INF-111

Redes Sem Fio

Aula 07
Redes Ad Hoc

Prof. João Henrique Kleinschmidt

Santo André, março de 2016

Roteiro

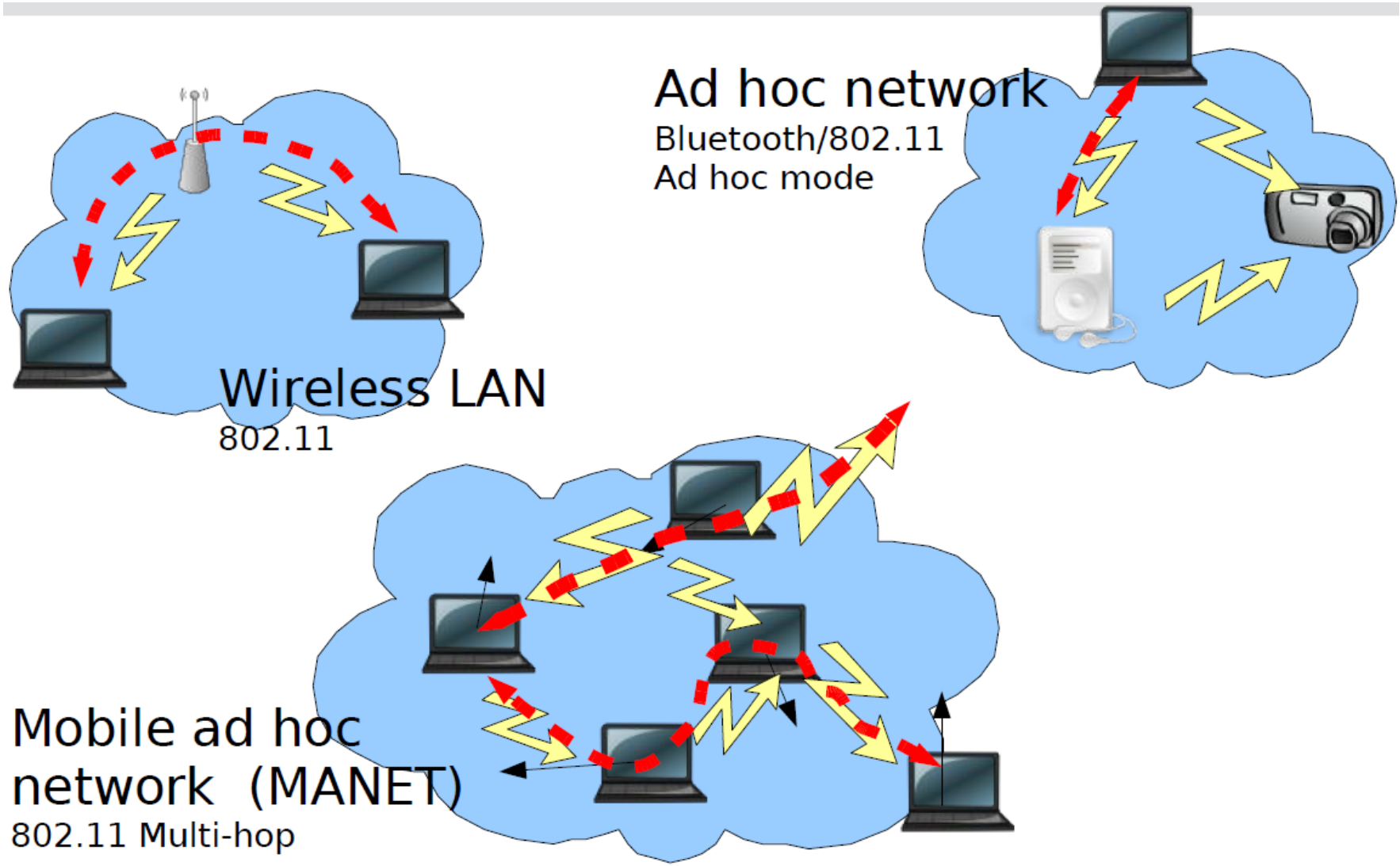
- **Introdução**
- **Camada de acesso ao meio**
- **Roteamento**
- **Protocolos pró-ativos, reativos e híbridos**
- **Protocolos da camada de transporte**

Introdução

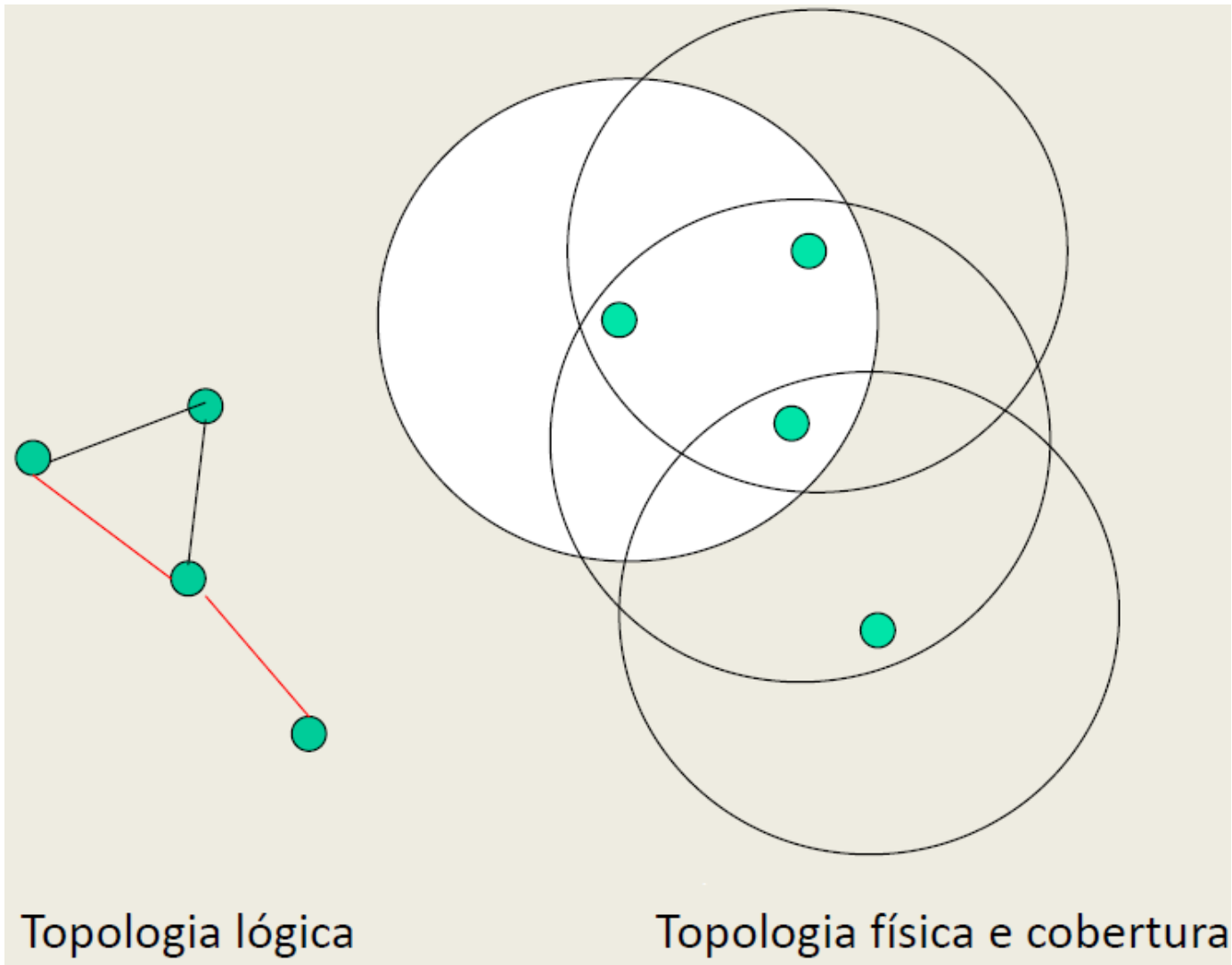
- Diversas tecnologias de redes sem fio dependem de uma infraestrutura existente Ex: redes celulares, WLANs
- Conectividade com um único salto

- **Mobile Ad hoc NETworks** (MANETs) não requer a existência de uma infraestrutura prévia
- Do latim: "Ad Hoc" é uma expressão que significa "para este fim“
- Formada espontaneamente a partir de um conjunto de nós (possivelmente móveis)
- Conectividade entre os nós se dá tipicamente por meio de múltiplos saltos
- Mobilidade provoca mudanças na topologia e, portanto, nas rotas

Introdução


Ad hoc network
Bluetooth/802.11
Ad hoc mode
Wireless LAN
802.11
Mobile ad hoc
network (MANET)
802.11 Multi-hop

Comunicação em uma rede ad hoc



MANETs

- Independência da existência de uma infraestrutura fixa
  - Úteis quando a infra estrutura está ausente, destruída ou sem possibilidade de uso
- Redes de curto alcance
- Facilmente implementadas

Características

- Não assume nenhuma infraestrutura de apoio
- Sem necessidade de administração, configuração ou projeto
- Fonte de energia limitada (*i.e., baterias)*
- Rotas entre nós podem conter múltiplos saltos (*"hops") Qualquer nó pode transmitir ou receber dados e pode atuar como intermediário na comunicação*
- Cada nó normalmente possui conectividade com mais de um nó

- Nós tem liberdade de locomoção: rotas podem mudar ao longo do tempo!
- Segurança é um problema intrínseco

Características

- Os nós tipicamente tem as mesmas responsabilidades e recursos:
  - Bateria (energia)
  - Rádio (alcance de transmissão)
  - Processamento
  - Roteamento
- Mas com capacidades assimétricas:
  - Alcance de transmissão e rádios podem diferir
  - Vida da bateria pode diferir
  - Nós podem ter diferentes capacidades de processamento
  - Nós podem ter diferentes padrões de velocidade e mobilidade

Características

- No entanto, em alguns tipos de MANETs as responsabilidades dos nós podem diferir
  – Somente alguns nós podem rotear pacotes
  – Alguns nós podem atuar como **líderes para outros nós próximos (*e.g., cluster-head)***

- Características de tráfego podem diferir em diferentes redes ad hoc:
  – Taxa de transmissão
  – Limitações de pontualidade na entrega de dados
  – Confiabilidade
  – *Unicast, multicast, broadcast*
  - Padrões de mobilidade
- Podem coexistir e cooperar com uma rede baseada em infraestrutura fixa

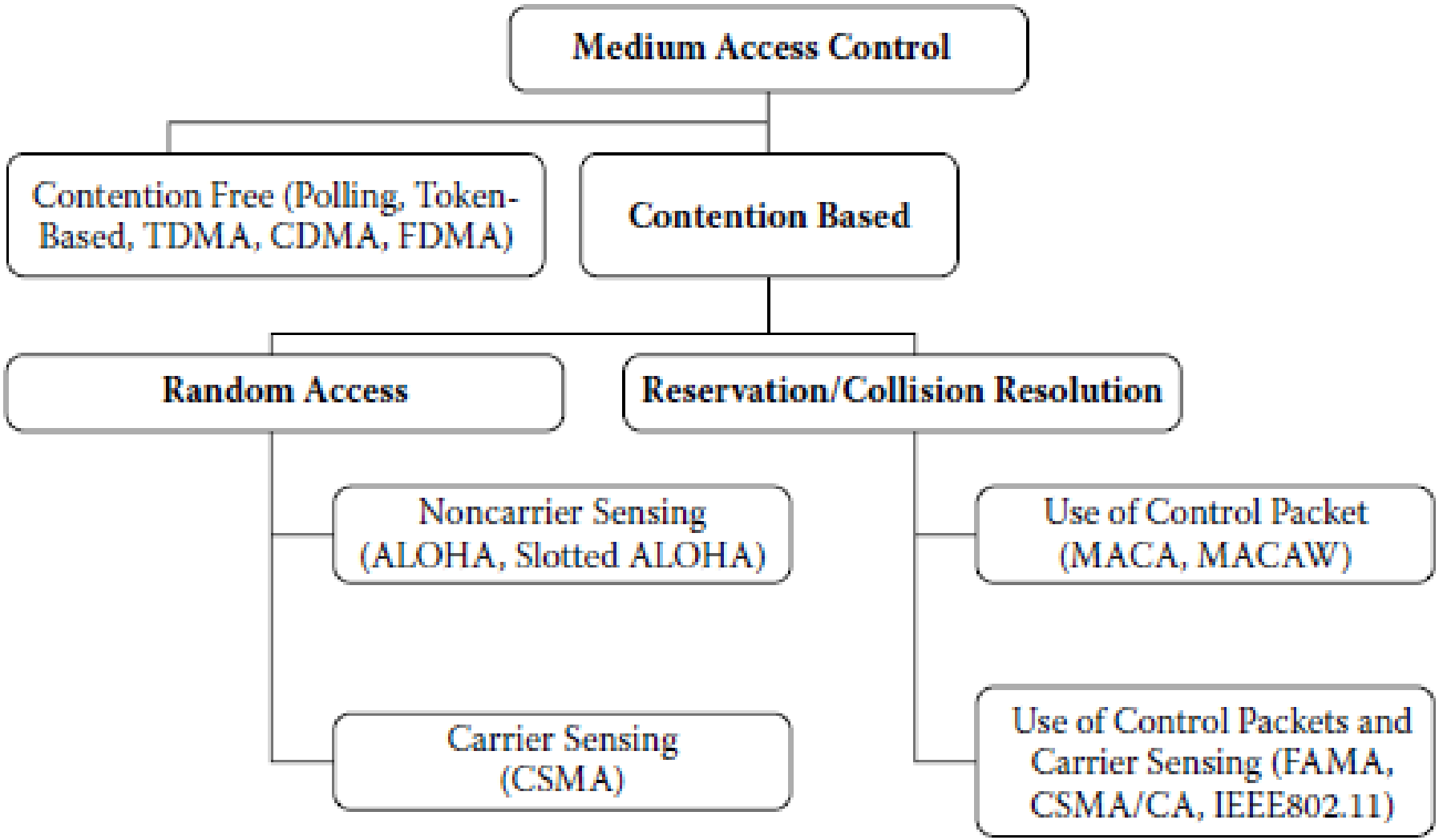
Aplicações

- Wireless Personal Area Networks (WPANs)
- Ambientes civis
  - Games, salas de encontro, carros
- Ambientes militares
  - Soldados, tanques e aviões
- Operações de emergência
  - Resgate, combate a incêndios, terremotos, desastres naturais

Desafios

- Os nós devem realizar comunicações de forma coordenada sem depender de nenhuma infraestrutura
  - Descoberta de serviços e nós na vizinhança
  - Controle de acesso ao meio
  - Roteamento unicast e multicast
  - Controle de topologia
  - Gerenciamento de grupos
  - Segurança
  - QoS
  - etc...
- Dificuldades
  - Limitações dos dispositivos, mobilidade, canal sem fio

Classificação de Protocolos MAC



Camada de acesso ao meio

- Protocolos MAC
  - MACA (*Medium Access with Collision Avoidance*)
    - RTS/CTS
    - MACAW (MACA Wireless)
  - MACA-BI (*MACA by Invitation*)
    - Protocolo iniciado no receptor
    - RTR (*Ready-to-Receive*)
    - Eficiente para modelos de tráfego previsíveis
  - CSMA/CA
  - Previsão de colisão e congestionamento
  - Protocolos MAC eficientes em energia
    - Desligar o rádio quando não for utilizado
  - Protocolos com *fairness* (justiça; equidade)
  - IEEE 802.11, 802.15.1 e 802.15.4 (ver aulas anteriores)

Roteamento

- Em redes ad hoc o roteamento é um problema bem mais complexo que em redes cabeadas
  - Redes cabeadas têm uma associação natural entre endereços IP e localização na rede (domínio administrativo)
  - Rotas variam pouco em redes cabeadas: em redes ad hoc o ambiente é altamente dinâmico

Algoritmos de roteamento tradicionais

- Vetor de distância
  - Troca periódica de mensagens com os vizinhos
  - Mensagem informa quem pode ser encontrado e a que distância
  - Seleção do trajeto mais curto, se possível
  - RIP (*Routing Information Protocol*)
  - Protocolo demora para convergir
- Estado de enlace
  - Notificação periódica de todos os roteadores sobre o estado de todos os enlaces
  - Cada roteador conhece toda a topologia da rede
  - OSPF (*Open Shortest Path First*)
  - Algoritmo de Dijkstra

Problemas em algoritmos tradicionais

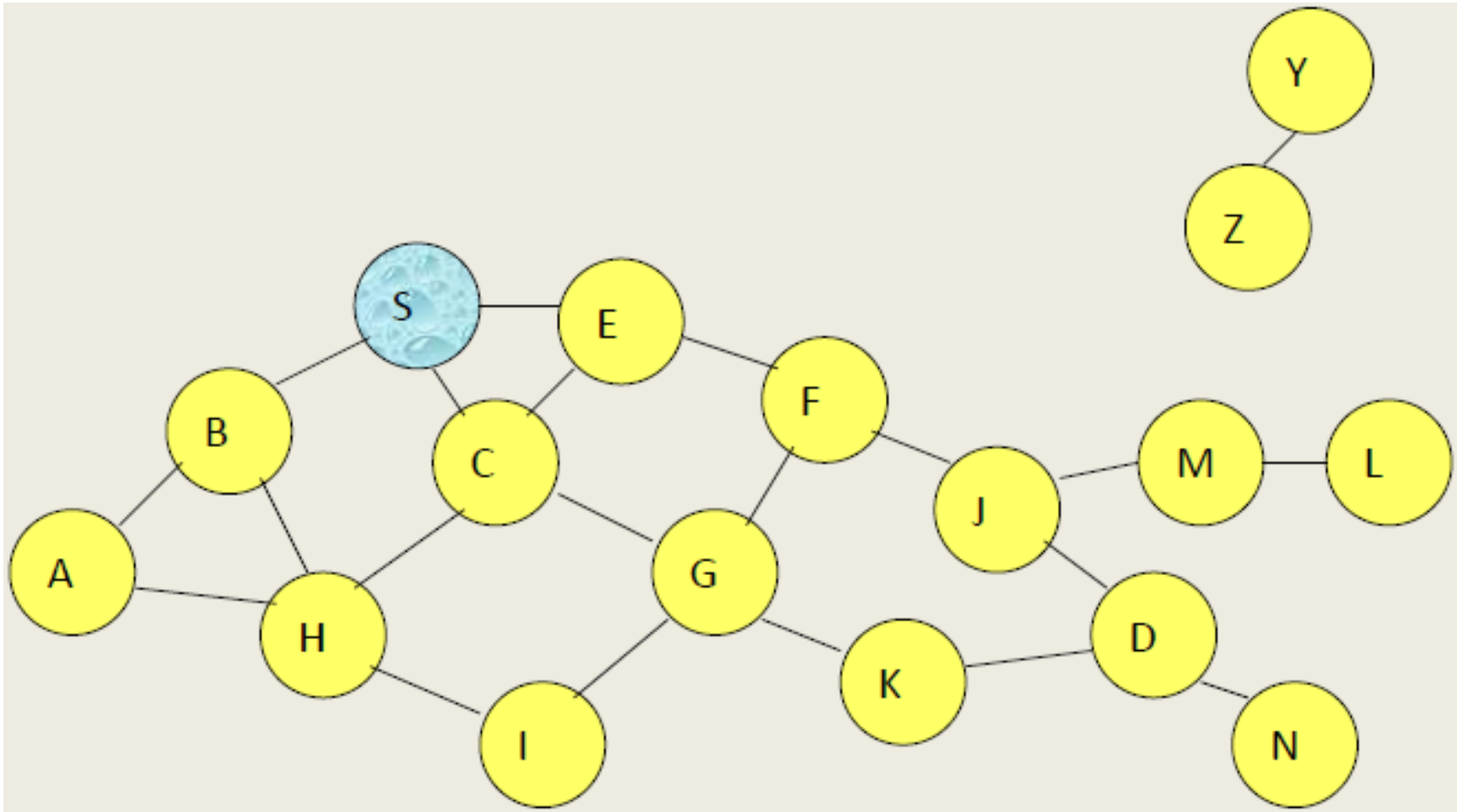
- **Topologia dinâmica**
  - Mudanças frequentes no alcance dos nós
  - Variações na qualidade dos enlaces
  - Nós com diferentes capacidades (energia, velocidade, etc.)

- **Limitação de recursos em redes móveis**
  - Atualizações periódicas em tabelas de roteamento demandam energia e não contribuem diretamente para uma transmissão mais eficiente
  - A largura de banda limitada é reduzida ainda mais devido à necessidade de troca de informações de roteamento
  - Enlaces podem ser assimétricos (N1 pode mandar, mas não receber dados de N2)
- **Principal Problema**
  - Protocolos de roteamento foram projetados para redes cabeadas, em que mudanças são pouco frequentes e enlaces são simétricos

Transferência de dados por inundação (*Flooding*)

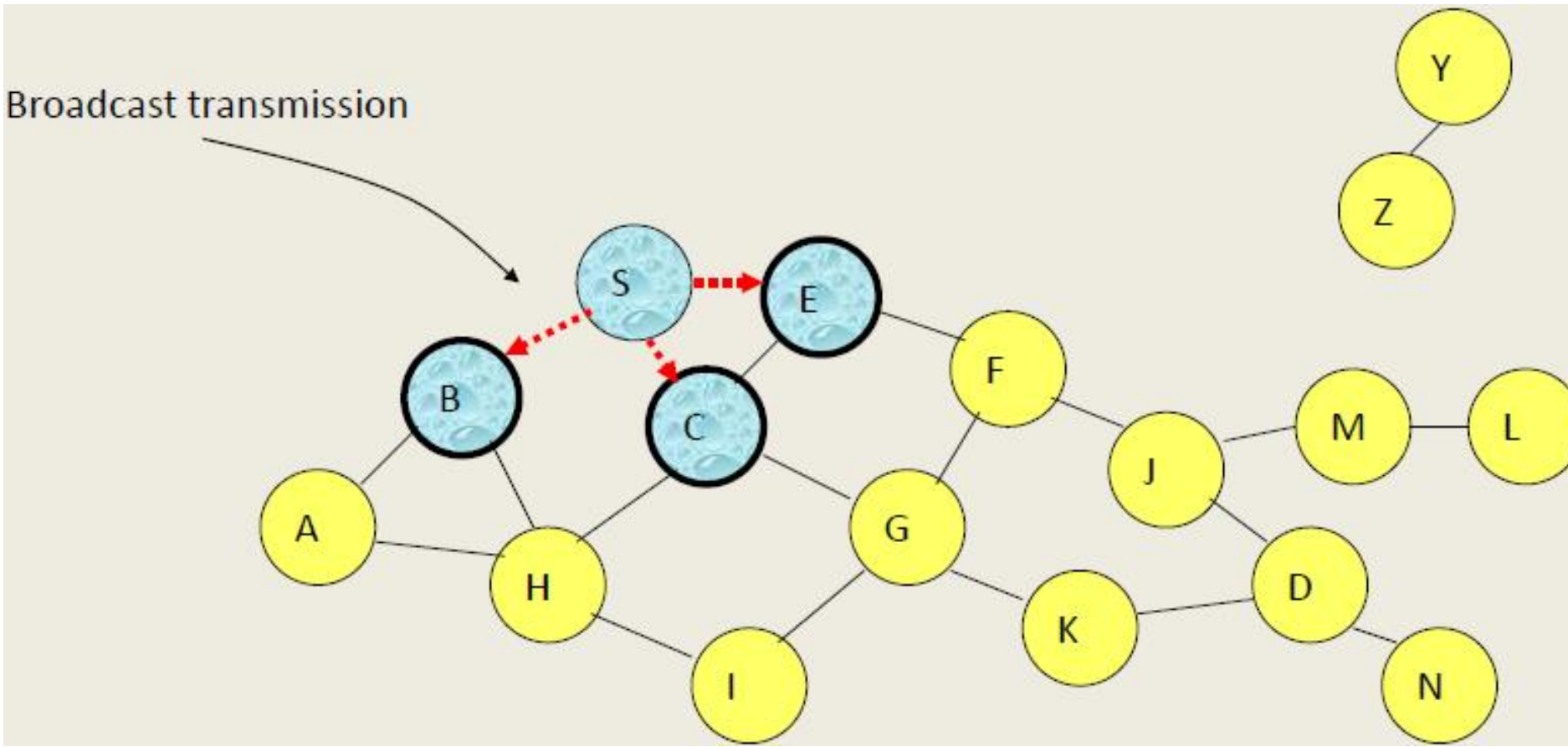
- O transmissor S envia em *broadcast* o pacote de dados *P* para todos os seus vizinhos
- Cada nó que receba P, o encaminha para seus vizinhos
- Números de sequência são utilizados para evitar a possibilidade de encaminhar o mesmo pacote mais de uma vez
- O pacote P alcança o destino D considerando que D é alcançável a partir do transmissor S
- O nó D não encaminha o pacote

Inundação

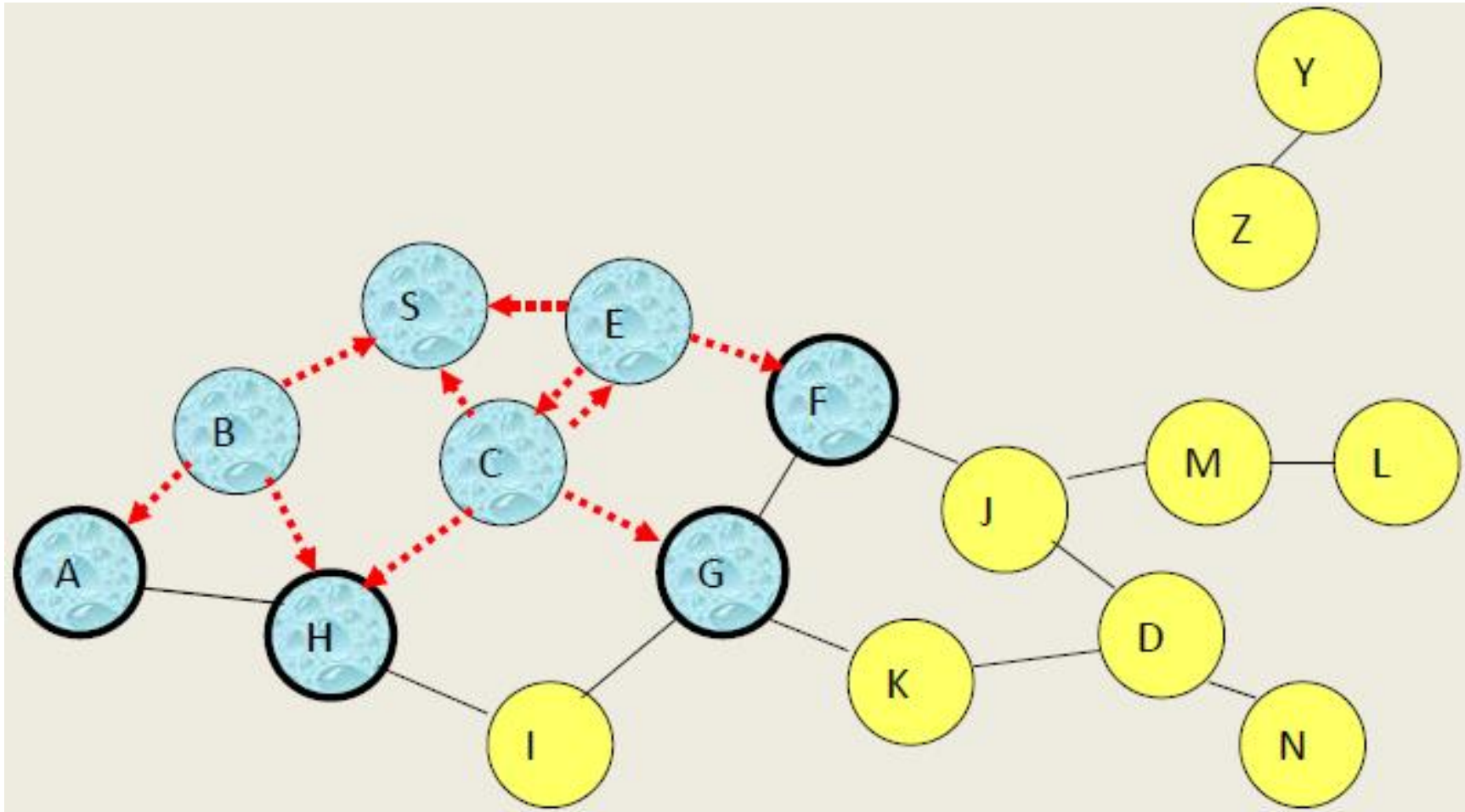


Nó S envia pacote P para D

Inundação

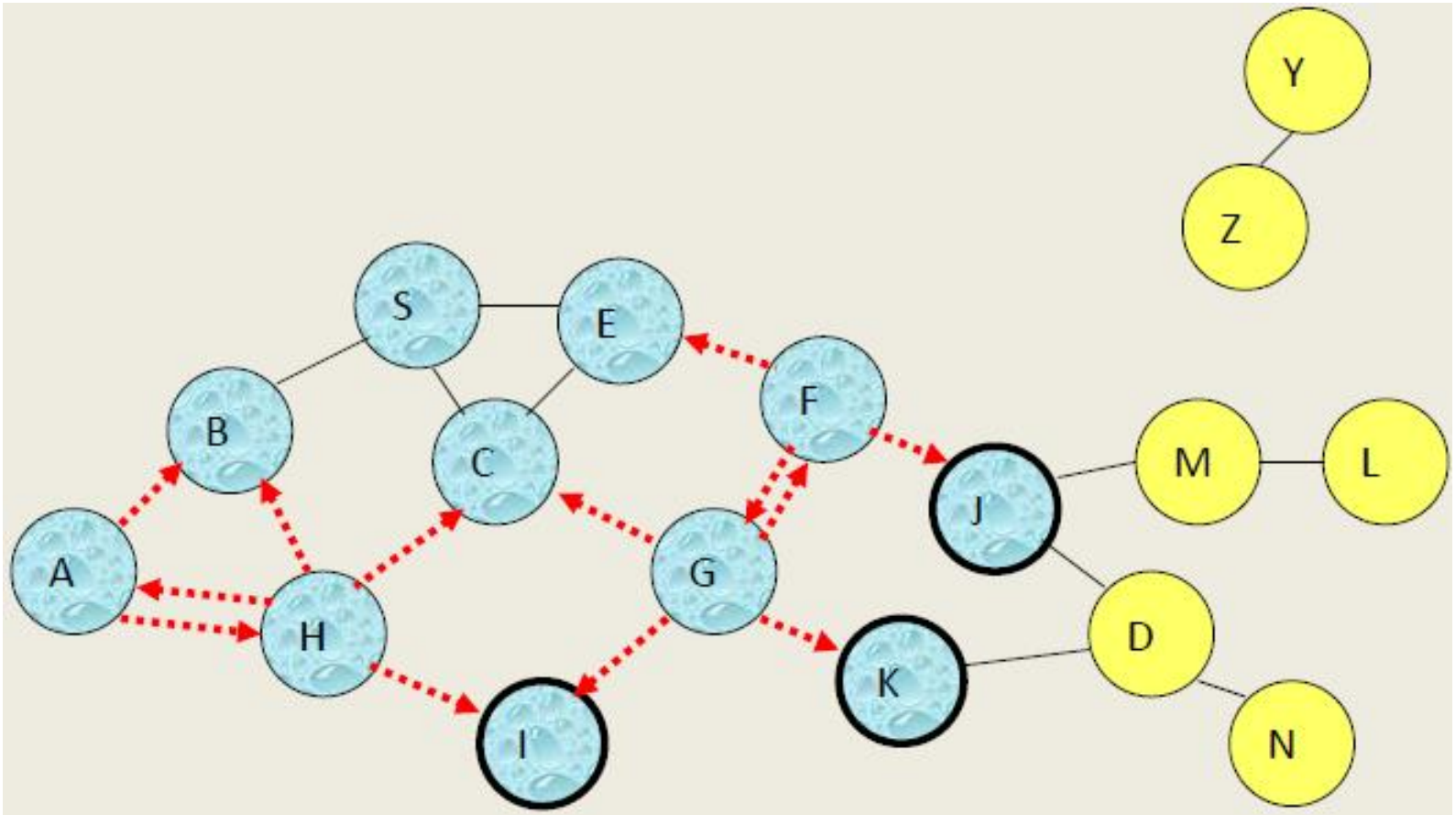


Inundação



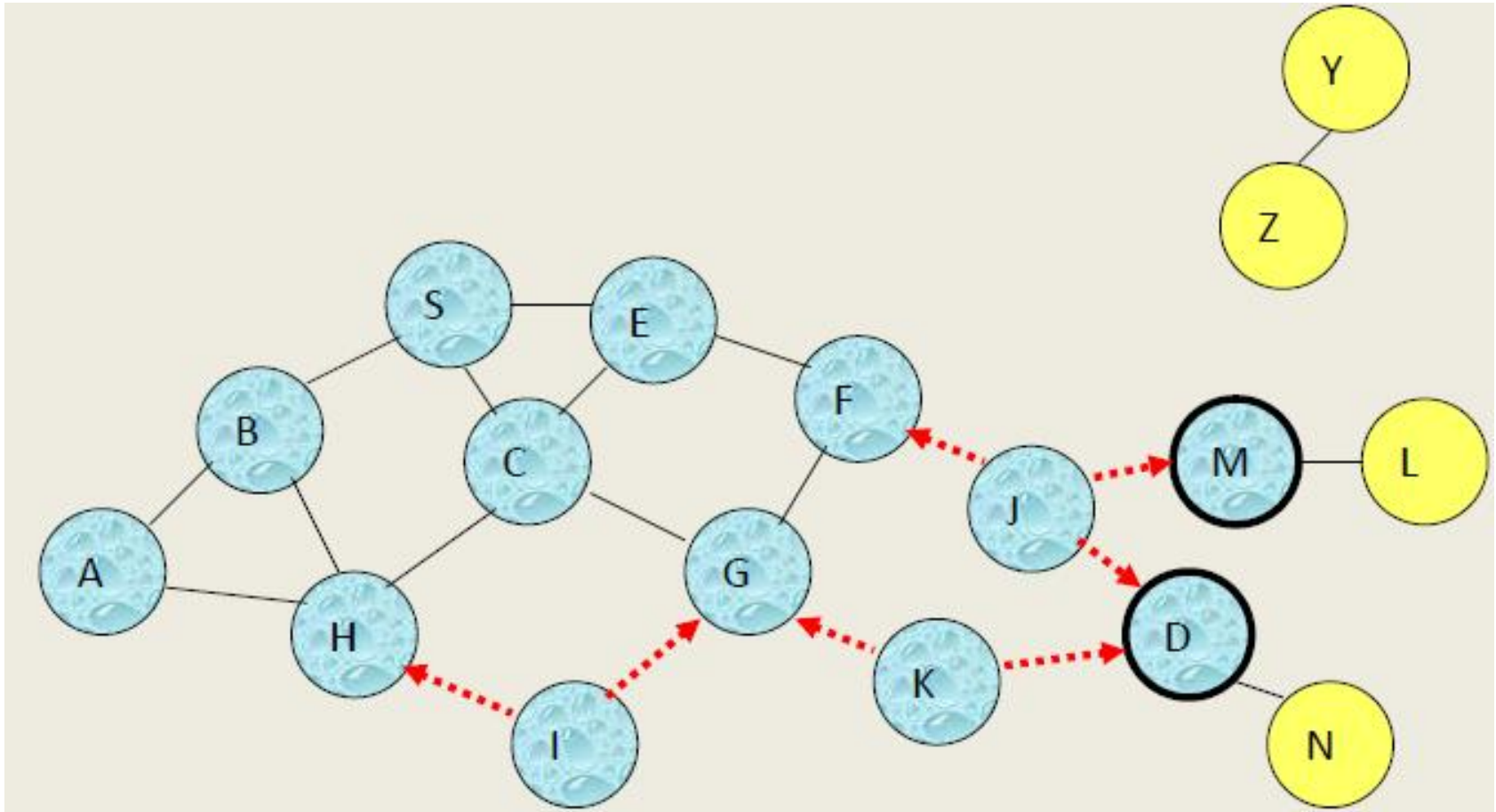
H recebe o pacote de 2 vizinhos: potencial colisão

Inundação



Nó C recebe pacote de G e H, mas não reencaminha.

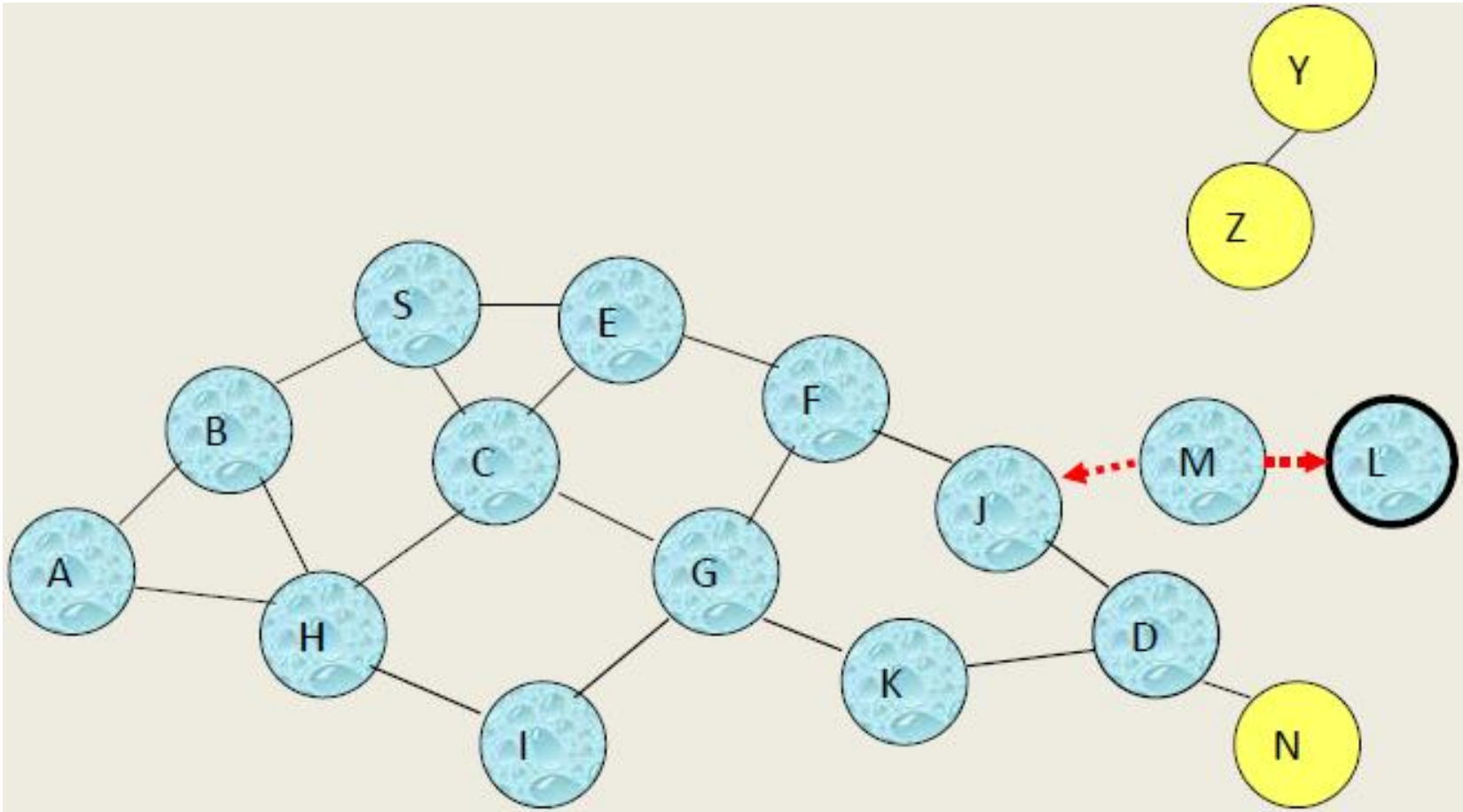
Inundação



Nós J e K enviam pacote a D.

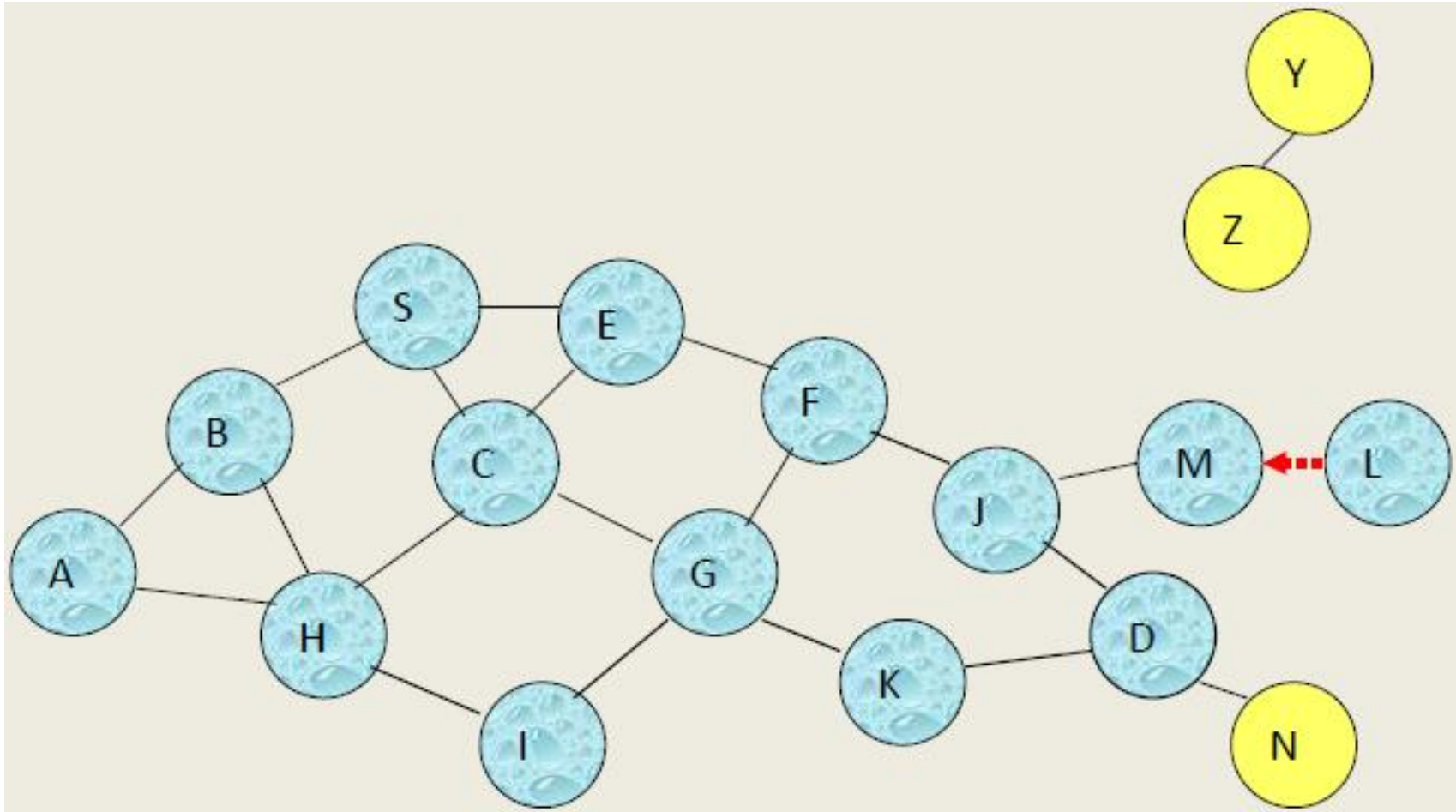
Transmissão pode colidir e não ser entregue a D.

Inundação



Nó D não reencaminha o pacote, pois é o destino.

Inundação



Inundação completa

Alguns nós não recebem o pacote (Y e Z são inalcançáveis e N não tem caminho até S que não passe por D)

Inundação: Vantagens

- Simplicidade
- Pode ser mais eficiente que outros protocolos quando a taxa de envio de dados é mais baixa que o overhead do envio de mensagens de descoberta e manutenção de rotas
- e.g., quando os nós transmitem pequenos pacotes de dados e relativamente infrequentes, e muitas mudanças de topologia ocorrem entre transmissões consecutivas
- Potencialmente alta confiabilidade de entrega de dados
  - Os pacotes podem ser entregues ao destino por diferentes caminhos

Inundação: Desvantagens

- Overhead alto
  - Os pacotes de dados podem ser entregues para muitos nós que não precisam recebê-los
- Potencial baixa confiabilidade na entrega dos dados

  Inundação utiliza broadcast – difícil de implementar broadcast confiável sem aumentar o overhead significativamente

  - No exemplo, os nós J e K podem transmitir para D ao mesmo tempo, resultando em uma colisão
  - Neste caso, o destino pode não receber o pacote

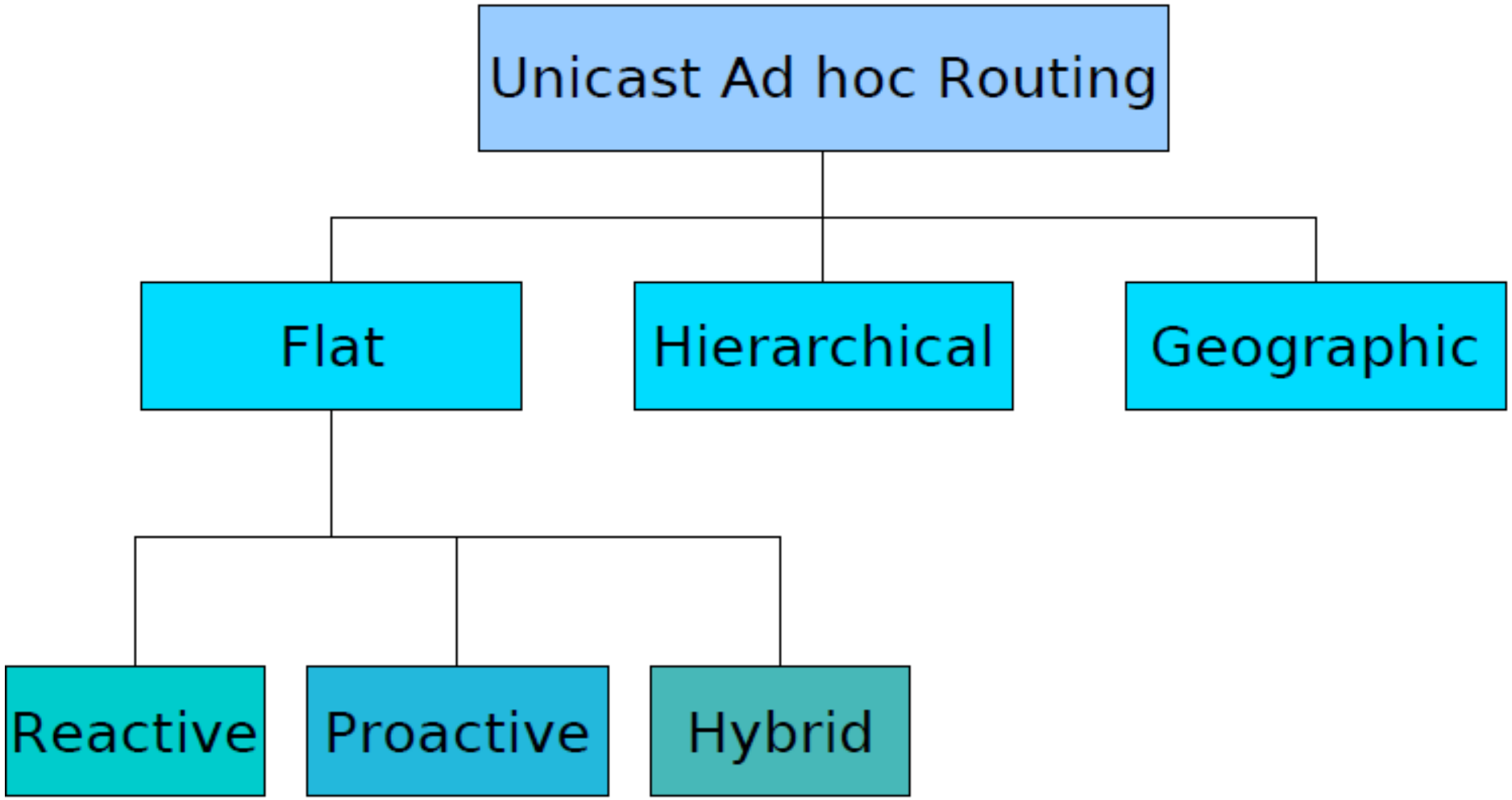
Inundação de pacotes de controle

- Muitos protocolos realizam inundação de pacotes de controle (potencialmente limitado), em vez de pacotes de dados
- Os pacotes de controle são utilizados para descobrir rotas
- Rotas descobertas são subsequentemente utilizadas para enviar pacotes de dados
- O overhead da inundação de pacotes de controle é amortizado pelos pacotes de dados transmitidos entre inundações consecutivas

Requisitos de algoritmos de roteamento

- Dinâmico e descentralizado
- Disseminação de informações de roteamento
  - Caminhos com múltiplos saltos (“*multi-hop”)*
  - Livre de loops durante todo o tempo ou quase livre de loops
  - Overhead de sinalização limitado
- Auto-configuração e adaptação dinâmica de topologia
- Baixo consumo de largura de banda de comunicação e energia
  - Escalável com o número de nós
  - Efeito localizado de mudanças de fluxo ou topologia

Taxonomia


Unicast Ad hoc Routing
Flat
Hierarchical
Geographic
Reactive
Proactive
Hybrid

Taxonomia

- Protocolos Pró-Ativos (*Table-Driven)*
  - Determina rotas independentemente de padrões de tráfego
  - Protocolos tradicionais de estado de enlace e vetor distância são pró-ativos
  - Atualização periódica; alta sobrecarga
- Protocolos Reativos (*On Demand)*
  - Mantém rotas somente se necessárias
- Protocolos Híbridos

Taxonomia

- Geográfico (Baseado em posição)*:*
- Se utiliza da posição física (por exemplo, coordenadas geográficas por GPS) dos nós para estabelecer uma rota

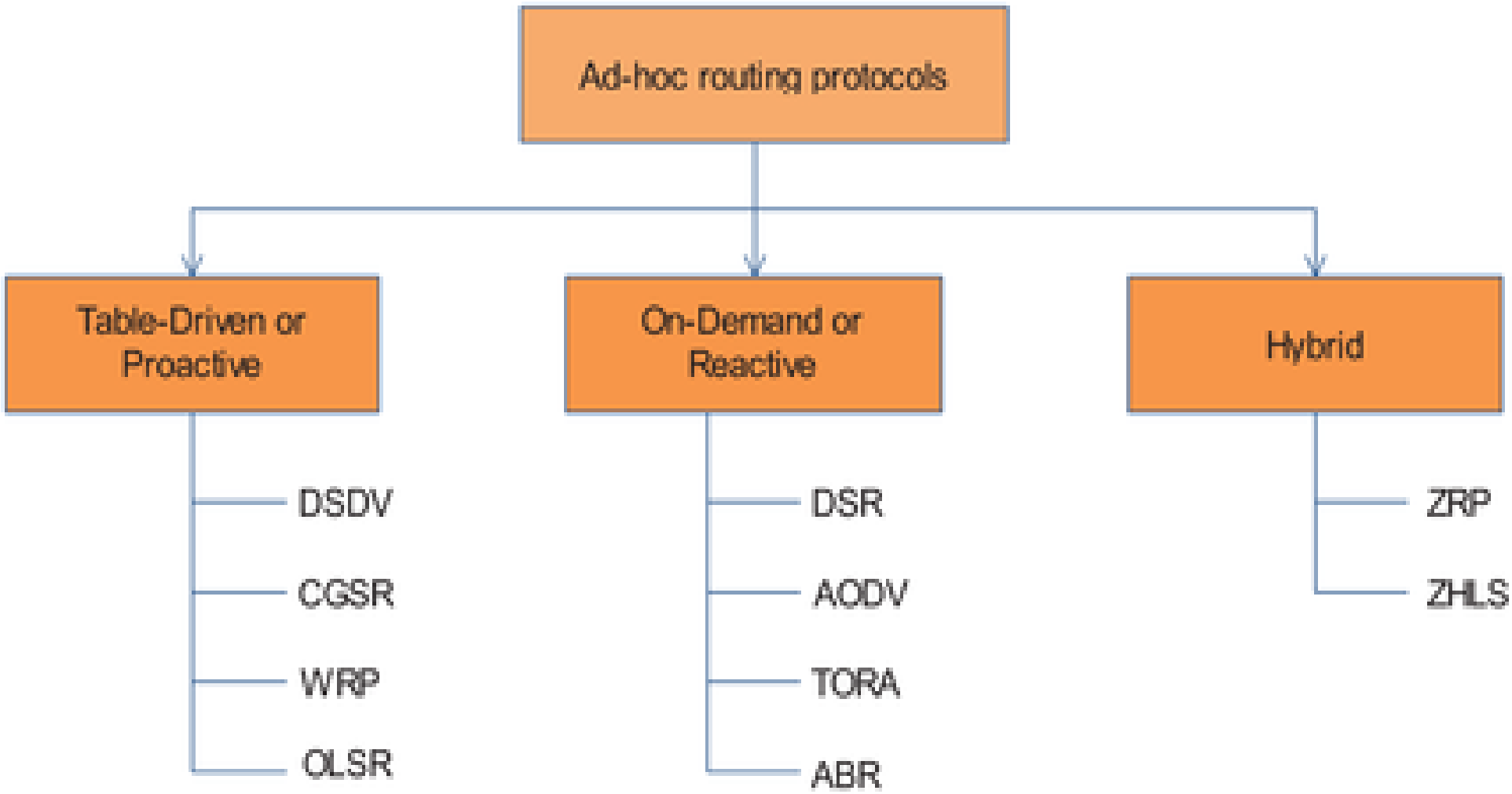
- *Energy-Aware:*
- Leva em consideração a energia contida em cada nó para fazer o roteamento. Este tipo de protocolo é particularmente interessante pois, além de minimizar a energia consumida para enviar um pacote, ele maximiza o tempo de vida da rede

- Podem ser classificados de acordo com a taxonomia apresentada anteriormente

Comparação

	Abordagem Pró-Ativa	Abordagem Reativa
Latência	**Baixa** • As rotas são mantidas todo o tempo	**Alta** • Uma rota nunca é mantida quando não é utilizada
Overhead	**Alta** • Disseminação frequente de informações de topologia é necessária	**Baixa** • Em geral, poucos pacotes de controle são transmitidos

- Não existe um protocolo que opere bem em todos os cenários (de tráfego e mobilidade)


Ad-hoc routing protocols
Table-Driven or Proactive
On-Demand or Reactive
Hybrid
DSDV
CGSR
WRP
OLSR
DSR
AODV
TORA
ABR
ZRP
ZHLS

Roteamento Pró-Ativo

- Tentativa de manter informações de roteamento entre quaisquer par de nós consistentes e atualizadas
  - Adequado para tráfego aleatório entre pares de nós

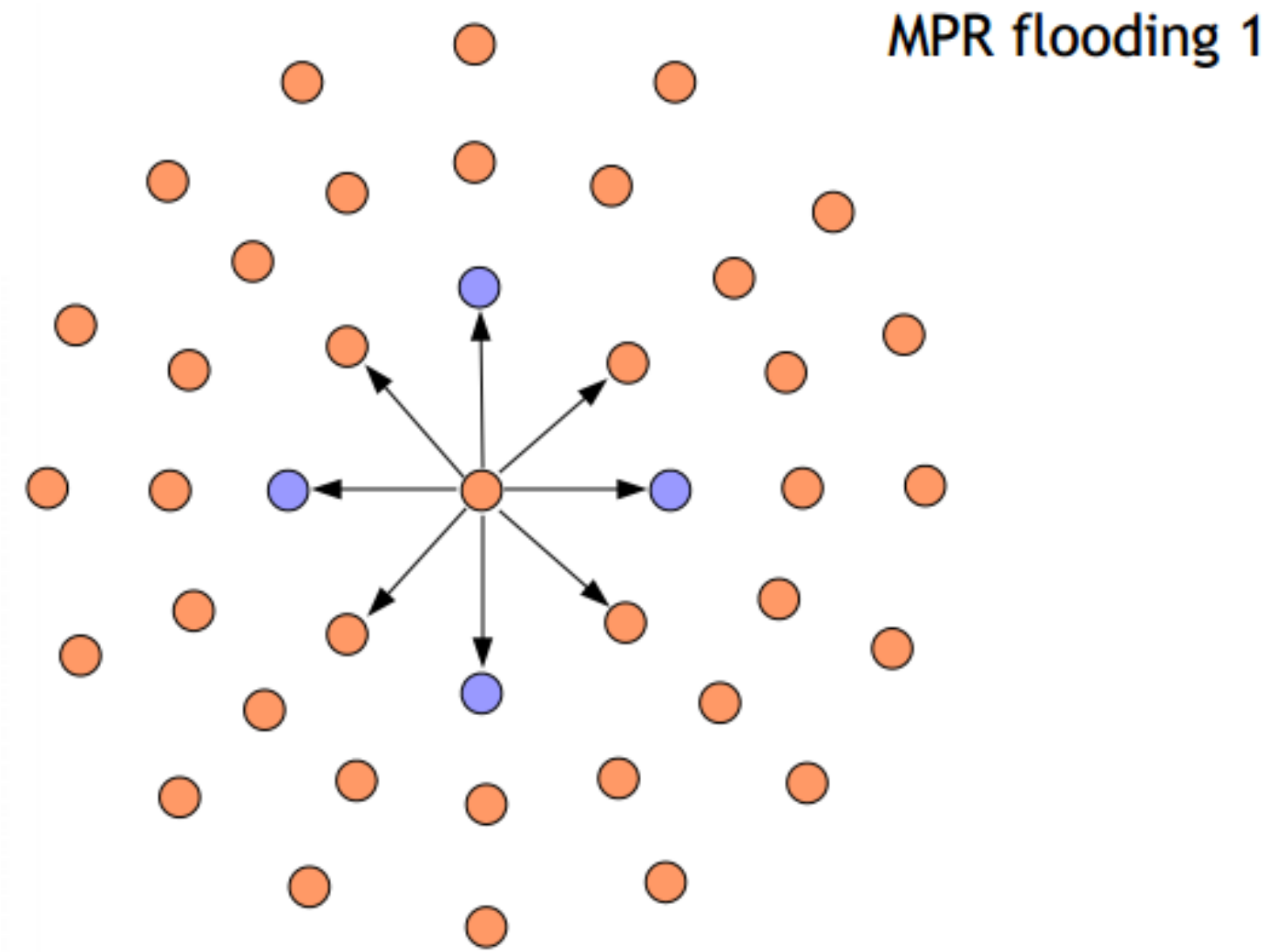
- Objetivo: Reagir às mudanças da topologia de rede, propagando as atualizações, a fim de obter uma visão consistente das distâncias/custos de roteamento na rede

- Apropriado para aplicações que geram um tráfego constante e bem distribuído entre os nós da rede ad hoc

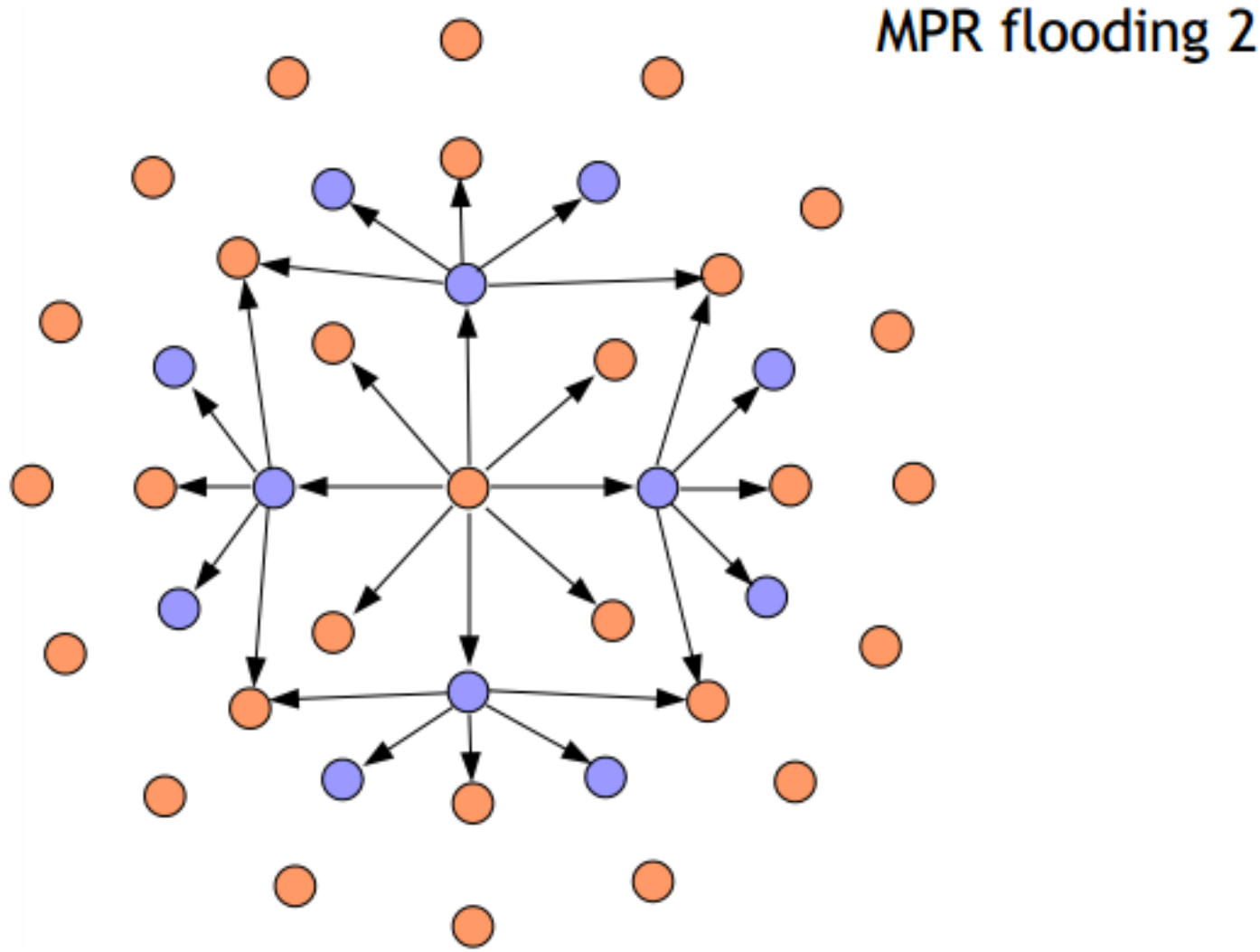
OLSR (Optimized Link State Routing)

- Protocolo de estado de enlace tradicional otimizado para MANETs
- Multi-Point Relays (MPRs) reduzem overhead
- OLSR tem duas otimizações:
  - Minimiza tamanho das mensagens: apenas MPRs são declarados nas mensagens
  - Minimiza o número de nós emitindo mensagens

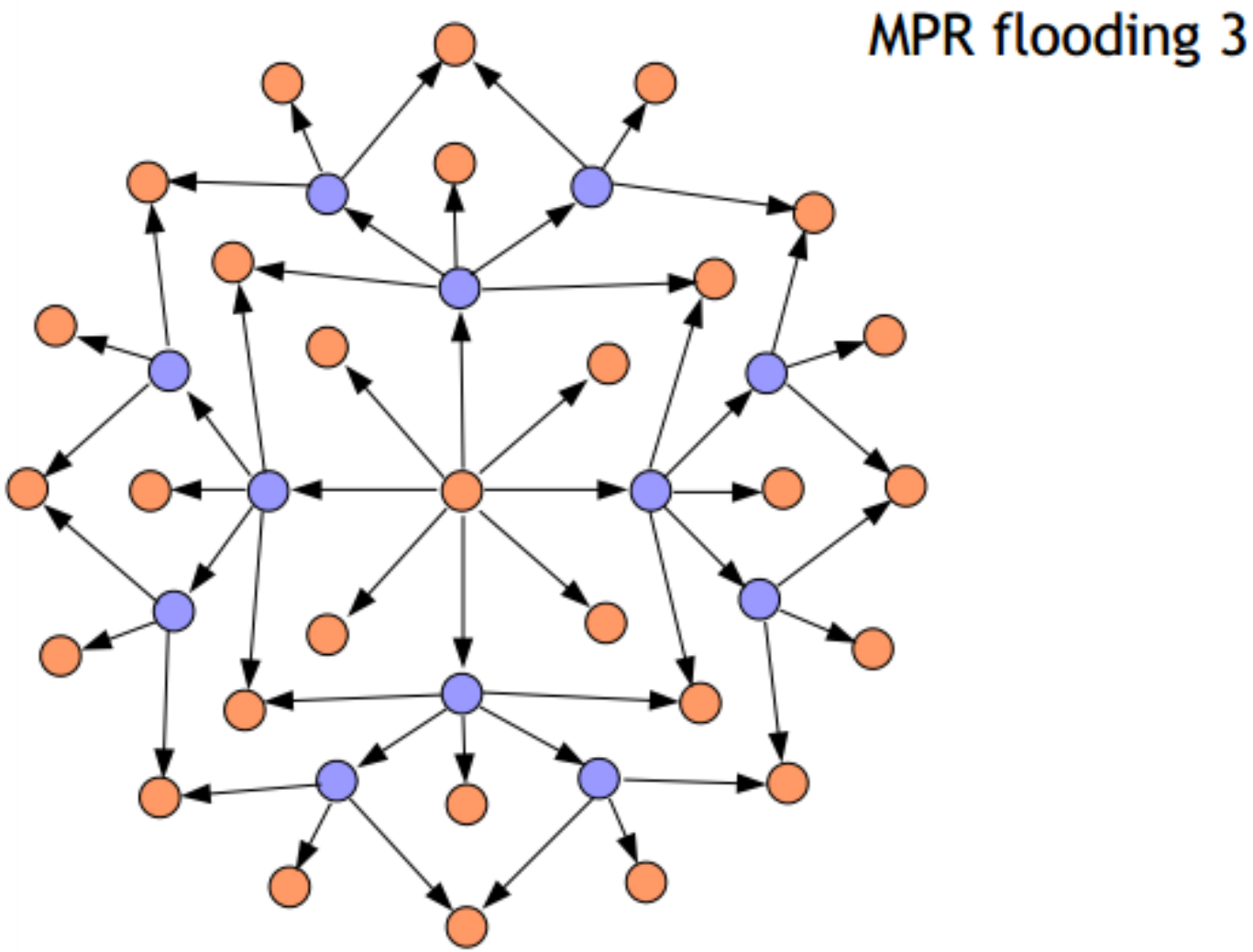
Multipoint relaying – exemplo OLSR



Multipoint relaying – exemplo OLSR



Multipoint relaying – exemplo OLSR



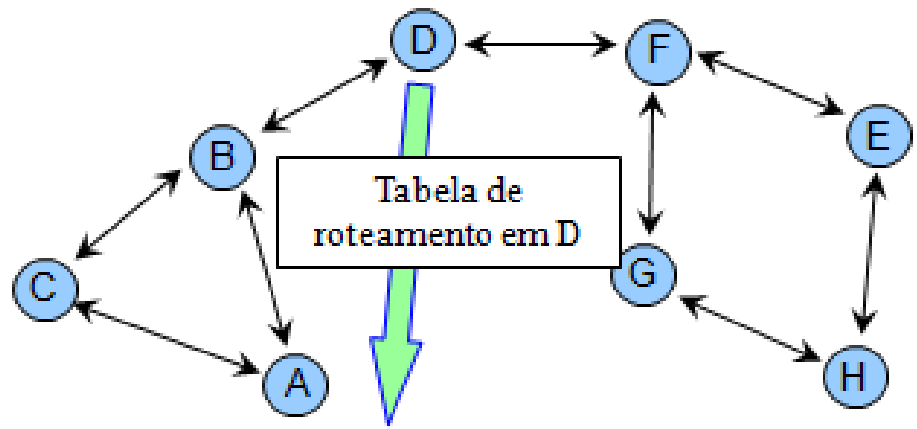
DSDV - *Destination Sequenced Distance Vector Routing*

- Adaptação do *Routing Information Protocol* (RIP) para redes MANET
- Cada nó mantém uma tabela de roteamento com o registro de todos os possíveis destinos e o número de hops para cada um destes destinos
- Além disso, cada entrada na tabela contém um número de sequência, que é usado para distinguir “rotas antigas” de “rotas novas”

DSDV

- Periodicamente, os nós enviam *Routing Advertisements* (RA) (com suas tabelas de roteamento) para todos os seus vizinhos
  - Mudanças de conectividade se difundem aos poucos
- RAs podem ser de 2 tipos:
  - "full dump": envio de toda a tabela de roteamento -> gera muito tráfego
  - Incremental: atualizações contém apenas as mudanças desde o último "full dump"
- Cada nó espera um certo tempo antes de atualizar as entradas antigas por entradas mais recentes a fim de "confirmar" se a mudança de topologia se estabilizou

DSDV - Exemplo



Destination	Next Hop	Metric	Sequence No	Install Time
A	B	2	S406_A	T001_D
B	B	1	S128_B	T001_D
C	B	2	S564_C	T001_D
D	D	0	S710_D	T001_D
E	F	2	S392_E	T002_D
F	F	1	S076_F	T001_D
G	F	2	S128_G	T002_D
H	F	3	S050_H	T002_D

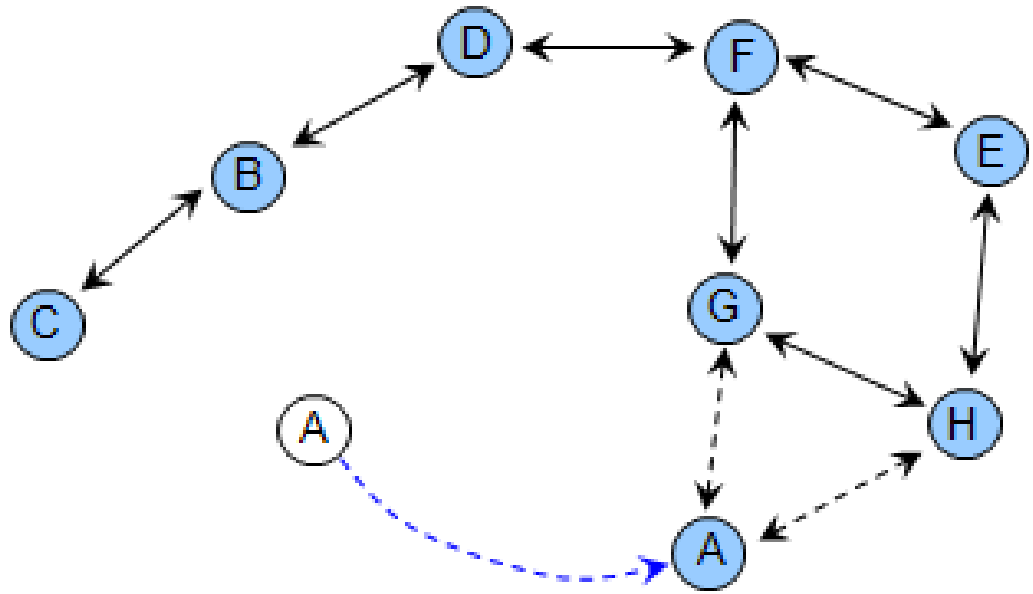
D's Advertised Routing Table		
Destination	Metric	Sequence No
A	2	S406_A
B	1	S128_B
C	2	S564_C
E	2	S392_E
F	1	S076_F
G	2	S128_G
H	3	S050_H

Métrica= número de saltos (hops) até o destino

Número de sequência = "freshness" da rota recebida, usada para evitar loops

Tempo de instalação= indica quando uma rota foi instalada, para minimizar flutuações de rota

DSDV– Exemplo (movimento de A)



- Quando A se move e é detectado por G e H, estes nós avisam com informações atualizadas de roteamento (incremental)
- Quando F recebe esta informação, atualiza sua tabela de roteamento e faz um broadcasta da nova informação
- D recebe esta atualização e modifica sua tabela de roteamento

DSDV– Exemplo (movimento de A)

D's Updated Routing Table				
Destination	Next Hop	Metric	Sequence No	Install Time
A	**F**	**3**	**S516_A**	**T810_D**
B	B	1	S238_B	T001_D
C	B	2	S674_C	T001_D
D	D	0	S820_D	T001_D
E	F	2	S502_E	T002_D
F	F	1	S186_F	T001_D
G	F	2	S238_G	T002_D
H	F	3	S160_H	T002_D

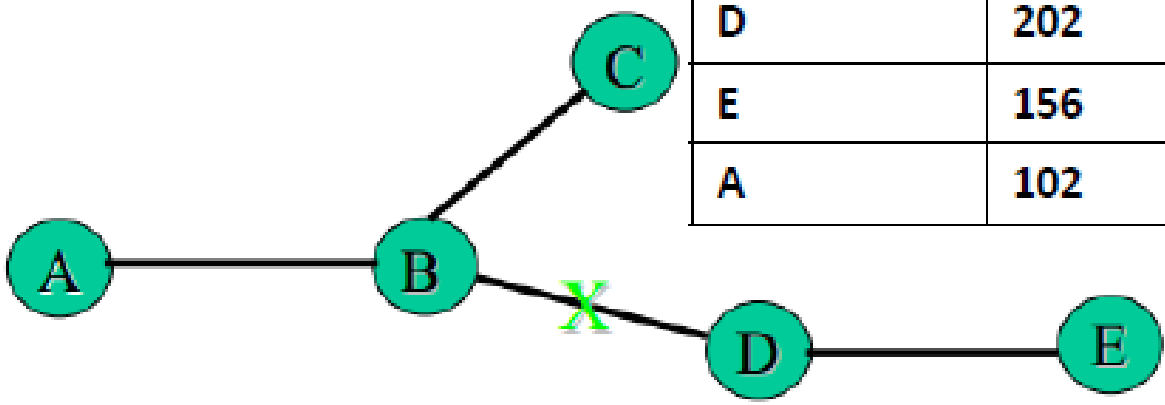
D's Updated Advertised Routing Table		
Destination	Metric	Sequence No
A	**3**	**S516_A**
B	1	S238_B
C	2	S674_C
E	2	S502_E
F	1	S186_F
G	2	S238_G
H	3	S160_H

DSDV – Quebra de Enlace

DSDV: Link Breaks

B's Route Table:

Destination Address	Destination Seq #	Next_Hop	Hop_count
B	132	B	0
C	144	C	1
D	202	D	1
E	156	D	2
A	102	A	1



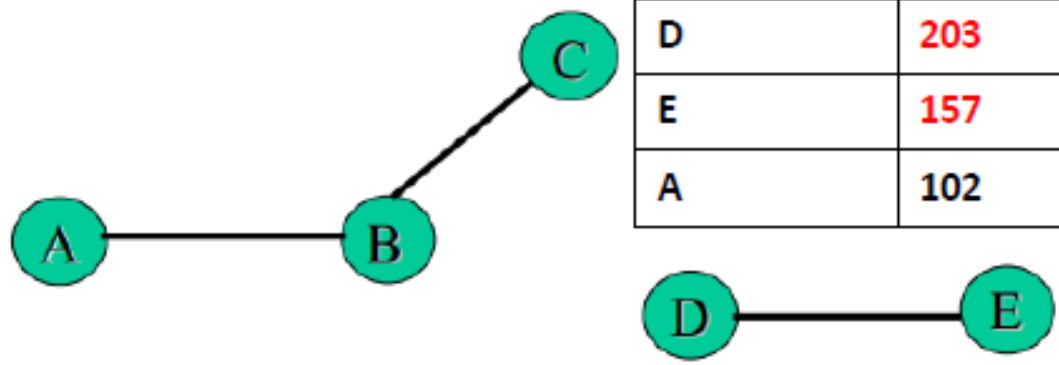
1. Link between B and D breaks

DSDV – Quebra de Enlace

DSDV: Link Breaks

B's Route Table:

Destination Address	Destination Seq #	Next_Hop	Hop_count
B	132	B	0
C	144	C	1
D	203	D	□
E	157	D	□
A	102	A	1



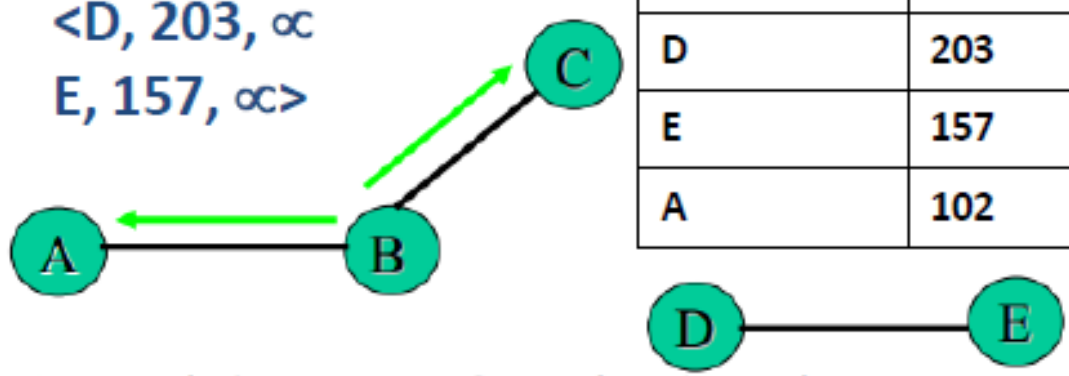
1. Link between B and D breaks
2. Node B notices break
   1. Updates hopcount for D & E to infinity
   2. Increments seq# for D & E

DSDV – Quebra de Enlace

DSDV: Link Breaks

B's Route Table:

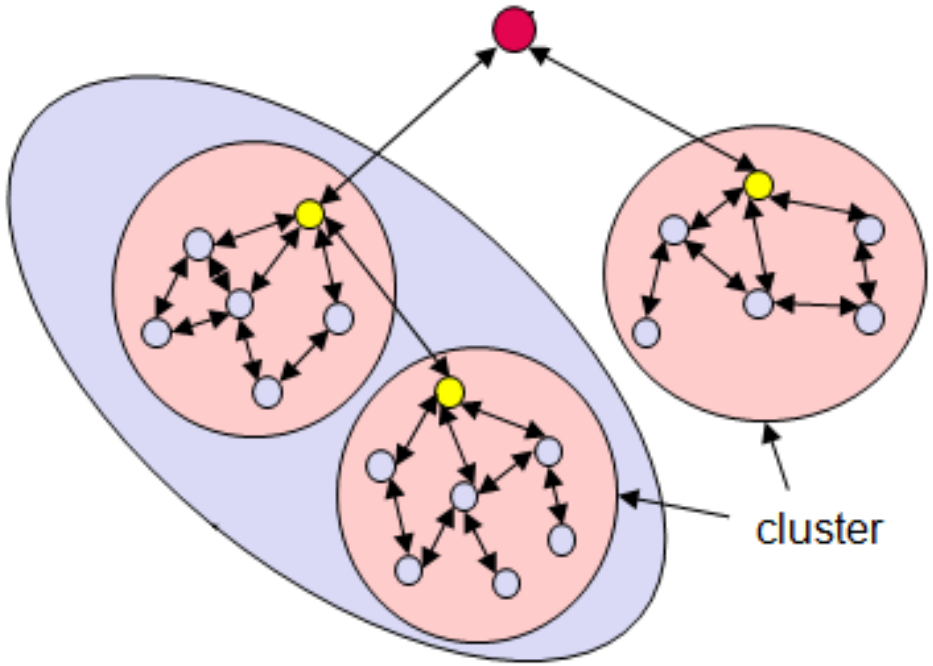
Destination Address	Destination Seq #	Next_Hop	Hop_count
B	132	B	0
C	144	C	1
D	203	D	∞
E	157	D	∞
A	102	A	1



1. Link between B and D breaks
2. Node B notices break
   1. Updates hopcount for D & E to infinity
   2. Increments seq# for D & E
3. Node B sends update with new route information

Clusterhead Gateway Switch Routing (CGSR)

- Nós organizados em hierarquia de clusters
- Cada nó tem um cluster head.
- Nós enviam pacotes pelos cluster heads.
- Cluster heads se comunicam usando DSDV.



Roteamento Reativo

- Uma rota somente é criada quando o nó fonte precisa enviar para o destino:
  - Quando um nó precisa de uma rota, inicia o processo de descoberta de rota
  - Este processo termina quando a rota é descoberta, ou quando todas os possíveis caminhos tiveram sido analisados

- Uma vez que a rota é criada, esta é mantida por um protocolo de manutenção
- Não há necessidade de atualizações periódicas! É adequado para MANETs com demanda de comunicação esporádica e seletiva (apenas entre alguns pares de nós) e em que a topologia apresenta uma variabilidade baixa

Roteamento Reativo

- Descoberta de Rotas:
- Difunde (broadcast para todos vizinhos diretos) um *Route Request* (RREQ) com endereço destino e identificação única ID
  - Quando um nó recebe este broadcast:
  - Se o próprio é o destinatário então envia ao remetente uma resposta RREP (contendo a rota registrada em RREQ)
  - Se RREQ já foi recebido anteriormente (detecção através do ID) então simplesmente descarta o pacote
  - Senão, adiciona o próprio endereço na lista de hops (no pacote) e também difunde RREQ para seus vizinhos

- O Remetente em algum momento recebe RREP contendo a rota descoberta

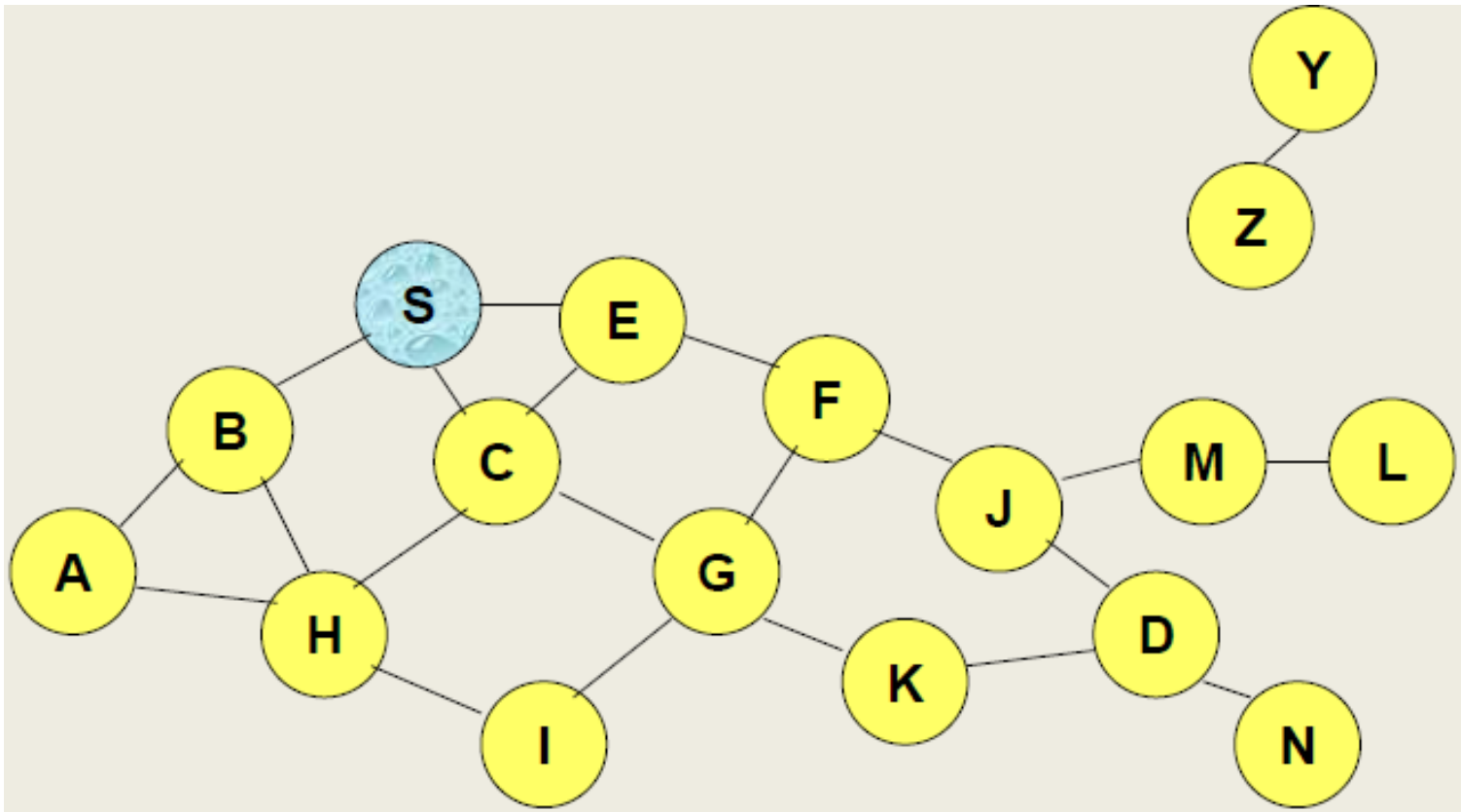
DSR – *Dynamic Source Routing*

- Quando o nó S deseja enviar um pacote para D, e não conhece uma rota para D, o nó S inicia uma descoberta de rota

- S inunda a rede com um *Route Request* (RREQ).

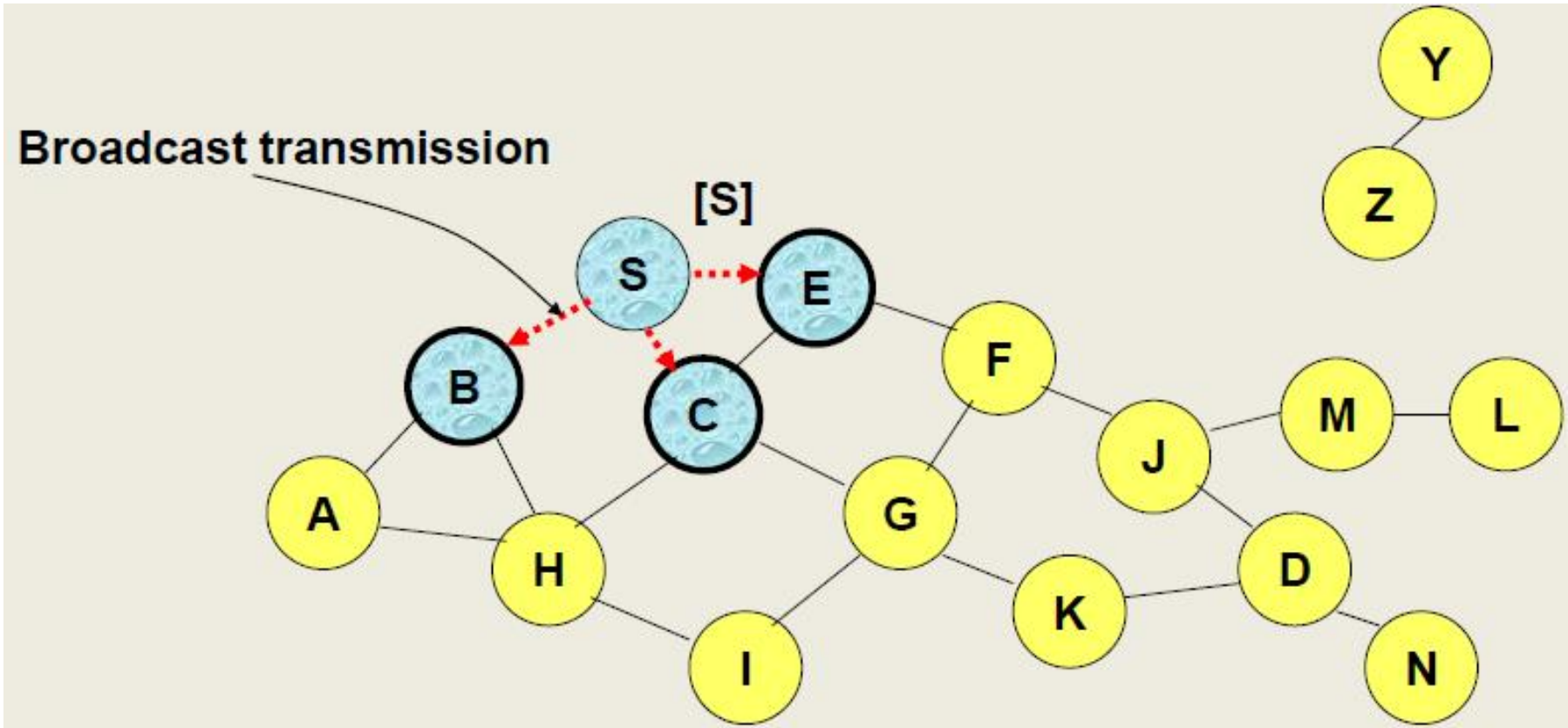
- Cada nó adiciona seu próprio identificador quando encaminhar o RREQ

Descoberta de rotas no DSR

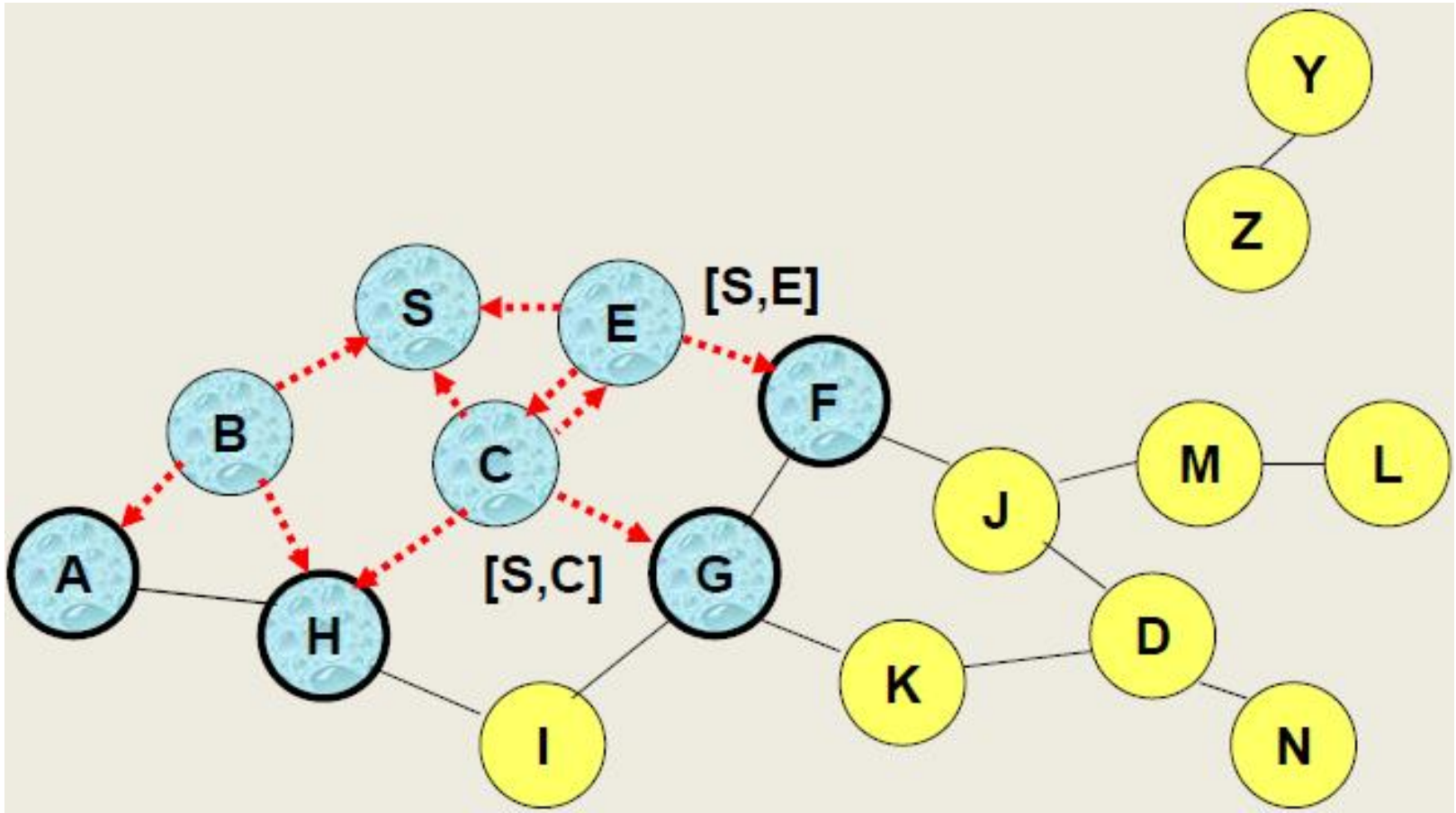


Nó S envia RREQ para descobrir destino D

Descoberta de rotas no DSR

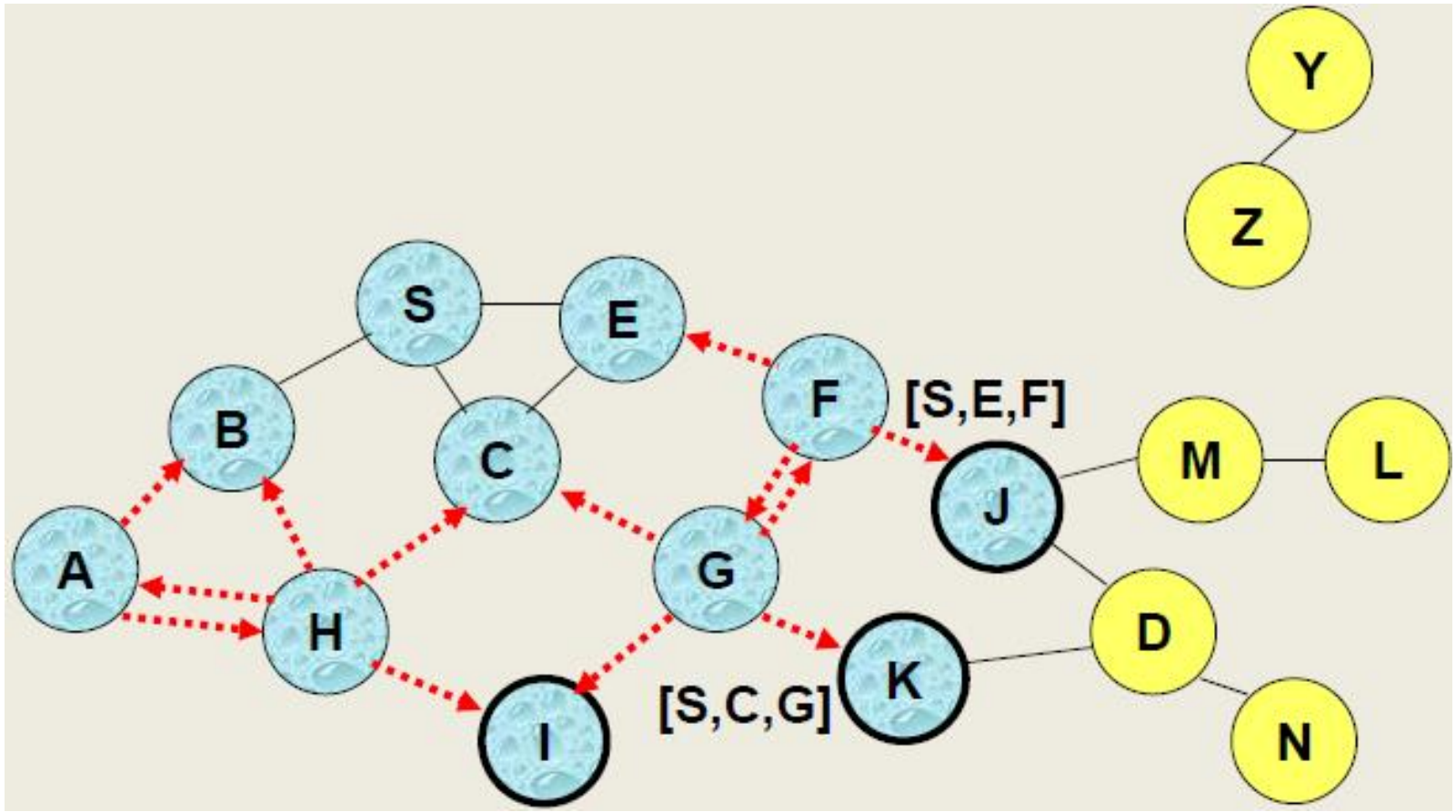


Descoberta de rotas no DSR



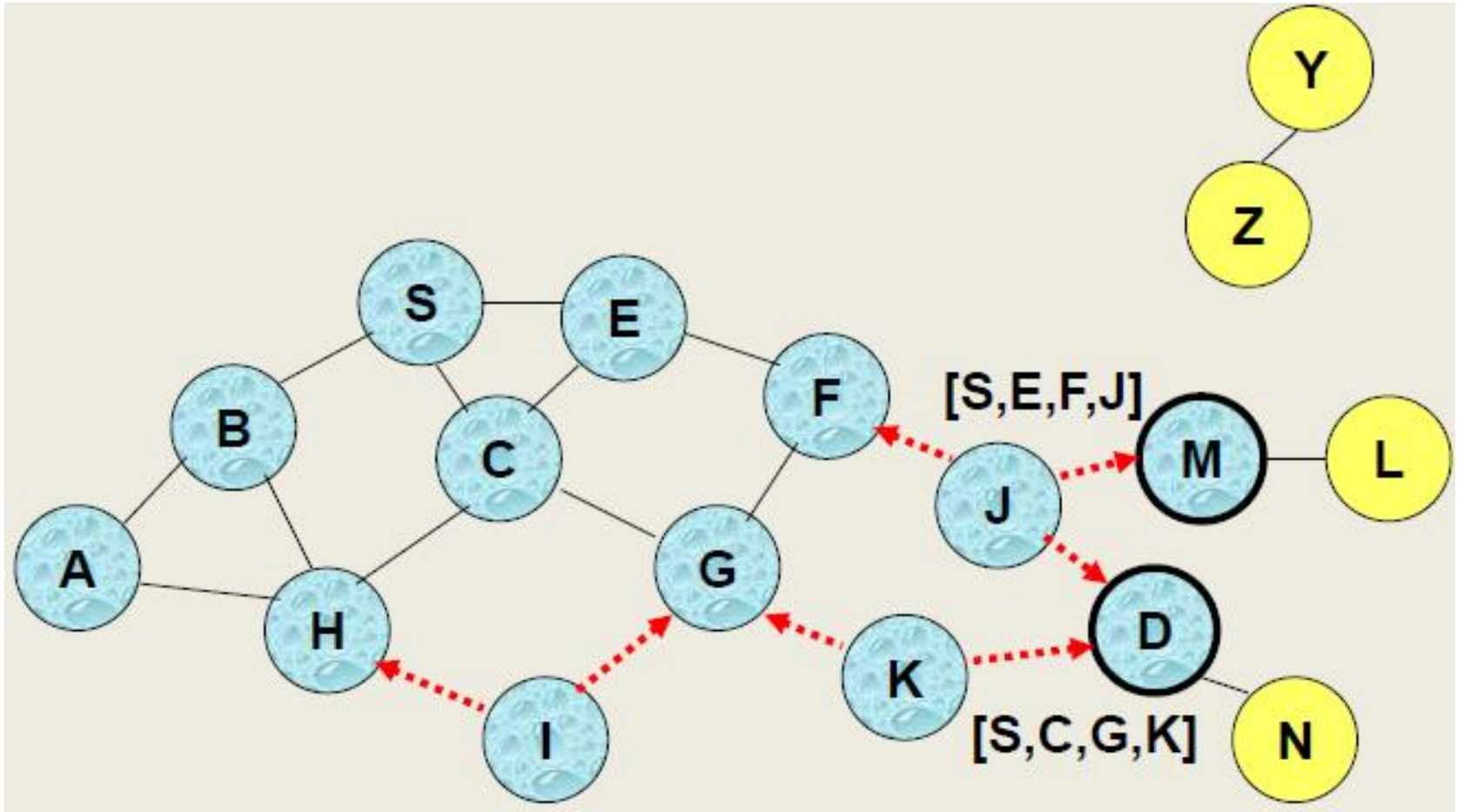
H recebe RREQ de dois vizinhos: potencial colisão

Descoberta de rotas no DSR



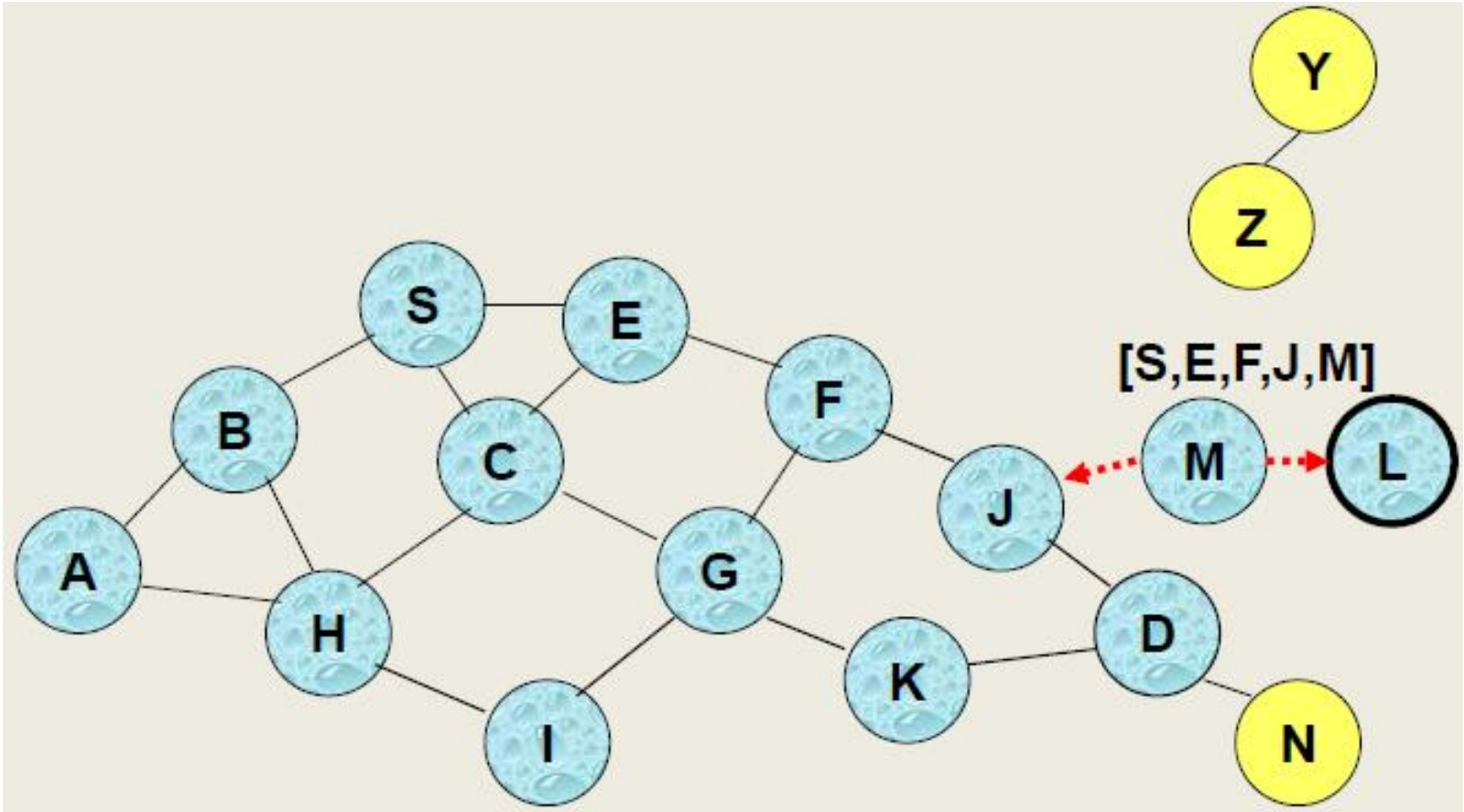
C recebe RREQ de G e H, mas não encaminha novamente (já enviou RREQ antes)

Descoberta de rotas no DSR



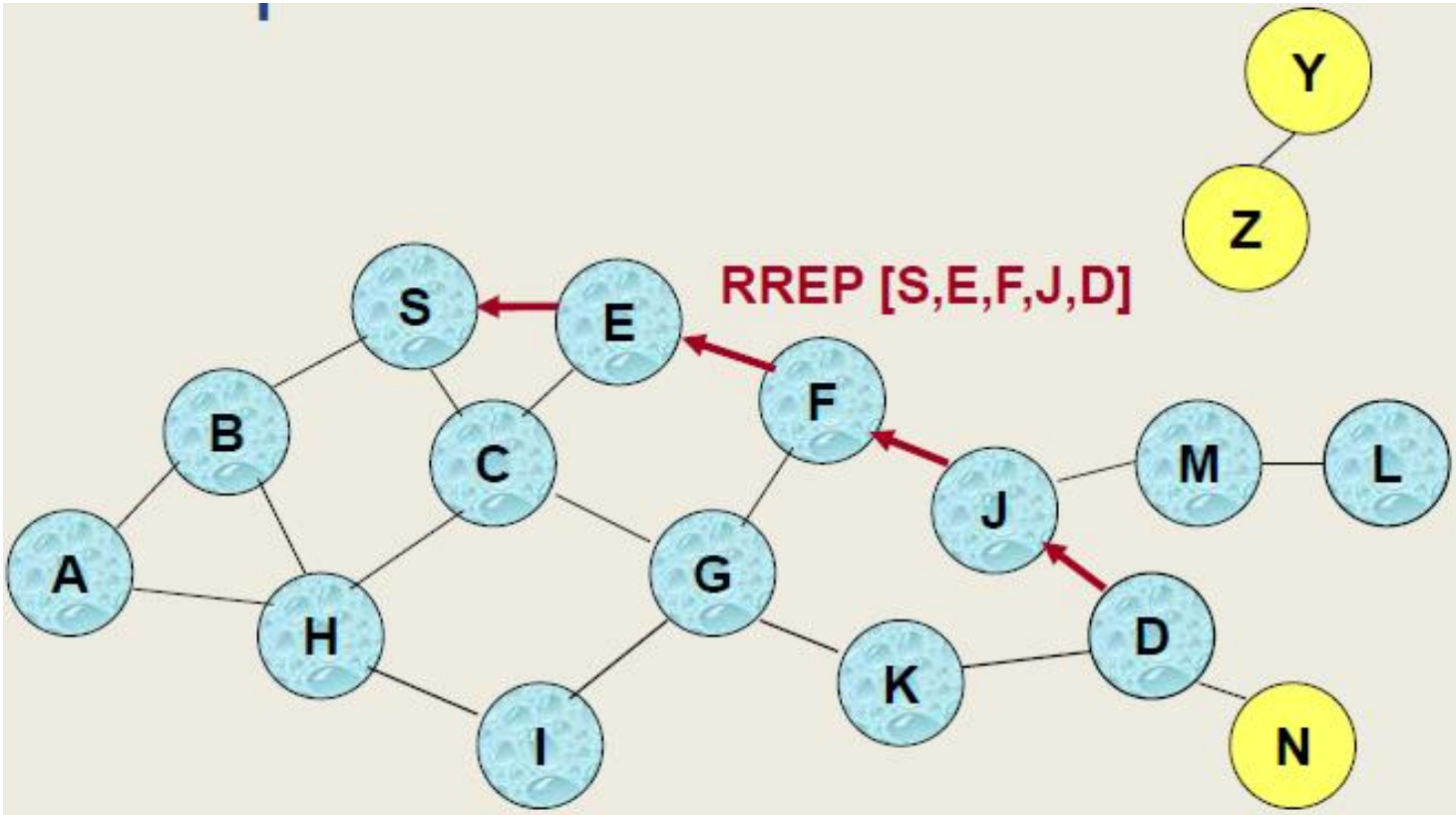
- J e K fazem o broadcast de RREQ ao nó D
- Como J e K estão “escondidos” um do outro, transmissões podem colidir

Descoberta de rotas no DSR



- D não encaminha RREQ, pois é o destino da descoberta de rota

Resposta de rota no DSR

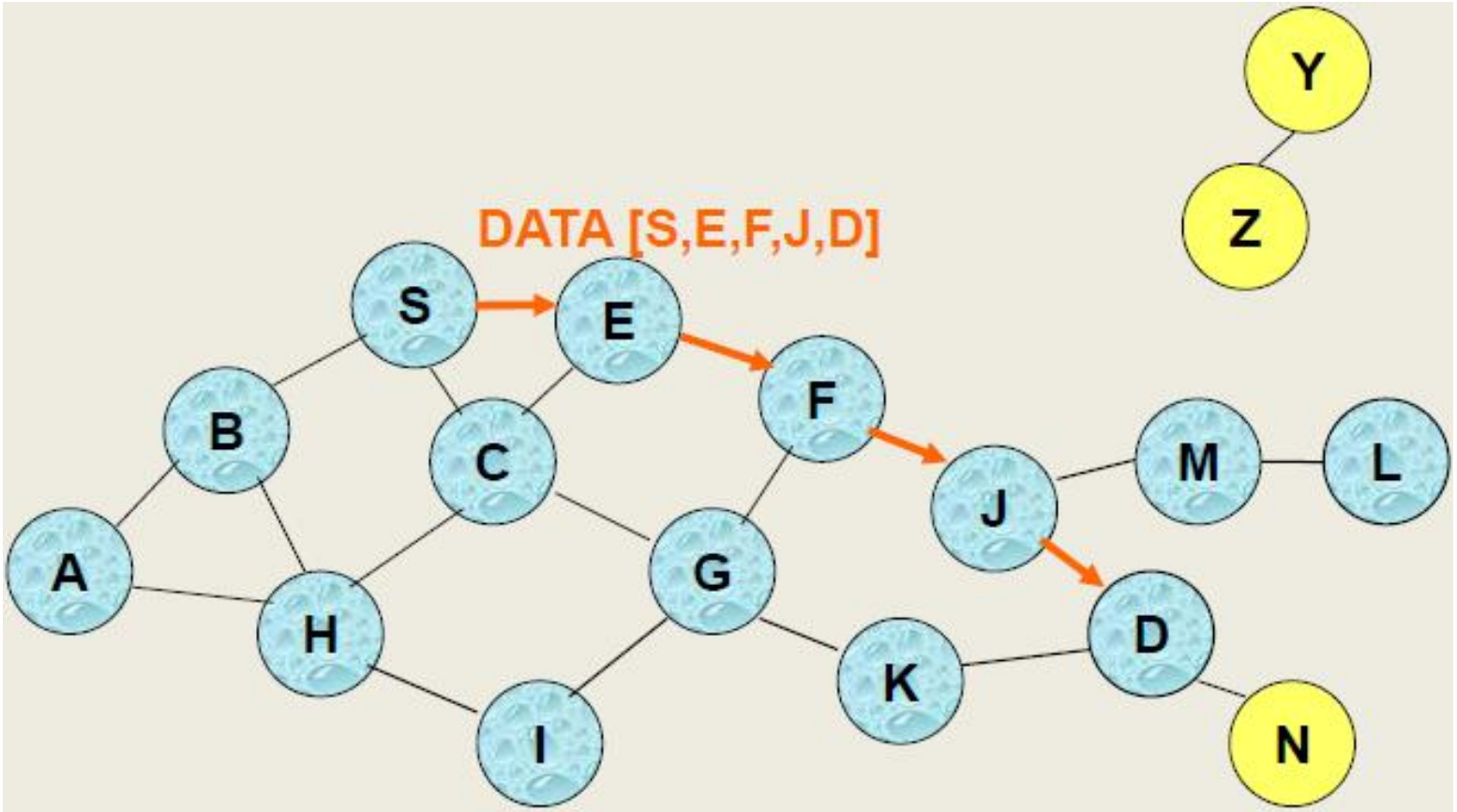


- O destino D, ao receber o primeiro RREQ, envia um *Route Reply* (RREP)
- RREP é enviado em uma rota obtida invertendo a rota adicionada ao RREQ recebido

Resposta de Rota no DSR

- RREP pode ser enviado invertendo a rota contida no RREQ somente se os enlaces forem garantidamente bidirecionais
- Se enlaces unidirecionais (assimétricos) forem permitidos, então o RREP pode precisar de uma descoberta de rota para S a partir de D
  - A menos que D já conheça uma rota para S
  - Se uma descoberta de rota é iniciada por D para obter uma rota a S, então o RREP é incluído no conteúdo do RREQ de D

Envio de dados no DSR



- Tamanho do cabeçalho do pacote aumenta com o comprimento da rota

DSR - Vantagens

- Rotas mantidas somente entre nós que precisam se comunicar
  - Reduz o overhead de manutenção de rotas
- Caching de rotas pode reduzir o overhead da descoberta de rotas
- Uma única descoberta de rota pode produzir muitas rotas ao destino, devido ao cache local dos nós intermediários

DSR - Desvantagens

- Cabeçalho do pacote cresce com o comprimento da rota
- Inundação de requisições de rota podem potencialmente alcançar todos os nós da rede
  - Cuidado deve ser tomado para evitar colisões entre requisições de rotas propagadas por nós vizinhos
- Um nó intermediário pode enviar um RREP contendo uma rota obsoleta de seu cache, poluindo outros caches
- Contenção aumenta caso muitas respostas de rota voltem ao transmissor devido a múltiplos nós respondendo utilizando seu cache local
  - Problema da “tempestade de RREPs”

AODV

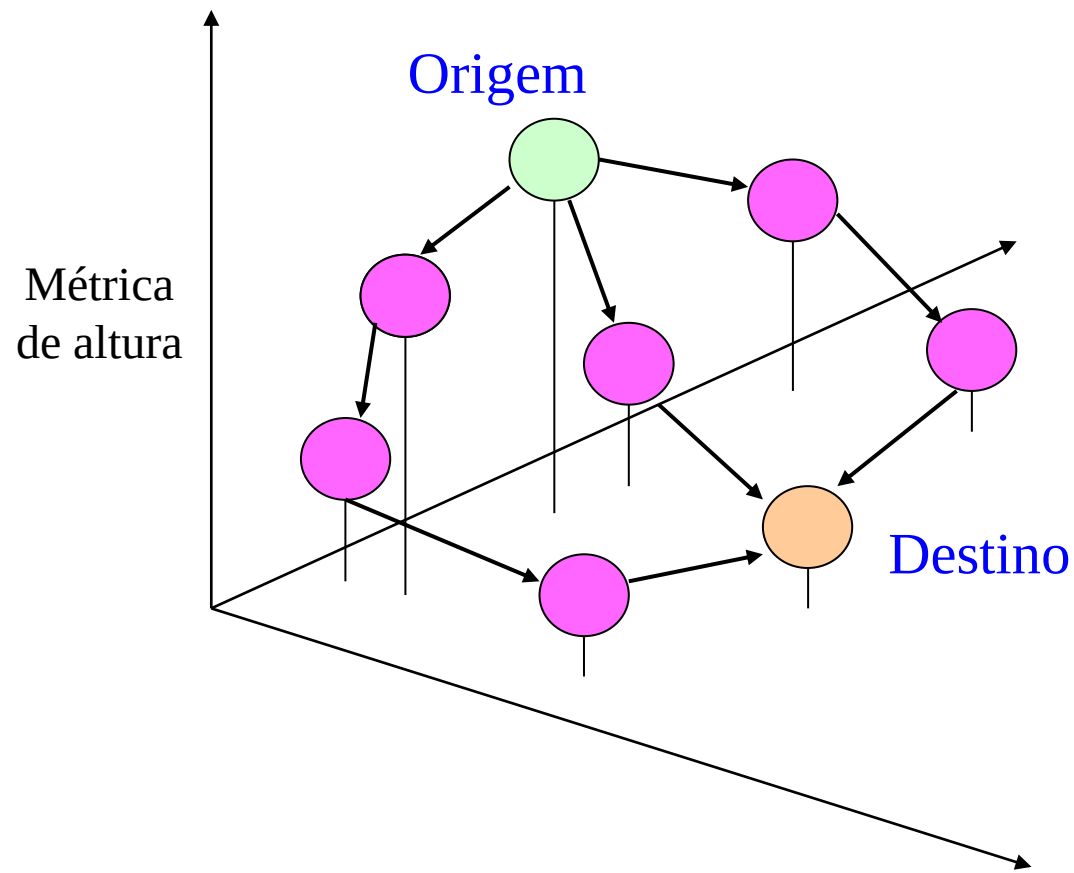
- *Ad-Hoc On Demand Distance Vector*
- AODV usa uma combinação de DSR e DSDV.
- Descoberta e manutenção de rotas do DSR
- Roteamento salto-a-salto, anúncios periódicos e números de sequência do DSDV.
- Números de sequência significam o “freshness” da rota – maior o número, mais atualizada a rota.
- Tabelas de roteamento ao invés de armazenar “*full routes*”.

TORA (*Temporally Order Routing Algorithm*)

- Roteamento iniciado na fonte (por demanda)
- Provê rotas “loop-free” e múltiplas rotas
- Minimiza reação a mudanças de topologia; faz a reação em nós próximos a mudança
- Rápida recuperação em falhas de rotas
- Detecta partição de rede e apaga rotas inválidas
- Três funções básicas:
  - *Criação de rota*
  - *Manutenção de rotaance*
  - *Eliminação de rota*

TORA – Criação de rota

Usa métrica de “altura” para construir um grafo direcionado acíclico.



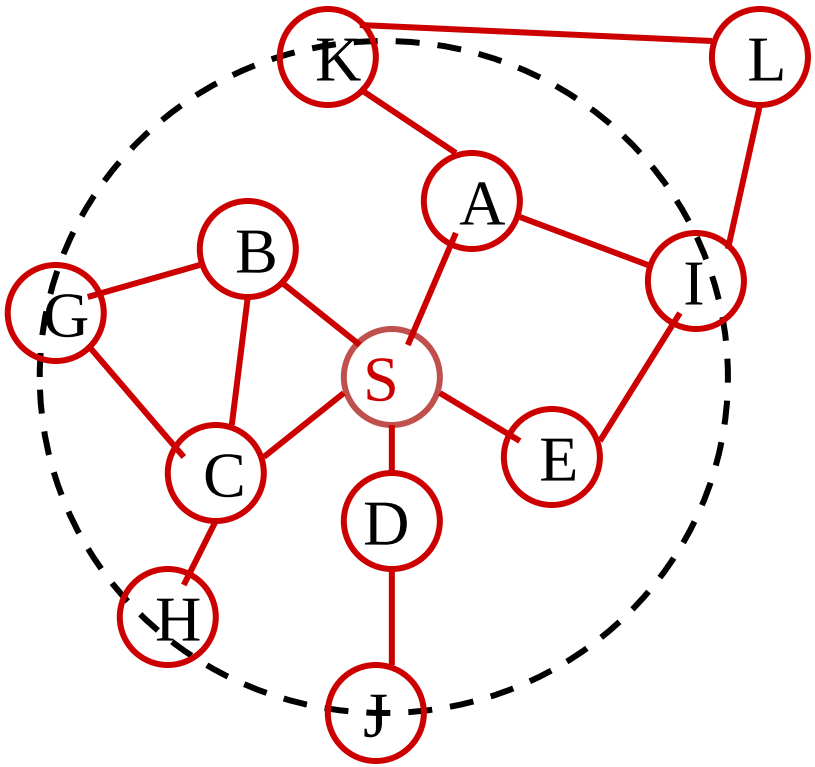
Algoritmos iniciados na fonte

- Resumo:
- Roteamento iniciado na fonte usa meio sem fio de forma mais eficiente (só quando há demanda)
- A descoberta de rotas é baseado em difusão na rede, o que têm alta latência e consome muita banda
  - Há problemas quando:
  - a topologia é altamente dinâmica
  - os enlaces não são bi-direcionais

- Funciona bem somente se demanda é pouco frequente por comunicação e para redes pequenas

Zone Routing Protocol (ZRP)

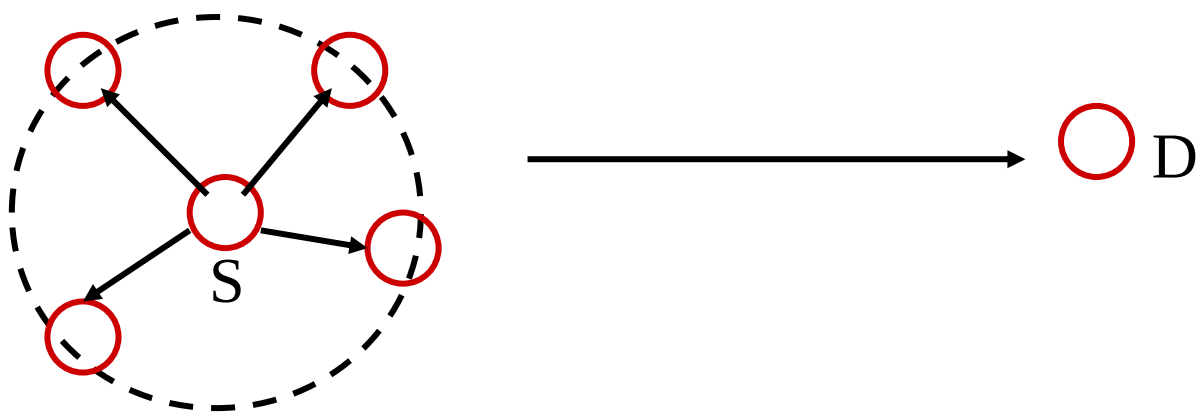
- Protocolo híbrido
- A parte proativa do protocolo é restrita a uma pequena vizinhança.
- A parte reativa é usada para roteamento ao longo da rede.



Todos os nós exceto L estão na zona de roteamento de S com raio 2.

Estratégia básica do ZRP

- O roteamento é dividido em duas partes:
  - Roteamento intra-zona: Considere que um nó origem (S) quer enviar um pacote ao destino (D). Se D está dentro da zona de roteamento de S, o roteamento acaba na fase intra-zona.
  - Senão, S envia o pacote aos nós periféricos (*bordercasting*).
  - Roteamento inter-zona: descobre a rota para o destino reativamente. O pacote é enviado dos nós periféricos para o destino.



ZRP

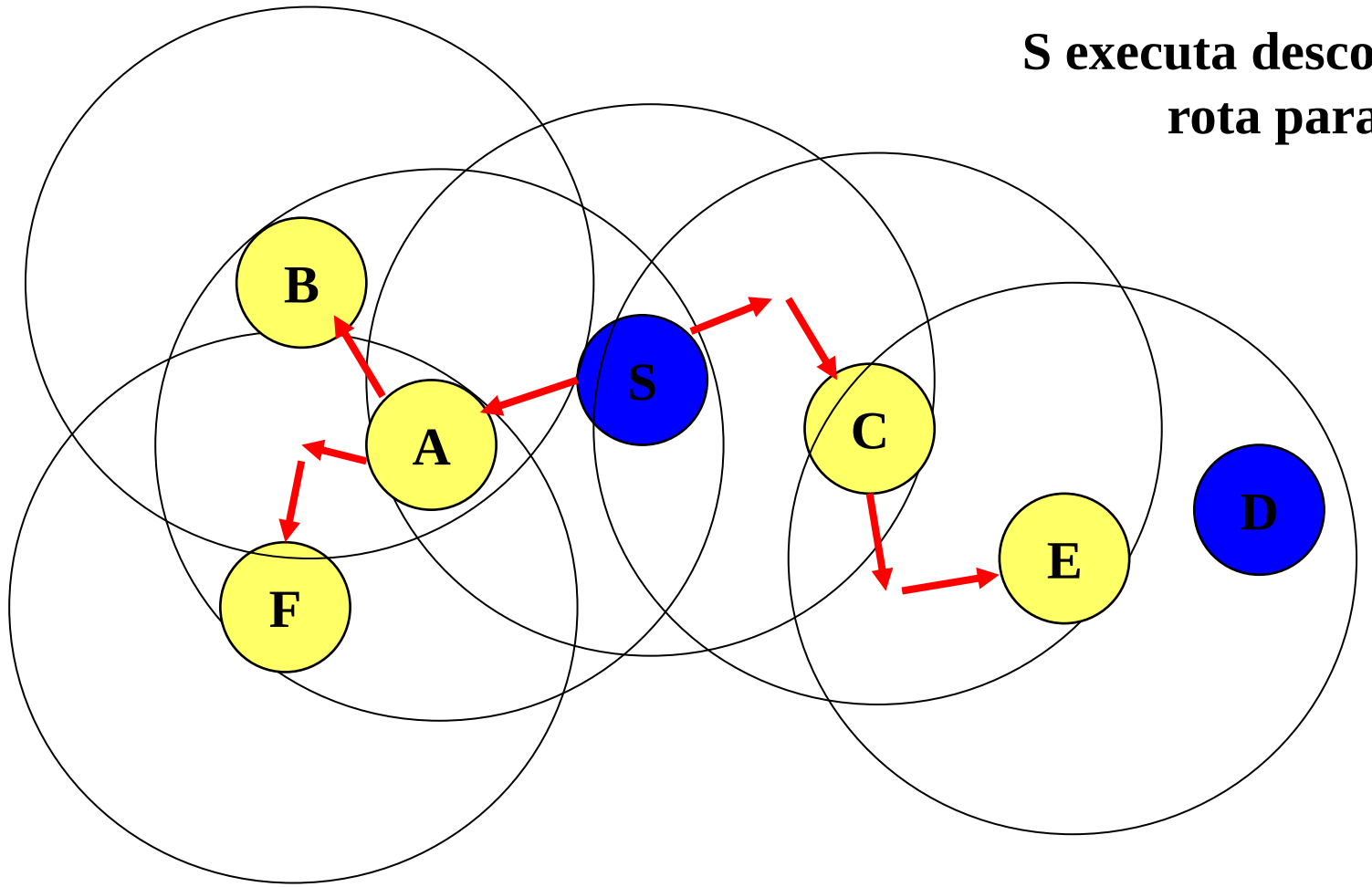
- Roteamento intra-zona: Cada nó coleta informação sobre os nós em sua zona proativamente. Similar ao protocolo DSDV.

- Cada nó mantém uma tabela de roteamento para sua zona.

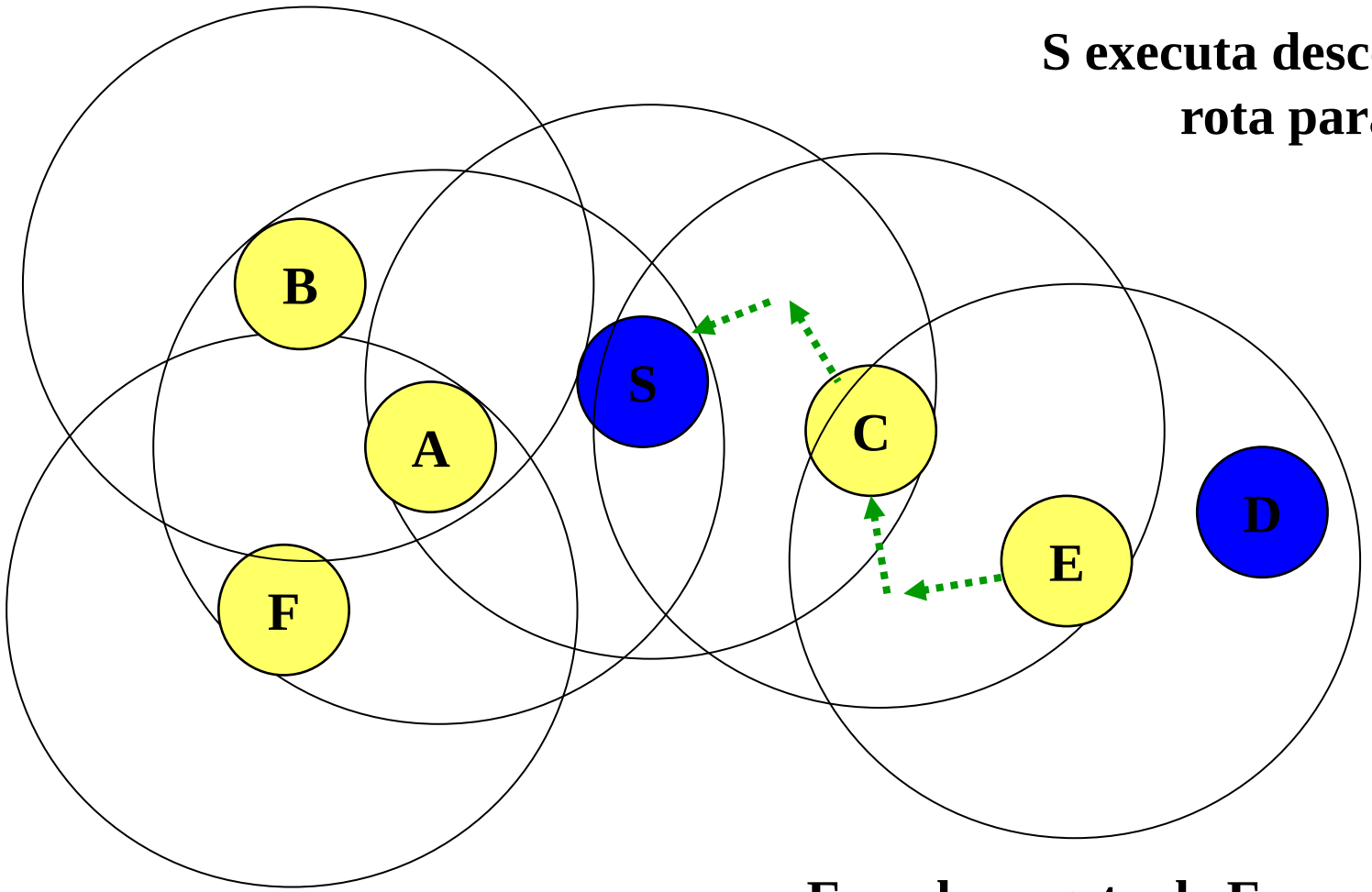
- Uma mensagem de notificação de zona morre após k saltos, ou seja, após atingir os vizinhos do nó em uma distância de k saltos.

ZRP: Exemplo com

Raio da Zona = *K = 2*



ZRP: Exemplo com K *= 2*

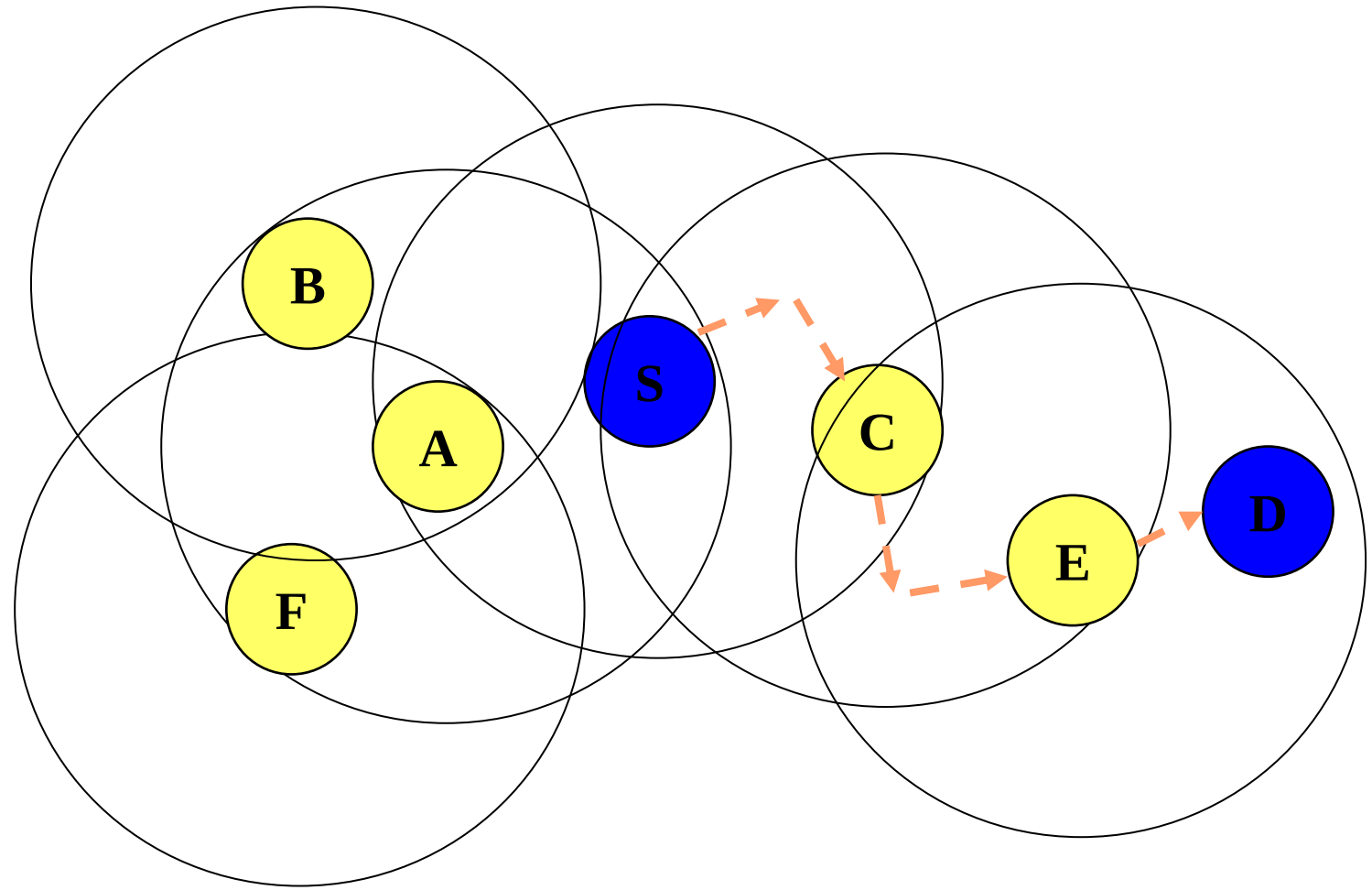


S executa descoberta de rota para D

Indica route reply

E conhece rota de E para D, então route request não precisa ser enviado de E para D

ZRP: Exemplo com *K = 2*



Indica rota dos dados

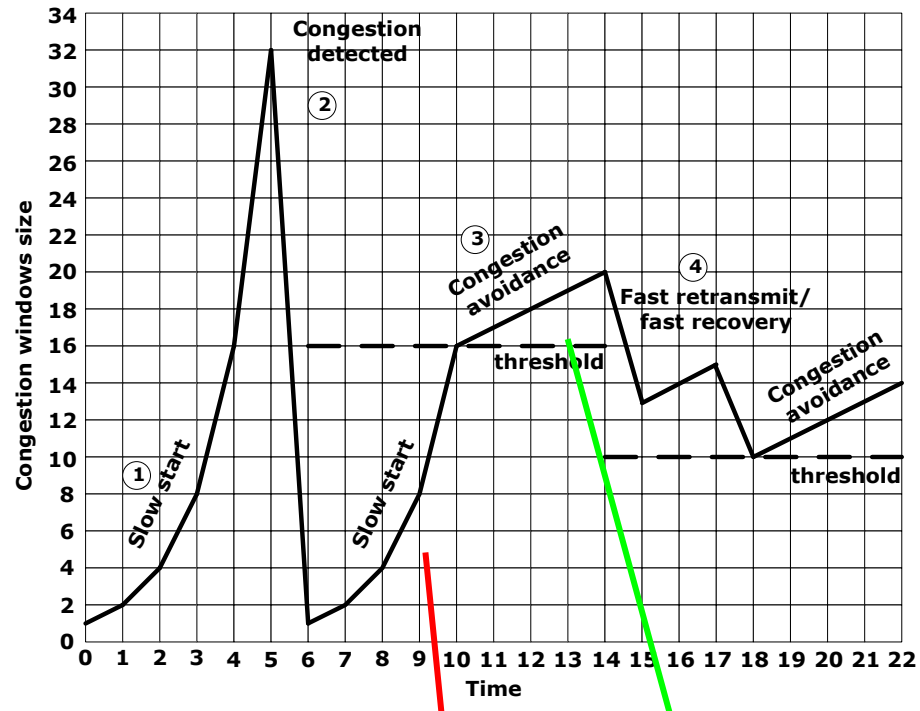
Métricas alternativas

- O número de saltos é somente uma das possíveis métricas para determinar a rota
- Pode-se considerar também a qualidade dos enlaces, energia residual nos nós, métricas de QoS, etc

Protocolos de Transporte para MANETs

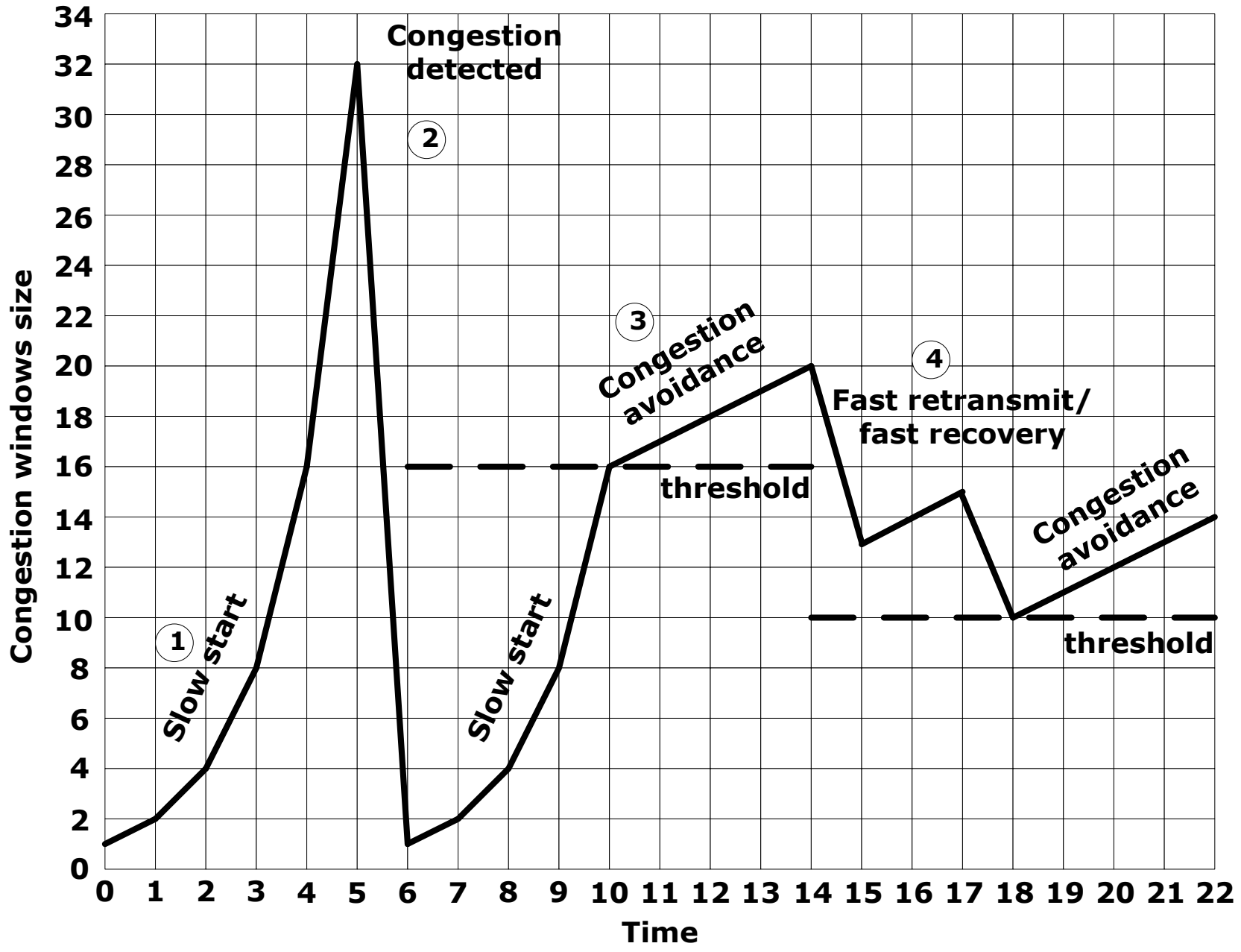
Conceitos TCP

- TCP Convencional: Tahoe, Reno, New-Reno
- Taxa de envio é controlada por
  - Janela de congestionamento (***cwnd***): limita o número de pacotes
  - Limiar de partida lenta (***ssthresh***): quando começa a prevenção de congestionamento
- Detecção de perda
  - 3 ACKs duplicados (mais rápido)
  - Timer de retransmissão expira
- Mecanismos de controle de congestionamento
  - Partida lenta: ***cwnd*** começa em 1 e aumenta exponencialmente
  - Prevenção de congestionamento: aumento linear
  - Retransmissão rápida e recuperação rápida: disparada por 3 ACKs duplicados



Partida lenta

Prevenção de congestionamento


Congestion windows size
Time
Congestion detected
1
2
3
4
Slow start
Slow start
Congestion avoidance
Congestion avoidance
Fast retransmit/ fast recovery
threshold
threshold
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34

Camada de Transporte

- Qual a melhor variante do TCP?
- É necessário um novo protocolo de transporte?
- Por que TCP tem desempenho ruim em MANETs?
  - Projetado para redes com fio
  - Considera todas as perdas como congestionamento.
- Existem outras alternativas?
  - A solução é usar protocolos não-TCP?

Diferenças das MANETs

1. **Mobilidade**
   - Estabilidade de rota e disponibilidade
2. **Alta taxa de erro de bits**
3. **Imprevisibilidade/Variabilidade**
   - Dificuldade de estimar timeout, RTT
4. **Contenção: pacotes competem pelo meio físico**
5. **Conexões longas tem desempenho ruim**

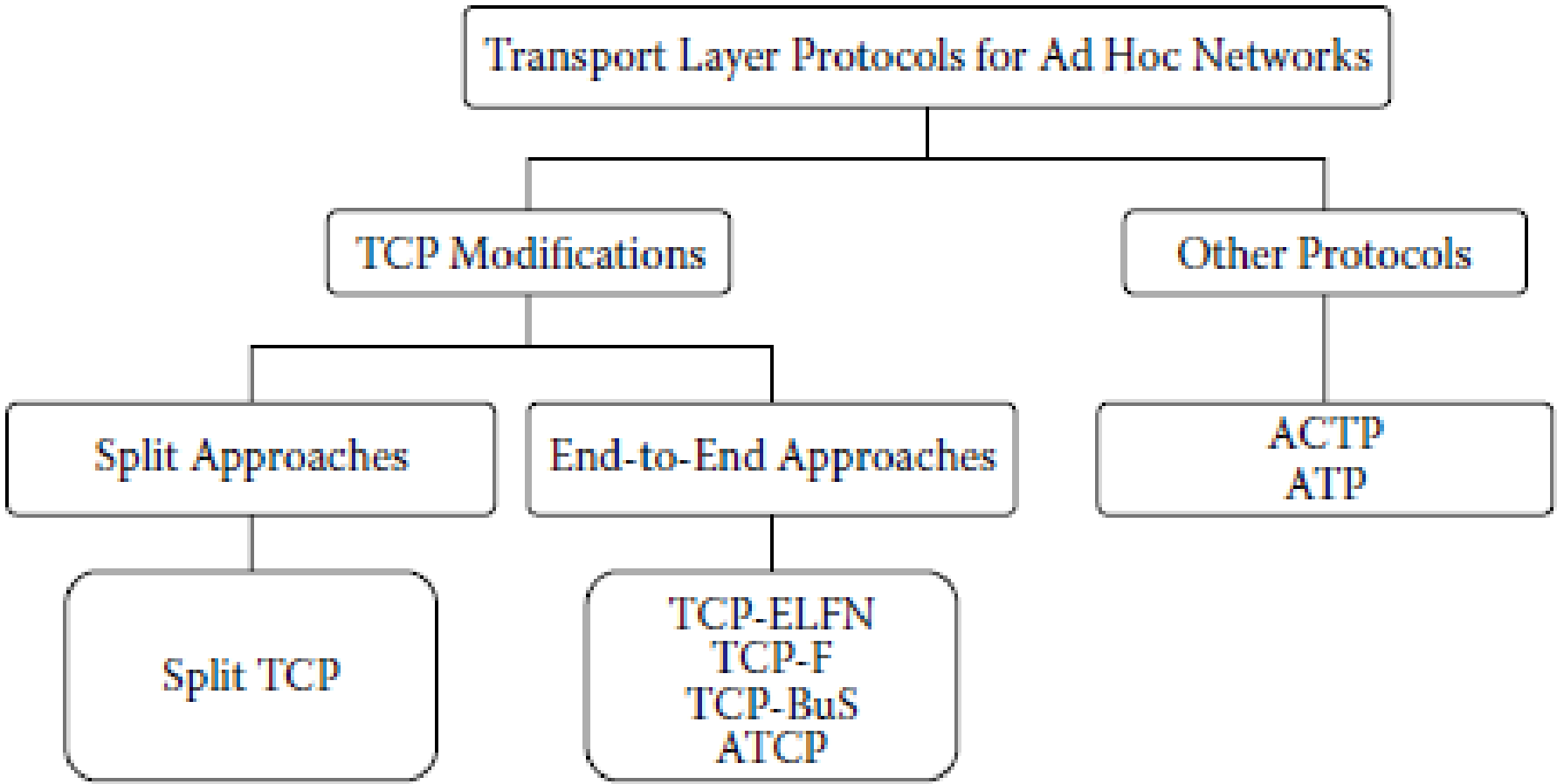
   - Muitos saltos vazão cai drasticamente

Por que TCP falha em MANETs?

Problemas específicos:

1. TCP interpreta falhas de rota como congestionamento
2. TCP interpreta erros do canal sem fio como congestionamento
3. Contenção reduz vazão e fairness
4. Aumento de atraso causa retransmissões desnecessárias
5. Ineficiência devido a perda de pacotes retransmitidos

Classificação de Protocolos de Transporte



Visão global

- **Melhores variantes do TCP:**
  - **TCP-Westwood** e **TCP-Jersey** parecem os melhores
  - Estimam melhor a largura de banda
- **Mecanismos TCP:**
  - Feedback de nós intermediários tem bons resultados.
- **Melhores abordagens não TCP:**
  - **Ad-hoc Transport Protocol (ATP)**
    - Estima taxa periodicamente
  - **Split-TCP:** promising new way of looking at transport layer
    - Proxies ao longo da conexão – várias conexões TCP menores
  - **Ponto-chave**: separação de controle de congestionamento da confiabilidade