

Sistemas Distribuídos

Roteamento em Redes Limitadas (LLN)

Noemi Rodriguez

2017

- uso de protocolos que supõem acoplamento entre nós roteadores
 - BGP: uso de conexões TCP/IP
 - OSPF: uso de *heartbeats*
 - mensagens enviadas a cada detecção de alteração em estado de links
 - gasto de processamento, memória e energia!

Lossy low-power networks

- conceito definido por grupo de trabalho IETF
- redes sem fio de sensores e atuadores com recursos limitados
 - LBR (roteadores de borda) conectam rede à Internet

características

- dispositivos não têm capacidade de manter muito estado
- ênfase em economia de energia
- padrões principais de comunicação:
 - MP2P
 - P2MP
 - P2P

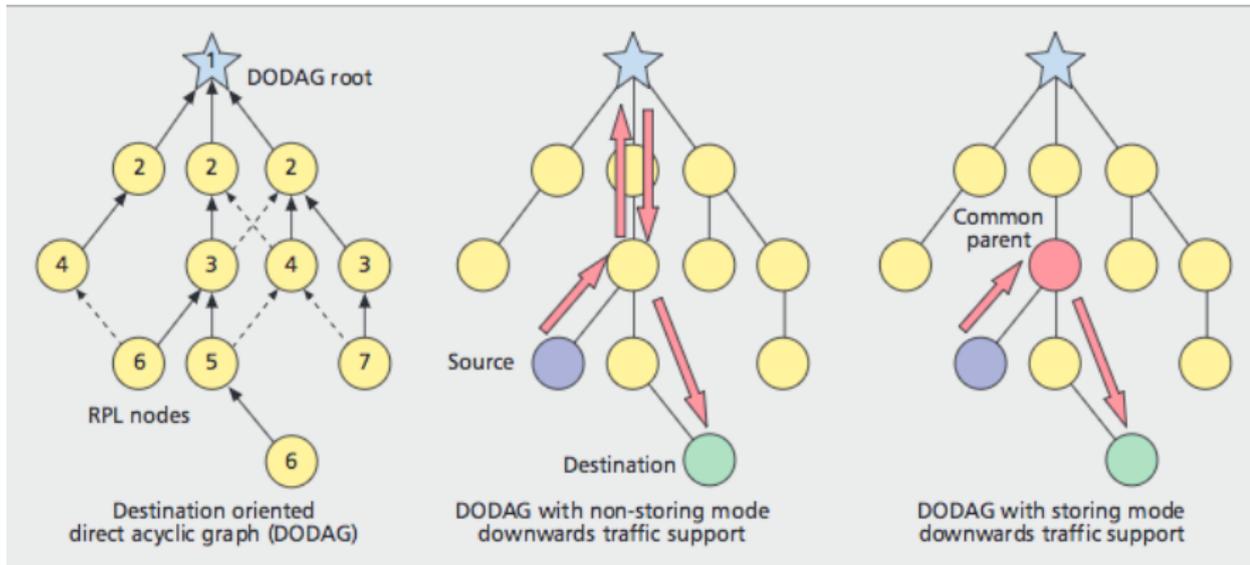
características

- salto a salto ou roteamento na origem
- roteamento pró-ativo ou reativo

RFC 6550

- sobre o 6LoWPAN
 - configuração da wsan: quem é roteador?
 - uso de algoritmo *trickle*
 - estrutura DODAG: *Destination-Oriented Directed Acyclic Graph*
-
- Olsson, Jonas. 6LoWPAN demystified. Technical Report. Texas Instruments. 2014.
(www.ti.com/lit/wp/swry013/swry013.pdf)

RPL — caminhos até uma *raiz* criam DAG



- 1 raiz inicia procedimento (como único nó no DODAG) com mensagem DIO (*DODAG Information Object*)
- 2 mensagem DIO indica *rank*: métricas variadas
- 3 número de sequência indica versão do DODAG
- 4 mensagem pode incluir configurações
 - mais simples: número de saltos
- 5 cada roteador recebe mensagens DIO e calcula conjunto de progenitores e progenitor *principal*
- 6 mensagem DAO (*Destination Advertisement Object*) enviadas de nós para a raiz anunciam rotas para eles

- roteador transmite DIO “regularmente”
 - ... mas pode detectar que seria redundante transmitir a partir de mensagens recebidas de outros roteadores
- ao receber uma mensagem DIO, roteador verifica se contém informação sobre mesma versão de DODAG que ele já tem ou sobre uma diferente.
 - versão posterior: receptor deve atualizar sua informação
 - versão anterior: atualizar se novo caminho melhor que o armazenado
- mensagens a serem enviadas são escalonadas para envio futuro em tempo t
 - se muitas mensagens redundantes forem recebidas, t é aumentado

- muito tendencioso para tráfego MP2P
- modo armazenamento rotas pode levar a carga de armazenamento pesada

"implementabilidade"

- implementação Contiki ocupa aproximadamente 50KBytes
 - restrições levam implementações a cortarem partes da especificação

- normalmente supõe nós estáticos e gateways possivelmente móveis
- redes podem ser homogêneas ou heterogêneas

fluxos de dados

- contínuos
 - disparados por eventos
 - sob demanda
 - roteamento como parte da aplicação!!!
-
- Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3), May 2005.

agregação

- supressão (filtragem)
- redução
 - mínimo, máximo, soma, média, ...

- sem endereçamento individual de nós
- as demandas por informação são baseadas em metadados ou atributos
 - SPIN
 - Directed Diffusion
 - Rumor Routing

inundação

- cada nó transmite o que recebe para todos os (demais) vizinhos

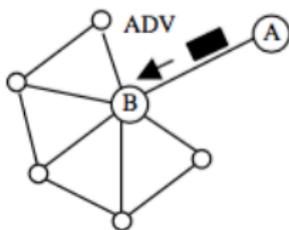
fofoca

- cada nó transmite o que recebe para um vizinho selecionado aleatoriamente

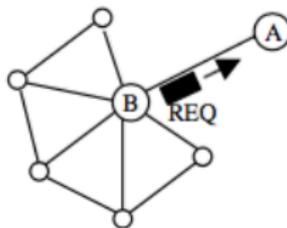
- simplicidade
- sobrecarga de energia e processamento

Protocolo SPIN

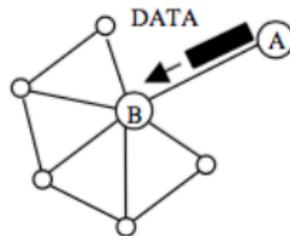
- anúncios, requisições e respostas
 - modelo, sem padronização de formatos de msgs



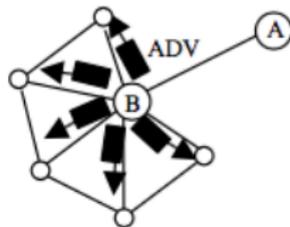
(a)



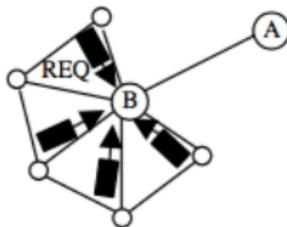
(b)



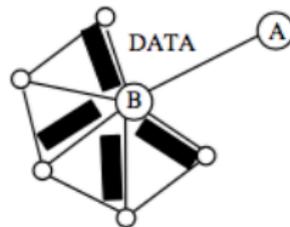
(c)



(d)

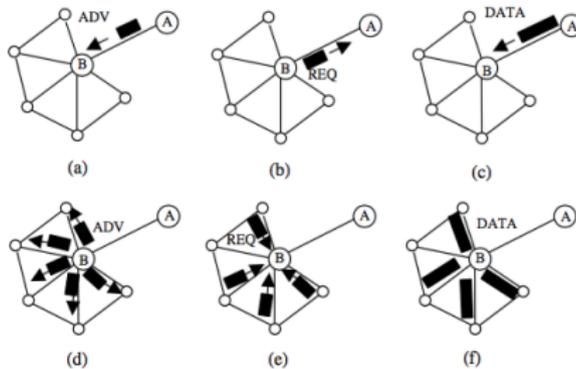


(e)



(f)

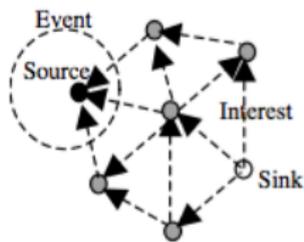
Protocolo SPIN



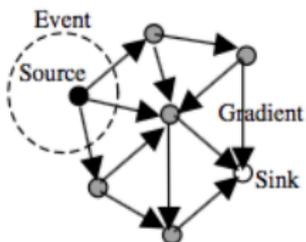
- economia em relação a inundação
- pouca garantia de chegada da informação

Difusão direcionada

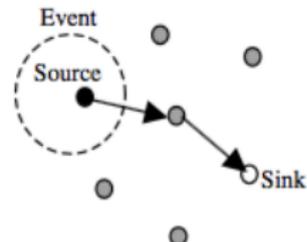
- anúncios de interesse e respostas com dados



(a)



(b)



(c)

- anúncios de interesse
 - difusão pela rede
 - possibilidade de restringir broadcast

```
type = four-legged animal
interval = 1s
rect = [-100, 200, 200, 400]
timestamp = 01:20:40
expiresat = 01:30:40
```

- anúncios de interesse
 - difusão pela rede
 - conceito de gradientes

```
type = four-legged animal
interval = 1s
duration = 20s
rect = [-100, 200, 200, 400]
timestamp = 01:20:40
expiresat = 01:30:40
gradient = nó vizinho interessado, taxa, ...
```

- nós na região de um interesse recebido podem ligar sensores
- ao receber dado de sensor ou por mensagem, busca por interesse que case
 - taxas de atendimento podem ser diminuídas no caminho
 - nós mantêm dados recebidos em cache

reforço de rotas

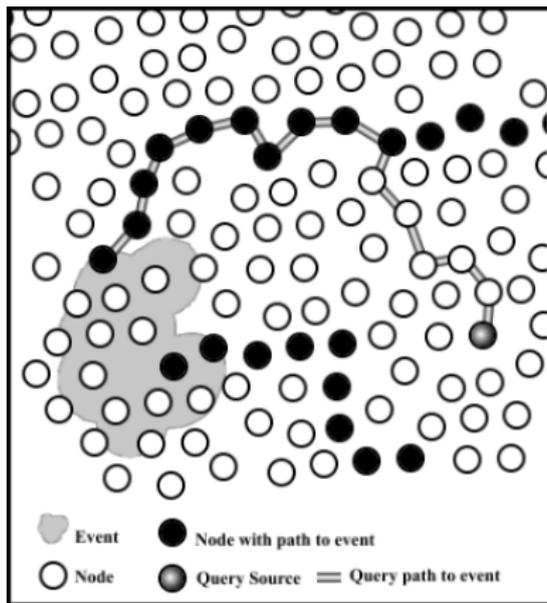
- *sink* pode escolher um vizinho para reenviar mensagem de interesse com maior taxa de atualização

- bom para aplicações baseadas em consultas
- não tão bom para aplicações de monitoramento contínuo...
- escolha de caminho preferencial pode detonar energia de determinados nós

Roteamento por rumores (rumor routing)

evitando inundação

- eventos geram *agentes* em caminhos aleatórios



Roteamento por rumores (2)

- agente carrega lista de evento (metadados) e número de saltos até ele
- ao chegar a novo nó, sincronização com a tabela local
- nó que recebe agente pode não ter caminho algum ou ter caminho pior até evento
- queries e agentes têm TTL para expiração

técnicas

- probabilidade para rotear entre um conjunto de nós vizinhos
- radios ligados apenas parte do tempo
 - funciona bem para aplicações baseadas em eventos

- noção de *clusters* ou grupos com um líder
 - líder pode ser escolhido dinamicamente
 - tratamento de falhas
 - preocupação com consumo de energia
- roteamento entre Internet e nós feito por líderes

uso de GPS em cada nó

- roteamento de demandas a regiões específicas
 - uso conjunto com difusão direcionada para rotear interesses a regiões específicas (GEAR)
- cálculo de distâncias entre nós para construção de rotas

- Leitura do survey de Akkaya e Younnis
- Escrever resumo sobre técnicas de roteamento centradas em dados (entrega dia 10/5)
- Estudo de roteamento por rumores:
 - D. Braginsky, D. Estrin, Rumor routing algorithm for sensor networks, in: Proceedings of the First Workshop on Sensor Networks and Applications (WSNA), Atlanta, GA, October 2002.