## CHƯƠNG 2: KIẾN THỨC NỀN TẢNG

## HỆ ĐIỀU HÀNH ANDROID :

**MỘT SỐ ĐẶC ĐIỂM CỦA ANDROID :**

**Tính Năng Mở của hệ điều hành Android là gì?**

Android được xây dựng  để cho phép các nhà phát triển để tạo ra các ứng dụng di động hấp dẫn tận dụng tất cả phần cứng của chiếc điện thoại nhà sản xuất cung cấp. Nó được xây dựng để được thực sự mở. Ví dụ, một ứng dụng có thể kêu gọi bất kỳ chức năng lõi của điện thoại như thực hiện cuộc gọi, gửi tin nhắn văn bản, hoặc bằng cách sử dụng máy ảnh, cho phép các nhà phát triển có thể tạo ra những ứng dụng phong phú hơn , mang đến cho người dùng nhiều trải nghiệm hơn. Android là hệ điều hành nguồn mở dựa trên Linux Kernel. Hơn nữa, nó sử dụng một máy ảo được tuỳ chỉnh thiết kế để tối ưu hóa bộ nhớ và tài nguyên phần cứng trong môi trường di động. Android là mã nguồn mở, nó có thể được tự do mở rộng. Nền tảng này sẽ tiếp tục tiến triển ,các cộng đồng nhà phát triển sẽ làm việc cùng nhau để xây dựng các ứng dụng di động ngày càng sáng tạo.

## ****Có thể tạo bất cứ ứng dụng nào cho Android :****

Android không phân biệt giữa các ứng dụng gốc của điện thoại với các ứng dụng của nhà phát triển thứ ba. Tất cả có thể được xây dựng để có thể truy cập bằng khả năng của một người cung cấp cho người sử dụng điện thoại với một dải rộng các ứng dụng và dịch vụ. Với các thiết bị xây dựng trên Hệ điều hành Android, người dùng có thể hoàn toàn thích ứng với điện thoại đến lợi ích của họ. Họ có thể trao đổi trên màn hình của điện thoại, những phong cách của dialer, hoặc bất kỳ ứng dụng. Họ thậm chí có thể hướng dẫn điện thoại của họ để sử dụng hình ảnh ưa thích của họ xem các ứng dụng để xử lý xem tất cả các hình ảnh.

## ****Android phá bỏ các rào cản của ứng dụng****

Android phá bỏ rào cản để xây dựng các ứng dụng mới và sáng tạo. Ví dụ, một nhà phát triển có thể kết hợp thông tin từ các trang web với dữ liệu trên điện thoại di động của một cá nhân – ví dụ như địa chỉ liên hệ của người dùng, lịch, hoặc vị trí địa lý - để cung cấp cho người dùng một trải nghiệm đầy đủ hơn. Với Android, một nhà phát triển có thể xây dựng một ứng dụng cho phép người dùng xem vị trí của bạn bè của họ và và có thể gửi gợi ý kết bạn cho họ nếu có những người ở gần họ sử dụng cùng phần mềm.

## ****Với Android ứng dụng được phát triển dễ dàng & nhanh chóng****

Android cung cấp truy cập đến một loạt các thư viện công cụ hữu ích và có thể được sử dụng để xây dựng các ứng dụng phong phú. Ví dụ, Android cho phép các nhà phát triển để có được vị trí của điện thoại, và cho phép các thiết bị để giao tiếp với nhau tạo điều kiện cho các ứng dụng tương tác phong phú hơn. Ngoài ra, Android bao gồm một tập hợp đầy đủ công cụ đã được xây dựng từ cấp thấp đến cấp cao ,cùng với việc cung cấp nền tảng phát triển, với năng suất cao và cái nhìn phát triển lâu dài vào các ứng dụng của họ.

**LỊCH SỬ :**

Ban đầu, Android là hệ điều hành cho các thiết bị cầm tay dựa trên nhân Linux do công ty Android Inc (California, Mỹ) thiết kế. Công ty này sau đó được Google mua lại vào năm 2005 và bắt đầu xây dựng Android Platform. Các thành viên chủ chốt tại Android Inc gồm có: Andy Rubin, Rich Miner, Nick Scars và Chris White.

Và sau tiếp, vào cuối năm 2007, thuộc về Open Handset Alliance gồm các thành viên nổi bật trong ngành viễn thông và thiết bị cầm tay như :Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, T-Mobile, ARM Holdings, Atheros Communications, Asutek Computer Inc, Garmin Ltd, Softbank, Sony Erisson,...

Mục tiêu của liên minh này là nhanh chóng đổi mới để đáp ứng tốt hơn cho nhu cầu người tiêu dùng và kết quả đầu tiên của nó chính là nền tảng Android. Android được thiết kế để phục vụ nhu cầu của các nhà sản xuất, các nhà khai thác và các lập trình viên thiết bị cầm tay.

Phiên bản SDK đầu tiên phát hành vào tháng 11 năm 2007, hãng T-Mobile cũng công bố chiếc điện thoại Android đầu tiên đó là chiếc T-Mobile G1, chiếc smartphone đầu tiên dựa trên nền tảng Android. Một vài ngày sau đó, Google lại tiếp tục công bố sự ra mắt phiên bản Android SDK realease Candidate 1.0. Trong tháng 10 năm 2008, Google được cấp phép mã nguồn mở cho Android Platform.

Khi Android được pháp hành thì một trong những mục tiêu trong kiến trúc của nó là cho phép các ứng dụng có thể tương tác được với nhau và có thể sử dụng lại các thành phần từ các ứng dụng khác. Việc tái sử dụng không chỉ được áp dụng cho các dịch vụ mà nó còn được áp dụng cho cả các thành phần dữ liệu và giao tiếp người dùng.

Vào cuối năm 2008, Google cho phát hàng một thiết bị cầm tay được gọi là Android Dev Phone 1 có thể chạy được các ứng dụng Android mà không bị ràng buộc vào các nhà chuyên cung cấp mạng điện thoại di động. Mục tiêu của thiết bị này là cho phép các nhà phát triển thực hiện các cuộc thí nghiệm trên một thiết bị thực có thể chạy hệ điều hành Android mà không phải ký một bản hợp đồng nào. Vào khoảng cùng thời gian đó thì Google cũng cho phát hành một phiên bản vá lỗi 1.1 cho hệ điều hành này. Ở cả 2 phiên bản 1.0 và 1.1 Android chưa hỗ trợ soft-keyboard mà đòi hỏi các thiết bị phải sử dụng bàn phím vật lý, Android cố định vấn đề này bằng cách phát hành SDK 1.5 vào tháng 4 năm 2009, cùng với một số chức năng khác. Chẳng hạn như nâng cao khả năng ghi âm truyền thông, vật dụng và các live folder.

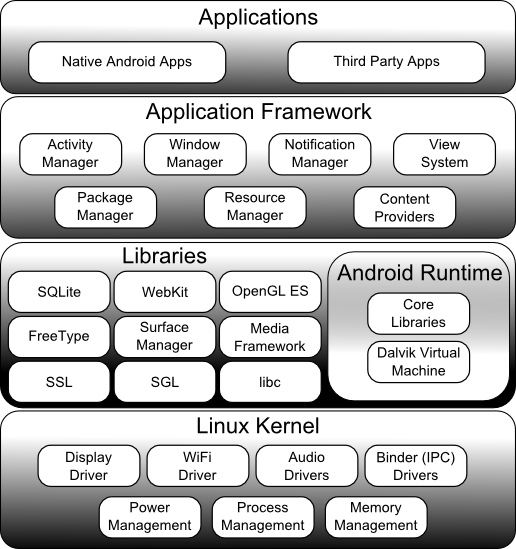
**DELVING VỚI MÁY ẢO DALVIK :**

Dalvik là máy ảo giúp các ứng dụng java chạy trên các thiết bị di động nền tảng Android. Nó chạy các ứng dụng đã được chuyển đổi thành file thực thi Dalvik (.dex). Định dạng này phù hợp cho các hệ thống mà thường bị hạn chế về bộ nhớ và tốc độ xử lý. Dalvik đã được thiết kế và viết bởi Dan Bornstein, người đã đặt tên cho nó sau khi đến thăm một ngôi làm đánh cá có tên Dalvik ở đảo Eyiafiorur, nơi mà một số tổ tiên ông sống.

Từ góc nhìn của một nhà phát triển, Dalvik trông giống như máy ảo Java (Java Virtual Machine) nhưng thực tế thì hoàn toàn khác nhau. Khi nhà phát triển viết một ứng dụng dành cho Android, anh ta thực hiện các đoạn mã trên môi trường Java. Sau đó, nó sẽ biên dịch sang các bytecode của Java, tuy nhiên để thực thi được ứng dụng này trên Android thì nhà phát triển phải thực thi một công cụ tên là DX. Đây là công cụ dùng để chuyển đổi bytecode sang một dạng gọi là dex bytecode. “Dex” là từ viết tắc của “Dalvik executable” đóng vai trò như cơ chế ảo thực thi các ứng dụng Android.

**KIẾN TRÚC CỦA ANDROID :**

Mô hình sau thể hiện một cách tổng quát các thành phân của hệ điều hành Android. Mỗi một phần sẽ được đặt tả chi tiết dưới đây.



### Tầng ứng dụng:

Android được tích hợp sẵn một số ứng dụng cần thiết, cơ bản như: Contacts, Browser, Camera, Phone,… Tất cả các ứng dụng trên đều được viết bằng Java.

### Tầng Application Framework:

Android cung cấp một nền tảng phát triển ứng dụng mở, xây dựng các API hỗ trợ người lập trình để phát triển các ứng dụng.

Người lập trình được phép sử dụng các API trong các ứng dụng tích hợp sẵn của Android. Kiến trúc ứng dụng được thiết kế nhằm mục đích đơn giản hóa việc sử dụng lại các thành phần; bất kỳ ứng dụng nào cũng có thể công bố tính năng (capability) của nó, và các ứng dụng khác có thể sử dụng tính năng này. Cơ bản, tất cả các ứng dụng là một bộ dịch vụ và các hệ thống, bao gồm:

* Một tập hợp phong phú và có thể mở rộng các View để xây dựng ứng dụng, gồm danh sách, hiển thị dạng lưới, hiển thị chữ dạng hộp, nút bấm và kể cả trình duyệt web được tích hợp.
* Trình cung cấp nội dung (Content Provider) cho phép ứng dụng truy xuất dữ liệu từ các ứng dụng khác, hay chia sẻ dữ liệu.
* Quản lý tài nguyên (Resource Manager) cung cấp quyền truy xuất đến tài nguyên không phải mã nguồn như chuỗi định vị (localized strings), các tập tin layout, hay hình ảnh.
* Quản lý thông báo (Notification Manager) cung cấp khả năng hiển thị các cảnh báo lên thanh trạng thái
* Quản lý hoạt động (Activity Manager) quản lý vòng đời của ứng dụng.

### Thư viện (Library):

Android bao gồm một tập các thư viện C/C++ được sử dụng bởi nhiều thành phần khác nhau trong hệ thống Android. Điều này được thể hiện thông qua nền tảng ứng dụng Android. Một số các thư viện cơ bản được liệt kê dưới đây:

* System C library: có nguồn gốc từ hệ thống thư viện C chuẩn (libc), được điều chỉnh để nhúng vào các thiết bị dựa trên Linux.
* Media Libraries: dựa trên PacketVideo của OpenCORE, các thư viện hỗ trợ phát và ghi các định dạng audio và video phổ biến.
* Surface Manager: Quản lý hiển thị nội dung 2D và 3D.
* LibWebCore: Công cụ cho các trình duyệt web hiện đại.
* SGL: Công cụ đồ họa 2D cơ sở.
* 3D libraries: Một thể hiện được xây dựng dựa trên OpenGL ES 1.0 APIs. Những thư viện này dùng để tối ưu hóa hiển thị 3D.
* Free Type: Biểu diễn font vec-tor và bitmap.
* QLite: Cơ sở dữ liệu.
* Webkit – Thư viện hiển thị HTML

### Android Runtime :

Android bao gồm một tập hợp các thư viện cơ bản mà cung cấp hầu hết các chức năng có sẵn trong các thư viện kernel của ngôn ngữ lập trình Java. Tất cả các ứng dụng Android đều chạy trong tiến trình riêng. Máy ảo Dalvik đã được viết lại để cho một thiết bị có thể chạy nhiều máy ảo hiệu quả. Các VM Dalvik thực thi các tập tin thực thi Dalvik (dex). Định dạng được tối ưu hóa cho bộ nhớ tối thiểu. VM là dựa trên register-based và chạy các lớp đã được biên dịch bởi một trình biện dịch Java để chuyển đổi thành các định dạng dex. Các VM Dalvik dựa vào nhân Linux cho các chức năng cơ bản như luồng và quản lý bộ nhớ cấp thấp.

### Linux Kernel :

Android dựa trên Linux phiên bản 2.6 cho hệ thống dịch vụ cốt lõi như bảo mật , quản lý bộ nhớ , quản lý các tiến trình , quản lý mạng , các driver cho các thiết bị. Kernel Linux hoạt động như một lớp trừu tượng giữa phần cứng và phần còn lại của hệ thống (Các thành phần ở trên).

**Một vài con số về Android sau 5 năm ra đời và phát triển:**

* Theo Google, hiện mỗi ngày có trung bình khoảng 1,3 triệu sản phẩm chạy Android mới được kích hoạt trên toàn cầu.
* Tính đến thời điểm hiện tại, trên toàn cầu có khoảng 570 triệu thiết bị sử dụng Android.
* Trung bình mỗi người Mỹ sở hữu 2 thiết bị chạy Android (smartphone và máy tính bảng) và gần 1/10 dân số thế giới đang sở hữu thiết bị chạy Android.
* Theo Google, số lượng thiết bị Android sẽ cán mốc 1 tỷ trước khi nền tảng di động này cán mốc 6 tuổi. Như vậy Android sẽ mất ít thời gian hơn so với Facebook để cán mốc 1 tỷ người dùng (Facebook mất gần 8 năm để đạt được điều này). Điều này thực sự ấn tượng bởi lẽ tài khoản Facebook có thể đăng ký miễn phí, trong khi muốn sở hữu Android, người dùng sẽ phải mất khoản tiền để sở hữu thiết bị.
* Mặc dù đã ra mắt khá lâu tuy nhiên Android 2.3 Gingerbread vẫn là phiên bản Android phổ biến nhất thế giới với 54% số lượng thiết bị Android. Đứng thứ 2 là Android 4.0 Ice Cream Sandwich với 25,8%. Nền tảng mới nhất Android 4.1 Jelly Bean hiện chỉ mới chiếm 2.7%.
* Hiện tại số lượng ứng dụng trên kho ứng dụng Google Play của Google dành cho nền tảng Android là 700.000 ứng dụng.

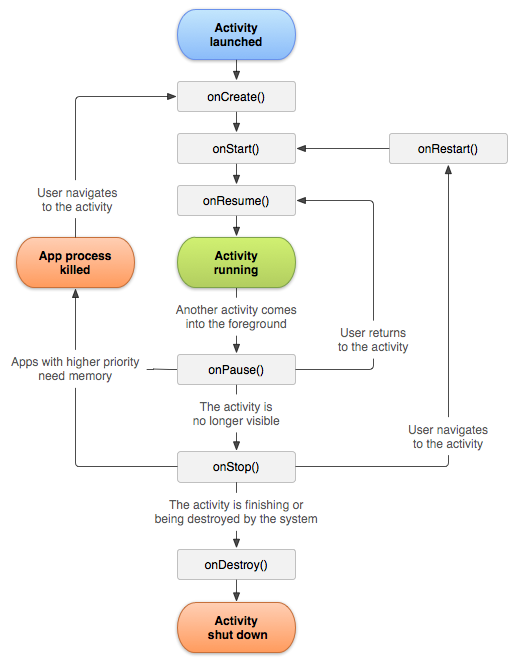
### CÁC THÀNH PHẦN CHÍNH CỦA ỨNG DỤNG ANDROID

*Activity*

Activity là thành phần của ứng dụng, hiển thị màn hình giao diện, cho phép người dùng tương tác.

Các Activity được quản lý dưới dạng Activity Stack-Last In First Out. Khi một Activity mới được khởi tạo, nó nằm trên đỉnh stack, các Activity trước đó muốn chạy trên nền trở lại cần phải chờ Activity mới này kết thúc.

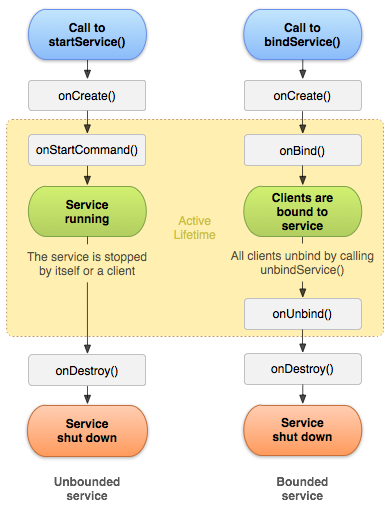
Một Activity có 4 trạng thái

* Active/Running: Khi Activity đang chạy trên màn hình.
* Paused: Khi một Activity vẫn đang chạy trên màn hình nhưng bị một Activity không chiếm toàn màn hình hiển thị phía trên.
* Stopped: Khi một Activity bị che khuất hoàn toàn bởi một Activity khác.Nó vẫn có thể được đưa trở lại để trở thành một Activity đang chạy (running) hoặc nó cũng có thể bị phá hủy và loại bỏ khỏi bộ nhớ.
* Destroyed hay Shut down: Khi một Activity đang paused hay stopped, hệ thống sẽ xóa Activity ấy ra khỏi bộ nhớ.
* 

**Service**

Service là các tác vụ chạy ngầm dưới hệ thống nhằm thực hiện một nhiệm vụ nào đó.Service được sử dụng để cập nhật dữ liệu, đưa ra các cảnh báo.

Có 2 cách để tạo Service: bằng cách gọi phương thức Context.startService() hoặc Context.bindService().



* Service được bắt đầu bởi phương thức startService()dùng để thực hiện các tác vụ ngầm dưới nền.
* Service được bắt đầu bởi phương thức bindService()dùng để cung cấp các chức năng cho các chương trình khác.

**Broadcast receiver**

Broadcast receiver là một thành phần thu nhận các Intent bên ngoài gửi tới. Có thể tạo instance cho lớp này bằng 2 cách: Context.registerReceiver() hoặc thông qua tag <receiver> trong file AndroidManifest.xml.

**Content Provider**

Content Provider được sử dụng để quản lý và chia sẻ dữ liệu giữa các ứng dụng.

Android cung cấp sẵn content providers cho 1 số kiểu dữ liệu thông dụng như âm thanh, video, danh bạ điện thoại,... Người lập trình cũng có thể tự tạo ra các class con (subclass) của Content Provider để lưu trữ kiểu dữ liệu của riêng mình.

**GOOGLE MAP API:**

Google Map một ứng dụng bản đồ tuyệt vời mà phổ biến nhất hiện nay được phát triển từ Google. Thêm một điểm cộng nữa đó là nó hoàn toàn miễn phí. Google Map hầu như có mặt trên tất cả các thiết bị di động thông minh.Và trên các ứng dụng bản đồ trên web tại Việt Nam. Chính vì vậy việc lựa chọn GoogleMap sẽ có được sự thiện cảm với người dùng khi sử dụng ứng dụng.

Google Maps API giúp dễ dàng thêm các chức năng bản đồ vào trong ứng dụng. Google Maps API là một tiện ích được thêm vào trong thư viện phụ (external library). Các thự viện viết sẵn sẽ hỗ trợ các tính năng như tải bản đồ về, lưu , vẽ và rất nhiều các tùy chọn về hiển thị và điều khiển.

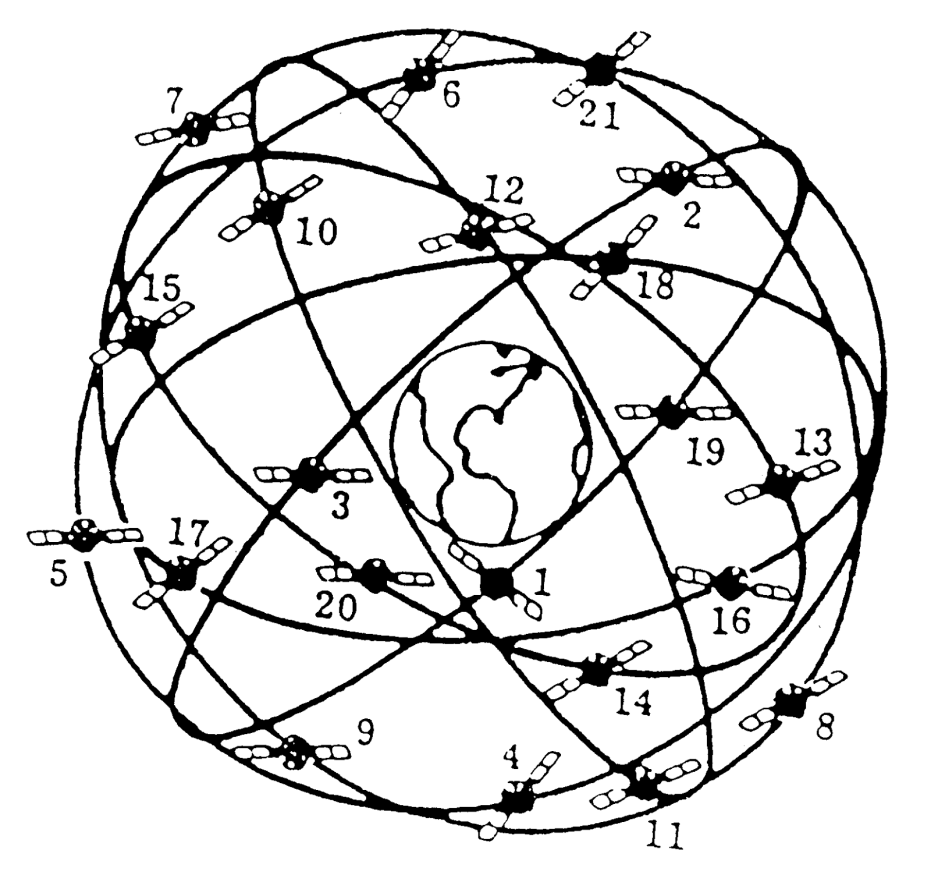
Lớp quan trọng nhất trong thư viện là lớp MapView,là subclass của lớp ViewGroup trong thư viện chuẩn của Android.MapView sẽ hiển thị bản đồ với dữ liệu nhận được từ dịch vụ Google Maps. Khi MapView được chọn , nó có thể được xoay và phóng to thu nhỏ bằng cách nhấn các phím một cách tự động , bao gồm cả việc xử lý các yêu cầu của mạng cho bản đồ.Bên cạnh đó nó cũng cung cấp tất cả các yếu tố giao diện người dùng cần thiết cho người sử dụng để kiểm soát bản đồ.Ứng dụng của bạn cũng có thể sử dụng các phương pháp điểu khiển lớp để điều khiển lớp MapView và có thể vẽ lên trên lớp MapView nhiều lớp Overlay khác. Đây là một tính năng rất quan trọng.

Nói chung , lớp MapView là lớp bao quanh Google Maps API , nó có thể giúp người lập trình điều khiển dữ liệu của Google Maps thông qua các phương thức của lớp.Giúp cho bạn làm việc với bản đồ như thể là làm việc với các View trong Android.

Thư viện bản đồ do Google Maps API cung cấp không phải là thư viện chuẩn của Android, vì thế nó có thể không hỗ trợ một số thiết bị Android. Tương tự như vậy thư viện Google Maps API cũng không bao gồm thư viện chuẩn của Android trong bộ công cụ phát triển (SDK).Các Google APIs cung cấp các tiện ích , các thư viện về bản đồ giúp cho các lập trình viên có thể phát triển , xây dựng, và chạy các ứng dụng dựa trên công cụ phát triển Android SDK với đầy đủ quyền truy cập vào dữ liệu của Google Maps.

**TỔNG QUAN VỀ GPS :**

Hệ thống Định vị Toàn cầu (tiếng Anh: Global Positioning System - GPS) là hệ thống xác định vị trí dựa trên vị trí của các vệ tinh nhân tạo, do Bộ Quốc phòng Hoa Kỳ thiết kế, xây dựng, vận hành và quản lý. Trong cùng một thời điểm, tọa độ của một điểm trên mặt đất sẽ được xác định nếu xác định được khoảng cách từ điểm đó đến ít nhất ba vệ tinh.



Tuy được quản lý bởi Bộ Quốc phòng Hoa Kỳ, chính phủ Hoa Kỳ cho phép mọi người trên thế giới sử dụng một số chức năng của GPS miễn phí, bất kể quốc tịch nào.

Các nước trong Liên minh châu Âu đang xây dựng Hệ thống định vị Galileo, có tính năng giống như GPS của Hoa Kỳ, dự tính sẽ bắt đầu hoạt động năm 2014.

**Phân loại**

Hệ thống định vị toàn cầu của Mỹ là hệ dẫn đường dựa trên một mạng lưới 24 quả vệ tinh được Bộ Quốc phòng Hoa Kỳ đặt trên quỹ đạo không gian.

Các hệ thống dẫn đường truyền thống hoạt động dựa trên các trạm phát tín hiệu vô tuyến điện. Được biết đến nhiều nhất là các hệ thống sau: LORAN – (LOng RAnge Navigation) – hoạt động ở giải tần 90-100 kHz chủ yếu dùng cho hàng hải, hay TACAN – (TACtical Air Navigation) – dùng cho quân đội Mỹ và biến thể với độ chính xác thấp VOR/DME – VHF (Omnidirectional Range/Distance Measuring Equipment) – dùng cho hàng không dân dụng.

Gần như đồng thời với lúc Mỹ phát triển GPS, Liên Xô cũng phát triển một hệ thống tương tự với tên gọi GLONASS. Hiện nay Liên minh Châu Âu đang phát triển hệ dẫn đường vệ tinh của mình mang tên Galileo. Trung Quốc thì phát triển hệ thống định vị toàn cầu của mình mang tên Bắc Đẩu bao gồm 35 vệ tinh.

Ban đầu, GPS và GLONASS đều được phát triển cho mục đích quân sự, nên mặc dù chúng dùng được cho dân sự nhưng không hệ nào đưa ra sự đảm bảo tồn tại liên tục và độ chính xác. Vì thế chúng không thỏa mãn được những yêu cầu an toàn cho dẫn đường dân sự hàng không và hàng hải, đặc biệt là tại những vùng và tại những thời điểm có hoạt động quân sự của những quốc gia sở hữu các hệ thống đó. Chỉ có hệ thống dẫn đường vệ tinh châu Âu Galileo (đang được xây dựng) ngay từ đầu đã đặt mục tiêu đáp ứng các yêu cầu nghiêm ngặt của dẫn đường và định vị dân sự.

GPS ban đầu chỉ dành cho các mục đích quân sự, nhưng từ năm 1980 chính phủ Mỹ cho phép sử dụng trong dân sự. GPS hoạt động trong mọi điều kiện thời tiết, mọi nơi trên Trái Đất, 24 giờ một ngày. Không mất phí thuê bao hoặc mất tiền trả cho việc thiết lập sử dụng GPS nhưng phải tốn tiền không rẻ để mua thiết bị thu tín hiệu và phần mềm nhúng hỗ trợ.

**Sự hoạt động của GPS**

Các vệ tinh GPS bay vòng quanh Trái Đất hai lần trong một ngày theo một quỹ đạo rất chính xác và phát tín hiệu có thông tin xuống Trái Đất. Các máy thu GPS nhận thông tin này và bằng phép tính lượng giác tính được chính xác vị trí của người dùng. Về bản chất máy thu GPS so sánh thời gian tín hiệu được phát đi từ vệ tinh với thời gian nhận được chúng. Sai lệch về thời gian cho biết máy thu GPS ở cách vệ tinh bao xa. Rồi với nhiều quãng cách đo được tới nhiều vệ tinh máy thu có thể tính được vị trí của người dùng và hiển thị lên bản đồ điện tử của máy.

Máy thu phải nhận được tín hiệu của ít nhất ba vệ tinh để tính ra vị trí hai chiều (kinh độ và vĩ độ) và để theo dõi được chuyển động. Khi nhận được tín hiệu của ít nhất 4 vệ tinh thì máy thu có thể tính được vị trí ba chiều (kinh độ, vĩ độ và độ cao). Một khi vị trí người dùng đã tính được thì máy thu GPS có thể tính các thông tin khác, như tốc độ, hướng chuyển động, bám sát di chuyển, khoảng hành trình, quãng cách tới điểm đến, thời gian Mặt Trời mọc, lặn và nhiều thứ khác nữa.

**Độ chính xác của GPS**

Các máy thu GPS ngày nay cực kì chính xác, nhờ vào thiết kế nhiều kênh hoạt động song song của chúng. Các máy thu 12 kênh song song (của Garmin) nhanh chóng khóa vào các quả vệ tinh khi mới bật lên và chúng duy trì kết nối bền vững, thậm chí trong tán lá rậm rạp hoặc thành phố với các toà nhà cao tầng. Trạng thái của khí quyển và các nguồn gây sai số khác có thể ảnh hưởng tới độ chính xác của máy thu GPS. Các máy thu GPS có độ chính xác trung bình trong vòng 15 mét.

Các máy thu mới hơn với khả năng WAAS (Wide Area Augmentation System) có thể tăng độ chính xác trung bình tới dưới 3 mét. Không cần thêm thiết bị hay mất phí để có được lợi điểm của WAAS. Người dùng cũng có thể có độ chính xác tốt hơn với GPS vi sai (Differential GPS, DGPS) sửa lỗi các tín hiệu GPS để có độ chính xác trong khoảng 3 đến 5 mét. Cục Phòng vệ Bờ biển Mỹ vận hành dịch vụ sửa lỗi này. Hệ thống bao gồm một mạng các đài thu tín hiệu GPS và phát tín hiệu đã sửa lỗi bằng các máy phát hiệu. Để thu được tín hiệu đã sửa lỗi, người dùng phải có máy thu tín hiệu vi sai bao gồm cả ăn-ten để dùng với máy thu GPS của họ.

**Các thành phần của GPS**

GPS hiện tại gồm 3 phần chính: phần không gian, kiểm soát và sử dụng.Không quân Hoa Kỳ phát triển, bảo trì và vận hành các phần không gian và kiểm soát. Các vệ tinh GPS truyền tín hiệu từ không gian, và các máy thu GPS sử dụng các tín hiệu này để tính toán vị trí trong không gian 3 chiều (kinh độ, vĩ độ và độ cao) và thời gian hiện tại.

**Phần không gian**

Phần không gian gồm 24 vệ tinh (21 vệ tinh hoạt động và 3 vệ tinh dự phòng) nằm trên các quỹ đạo xoay quanh trái đất. Chúng cách mặt đất 20.200 km, bán kính quỹ đạo 26.600 km. Chúng chuyển động ổn định vá quay hai vòng quỹ đạo trong khoảng thời gian gần 24 giờ với vận tốc 7 nghìn dặm một giờ. Các vệ tinh trên quỹ đạo được bố trí sao cho các máy thu GPS trên mặt đất có thể nhìn thấy tối thiểu 4 vệ tinh vào bất kỳ thời điểm nào.

Các vệ tinh được cung cấp bằng năng lượng Mặt Trời. Chúng có các nguồn pin dự phòng để duy trì hoạt động khi chạy khuất vào vùng không có ánh sáng Mặt Trời. Các tên lửa nhỏ gắn ở mỗi quả vệ tinh giữ chúng bay đúng quỹ đạo đã định.

**Phần kiểm soát**

Mục đích trong phần này là kiểm soát vệ tinh đi đúng hướng theo quỹ đạo và thông tin thời gian chính xác. Có 5 trạm kiểm soát đặt rải rác trên trái đất. Bốn trạm kiểm soát hoạt động một cách tự động, và một trạm kiểm soát là trung tâm. Bốn trạm này nhận tín hiệu liên tục từ những vệ tinh và gửi các thông tin này đến trạm kiểm soát trung tâm. Tại trạm kiểm soát trung tâm, nó sẽ sửa lại dữ liệu cho đúng và kết hợp với hai an-ten khác để gửi lại thông tin cho các vệ tinh. Ngoài ra, còn một trạm kiểm soát trung tâm dự phòng và sáu trạm quan sát chuyên biệt.

Trạm trung tâm cũng có thể truy cập từ các ăng-ten mặt đất của U.S. Air Force Satellite Control Network (AFSCN) và các trạm quan sát NGA (National Geospatial-Intelligence Agency). Các đường bay của vệ tinh được ghi nhận bởi các trạm quan sát chuyên dụng của Không quân Hoa Kỳ đặt ở Hawaii, Kwajalein, Đảo Ascension, Diego Garcia, Colorado Springs, Colorado và Cape Canaveral, cùng với các trạm quan sát NGA được vận hành ở Anh, Argentina, Ecuador, Bahrain, Úc và Washington DC. Thông tin đường bay của vệ tinh đi được gởi đến Air Force Space Command's MCS ở Schriever Air Force Base 25 km đông đông nam của Colorado Springs, do 2nd Space Operations Squadron (2 SOPS) của U.S. Air Force vận hành. Sau đó 2 SOPS liên lạc thường xuyên với mỗi vệ tinh GPS thông qua việc cập nhật định vị sử dụng các ăng-ten mặt đất chuyên dụng hoặc dùng chung (AFSCN)(các ăng-ten GPS mặt đất chuyên dụng được đặt ở Kwajalein, đảo Ascension, Diego Garcia, và Cape Canaveral). Các thông tin cập nhật này đồng bộ hóa với các đồng hồ nguyên tử đặt trên vệ tinh trong vòng một vài phần tỉ giây cho mỗi vệ tinh, và hiệu chỉnh lịch thiên văn của mô hình quỹ đạo bên trong mỗi vệ tinh. Việc cập nhật được tạo ra bở bộ lọc Kalman sử dụng các tín hiệu/thông tin từ các trạm quan sát trên mặt đất, thông tin thời tiết không gian, và các dữ liệu khác.

**Phần sử dụng**

Phần sử dụng là thiết bị nhận tín hiệu vệ tinh GPS và người sử dụng thiết bị này.

Dưới đây là một số thông tin đáng chú ý về các vệ tinh GPS (còn gọi là NAVSTAR, tên gọi chính thức của Bộ Quốc phòng Mỹ cho GPS):

Vệ tinh GPS đầu tiên được phóng năm 1978.

Hoàn chỉnh đầy đủ 24 vệ tinh vào năm 1994.

Mỗi vệ tinh được làm để hoạt động tối đa là 10 năm.

Vệ tinh GPS có trọng lượng khoảng 1500 kg và dài khoảng 17 feet (5 m) với các tấm năng lượng Mặt Trời mở (có độ rộng 7 m²).

Công suất phát bằng hoặc dưới 50 watts.

Tín hiệu GPS

Các vệ tinh GPS phát hai tín hiệu vô tuyến công suất thấp dải L1 và L2. (dải L là phần sóng cực ngắn của phổ điện từ trải rộng từ 0,39 tới 1,55 GHz). GPS dân sự dùng tần số L1 1575.42 MHz trong dải UHF. Tín hiệu truyền trực thị, có nghĩa là chúng sẽ xuyên qua mây, thuỷ tinh và nhựa nhưng không qua phần lớn các đối tượng cứng như núi và nhà.

L1 chứa hai mã "giả ngẫu nhiên"(pseudo random), đó là mã Protected (P) và mã Coarse/Acquisition (C/A). Mỗi một vệ tinh có một mã truyền dẫn nhất định, cho phép máy thu GPS nhận dạng được tín hiệu. Mục đích của các mã tín hiệu này là để tính toán khoảng cách từ vệ tinh đến máy thu GPS.

Tín hiệu GPS chứa ba mẩu thông tin khác nhau – mã giả ngẫu nhiên, dữ liệu thiên văn và dữ liệu lịch. Mã giả ngẫu nhiên đơn giản chỉ là mã định danh để xác định được quả vệ tinh nào là phát thông tin nào. Có thể nhìn số hiệu của các quả vệ tinh trên trang vệ tinh của máy thu Garmin để biết nó nhận được tín hiệu của quả nào.

Dữ liệu thiên văn cho máy thu GPS biết quả vệ tinh ở đâu trên quỹ đạo ở mỗi thời điểm trong ngày. Mỗi quả vệ tinh phát dữ liệu thiên văn chỉ ra thông tin quỹ đạo cho vệ tinh đó và mỗi vệ tinh khác trong hệ thống.

Dữ liệu lịch được phát đều đặn bởi mỗi quả vệ tinh, chứa thông tin quan trọng về trạng thái của vệ tinh (lành mạnh hay không), ngày giờ hiện tại. Phần này của tín hiệu là cốt lõi để phát hiện ra vị trí.

**Nguồn lỗi của tín hiệu GPS**

Những yếu tố có thể làm giảm tín hiệu GPS và vì thế ảnh hưởng tới chính xác bao gồm:

Giữ chậm của tầng đối lưu và tầng ion – Tín hiệu vệ tinh bị chậm đi khi xuyên qua tầng khí quyển.

Tín hiệu đi nhiều đường – Điều này xảy ra khi tín hiệu phản xạ từ nhà hay các đối tượng khác trước khi tới máy thu.

Lỗi đồng hồ máy thu – Đồng hồ có trong máy thu không chính xác như đồng hồ nguyên tử trên các vệ tinh GPS.

Lỗi quỹ đạo – Cũng được biết như lỗi thiên văn, do vệ tinh thông báo vị trí không chính xác.

Số lượng vệ tinh nhìn thấy – Càng nhiều quả vệ tinh được máy thu GPS nhìn thấy thì càng chính xác. Nhà cao tầng, địa hình, nhiễu loạn điện tử hoặc đôi khi thậm chí tán lá dầy có thể chặn thu nhận tín hiệu, gây lỗi định vị hoặc không định vị được. Nói chung máy thu GPS không làm việc trong nhà, dưới nước hoặc dưới đất.

Che khuất về hình học – Điều này liên quan tới vị trí tương đối của các vệ tinh ở thời điểm bất kì. Phân bố vệ tinh lí tưởng là khi các quả vệ tinh ở vị trí tạo các góc rộng với nhau. Phân bố xấu xảy ra khi các quả vệ tinh ở trên một đường thẳng hoặc cụm thành nhóm.

Sự giảm có chủ tâm tín hiệu vệ tinh – Là sự làm giảm tín hiệu cố ý do sự áp đặt của Bộ Quốc phòng Mỹ, nhằm chống lại việc đối thủ quân sự dùng tín hiệu GPS chính xác cao. Chính phủ Mỹ đã ngừng việc này từ tháng 5 năm 2000, làm tăng đáng kể độ chính xác của máy thu GPS dân sự. (Tuy nhiên biện pháp này hoàn toàn có thể được sử dụng lại trong những điều kiện cụ thể để đảm bảo gậy ông không đập lưng ông. Chính điều này là tiềm ẩn hạn chế an toàn cho dẫn đường và định vị dân sự.)

**Ứng dụng GPS**

GPS chủ yếu được sử dụng trong các ứng dụng dân dụng

*Quản lý và điều hành xe :* Giám sát quản lý vận tải, theo dõi vị trí, tốc độ, hướng di chuyển,… Giám sát mại vụ, giám sát vận tải hành khách,.. Chống trộm cho ứng dụng thuê xe tự lái, theo dõi lộ trình của đoàn xe Liên lạc, theo dõi định vị cho các ứng dụng giao hàng GPS có nhiều ứng dụng mạnh mẽ trong quản lý xe ô tô, đặc biệt là các loại xe như: Xe taxi, xe tải, xe công trình, xe bus, xe khách, xe tự lái. Với nhiều tính năng như:

Giám sát lộ trình đường đi của phương tiện theo thời gian thực: vận tốc, hướng di chuyển và trạng thái tắt/mở máy, quá tốc độ của xe….

Xác định vị trí xe chính xác ở từng góc đường ( vị trí xe được thể hiện nháp nháy trên bản đồ), xác định vận tốc và thời gian xe dừng hay đang chạy, biết được lộ trình hiện tại xe đang đi (real time)

Lưu trữ lộ trình từng xe và hiển thị lại lộ trình của từng xe trên cùng một màn hình

Xem lại lộ trình xe theo thời gian và vận tốc tùy chọn

Quản lý theo dõi một hay nhiều xe tại mỗi thời điểm

Báo cáo cước phí và tổng số km của từng xe (ngày/tháng)

Cảnh báo khi xe vượt quá tốc độ, vượt ra khỏi vùng giới hạn

Chức năng chống trộm

Khảo sát trắc địa, môi trường

Các hạn chế trong ứng dụng dân dụng

Chính phủ Hoa Kỳ kiểm soát vệc xuất khẩu một số máy thu dân dụng. Tất cả máy thu GPS có khả năng hoạt động ở độ cao trên 18 kilômét (11 mi) và 515 mét trên giây (1.690 ft/s) được phân loại vào nhóm vũ khí theo đó cần phải có phép sử dụng của Bộ ngoại giao Hoa Kỳ. Những hạn chế này nhắm mục đích ngăn ngừa việc sử dụng các máy thu trong tên lửa đạn đạo, trừ việc sử dụng trong tên lửa hành trình do độ cao và tốc độ của các loại này tương tự như các máy bay.

HỆ CƠ SỞ DỮ LIỆU SQLite :

Nói một cách đơn giản SQLite là phần mềm quản lý cơ sở dữ liệu (DBMS) tương tự như Mysql, PostgreSQL... Đặc điểm của SQLite là gọn, nhẹ, đơn giản. Chương trình gồm một file duy nhất vỏn vẹn chưa đến 500kB, không cần cài đặt, không cần cấu hình hay khởi động mà có thể sử dụng ngay. Dữ liệu database cũng được lưu ở một file duy nhất. Không có khái niệm user, password hay quyền hạn trong SQLite database.SQLite không thích hợp với những hệ thống lớn nhưng ở quy mô vừa tầm thì SQLite phát huy uy lực và không hề yếu kém về mặt chức năng hay tốc độ.

Mỗi ứng dụng đều sử dụng dữ liệu, dữ liệu có thể đơn giản hay đôi khi là cả 1 cấu trúc. Trong Android thì hệ cơ sở dữ liệu được sử dụng là SQLite Database, đây là hệ thống mã nguồn mở được sử dụng rộng rãi trong các ứng dụng và các hệ điều hành di động để lưu trữ thông tin vì tính hiệu quả của nó.Trong Android , cơ sở dữ liệu mà bạn tạo cho một ứng dụng thì chỉ ứng dụng đó có quyền truy cập và sử dụng, các ứng dụng khác thì không. Điều này có ý nghĩa quan trọng trong việc bảo mật thông tin cho các ứng dụng.Nó cũng giúp phân chia cơ sở dữ liệu của các ứng dụng khác nhau ra riêng biệt.Tránh được những nhập nhằng khi truy xuất dữ liệu.

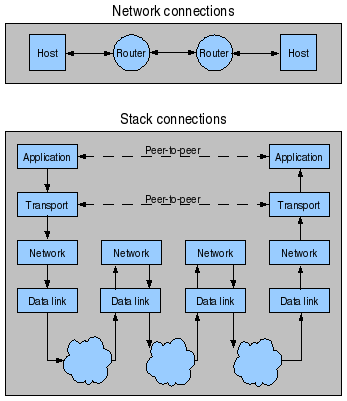
Một vài nét chính :

* Cơ sở dữ liệu dạng quan hệ (RDBMS), hỗ trợ chuẩn SQL-92
* Sử dụng dưới dạng thư viện nhúng, không chạy ở theo kiểu server độc lập
* Hỗ trợ các ngôn ngữ phổ biến: C, C++, C#, Basic, Perl, Ruby, Python, PHP, Java ...
* Không cần chỉ định kiểu dữ liệu (SQLite is typeless)
* Hỗ trợ mã UTF8
* Hỗ trợ command line
* Hỗ trợ transaction
* Hỗ trợ view
* Hỗ trợ C extensions
* Có các công cụ quản lý bằng đồ hoạ (xem chi tiết ở dưới)
* Download & sử dụng miễn phí

TCP/IP

Bộ giao thức TCP/IP, ngắn gọn là TCP/IP (tiếng Anh: Internet protocol suite hoặc IP suite hoặc TCP/IP protocol suite - bộ giao thức liên mạng), là một bộ các giao thức truyền thông cài đặt chồng giao thức mà Internet và hầu hết các mạng máy tính thương mại đang chạy trên đó. Bộ giao thức này được đặt tên theo hai giao thức chính của nó là TCP (Giao thức Điều khiển Giao vận) và IP (Giao thức Liên mạng). Chúng cũng là hai giao thức đầu tiên được định nghĩa.

Mô hình OSI miêu tả một tập cố định gồm 7 tầng mà một số nhà sản xuất lựa chọn và nó có thể được so sánh tương đối với bộ giao thức TCP/IP. Sự so sánh này có thể gây nhầm lẫn hoặc mang lại sự hiểu biết sâu hơn về bộ giao thức TCP/IP.



**Các tầng**

Sau đây là miêu tả từng tầng trong bộ giao thức TCP/IP.

**Tầng ứng dụng :**

Tầng ứng dụng là nơi các chương trình mạng thường dùng nhất làm việc nhằm liên lạc giữa các nút trong một mạng.

Giao tiếp xảy ra trong tầng này là tùy theo các ứng dụng cụ thể và dữ liệu được truyền từ chương trình, trong định dạng được sử dụng nội bộ bởi ứng dụng này, và được đóng gói theo một giao thức tầng giao vận.

Hai giao thức tầng thấp thông dụng nhất là TCP và UDP. Mỗi ứng dụng sử dụng dịch vụ của một trong hai giao thức trên đều cần có cổng. Hầu hết các ứng dụng thông dụng có các cổng đặc biệt được cấp sẵn cho các chương trình phục vụ (server)(HTTP - Giao thức truyền siêu văn bản dùng cổng 80; FTP - Giao thức truyền tệp dùng cổng 21, v.v..) trong khi các trình khách (client) sử dụng các cổng tạm thời (ephemeral port).

**Tầng giao vận :**

Trách nhiệm của tầng giao vận là kết hợp các khả năng truyền thông điệp trực tiếp (end-to-end) không phụ thuộc vào mạng bên dưới, kèm theo kiểm soát lỗi (error control), phân mảnh (fragmentation) và điều khiển lưu lượng. Việc truyền thông điệp trực tiếp hay kết nối các ứng dụng tại tầng giao vận có thể được phân loại như sau:

1. định hướng kết nối (connection-oriented), ví dụ TCP

2. phi kết nối (connectionless), ví dụ UDP

Tầng giao vận có thể được xem như một cơ chế vận chuyển thông thường, nghĩa là trách nhiệm của một phương tiện vận tải là đảm bảo rằng hàng hóa/hành khách của nó đến đích an toàn và đầy đủ.

Tầng giao vận cung cấp dịch vụ kết nối các ứng dụng với nhau thông qua việc sử dụng các cổng TCP và UDP. Do IP chỉ cung cấp dịch vụ phát chuyển nỗ lực tối đa (best effort delivery), tầng giao vận là tầng đâu tiên giải quyết vấn đề độ tin cậy.

Giao thức mới hơn, SCTP (Stream Control Transmission Protocol|), cũng là một cơ chế giao vận định hướng kết nối "đáng tin cậy". Giao thức này định hướng dòng (stream-oriented), chứ không định hướng byte như TCP, và cung cấp nhiều dòng đa công (multiplexed) trên một kết nối. Nó còn hỗ trợ multi-homed, trong đó một đầu của kết nối có thể được đại diện bởi nhiều địa chỉ IP (đại diện cho nhiều giao diện vật lý), sao cho, nếu một giao diện vật lý thất bại thì kết nối vẫn không bị gián đoạn. Giao thức này ban đầu được phát triển dành cho các ứng dụng điện thoại (để vận chuyển SS7 trên giao thức IP), nhưng nó cũng có thể được sử dụng cho các ứng dụng khác.

**Tầng mạng**

Theo định nghĩa ban đầu, tầng mạng giải quyết các vấn đề dẫn các gói tin qua một mạng đơn. Một số ví dụ về các giao thức như vậy là X.25, và giao thức Host/IMP của mạng ARPANET.

Với sự xuất hiện của khái niệm liên mạng, các chức năng mới đã được bổ sung cho tầng này, đó là chức năng dẫn đường cho dữ liệu từ mạng nguồn đến mạng đích. Nhiệm vụ này thường đòi hỏi việc định tuyến cho gói tin quan một mạng lưới của các mạng máy tính, đó là liên mạng.

Trong bộ giao thức liên mạng, giao thức IP thực hiện nhiệm vụ cơ bản dẫn đường dữ liệu từ nguồn tới đích. IP có thể chuyển dữ liệu theo yêu cầu của nhiều giao thức tầng trên khác nhau; mỗi giao thức trong đó được định danh bởi một số hiệu giao thức duy nhất: giao thức ICMP (Internet Control Message Protocol) là giao thức 1 và giao thức IGMP (Internet Group Management Protocol) là giao thức 2.

Một số giao thức truyền bởi IP, chẳng hạn ICMP (dùng để gửi thông tin chẩn đoán về truyền dữ liệu bằng IP) và IGMP (dùng để quản lý dữ liệu đa truyền (multicast)), được đặt lên trên IP nhưng thực hiện các chức năng của tầng liên mạng, điều này minh họa một sự bất tương thích giữa liên mạng và chồng TCP/IP và mô hình OSI. Tất cả các giao thức định tuyến, chẳng hạn giao thức BGP (Border Gateway Protocol), giao thức OSPF, và giao thức RIP (Routing information protocol|), đều thực sự là một phần của tầng mạng, mặc dù chúng có thể có vẻ thuộc về phần trên của chồng giao thức.

**Tầng liên kết**

Tầng liên kết - phương pháp được sử dụng để chuyển các gói tin từ tầng mạng tới các máy chủ (host) khác nhau - không hẳn là một phần của bộ giao thức TCP/IP, vì giao thức IP có thể chạy trên nhiều tầng liên kết khác nhau. Các quá trình truyền các gói tin trên một liên kết cho trước và nhận các gói tin từ một liên kết cho trước có thể được điều khiển cả trong phần mềm điều vận thiết bị (device driver) dành cho cạc mạng, cũng như trong phần sụn (firmware) hay các chipset chuyên dụng. Những thứ đó sẽ thực hiện các chức năng liên kết dữ liệu chẳng hạn như bổ sung một tín đầu (packet header) để chuẩn bị cho việc truyền gói tin đó, rồi thực sự truyền frame dữ liệu qua một môi trường vật lý.

Tầng liên kết còn có thể được xem là bao gồm cả tầng vật lý - tầng là kết hợp của các thành phần mạng vật lý thực sự (hub, các bộ lặp (repeater), cáp mạng, cáp quang, cáp đồng trục (coaxial cable), cạc mạng, cạc HBA (Host Bus Adapter) và các thiết bị nối mạng có liên quan: RJ-45, BNC, etc), và các đặc tả mức thấp về các tín hiệu (mức hiệu điện thế, tần số, v.v..).

ƯU NHƯỢC ĐIỂM VÀ BẢO MẬT

ƯU ĐIỂM

* Là phương thức truyền thông tin cậy, đảm bảo cho các gói tin đến đúng thứ tự như đáp ứng yêu cầu của ứng dung.
* Được dùng để thiết lập kết nối giữ nhiều loại thiết bị khác nhau theo mô hình server-client. Đồng thời là khả năng mở rộng của hệ thống.
* Là một chuẩn giao thức công nghiệp mở, không bị quản lý bởi một tổ chức công nghiệp nào.
* Gồm nhiều giao thức định tuyến cho phép truyền dữ liệu thông qua các mạng khác nhau.
* Hoạt động độc lập với hệ điều hành.
* Cung cấp rất nhiều dịch vụ cần thiết như phân giải tên miền, cấu hình địa chỉ máy chủ linh hoạt…

NHƯỢC ĐIỂM

* Chi phí để duy kết nối TCP/IP là rất cao.
* TCP/IP khá phức tạp để có thể thiết lập và quản lý.

## GPX

GPX là định dạng dữ liệu XML cho việc trao đổi dữ liệu GPS giữa các ứng dụng và dịch vụ Web trên Internet.

GPX là một định dạng mở, và có thể được sử dụng miễn phí.

|  |
| --- |
| <?xml version='1.0' encoding='UTF-8' standalone='yes' ?>  <gpx version="1.1" creator="Traffic Direction" xsi:schemaLocation="http://www.topografix.com/GPX/1/1 http://www.topografix.com/gpx/1/1/gpx.xsd" xmlns="http://www.topografix.com/GPX/1/1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gpx10="http://www.topografix.com/GPX/1/0" xmlns:ogt10="http://gpstracker.android.sogeti.nl/GPX/1/0">  <metadata>  <time>2012-12-09T11:15:00+0700</time>  </metadata>  <trk>  <name>Track2012-12-9 11-15</name>  <trkpt lat="10.753101666666664" lon="106.66313666666669">  <ele>0.0</ele>  <time>2012-12-09T10:54:23+0700</time  <extensions>  <ogt10:accuracy>0</ogt10:accuracy>  <speed>0.0</speed>  </extensions>  <trkpt>  </trk>  </gpx> |

* Thẻ xml: bắt đầu file GPX
* Thẻ gpx: là phần tử gốc trong file GPX
* Thẻ trk: chứa thông tin track, bao gồm nhiều điểm.
* Thẻ trkpt: chứa thông tin điểm, bao gồm kinh độ, vĩ độ, độ cao, thời gian track, và các thông số do người dùng định nghĩa như vận tốc, độ chính xác.