

1. Сравнете динамичните маршрутизиращи протоколи: "distance vector" и „link-state“.

Динамичните маршрутизиращи протоколи "distance vector" (протоколи на база на разстояния) и "link-state" (протоколи на база на състоянието на връзката) са две различни стратегии за определяне на маршрутите в мрежата. Ето някои от основните разлики между тях:

1. Информацията, използвана за вземане на решения:

Distance vector: Тези протоколи използват информация за разстоянието и посоката (като брой хопове или метрики) до различни дестинации. Изпращането на обновления на разстоянието към съседните рутери се извършва периодично.

Link-state: Тези протоколи използват информация за състоянието на връзката с всички други рутери в мрежата. Информацията се изпраща само при промяна в състоянието на връзката.

2. Изпращане на обновления:

Distance vector: Рутерите, използващи протоколи на база разстояние, изпращат само информацията за разстоянието до своите съседни. Това означава, че рутерите имат само локална представа за мрежата и разстоянието до другите дестинации се определя на основата на информацията, получена от съседите.

Link-state: Рутерите, използващи протоколи на база състояние на връзката, изпращат пълна информация за състоянието на връзката с всички рутери в мрежата. Това позволява на всеки рутер да има цялостна представа за мрежата и да пресмята най-кратките пътища до всички дестинации.

3. Обработка на промени в мрежата:

Distance vector: При промяна в мрежата, като например прекъсване на връзка или промяна в метриката, рутерите изпращат своите обновления на разстоянието към съседите си. Тези обновления се препращат от съседни на съседни, докато всички рутери се актуализират с новата информация.

Link-state: При промяна в мрежата, като например прекъсване на връзка или промяна в състоянието на връзката, само засегнатите рутери изпращат обновления на състоянието на

връзката към всички други рутери. Това намалява обема на преносена информация и ускорява сходимостта на мрежата.

4. Проблем на "конвергенция":

Distance vector: Протоколите на база разстояние могат да изпитат проблем на "конвергенция", когато информацията за разстоянието се разпространява по мрежата и се обновява. Това може да доведе до временни неправилни маршрути, преди конвергенцията да бъде постигната.

Link-state: Протоколите на база състояние на връзката имат по-бърза "конвергенция", тъй като всяка промяна в мрежата се разпространява само до съседните рутери, които актуализират своите бази данни за състоянието на връзката.

5. Скалируемост:

Distance vector: Протоколите на база разстояние са по-малко скалируеми от протоколите на база състояние на връзката. Това се дължи на факта, че рутерите изпращат своите обновления на разстоянието към съседите си периодически, независимо от това дали има промяна в мрежата или не. При големи мрежи това може да доведе до излишен трафик и натоварване на мрежата.

Link-state: Протоколите на база състояние на връзката са по-скалируеми, тъй като само промените в мрежата се разпространяват до съседните рутери. Това намалява излишния трафик и позволява по-ефективно управление на големи мрежи.

6. Използване на ресурси:

Distance vector: Протоколите на база разстояние изискват по-малко изчислителни ресурси и памет в сравнение с протоколите на база състояние на връзката. Това ги прави по-подходящи за по-малки мрежи или устройства с по-ограничени ресурси.

Link-state: Протоколите на база състояние на връзката изискват по-големи изчислителни ресурси и памет, тъй като рутерите трябва да поддържат и обработват пълна информация за състоянието на връзката в мрежата. Това ги прави по-подходящи за по-големи и по-сложни мрежови среди.

И двата типа протоколи имат своите предимства и недостатъци, и изборът между тях зависи от специфичните изисквания и характеристики на мрежата.

2. Опишете стъпките за процеса на договаряне при транспортния протокол TCP.

Процесът на договаряне при транспортния протокол TCP (Transmission Control Protocol) се нарича "TCP handshake" и се извършва между два устройства (клиент и сървър), за да се установи стабилна връзка между тях. Ето стъпките на процеса на договаряне при TCP:

Стъпка 1: Установяване на връзка (SYN)

Клиентът инициира процеса на договаряне като изпраща TCP сегмент, наречен "SYN" (символ на синхронизация), до сървъра.

Сегментът съдържа номер на порт, от който клиентът иска да комуникира, и избран случаен номер на последователността, който се генерира от клиента.

Стъпка 2: Приемане на връзка (SYN-ACK)

Сървърът получава сегмента SYN от клиента и отговаря със сегмент "SYN-ACK" (символ на синхронизация и приемане на синхронизация).

Сегментът SYN-ACK включва потвърждение на случайния номер на клиента и собствен случаен номер на последователността, генериран от сървъра.

Стъпка 3: Потвърждение на връзка (ACK)

Клиентът получава сегмента SYN-ACK от сървъра и изпраща сегмент "ACK" (потвърждение) за потвърждение на свързването.

Сегментът ACK включва потвърждение на случайните номера на последователността от клиента и сървъра.

След успешното изпълнение на тези три стъпки, връзката се счита за установена и клиентът и сървърът могат да започнат обмен на данни. В случай на неуспешно договаряне или възникване на проблем по време на процеса, стъпките на договаряне се повтарят или се генерира грешка, което води до неуспешна връзка. Процесът на TCP handshake осигурява надеждна и сигурна връзка между две устройства, които използват TCP протокола за комуникация.

3. Посочете транспортния протокол и портовете по подразбиране (well-known ports) на следните услуги:

SMTP (Simple Mail Transfer Protocol) - TCP порт 25

SNMP (Simple Network Management Protocol) - UDP порт 161

DNS (Domain Name System) - UDP порт 53

DHCP (Dynamic Host Configuration Protocol) - UDP порт 67 (сървър) и 68 (клиент)

HTTP (Hypertext Transfer Protocol) - TCP порт 80

HTTPS (Hypertext Transfer Protocol Secure) - TCP порт 443

FTP (File Transfer Protocol) - TCP порт 21

SSH (Secure Shell) - TCP порт 22

POP3 (Post Office Protocol version 3) - TCP порт 110

IMAP (Internet Message Access Protocol) - TCP порт 143

TELNET (Teletype Network Protocol)- TCP порт 23