Algoritmos Distribuídos para Roteamento em Redes Ad Hoc

Tiago Rodrigues Chaves

Orientador: Ricardo Augusto Rabelo de Oliveira

Programa de Pós-Graduação em Ciência da Computação
PPGCC/UFOP

26 de julho de 2011

Universidade Federal de Ouro Preto

Universidade Federal de Ouro Preto

Redes Ad Hoc/MANET

- Uma rede *ad hoc*, também é conhecida como *Mobile Ad hoc NETwork* (MANET).

Figura: Rede *ad hoc*

Rede Convencional

- É necessário que exista um ponto de acesso centralizador.

Figura: Rede convencional

Redes Ad Hoc/MANET

- Redes *ad hoc* não são restritas apenas para uso de computadores.

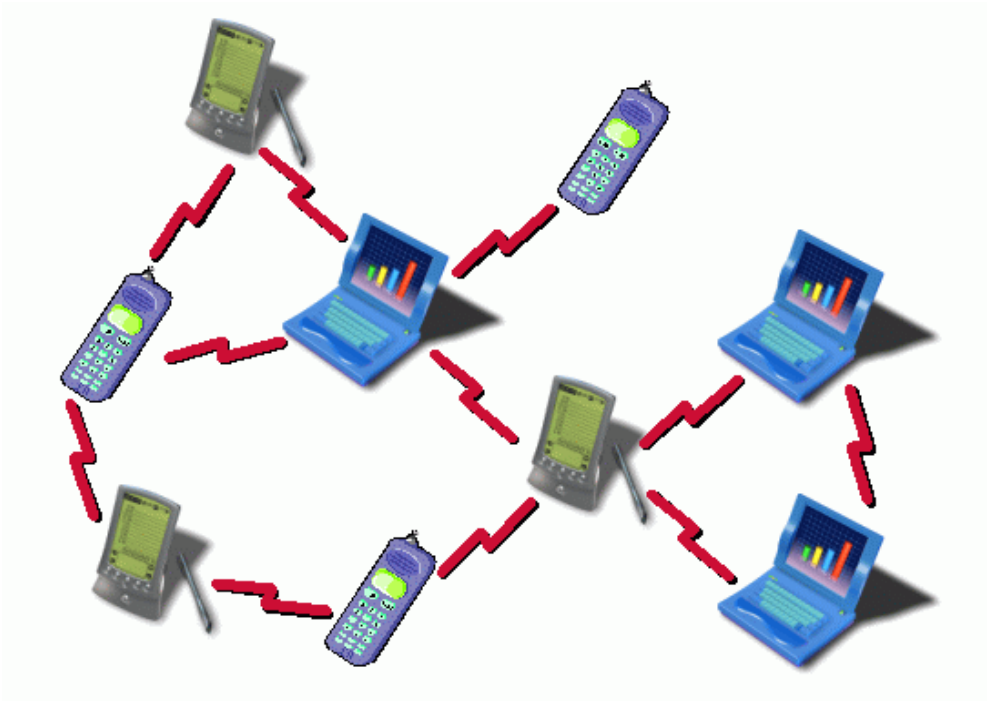


Figura: Rede *ad hoc* com diferentes dispositivos.

Protocolos de Roteamento

Protocolos Pró-ativos X Protocolos Reativos

1. **Pró-ativos:** todo nó da rede possui informações para todos os possíveis destinos.
2. **Reativos:** uma rota só é determinada quando um nó deseja enviar um pacote.

Algoritmos Distribuídos

Algoritmo + Distribuído (processamento e comunicação)

- Cada processo tipicamente executa o mesmo algoritmo;
- Interligados por um canal de comunicação;
- Processos comunicam entre si apenas usando mensagens.

Modelagem do Sistema de Memória Distribuída

- Sistema de memória distribuída pode ser modelado por um grafo não dirigido $Gp = (Np, Ep)$, onde:

- Np: conjunto de processadores.
- Ep: conjunto de enlaces de comunicação *full-duplex*.

- O roteamento da mensagem (q, Msg) é feito por um processador r usando a função $next_r(q)$.

Algoritmo Auto-estabilizável

- A propriedade de auto-estabilização modela a habilidade de um sistema se recuperar automaticamente de falhas transientes.

Universidade Federal de Ouro Preto

Objetivos

- Estudar e apresentar os algoritmos distribuídos de roteamento *Flooding* e AODV;
- Comparar os algoritmos em relação ao número de troca de mensagens;
- Simular os algoritmos em diferentes topologias de rede.

Metodologia

- Foram implementados em linguagem Java;
- O Algoritmo de *Flooding* foi simulado no *Distributed Algorithms in Java*(DAJ);
- As simulações consideraram diferentes topologias de rede.

Distributed Algorithms in Java(DAJ)

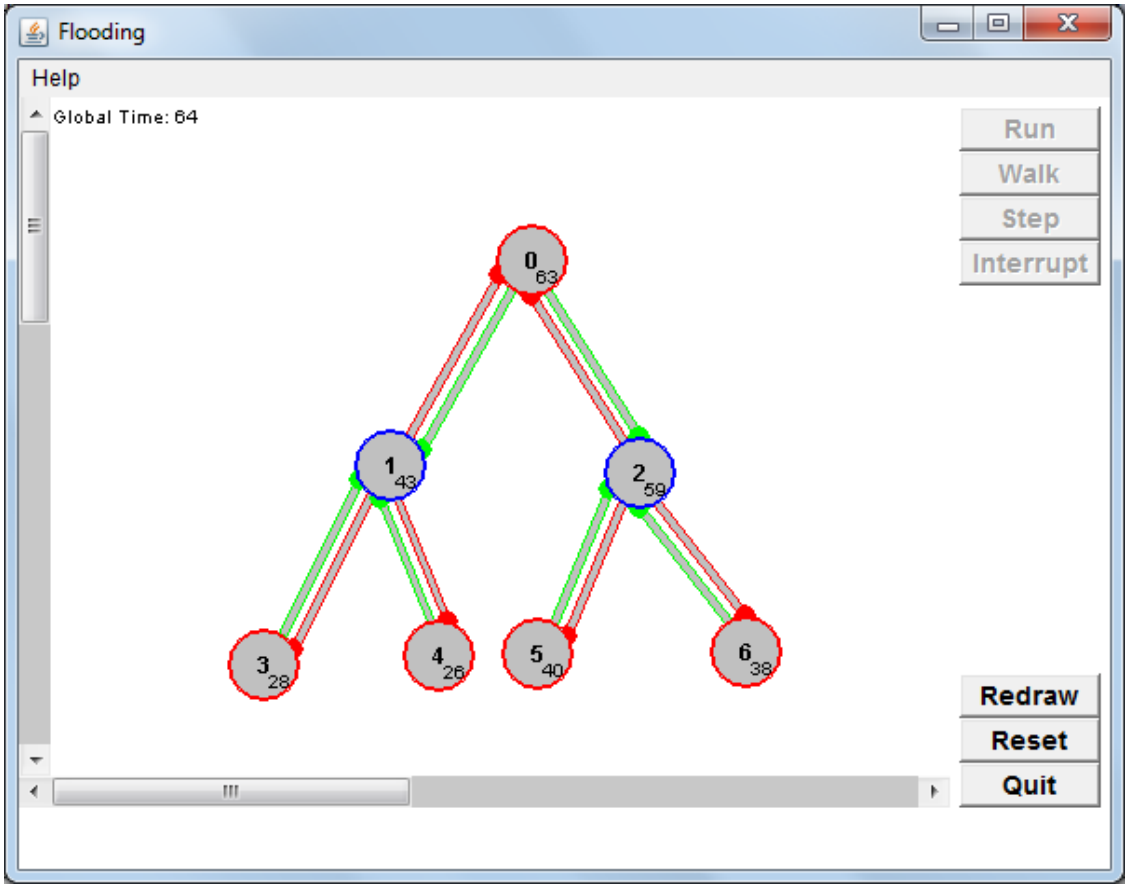


Figura: DAJ

Topologias de Rede

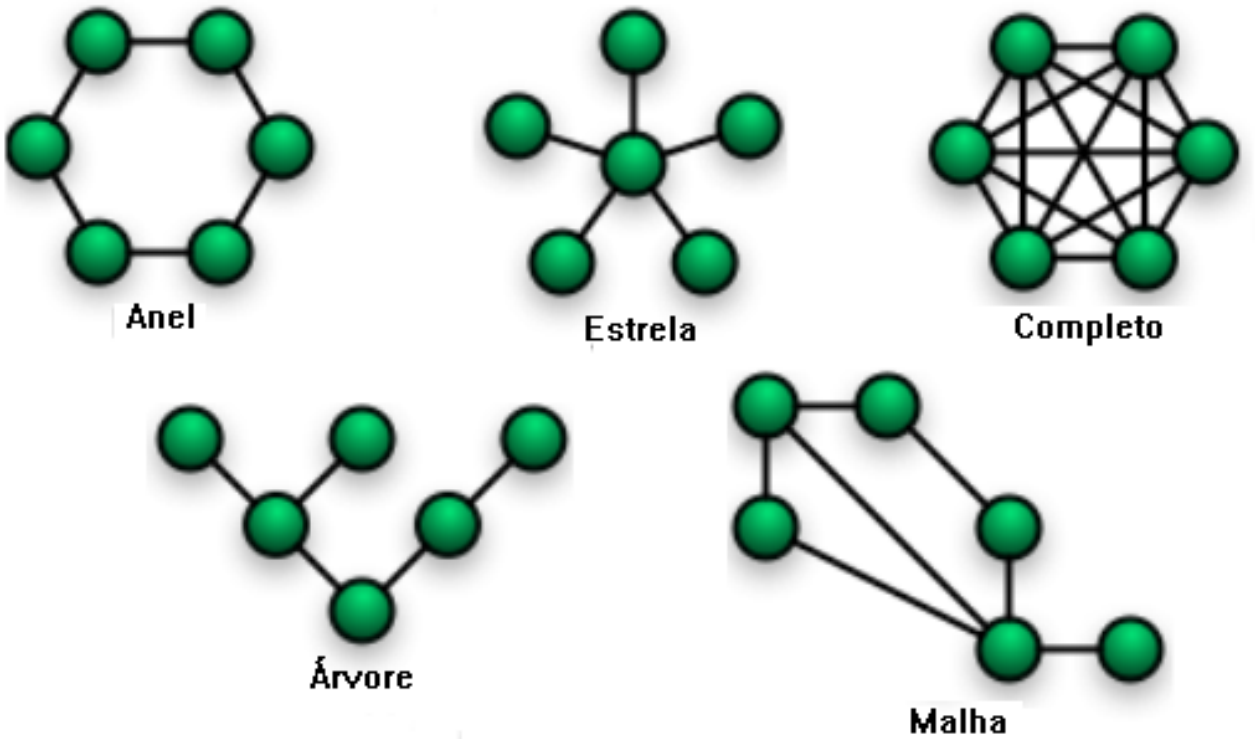


Figura: Topologias de Rede

Algoritmo de inundação (Flooding)

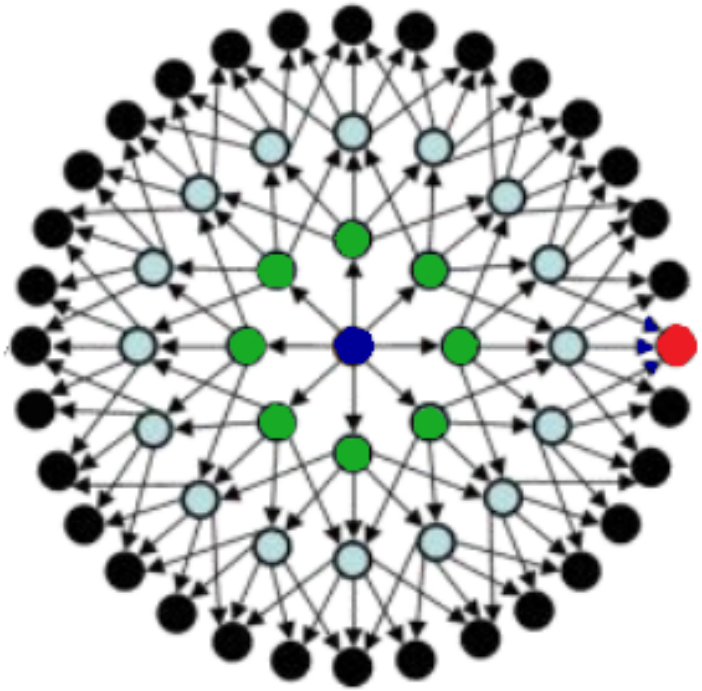


Figura: Estado inicial: nó azul deseja enviar uma mensagem para o nó vermelho.

Ad hoc On Demand Distance Vector (AODV)

Este algoritmo utiliza as seguintes variáveis:

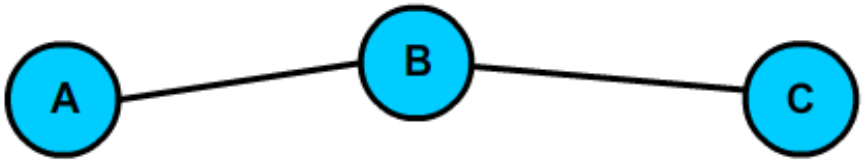
- Identificador fonte (*SrcID*);
- Identificador do destino (*DestID*);
- Número de sequência da fonte (*SrcSeqNum*);
- Número de sequência do destino (*DestSeqNum*);
- Identificador de *broadcast* (*BcastID*);
- Tempo de vida das mensagens (*TTL*);
- Número de *hops*.

Ad hoc On Demand Distance Vector (AODV)

O algoritmo possui as seguintes mensagens:

- *Route Request* (*RREQ*);
- *Route Reply* (*RREP*);
- Mensagens *HELLO*;
- *Route Error* (*RERR*).

Ad hoc On Demand Distance Vector (AODV)



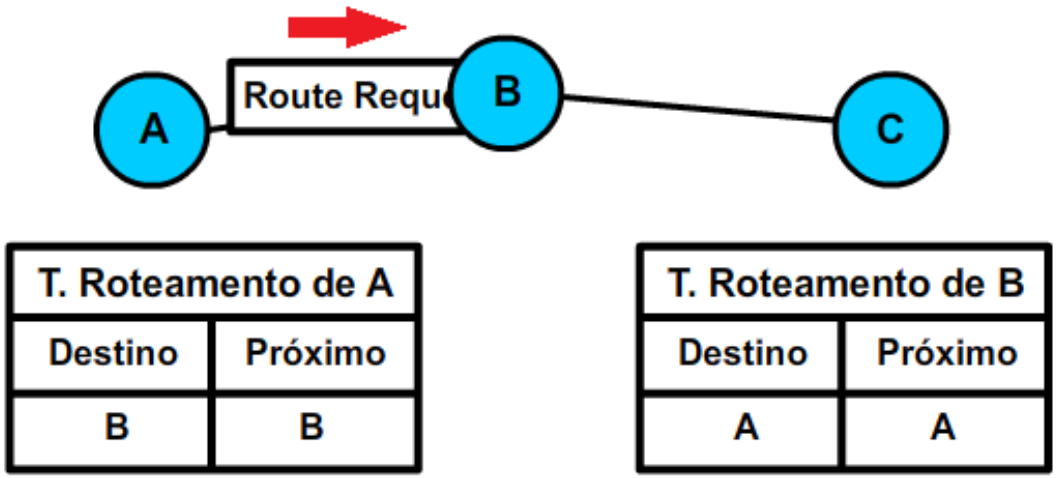
T. Roteamento de A	
Destino	Próximo
B	B

T. Roteamento de B	
Destino	Próximo
A	A

O nó "A" deseja enviar um pacote ao nó "C"

Figura: AODV - Descobrimento de Rotas

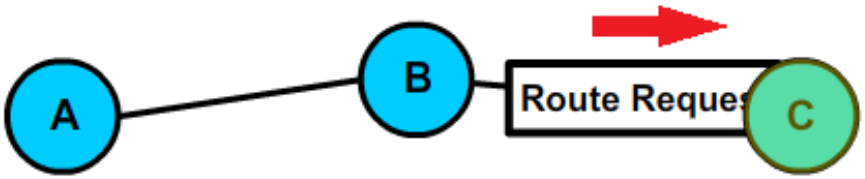
Ad hoc On Demand Distance Vector (AODV)



O nó "A" envia um pacote ROUTE REQUEST aos seus vizinhos

Figura: AODV - Descobrimento de Rotas

Ad hoc On Demand Distance Vector (AODV)



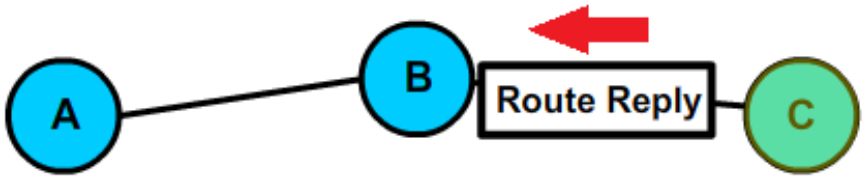
T. Roteamento de A	
Destino	Próximo
B	B

T. Roteamento de B	
Destino	Próximo
A	A

O processo continua, até que o nó destino, ou um nó intermediário que conheça a rota desejada, é encontrado

Figura: AODV - Descobrimento de Rotas

Ad hoc On Demand Distance Vector (AODV)



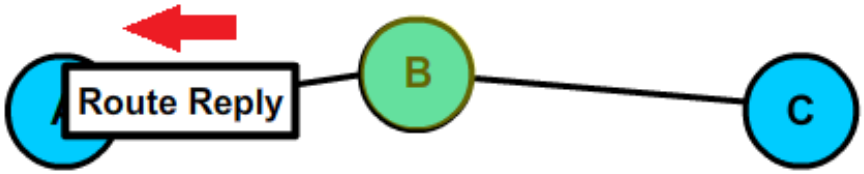
T. Roteamento de A	
Destino	Próximo
B	B

T. Roteamento de B	
Destino	Próximo
A	A
C	C

O nó destino (ou intermediário conhecedor da rota) envia um pacote ROUTE REPLY de volta ao nó fonte (A). Nesse processo, as tabelas de roteamento dos nós da rota encontrada são atualizados

Figura: AODV - Descobrimento de Rotas

Ad hoc On Demand Distance Vector (AODV)



T. Roteamento de A	
Destino	Próximo
B	B
C	B

T. Roteamento de B	
Destino	Próximo
A	A
C	C

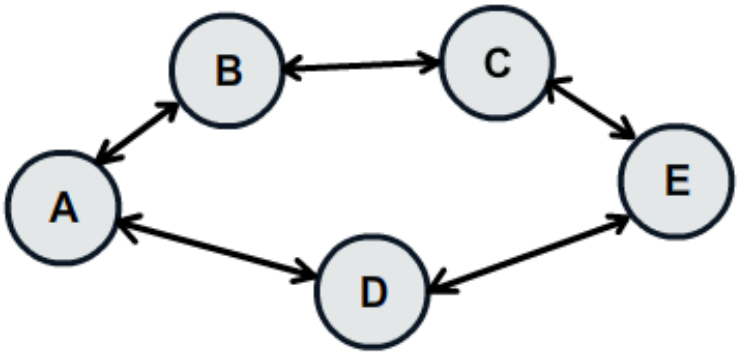
Com a rota descoberta, o nó "A" pode enviar seu pacote ao nó "C"

Figura: AODV - Descobrimento de Rotas

Ad hoc On Demand Distance Vector (AODV)

Cache Nó A	
Destino	Rota
⋮	⋮
E	A - B - C - E
E	A - D - E

Cache Nó B	
Destino	Rota
⋮	⋮
⋮	⋮
E	C - E



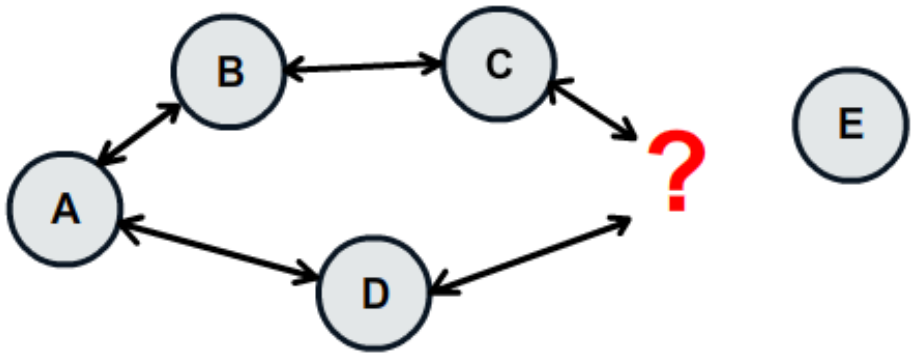
A rede opera normalmente até que ...

Figura: AODV - Manutenção de Rotas

Ad hoc On Demand Distance Vector (AODV)

Cache Nó A	
Destino	Rota
⋮	⋮
E	A - B - C - E
E	A - D - E

Cache Nó B	
Destino	Rota
⋮	⋮
⋮	⋮
E	C - E



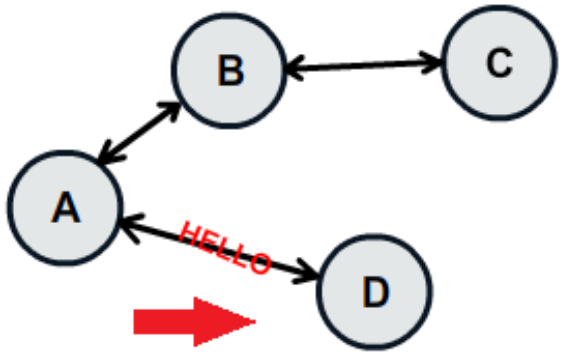
... um nó se mova e o enlace caia.

Figura: AODV - Manutenção de Rotas

Ad hoc On Demand Distance Vector (AODV)

Cache Nó A	
Destino	Rota
⋮	⋮
E	A - B - C - E
E	A - D - E

Cache Nó B	
Destino	Rota
⋮	⋮
⋮	⋮
E	C - E



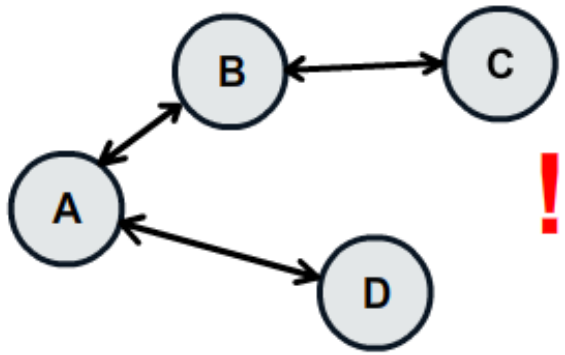
É disparado um hello de rotina ...

Figura: AODV - Manutenção de Rotas

Ad hoc On Demand Distance Vector (AODV)

Cache Nó A	
Destino	Rota
⋮	⋮
E	A - B - C - E
E	A - D - E

Cache Nó B	
Destino	Rota
⋮	⋮
⋮	⋮
E	C - E



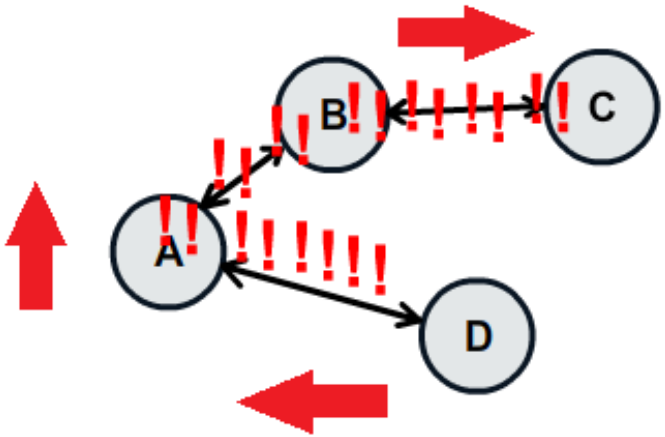
... que não é recebido de volta, fazendo com que o nó "D" detecte queda de enlace.

Figura: AODV - Manutenção de Rotas

Ad hoc On Demand Distance Vector (AODV)

Cache Nó A	
Destino	Rota
⋮	⋮
E	A - B - C - E
E	A - D - E

Cache Nó B	
Destino	Rota
⋮	⋮
⋮	⋮
E	C - E



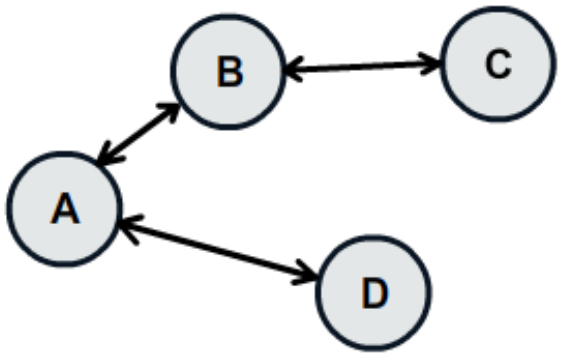
Um sinal de erro é disparado para os outros nós, em unicast ou, se for preciso, em broadcast para que ...

Figura: AODV - Manutenção de Rotas

Ad hoc On Demand Distance Vector (AODV)

Cache Nó A	
Destino	Rota
⋮	⋮
E	A - B - C - E
E	A - D - E

Cache Nó B	
Destino	Rota
⋮	⋮
⋮	⋮
E	C - E



... os nós possam eliminar "E" e as sub-rotas posteriores a esse nó de sua tabela de roteamento.

Figura: AODV - Manutenção de Rotas

Universidade Federal de Ouro Preto

Experimentos com o Algoritmo Flooding

Topologia	RREQ	RREP	RREP-ENC
Ring	7	14	0
Mesh	7	18	0
Star	7	12	0
Line	7	12	0
Tree	7	10	72
Fully Conected	7	42	0

Tabela: Experimentos com o Algoritmo de *Flooding*.

Análise de Complexidade

O fator a ser analisado é a ordem de complexidade das trocas de mensagem entre os nós de uma rede.

- Considerando a topologia de um grafo completo, onde todos os nós estão interligados a todos os outros nós.

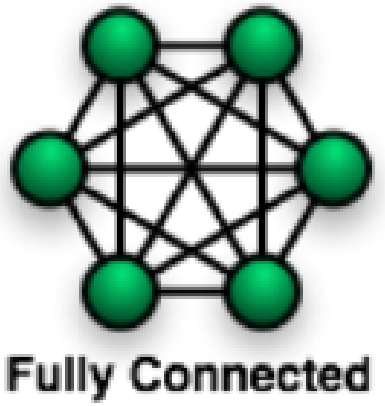


Figura: Grafo Completo G

Análise de Complexidade

Suponha:

- n é o número de vértices em um grafo completo G.
- $d(n)$ é o grau do vértice, ou seja, número de arestas que chegam e saem do vértice.

Cada vértice do grafo contribuirá com:

$$\Theta(d(n)) = \Theta(n-1) \tag{1}$$

Análise de Complexidade

Assim, a ordem de complexidade das trocas de mensagem é representada por
$(numero - de - vertices) \times (grau - do - vertice)$.

$$\Theta(n \times d(n)) = \Theta(n \times (n-1)) = \Theta(n^2 - n) = \Theta(n^2) \qquad (2)$$

Universidade Federal de Ouro Preto

Conclusão

- Ainda não existe nenhum algoritmo que tenha um desempenho razoável em todos os possíveis ambientes;
- O *Flooding* é uma abordagem simples e ingênua, onde a demanda por escalabilidade da rede torna o algoritmo proibitivo;
- O AODV com estabelecimento dinâmico das tabelas de roteamento evita a sobrecarga de troca de mensagens na rede.

Trabalhos Futuros

- Estudar o algoritmo distribuído de roteamento, o ATR (*Augmented Tree-based Routing*);
- Combinar os algoritmos AODV e ATR
- Propor um algoritmo distribuído de roteamento, que apresente resultados melhores em relação ao número de troca de mensagens em redes *ad hoc*.