual service parameters when transmitting a datagram through a particular network. Several networks offer service precedence, which somehow treats high precedence traffic as more important than other traffic (generally by accepting only traffic above a certain precedence at time of high load). The major choice is a three way tradeoff between low-delay, high-reliability, and high-throughput.

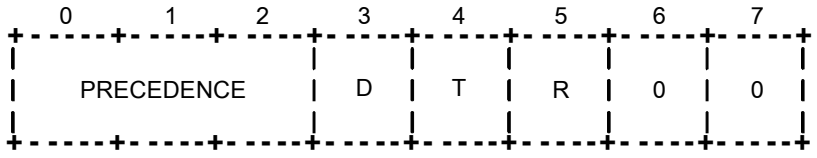
Bits 0-2: Precedence

Bit 3: 0 = Normal Delay, 1 = Low Delay

Bits 4: 0 = Normal Throughput, 1 = High Throughput

Bits 5: 0 = Normal Relibility, 1 = High Relibility

Bit 6-7: Reserved for Future Use



Precedence

111 - Network Control

110 - Internetwork Control

101 - CRITIC/ECP

100 - Flash Override

011 - Flash

010 - Immediate

001 - Priority

000 - Routine

The use of the Delay, Throughput, and Reliability indications may increase the cost (in some sense) of the service. In many networks better performance for one of these parameters is coupled with worse performance on another. Except for very unusual cases at most two of these three indications should be set.

The type of service is used to specify the treatment of the datagram during its transmission through the internet system.

The Network Control precedence designation is intended to be used within a network only. The actual use and control of that designation is up to each network. The Internetwork Control designation is intended for use by gateway control originators only. If the actual use of these precedence designations is of concern to a particular network, it is the responsibility of that network to control the access to, and use of, those precedence designations.

***Total Length*:** 16 bits

Total Length is the length of the datagram, measured in octets, including internet header and data. This field allows the length of a datagram to be up to 65,535 octets. Such long datagrams are impractical for most hosts and networks. All hosts must be prepared to accept datagrams of up to 576 octets (whether they arrive whole or in fragments). It is recommended that hosts only send datagrams larger than 576 octets if they have assurance that the destination is prepared to accept the larger datagrams.

The number 576 is selected to allow a reasonable sized data block to be transmitted in addition to the required header information. For example, this size allows a data block of 512 octets plus 64 header octets to fit in a datagram. The maximal internet header is 60 octets, and a typical internet header is 20 octets, allowing a margin for headers of higher level protocols.

***Identification*:** 16 bits

An identifying value assigned by the sender to aid in assembling the fragments of a datagram.

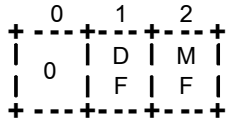
***Flags*:** 3 bits

Various Control Flags

Bit 0: reserved, must be zero

Bit 1: (DF) 0 = May Fragment, 1 = Don’t Fragment

Bit 2: (MF) 0 = Last Fragment, 1 = More Fragments



***Fragment Offset*:** 13 bits

This field indicates where in the datagram this fragment belongs.

The fragment offset is measured in units of 8 octets (64 bits). The first fragment has offset zero.

***Time to Live*:** 8 bits

This field indicates the maximum time the datagram is allowed to remain in the internet system. If this field contains the value zero, then the datagram must be destroyed. This field is modified in internet header processing. The time is measured in units of seconds, but since every module that processes a datagram must decrease the TTL by at least one even if it process the datagram in less than a second, the TTL must be thought of only as an upper bound on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded, and to bound the maximum datagram lifetime.

***Protocol*:** 8 bits

This field indicates the next level protocol used in the data portion of the internet datagram.

***Header Checksum*:** 16 bits

A checksum on the header only. Since some header fields change (e.g., time to live), this is recomputed and verified at each point that the internet header is processed.

The checksum algorithm is:

The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.

This is a simple to compute checksum and experimental evidence indicates it is adequate, but it is provisional and may be replaced by a CRC procedure, depending on further experience.

***Source Address*:** 32 bits

The source address. See section 3.2.

***Destination Address*:** 32 bits

The destination address. See section 3.2.

***Options*:** variable

The options may appear or not in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation.

In some environments the security option may be required in all datagrams.

The option field is variable in length. There may be zero or more options. There are two cases for the format of an option:

Case 1: A single octet of option-type.

Case 2: An option-type octet, an option-length octet, and the actual option-data octets.

The option-length octet counts the option-type octet and the option-length octet as well as the option-data octets.

The option-type octet is viewed as having 3 fields:

1 bit copied flag;

2 bits option class;

5 bits option number.

The copied flag indicates that this option is copied into all fragments on fragmentation.

0 = not copied

1 = copied

The option classes are:

0 = control

1 = reserved for future use

2 = debugging and measurement

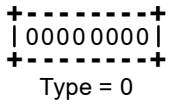
3 = reserved for future use

The following internet options are defined:

CLASS	NUMBER	LENGTH	DESCRIPTION
0	0	-	End of Option list. This option occupies only 1 octet; it has no length octet.
0	1	-	No Operation. This option occupies only 1 octet; it has no length octet.
0	2	11	Security. Used to carry Security, Compartmentation, User Group (TCC), and Handling Restriction Codes compatible with DOD requirements.
0	3	var.	Loose Source Routing. Used to route the internet datagram based on information supplied by the source.
0	9	var.	Strict Source Routing. Used to route the internet datagram based on information supplied by the source.
0	7	var.	Record Route. Used to trace the route an internet datagram takes.
0	8	4	Stream ID. Used to carry the stream identifier.
2	4	var.	Internet Timestamp.

Specific Option Definitions

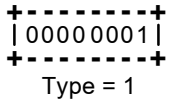
End of Option List



This option indicates the end of the option list. This might not coincide with the end of the internet header according to the internet header length. This is used at the end of all options, not the end of each option, and need only be used if the end of the options would not otherwise coincide with the end of the internet header.

May be copied, introduced, or deleted on fragmentation, or for any other reason.

No Operation

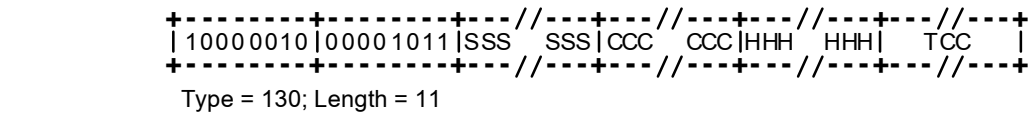


This option may be used between options, for example, to align the beginning of a subsequent option on a 32 bit boundary.

May be copied, introduced, or deleted on fragmentation, or for any other reason.

Security

This option provides a way for hosts to send security, compartmentation, handling restrictions, and TCC (closed user group) parameters. The format for this option is as follows:



Security (S field): 16 bits

Specifies one of 16 levels of security (eight of which are reserved for future use).

00000000 00000000 - Unclassified

11110001 00110101 - Confidential

01111000 10011010 - EFTO

10111100 01001101 - MMMM

01011110 00100110 - PROG

10101111 00010011 - Restricted

11010111 10001000 - Secret

01101011 11000101 - Top Secret

00110101 11100010 - (Reserved for future use)

10011010 11110001 - (Reserved for future use)

01001101 01111000 - (Reserved for future use)

00100100 10111101 - (Reserved for future use)

00010011 01011110 - (Reserved for future use)

10001001 10101111 - (Reserved for future use)

11000100 11010110 - (Reserved for future use)

11100010 01101011 - (Reserved for future use)

Compartments (C field): 16 bits

An all zero value is used when the information transmitted is not compartmented. Other values for the compartments field may be obtained from the Defense Intelligence Agency.

Handling Restrictions (H field): 16 bits

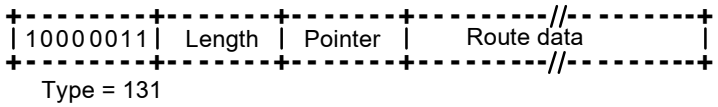
The values for the control and release markings are alphanumeric digraphs and are defined in the Defense Intelligence Agency Manual DIAM 65-19, "Standard Security Markings".

Transmission Control Code (TCC field): 24 bits

Provides a means to segregate traffic and define controlled communities of interest among subscribers. The TCC values are trigraphs, and are available from HQ DCA Code 530.

Must be copied on fragmentation. This option appears at most once in a datagram.

Loose Source and Record Route



The loose source and record route (LSRR) option provides a means for the source of an internet datagram to supply routing information to be used by the gateways in forwarding the datagram to the destination, and to record the route information.

The option begins with the option type code. The second octet is the option length which includes the option type code and the length octet, the pointer octet, and length-3 octets of route data. The third octet is the pointer into the route data indicating the octet which begins the next source address to be processed. The pointer is relative to this option, and the smallest legal value for the pointer is 4.

A route data is composed of a series of internet addresses. Each internet address is 32 bits or 4 octets. If the pointer is greater than the length, the source route is empty (and the recorded route full) and the routing is to be based on the destination address field.

If the address in destination address field has been reached and the pointer is not greater than the length, the next address in the source route replaces the address in the destination address field, and the recorded route address replaces the source address just used, and pointer is increased by four.

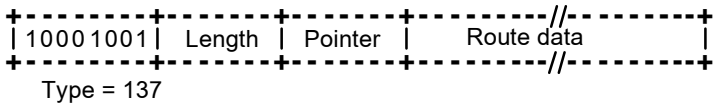
The recorded route address is the internet module’s own internet address as known in the environment into which this datagram is being forwarded.

This procedure of replacing the source route with the recorded route (though it is in the reverse of the order it must be in to be used as a source route) means the option (and the IP header as a whole) remains a constant length as the datagram progresses through the internet.

This option is a loose source route because the gateway or host IP is allowed to use any route of any number of other intermediate gateways to reach the next address in the route.

Must be copied on fragmentation. Appears at most once in a datagram.

Strict Source and Record Route



The strict source and record route (SSRR) option provides a means for the source of an internet datagram to supply routing information to be used by the gateways in forwarding the datagram to the destination, and to record the route information.

The option begins with the option type code. The second octet is the option length which includes the option type code and the length octet, the pointer octet, and length-3 octets of route data. The third octet is the pointer into the route data indicating the octet which begins the next source address to be processed. The pointer is relative to this option, and the smallest legal value for the pointer is 4.

A route data is composed of a series of internet addresses. Each internet address is 32 bits or 4 octets. If the pointer is greater than the length, the source route is empty (and the recorded route full) and the routing is to be based on the destination address field.

If the address in destination address field has been reached and the pointer is not greater than the length, the next address in the source route replaces the address in the destination address field, and the recorded route address replaces the source address just used, and pointer is increased by four.

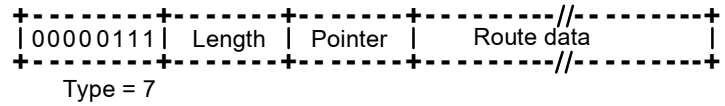
The recorded route address is the internet module's own internet address as known in the environment into which this datagram is being forwarded.

This procedure of replacing the source route with the recorded route (though it is in the reverse of the order it must be in to be used as a source route) means the option (and the IP header as a whole) remains a constant length as the datagram progresses through the internet.

This option is a strict source route because the gateway or host IP must send the datagram directly to the next address in the source route through only the directly connected network indicated in the next address to reach the next gateway or host specified in the route.

Must be copied on fragmentation. Appears at most once in a datagram.

Record Route



The record route option provides a means to record the route of an internet datagram.

The option begins with the option type code. The second octet is the option length which includes the option type code and the length octet, the pointer octet, and length-3 octets of route data. The third octet is the pointer into the route data indicating the octet which begins the next area to store a route address. The pointer is relative to this option, and the smallest legal value for the pointer is 4.

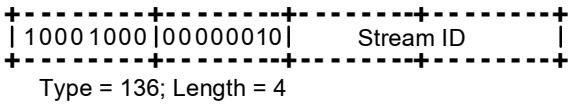
A recorded route is composed of a series of internet addresses.Each internet address is 32 bits or 4 octets. If the pointer is greater than the length, the recorded route data area is full. The originating host must compose this option with a large enough route data area to hold all the address expected. The size of the option does not change due to adding addresses. The intitial contents of the route data area must be zero.

When an internet module routes a datagram it checks to see if the record route option is present. If it is, it inserts its own internet address as known in the environment into which this datagram is being forwarded into the recorded route begining at the octet indicated by the pointer, and increments the pointer by four.

If the route data area is already full (the pointer exceeds the length) the datagram is forwarded without inserting the address into the recorded route. If there is some room but not enough room for a full address to be inserted, the original datagram is considered to be in error and is discarded. In either case an ICMP parameter problem message may be sent to the source host.

Not copied on fragmentation, goes in first fragment only. Appears at most once in a datagram.

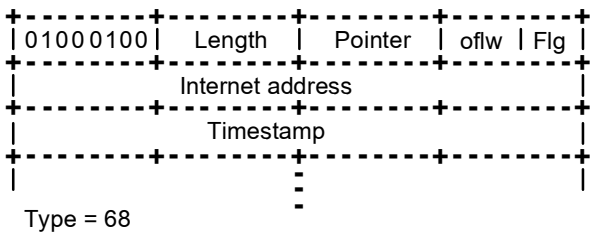
Stream Identifier



This option provides a way for the 16-bit SATNET stream identifier to be carried through networks that do not support the stream concept.

Must be copied on fragmentation. Appears at most once in a datagram.

Internet Timestamp



The Option Length is the number of octets in the option counting the type, length, pointer, and overflow/flag octets (maximum length 40).

The Pointer is the number of octets from the beginning of this option to the end of timestamps plus one (i.e., it points to the octet beginning the space for next timestamp). The smallest legal value is 5. The timestamp area is full when the pointer is greater than the length.

The Overflow (oflw) [4 bits] is the number of IP modules that cannot register timestamps due to lack of space.

The Flag (flg) [4 bits] values are

0: Time stamps only, stored in consecutive 32-bit words,

1: Each timestamp is preceded with internet address of the registering entity,

3: The internet address fields are prespecified. An IP module only registers its timestamp if it matches its own address with the next specified internet address.

The Timestamp is a right-justified, 32-bit timestamp in milliseconds since midnight UT. If the time is not available in milliseconds or cannot be provided with respect to midnight UT then any time may be inserted as a timestamp provided the high order bit of the timestamp field is set to one to indicate the use of a non-standard value.

The originating host must compose this option with a large enough timestamp data area to hold all the timestamp information expected. The size of the option does not change due to adding timestamps. The intitial contents of the timestamp data area must be zero or internet address/zero pairs.

If the timestamp data area is already full (the pointer exceeds the length) the datagram is forwarded without inserting the timestamp, but the overflow count is incremented by one.

If there is some room but not enough room for a full timestamp to be inserted, or the overflow count itself overflows, the original datagram is considered to be in error and is discarded. In either case an ICMP parameter problem message may be sent to the source host.

The timestamp option is not copied upon fragmentation. It is carried in the first fragment. Appears at most once in a datagram.

***Padding*:** variable.

The internet header padding is used to ensure that the internet header ends on a 32 bit boundary. The padding is zero.

4.2.2 Description

The implementation of a protocol must be robust. Each implementation must expect to interoperate with others created by different individuals. While the goal of this specification is to be explicit about the protocol there is the possibility of differing interpretations. In general, an implementation must be conservative in its sending behavior, and liberal in its receiving behavior. That is, it must be careful to send well-formed datagrams, but must accept any datagram that it can interpret (e.g., not object to technical errors where the meaning is still clear).

The basic internet service is datagram oriented and provides for the fragmentation of datagrams at gateways, with reassembly taking place at the destination internet protocol module in the destination host. Of course, fragmentation and reassembly of datagrams within a network or by private agreement between the gateways of a network is also allowed since this is transparent to the internet protocols and the higher-level protocols. This transparent type of fragmentation and reassembly is termed "network-dependent" (or intranet) fragmentation and is not discussed further here.

Internet addresses distinguish sources and destinations to the host level and provide a protocol field as well. It is assumed that each protocol will provide for whatever multiplexing is necessary within a host.

4.2.2.1 Addressing

To provide for flexibility in assigning address to networks and allow for the large number of small to intermediate sized networks the interpretation of the

address field is coded to specify a small number of networks with a large number of host, a moderate number of networks with a moderate number of hosts, and a large number of networks with a small number of hosts. In addition there is an escape code for extended addressing mode.

Address Formats

High Order Bits	Format	Class
0	7 bits of net, 24 bits of host	a
10	14 bits of net, 16 bits of host	b
110	21 bits of net, 8 bits of host	c
111	escape to extended addressing mode	

A value of zero in the network field means this network. This is only used in certain ICMP messages. The extended addressing mode is undefined. Both of these features are reserved for future use.

The local address, assigned by the local network, must allow for a single physical host to act as several distinct internet hosts. That is, there must be a mapping between internet host addresses and network/host interfaces that allows several internet addresses to correspond to one interface. It must also be allowed for a host to have several physical interfaces and to treat the datagrams from several of them as if they were all addressed to a single host.

4.2.2.2 Fragmentation and Reassembly

The internet identification field (ID) is used together with the source and destination address, and the protocol fields, to identify datagram fragments for reassembly.

The More Fragments flag bit (MF) is set if the datagram is not the last fragment. The Fragment Offset field identifies the fragment location, relative to the beginning of the original unfragmented datagram. Fragments are counted in units of 8 octets. The fragmentation strategy is designed so than an unfragmented datagram has all zero fragmentation information (MF = 0, fragment offset = 0). If an internet datagram is fragmented, its data portion must be broken on 8 octet boundaries.

This format allows 2^{13} = 8192 fragments of 8 octets each for a total of 65536 octets. Note that this is consistent with the the datagram total length field (of course, the header is counted in the total length and not in the fragments).

When fragmentation occurs, some options are copied, but others remain with the first fragment only.

Every internet module must be able to forward a datagram of 68 octets without further fragmentation. This is because an internet header may be up to 60 octets, and the minimum fragment is 8 octets.

Every internet destination must be able to receive a datagram of 576 octets either in one piece or in fragments to be reassembled.

The fields which may be affected by fragmentation include:

(1) Options field

(2) More fragments flag

(3) Fragment offset

(4) Internet header length field

(5) Total length field

(6) Header checksum

If the Don’t Fragment flag (DF) bit is set, then internet fragmentation of this datagram is NOT permitted, although it may be discarded. This can be used to prohibit fragmentation in cases where the receiving host does not have sufficient resources to reassemble internet fragments.

One example of use of the Don’t Fragment feature is to down line load a small host. A small host could have a boot strap program that accepts a datagram stores it in memory and then executes it.

The fragmentation and reassembly procedures are most easily described by examples. The following procedures are example implementations.

General notation in the following pseudo programs: "=<" means "less than or equal", "#" means "not equal", "=" means "equal", "<-" means "is set to". Also, "x to y" includes x and excludes y; for example, "4 to 7" would include 4, 5, and 6 (but not 7).

4.2.2.2.1 An Example Fragmentation Procedure

The maximum sized datagram that can be transmitted through the next network is called the maximum transmission unit (MTU).

If the total length is less than or equal the maximum transmission unit then submit this datagram to the next step in datagram processing; otherwise cut the

datagram into two fragments, the first fragment being the maximum size, and the second fragment being the rest of the datagram. The first fragment is submitted to the next step in datagram processing, while the second fragment is submitted to this procedure in case it is still too large.

Notation:

FO - Fragment Offset

IHL - Internet Header Length

DF - Don't Fragment flag

MF - More Fragments flag

TL - Total Length

OFO - Old Fragment Offset

OIHL - Old Internet Header Length

OMF - Old More Fragments flag

OTL - Old Total Length

NFB - Number of Fragment Blocks

MTU - Maximum Transmission Unit

Procedure:

IF TL =< MTU THEN Submit this datagram to the next step in datagram processing

ELSE IF DF = 1 THEN discard the datagram ELSE

To produce the first fragment:

(1) Copy the original internet header;

(2) OIHL <- IHL; OTL <- TL; OFO <- FO; OMF <- MF;

(3) NFB <- (MTU - IHL*4)/8;

(4) Attach the first NFB*8 data octets;

(5) Correct the header:

MF <- 1; TL <- (IHL*4) + (NFB*8);

Recompute Checksum;

(6) Submit this fragment to the next step in datagram processing;

To produce the second fragment:

(7) Selectively copy the internet header (some options are not copied, see option definitions);

(8) Append the remaining data;

(9) Correct the header:

IHL <- (((OIHL*4) - (length of options not copied)) + 3)/4;

TL <- OTL - NFB*8 - (OIHL-IHL)*4);

FO <- OFO + NFB; MF <- OMF; Recompute Checksum;

(10) Submit this fragment to the fragmentation test; DONE.

In the above procedure each fragment (except the last) was made the maximum allowable size. An alternative might produce less than the maximum size datagrams. For example, one could implement a fragmentation procedure that repeatly divided large datagrams in half until the resulting fragments were less than the maximum transmission unit size.

4.2.2.2.2 An Example Reassembly Procedure

For each datagram the buffer identifier is computed as the concatenation of the source, destination, protocol, and identification fields. If this is a whole datagram (that is both the fragment offset and the more fragments fields are zero), then any reassembly resources associated with this buffer identifier are released and the datagram is forwarded to the next step in datagram processing.

If no other fragment with this buffer identifier is on hand then reassembly resources are allocated. The reassembly resources consist of a data buffer, a header buffer, a fragment block bit table, a total data length field, and a timer. The data from the fragment is placed in the data buffer according to its fragment offset and length, and bits are set in the fragment block bit table corresponding to the fragment blocks received.

If this is the first fragment (that is the fragment offset is zero) this header is placed in the header buffer. If this is the last fragment (that is the more fragments field is zero) the total data length is computed. If this fragment completes the datagram (tested by checking the bits set in the fragment block table), then the datagram is sent to the next step in datagram processing; otherwise the timer is set to the maximum of the current timer value and the value of the time to live field from this fragment; and the reassembly routine gives up control.

If the timer runs out, the all reassembly resources for this buffer identifier are released. The initial setting of the timer is a lower bound on the reassembly waiting time. This is because the waiting time will be increased if the Time to Live in the arriving fragment is greater than the current timer value but will not be decreased if it is less. The maximum this timer value could reach is the maximum time to live (approximately 4.25 minutes). The current recommendation for the initial timer setting is 15 seconds. This may be changed as experience with this protocol accumulates. Note that the choice of this parameter value is related to the buffer capacity available and the data rate of the transmission medium; that is, data rate times timer value equals buffer size (e.g., 10 kbit/s * 15 s = 150 kbit).

Notation:

FO - Fragment Offset

IHL - Internet Header Length

MF - More Fragments flag

TTL - Time To Live

NFB - Number of Fragment Blocks

TL - Total Length

TDL - Total Data Length

BUFID - Buffer Identifier

RCVBT - Fragment Received Bit Table

TLB - Timer Lower Bound

Procedure:

(1) BUFID <- source|destination|protocol|identification;

(2) IF FO = 0 AND MF = 0

(3) THEN IF buffer with BUFID is allocated

(4) THEN flush all reassembly for this BUFID;

(5) Submit datagram to next step; DONE.

(6) ELSE IF no buffer with BUFID is allocated

(7) THEN allocate reassembly resources with BUFID;

TIMER <- TLB; TDL <- 0;

(8) put data from fragment into data buffer with BUFID from octet FO*8
to octet (TL - (IHL*4)) + FO*8;

(9) set RCVBT bits from FO to FO + ((TL - (IHL*4) + 7)/8);

(10) IF MF = 0 THEN TDL <- TL - (IHL*4) + (FO*8)

(11) IF FO = 0 THEN put header in header buffer

(12) IF TDL # 0

(13) AND all RCVBT bits from 0

to (TDL + 7)/8 are set

(14) THEN TL <- TDL + (IHL*4)

(15) Submit datagram to next step;

(16) free all reassembly resources

for this BUFID; DONE.

(17) TIMER <- MAX(TIMER,TTL);

(18) give up until next fragment or timer expires;

(19) timer expires: flush all reassembly with this BUFID; DONE.

In the case that two or more fragments contain the same data either identically or through a partial overlap, this procedure will use the more recently arrived copy in the data buffer and datagram delivered.

4.2.2.3 Identification

The choice of the Identifier for a datagram is based on the need to provide a way to uniquely identify the fragments of a particular datagram. The protocol module assembling fragments judges fragments to belong to the same datagram if they have the same source, destination, protocol, and Identifier. Thus, the sender must choose the Identifier to be unique for this source, destination pair and protocol for the time the datagram (or any fragment of it) could be alive in the internet.

It seems then that a sending protocol module needs to keep a table of Identifiers, one entry for each destination it has communicated with in the last maximum packet lifetime for the internet.

However, since the Identifier field allows 65,536 different values, some host may be able to simply use unique identifiers independent of destination.

It is appropriate for some higher level protocols to choose the identifier. For example, TCP protocol modules may retransmit an identical TCP segment, and the probability for correct reception would be enhanced if the retransmission carried the same identifier as the original transmission since fragments of either datagram could be used to construct a correct TCP segment.

4.2.2.4 Type of Service

The type of service (TOS) is for internet service quality selection. The type of service is specified along the abstract parameters precedence, delay, throughput, and reliability. These abstract parameters are to be mapped into the actual service parameters of the particular networks the datagram traverses.

Precedence. An independent measure of the importance of this datagram.

Delay. Prompt delivery is important for datagrams with this indication.

Throughput. High data rate is important for datagrams with this indication.

Reliability. A higher level of effort to ensure delivery is important for datagrams with this indication.

For example, the ARPANET has a priority bit, and a choice between "standard" messages (type 0) and "uncontrolled" messages (type 3), (the choice between single packet and multipacket messages can also be considered a service parameter). The uncontrolled messages tend to be less reliably delivered and suffer less delay. Suppose an internet datagram is to be sent through the ARPANET. Let the internet type of service be given as:

Precedence: 5

Delay: 0

Throughput: 1

Reliability: 1

In this example, the mapping of these parameters to those available for the ARPANET would be to set the ARPANET priority bit on since the Internet precedence is in the upper half of its range, to select standard messages since the throughput and reliability requirements are indicated and delay is not.

4.2.2.5 Time to Live

The time to live is set by the sender to the maximum time the datagram is allowed to be in the internet system. If the datagram is in the internet system longer than the time to live, then the datagram must be destroyed.

This field must be decreased at each point that the internet header is processed to reflect the time spent processing the datagram. Even if no local information is available on the time actually spent, the field must be decremented by 1. The time is measured in units of seconds (i.e. the value 1 means one second). Thus, the maximum time to live is 255 seconds or 4.25 minutes. Since every module that processes a datagram must decrease the TTL by at least one even if it process the datagram in less than a second, the TTL must be thought of only as an upper bound on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded, and to bound the maximum datagram lifetime.

Some higher level reliable connection protocols are based on assumptions that old duplicate datagrams will not arrive after a certain time elapses. The TTL is a way for such protocols to have an assurance that their assumption is met.

4.2.2.6 Options

The options are optional in each datagram, but required in implementations. That is, the presence or absence of an option is the choice of the sender, but each internet module must be able to parse every option. There can be several options present in the option field.

The options might not end on a 32-bit boundary. The internet header must be filled out with octets of zeros. The first of these would be interpreted as the end-of-options option, and the remainder as internet header padding.

Every internet module must be able to act on every option. The Security Option is required if classified, restricted, or compartmented traffic is to be passed.

4.2.2.7 Checksum

The internet header checksum is recomputed if the internet header is changed. For example, a reduction of the time to live, additions or changes to internet options, or due to fragmentation. This checksum at the internet level is intended to protect the internet header fields from transmission errors.

There are some applications where a few data bit errors are acceptable while retransmission delays are not. If the internet protocol enforced data correctness such applications could not be supported.

4.2.2.8 Errors

Internet protocol errors may be reported via the ICMP messages.

4.2.3 Interfaces

The functional description of user interfaces to the IP is, at best, fictional, since every operating system will have different facilities. Different IP implementations may have different user interfaces. However, all IPs must provide a certain minimum set of services to guarantee that all IP implementations can support the same protocol hierarchy. This section specifies the functional interfaces required of all IP implementations.

Internet protocol interfaces on one side to the local network and on the other side to either a higher level protocol or an application program. In the following, the higher level protocol or application program (or even a gateway program) will be called the "user" since it is using the internet module. Since internet protocol is a datagram protocol, there is minimal memory or state maintained between datagram transmissions, and each call on the internet protocol module by the user supplies all information necessary for the IP to perform the service requested.

APPENDIX A

EXAMPLE & SCENARIOS

Example 1: This is an example of the minimal data carrying internet datagram:

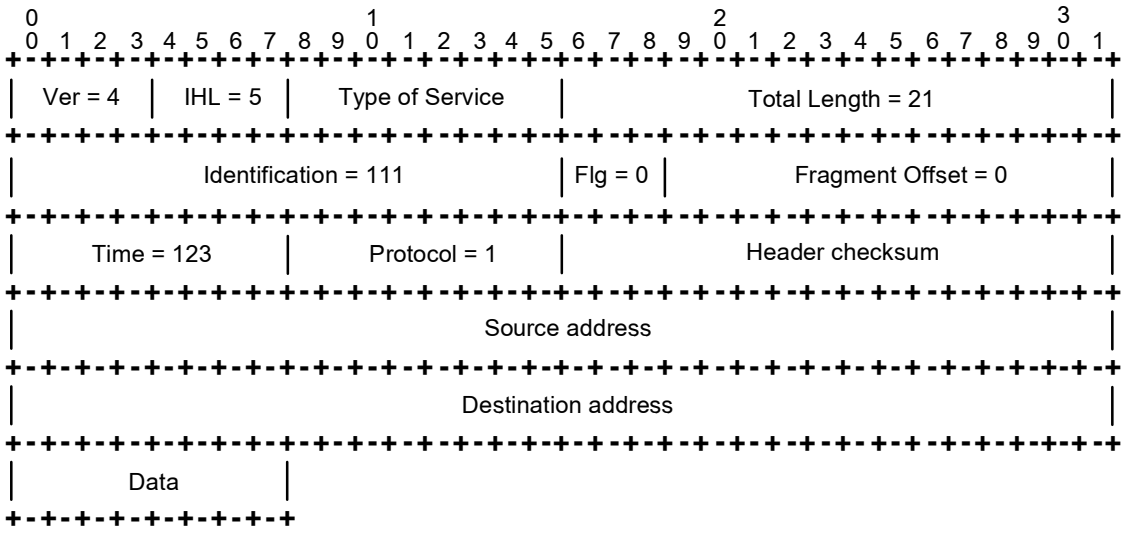


Figure 5: Example Internet Datagram

Note that each tick mark represents one bit position.

This is a internet datagram in version 4 of internet protocol; the internet header consists of five 32 bit words, and the total length of the datagram is 21 octets. This datagram is a complete datagram (not a fragment).

Example 2:

In this example, we show first a moderate size internet datagram (452 data octets), then two internet fragments that might result from the fragmentation of this datagram if the maximum sized transmission allowed were 280 octets.

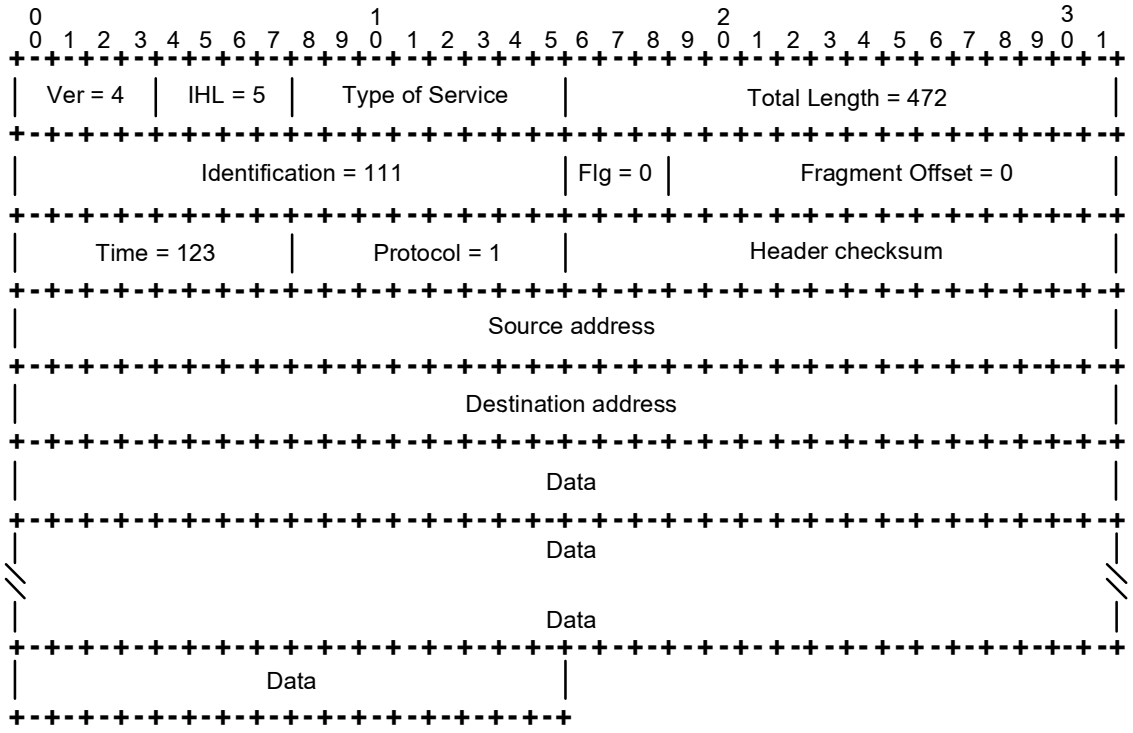


Figure 6: Example Internet Datagram

Now the first fragment that results from splitting the datagram after 256 data octets.

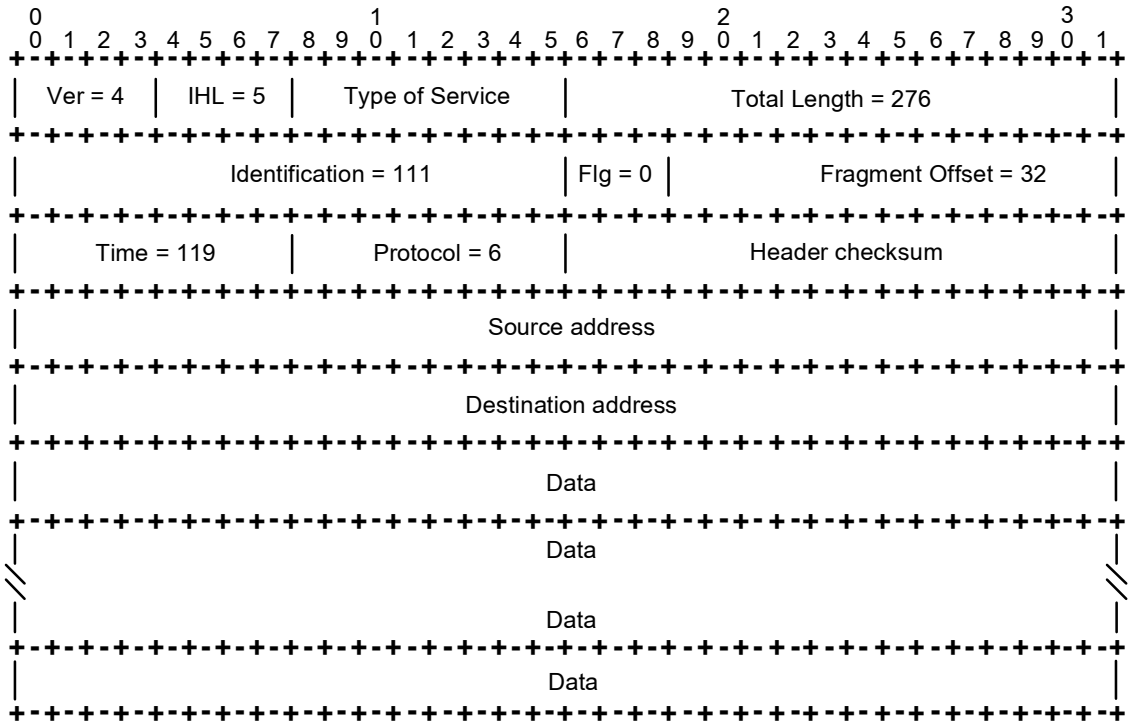


Figure 7: Example Internet Fragment

And the second fragment.

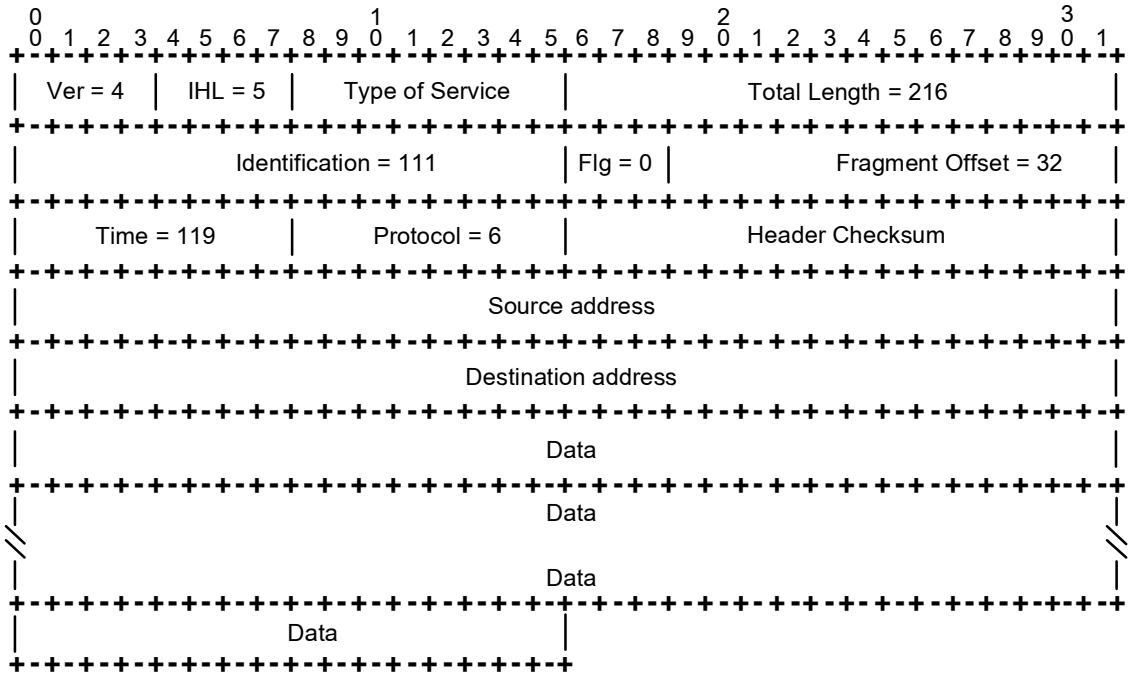


Figure 8: Example Internet Fragment

Example 3:

Here, we show an example of a datagram containing options:

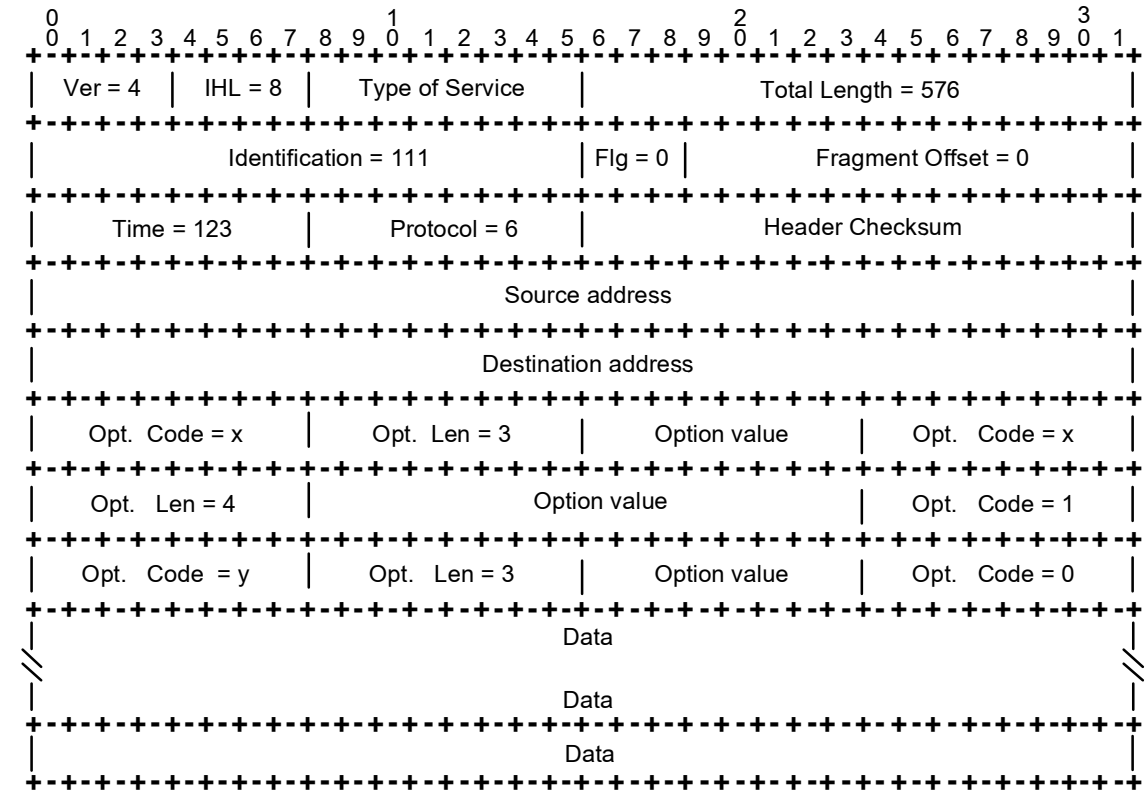


Figure 9: Example Internet Datagram

APPENDIX B
DATA TRANSMISSION ORDER

The order of transmission of the header and data described in this document is resolved to the octet level. Whenever a diagram shows a group of octets, the order of transmission of those octets is the normal order in which they are read in English. For example, in the following diagram the octets are transmitted in the order they are numbered.

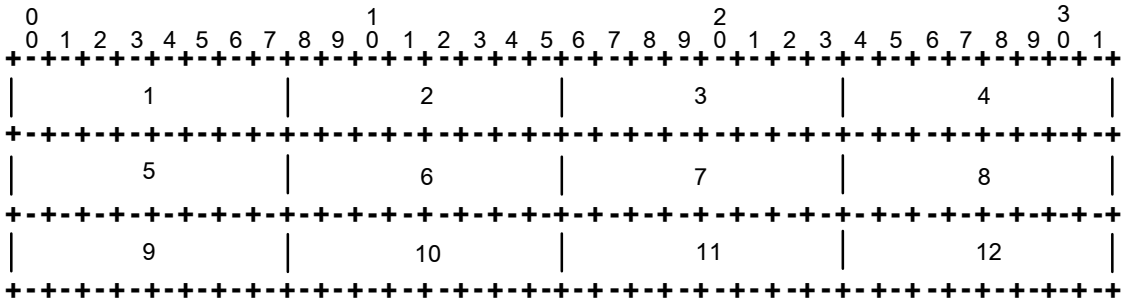


Figure 10: Transmission Order of Bytes

Whenever an octet represents a numeric quantity the left most bit in the diagram is the high order or most significant bit. That is, the bit labeled 0 is the most significant bit. For example, the following diagram represents the value 170 (decimal).

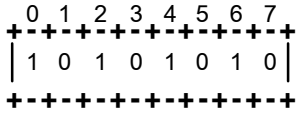


Figure 11: Significance of Bits

Similarly, whenever a multi-octet field represents a numeric quantity the left most bit of the whole field is the most significant bit. When a multi-octet quantity is transmitted the most significant octet is transmitted first.

APPENDIX C
AN EXAMPLE UPPER LEVEL INTERFACE

The following two example calls satisfy the requirements for the user to internet protocol module communication ("=>" means returns):

SEND (src, dst, prot, TOS, TTL, BufPTR, len, Id, DF, opt => result)

where:

src = source address

dst = destination address

prot = protocol

TOS = type of service

TTL = time to live

BufPTR = buffer pointer

len = length of buffer

Id = Identifier

DF = Don’t Fragment

opt = option data

result = response

OK = datagram sent ok

Error = error in arguments or local network error

Note that the precedence is included in the TOS and the security/compartment is passed as an option.

RECV (BufPTR, prot, => result, src, dst, TOS, len, opt)

where:

BufPTR = buffer pointer

prot = protocol

result = response

OK = datagram received ok

Error = error in arguments

len = length of buffer

src = source address

dst = destination address

TOS = type of service

opt = option data

When the user sends a datagram, it executes the SEND call supplying all the arguments. The internet protocol module, on receiving this call, checks the arguments and prepares and sends the message. If the arguments are good and the datagram is accepted by the local network, the call returns successfully. If either the arguments are bad, or the datagram is not accepted by the local network, the call returns unsuccessfully. On unsuccessful returns, a reasonable report must be made as to the cause of the problem, but the details of such reports are up to individual implementations.

When a datagram arrives at the internet protocol module from the local network, either there is a pending RECV call from the user addressed or there is not. In the first case, the pending call is satisfied by passing the information from the datagram to the user. In the second case, the user addressed is notified of a pending datagram. If the user addressed does not exist, an ICMP error message is returned to the sender, and the data is discarded.

The notification of a user may be via a pseudo interrupt or similar mechanism, as appropriate in the particular operating system environment of the implementation.

A user's RECV call may then either be immediately satisfied by a pending datagram, or the call may be pending until a datagram arrives.

The source address is included in the send call in case the sending host has several addresses (multiple physical connections or logical addresses). The internet module must check to see that the sourceaddress is one of the legal address for this host.

An implementation may also allow or require a call to the internet module to indicate interest in or reserve exclusive use of a class of datagrams (e.g., all those with a certain value in the protocol field).

This section functionally characterizes a USER/IP interface. The notation used is similar to most procedure of function calls in high level languages, but this usage is not meant to rule out trap type service calls (e.g., SVCs, UUOs, EMTs), or any other form of interprocess communication.