

Redes de Computadores 2
EEL 879
Parte III
Roteamento Unicast na Internet
Estado do Enlace

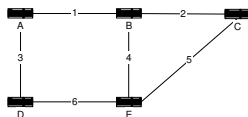
Luís Henrique M. K. Costa
luish@gta.ufrj.br

Universidade Federal do Rio de Janeiro - PEE/COPPE
P.O. Box 68504 - CEP 21945-970 - Rio de Janeiro - RJ
Brasil - <http://www.gta.ufrj.br>

Protocolos de Estado do Enlace

- Baseiam-se em um *mapa distribuído* da topologia da rede
- O mapa deve ser atualizado a cada mudança na topologia
- Cada nó é capaz de calcular a melhor rota entre quaisquer 2 pontos da rede, a partir do mapa local

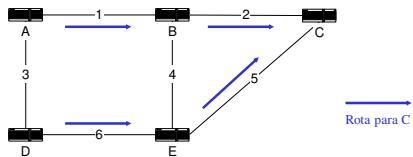
Base de Dados de Estados do Enlace



De	Para	Enlace	Dist.
A	B	1	1
A	D	3	1
B	A	1	1
B	C	2	1
B	E	4	1
C	B	2	1
C	E	5	1
D	A	3	1
D	E	6	1
E	B	4	1
E	C	5	1
E	D	6	1

Estados do Enlace

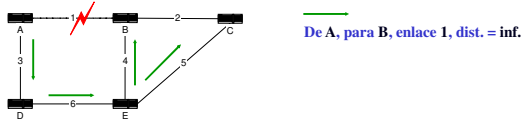
- Cada nó pode calcular o caminho mais curto para todos os outros nós
- Todos os nós possuem a mesma base de dados
 - Não podem ocorrer loops



GTA/UFRJ

Protocolo de Inundação

- Detecção de falhas deve ser rápida e confiável
 - Protocolo de inundação



GTA/UFRJ

Protocolo de Inundação

- Mensagens "antigas" não devem contaminar as bases de dados
 - Mensagens devem ser identificadas (datadas)
 - Estampa de tempo
 - Número de mensagem
- Algoritmo
 1. Recepção da mensagem. Busca do registro na base de dados.
 2. Se o registro não existia, adicioná-lo à base. Enviar a mensagem em *broadcast*.
 3. Senão, se o número na base de dados é menor que o número recebido na mensagem, substituir o registro pelo novo valor. Enviar a mensagem em *broadcast*.
 4. Senão, se o número na base de dados é maior, transmitir o registro da base de dados em uma nova mensagem, na interface de entrada.
 5. Senão, se os números são iguais, não fazer nada.

GTA/UFRJ

Proteção dos Mapas no OSPF

- A inundação inclui reconhecimentos salto-a-salto
- Os pacotes de descrição são transmitidos de maneira segura
- Cada registro de estado do enlace é associado a um temporizador e é retirado se não for devidamente atualizado
- Registros são protegidos por um checksum
- As mensagens podem ser autenticadas
 - Ex. senhas

GTA/UFRJ

Problema após falha de um nó

- Nó X desliga e religa após um curto intervalo
- X envia estado do enlace com no. de seq. = 1
- X recebe estados de enlace de nós vizinhos com no. de seq. maior
 - Mas na verdade, menos atuais
- Solução
 - X deve re-enviar os "seus" registros, com no. de seq. igual ao no. de seq. recebido + 1

GTA/UFRJ

Algoritmo Shortest Path First

- *Shortest Path First* (SPF) – Dijkstra
 - Cálculo do caminho mais curto entre um nó e todos os outros nós da rede
- Funcionamento
 - Rede
 - conjunto V contendo N nós, conjunto E contendo M enlaces
 - Separa os nós em 2 grupos
 - C – nós para os quais o caminho mais curto é conhecido
 - R – nós restantes
 - O – lista ordenada de caminhos
 - Nó fonte = S

GTA/UFRJ

Algoritmo Shortest Path First

1. $C = \{S\}$; $R = V - \{S\}$; $O = \{\text{caminhos de 1 salto a partir de } S\}$
Os caminhos em O possuem custo igual à métrica do seu enlace.
Ordenar os caminhos em O por ordem crescente de custos.
2. Se $R = \{\}$, $O = \{\}$, ou se o primeiro caminho de O possui custo inf.,
marcar todos os nós em R como inalcançáveis. O algoritmo terminou.
3. Primeiro, examine P , o caminho mais curto em O . Remova P de O .
Seja U o último nó em P . Se U já pertence a C , vá para o passo 2.
Senão, P é o caminho mais curto de S para U . Mova U de R para C .
4. Construa um novo conjunto de caminhos candidatos através da
concatenação de P e dos enlaces saindo de U .
Insira os novos caminhos na lista O , mantendo sua ordenação.
Vá para o passo 2.

GTA/UFRJ

Algoritmo Shortest Path First

- o Complexidade
 > $O(M \cdot \log M)$
- o Número de passos para convergir menor que
 Bellman-Ford ($O(N \cdot M)$)

GTA/UFRJ

Vantagens do Estado do Enlace

- o Convergência rápida e livre de *loops*
- o Suporte de métricas precisas, ou múltiplas métricas
- o Suporte de múltiplos caminhos
- o Representação separada de rotas externas

GTA/UFRJ

Convergência Rápida

- DV – Bellman-Ford
 - > No. de passos proporcional ao número de nós
 - Igual ao número de saltos do mais longo caminho, no pior caso
- LS
 - > Transmissão da atualização através de inundação
 - > Cálculo de rotas realizado localmente (Dijkstra)
- LS mais rápido
 - > Atualizações disparadas (DV) ~ inundação (LS)
 - No melhor caso, em que 1 DV de atualização é suficiente
 - Intervalo entre atualizações disparadas
 - 1 a 5s no RIP
 - > Cálculo de rotas no LS
 - 200ms para rede com ~ 200 nós
 - > Falha de enlace > vários destinos são “atingidos”
 - DV: tamanho de mensagem ~ número de destinos
- Livre de *loops*
 - > Logo após inundação e cálculo de rotas

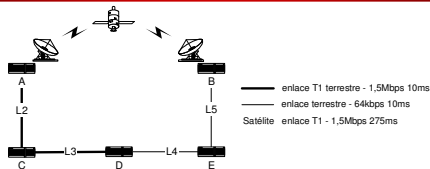
GTA/UFRJ

Múltiplas Métricas

- Cálculo do caminho mais curto
 - > Conhecimento **completo** da topologia
 - > Métrica podem ser arbitrariamente precisa
 - > DV
 - Granularidade fina + grande diferença entre a menor e a maior métrica dos enlaces = risco de convergência **muito** lenta
- Várias métricas
 - > Múltiplos estados do enlace
 - > Cálculo de múltiplas tabelas de roteamento, uma para cada métrica
 - > Indicação de que rota um determinado pacote deve seguir

GTA/UFRJ

Exemplo de Múltiplas Métricas



- $D > C > A > B$
 - > Vazão de 1,5Mbps, atraso de 295ms (escolhido pela métrica vazão)
- $D > E > B$
 - > Vazão de 64kbps, atraso de 20ms (escolhido pela métrica atraso)
- Escolha da métrica deve ser coerente
 - > Risco de formação de *loops*

GTA/UFRJ

Suporte a Múltiplos Caminhos

- Balanceamento de tráfego
 - > Diminui o atraso médio dos pacotes
 - A capacidade de transmissão é maior
 - > Redução da variação de atraso
 - Correlação entre as chegadas de pacote em um dado caminho é menor
 - > Após a falha de um caminho
 - Apenas uma parte do tráfego é re-roteada
 - Todo o tráfego é desviado se apenas uma rota é utilizada
 - > Algoritmo SPF pode ser modificado
 - Múltiplos caminhos de mesmo custo
- Mas
 - > Pode haver desordem de pacotes

GTA/UFRJ

Algoritmo SPF – Múltiplos Caminhos

1. $C = \{S\}$; $R = V - \{S\}$; $O = \{\text{caminhos de 1 salto a partir de } S\}$
Os caminhos em O possuem custo igual à métrica do seu enlace.
Ordenar os caminhos em O por ordem crescente de custos.
2. Se $O = \{\}$, ou se o primeiro caminho de O possui custo inf., marcar todos os nós em R como inalcançáveis. O algoritmo terminou.
3. Primeiro, examine P , o caminho mais curto em O . Remova P de O .
Seja U o último nó em P . Se U já pertence a C , vá para o passo 4.
Senão, P é o caminho mais curto de S para U . Mova U de R para C . Continue no passo 5.
4. Se a métrica do caminho P entre S e U é igual à distância previamente calculada entre S e U , foi encontrado um caminho de custo igual para U . Senão, ignorar P . Ir para o passo 2.
5. Construa um novo conjunto de caminhos candidatos através da concatenação de P e dos enlaces saindo de U .
Insira os novos caminhos na lista O , mantendo sua ordenação. Vá para o passo 2.

GTA/UFRJ

Suporte a Múltiplos Caminhos

- OSPF
 - > Roteamento por múltiplos caminhos de custo igual ("equal cost multi-path")
 - > Pode haver desordem de pacotes
 - Potencialmente prejudicial para o TCP
 - > Roteadores devem dividir os fluxos TCP entre as rotas
 - Exemplo
 - Hash code calculado a partir dos endereços fonte e destino e do número de porta de transporte
 - $\text{hash}(\text{pacote}) > H1$, caminho 1
 - $\text{hash}(\text{pacote}) < H1$, caminho 2
- De forma mais geral, a carga pode ser usada como métrica
 - > Deve-se tomar cuidado com a realimentação de informação

GTA/UFRJ

Rotas Externas

- Cenário: Rota de saída da rede é **única**
 - Anúncio de rota *default* (DV e LS)
- Cenário: **Múltiplas** rotas de saída
 - Anúncio de rota *default*
 - DV: caminho mais curto para a saída é o utilizado
 - Anúncio de rotas específicas
 - DV: mais entradas nos vetores de distância
 - LS: estados de enlace especiais
- Comparação
 - LS: *gateway* anuncia a métrica que convier ao enlace de saída
 - DV: métrica limitada pelo valor de infinito
 - Número de rotas externas
 - LS: tamanho da base de dados
 - DV: tamanho dos vetores de distância
 - **Mas** custo SPF = $O(N \cdot \log N)$, custo Bellman-Ford = $O(N^2)$

GTA/UFRJ

O Projeto do OSPF

- OSPF
 - Toda a funcionalidade de protocolos de estado do enlace (LS)
 - Base de dados distribuída
 - Procedimento de inundação
 - Descoberta de adjacências
 - Registros especiais para rotas externas
 - Mais
 - Separação entre estações e roteadores
 - Suporte de redes *broadcast* (Ethernet, Token Ring, FDDI)
 - Suporte de redes *não-broadcast* (X.25, ATM)
 - Divisão de redes muito grandes em áreas (roteamento hierárquico)

GTA/UFRJ

Separação de Estações e Roteadores

- Numa rede local, um estado de enlace por estação
 - não escalável
- OSPF
 - Estados de enlace
 - Enlace de roteador (*router link*)
 - Conexão entre roteadores
 - Endereço IP do vizinho
 - Enlace para rede stub (*link to a stub network*)
 - Conexão a uma rede local
 - Número de rede ou de sub-rede

GTA/UFRJ

Redes Broadcast

- Conectividade total
 - Todas as estações podem se falar diretamente
- Capacidade de difusão (nativa)
 - *broadcast* – todas as estações recebem
 - *multicast* – um grupo de estações recebe
- Problema: adjacências

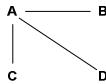


- Dados N roteadores, $N(N-1) / 2$ adjacências

GTA/UFRJ

Redes Broadcast

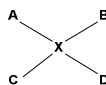
- Cada roteador anuncia
 - $N-1$ estados de enlace (enlaces para os outros roteadores)
 - 1 estado para as estações na rede (*stub network link*)
- Total = N^2 mensagens
- Redução do número de adjacências
 - Um roteador é escolhido como roteador "designado"
 - Outros roteadores na rede local estabelecem adjacências apenas com este roteador



GTA/UFRJ

Redes Broadcast

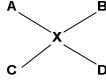
- Funcionamento
 - Primeiro passo
 - Eleição do roteador designado
 - Segundo passo
 - Outros roteadores recuperam adjacência com o roteador designado (ou se sincronizam com ele)
 - Se todos se sincronizam com **A**, todos estão sincronizados
 - N adjacências em vez de $N(N-1) / 2$
- Além disso, o número de estados do enlace é reduzido
 - Roteador virtual (**X**) representa a rede broadcast



GTA/UFRJ

Redes Broadcast

- Base de dados
 - > 2 enlaces por roteador
 - *para e do* nó virtual, X



- Anúncios
 - > roteador para X – pelo *próprio* roteador
 - > X para roteador – pelo roteador *designado*
 - Endereço IP do roteador designado *na rede broadcast*
 - "Enlaces de rede" (*network links*)
 - Métrica nula para não causar problemas de cálculo de rotas
- Procedimento de inundação

GTA/UFRJ

Redes Broadcast

- Inundação
 - > Envio de LSA (*link state advertisement*)
 - Apenas para o roteador designado
 - "all-designated-routers" – 224.0.0.6
 - > Se o LSA é novo, o roteador designado
 - Re-envia em todas suas interfaces
 - E na rede broadcast
 - "all-OSPF-routers" – 224.0.0.5
- Problema
 - > Roteador designado – ponto de falha
- Roteador designado de *backup*
 - > Roteadores mantêm adjacências com o designado e o backup
 - > O backup escuta os anúncios, silenciosamente
 - Falha do roteador designado detectada pelo protocolo *Hello*

GTA/UFRJ

Redes Não-broadcast

- Circuitos virtuais (X.25, frame-relay, ATM)
- Conectividade total
 - > Todas as estações podem se falar diretamente
- **Não** há difusão nativa
- Primeira solução
 - > Conjunto de circuitos virtuais configurados estaticamente
 - $N(N-1) / 2$ circuitos
 - Informação de roteamento inundada em todos os circuitos
 - Custo alto, considerando tarifação por volume de tráfego
- Segunda solução
 - > Redução do número de circuitos virtuais
 - Pode fazer com que o tráfego passe várias vezes pela rede paga

GTA/UFRJ

Redes Não-broadcast

- OSPF - Solução semelhante às redes *broadcast*
 - Roteador designado + roteador *backup*
 - Informação de roteamento trocada apenas com estes roteadores
 - Circuitos virtuais podem ser estabelecidos entre qualquer par de roteadores, porém *sob-demanda*
 - Apenas os circuitos entre "roteador comum" e roteador designado (e *backup*) são utilizados permanentemente
- Diferença: pacotes enviados ponto-a-ponto
 - Anúncio enviado pelo roteador designado
 - Várias mensagens ponto-a-ponto
 - Anúncio enviado por um "roteador comum"
 - Mensagem para o roteador designado + mensagem para o *backup*

GTA/UFRJ

Áreas Múltiplas

- Roteamento hierárquico
 - Divisão da rede em diferentes partes conectadas por uma espinha dorsal
 - OSPF: áreas conectadas através da área *backbone*
- Cada área se comporta como uma rede independente
 - Base de dados de estados do enlace
 - Inundação termina nas fronteiras da área
 - Roteadores calculam rotas dentro da área
- Custo proporcional ao tamanho da área, não da rede

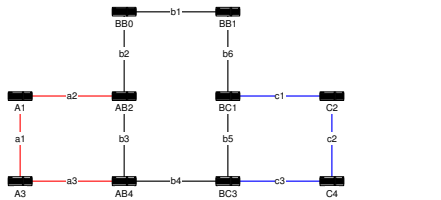
GTA/UFRJ

Conexão das Áreas Múltiplas

- Roteadores de borda de área (*area-border routers*)
 - Pertencem a várias áreas
 - Tipicamente, área de baixo nível + área *backbone*
 - Mantém várias bases de dados, uma para cada área à qual pertencem
 - Cada área deve ter pelo menos um roteador de borda de área, conectando-a ao *backbone*
 - Anunciam "enlaces de sumário" (*summary links*)
 - Descrevem rotas internas
- Roteadores de borda (*border routers*)
 - Anunciam "enlaces externos" (*external links*)
 - Descrevem rotas externas

GTA/UFRJ

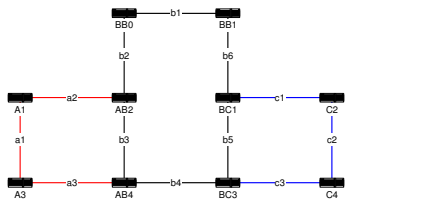
Exemplo de Áreas Múltiplas



- Área A: roteadores A1, AB2, AB4 e A3; enlaces a1, a2 e a3
- Área C: roteadores C2, C4, BC3 e BC1; enlaces c1, c2 e c3
- Área B: (*backbone*): roteadores de borda BB0 e BB1, roteadores de borda de área AB2, AB4, BC3 e BC1; enlaces b1, b2, b3, b4, b5 e b6

GTA/UFRJ

Exemplo de Áreas Múltiplas



- Base de dados Área A:
 - Estados de enlace para a1, a2 e a3, enviados por A1, AB2, AB4 e A3
 - Registros *sumário* emitidos por AB2 e AB4, descrevendo redes e sub-redes da área *backbone* e da **área C**
 - Registros *externos* emitidos por BB0 e BB1, e retransmitidos por AB2 e AB4

GTA/UFRJ

Registros Sumário

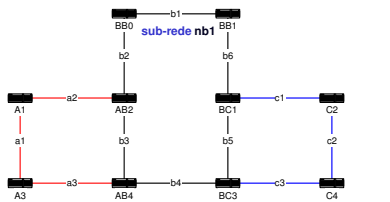
- Registros sumário
 - Representam "enlaces" entre um roteador de borda de área e uma rede na área *backbone*, ou outra área
 - Métrica igual à distância entre o roteador e a rede

 - Propagação semelhante a vetores distância, mas sem riscos de loops, devido à hierarquia estrita
 - Áreas são conectadas apenas através da área *backbone*

 - Exemplo
 - b1: rede Ethernet identificada pelo número de sub-rede nb1

GTA/UFRJ

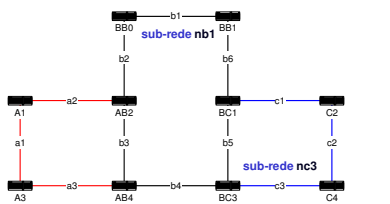
Exemplo de Áreas Múltiplas



- AB2 anuncia enlace sumário para nb1 com métrica = $b1 + b2$
- AB4 anuncia enlace sumário para nb1 com métrica = $b1 + b2 + b3$

GTA/UFRJ

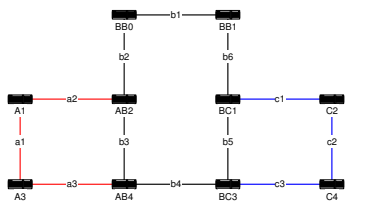
Exemplo de Áreas Múltiplas



- BC3 anuncia enlace sumário para nc3 com métrica = $c3$
- BC1 anuncia enlace sumário para nc3 com métrica = $c1 + c2 + c3$
- AB4 e AB2 calculam o caminho mais curto para nc3 através de BC3
- AB4 anuncia na área A enlace sumário para nc3 com métrica = $c3 + b4$
- AB2 anuncia na área A enlace sumário para nc3 com $m. = c3 + b4 + b3$

GTA/UFRJ

Rotas Externas

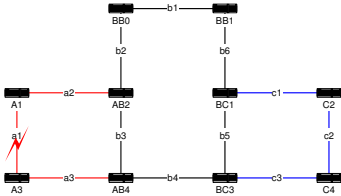


- Registros externos, gerados por BB0 e BB1, são copiados inalterados na base de dados da **área A** (e C)
- Registros de sumário enviados por AB2 e AB4 dizem como chegar a cada roteador na área de *backbone*
 - Portanto, rotas externas passando por BB0 e BB1 podem ser calculadas precisamente

GTA/UFRJ

Área Particionada

- Não há, *a priori*, como diferenciar destinos dentro de uma área
- Como evitar o envio de pacotes para A3 através de AB2?

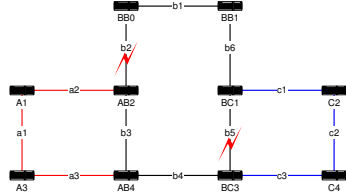


- Roteadores de borda de área enviam registros sumário apenas para redes e sub-redes **alcançáveis**, em vez de para todas as redes e sub-redes da área
- Entradas na área *backbone* são tão precisas *quanto necessário*

GTA/UFRJ

Área Backbone Particionada

- Não há conectividade através de enlaces do *backbone*, apenas
- Poderia ser restaurada através da área C



- Solução: enlace virtual entre BC1 e BC3
 - Atualizações da base de dados do *backbone* trocados através da rede C
 - Descrição do enlace virtual na base de dados
 - Métrica: $c1 + c2 + c3$

GTA/UFRJ

Áreas Stub

- Geralmente, os registros mais numerosos na base de dados de estados de enlace correspondem às rotas externas
 - ~60.000 em 1999
- Algumas *pequenas* áreas são conectadas por apenas um roteador de borda de área ao *backbone*
- OSPFv2
 - Área *stub* (*stub area*)
 - Todas as rotas externas são resumidas por uma rota *default*
 - A área pode ter mais de um roteador de borda de área, **mas**
 - Não há como escolher o roteador de saída de acordo com o destino
 - Não há como configurar enlaces virtuais passando por uma área *stub*

GTA/UFRJ

Base de Dados de Estados do Enlace

- Composta de *registros de estado do enlace*
 - *Link State (LS) records*
- Tipos de LS
 - Enlace de roteador (tipo 1)
 - Enlace de rede (tipo 2)
 - Enlace sumário (de rede IP) (tipo 3)
 - Enlace sumário (para um roteador de borda) (tipo 4)
 - Enlace externo (tipo 5)

GTA/UFRJ

Cabeçalho de Estado do Enlace

- Comum a todos tipos de registro de estado do enlace

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1	1	2	3
LS age	options	LS type	
Link State ID			
Advertising router			
LS sequence number			
LS checksum		length	

- Advertising router – um dos endereços IP do roteador (identificador OSPF)
- Age – tempo em segundos desde o primeiro anúncio deste LS
- Option
 - E – enlace externo (utilizado pelo prot. Hello)
 - T – indica se o roteador suporta roteamento por TOS

GTA/UFRJ

Cabeçalho de Estado do Enlace

- Link State ID – identificação do enlace
 - geralmente, endereço IP (depende do tipo de enlace)
 - (link state ID, advertising router, LS type) devem identificar unicamente o registro
- Checksum – calculado como no cabeçalho IP
 - Sobre cabeçalho + conteúdo
- Length – comprimento total do registro
- LS sequence number
 - $N = 2^{31}$ (usados negativos)
 - Números variam entre 1 - N e N - 2; - N e N -1 não utilizados

GTA/UFRJ

Cabeçalho de Estado do Enlace

- Roteador é ligado
 - > Começa de 1 – N e incrementa o número
 - > Em N - 2, próximo número = 0 (N - 1 não usado)
 - > Números agora em seqüência cíclica na parte positiva
- Comparação
 - > Um número negativo – comparação direta
 - > Ambos positivo ou nulo – comparação cíclica
 - a e b, com a menor que b
 - Se $(b - a) < (N - 1) / 2$
 - b mais recente que a
 - Senão, a mais recente que b
 - > Problema se um roteador re-inicia várias vezes em seqüência

GTA/UFRJ

Enlace de Roteador

- O “registro de estado de enlace de roteador” lista todos os enlaces que saem do roteador

0	1	2	3
0	0	0	0
V	E	B	
Link ID		number of links	
Link data			
Type	# TOS	TOS	0 metric
TOS=x	0	TOS	x metric
TOS=y	0	TOS	y metric
---	---	---	---
TOS=z	0	TOS	z metric

- Link ID – identificador OSPF do roteador
- E = 1 – roteador de borda de área
- B = 1 – roteador de borda
- V = 1 – roteador é ponta de enlace virtual passando pela área

GTA/UFRJ

Enlace de Roteador

- Type (tipo do enlace)
 - = 1 – enlace ponto-a-ponto
 - Link ID = identificador OSPF
 - Link data = endereço IP desta interface do roteador
 - = 2 – conexão a uma rede de trânsito
 - Link ID = endereço IP da interface do roteador designado
 - Link data = endereço IP desta interface do roteador
 - = 3 – conexão a uma rede stub
 - Link ID = número da rede ou sub-rede IP
 - Link data = máscara de rede ou sub-rede correspondente
- #TOS – número de tipos de serviço anunciados
 - > TOS 0 obrigatório
 - > TOS = x – número do tipo de serviço

GTA/UFRJ

Enlace de Rede

- Enviados por roteadores designados para redes de trânsito
- Link state ID = endereço IP da interface nesta rede

0 0 1 2 3 4 5 6 7 8 9 0	1 1 2 3 4 5 6 7 8 9 0	2 1 2 3 4 5 6 7 8 9 0	3 1 2 3 4 5 6 7 8 9 0 1
Network mask			
Attached router			
....			
Attached router			

- Máscara de rede ou sub-rede IP
- Identificador OSPF de todos os roteadores conectados à rede (que possuem adjacência com o roteador designado)

GTA/UFRJ

Enlace de Sumário

- Enlaces de sumário para redes IP (tipo 3) e para roteadores de borda (tipo 4) são enviados por roteadores de borda de área

- Um anúncio separado para cada destino

- Link State ID
 - Número de rede ou sub-rede IP (tipo 3)
 - Endereço IP do roteador (tipo 4)

0 0 1 2 3 4 5 6 7 8 9 0	1 1 2 3 4 5 6 7 8 9 0	2 1 2 3 4 5 6 7 8 9 0	3 1 2 3 4 5 6 7 8 9 0 1
Network mask			
TOS=0	0	TOS	0 metric
TOS=x	0	TOS	x metric
...
TOS=z	0	TOS	z metric

- Máscara
 - da rede ou sub-rede (tipo 3)
 - 0xFFFFFFFF se roteador de borda (tipo 4)

- Métricas
 - Similar ao "enlace de roteador"
 - Número de TOS não é necessário (deduzido do tamanho)

GTA/UFRJ

Enlace Externo

- Enlaces Externos são anunciados por roteadores de borda

- Link State ID
 - Número de rede ou sub-rede IP do destino

0 0 1 2 3 4 5 6 7 8 9 0	1 1 2 3 4 5 6 7 8 9 0	2 1 2 3 4 5 6 7 8 9 0	3 1 2 3 4 5 6 7 8 9 0 1
Network mask			
E,TOS=0	0	TOS	0 metric
External route tag (0)			
E,TOS=x	0	TOS	x metric
External route tag (x)			
...
TOS=z	0	TOS	z metric
External route tag (z)			

- Campos TOS
 - E=1 – a métrica anunciada para este TOS não é compatível com a métrica usada internamente
 - E=0 – ok, comparação entre métrica interna e externa é válida

- External route tag
 - Utilizada pelos roteadores de borda, não é examinada pelo OSPF

GTA/UFRJ

Cálculo de Rotas e TOS

Código OSPF	Valor de TOS (RFC 1349)
0	0000 serviço normal
2	0001 minimizar custo financeiro
4	0010 maximizar confiabilidade
8	0100 maximizar vazão
16	1000 minimizar atraso

- Pacote roteado segundo as rotas para o TOS correspondente, mas
 - > Roteadores não são obrigados a anunciar métricas para todo TOS
 - Se TOS≠0, sem métrica, é tomada a métrica do TOS 0
 - > Roteadores que não suportam TOS o informam no cabeçalho do LS
 - Rotas calculadas para TOS ≠ 0 *sem* estes roteadores
 - Se destino inalcançável, rota para TOS 0 é usada
- Métricas
 - > Número inteiro
 - > Quanto menor melhor

GTA/UFRJ

Cálculo de Rotas e TOS

- Banda passante
 - > 10^8 / capacidade(bps) – número de segundos para transmitir 10^8 bits
 - > Métrica = 1 para 100Mbps ou mais [RFC1850 – OSPF MIB]
 - > 1 também é o valor *default* desta métrica
- Atraso, confiabilidade e custo não definidos
 - > Administrador pode configurá-los
 - > Regra geral: usar a métrica *default*, aumentar a métrica para enlaces "especiais"
 - Atraso: enlaces de satélite, alto tempo de propagação
 - Custo: enlaces pagos por volume de tráfego
 - Confiabilidade: enlaces de rádio

GTA/UFRJ

Cálculo de Rotas: Outras Métricas

- Seja caminho C contendo os enlaces l_1, l_2 , e l_3
- Probabilidade de perda
 - > $p(C) = 1 - ((1 - p(l_1)) * (1 - p(l_2)) * (1 - p(l_3)))$
- Probabilidade de sucesso
 - > $s(C) = s(l_1) * s(l_2) * s(l_3)$
 - > $\log(s(C)) = \log(s(l_1)) + \log(s(l_2)) + \log(s(l_3))$
 - > Obs.: $0 \leq s(l) \leq 1 \rightarrow -\infty \leq \log(s(l)) \leq 0$
 - > $|\log(s(C))| = |\log(s(l_1))| + |\log(s(l_2))| + |\log(s(l_3))|$
 - > (Pode-se utilizar Dijkstra)

GTA/UFRJ

Os Protocolos dentro do OSPF

o OSPF

- > Executado sobre o IP
 - Tipo de protocolo = 89
- > Sub-protocolos
 - **Protocolo Hello**
 - Alcançabilidade de vizinhos
 - Eleição do roteador designado
 - **Protocolo de Troca (Exchange)**
 - Sincronização inicial da base de dados de estados do enlace
 - **Protocolo de Inundação (Flooding)**
 - Atualização da base de dados de estados do enlace
 - Sincronização da idade dos registros

GTA/UFRJ

Cabeçalho Comum dos Pacotes OSPF

- o Version # - 2
- o Type – tipo de pacote OSPF
- o Length – comprimento
- o Router ID
 - > identificador OSPF (endereço IP)
- o Area ID
 - > 0 – área backbone
 - > Utiliza-se na prática o número de rede como Area ID
- o Checksum
 - > Como no IP, sobre todo o pacote *menos* os 8 bytes de autenticação
- o Autype
 - > 0 – sem autenticação
 - > 1 – autenticação simples
 - > 2 – autenticação criptográfica

Version #	Type	Packet length
Router ID		
Area ID		
Checksum	Autype	
Authentication		
Authentication		

GTA/UFRJ

Autenticação Criptográfica

o Campo de autenticação redefinido

0	1	2	3
0	Key ID	Auth. data length	
Cryptographic sequence number			

- o Key-ID
 - > identifica a senha secreta (chave) e algoritmo (MD5)
- o Auth Data Len
 - > comprimento dados de autenticação
- o MD5 digest (16 bytes)
 - > calculado sobre o pacote OSPF, chave secreta, campos de enchimento e comprimento
 - > concatenado no fim do pacote (trailer)
 - tamanho não conta no tamanho do pacote

GTA/UFRJ

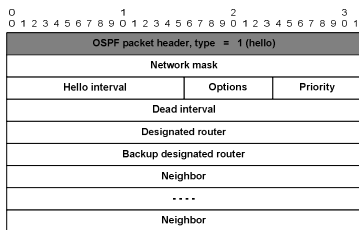
Autenticação Criptográfica

- Na recepção, verificação através da utilização da mesma chave secreta
- Número de seqüência (Cryptographic sequence number)
 - > Proteção contra *replay attacks*
 - > Roteador rejeita mensagem com número de seqüência não superior ao recebido anteriormente para mesmo KeyID
- A cada chave são associadas 4 constantes de tempo, em que o roteador
 - > começa a receber mensagens com esta chave
 - > começa a enviar mensagens com esta chave
 - > pára de enviar mensagens com esta chave
 - > pára de aceitar mensagens com esta chave

GTA/UFRJ

O Protocolo Hello

- Objetivos
 - > Checar que enlaces estão operacionais
 - > Eleger os roteadores designado e designado de backup, em redes *broadcast e não-broadcast*



GTA/UFRJ

O Protocolo Hello

- Network mask
 - > Se não houver, máscara correspondente à rede (A, B ou C)
- Hello interval
 - > Intervalo entre as mensagens, em segundos
- Designated router
 - > Endereço do roteador designado, ou zero se não definido
- Backup designated router
 - > Endereço do roteador *backup* designado, ou zero se não definido
- Neighbor (n vezes)
 - > Lista dos vizinhos dos quais foi ouvido um *hello*, nos últimos *dead interval* segundos
- Intervalos hello e dead
 - > Parâmetros do enlace, configurados pelo administrador

GTA/UFRJ

Conectividade

- Options
 - T = 1 – roteador é capaz de roteamento por TOS
 - E = 1 – roteador é capaz de enviar e receber rotas externas (E = 0 – roteador conectado à área *stub*)
- Enlace operacional se
 - Dados podem fluir nas duas direções
 - Roteadores de acordo com valor do bit E
- Conectividade bi-direcional
 - Roteador presente na lista de vizinhos anunciada pelo roteador remoto
 - Se não, conectividade uni-direcional (por enquanto)
 - O enlace não deve ser utilizado para o roteamento
- Dead interval
 - Não é a única forma de reconhecer morte do vizinho
 - Notificação da camada inferior pode ser utilizada

GTA/UFRJ

Eleição do Roteador Designado

- Procedimento usa campo *priority* do pacote Hello
 - Prioridade de cada roteador vai de 0 a 255
 - Roteador com maior prioridade é selecionado, **mas**
 - Quando roteador designado cai, outro roteador é selecionado
 - Quando o primeiro é re-ligado, não retoma o papel de designado imediatamente
 - Limita-se a frequência de mudança do roteador designado
 - Prioridade = 0 - roteador **não** é selecionado como designado
- Quando roteador é ligado
 - Estado de espera (durante "dead interval")
 - Envio de pacotes *hello*; designado = 0, backup = 0
 - Não se candidata
 - Ouve pacotes *hello*
 - Para cada vizinho, armazenar
 - Prioridade
 - Estado da conectividade
 - Se é candidato a designado ou backup

GTA/UFRJ

Eleição do Roteador Designado

- Eleição
 1. Se um ou vários vizinhos se propuseram como roteador de backup, o de maior prioridade é selecionado roteador de backup (roteadores que se propuseram designado não são elegíveis backup)
 2. Se nenhum vizinho se propôs como backup, o vizinho com a maior prioridade é selecionado
 3. Se um ou vários vizinhos se propuseram como roteador designado, o de maior prioridade é selecionado roteador designado.
 4. Se nenhum vizinho se propôs como roteador designado, o roteador de backup se torna o designado
- Algoritmo executado permanentemente
 - Passo 2 é executado apenas ao fim do período de espera
 - Após mudança de vizinho, eleição e construção de adjacências
- Endereços de envio
 - Configurados em redes não-broadcast

GTA/UFRJ

O Protocolo de Troca (*Exchange*)

- Objetivo
 - > Sincronização inicial da base de dados
 - Após estabelecimento de conectividade
 - Com vizinho no enlace ponto-a-ponto
 - Com roteador designado senão
 - > Após a sincronização inicial, protocolo de inundação
- Protocolo Assimétrico
 - > Passo 1: escolha do mestre e escravo
 - > Passo 2: troca da descrição das bases de dados
 - > Passo 3: listagem dos registros a serem pedidos
- Utiliza pacotes de descrição da base de dados OSPF

GTA/UFRJ

Funcionamento do Protocolo de Troca

- Funcionamento
 - > Roteador que inicia o procedimento
 - Envia pacote com $I=1, M=1, MS=1$
 - Núm. de Seq. DD = número inédito no outro roteador
 - > Vizinho concorda em ser escravo
 - Envio de pacote $I=1, M=1, MS=0$
- Problemas
 - > Perdas
 - Reconhecida por *timeout*, resolvidas pelo re-envio do pacote
 - > Colisões
 - Roteador recebe pedido de *mestre*, enquanto estava esperando resposta ao *próprio* pedido de *mestre*
 - O *mestre* é o roteador de maior endereço IP

GTA/UFRJ

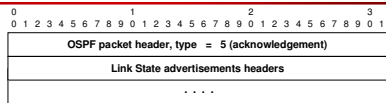
Funcionamento do Protocolo de Troca

- Descrição da base de dados é transmitida pelo *mestre*
 - > $I=0, MS=1, M=1$ para todos os pacotes, exceto o último que possui $M=0$
 - > Cada pacote provoca o envio de um reconhecimento
 - Mesmo núm. de seq. DD, $MS=0$
 - Descrição do registro na base de dados do escravo
- Proteção contra perdas de reconhecimentos
 - > Se o mestre não recebe *ack* em *x* segundos, re-envio da descrição
 - > Escravo deve re-enviar *ack* para mesmo núm. de seq. DD
- Critério de parada
 - > Quando mestre terminou, pacote com $M=0$
 - > Se escravo ainda possui registros a transmitir
 - *Ack* enviado com $M=1$
 - Mestre envia pacotes com descrição vazia até *ack* enviado com $M=0$

GTA/UFRJ

Protocolo de Inundação

○ Reconhecimento



- Pode reconhecer vários LSs, portanto
 - Envio não é imediato
 - Nem tão longo, que cause retransmissão
- Redes broadcast
 - LSs de vários vizinhos podem ser reconhecidos no mesmo pacote
 - Reconhecimento enviado a "all-OSPF-routers"
- Anúncios duplicados devem ser reconhecidos imediatamente
- Com roteador designado
 - Reconhecimento desnecessário, re-envio do update cumpre esta tarefa

GTA/UFRJ

Controle da Idade dos Registros

○ Remoção de LSs desatualizados deve ser sincronizada

○ Mecanismo de envelhecimento (*aging*)

- Idade = 0 no envio do registro
- Idade++ a cada re-envio
- Idade++ a cada segundo, daí por diante
- Se Idade = IdadeMax (1 hora)
 - Registro não utilizado para cálculo de rotas
 - Remoção condicionada ao aviso aos vizinhos

GTA/UFRJ

Controle da Idade dos Registros

○ Recepção de anúncio duplicado

- Se idade recebida = IdadeMax
 - Aceito (e será eventualmente apagado) e Re-enviado
- Se pequena diferença (maioria dos casos)
 - Ignorado e Não re-enviado
 - (conseqüência do envio por diferentes caminhos)
- Grande diferença (> 15min., casos raros)
 - Registro mais recente é guardado
 - Pode acontecer com um roteador que rebootou e utilizou o mesmo número de seqüência

○ Registros têm a idade re-iniciada periodicamente (ao serem re-enviados)

- ~30 min., pelo menos

GTA/UFRJ

Balanceamento de Carga

- Estado de Enlace Opaco (*opaque LS record* [RFC2370])
 - > pode ser enviado a todos os nós, mesmo os que não o entendem

0								1								2								3																	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Link State age																Options								LS type																	
Opaque type								Opaque ID																																	
Advertising router																																									
Link State sequence number																																									
Link State checksum																LS length																									
Opaque Content																																									
...																																									

GTA/UFRJ

LSA Opaco

- LS Type
 - > 9 – inunda apenas um enlace
 - > 10 – inunda apenas a área OSPF
 - > 11 – inunda toda a rede OSPF
- 32-bit Link State ID
 - > Opaque Type
 - 0 – 127 – tipos globais, definidos pelo IANA
 - 128 – 255 – Uso Experimental
 - > Opaque ID
 - Identificador escolhido pelo roteador anunciante

GTA/UFRJ

Optimized Multi Path

- Proposta de Curtis Villamizar
- Objetivo
 - > Direcionar mais tráfego para o enlace menos carregado
- Utiliza LSA opaco de tipo 10 (LS Type = 10)
 - > inunda a área OSPF
- Cada “LSA de carga do enlace” é paralelo a um estado do enlace comum (LS Type = 1..5)
 - > Utiliza-se também “LSA de carga do *caminho*”

GTA/UFRJ

Algoritmo OMP

1. Algoritmo SPF – cálculo dos caminhos para os destinos a partir dos vizinhos do roteador.
2. Calcular um conjunto de caminhos candidatos a partir dos enlaces locais e dos caminhos a partir dos vizinhos.
3. Guardar os *caminhos de custo mínimo, ou dentro de margem de custo aceitável.*

Para evitar loops, caminhos não minimais são guardados apenas se não contêm enlaces virtuais, e se o roteador vizinho está mais próximo do destino que o roteador de origem.

GTA/UFRJ

Algoritmo OMP

- o Conjunto inicial de caminhos
 - > Tráfego distribuído de acordo com técnica *hash* sobre o cabeçalho dos pacotes
- o Roteadores monitoram o tráfego e enviam informação quando a carga varia
- o Intervalo de inundação
 - > ~20 min. em baixa carga
 - > 30~60s em situações de carga alta e variável
- o Balanceamento reavaliado
 - > nova informação de tráfego foi recebida
 - > a cada 30s em alta carga, ou 5 min. em baixa carga
- o A idéia é convergir, lentamente, para o melhor balanceamento de carga

GTA/UFRJ

Algoritmo OMP

- o Após mudança de topologia
 - > Cálculo de novo conjunto de caminhos inicial
- 1. Se houve caminho removido, a carga é dividida entre os caminhos restantes
- 2. Novos caminhos não recebem carga.
Sua parcela de carga aumenta gradualmente após cada intervalo de cálculo de carga. (Novos caminhos são "suspeitos": podem conter enlaces instáveis.)
- 3. Se todos os caminhos foram substituídos por novos caminhos, distribuir a carga entre estes.

GTA/UFRJ

OSPF x RIP

- RIP
 - > Vetores de distância
 - > 2 mensagens de controle
 - > DVs enviados a cada 30s
 - > Tabela de roteamento
- OSPF
 - > Estados do enlace
 - > 5 mensagens de controle e 3 "sub-protocolos"
 - > LSAs com reconhecimentos
 - > Tabela de roteamento + tabela de estados do enlace
- OSPF é mais complexo, porém mais eficiente que o RIP
 - > "O OSPF calcula melhores rotas com menos mensagens"

GTA/UFRJ
