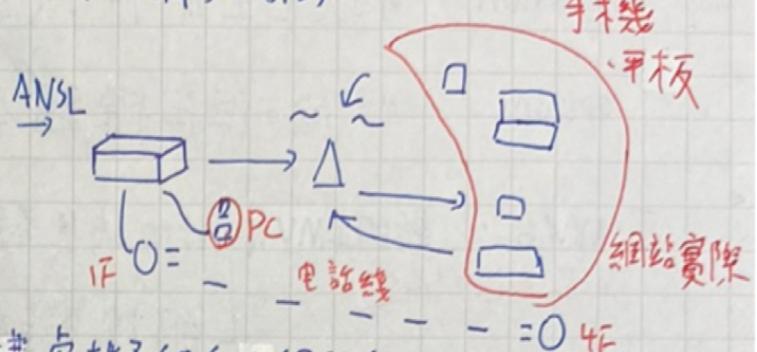


每部電腦的IP位址都是獨一無二的，(有32 bit)

NAT 網路位置轉換



VPN 利用公開的網路系統來建構虛擬的私人網路位址

→ (IP位址是私有的)

DHCP 動態主機設定協議，DHCP伺服器會自動分配給每個用戶IP位址並指定

Port forwarding 端口轉發，是SSH為網路安全通信的一種方法

DDNS 一種自動更新名稱伺服器內容的技術

DNS伺服器

192.168.X.X 都是
10.X.X.X 私有IP
範圍

TCP ≥ 20 byte IPv4 ≥ 20 byte IPv6 ≥ 40 byte 標頭

IPV6：動機（能負荷遠超 IPV4 的位址數量）

能將推出的32位地址空間完全分配

標頭格式有助於加快處理或轉發速度

標頭更改以促進 QoS

標頭：數據報格式：固定長度是 40 byte，字節標頭不允許分片

優先級：識別流中數據的優先級

→ 沒有定義的很好

流標頭：標識同一（流）中的數據報

下一個標頭：標識數據的上層協議

| Ver | Pri | flow label | |
|--------------------------------|-----|------------|----------|
| payload len | | next hdr | hoplimit |
| Source address (128 bits) | | | |
| destination address (128 bits) | | | |
| data | | | |
| ← | | 32 bits | → |

IPv4 的其他變化

吳家穎

checksum：可以完全刪除以減少每一跳的處理時間。
送錯就丟掉不影響

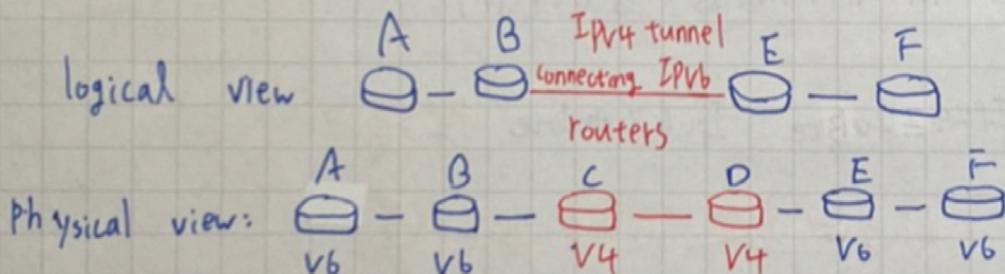
options：允許的，但在標題之外，由下一個標題字段指示。

ICMPv6：新的ICMP附加消息類型，像 Packet Too big
IP層輔助協定
組播管理功能

並非所有路由器都可同時升級

- no "flag days"
- 混合 IPv4 和 IPv6 路由器的網路將如何運行？

tunneling：IPv6 作為有效載荷在 IPv4 跟 IPv4 之間傳送



A → B, E → F (flow: X, src: A, dest: F)

B → C IPv6 inside IPv4

IPv6：採用

Google 有 8% 的用戶同意 services via IPv6

USA 政府有 $\frac{1}{3}$ 支持 IPv6

SDN：軟體定義的網路的一種方法，會使用網路架構的控制

或 API (應用程式設計介面) 來引導網路流量。

新的拓撲架構由一個中央集權的控制器，來控制底下交換器的行為

分成 data plane 跟 control plane



去查 forwarding table、flow table
怎麼弄出來的

每一個 switch 是如何做到
封包進來，從那邊送出去

當中重要元件就是 forwarding table



SDN 裡稱為 flow table

Ch5. 章節目標：了解網路控制平面背後的原理

傳統路由的算法

SDN 控制器

互聯網控制消息協議 (ICMP)

網絡管理

在互聯網上的實例

OSPF、BGP、Openflow、ODL

OIVOS 控制器、ICMP

SNMP 等等

網路層的功能分成 data plane, control plane

data plane 做 forwarding；封包進來依照 forwarding table，從選定出口
出去

control plane 做 routing：用來決定從源到目的地所採用的路由

兩種 { per-router control (traditional) 每個路由器中的單獨路由算法組件在
SDN 控制平面中相互計算以轉發

Routing protocol goal：通過路由器網路確定從發送主機
到接收主機“好”的路徑

→ least (cost, fastest, least)

演算法可以跟資料結構做相關聯

全球的：

所有路由器都有完整拓撲，鏈連成本信息，叫 link state

去中心化：

路由器只知道連接到鄰居的成本；計算過程會與鄰居交換信息

叫 "distance vector"

link state

Dijkstra's algorithm: 可以算出無向圖中一點到某一點的最短路
distance vector 徑

Bellman-Ford equation: 同樣可找出最短路徑

LS 和 DV 算法比較

消息複雜度: DV 勝 LS (LS 有 n 個節點 · E 個鏈接 · 發送了 $O(nE)$ 個)
DV 僅在鄰居間交換，收斂時間不同

收斂速度: LS 勝 DV (LS $O(n^2)$ 算法需要 $O(nE)$ msgs, 可能有振盪)
DV 收斂時間不同，因路由環路數到 ∞

強健性: 如果路由
故障

LS 節點可以通告不正確鏈接成本
每個節點只計算自己的表

DV 節點可通告不正確鏈接

其他人使用每個節點的表

使路由可擴展 (對路由研究太理想化)

所有路由都相同， network "flat"， 實踐中不正確

scale 規模: 數十億目的地

不能將所有目的地存儲在路由表中

路由表交換會淹沒鏈接

administrative autonomy

互聯網 = 網路中各網路

每個管理員都希望自己控制路由

可擴展路由的 Internet 方法 \Rightarrow 將路由聚合到自治系統-As 的 area

As 中所有路由都要用 相同域內 協議

網關執行域間路由

不同As 的路由 可以 運行 不同域內 協議

2022/5/18 3個問題

1. Q: 請問路由演算法有哪兩大類，並說明優缺？

A: 有 link states 跟 distance vector

| | LS | DV |
|-------|-------------------------------------|---------------------------|
| 訊息複雜度 | 每個節點都必須知道 網路中所有連結的成本 | 每次循環時再直接相鄰 的節點之間進行訊息交換 |
| 收斂速度 | 實作 $O(N^2)$ ，比訊息 $O(NE)$ | 可能很慢，也可能產生繞送迴圈 |
| 強健性 | (節點可以通告不正確鏈接 成本)，每個節點只計算自己的 表 | 同左，() 但每個節點都會使用其他節點的表 |

2. Q: 請問網路層功能分成哪兩個 plane，並說明分別做什麼？

A: 分成 data plane 跟 control plane

data plane 做 forwarding \Rightarrow 封包進來依照 forwarding table，從選定出口出去
control plane 做 routing \Rightarrow 用來決定從源到目的地所採用的路由

3. Q: SDN 是什麼，能做什麼功能？

A: SDN (軟體定義的網路) 是一種網路方法

會使用網路架構的控制器或應用程式設計介面 (API) 來引導
網路流量，並與底層基礎架構進行通訊。