



# T04- Camada rede



Subjects

RC

## Serviços da camada de rede

- Transporte de datagramas entre computadores
- Os protocolos desta camada correm nos nós terminais e nos routers
- Os routers examinam os cabeçalhos IP de todos os datagramas

## Duas funções chave

- Routing → Determine o caminho origem-destino , Ação global
- Forwarding → Mover pacotes que chega no router para a saída apropriada; Ação local no router
- Analogia:
  - O routing e planeamento de uma viagem
  - O forwarding e a passar um cruzamento

## Routing vs forwarding

algoritmo de routing → Determina a rota através da rede

tabela forwarding local → Determina como o pacote deve ser encaminhado neste router

## Plano de dados e Plano de controlo

Plano de dados:

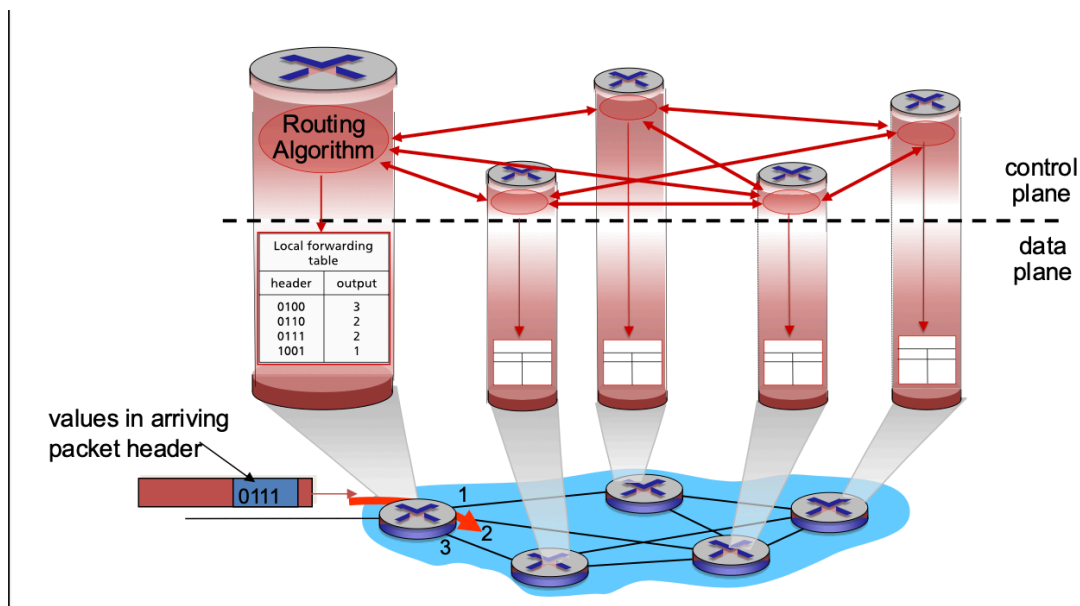
Plano de controlo:

- função local em cada router
- determina o caminho de datagrama e a saída
- encaminhamento
- logica global
- routing e definido aqui
- duas abordagens:
  - *tradicional* : algoritmos de routing implementados nos routers
  - redes definidas por software : implementado em servidores remotos

## Plano de controlo tradicional

### Algoritmo distribuido de routing

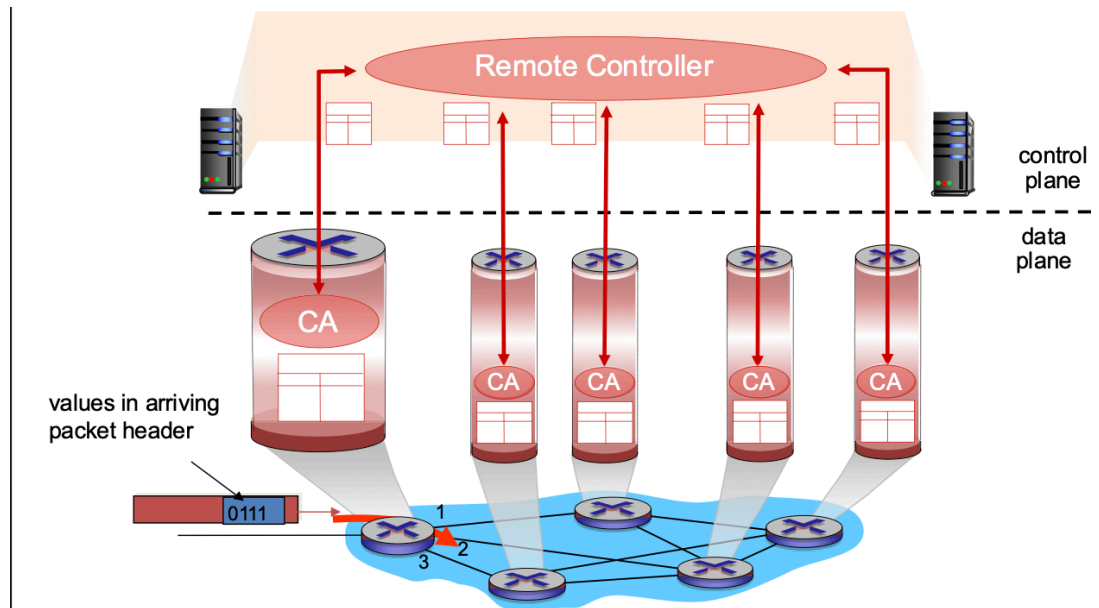
Componentes individuais do algoritmo de routing correm em cada router, e todos interagem entre si



## Plano de controlo definido por software

Logicamente centralizado

Um controlador remote interage com agentes locais em cada router



## Plano de dados

### Anatomia de um router

Um router tem 2 características : muitas placas de rede e comutar pacotes

Função chaves: correr protocolos de routing e forwarding datagramas da entrada para a saída

### Porta de entrada

Data um endereço destino do datagram, vai fazer um lookup na tabela de encaminhamento

objetivo: processamento a velocidade da linha (daca linia este de 10Gbps , procesul trebuie sa fie de minim 10Gbps pentru nu a incetini pachetele)

Se o datagrama chega mais rapido do que a velocidade do modulo de switching

## Tabela de encaminhamento

128.16.16.0 → 128.16.23.255 = interface de saída 0

128.16.24.0 → 128.16.24.255 = interface de saída 1

128.16.24.0 → 128.16.31.255 = interface de saída 2

caso contrario = interface de saída 3

Gamas de endereço destino	Interface saída
10000000 00010000 00010 <b>000 00000000</b> → 10000000 00010000 00010 <b>111 11111111</b>	0
10000000 00010000 00011000 <b>00000000</b> → <b>10000000 00010000 00011000 11111111</b>	1
10000000 00010000 00011 <b>000 00000000</b> → 10000000 00010000 00011 <b>111 11111111</b>	2
Caso contrario	3

## Module de comutação

Transfere pacote do buffer de entrada para buffer de saída

Taxa de comutação : Taxa do pacote que esta enviado da entrada ate a saída

Temos 3 tipos : memoria , bus , crossbar

## Switching atraves da memoria

Computador faz a comutação sob controlo direto do CPU

Os pacotes eram copiadas para a memoria do sistema

Velocidade era limitada pela banda de memoria

## Switching atraves de bus

Datagrama e enviada da entrada para saída atraves de um bus partilhado

a taxa e limitada pela largura de banda do bus

## Portas de saída

Necessário buffering para quando o datagrama chega mais rápido do que velocidade da linha

Algoritmo de scheduling é usado para escolher qual o próximo pacote transmitir

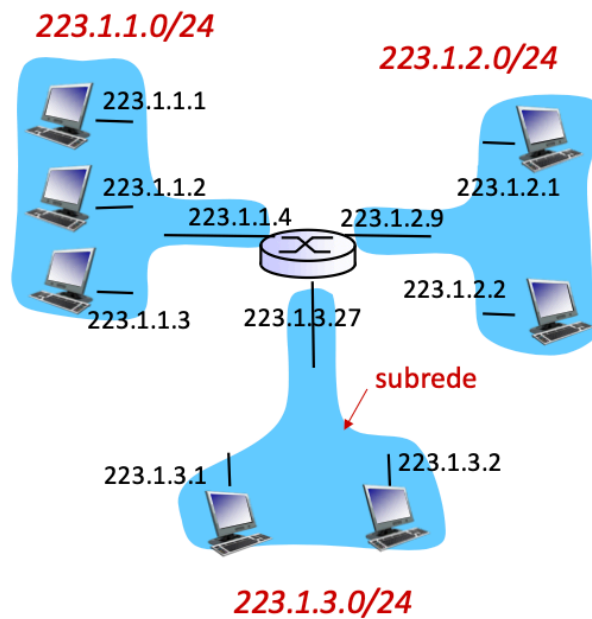
Aqui se pode perder pacotes

## Endereçamento IP

- As ligações têm um MTU (Maximum Transmission Unit) → o tamanho máximo que um datagrama pode ter
- Um pacote IP de tamanho grande pode ser dividido na rede e depois remontado no destino
- O endereço IPv4 tem 32 bits
- Interface: Ligação entre routers, computadores e uma ligação física
  - Os routers têm múltiplas interfaces mas o computador tem 1 ou 2

## Subredes

- Endereço IP
  - bits mais significativo : subrede
  - bits menos significativo: computador/router
- O que é um subrede:
  - Conjunto de dispositivos que pode falar entre si sem necessitar de um router



Máscara da subrede: /24

- CIDR(Classless InterDomain Routing)
  - A porção do endereço que identifica a rede tem tamanho arbitrario
  - formato: a.b.c.d/x , onde x e o numero de bits da porção do endereço que identifica a subrede

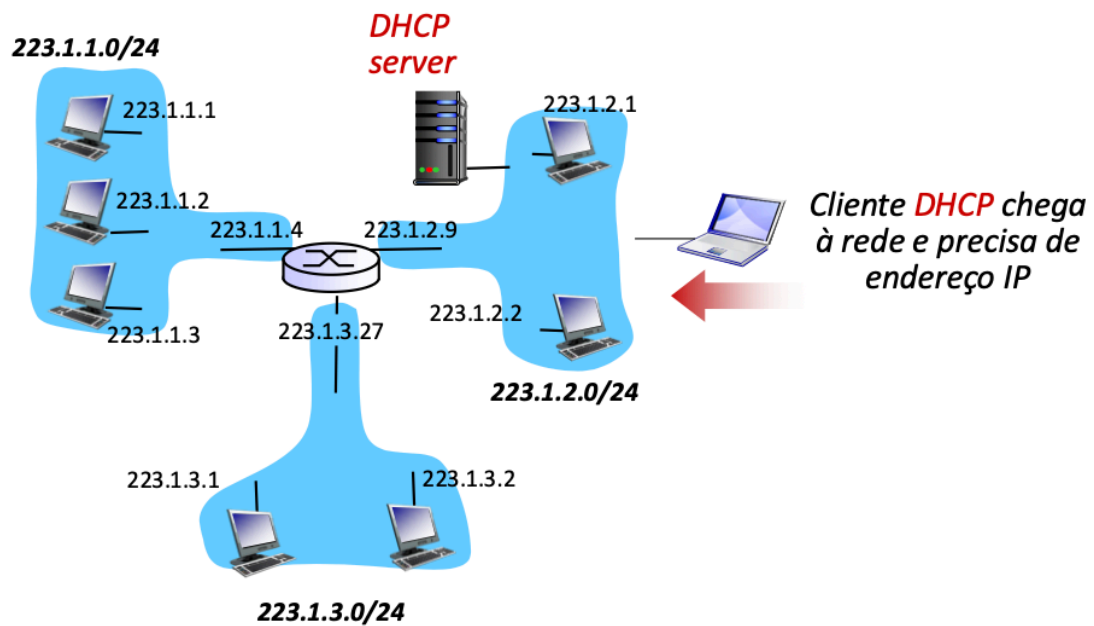
O computador recebe o seu IP em 2 maneiras :

- Manualmente → configurado pelo admin
- Por DHCP(Dynamic Host Configuration Protocol) → permite obter endereço IP automaticamente

DHCP :

- Durante un tempo de reserva , renovavel

- Permite reutilização de endereços
- Utilizadores moveis podem entrar na rede facilmente

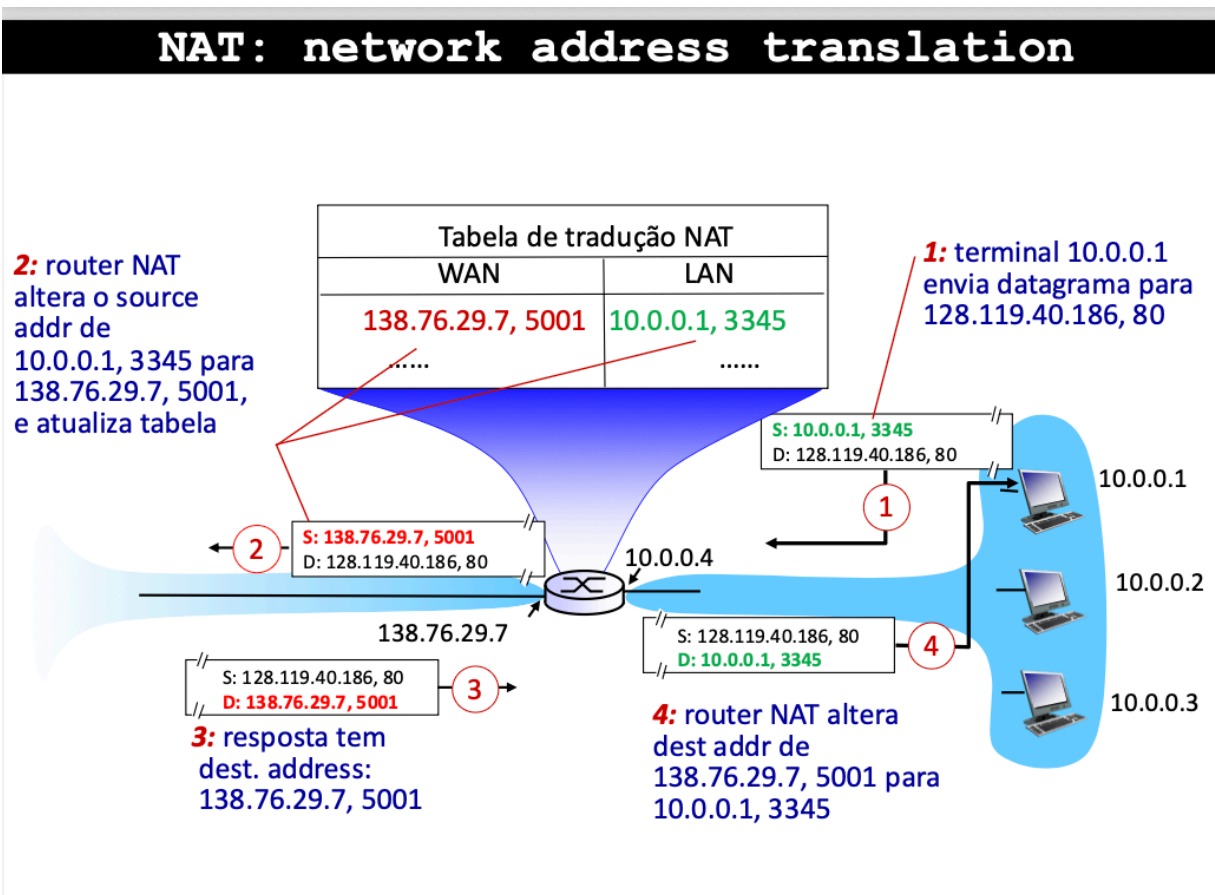


Como funciona o protocolo:

1. O computador faz um broadcast com a mensagem "DHCP discover"
2. O servidor DHCP manda uma mensagem "DHCP offer" com um IP
3. Faz pedido de endereço com "DHCP request"
4. O servidor DHCP confirma com "DHCP ack"

## NAT ; ICMP ; IPv6

- Quando o datagrama sai da rede local , ela vai pegar o IP do router NAT com port diferente
- Vantagens:
  - E mais seguro
  - Os dispositivos da rede local não são visíveis de fora



- Como resolver a problema de NAT



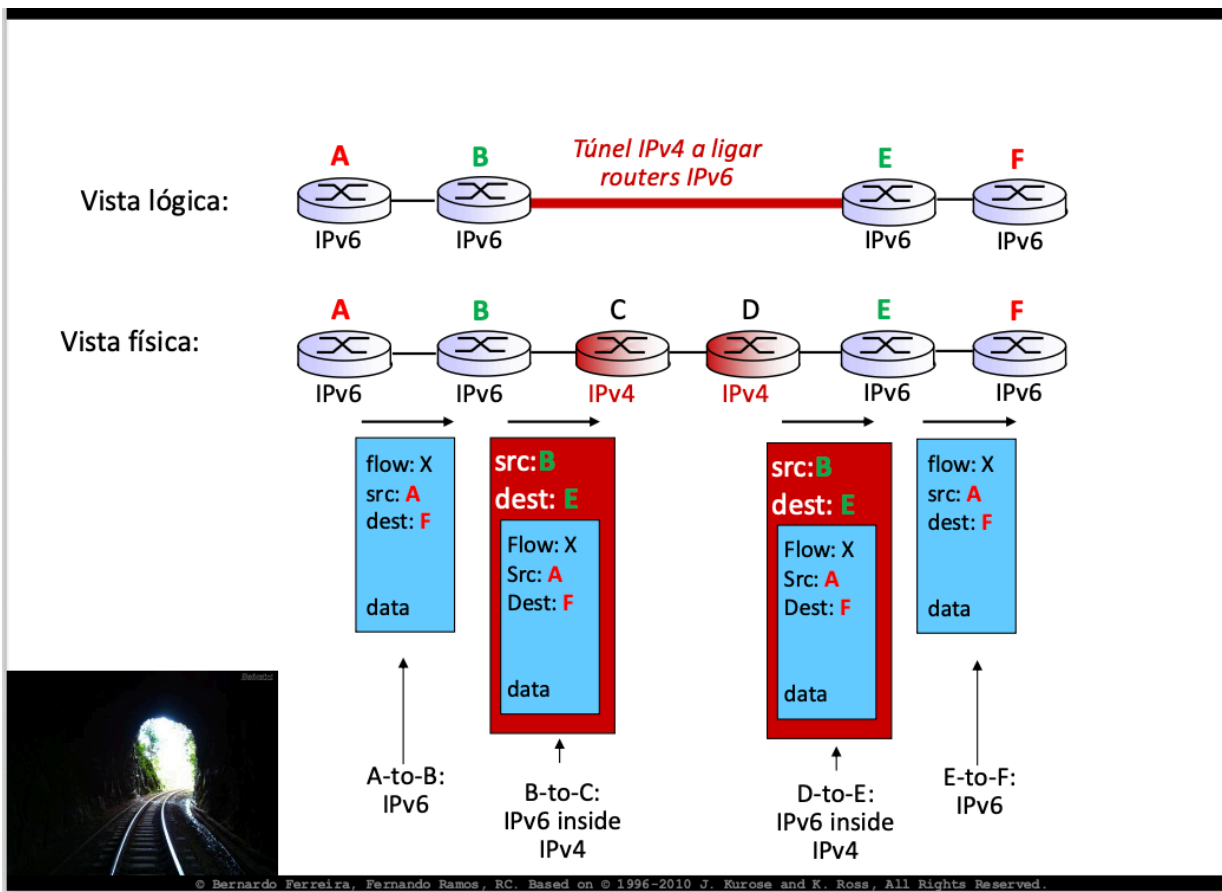
- Usar um dispositivo Universal Plug and Play e protocolo Internet Gateway Device
- Permite que um terminal estar atrás de NAT

## **ICMP : Internat Control Message Protocol**

- Usado pelo computador e routers para trocar informações com a camada de rede (Errors , Echos → ping , requests ,reply)
- Structura : Tipo + Codigo + primeiros 8 bits da datagrama

## **IPv6**

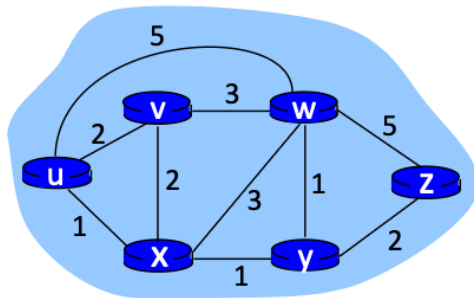
- E um upgrade comparado com IPv4 porque o IPv4 esta a acabar
- O cabeçalho do IPv6 permite aumentar a velocidade de processamento e de forwarding
- O cabeçalho tem 40 bits fixo
- Transição do IPv4 para IPv6
  - Usamos pilhas protocolares duplas (os nos podem enviar IPv4 e IPv6)
  - Usando tuneis
    - Entre routers IPv4 os datagramas IPv6 são enviadas como carga (payload) de uma datagrama IPv4



## Plano de controlo

## Algoritmos de routing

## Grafo da rede: custos



$c(x, x') =$  custo da ligação  $(x, x')$   
e.g.,  $c(w, z) = 5$

Custo pode ser sempre 1, ou então ser inversamente relacionado com largura de banda ou congestão, custo monetário, etc.

Custo do percurso  $(x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$

**Questão chave:** qual é o percurso de **custo mínimo** entre u e z?

**Algoritmo de routing:** algoritmo que encontra o percurso de **custo mínimo**.

## Classificação de algoritmos de routing

Informação global ou parcial

global	parcial
Routers tem noção da topologia completa da rede	Routers conhece os nos vizinhos
Routers distribuem informação local com todos	Routers distribuem a informação global apenas com os vizinhos
Exemplos: algoritmo "link state"	Exemplo: algoritmo "distance vector"

Algoritmos estaticos ou dinamicos

Estaticos	Dinamicos
Rotas mudam muito lentamento	Rotas mudam frequentemente
Tabela de encaminhamento pode ser configurada manualmente	São feitas atualizações periodicas em resposta a mudanças

## Algoritmo Estado de Ligação

- Exemplo: algoritmo de Dijkstra
- Características:
- A topologia da rede e os custos da ligação são conhecidos por todos os nós (atraves de "link state broadcast")
- Cada nó calcula os percursos de custo minimo entre si e todos os outros (para construir a tabela de encaminhamento)
- Recalcula periodicamente ou quando alguma ligação sofre alteração

## Algoritmo de Dijkstra

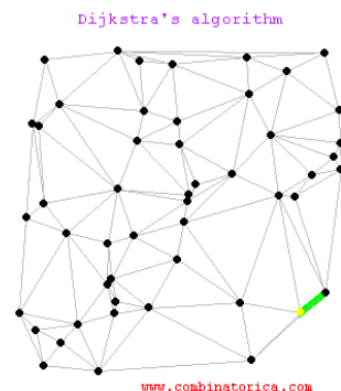
### #0 Entrada

- nó\_a\_testar: TODOS
- nó\_resolvido: NENHUM

### #1 Encontrar o nó ao qual eu chego com custo mínimo

### #2 Adicionar esse nó a nó\_resolvido e remover de nó\_a\_testar.

### #3 Repetir #1 e #2 até nó\_a\_testar estar vazio



## Distance vector algorithm

- $D_x(y)$  = Estimativa do custo minimo entre x e y
- O nó x :
  - Conhece o custo para cada vizinho
  - Guarda os vetores distancia dos vizinhos
- Ideia chave :
  - cada nó envia a sua estimativa de vetor distancia aos vizinhos , periodicamente
  - Quando recebe um novo vetor distancia , atualiza o seu usando a equação de B-F
- Algoritmo iterativo sendo cada iteração local causada por:
  - Mudança de custo das ligações locais
  - Receção de um novo vetor distância de um vizinho
- Algoritmo distribuido :
  - Cada nó apenas notifica os seus vizinhos quando o seu vetor distância sofre alteração

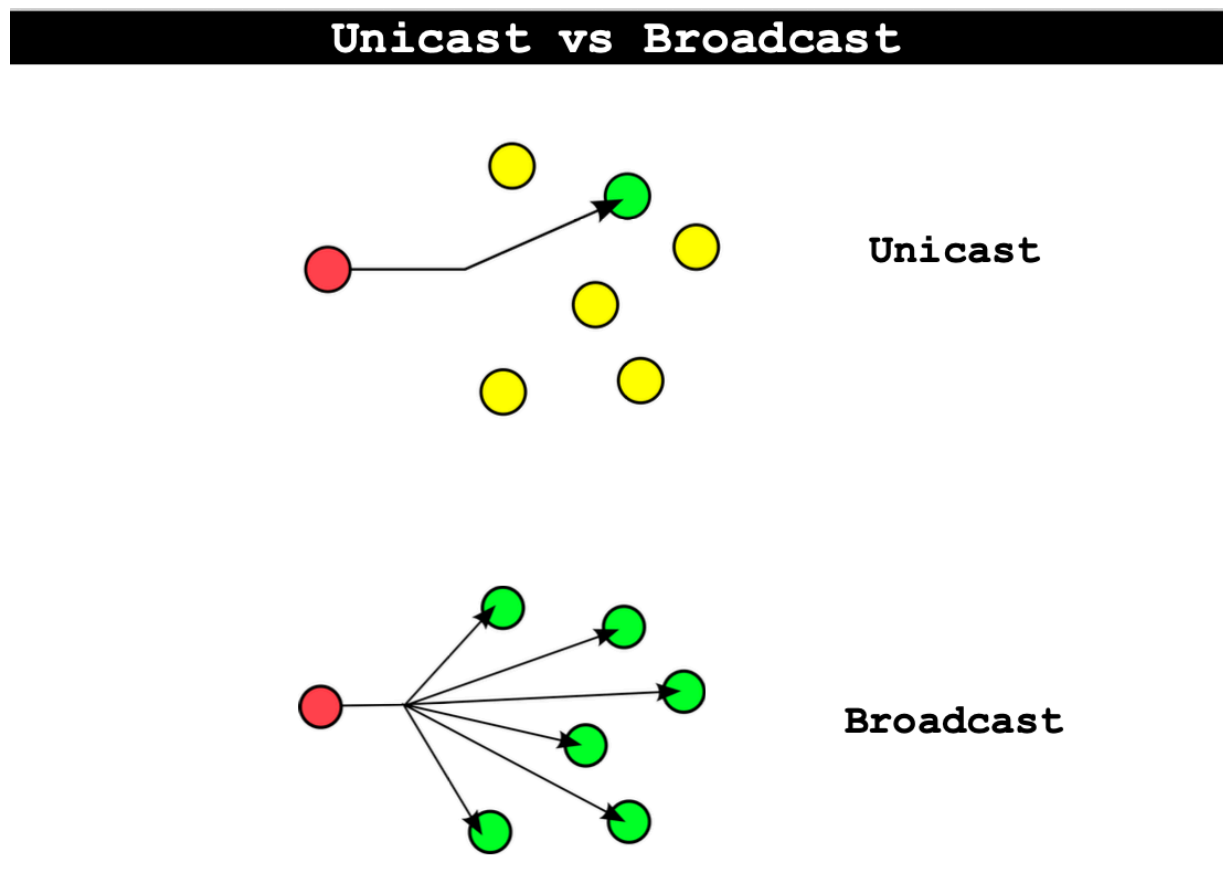
## Routing hierarquico

- O routing na Internet e hierarquico com as routers agregados em regioes os sistemas autonomos (AS)
- Routers de mesmo AS correm o mesmo protocolo de routing "intra-AS"
- Como ligar as diferentes redes?
  - Routing gateway na edge do seu AS tem ligações com routers de outros AS
  - Correm protocolo de routing "inter-AS"
  - No caso da Internet este protocolo e o BGP(Border Gateway Protocol)

## Interligação entre Ases

- A tabela de encaminhamento é configurada pelos dois protocolos : intra-AS e inter-AS
  - O intra-AS define as entradas relacionadas com os destinos internos
  - O inter-AS define as entradas relacionadas com os destinos externos

## Broadcast e multicast



## Routing Broadcast

- uma possibilidade: enviar os pacotes do emissor para todos os outros nós

- Muito ineficiente

## **Tecnicas de duplicação na rede**

- Flooding: quando um no recebe um pacote broadcast, envia a copia para todos os vizinhos
  - problema: criação de ciclos e broadcast storms
- Flooding controlado: o no faz broadcast do pacote apenas se não o tiver feito anteriormente
- Spanning tree: permite que nenhum no receba pacotes repetidos

## **Spanning tree**

- Spanning tree → un arvore a qual pertencem todos os nos da rede
- Os nos depois encaminham os pacotes apenas pela spanning tree

## **Routing multicast**

Multicast → envia pacotes para os nos que fazem parte da sua rede

Broadcast → envia pacotes para todos nos

## **Como unir "ilhas" multicast**

- Para ligar as "ilhas" temos de usar os tuneis
- Encapsulando o pacote multicast dentro de um pacote "normal"
- Esse pacote lp"normal" e enviado atraves de um tunel usando unicast IP para o proximo router que usa multicast
- O router multicast que recebe esta datagrama "normal" desencapsula e obtem o pacote multicast

## Routing multicast:problema

Arvores podem ser partilhadas ou podem ser baseadas no emissor

- Abordagens
  - Arvores baseadas no emissor
    - shortest path trees
    - reverse path forwarding (with pruning)
  - Arvores partilhadas:
    - minimal Steiner tree
    - center -base trees

## Shortest path tree

- A arvore dos caminhos mais curtos entre o emissor e todos os destinatarios
- Problema: os destinatarios teriam de ter a visão da rede do ponto de vista do emissor , e conhecer todos os outros destinatarios

## Reverse path forwarding with pruning

- Inicialmente igual ao Broadcast
- Mas a arvore construida pode ter sub-arvores sem membros
  - E ineficiente encaminhar pacotes para este nos
  - Solução: os routers sem membros enviam mensagens "prune" em direção ao emissor ate chegar a um router com membros

Exemplu:

Desigur! Hai să explicăm **Reverse Path Forwarding cu Pruning** în termeni simpli, ca și cum ai povesti cuiva care nu e expert în rețele:



**Ce e Reverse Path Forwarding (RPF)?**



Imaginează-ți că vrei să trimiți un anunț la toți vecinii tăi, dar nu vrei să-l trimiți de mai multe ori la aceeași casă.

RPF este o metodă folosită pentru a trimite pachete **multicast** (adică de la o sursă la mai mulți destinatari), fără să le trimită în cerc (fără loop-uri).

🔄 **"Reverse"** se referă la faptul că routerul verifică dacă pachetul a venit **pe același drum** pe care l-ar folosi el ca să ajungă înapoi la sursă. Dacă da, păstrează pachetul și îl trimite mai departe. Dacă nu, îl aruncă (ca să evite dubluri sau bucle).

---

## Ce înseamnă „Pruning”?

Gândește-te că ai trimis anunțul către toate casele, dar unele îți spun:

“Hei, la noi nu locuiește nimeni care vrea acest anunț, nu mai trimite aici!”

Asta e **pruning** – o curățare.

Routerele care nu au destinatari pentru acel flux multicast **trimit un semnal înapoi**, spunând:

„Te rog, nu-mi mai trimite pachetele astea”.

Astfel, fluxul de date nu mai trece prin ele inutil – ceea ce salvează **bandă și resurse**.

---

## Cum funcționează împreună?

1. La început, pachetele multicast se trimit cam peste tot (flooding controlat cu RPF).
  2. Routerele verifică dacă pachetul a venit pe drumul corect spre sursă (RPF).
  3. Dacă routerul nu are clienți interesați de acel flux, trimite un mesaj de „prune” în sus.
  4. După ce se stabilizează, doar routerele care au nevoie de pachet îl mai primesc.
-

## Un exemplu:

- Sursa: un server video live.
  - Destinatari: doar 3 din 10 calculatoare din rețea vor să vadă.
  - La început, video-ul se trimite peste tot.
  - 7 calculatoare spun "nu ne interesează" (prune).
  - Pachetele ajung doar la cele 3 care vor – eficiență!
- 

## Arvorea partilhas: steiner tree

- Steiner tree: a arvora de custo minimo que liga todos os routers que tem membros
- Solução ótima, mas não existe nenhum algoritmo eficiente que permita chegar a esse ótimo

## Center-based tree

- Também uma única árvore partilhada por todos
- Um dos nós é identificado como sendo o centro ou a raiz da árvore
- Para o nó se juntar à árvore, o router envia mensagem unicast join para o router central

## Ce e Center-Based Tree?

Este o metodă de trimiteră a datelor **multicast**, dar în loc să le trimiți peste tot (ca la început în RPF), alegi **un punct central** numit **rendezvous point (RP)**, adică un **punct de întâlnire**.

Toți cei care vor datele (abonatul A, B, C...) vin și se conectează la acest RP.

## Cum funcționează:

1. **Sursa** trimite datele către **RP (centrul)**.

2. **Destinatarii** (cei care vor pachetul) se conectează și ei la RP.
3. Datele merg **de la sursă → RP → destinatari**, printr-un **arbore (tree)** care pleacă din RP.
4. Arborele are ramuri doar către nodurile care chiar au nevoie de date – foarte eficient!