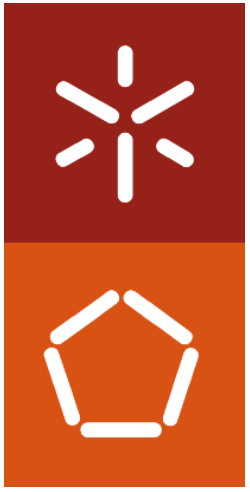
Redes Móveis Adhoc

Novos Paradigmas de Redes

2023/2024

2°semestre



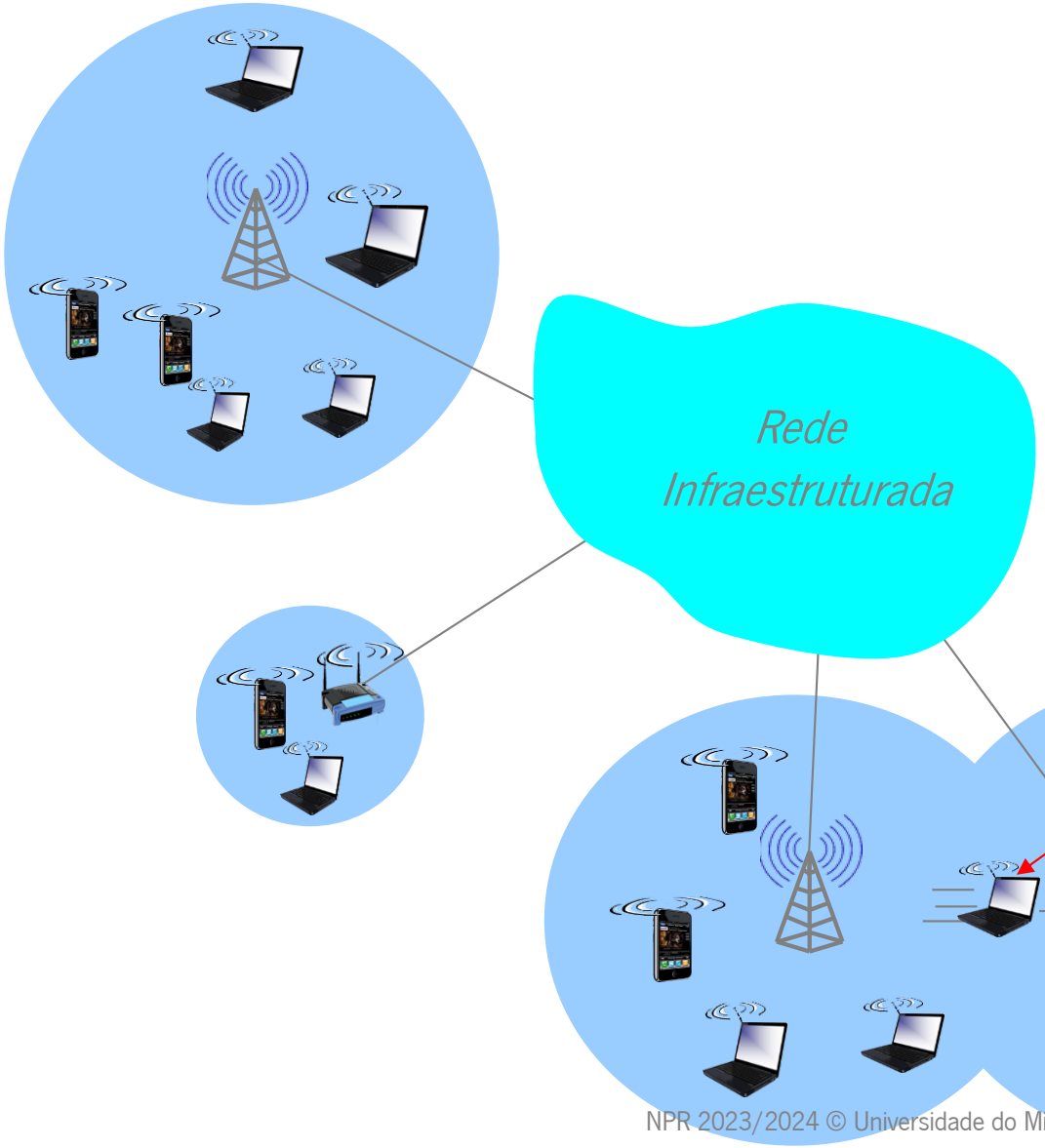
Redes Móveis Sem Fios

- **Porquê?**
  - Número de assinantes de telefones móveis excede largamente o número de assinantes de telefones fixos
  - Redes de computadores com cada vez mais dispositivos móveis: laptops, palmtops, PDAs, telefones VOIP
  - **Surgimento de contextos em que a mobilidade é intrínseca (por exemplo, as redes veiculares)**
- **Dois desafios importantes mas diferentes**
  - *Comunicações sem fios :* comunicação através de ligações sem fios
  - *Mobilidade:* gestão do utilizador móvel que muda constantemente o seu ponto de ligação à rede

Redes Móveis Sem Fios

- **A mobilidade dos nós da rede constituiu um desafio já que não estava prevista na arquitetura original da Internet**
  - Endereçamento e encaminhamento do tráfego na Internet são efetuados com base em endereços IP, que servem por isso, não só para identificar o destino, mas também para o localizar. Quando um nó se move, muda de localização e por isso TEM que mudar de endereço!
  - Os nós móveis estão normalmente associados a redes sem fios, onde o protocolo TCP apresenta um mau desempenho graças ao modelo de controlo de congestionamento que é utilizado

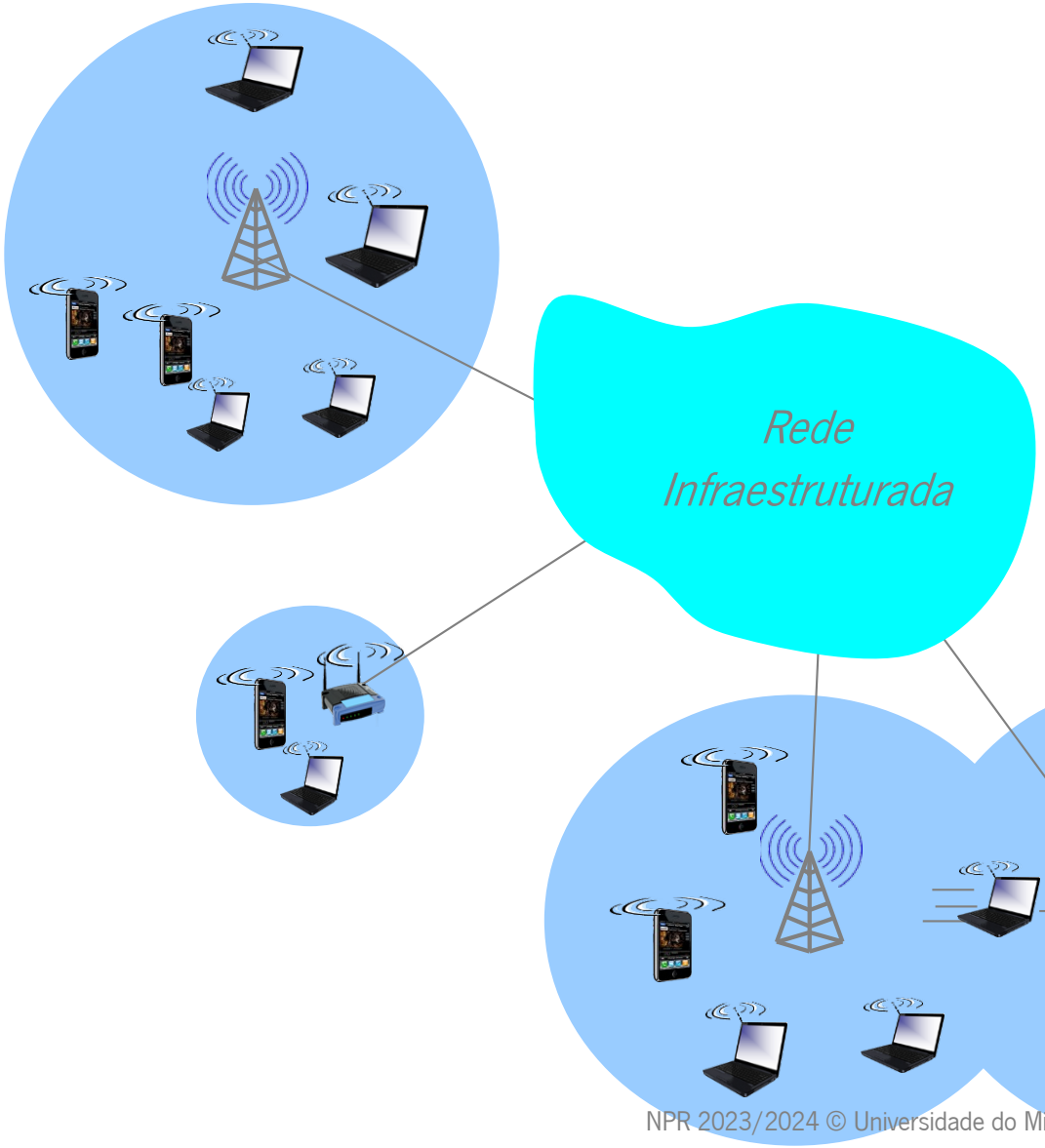
Elementos de uma rede sem fios



Hosts sem fios

- *Portáteis , Tablets, Telemóveis*
- *Executam Aplicações*
- *Podem ser estacionários ou móveis*
  - *Sem fios não significa sempre móveis*

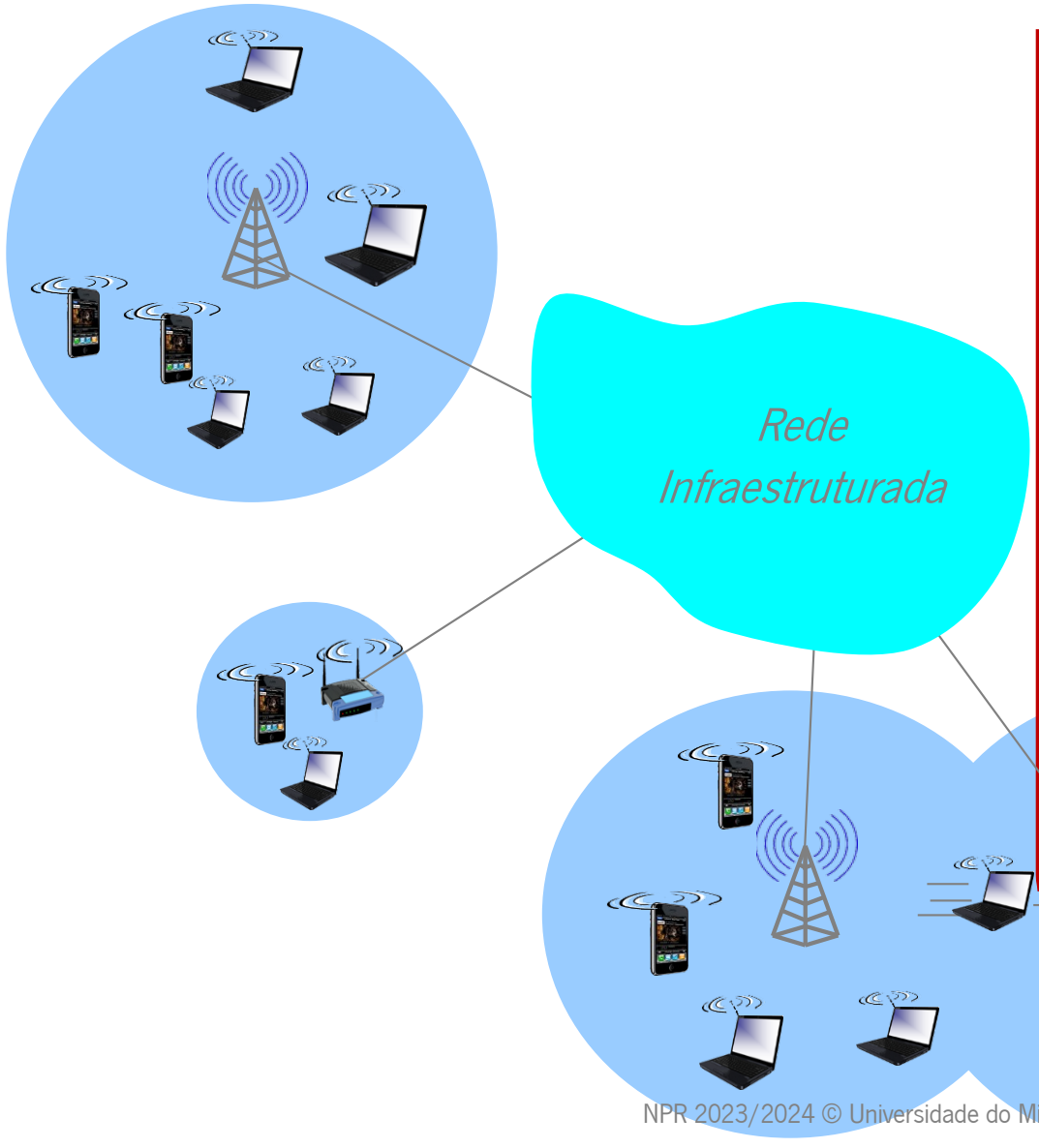
Elementos de uma rede sem fios



Estação Base

- *Tipicamente ligada à rede infraestruturada*
- *Atua como "relay" – responsável por transmitir os pacotes entre a rede infra-estruturada e a rede sem fios. Exemplos: pontos de acesso Wifi, torres celulares.*

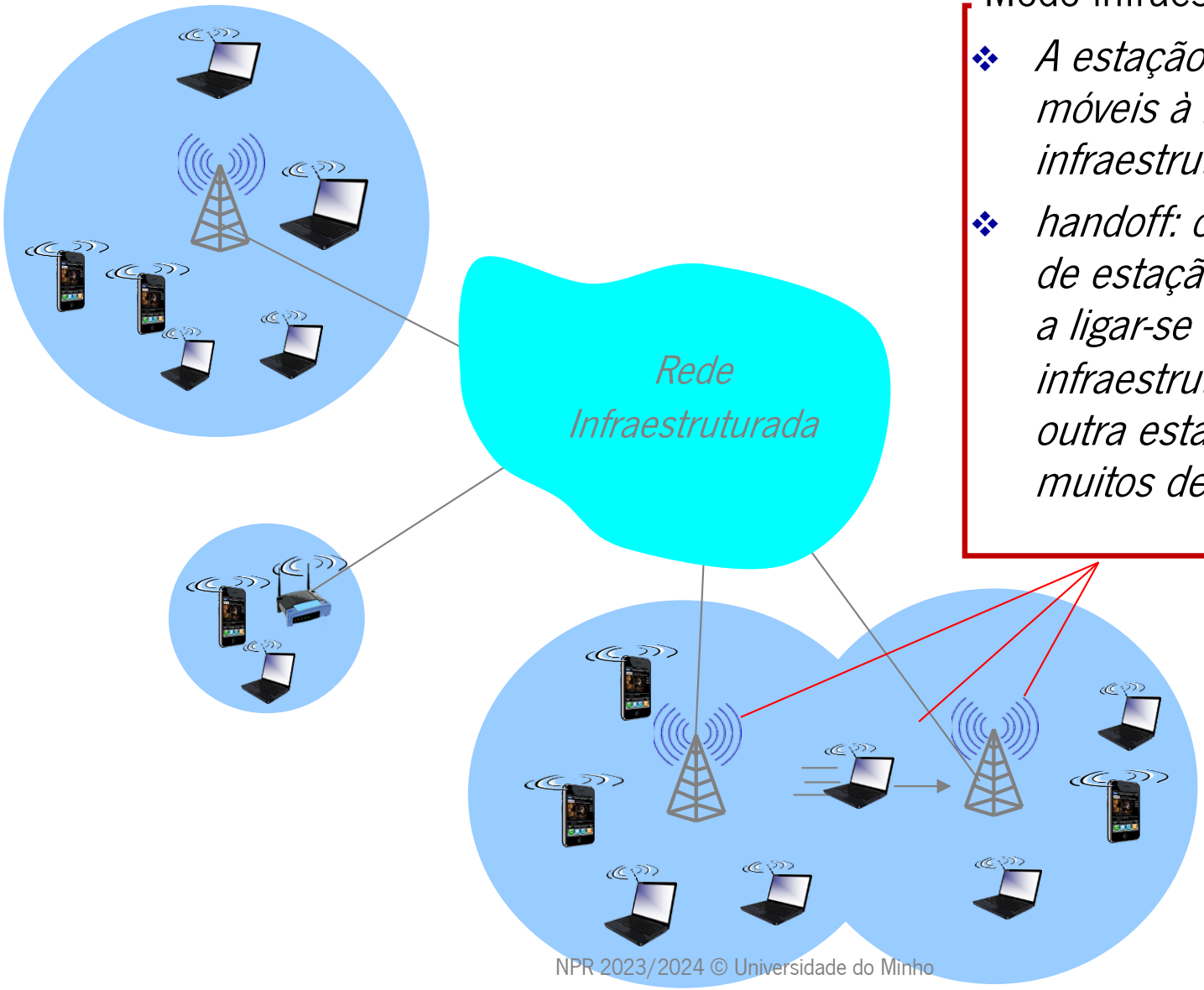
Elementos de uma rede sem fios



Ligação Sem Fios

- *Tipicamente usada para ligar os nós móveis à Estação Base*
- *Também pode ser usada como ligação de backbone*
- *Protocolo de Acesso múltiplo para coordenar o acesso ao meio físico de transmissão*
- *Diferentes tecnologias com diferentes taxas de transmissão e distâncias*

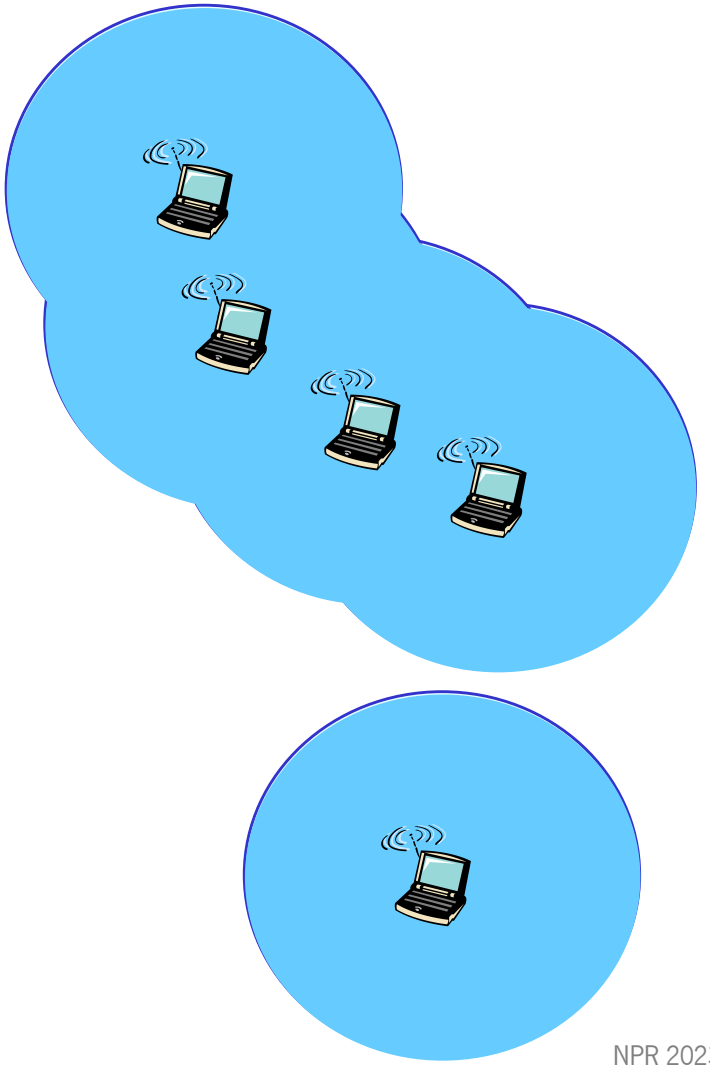
Elementos de uma rede sem fios



Modo infraestrutura

- *A estação base liga os nós móveis à rede infraestruturada*
- *handoff: o nó móvel muda de estação base passando a ligar-se à rede infraestruturada através de outra estação base (levanta muitos desafios)*

Elementos de uma rede sem fios



Modo Ad-Hoc

- *Sem estações Base*
- *Os nós podem apenas transmitir pacotes para outros nós que estejam dentro da sua área da cobertura. nodes*
- *Os nós auto-organizam-se numa rede fazendo o encaminhamento de tráfego entre eles.*

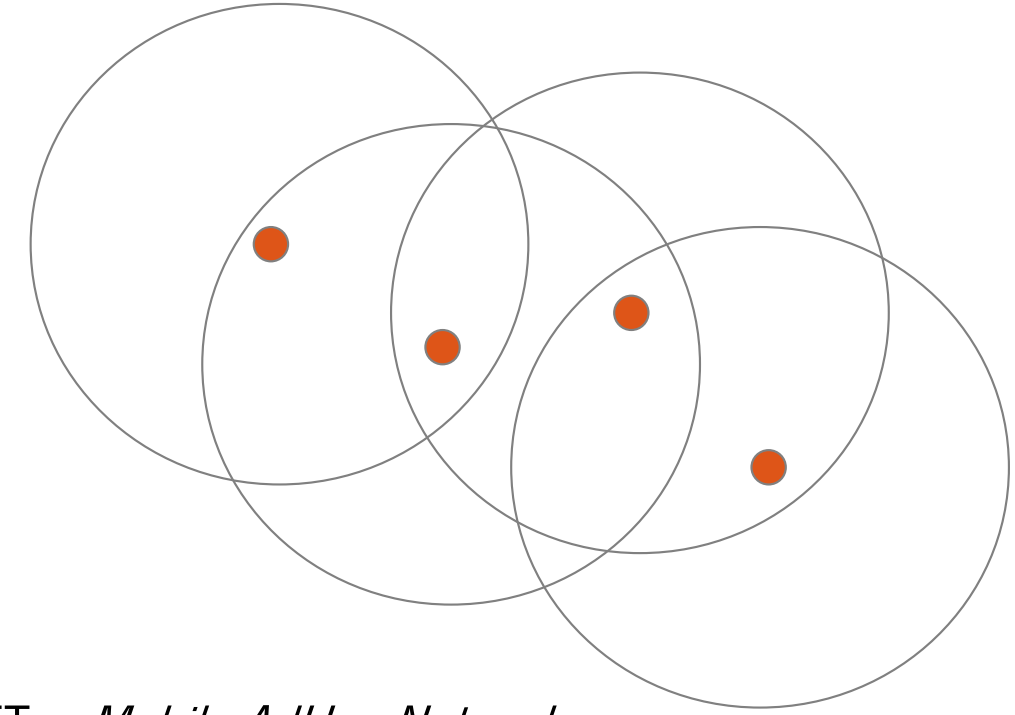
Taxonomia de Redes sem Fios

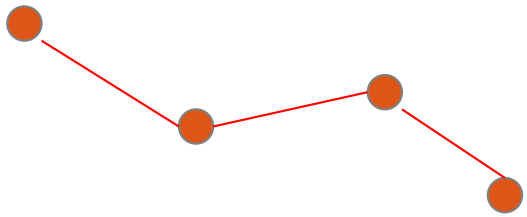
	Salto único	*Saltos múltiplos*
Infraestrutura (i.e., APs)	O host liga-se à estação de base (WiFi,WiMAX, celular) que se liga à Internet	O host pode ter que passar por vários nós relay sem fios para se ligar à Internet: mesh net
Sem Infraestrutura	Sem estação de base, nem ligação à Internet (Bluetooth, redes adhoc)	Sem estação de base, sem ligação à Internet. Pode ter que usar relays para atingir um dado nó sem fios na MANET, VANET

Redes móveis Ad-Hoc

- **Em que cenários?**
  - Onde não há infraestruturas...
  - Onde as redes celulares (3G, LTE, etc) e redes de acesso (WiFi) não são atrativas (custos, etc) ou estão sobrecarregadas
- **Que aplicações?**
  - Cenários de catástrofe e militares...
  - Conferência/trabalho/diversão em grupo de formação espontânea em qualquer lugar...
  - Partilha de informação em eventos públicos localizados
  - Rede pessoal de dispositivos locais
- **Como?**
  - Cada dispositivo é um nó da rede que participa no encaminhamento... enviando/recebendo dos vizinhos ao seu alcance
  - Neste cenário todos são routers... E estes "routers" movem-se!

Redes móveis AdHoc





Source: it644 course material
Prof. Sridhar Iyer

- MANET – *Mobile AdHoc Network*
- Múltiplos dispositivos móveis, com tecnologia de comunicação sem fios heterogénea e limitada no alcance de transmissão
- Encaminhamento pode permitir comunicação em mais que um salto
- Desafios: bateria, obstáculos, perdas por erros de transmissão, mobilidade dos dispositivos

Principais desafios

- **Arquitetura de encaminhamento plana ou hierárquica**
  - Numa arquitetura plana o endereço de um nó está visível para todos os outros pela informação de encaminhamento → problema de escalabilidade
  - Numa organização hierárquica reduz-se a sobrecarga de informação → implica uma organização em aglomerados disjuntos (clusters) com um líder de cluster que conhece os outros membros
- **Ligações unidirecionais ou bidirecionais**
  - Não se pode assumir que os links são bidirecionais:
    - Diferentes potências de transmissão e sensibilidade de receção dos dispositivos
    - Interferência: um dispositivo pode estar mais exposto a interferências que o impedem de receber mas podem permitir o envio
    - Modo "silêncio" imposto ao dispositivo não o impede de receber
    - etc...

Principais desafios

- **Utilização de "Super-Dispositivos"**
  - Assume-se normalmente que todos os dispositivos possuem as mesmas características mas isso pode não ser verdade...
  - Nós com mais bateria ou maior capacidade de CPU/memória ou de transmissão podem constituir-se como Super-Dispositivos numa rede de backbone...
- **Encaminhamento com QoS**
- **Encaminhamento Multicast**

Classificação dos protocolos de encaminhamento

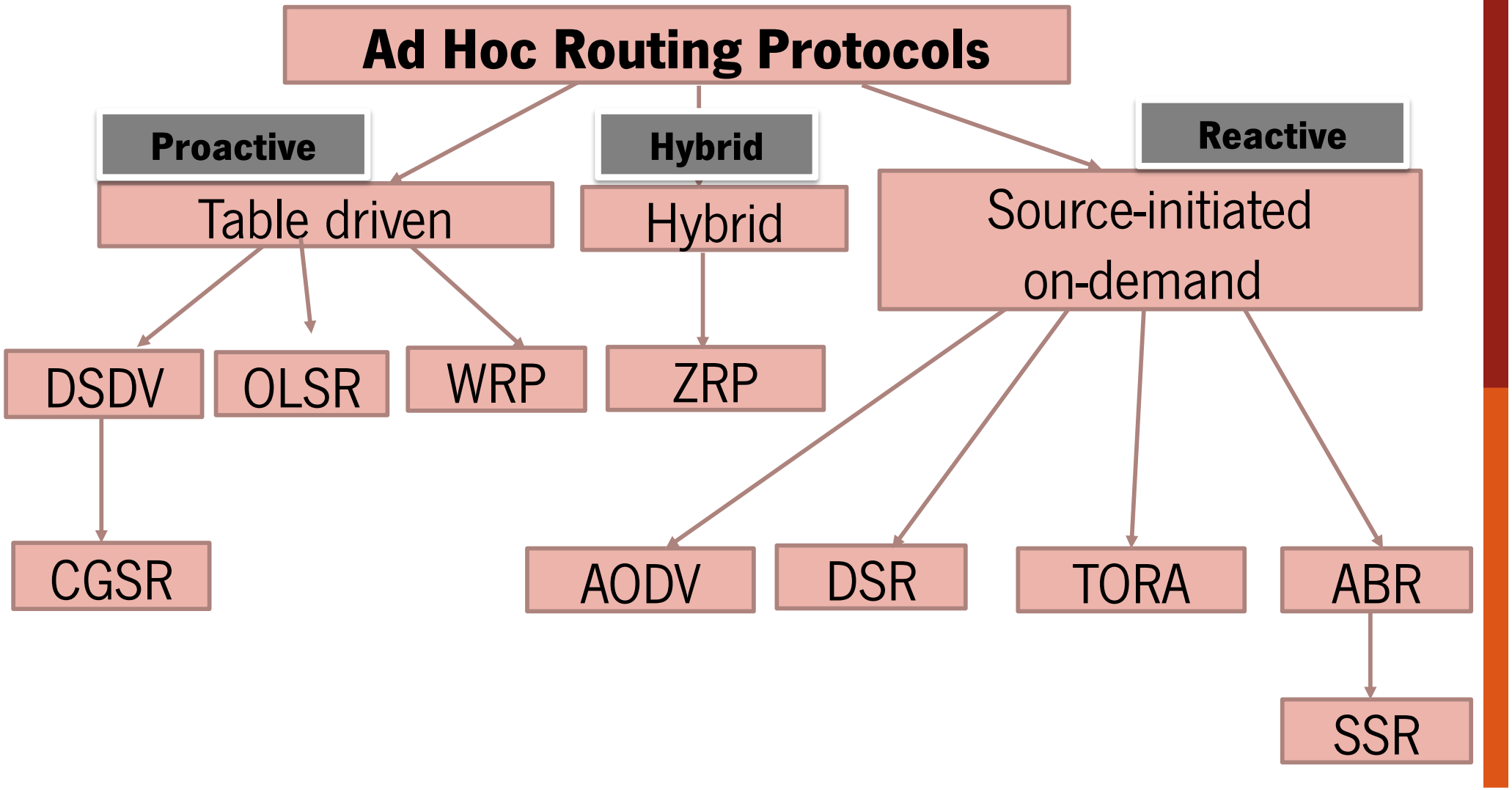
- **Quando devem atuar?**
  - *Opção 1*: protocolo tenta ***sempre*** manter informação de encaminhamento atualizada!
    - Protocolos **pro-activos** / **baseados em tabela** (atualizam as tabelas com informação ainda antes dela ser necessária)
  - *Opção 2*: protocolo só calcula rota quando ***ela é necessária***
    - Protocolos **reactivos** / **a pedido**
  - *Opção 3:* protocolo combina as duas opções anteriores
    - Protocolos híbridos

Classificação dos protocolos

	Pro-Activos	Reativos
Latência das rotas	**Baixa** As rotas estão sempre pre-calculadas e prontas a usar	**Alta** Não se guardam rotas que não estão em uso, pelo que é sempre preciso calcular rota
Sobrecarga do encaminhamento	**Alta** Disseminação de dados na topologia é frequente e gera sobrecarga	**Baixa** Em geral são necessárias menos mensagens de controlo

Estudos mostram que as abordagens reativas têm normalmente melhor desempenho e melhor escalabilidade, no entanto não há uma solução melhor para todos os cenários; resultados dependem de padrões de movimento e de tráfego....

Classificação dos protocolos



Protocolos pro-activos

- **A ideia base é tentar adaptar os protocolos tradicionais bem conhecidos para o ambiente AdHoc**
- **Baseados no algoritmo de Vector Distância de Bellman-Ford:**
  - WRP – Wireless Routing Protocol
  - DSDV – Destination-Sequenced Distance Vector
- **Baseados no Estado das Ligações (Dijkstra)**
  - OLSR – Optimized Link State Routing
  - FSR – FishEye State Routing
- **Combinado dos dois anteriores (LS & DV)**

WRP – Wireless Routing Protocol

- Proposto por: J. Garcia-Luna-Aceves e S. Murthy
- Protocolo pró-ativo baseado no algoritmo de Vetores de Distância
- A métrica usada para a escolha do melhor caminho é o custo ou número de saltos (caminho de menor custo ou caminho mais curto)
- Divulga o penúltimo nó do caminho mais curto para o destino (*predecessor*) com o objetivo de evitar ciclos e ultrapassar o problema da contagem até ao infinito.

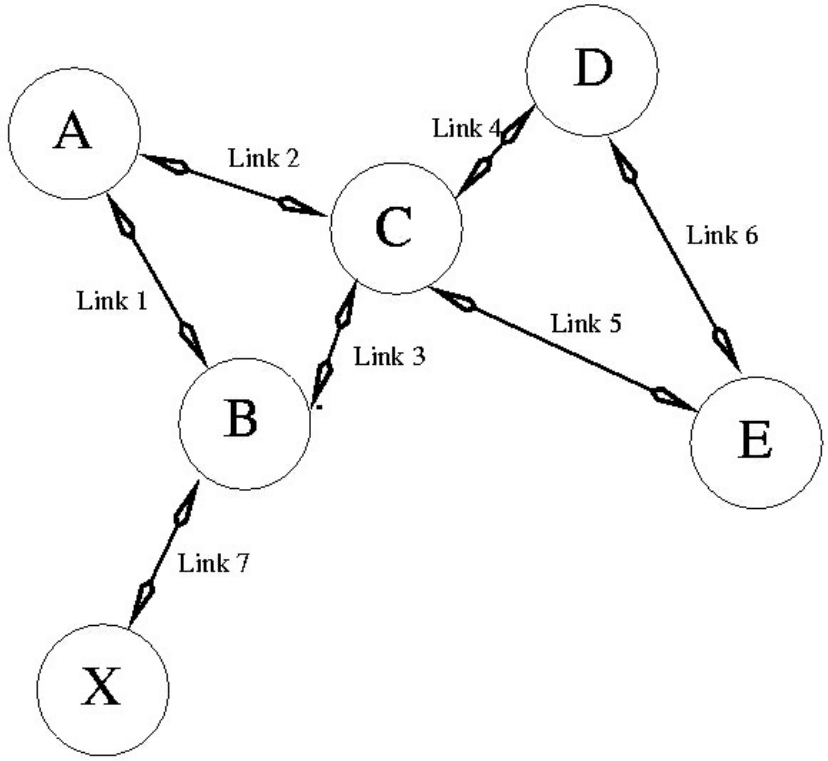
WRP – Wireless Routing Protocol

- **Cada nó mantém a seguinte informação de estado:**
  - Uma *tabela de distâncias* – matriz com predecessor e distância de cada vizinho para cada destino possível;
  - Uma *tabela de rotas* – um vector com uma entrada por cada destino, que especifica o destino, a distância, o predecessor, o sucessor (*next hop*) e uma etiqueta que indica se a rota está correta ou em erro
  - Uma *tabela de ligações* – com custo das ligações a cada um dos vizinhos e timestamp desde a ultima mensagem de atualização recebida sem erro
  - Uma *lista de mensagens a retransmitir (MRL)* – número de sequência da mensagem de atualização, contador de retransmissões a decrementar a cada reenvio, indicação dos vizinhos que enviaram ACKs

WRP – Wireless Routing Protocol

- **Informação trocada entre nós:**
  - Cada nó envia *mensagens de actualização* de rotas aos seus vizinhos; as mensagens de actualização da tabela são retransmitidas até serem *confirmadas* pelos vizinhos (em caso de falha descartar vizinho);
  - As mensagens possuem um número de sequência, uma ou mais actualizações de rotas e uma lista de confirmações de actualizações;

WRP – Exemplo



(todos os custos iguais a 1, com exceção do link 2, que é 10)

ROUTING TABLE AT A

Dest	Cost	Pred	Succ
A	0	A	A
B	1	A	B
C	2	B	B
D	3	C	B
E	3	C	B
X	2	B	B

WRP – Exemplo

- **Suponha que a ligação 1 se quebra. Como reage o nó A ao aperceber-se desta falha?**
  - Considerando por exemplo o nó X, o nó A coloca a distância para X igual a infinito e os nós predecessor e sucessor a nulo;
  - Difunde esta informação, que atinge o nó C (no raio de alcance);
  - O nó C calcula uma rota alternativa para nó A através do link 2 e transmite o seu vetor de distância ao nó A através do link 2.
  - O nó A descobre que consegue chegar aos outros nós da rede através do nó C

WRP – Exemplo

- **Nova tabela de encaminhamento do nó A**

Dest	Cost	Pred	Succ
A	0	A	A
B	11	C	C
C	10	A	C
D	11	C	C
E	11	C	C
X	12	B	C

WRP – Vantagens/Limitações

- **Vantagens**
  - Lida com o problema da contagem até ao infinito e não tem ciclos
  - Converge depressa depois de uma falha
- **Limitações**
  - As mensagens de atualização são muito grandes
  - Obriga à manutenção de quatro tabelas por nó
  - Um nó tem que estar sempre ativo (o modo "silêncio" não é permitido)
    - Se não há atualizações a fazer, envia "hello"

DSDV – Destination-Sequenced DV

- **Tal como o WRP é baseado no algoritmo de Bellman-Ford (DV)**
- **Introduz mecanismos de melhoria de desempenho em AdHoc**
  - Na tabela de rotas, além do destino, próximo salto e do custo, é acrescentado um *número de sequência* **definido pelo destino**
  - O número de sequência permite distinguir rotas obsoletas de outras mais frescas de um modo simples, evitando formação de ciclos!
  - Cada nó acrescenta também um número de sequencia da atualização, em complemento ao número de sequencia do originador
- **Atualizações são despoletadas por eventos e por tempo**
  - Cada nó transmite de forma periódica atualizações aos vizinhos...
  - Ou sempre que há alterações significativas
  - Distinguem-se as atualizações em completas e incrementais...
- **Evitam-se os mecanismos de resolução de ciclos tradicionais**

DSDV – Destination-Sequenced DV

- **Quando X recebe informação de Y com rota Z**
  - Seja S(X) o número de sequência da rota para Z em X e S(Y) o da rota para Z enviado por Y

  - Se S(X) > S(Y), então X ignora a informação de encaminhamento recebida de Y
  - Se S(X) = S(Y), e o custo (métrica) via Y é menor que o da rota que X conhece para Y, então X torna Y no próximo salto para Z
  - Se S(X) < S(Y), então X torna Y no próximo salto para Z, e atualiza S(X) para o valor de S(Y)

DSDV – Destination-Sequenced DV

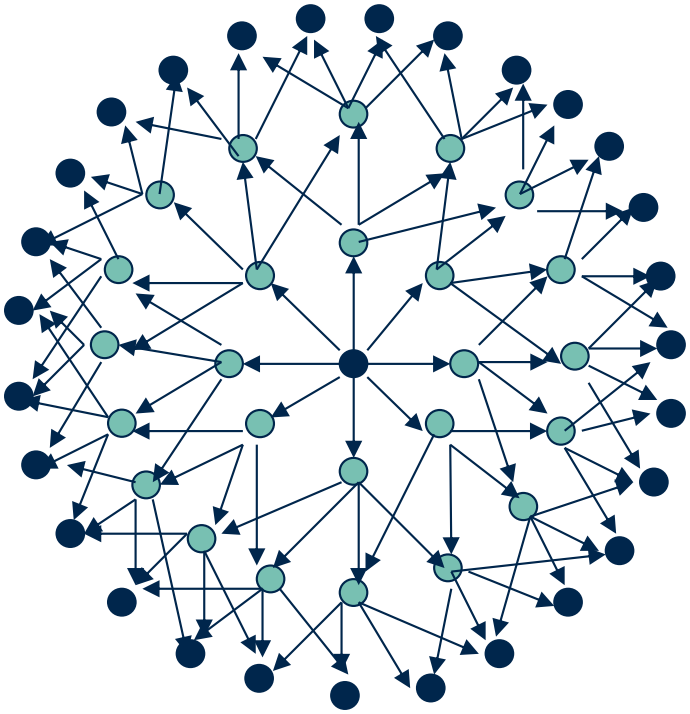
- **Propagação das atualizações:**
  - Um nó deve preferir sempre rotas com maior número de sequência
  - Se tiverem o mesmo número de sequência, preferir as com melhor métrica (neste caso número de saltos)
  - Se resultar numa alteração da tabela deve enviar a todos os vizinhos, que por sua vez fazem o mesmo...
  - Rotas novas implicam sempre envio de atualizações...
  - No caso de atualizações de rotas já existentes pode acontecer que as piores rotas cheguem sistematicamente primeiro que as melhores → isto pode levar a "tempestades" de atualizações...
  - Para evitar isso atrasa-se deliberadamente o envio da atualização enquanto for provável receber novas atualizações;
    - Essa probabilidade determina-se pelo histórico de atualizações que deve ser mantido
  - O movimento de um nó poderia provocar atualizações permanentes que é preciso evitar...

OLSR – Optimized Link-State Routing

- **Baseado no conhecimento da topologia: implica trocas regulares de informação topológica**
- **Introduz o conceito de MPR – Multipoint Relays**
  - Objetivo principal: minimizar a informação de controlo e evitar que todos tenham de mandar para todos na topologia
  - Só os nós eleitos como MPR enviam informação de controlo!
  - Os nós MPR enviam informação de estado das ligações a todos os nós vizinhos que o elegeram como MPR...
- **Seleção dos MPR – passo importante**
  - Cada nó elege os seus MPRs entre os vizinhos com os quais tem ligações bidirecionais;
  - Se um nó N difundir uma mensagem para todos os seus MPR(N) eleitos, e estes a retransmitirem, ela chega a todos os nós vizinhos a dois saltos de distância!
  - Quanto menor for o conjunto de MPRs menos tráfego de controlo se gera!

OLSR – Optimized Link-State Routing

- **Sem MPRs, cada nó terá de:**
  - fazer *flood* periódico do estado dos seus *links*
  - retransmitir informação de estado recebida dos vizinhos
  - manter informação de estado recebida de todos os outros nós
  - calcular rotas para todos os outros



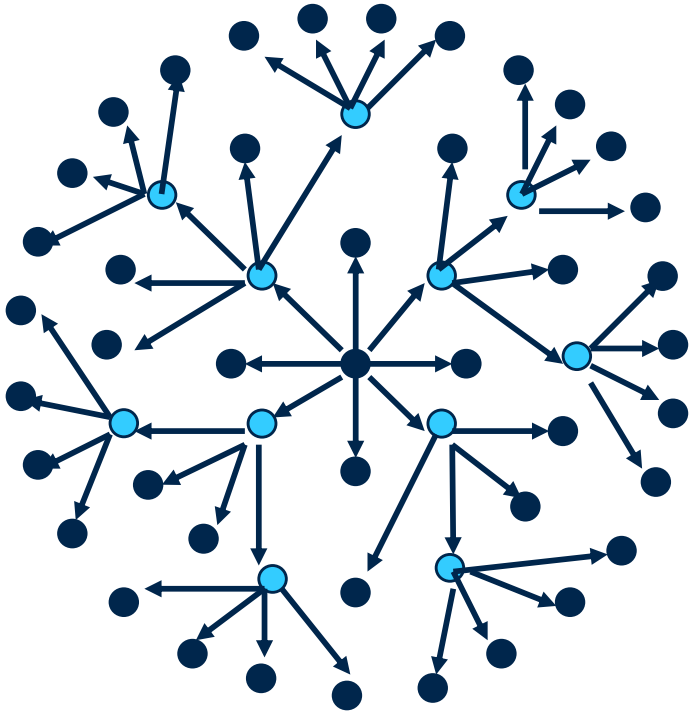
25 retransmissões para difundir mensagem a 3 saltos

Nó retransmissor

ant.comm.ccu.edu.tw/course/96_Network.../1.../UM-OLSR.ppt

OLSR – Optimized Link-State Routing

- Só os nós MPR selecionados retransmitem mensagens
- Selecionar conjunto de MPR que cobrem os vizinhos a 2-saltos
- Os vizinhos de 2-saltos são obtidos pelas mensagens de HELLO
- As escolhas MPR também são anunciadas nos HELLO



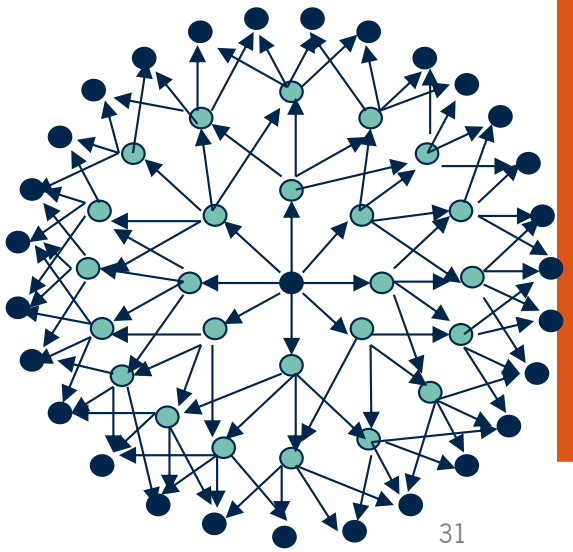
11 retransmissões para difundir uma mensagem a 3 saltos

*Nó retransmissor – **MPR***

ant.comm.ccu.edu.tw/course/96_Network.../1.../UM-OLSR.ppt

Algoritmo de seleção de MPR

- **Eurísticas... porque problema é NP-Completo**
  - Periodicamente todos os nós enviam mensagens **HELLO** por broadcast, incluindo no pacote de **HELLO** a lista dos seus vizinhos a 1-salto
    - ➔ todos os nós conhecem **N1 (vizinhos a 1-salto)** e **N2 (vizinhos a 2-saltos)**
  - Seja D(y) o grau de um vizinho pertencente a N1 ($y \in N1$): número de vizinhos de y, excluindo o nó x e todos os nós vizinhos de x pertencentes a N1
  - Cálculo para um dado nó **X:**
    - São candidatos a MPR apenas os nós que disseram que poderiam ser MPR (excluem-se os outros)
    - Seja **MPRx** o conjunto vazio com os MPR já seleccionados
    - Calular D(y), sendo y membro de N1, para todos os membros de N1
    - Fase 1: Adicionar a MPRx todos os vizinhos em N1 que sejam os *únicos* a dar acessibilidade a algum nó em N2
      - Remover de N2 os nós nestas circunstâncias já cobertos
    - Fase 2: enquanto existirem nós em N2 ainda não cobertos por nenhum candidato a MPR incluido em MPRx, fazer:
      - Selecione o nó em N1 que fornece acessibilidade ao número máximo de nós em N2.
      - No caso de múltiplos nós que fornecem a mesma quantidade, selecione o nó como MPR cujo d(y) é maior.
      - Remova os nós da N2 que agora estão cobertos por um nó no conjunto MPR.



https://tools.ietf.org/html/rfc3626#section-8.3

Protocolos reactivos

- **Como funcionam?**
  - Um emissor que queira enviar um pacote para um dado destino só tem uma (?) hipótese: envia pedido de rota a todos os vizinhos!...
  - O pedido vai sendo propagado até que – eventualmente! – o destino recebe o pedido e responde de volta (pelo mesmo percurso)

- **Exemplos:**
  - DSR – Dynamic Source Routing
  - AODV – AdHoc On-Demand Distance Vector
  - ZRP – Zone Routing Protocol

DSR – Dynamic Source Routing

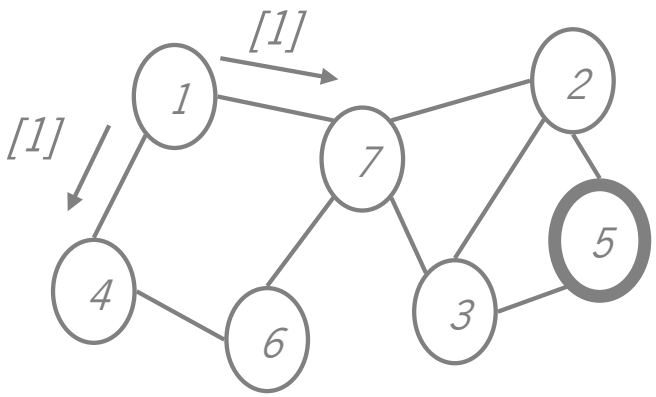
- **DSR permite que os nós descubram rotas a pedido**
  - Protocolo reativo
- **Baseado no conceito de "Source Routing":**
  - Cada pacote enviado carrega no cabeçalho a rota completa com indicação explicita de todos os nós intermédios até ao destino
- **Dois mecanismos distintos:**
  - *Descoberta da rota* – iniciado a pedido pelo originador quando pretende enviar um pacote para o destinatário
  - *Manutenção da rota* – iniciado a pedido quando uma rota se quebra durante uma transmissão...

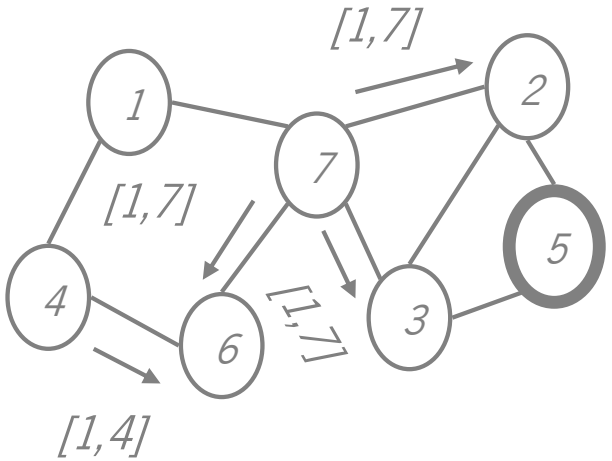
DSR – Dynamic Source Routing

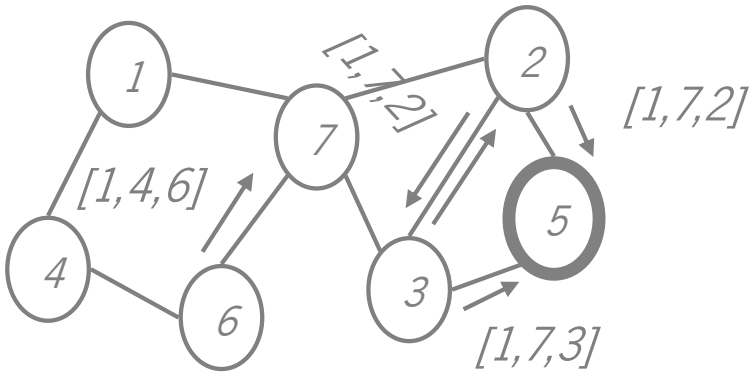
- **Descoberta da rota (de S para D)**
  - **S** analisa a sua cache de rotas a ver se tem alguma recente
  - Se não existir, **S** envia um broadcast com um *Route Request* para **D**, que vai ser recebida por todos os nós ao alcance
    - Cada pedido identifica **S** e **D** e tem um **RequestID** único
    - Pedido acumula o percurso já percorrido numa lista de nós (iniciada a vazio)
  - Quando um nó recebe um *Route Request* verifica se é o destinatário **D**
    - Se for, responde com um *Route Reply*, usando a rota acumulada no *Route Request*
    - Se não for, Ignora o pedido se o **RequestId** já tiver sido processado anteriormente, ou se o seu endereço estiver na lista dos nós já percorridos
    - Senão, acrescenta o seu próprio identificador na lista de nós percorridos e reenvia o pedido por broadcast a todos os nós no seu alcance rádio
- **Pacotes ficam num buffer de espera em S até haver resposta de D**

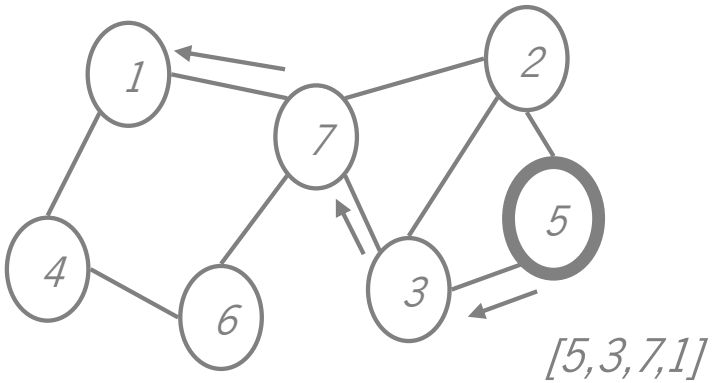
DSR – Dynamic Source Routing

Exemplo de um pedido de rota de 1 para 5









Nó 5 responde de volta usando a rota que foi armazenada no pedido RREQ (source routing)

DSR – Dynamic Source Routing

- **Processo de manutenção das rotas**
  - As entregas dos pacotes devem ser confirmadas, explicitamente, de forma passiva, ou mesmo por mecanismos de nível MAC
  - Se o pacote é retransmitido um número máximo de vezes por um nó **I** sem confirmação a rota está obsoleta!
  - O nó intermediário **I** que não conseguir reenviar o pacote retorna uma notificação de erro de rota (*Route Error*) ao originador **S**
  - Sempre que receber uma indicação de *Route Error*, o nó originador deve invalidar a rota da sua cache
  - A retransmissão do pacote será originada no nó S mas pelo protocolo de transporte (se for o caso)
  - Pacotes seguintes seguem o processo de descoberta normal...

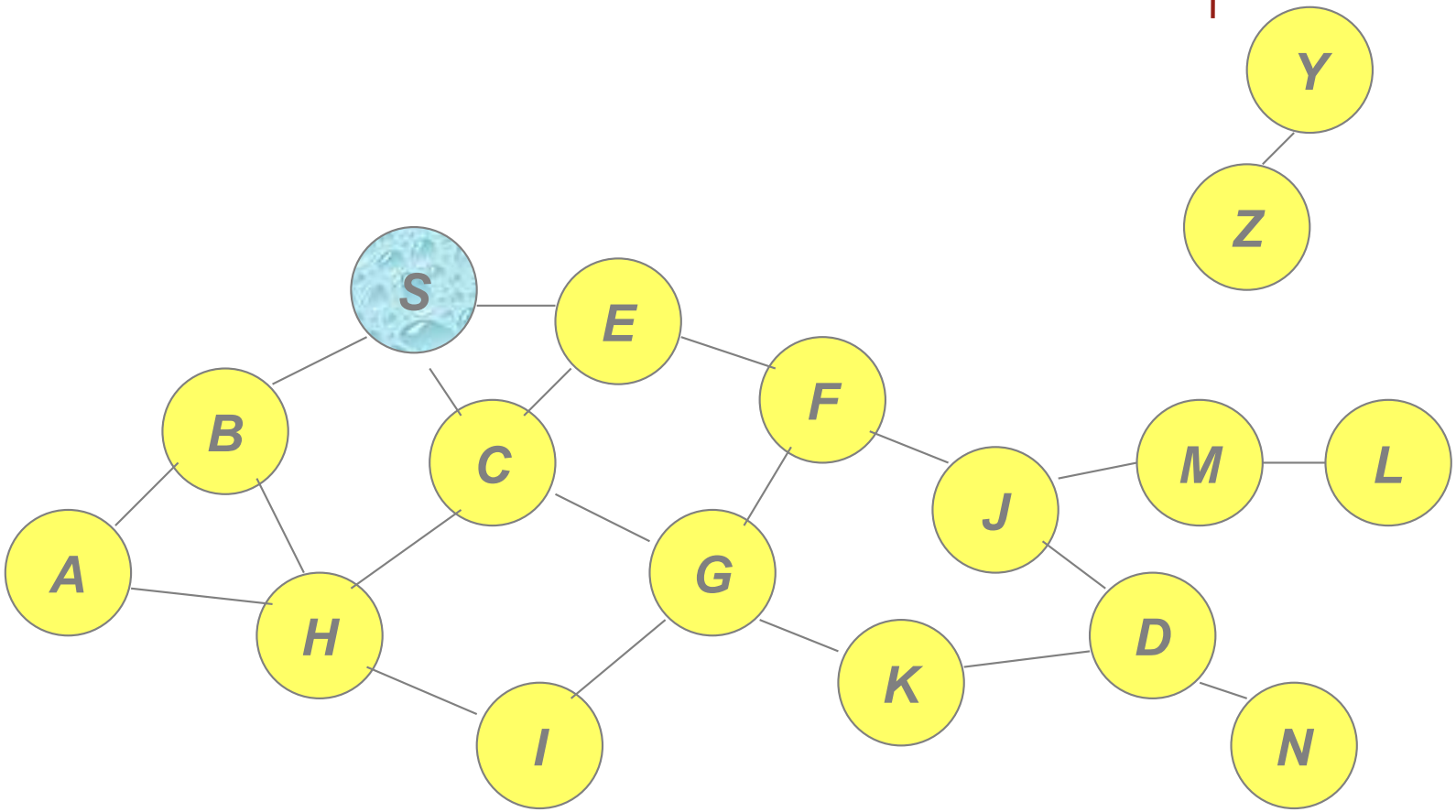
DSR – Dynamic Source Routing

- **Otimizações possíveis**
  - Cache de rotas: os nós intermédios que escutam mensagens (sinal rádio é broadcast) com pedidos de outros nós podem tomar conhecimento da topologia e atualizar cache
  - Um nó intermédio pode responder com rotas que tem em cache
  - Mas: as caches mal parametrizadas induzem rotas erradas!...
    - Necessários mecanismos inteligentes de gestão das caches
  - A reparação de uma rota pode ser feita localmente no nó que deteta o erro... Iniciando ele um pedido de rota para o destino
  - As respostas podem ser deliberadamente atrasadas (um tempo aleatório) para evitar sobrecargas...
  - Uso de TTLs limitados para evitar o "*flooding*" descontrolado.

AODV – AdHoc On Demand DV

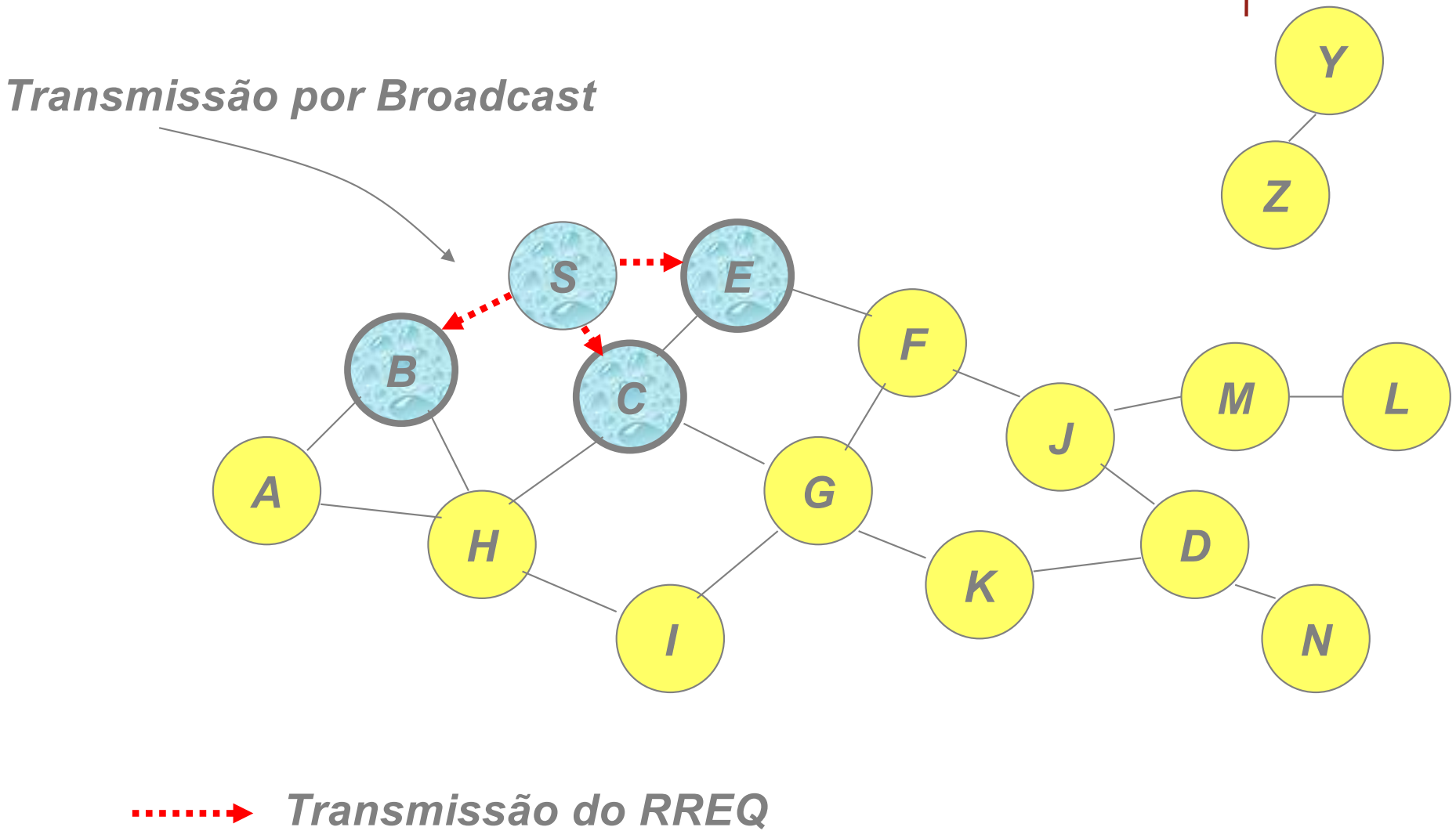
- **Protocolo reativo**
  - Essencialmente a ideia é a mesma do DSR... no que diz respeito à estratégia de descoberta de rotas...
  - Não se utiliza source routing mas sim *tabelas de rotas* em cada nó
    - Cada pedido de *route request* introduz estado na tabela que ou expira ou é consolidado quando receber de volta um *route reply*
    - Os nós memorizam de onde o pedido chegou ...
    - Utilizam-se números de sequência para distinguir pedidos sucessivos

Pedidos de Rota no AODV

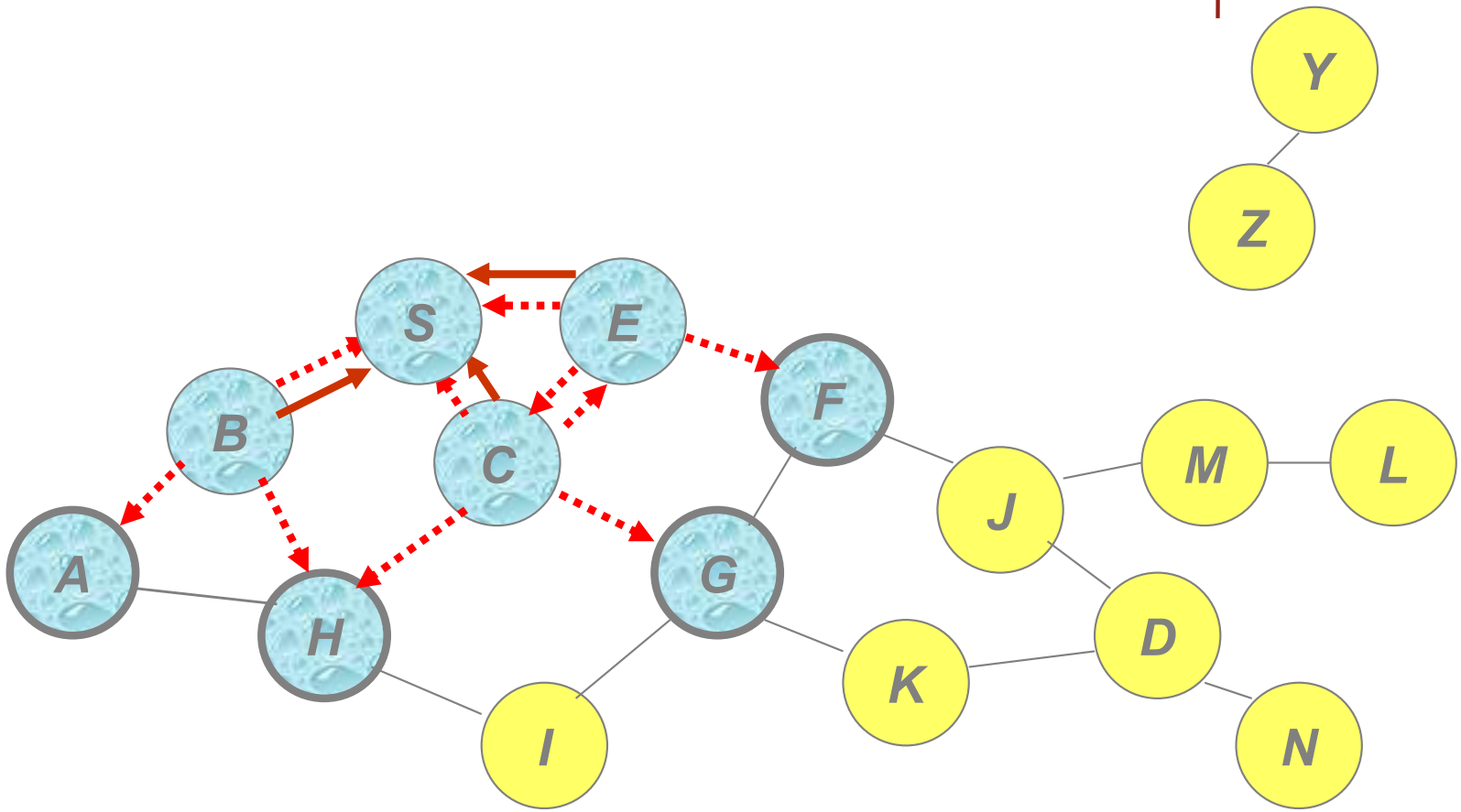


Representa um nó que recebeu um RREQ para D de S

Pedidos de Rota no AODV

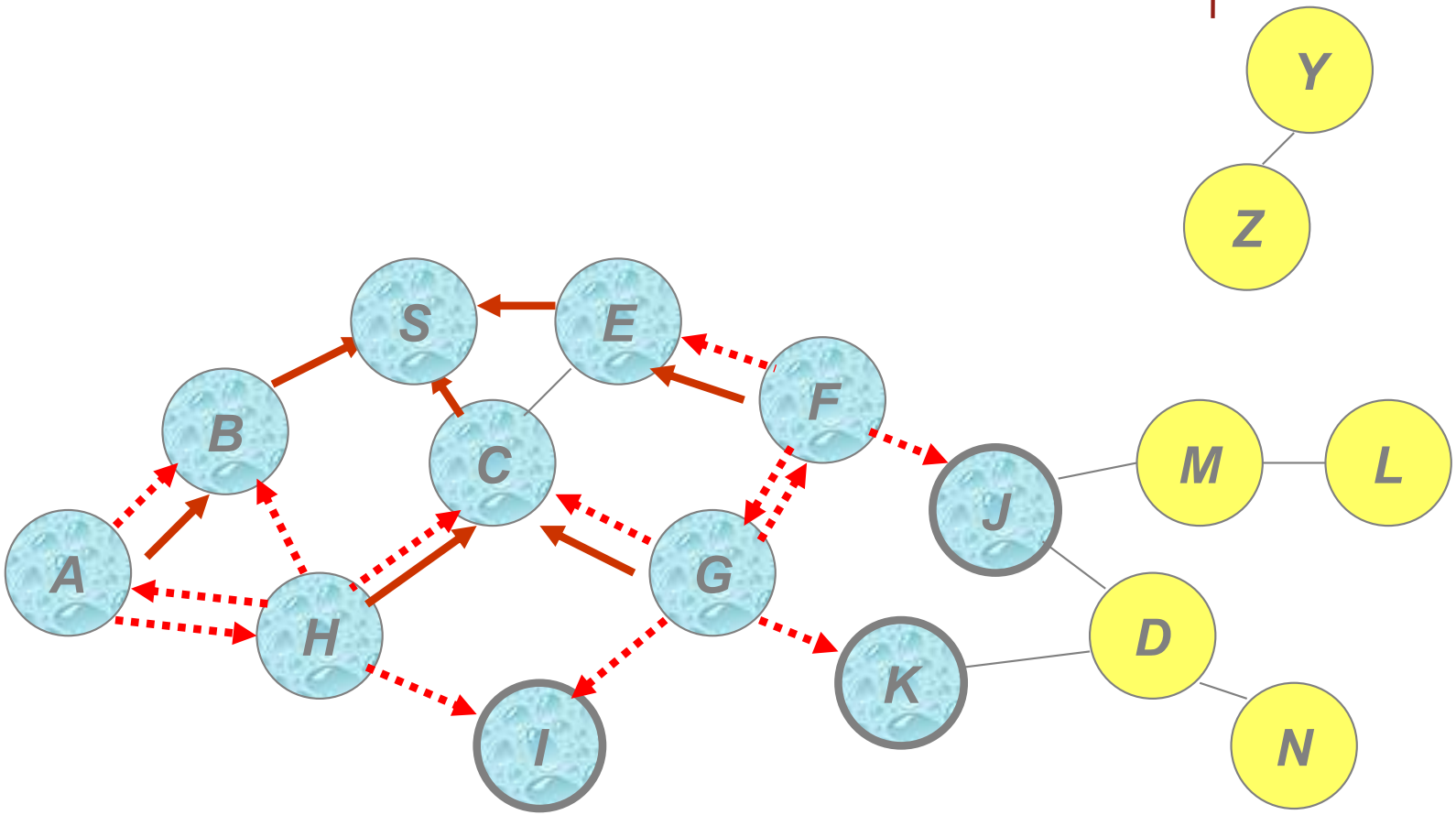


Pedidos de Rota no AODV



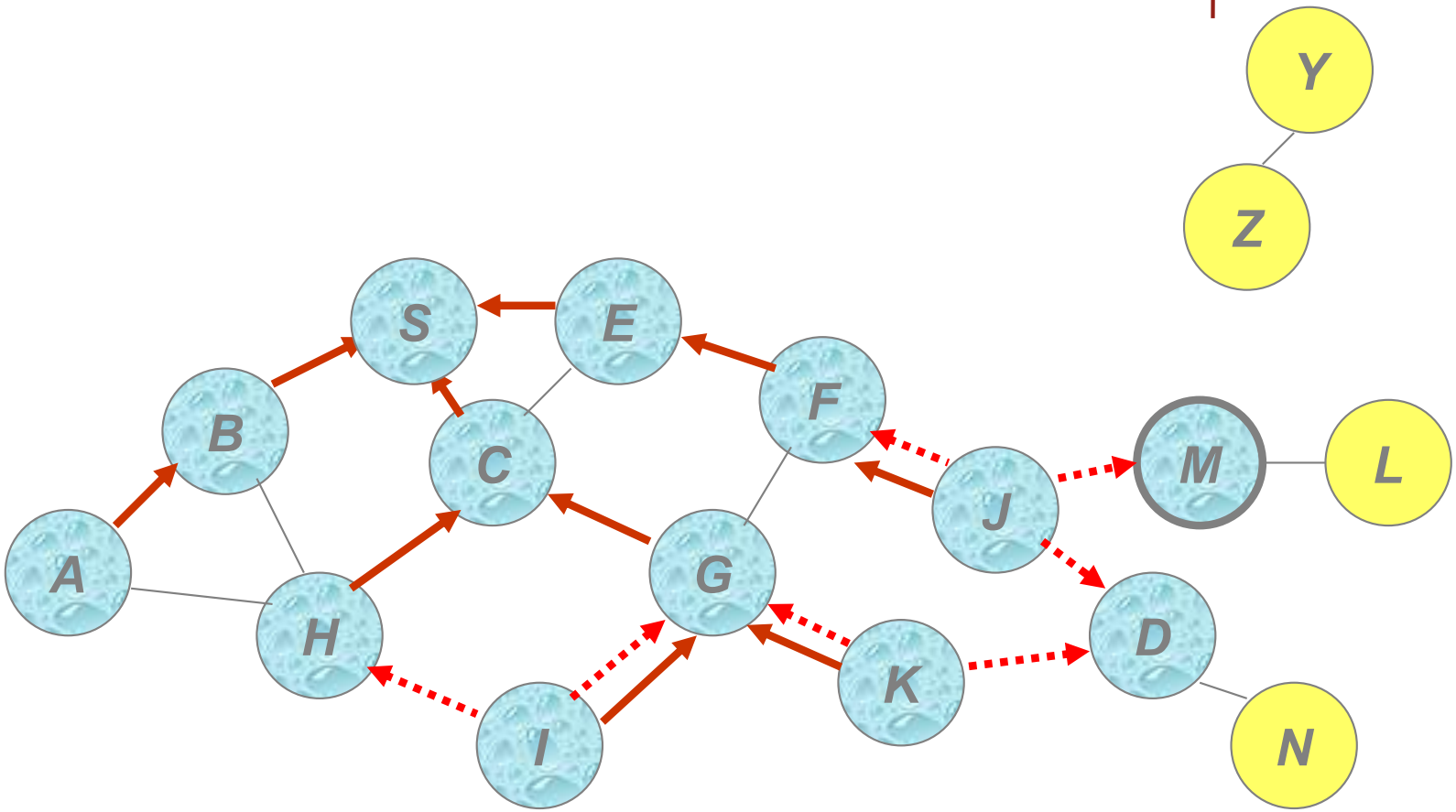
← *Representa as ligações no caminho inverso*

Caminho inverso no AODV

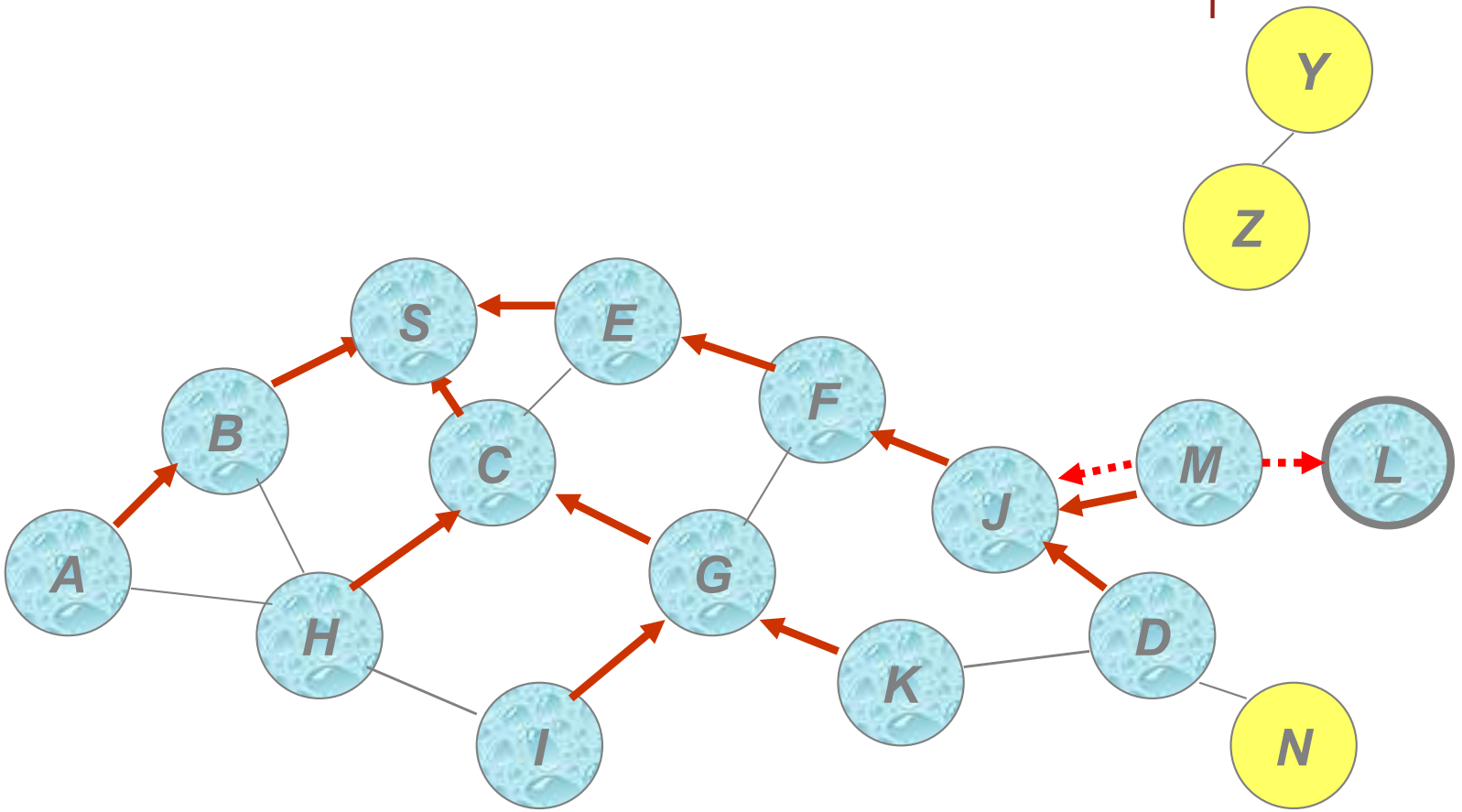


O nó C recebeu RREQs do nó G e H, mas não os retransmitiu novamente porque já o tinha feito uma vez!

Caminho inverso no AODV

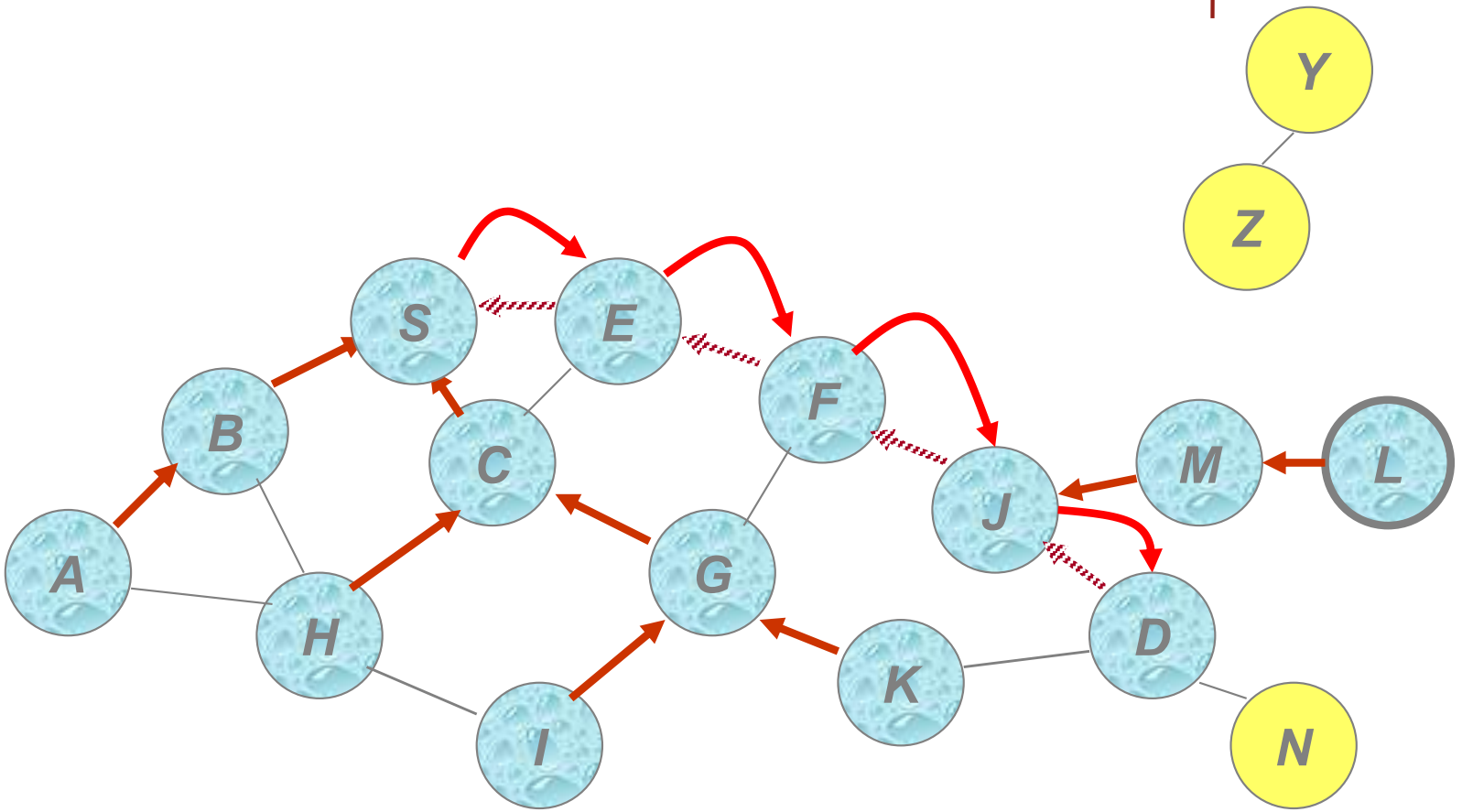


Caminho inverso no AODV



D não retransmitiu o RREQ porque o nó D é o nó destino do RREQ

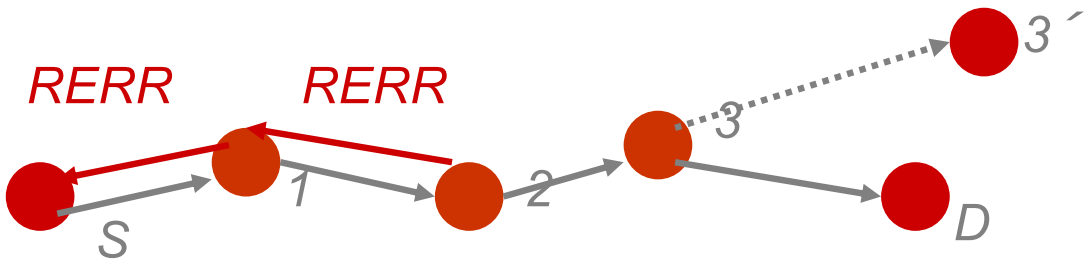
Estabelecimento do caminho no AODV



A rota é estabelecida quando o RREP viaja ao longo do caminho inverso

Representa uma ligação incluída na rota estabelecida

Manutenção das rotas



- A ligação do nó 3 para D quebrou-se porque o nó 3 moveu-se.
- O nó 2 envia uma mensagem RERR para 1 e 1 envia uma mensagem de volta para S.
- S reinicia o processo de descoberta de rota, se ainda necessitar de uma rota para D.

ZRP – Zone Routing Protocol

- **Trata-se de um protocolo híbrido**
- **Para cada nó define-se uma "Zona" (pode ser em nº saltos)**
- **Protocolo pro-activo dentro da zona…**
  - ***IARP (IntrA Zone Routing Protocol),*** baseado no vector distância ou no estado das ligações…
- **Protocolo reactivo fora da zona**
  - ***IERP (IntEr Zone Routing Protocol)***, baseado em procura a pedido de rotas (*route request/route reply*)
- **Utilizar um conceito de "Bordercast" em vez de "broadcast" nos pedidos de rota**
  - Enviar para todos os nós fronteira da zona

ZRP – Zone Routing Protocol

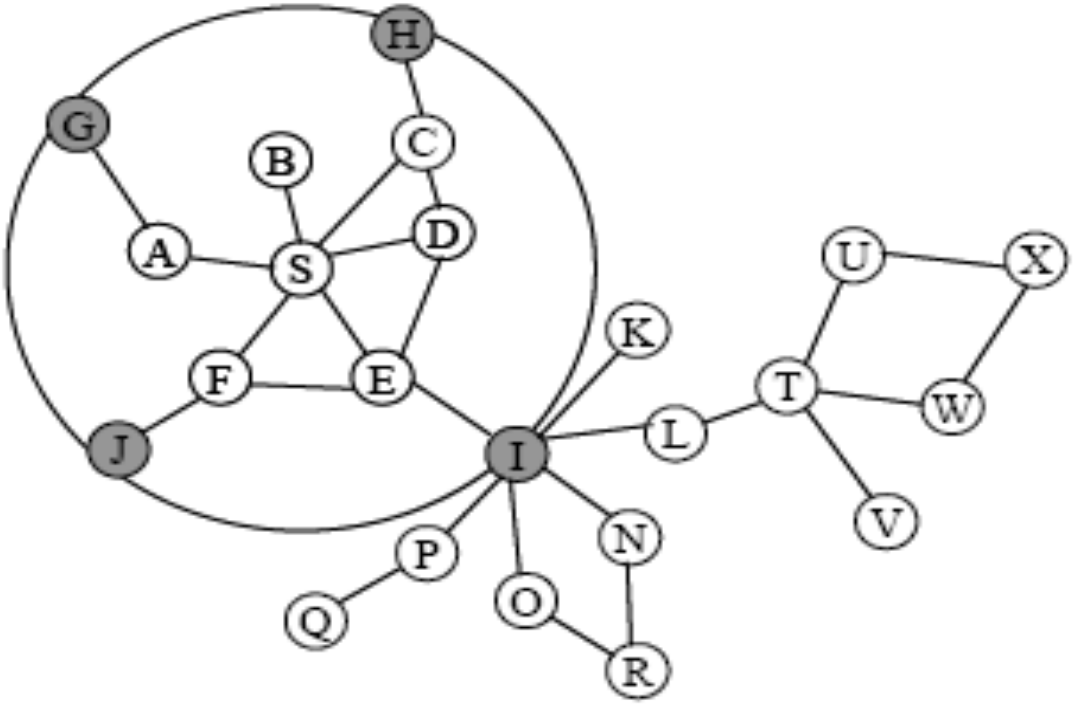


Figure 3: The routing zone of node S

ZRP – Zone Routing Protocol

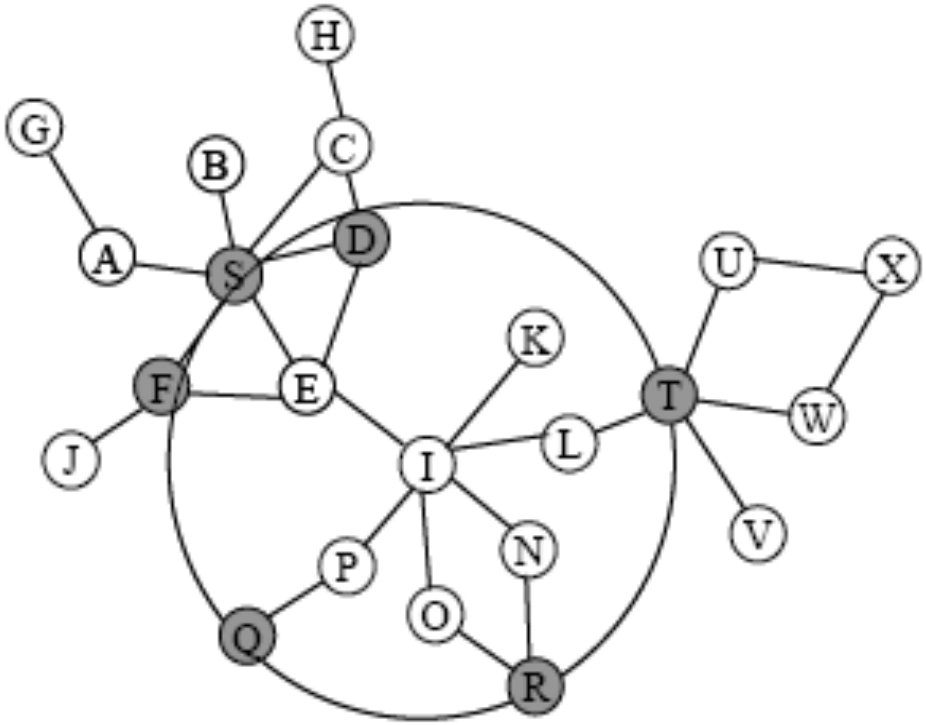


Figure 4: The routing zone of node I

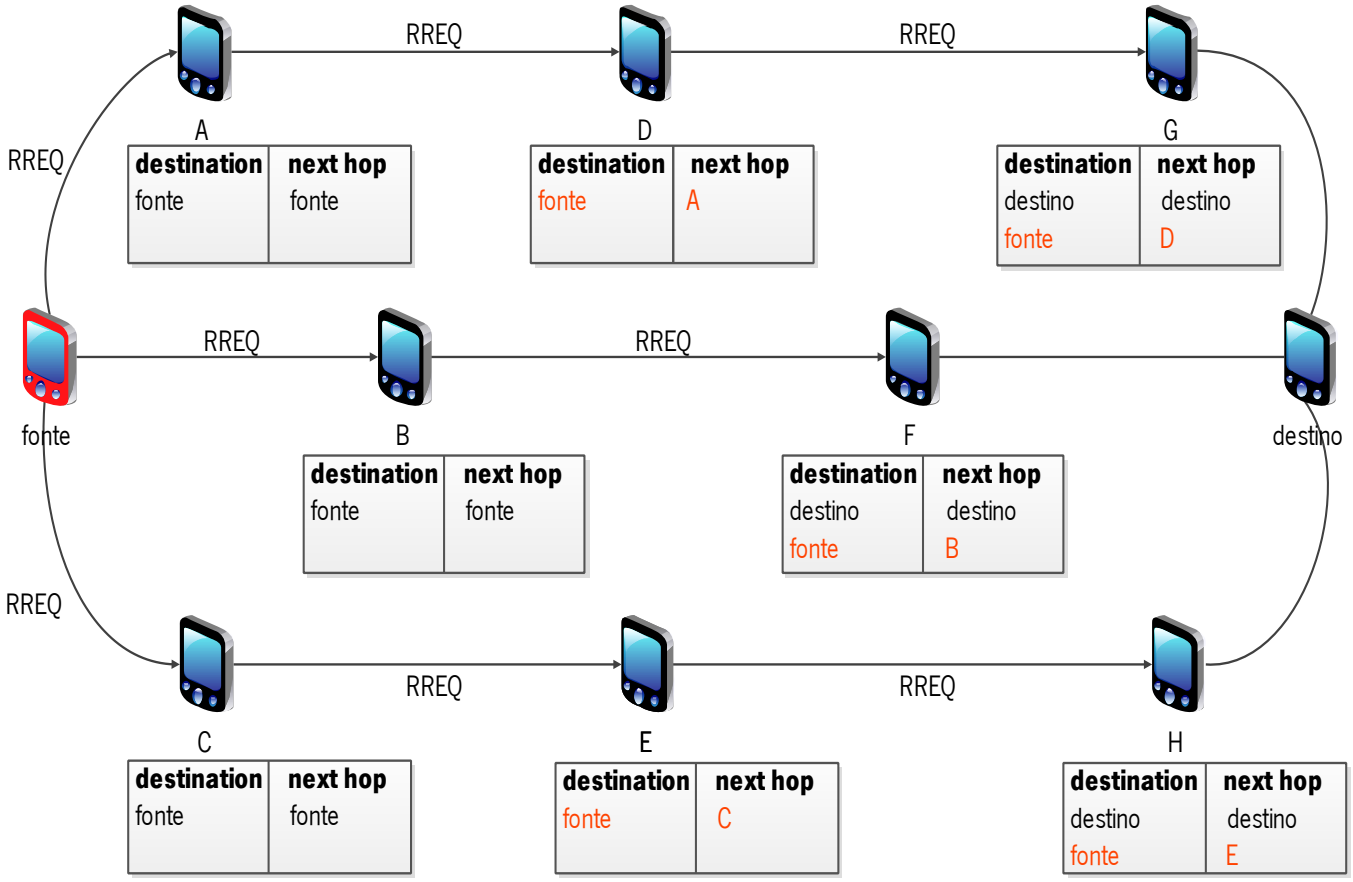
Outras abordagens

- **Encaminhamento por múltiplos caminhos**
  - Objetivo: Em vez de um único caminho para o destino calculam-se vários
    - Rotas de backup que permitam lidar com as alterações muito frequentes da topologia,
    - Balanceamento de carga,
    - Encaminhamento com QoS, etc.
- **Exemplo: AOMDV (Ad-hoc On-demand Multipath Distance Vector Routing Protocol)**

AOMDV - Ad-hoc On-demand Multipath Distance Vector

Processo de descoberta de rota

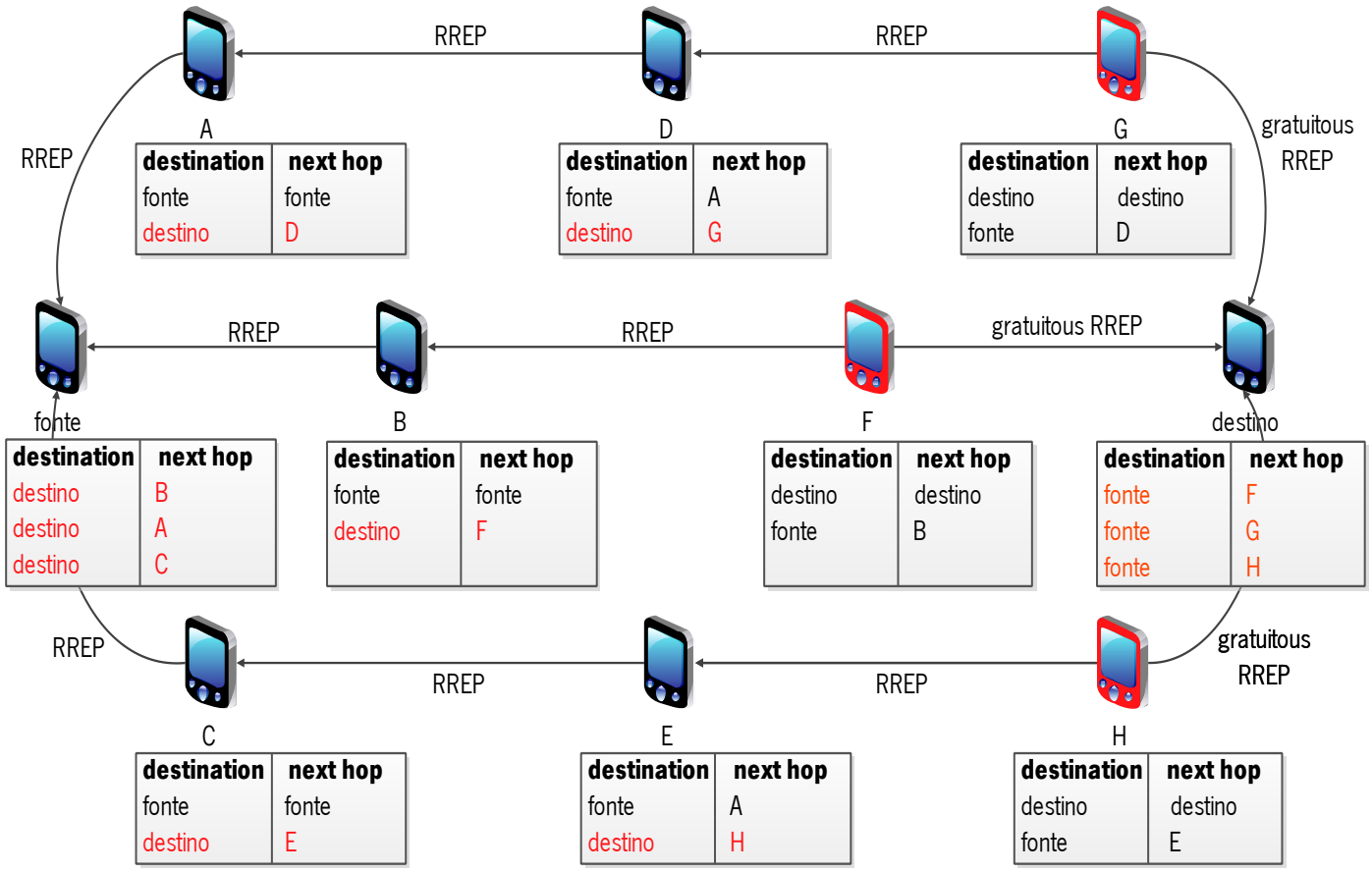
- *Pedido de rota – mensagem Route Request (RREQ)*



AOMDV - Ad-hoc On-demand Multipath Distance Vector

Processo de descoberta de rota

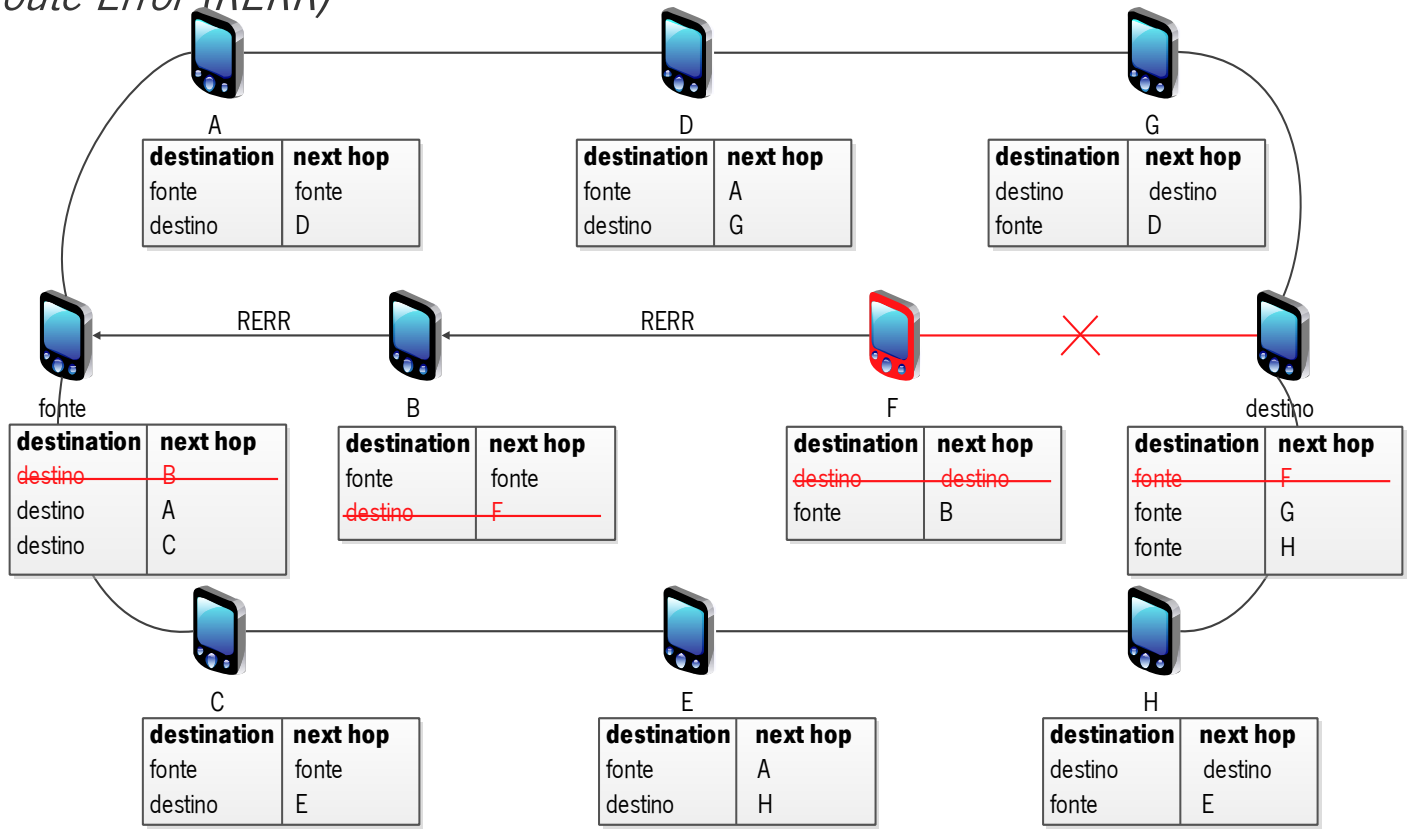
- *Resposta ao pedido de rota - mensagem Route Reply (RREP)*



AOMDV - Ad-hoc On-demand Multipath Distance Vector

Manutenção de rotas

- *Mensagens Hello*
- *Mensagens Route Error (RERR)*



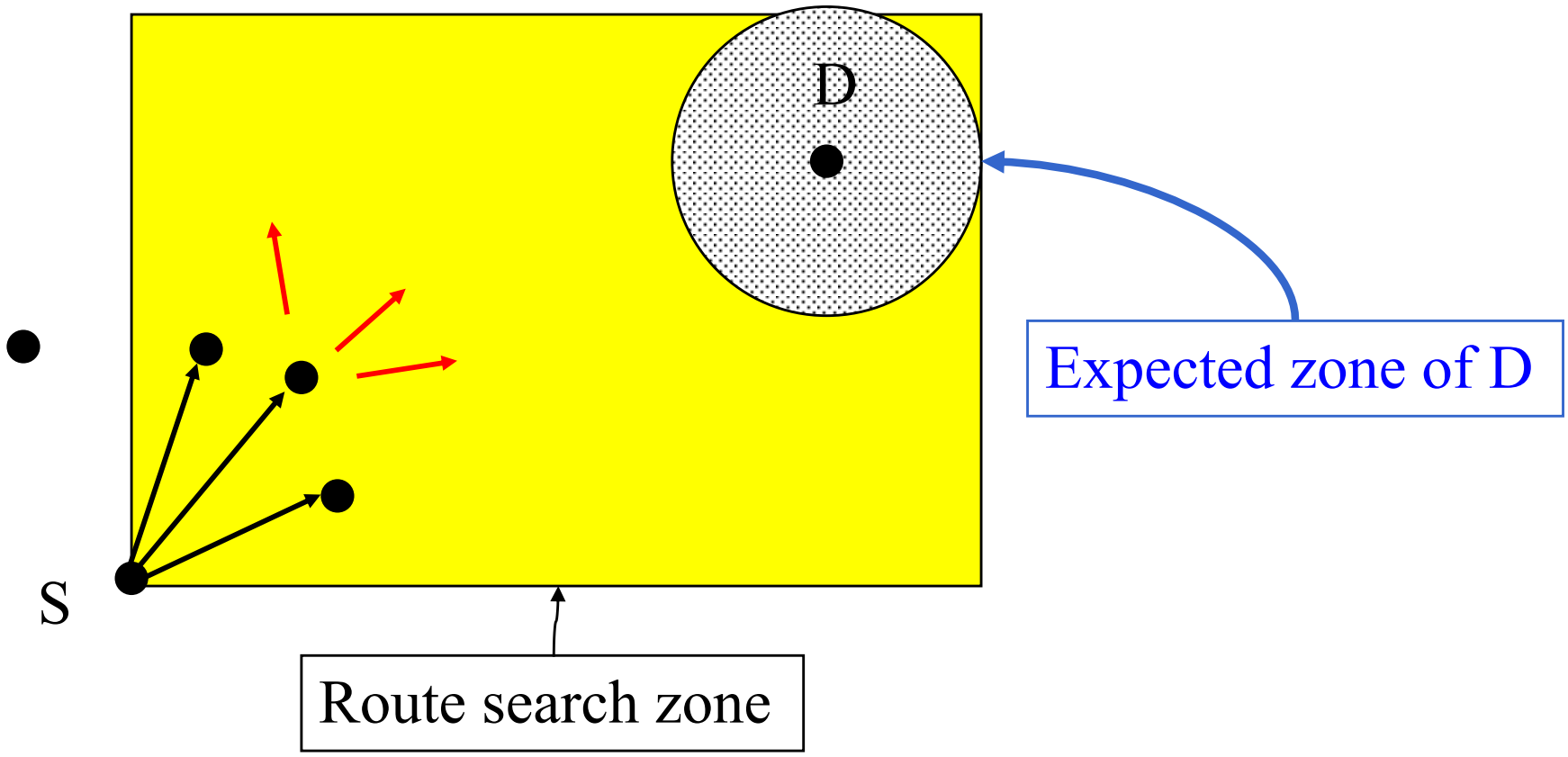
Outras abordagens

- **Encaminhamento Geográfico**
  - Objetivo: tirar partido da informação de localização do nós para encaminhar o tráfego;
  - Pressupostos:
    - Todos os nós conhecem a sua própria localização (para isso poderão recorrer por exemplo ao GPS);
    - A localização do nó destino também é conhecida à partida (através de um serviço de localização);
  - Exemplo
    - LAR (Location-Aided Routing)

Encaminhamento Geográfico

- **LAR (Location-Aided Routing)**
  - Todos os pacotes transportam a posição atual da fonte para permitir que todos os nós aprendam a posição uns dos outros;
  - A partir daí o funcionamento é semelhante ao funcionamento do DSR, com uma diferença significativa. Quando a localização do destino for conhecida o pacote de ROUTE_REQ é reencaminhado na direção do destinatário ("route search zone");

LAR (Location-Aided Routing)



Bibliografia

- Azzedine Boukerche, Begumhan Turgut, Nevin Aydin, Mohammad Z. Ahmad, Ladislau Bölöni, Damla Turgut, **Routing protocols in ad hoc networks: A survey**, Computer Networks, Volume 55, Issue 13, 2011,
- Eiman Alotaibi and Biswanath Mukherjee. **A survey on routing algorithms for wireless Ad-Hoc and mesh networks,** Computer Networks, Volume 56, Issue 2, 2012, 940-965.