**Ôn tập MMT nâng cao**

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

**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**

[**https://www.dropbox.com/sh/zwm8gpnkr5xetue/AABPjDJNGVu2guiEDzkA1Lz2a?dl=0&fbclid=IwAR1zK7VMVu5uX\_HEicpymTcZD4hC6iJhzYFzO4XpyT2zPA7m1OuzpHnkQUY**](https://www.dropbox.com/sh/zwm8gpnkr5xetue/AABPjDJNGVu2guiEDzkA1Lz2a?dl=0&fbclid=IwAR1zK7VMVu5uX_HEicpymTcZD4hC6iJhzYFzO4XpyT2zPA7m1OuzpHnkQUY)

C**âu 1: Mô hình hybric cloud** => Nếu như Public Cloud có thể mở rộng và triển khai nhanh chóng, Private Cloud được quản lý nội bộ, an toàn hơn nhưng ít khả năng mở rộng hơn, thì Hybrid Cloud là tổng hòa của 2 mô hình nói trên. Hybric sử dụng các môi trường đám mây công cộng và đám mây riêng, phối hợp tạo ra môi trường phù hợp nhất cho doanh nghiệp. Đây được coi là mô hình tốt nhất bởi nó cho phép doanh nghiệp xây dựng một giải pháp tùy chỉnh và đáp ứng hết các nhu cầu.

**Câu 2: BGP sử dụng cơ chế gì phát hiện routing loop ? Giải thích**

=> BGP sử dụng AS path để phát hiện routing loop Vì S\_PATH lưu danh sách các AS mà nó đi qua, nếu bất kì AS nào thấy chính nó trong list đã có thì phát hiện là tồn tại vòng lặp.

**Câu 3: Sự khác biệt flowcontrol 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.

**Câu 3: Tiêu chí lựa chọn đường đi trong BGP là gì ? Có sử dụng thuật toán gì ko ?**

Việc lựa chọn tuyến đường tốt nhất dựa trên các thuộc tính của tuyến đường

A picture containing text, newspaper, document

Description automatically generated

**Câu 4: Thế nào là mạng ko liên tục ? Tại sao mạng RIPv1 lại ko hoạt động được ở mạng ko liên tục?**

**KHÓ**

**Câu 5: Docker là gì ? Phân biệt docker và Kubernetes ?**

* Docker là nền tảng phần mềm cho phép bạn dựng, kiểm thử và triển khai ứng dụng một cách nhanh chóng. Docker đóng gói phần mềm vào các đơn vị tiêu chuẩn hóa được gọi là container có mọi thứ mà phần mềm cần để chạy, trong đó có thư viện, công cụ hệ thống, mã và thời gian chạy. Bằng cách sử dụng Docker, bạn có thể nhanh chóng triển khai và thay đổi quy mô ứng dụng vào bất kỳ môi trường nào và biết chắc rằng mã của bạn sẽ chạy được.
* Phân biệt

+ Docker chạy trên một node duy nhất, trong khi Kubernetes được thiết kế đểchạytrênmộtcụm-cluster.  
+ Docker có thể được sử dụng mà không cần Kubernetes, trong khi Kubernetes cần môi trường container thời gian thực để tổ chức.

**Câu 6: Cho biết 3 khác nhau giữa linkstate và distance vector ?**

* **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

**Câu 7: Giao thức ARP còn sử dụng trong IPv6 ko ? Nếu ko có giao thức nào thay thế ?**

* Ko, giao thức thay thế là NDP(Neighbor Discovery Protocol) vì vấn đề bảo mật ARP giả mạo, sao chép MAC

**Câu 8: Phân biệt CI, CD với deploy ?**

* **CI (Continuous Integration)** và **CD (Continuous Delivery)**, ý nói là quá trình tích hợp (integration) thường xuyên, nhanh chóng hơn khi code cũng như thường xuyên cập nhật phiên bản mới (delivery).
* Deploy là cách thủ công để deploy code lên server

**Câu 9: Elasticserch là gì ? cho biết chức năng quan trọng của nó ?**

* Elasticsearch là một search engine.
* Elasticsearch được kế thừa từ Lucene Apache
* Elasticsearch thực chất hoặt động như 1 web server, có khả năng tìm kiếm nhanh chóng (near realtime) thông qua giao thức RESTful
* Elasticsearch có khả năng phân tích và thống kê dữ liệu
* Elasticsearch chạy trên server riêng và đồng thời giao tiếp thông qua RESTful do vậy nên nó không phụ thuộc vào client viết bằng gì hay hệ thống hiện tại của bạn viết bằng gì. Nên việc tích hợp nó vào hệ thống bạn là dễ dàng, bạn chỉ cần gửi request http lên là nó trả về kết quả.
* Elasticsearch là 1 hệ thống phân tán và có khả năng mở rộng tuyệt vời (horizontal scalability). Lắp thêm node cho nó là nó tự động auto mở rộng cho bạn.

**Câu 10: Routing loop xảy ra ở distance hay linkstate hay cả 2 ? Giải thích ?**

Routing loop xảy ra ở distance vector ? vì distance vector Router hội tụ chậm , dẫn đến việc sai lệch trong bảng định tuyến gây nên hiện tượng loop còn Link State không gửi bảng định tuyến của mình , mà chỉ gửi tình trạng của các đường link trong linkstate-database của mình đi cho các router khác, các router sẽ áp dụng giải thuật SPF (shortest path first ) , để tự xây dựng routing-table riêng cho mình

\* router sẻ gửi routing-table của mình cho tất cả các router được nối trực tiếp với nó . Các router đó sau đó so sánh với bảng routing-table mà mình hiện có và kiểm tra lại các tuyến đường của mình với các tuyến đường mới nhận được, tuyến đường nào tối ưu hơn sẻ được đưa vào routing-table

**Câu 11: kubernet la gi ? cac chuc nang chinh ?**

* Kubernetes là một nền tảng nguồn mở, khả chuyển, có thể mở rộng để quản lý các ứng dụng được đóng gói và các service, giúp thuận lợi trong việc cấu hình và tự động hoá việc triển khai ứng dụng. Kubernetes là một hệ sinh thái lớn và phát triển nhanh chóng. Các dịch vụ, sự hỗ trợ và công cụ có sẵn rộng rãi.
* Chính

+ Điều phối bộ nhớ: Kubernetes cho phép bạn tự động mount một hệ thống lưu trữ mà bạn chọn, như local storages, public cloud providers, v.v.

+ **Tự động rollouts và rollbacks**

**+ Đóng gói tự động**

**+ Tự phục hồi**

**+ Quản lý cấu hình và bảo mật**

**Câu 12: ansible là gì ? Dùng để làm gì**

* Ansible là một trong những công cụ quản lý cấu hình hiện đại, nó tạo điều kiện thuận lợi cho công việc cài đặt, quản lý và bảo trì các server từ xa, với thiết kế tối giản giúp người dùng cài đặt và chạy nhanh chóng.
* Dùng
  + Sử dụng hệ thống kiểm soát version để quản lý mọi thay đổi trong cơ sở hạ tầng.
  + Sử dụng lại các tập lệnh cấp phép cho nhiều môi trường server như phát triển, thử nghiệm và production.
  + Làm việc trong môi trường phát triển chuẩn hóa bằng cách chia sẻ các tập lệnh cấp phép.
  + Hợp lý hóa quá trình sao chép server, tạo điều kiện cho việc khôi phục các lỗi nghiêm trọng.
  + Cung cấp cách kiểm soát một đến hàng trăm server từ một vị trí tập trung, giúp cải thiện đáng kể hiệu quả và tính toàn vẹn của cơ sở hạ tầng server.

**Câu 13: Docker là gì ? dùng làm gì ?**

* Docker là nền tảng phần mềm cho phép bạn dựng, kiểm thử và triển khai ứng dụng một cách nhanh chóng. Docker đóng gói phần mềm vào các đơn vị tiêu chuẩn hóa được gọi là [container](https://aws.amazon.com/vi/containers/) có mọi thứ mà phần mềm cần để chạy, trong đó có thư viện, công cụ hệ thống, mã và thời gian chạy. Bằng cách sử dụng Docker, bạn có thể nhanh chóng triển khai và thay đổi quy mô ứng dụng vào bất kỳ môi trường nào và biết chắc rằng mã của bạn sẽ chạy được.
* Việc setup và deploy application lên một hoặc nhiều server rất vất vả từ việc phải cài đặt các công cụ, môi trường cần cho application đến việc chạy được ứng dụng chưa kể việc không đồng nhất giữa các môi trường trên nhiều server khác nhau. Chính vì lý do đó Docker được ra đời để giải quyết vấn đề này.