**Ôn tập MMT nâng cao [1](C:\\Users\\ASUS\\Desktop\\mmtnc\\Network.docx) [2](C:\\Users\\ASUS\\Desktop\\mmtnc\\decuong-mmt-160420174825.pdf)**

1. **Chương 1&2: Ôn lại MMT và chương IPv6**

### 1. Địa chỉ IPv6 là gì?

Internet Protocol Version 6 (IPv6) là giao thức lớp mạng cho phép truyền dữ liệu diễn ra qua mạng internet. IPv6 ra đời vào năm 1998 với mục đích duy nhất là tiếp quản và thay thế giao thức IPv4.

Giao thức IPv4, tiêu chuẩn trước đó, bao gồm bốn chuỗi số – mỗi chuỗi chứa ba chữ số được phân tách bằng dấu chấm. Một địa chỉ IPv4 tiêu chuẩn là 32 bit ví dụ như 255.255.255.255 và cho phép đưa vào hoạt động 4.2 tỷ địa chỉ IP duy nhất.

Với các thiết bị không dây và kết nối mạng tăng nhanh như hiện nay, theo dự kiến thì năm 2010, internet đã cạn kiệt tất cả các địa chỉ IPv4 duy nhất. Để đưa ra một tiêu chuẩn mới của giao thức lớp mạng cho phép tạo ra nhiều địa chỉ IP duy nhất, IPv6 đã được chuẩn hóa.

Giao thức IPv6, gồm 128 bit (2 mũ 6 bit), bao gồm tám chuỗi được đánh số, mỗi chuỗi chứa bốn ký tự (chữ số), được phân tách bằng dấu hai chấm. Và giao thức IPv6 này mang lại một lượng địa chỉ IP duy nhất có thể coi là vô hạn – chính xác là 340,282,366,920,938,463,463,374,607,431,768,211,456 địa chỉ IP. Nó cũng đảm bảo rằng sẽ không bao giờ hết các địa chỉ IP duy nhất để có thể gán cho các thiết bị.

Graphical user interface, text, application, email

Description automatically generated

* + - 1. **Địa chỉ IP là gì ?**

Là định danh một máy tính khi giao tiếp trong mạng như địa chỉ nhà

1. **Mô hình OSI khác giao thức TCP/IP như thế nào ? Hiện nay đa phần network được sử dụng mô hình nào ?**

**Diagram

Description automatically generated**

* Hiện nay đa phần network được sử dụng **giao thức TCP/IP**
* Điểm khác biệt lớn nhất giữa hai giao thức này có lẽ là sự kết hợp giữa các tầng với nhau. Đối với giao thức TCP/IP **thì tầng session và tầng presentation** được kết hợp với nhau trong **tầng ứng dụng**. Còn đối với mô hình OSI thì mỗi tầng khác nhau sẽ thực hiện một nhiệm vụ khác nhau.

1. **Các giao thức trong tầng của mô hình OSI cho ví dụ ?**

**Graphical user interface, text

Description automatically generated**

1. **Gói tin trong mạng được truyền như thế nào ?**

Gói tin trong mạng được truyền bằng cách đóng gói(escapsulation) sau đó phân rã, được truyền từ source nguồn tới đích(destination)

Diagram

Description automatically generated

1. **Có mấy loại địa chỉ mạng ?**

Địa chỉ public và địa chỉ private

Địa chỉ public thì giao tiếp với nhau trên internet external mọi người đều thấy, địa chỉ private thì hoạt động trong internal chỉ trong tổ chức đó thấy ví dụ như hộ gia đình, cty,…

1. **Địa chỉ IP classfull khác gì với IP classless**

Các địa chỉ IPv4 từ classfull được chia nhỏ ra thành các subnet(classless)

Vấn đề phát sinh ra ip classless là do địa chỉ ip classfull nếu sử dụng quá lãng phí khó bảo trì

Text

Description automatically generated

Table

Description automatically generated

Cách tính 2^m -2 với m là số **host hợp lệ** ví dụ **lớp A (lấy 32 - 8)** **là 2^24 – 2 host vì có 1 network**

Còn class B là lấy (2^16-2)x16 vì có 16 network tương tự lớp C

**Kích thước là 4byte tương đương 1 byte = 8bit sẽ có 32 bit tất cả**

1. **IPv4 khác IPv6 như thế nào ? Tại sao phát sinh ra IPv6**

**Text, letter

Description automatically generated**

Vấn đề phát sinh ra IPv6 là do địa chỉ IPv4 là 32bit ta tính đơn giản 2^32-2 ~ 4,3 tỉ mà vấn đề thiết bị tham gia môi trường mạng ngày càng tăng nên ko thể đáp ứng đủ nhu cầu nên phải phát sinh ra địa chỉ IPv6

1. **Chia subnet cơ bản xem lại bài tập của thầy**
2. **Phần về gói tin truyền từ nguồn tới đích gồm những gì các bạn tự xem. Đại khái là sẽ mang theo MAC và IP khi qua các router/modem/switch sẽ có sự thay đổi phần ví dụ của thầy có nếu bạn chép bài**
3. **Chương 3: TCP congestion control(Kiểm soát tắt nghẽn)**

**TCP reno**

**Chart, line chart

Description automatically generated**

1. **Xác định các khoảng thời gian mà TCP slow-start đang hoạt động**

Slow-start ta xem cột transmission round

Vậy ta có là [1,6] và [23,26] vì nó đi nhanh dần đều lên

1. **Xác định các khoảng thời gian mà TCP congestion-avoidance đang hoạt động**

Slow-start ta xem cột transmission round

Vậy ta có là [6,16] và [17,22] vì nó đi thẳng lên sẽ

1. **After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout event?(Sau 16 chỗ transmission round nó hỏi là mất dữ liệu cập nhật hay 3 ACK)**

**Chart, line chart

Description automatically generated**

* Vì chỗ 16 ta thấy đi xuống nhưng ko về đến 1 nên nó là mất do **3 ACK**

1. **After the 22ndtransmission round, is segment loss detected by a triple duplicate ACK or by a timeout event?(Sau 22 thì mất do 3 ACK hay time out)**

**Chart, line chart

Description automatically generated**

* Vì nhìn vào hình ta thấy mất do **timeout** do ta thấy nó xuống 1

1. **What is the *ssthreshold* value at the first transmission round? (Đại khái nó hỏi giá trị threshold đầu tiên là bao nhiêu)**

**Chart, line chart

Description automatically generated**

* Nhìn vào hình ta thấy từ **32** chỗ đường thẳng nhìn qua cột dọc

1. **What is the *ssthreshold* value at the 18th transmission round?(Nó hỏi chỗ cột ngang á là 18 thì giá trị threshold là bao nhiêu)**

**Chart, line chart

Description automatically generated**

* Ta lấy từ chỗ mà nó bị 3 ACK là lúc nó chưa mất là giá trị 42 sau khi mất nó sẽ giảm 1 nửa(3 ACK là luôn luôn giảm 1 nửa)
* **Vậy kq sẽ là 21**

1. **What is the *ssthreshold* value at the 24th transmission round?(Giá trị chỗ cột nằm ngang 24 là bao nhiêu)**

**Chart, line chart

Description automatically generated**

* Tương tự như câu trên ta thấy lúc mà chưa bị mất gói chỗ cột nằm ngang là 22 giá trị qua bên cột dọc là 26 ta chia 2 ra
* **Kết quả bằng 13**

1. **What will be the values of *cwind* and *ssthreshold* if packet loss is detected after the 26th round by receipt of triple duplicate ACKs?(Đại khái nó hỏi giả sử sau giá trị của cột ngang là 26 nó giả sử mất gói do 3 ACK thì điểm threshold đi lên sẽ là bao nhiêu)**

**Chart, line chart

Description automatically generated**

* Đơn giản thôi nhìn vào note ta thấy mất do 3 ACK thì từ threshold giảm một nửa mà threshold lúc đó là 4 thôi
* **NOTE**
* **Nếu mất gói do timeout thì kênh truyền nghẽn để an toàn sẽ xuống 1**
* **Nếu do 3 ACK thì từ threshold(giá trị lúc mất giảm đi 1 nửa) đi lên**

tham số *cwind* được lấy từ: (e) - nó không nằm trong header TCP, mà chỉ được xác định bằng cách gửi TCP.

1. Khi TCP khởi động chậm, việc nhận 2 ACK trước khi hết thời gian dẫn đến *tốc* độ tăng 2 byte MSS
2. a) tham số *ssthreshold* xác định thời điểm bắt đầu chậm kết thúc và tính năng tránh tắc nghẽn diễn ra.
3. Trong giai đoạn tránh tắc nghẽn, tốc độ gửi của TCP tăng tuyến tính.
4. What is “fast re-transmit: Trong TCP Reno, khi nhận được ba ACK trùng lặp, TCP sẽ truyền lại gói được cho là bị mất mà không cần chờ hết thời gian. Đây được gọi là truyền lại nhanh.
5. What is “fast recovery”? Trong TCP Reno, khi TCP thực hiện truyền lại nhanh (xem câu hỏi ở trên), thay vì bắt đầu giai đoạn khởi động chậm với *cwind* được đặt thành 1, nó chuyển trực tiếp đến giai đoạn tránh tắc nghẽn, với cwind được đặt thành ½ giá trị của nó khi mất gói. xảy ra. Đây được gọi là phục hồi nhanh. Truyền lại nhanh và phục hồi nhanh thường được thực hiện cùng nhau.

**Bài này ko hiểu có thể contact mình**

1. **So sánh flow control và congestion control ?**

* Kiểm soát luồng là người nhận kiểm soát mức độ mà người gửi đưa vào mạng
* Kiểm soát tắc nghẽn là người gửi cảm nhận tắc nghẽn trên mạng bằng cách định thời gian ACK và kiểm soát tốc độ gửi của nó.

1. Các dữ liệu mất gói do

* 3 ACK
* Timeout

1. TCP reno vs TCP Tahoe

* TCP: reno
* Mất gói do 3 ACK
* Sequence từ threshold đi lên
* TCP Tahoe
* Mất gói do timeout
* Đi từ 1 đi lên

1. TCP vs UDP

* UDP:
* Bỏ qua các kết nối lỗi
* Không thiết lập kết nối
* Kích thước header nhỏ
* TCP:
* Truyền tin cậy
* Kiểm soát luồng
* Kiểm soát tắt nghẽn
* Thiết lập kết nối

**Kiểm soát lưu lượng(flow control):**

* Nó đảm bảo rằng người gửi không quá tải người nhận.
* Đó là một hiện tượng cục bộ, không giống như kiểm soát tắc nghẽn.
* **Nó thường được khởi xướng bởi người gửi.**

**Điều khiển tắc nghẽn(congestion control):**

* Nó đảm bảo rằng mạng có thể xử lý tải các gói.
* Đó là một hiện tượng toàn cầu và ảnh hưởng đến mọi Máy chủ được kết nối với mạng đó.
* **Nó được điều khiển bởi bộ định tuyến(router).**

1. **Chương 4 & 5: Internet routing**

* Có 2 loại định tuyến
* Tĩnh: Do ta tự cấu hình
* Động:
* Distance vector: gởi theo định kì, gởi toàn bộ bảng định tuyến, vd: RIP,IGRP,
* Link state: Gửi khi có thay đổi, gửi tình trạng kết nối vd: OSPF, ISIS,

**Sơ đồ Routing**

**Routing**

**IGP(trong 1 AS) EGP(Giữa các AS với nhau)**

**Distance vector LinkState BGP**

**RIP OSPF**

1. **RIP khác gì so với OSPF(hay distance vector khác gì so với linkstate)**

* **RIP(distance vector):**
* Xác định hướng tới mạng đích
* Cập nhật định kì các router khi có thay đổi
* Xác định khoảng cách tới mạng đích
* **OSPF(linkstate):**
* Trạng thái của từng link
* Biết được topo của mạng hiện hành
* Không cập nhật định kì

1. **Các mạng hiện nay giao tiếp với nhau thông qua giao thức nào ? với BGP nó tìm cách nào đi từ nguồn tới đích**

BGP(path vector), BGP sẽ

1. **BGP định tuyến theo cách nào ?**

BGP định tuyến bằng cách sử dụng các thuộc tính của các tuyến đường. Mỗi tuyến đường là danh sách các AS cần phải đi qua.

1. **Bản tin BGP có 4 loại thông điệp?**

* **Open:** Thiết lập kết nối với hàng xóm
* **Keep Alive:** Bắt tay thường xuyên với hàng xóm
* **Notif:** Thông báo với hàng xóm
* **Update:** Thông báo tuyến đường mới hoặc hủy 1 tuyến đường đã quảng bá trước đó

1. **Routing loop là gì ? Nguyên nhân và tác hại ? Cách ngăn ngừa**

* Routing loop là tình trạng gói tin đi qua nhiều router mà không đến đích
* Nguyên nhân do cấu hình sai, hội tụ chậm,
* Tác hại: Tiêu tốn bandwidth, mạng không hội tụ, thông tin cập nhật định tuyến bị mất hoặc ko xử lý kịp
* Cách chống loop: với distance vector thì thiết lập giá trị metric lớn nhất để xác định đường đi đó không khả dụng

1. **Fdsf**
2. **Fsdf**
3. **Dfsf**
4. **Sfsdf**
5. **Dsfsd**
6. **Chương 7: Ảo hóa & cloud**

Phần ảo hóa này các nhóm đã làm nên mình ko note nhiều, nếu ko chỗ nào có câu hỏi cần hỏi mình ,mình sẽ trả lời trong sự hiểu biết và kinh nghiệm làm việc của mình

* Virtual machine(VM): hoạt động như một hệ điều hành mà trong đó các phần cứng sẽ chia sẻ các tài nguyên với nhau, ta có thể dễ dàng snapshort,clone,…
* Có các loại ảo hóa: Ảo hóa server, ảo hóa desktop, ảo hóa về memory, disk, network,..
* Ảo hóa sử dụng KVM, VMWare
* Tóm lại ảo hóa khác gì so với cloud

Máy ảo khi request 1 VPS thì thời gian có thể lâu còn khi cloud chúng ta có thể tạo trực tiếp trên đó mà ko cần thông qua trung gian nào, có thể chia sẻ resource pool dễ dàng

* Các tính năng của máy ảo khác với máy thật

+ Có thể tạo nhiều VM trên đó sử dụng nhiều services

+ Tiết kiệm chi phí hơn so với mỗi hardware sử dụng 1 distro (OS)

* Ảo hóa
* Được cô lập
* Bảo mật nâng cao
* Dễ dàng mô phỏng kiến trúc khác nhau và cùng tồn tại

1. **Chương 8: Container**

**Phần này cũng ko có gì nhiều tùy cơ ứng biến**

* Khác với ảo hóa và cloud container hoạt động độc lập không dựa vào các tài nguyên như cpu,ram,disk,…
* Không sử dụng kernel
* Có thể dễ dàng di chuyển qua các máy khác nhau
* Hạn chế của máy ảo

+ Máy ảo vẫn yêu cầu cpu,ram,disk

+ Chạy nhiều máy ảo thì càng nhiều tài nguyên

+ Sử dụng hđh ko sử dụng hết gây lãng phí

+ Tính di động ko được đảm bảo

* Docker image & container

+ Image:

* Chỉ đọc để tạo ra container
* Được sử dụng bởi mình hoặc docker ng khác tạo
* Được lưu trữ trong docker hub

+ Container:

* Chứa mọi thứ cần thiết để chạy ứng dụng
* Dựa trên 1 hoặc nhiều image

1. **Các bài các nhóm khác seminar có thể tham khảo**

[TÌM HIỂU VÀ KHAI TRIỂN CI/CD CƠ BẢN](file:///C:\Users\ASUS\Desktop\mmtnc\Report_Seminar_CICDCoBan_Nhom07_3ML.docx)

[TÌM HIỂU VÀ TRIỂN KHAI CƠ BẢN Software Defined Networking(SDN)](file:///C:\Users\ASUS\Desktop\mmtnc\Đạt%20-%2012%20-%20Nguyễn%20Anh%20Khương%20-%20SDN.pptx)

## [Tìm hiểu và triển khai Elastic Search](file:///C:\Users\ASUS\Desktop\mmtnc\Báo%20cáo.docx)

[**KUBERNETES**](file:///C:\Users\ASUS\Desktop\mmtnc\Bao%20cao.docx)

[TÌM HIỂU SOFTWARE DEFINE NETWORK, TRIỂN KHAI VÀ THỬ NGHIỆM](file:///C:\Users\ASUS\Desktop\mmtnc\My%20Nguyen%20-%20Seminar_Socket.pdf)

[Báo cáo Chủ đề: CDN](file:///C:\Users\ASUS\Desktop\mmtnc\20424025_20424038_20424056_BaoCao_CDN.pdf)

[Tìm hiểu Ansible và triển khai thử nghiệm](file:///C:\Users\ASUS\Desktop\mmtnc\Report.pdf)

[Tìm hiểu Kubernetes và triển khai thử nghiệm](file:///C:\Users\ASUS\Desktop\mmtnc\Kubernetes.pdf)

Graphical user interface, text, application, email

Description automatically generated

A picture containing table

Description automatically generated

# Socket là gì? Khái niệm cần biết về giao thức TCP/IP và UDP

Hãy cùng Mắt Bão tìm hiểu chi tiết về khái niệm Socket là gì, sự quan trọng, cách hoạt động và tất cả những thứ liên quan đến Socket và TCP/IP, UDP!

## Socket là gì?

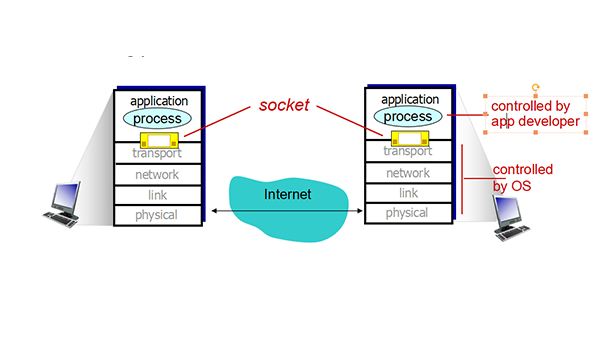
*Socket là điểm cuối end-point trong liên kết truyền thông hai chiều (two-way communication) biểu diễn kết nối giữa Client – Server*. Các lớp Socket được ràng buộc với một cổng port (thể hiện là một con số cụ thể) để các tầng TCP (TCP Layer) có thể định danh ứng dụng mà dữ liệu sẽ được gửi tới. Ứng dụng thực tiễn của Socket là gì? Cùng Mắt Bão tìm hiểu nhé!

Socket hoạt động thông qua các tầng TCP hoặc TCP Layer định danh ứng dụng, từ đó truyền dữ liệu thông qua sự ràng buộc với một cổng port

Socket là giao diện lập trình ứng dụng mạng được dùng để truyền và nhận dữ liệu trên internet. Giữa hai chương trình chạy trên mạng cần có một liên kết giao tiếp hai chiều, hay còn gọi là two-way communication để kết nối 2 process trò chuyện với nhau. Điểm cuối (endpoint) của liên kết này được gọi là socket.

Một chức năng khác của socket là giúp các tầng TCP hoặc TCP Layer định danh ứng dụng mà dữ liệu sẽ được gửi tới thông qua sự ràng buộc với một cổng port (thể hiện là một con số cụ thể), từ đó tiến hành kết nối giữa client và server.

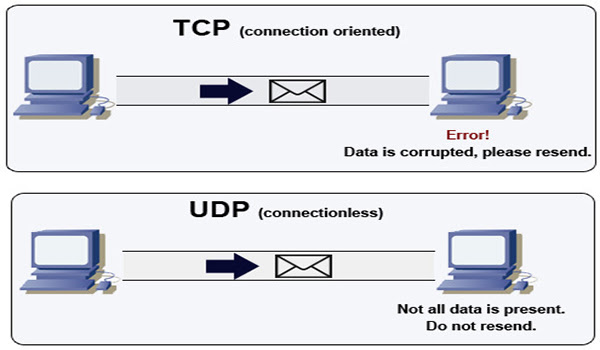
## Tại sao người dùng lại cần đến socket?

****Có thể sử dụng cùng lúc nhiều socket liên tục để tiết kiệm thời gian cũng như nâng cao năng suất làm việc

Ưu điểm lớn nhất của socket là hỗ trợ hầu hết các hệ điều hành bao gồm MS Windows, Linux,… Ngoài ra, socket cũng được sử dụng với nhiều ngôn ngữ lập trình, gồm C, C++, Java, Visual Basic, Visual C++,… nên nó có thể tương thích với hầu hết mọi đối tượng người dùng với những cấu hình máy khác nhau.

Đặc biệt, người dùng cũng có thể chạy cùng một lúc nhiều socket liên tục, giúp nâng cao hiệu suất làm việc, cũng như tiết kiệm thêm nhiều thời gian và công sức hơn.

## Socket hoạt động như thế nào?

****Chức năng của socket là kết nối giữa client và server thông qua TCP/IP và UDP để truyền và nhận giữ liệu qua Internet

Như đã đề cập trước đó, chức năng của socket là kết nối giữa client và server thông qua TCP/IP và UDP để truyền và nhận giữ liệu qua Internet. Giao diện lập trình ứng dụng mạng này chỉ có thể hoạt động khi đã có thông tin về thông số IP và số hiệu cổng của 2 ứng dụng cần trao đổi dữ liệu cho nhau.

2 ứng dụng cần truyền thông tin phải đáp ứng điều kiện sau thì socket mới có thể hoạt động:

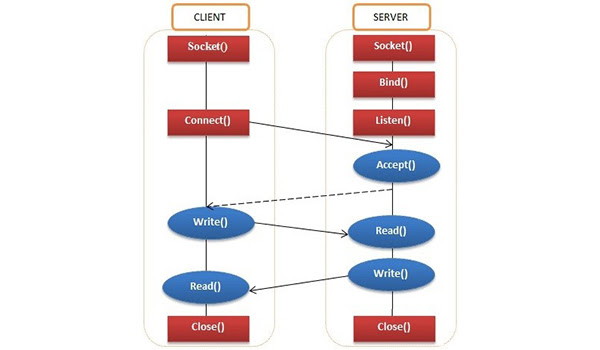
* 2 ứng dụng có thể nằm cùng trên một máy hoặc 2 máy khác nhau
* Trong trường hợp 2 ứng dụng cùng nằm trên một máy, số hiệu cổng không được trùng nhau.

## Phân loại Socket

Socket được chia làm 3 phân loại khác nhau. Dưới đây, hãy cùng Mắt Bão tìm hiểu chi tiết về:

* Stream Socket
* Datagram Socket
* Websocket
* Unix socket

### *Stream Socket là gì?*

******Stream Socket chỉ hoạt động khi server và client đã kết nối với nhau

Stream Socket hay còn gọi là socket hướng kết nối, là socket hoạt động thông qua giao thức TCP (Transmission Control Protocol). Stream Socket chỉ hoạt động khi server và client đã kết nối với nhau. Ngược lại đây chính là định nghĩa cho câu hỏi: “*Giao thức TCP là gì?*“

Ngoài ra, bạn cũng nên tham khảo qua bài viết: “[*TCP/IP là gì?*](https://wiki.matbao.net/kb/tcp-ip-la-gi-uu-va-nhuoc-diem-cua-tcp-ip-so-voi-mo-hinh-osi/)”

*TCP/IP viết tắt của Transmission Control Protocol (TCP) và Internet Protocol (IP) là giao thức cài đặt truyền thông, chồng giao thức mà hầu hết các mạng máy tính ngày nay đều sử dụng để kết nối*. TCP/IP được đặt theo tên của 2 giao thức là giao thức điều khiển giao vận và giao thức liên mạng. Đây là 2 giao thức đầu tiên trên thế giới được định nghĩa. TCP/IP được phát triển vào năm 1978 bởi Bob Kahn và Vint Cerf.

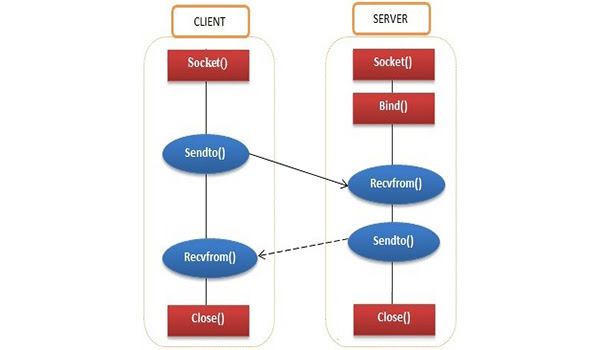
#### Ưu điểm của Stream Socket là gì?

* Dữ liệu truyền đi được đảm bảo truyền đến đúng nơi nhận, đúng thứ tự với thời gian nhanh chóng
* Mỗi thông điệp gửi đi đều có xác nhận trả về để thông báo cho người dùng thông tin về quá trình truyền tải.

#### Nhược điểm của Stream Socket là gì?

* Giữa máy chủ và máy nhận chỉ có 1 IP, nên khi kết nối, 1 máy phải chờ máy còn lại chấp nhận kết nối.

### *Datagram Socket là gì?*

******Datagram Socket có thể hoạt động kể cả khi không có sự thiết lập kết nối giữa 2 máy với nhau

Datagram Socket hay còn gọi là socket không hướng kết nối, là socket hoạt động thông qua giao thức UDP ( User Datagram Protocol). Datagram Socket có thể hoạt động kể cả khi không có sự thiết lập kết nối giữa 2 máy với nhau. Nói cách khác, đây cũng chính là định nghĩa cho câu hỏi: “Giao thức UDP là gì?“

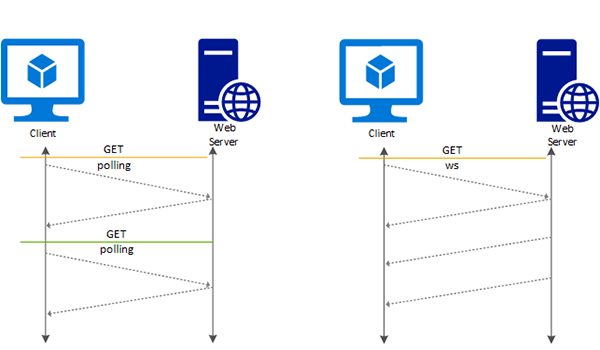
#### Ưu điểm của Datagram Socket là gì?

* Quá trình kết nối và truyền tải thông tin đơn giản, không cần thực hiện nhiều thao tác.
* Thời gian truyền tải dữ liệu cực nhanh.

#### Nhược điểm của Datagram Socket là gì?

* Quá trình truyền thông tin không đảm bảo tin cậy, thông tin có thể truyền sai thứ tự hoặc bị lặp.

### *Websocket là gì?*

******Websocket có chức năng giúp cho việc kết nối qua lại trên internet giữa client và server diễn ra một cách nhanh chóng và hiệu quả hơn

Websocket là công cụ hỗ trợ việc kết nối qua lại trên internet giữa client và server. Giúp diễn ra nhanh chóng và hiệu quả hơn thông qua việc sử dụng TCP socket. Không chỉ sử dụng riêng cho ứng dụng web, Websocket có thể áp dụng cho bất kì ứng dụng nào khác cần có sự trao đổi thông tin trên Internet.

#### Ưu điểm của Websocket là gì?

Websocket mang lại nhiều ưu điểm trong việc kết nối giữa client và server. Cụ thể như sau:

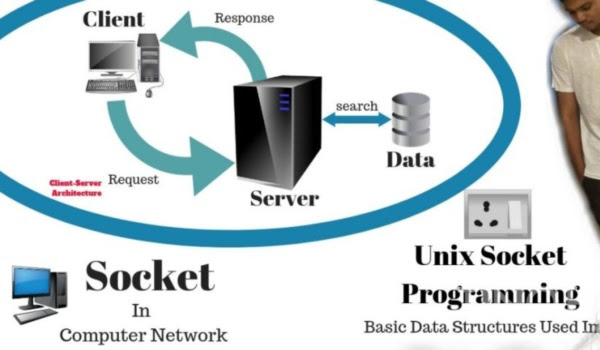
* Tăng tốc độ truyền tải thông tin giữa 2 chiều
* Dễ phát hiện và xử lý trong trường hợp có lỗi xảy ra
* Dễ dàng sử dụng, không cần cài đặt thêm các phần mềm bổ sung khác
* Không cần sử dụng nhiều phương pháp kết nối khác nhau

#### Nhược điểm của Websocket là gì?

Một số nhược điểm của Websocket mà bạn cần lưu ý khi sử dụng có thể kể đến như:

* Chưa hỗ trợ trên tất cả các trình duyệt
* Với các dịch vụ có phạm vi yêu cầu, Websocket chưa hỗ trợ hoàn toàn.

### *Unix socket là gì?*

******Unix socket giúp cho tốc độ kết nối và truyền tải dữ liệu giữa các ứng dụng trên cùng một máy tính diễn ra nhanh, nhẹ và hiệu quả hơn.

Unix socket là điểm giao tiếp hỗ trợ trao đổi giữa các ứng dụng khác nhau ngay trên cùng máy tính. Mọi hoạt động Unix socket diễn ra ngay ở nhân của hệ điều hành. Nhờ vậy, tốc độ kết nối và truyền tải giữa các ứng dụng nhanh, nhẹ và hiệu quả hơn.

Unix socket có thể tránh được các bước kiểm tra hoặc routing. Vì thế mà quá trình truyền tin sẽ đảm bảo và dễ dàng hơn.

#### Ưu điểm Unix socket là gì?

Unix socket có nhiều ưu điểm vượt trội. Nó hỗ trợ cho việc truyền tải nhanh hơn và hiệu quả hơn. Trong đó, tiêu biểu phải kể đến các ưu điểm như:

* Tăng tốc độ truy cập MySQL lên đến 30-50%
* Giảm thời gian latency xuống, từ 60ms còn 5ms
* Tăng PostgreSQL lên  hơn 30%
* Tăng Redis lên 50%
* …

#### Nhược điểm Unix socket là gì?

Bên cạnh các ưu điểm nổi bật phía trên, Unix socket cũng còn tồn tại một số nhược điểm như sau:

* Trong trường hợp các ứng dụng nằm trên những máy chủ khác nhau, sẽ không thể kết nối bằng Unix socket.
* Vấn đề phân quyền giữa các tệp tin trên Unix socket đôi khi vẫn xảy ra, ảnh hưởng đến việc sử dụng và thao tác.

# **4 Mô Hình Dịch Vụ Cloud Computing (Điện Toán Đám Mây)**

Cùng Netsa tìm hiểu các mô hình dịch vụ [**Cloud Computing**](https://netsa.vn/giai-phap-cloud-computing/) (Điện Toán Đám Mây) phân tích ưu điểm, nhược điểm từng mô hình theo nhu cầu sử dụng mà doanh nghiệp chọn mô hình nào là hợp lý nhất.

**Nói sơ qua “độ tăng – độ giảm” của điện toán đám mây**  
\* Sử dụng các tài nguyên tính toán động (Dynamic computing resources) : Các tài nguyên được cấp phát cho doanh nghiệp đúng như những gì doanh nghiệp muốn một cách tức thời. Thay vì việc doanh nghiệp phải tính toán xem có nên mở rộng hay không, phải đầu tư bao nhiêu máy chủ thì nay doanh nghiệp chỉ cần yêu cầu chúng tôi cần thêm tài nguyên tương đương với …và đám mây sẽ tự tìm kiếm tài nguyên rỗi để cung cấp cho bạn.

\* Giảm chi phí : Doanh nghiệp sẽ có khả năng cắt giảm chi phí để mua bán, cài đặt và bảo trì tài nguyên. Rõ ràng thay vì việc phải cử một chuyên gia đi mua máy chủ, cài đặt máy chủ, bảo trì máy chủ thì nay bạn chẳng cần phải làm gì ngoài việc xác định chính xác tài nguyên mình cần và yêu cầu. Quá tiện!.

\* Giảm độ phức tạp trong cơ cấu của doanh nghiệp : Doanh nghiệp sản xuất hàng hóa mà lại phải có cả một chuyên gia IT để vận hành, bảo trì máy chủ thì quá tốn kém. Nếu outsource được quá trình này thì doanh nghiệp sẽ chỉ tập trung vào việc sản xuất hàng hóa chuyên môn của mình và giảm bớt được độ phức tạp trong cơ cấu.

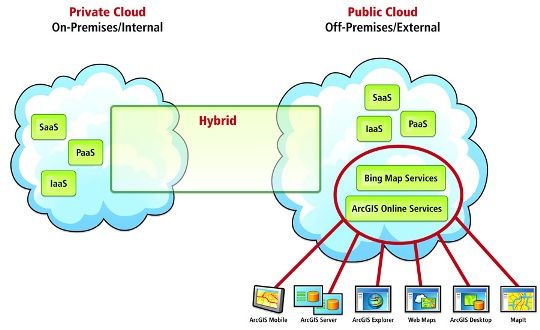
\* Tăng khả năng sử dụng tài nguyên tính toán : Một trong những câu hỏi đau đầu của việc đầu tư tài nguyên (ví dụ máy chủ) là bao lâu thì nó sẽ hết khấu hao, đầu tư như thế có lãi hay không, có bị outdate về công nghệ hay không … Khi sử dụng tài nguyên trên đám mây thì bạn không còn phải quan tâm tới điều này nữa.

***Cloud Computing có 4 mô hình dịch vụ (mô hình sản phẩm ):***

* **Public Cloud:** Đám mây công cộng (là các dịch vụ trên nền tảng Cloud Computing để cho các cá nhân và tổ chức thuê, họ dùng chung tài nguyên).
* **Private Cloud:** Đám mây riêng (dùng trong một doanh nghiệp và không chia sẻ với người dùng ngoài doanh nghiệp đó)
* **Hybrid Cloud:** Là mô hình kết hợp (lai) giữa các mô hình Public Cloud và Private Cloud.
* **Community Cloud:** Đám mây cộng đồng (là các dịch vụ trên nền tảng Cloud computing do các công ty cùng hợp tác xây dựng và cung cấp các dịch vụ cho cộng đồng).

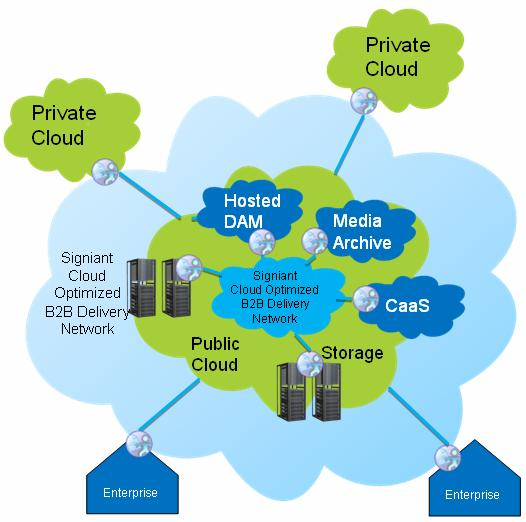
**1. Public Cloud (Đám mây “công cộng”)**

Định nghĩa: Là các dịch vụ được bên thứ 3 (người bán) cung cấp. Chúng tồn tại ngoài tường lửa của công ty và được nhà cung cấp đám mây quản lý. Nó được xây dựng nhằm phục vụ cho mục đích sử dụng công cộng, người dùng sẽ đăng ký với nhà cung cấp và trả phí sử dụng dựa theo chính sách giá của nhà cung cấp. Public cloud là mô hình triển khai được sử dụng phổ biến nhất hiện nay của cloud computing.

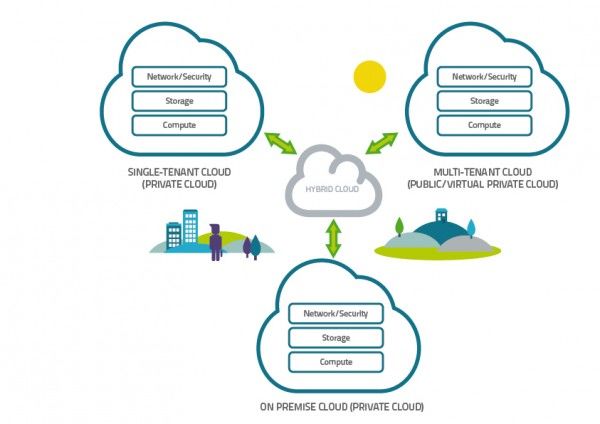
  
Đối tượng sử dụng: Bao gồm người dùng bên ngoài internet. Đối tượng quản lý là nhà cung cấp dịch vụ.  
Ưu điểm:  
Phục vụ được nhiều người dùng hơn, không bị giới hạn bởi không gian và thời gian.  
Tiết kiệm hệ thống máy chủ, điện năng và nhân công cho doanh nghiệp.  
Nhược điểm:  
Các doanh nghiệp phụ thuộc vào nhà cung cấp không có toàn quyền quản lý.  
Gặp khó khăn trong việc lưu trữ các văn bản, thông tin nội bộ.

Tuy nhiên Public Cloud có một trở ngại, đó là vấn đề mất kiểm soát về dữ liệu và vấn đề an toàn dữ liệu. Trong mô hình này mọi dữ liệu đều nằm trên dịch vụ Cloud, do nhà cung cấp dịch vụ Cloud đó bảo vệ và quản lý. Chính điều này khiến cho khách hàng, nhất là các công ty lớn cảm thấy không an toàn đối với những dữ liệu quan trọng của mình khi sử dụng dịch vụ Cloud.

**2. Private Cloud (Đám mây “doanh nghiệp”)**  
Định nghĩa: Private cloud là các dịch vụ điện toán đám mây được cung cấp trong các doanh nghiệp. Những “đám mây” này tồn tại bên trong tường lửa của công ty và được các doanh nghiệp trực tiếp quản lý. Đây là xu hướng tất yếu cho các doanh nghiệp nhằm tối ưu hóa hạ tầng công nghệ thông tin.

  
Đối tượng sử dụng: Nội bộ doanh nghiệp sử dụng và quản lý  
Ưu điểm: Chủ động sử dụng, nâng cấp, quản lý, giảm chi phí, bảo mật tốt,…  
Nhược điểm:  
Khó khăn về công nghệ khi triển khai và chi phí xây dựng, duy trì hệ thống.  
Hạn chế sử dụng trong nội bộ doanh nghiệp, người dùng ở ngoài không thể sử dụng.

**3. Hybrid Cloud (Đám mây “lai”)**  
Định nghĩa: Là sự kết hợp của private cloud và public cloud. Cho phép ta khai thác điểm mạnh của từng mô hình cũng như đưa ra phương thức sử dụng tối ưu cho người sử dụng. Những “đám mây” này thường do doanh nghiệp tạo ra và việc quản lý sẽ được phân chia giữa doanh nghiệp và nhà cung cấp điện toán đám mây công cộng.

  
Đối tượng sử dụng: Doang nghiệp và nhà cung cấp quản lý theo sự thỏa thuận. Người sử dụng có thể sử dụng các dịch vụ của nhà cung cấp và dịch vụ riêng của doanh nghiệp.  
Ưu điểm: Doanh nghiệp 1 lúc có thể sử dụng được nhiều dịch vụ mà không bị giới hạn.  
Nhược điểm: Khó khăn trong việc triển khai và quản lý. Tốn nhiều chi phí.

Doanh nghiệp có thể chọn để triển khai các ứng dụng trên Public, Private hay Hybrid Cloud tùy theo nhu cầu cụ thể. Mỗi mô hình đều có điểm mạnh và yếu của nó. Các doanh nghiệp phải cân nhắc đối với các mô hình Cloud Computing mà họ chọn. Và họ có thể sử dụng nhiều mô hình để giải quyết các vấn đề khác nhau. Nhu cầu về một ứng dụng có tính tạm thời có thể triển khai trên Public Cloud bởi vì nó giúp tránh việc phải mua thêm thiết bị để giải quyết một nhu cầu tạm thời. Tương tự, nhu cầu về một ứng dụng thường trú hoặc một ứng dụng có những yêu cầu cụ thể về chất lượng dịch vụ hay vị trí của dữ liệu thì nên triển khai trên Private hoặc Hybrid Cloud.

1. **Community Cloud:** Đám mây cộng đồng (là các dịch vụ trên nền tảng Cloud computing do các công ty cùng hợp tác xây dựng và cung cấp các dịch vụ cho cộng đồng).

**Tuy nhiên, “Kubernetes và Docker” cũng là một cụm từ hơi gây hiểu lầm. Khi bạn phân tích nó, những từ này không có nghĩa giống như nghĩa mà nhiều người nghĩ tới, vì Docker và Kubernetes không phải là đối thủ cạnh tranh trực tiếp. Docker là một nền tảng container hóa và Kubernetes là một bộ điều phối container cho các nền tảng container như Docker.**

## Docker là gì?

Docker hiện là nền tảng container phổ biến nhất. Docker xuất hiện trên thị trường vào đúng thời điểm và ngay từ đầu đã là mã nguồn mở , điều này làm cho nó vươn lên vị trí thống trị thị trường hiện nay. 30% doanh nghiệp hiện đang sử dụng Docker trong môi trường AWS của họ và con số đó tiếp tục tăng lên.

### Docker được dùng để làm gì?

Khi hầu hết mọi người nói về Docker, họ đang nói về Docker Engine, môi trường thực thi (runtime) cho phép bạn build và chạy các containers. Nhưng trước khi bạn có thể chạy một container Docker, chúng phải được build và định nghĩa từ một Docker File. Docker File xác định mọi thứ cần thiết để chạy image bao gồm thông số kỹ thuật cho OS network và vị trí các file. Bây giờ bạn đã có một Docker file, bạn có thể build một image Docker, đó là thành phần tĩnh, có thể vận chuyển được và được chạy trên Docker Engine. Và Docker thậm chí còn có một dịch vụ gọi là Docker Hub, nơi bạn có thể lưu trữ và chia sẻ các image.