



IMD0043

SPANNING TREE PROTOCOL (STP)

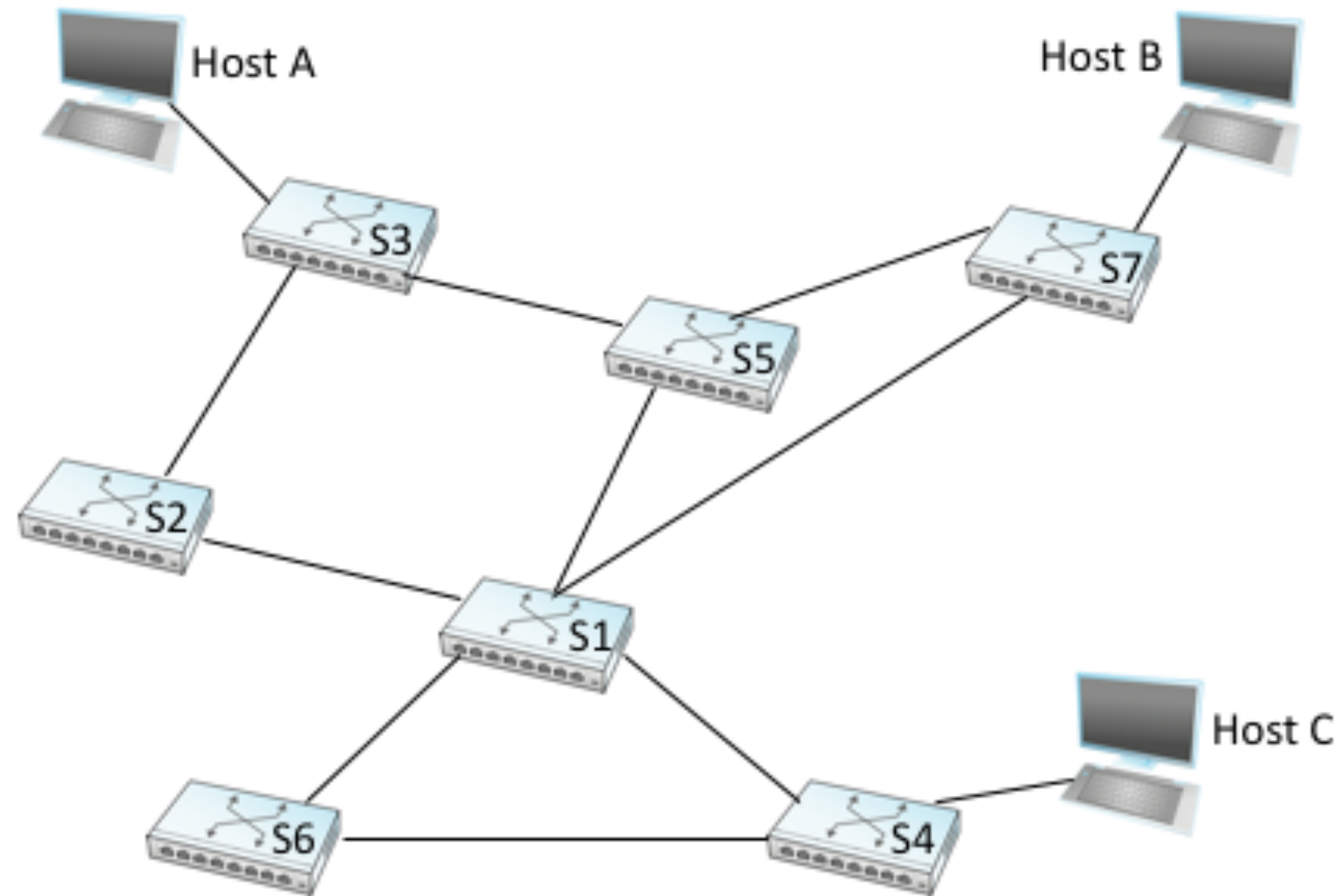
REVISANDO...

Forwarding acontece quando o *switch* tem a entrada para o MAC de destino. O quadro será encaminhado somente para a porta indicada na tabela.

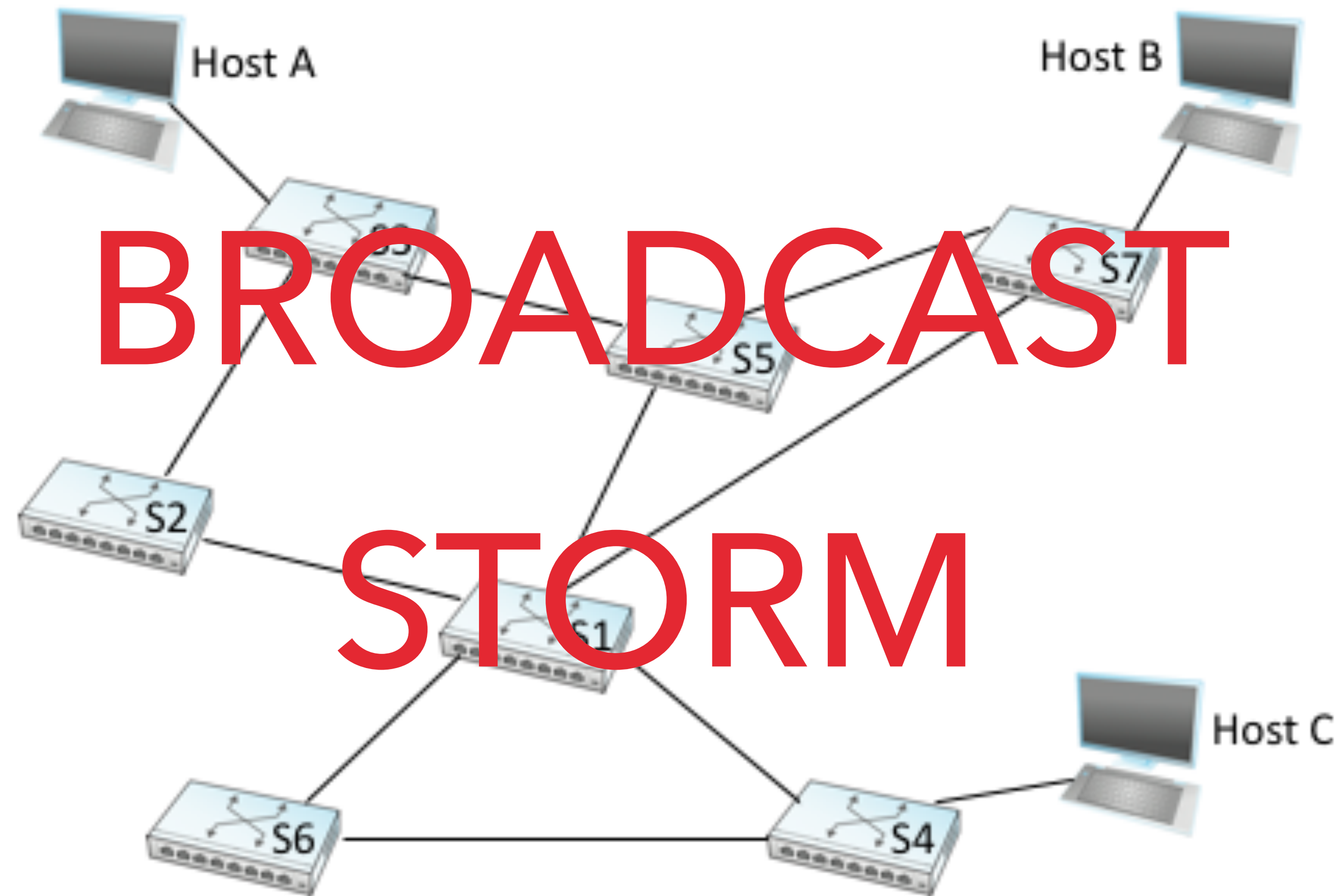
Flooding acontece quando o *switch* não tem entrada para o MAC de destino. O quadro é enviado para todas as portas, com exceção da porta de origem.

Filter acontece quando o MAC de origem e de destino estão localizados na mesma porta. O quadro é simplesmente descartado.

ENCAMINHAMENTO EM REDES LOCAIS (LAN)



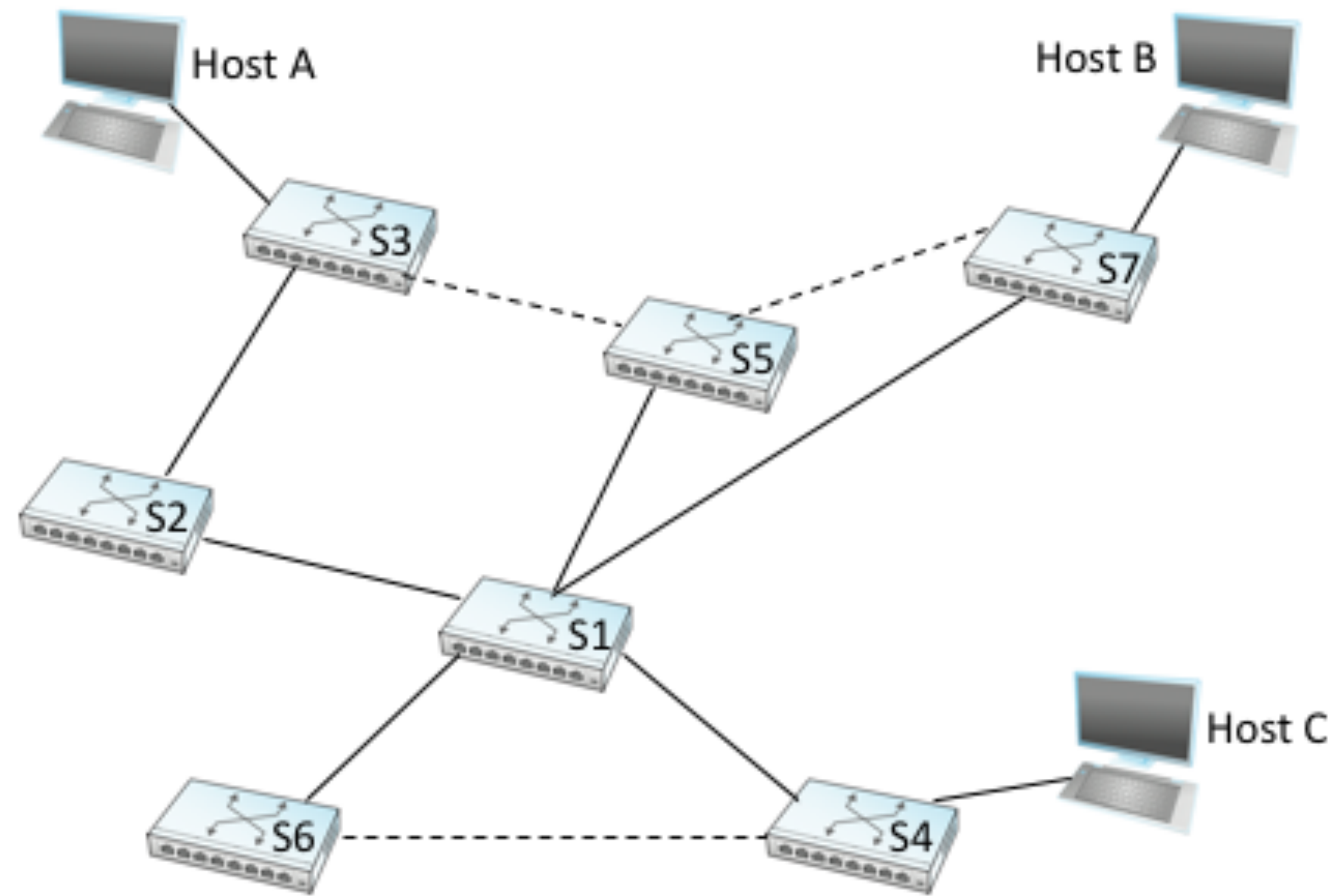
ENCAMINHAMENTO EM REDES LOCAIS (LAN)



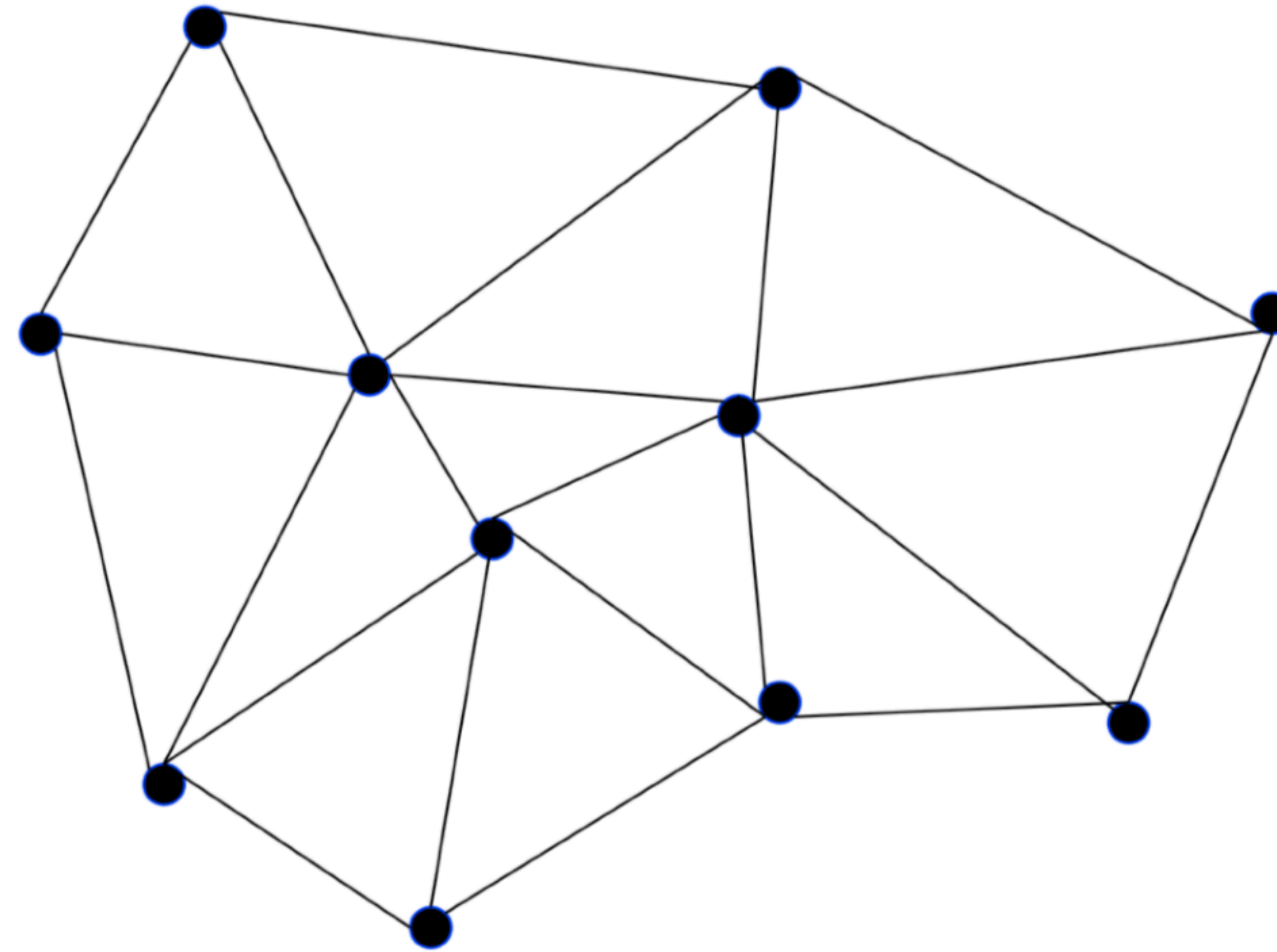
COMO EVITAR A TEMPESTADE DE BROADCAST?

Se livrar dos **loops**!

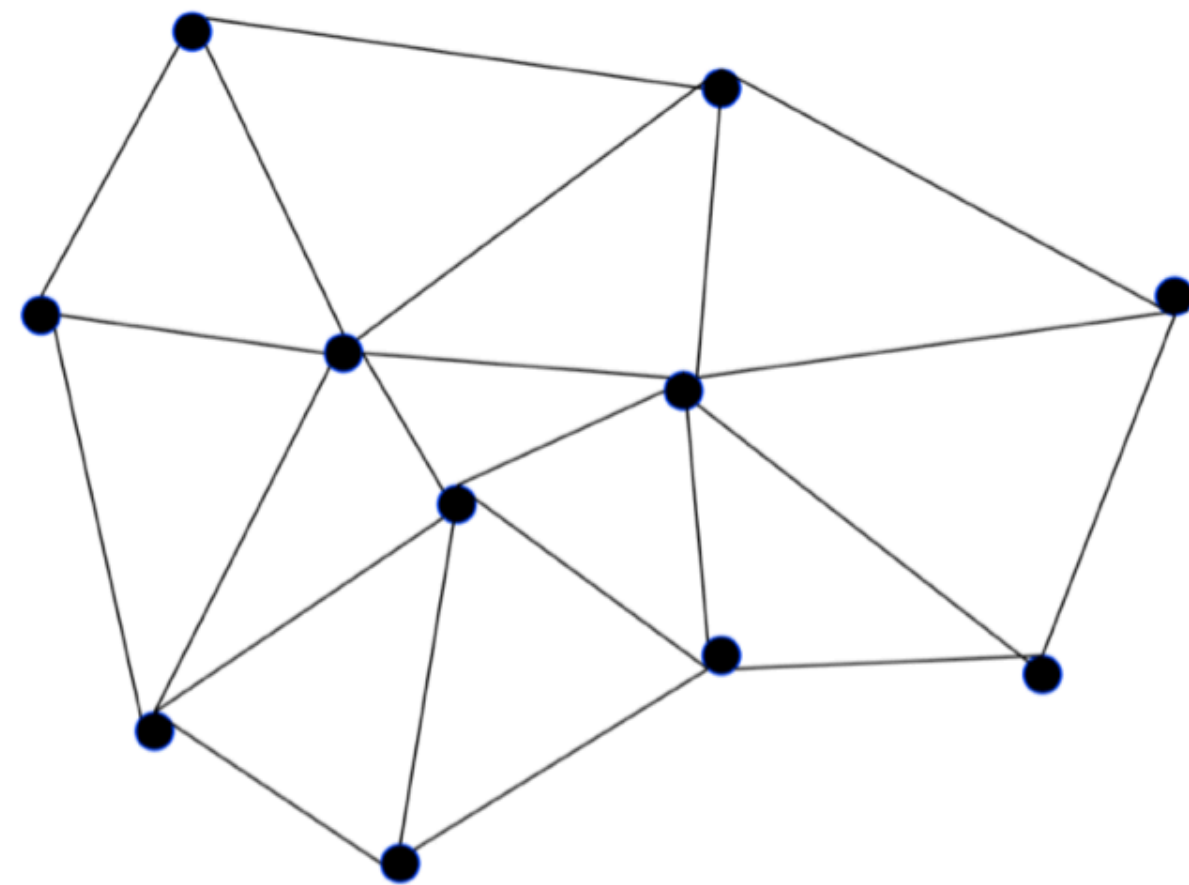
SPANNING TREE PROTOCOL (STP)



SPANNING TREE PROTOCOL (STP)

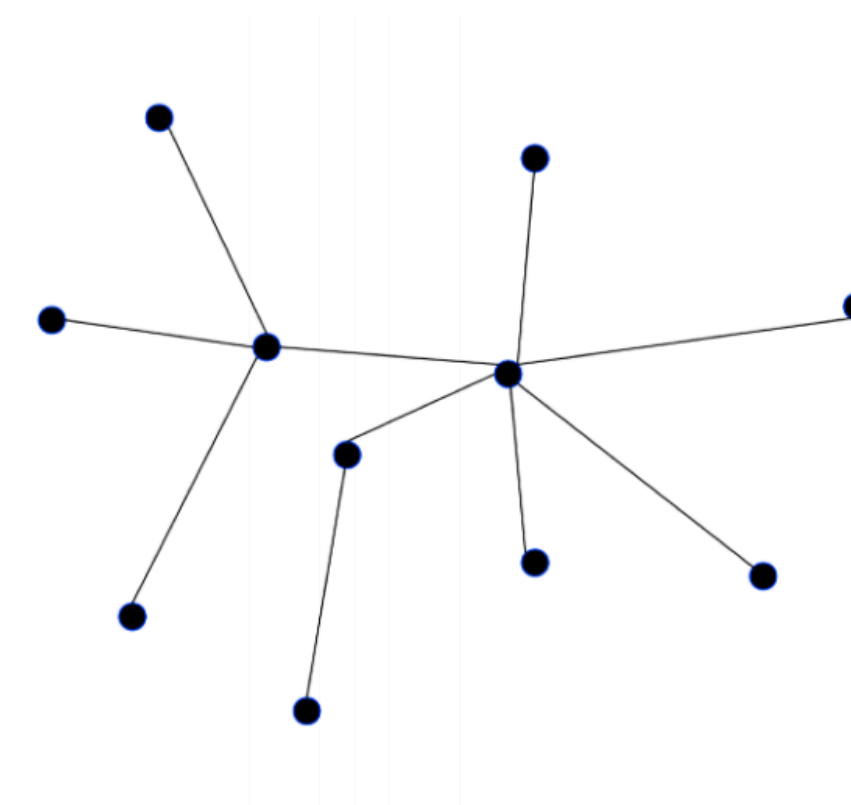
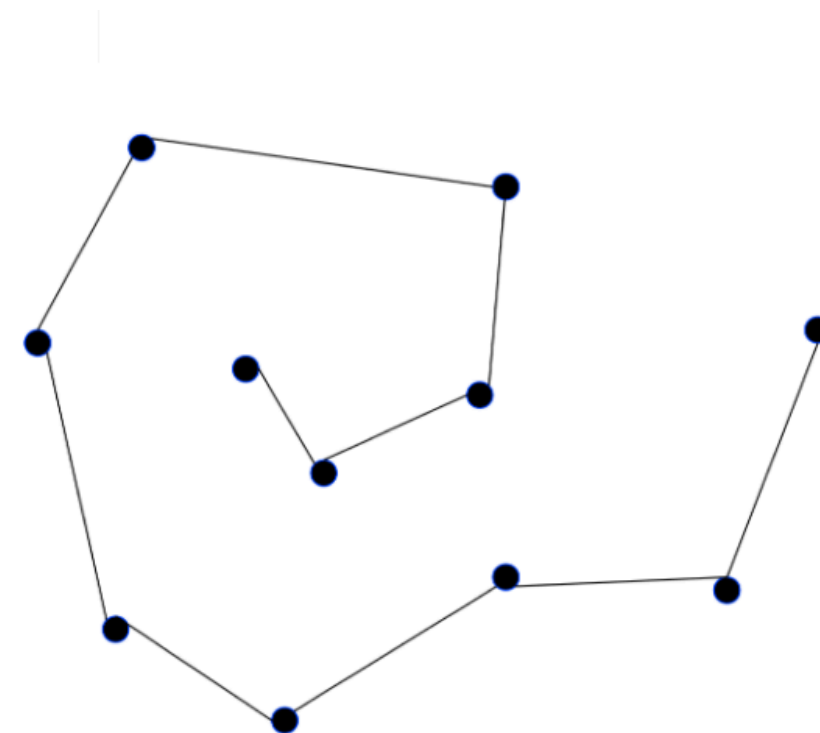
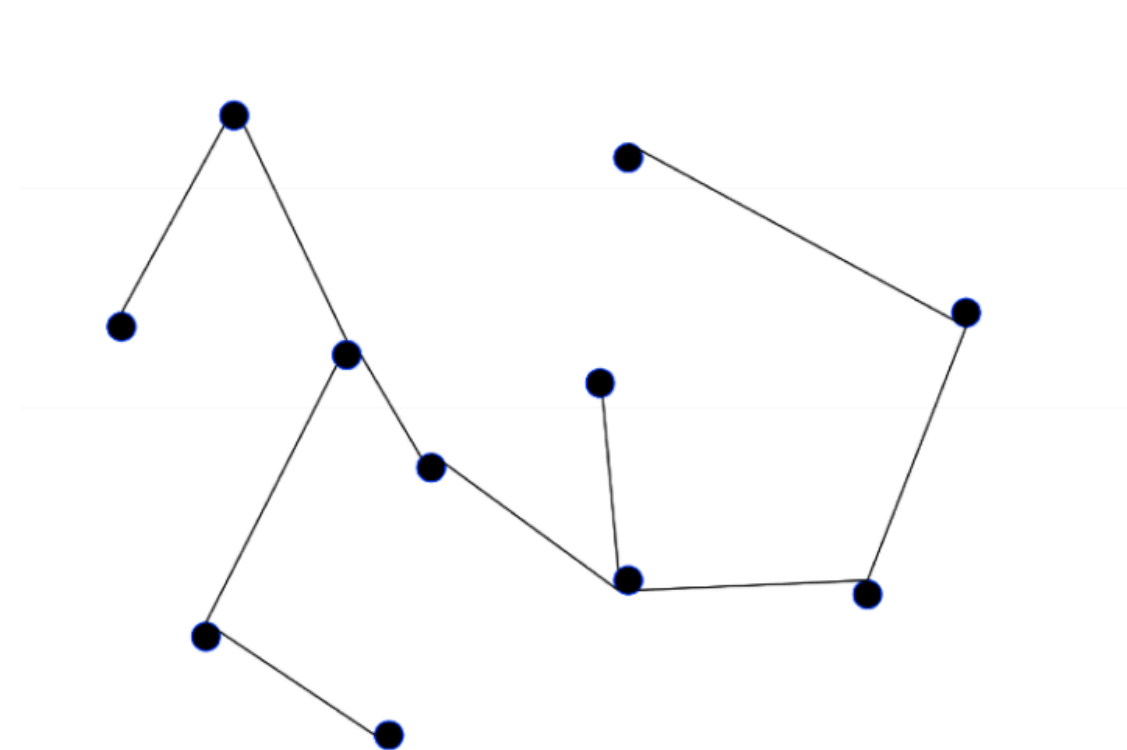


SPANNING TREE PROTOCOL (STP)



Subgrafo que contém todos os nós mas nem todas as arestas

Links fora da *spanning tree* não realizam encaminhamento



ABORDAGEM DO ALGORITMO DE SPANNING TREE

- ▶ Pegue topologia arbitrária;
- ▶ Selecione um subconjunto de *links* que formem uma *spanning tree*;
- ▶ Só encaminhe quadros pela *spanning tree*:
 - ▶ Sem *loops*
 - ▶ Sem *broadcast storm*

SPANNING TREE PROTOCOL

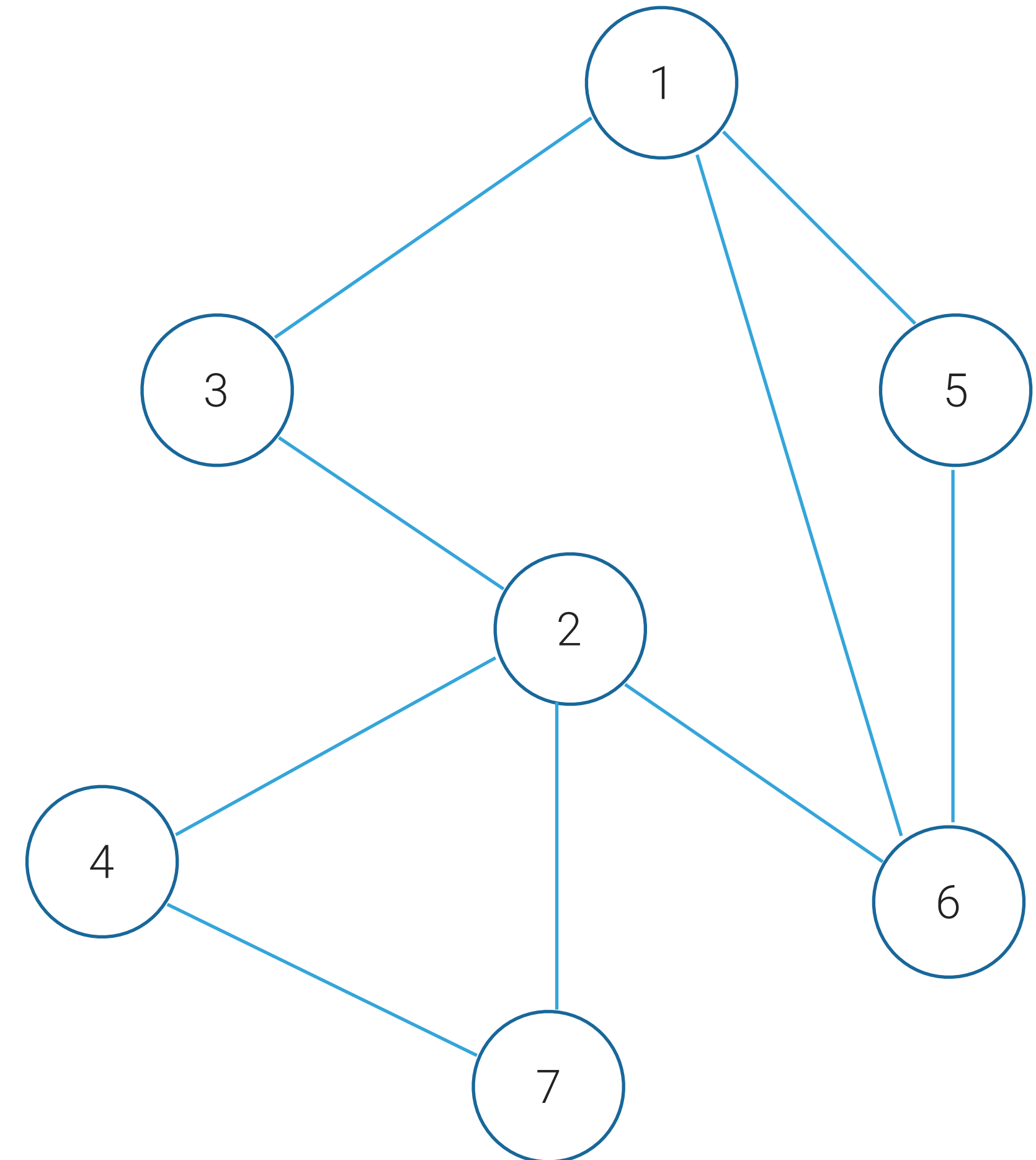
- ▶ Protocolo no qual os *switches* definem a *spanning tree*
- ▶ Propriedades interessantes:
 - ▶ não requer configuração;
 - ▶ se auto-adapta (*plug-n-play*).

ALGORITMO TEM DOIS ASPECTOS...

- ▶ Selecione uma raiz (*root*)
 - ▶ Destino no qual todos os menores caminhos chegarão
 - ▶ Seleciona o nó com o menor identificador (MAC)
- ▶ Calcule os menores caminhos para a raiz
 - ▶ Nenhum menor caminho pode ter um *loop*
 - ▶ Mantenha somente os *links* que fazem parte de um menor caminho
 - ▶ Defina alguma regra de desempate para que se mantenha somente um menor caminho para nó
 - ▶ ex: escolher o caminho a partir do vizinho com menor identificador

EXEMPLO STP

- ▶ Vamos executar o STP neste exemplo:
 - ▶ Assuma que todos os *links* tem distância 1
 - ▶ Mensagens (Y, d, X)
 - ▶ Propondo Y como *root*
 - ▶ Da origem X
 - ▶ com distância d entre X e Y
 - ▶ Switches elegem o nó com menor ID (MAC)
 - ▶ Y nas mensagens
 - ▶ Cada *switch* determina se um *link* está no menor caminho para o *root*
 - ▶ Se não está, exclui da árvore

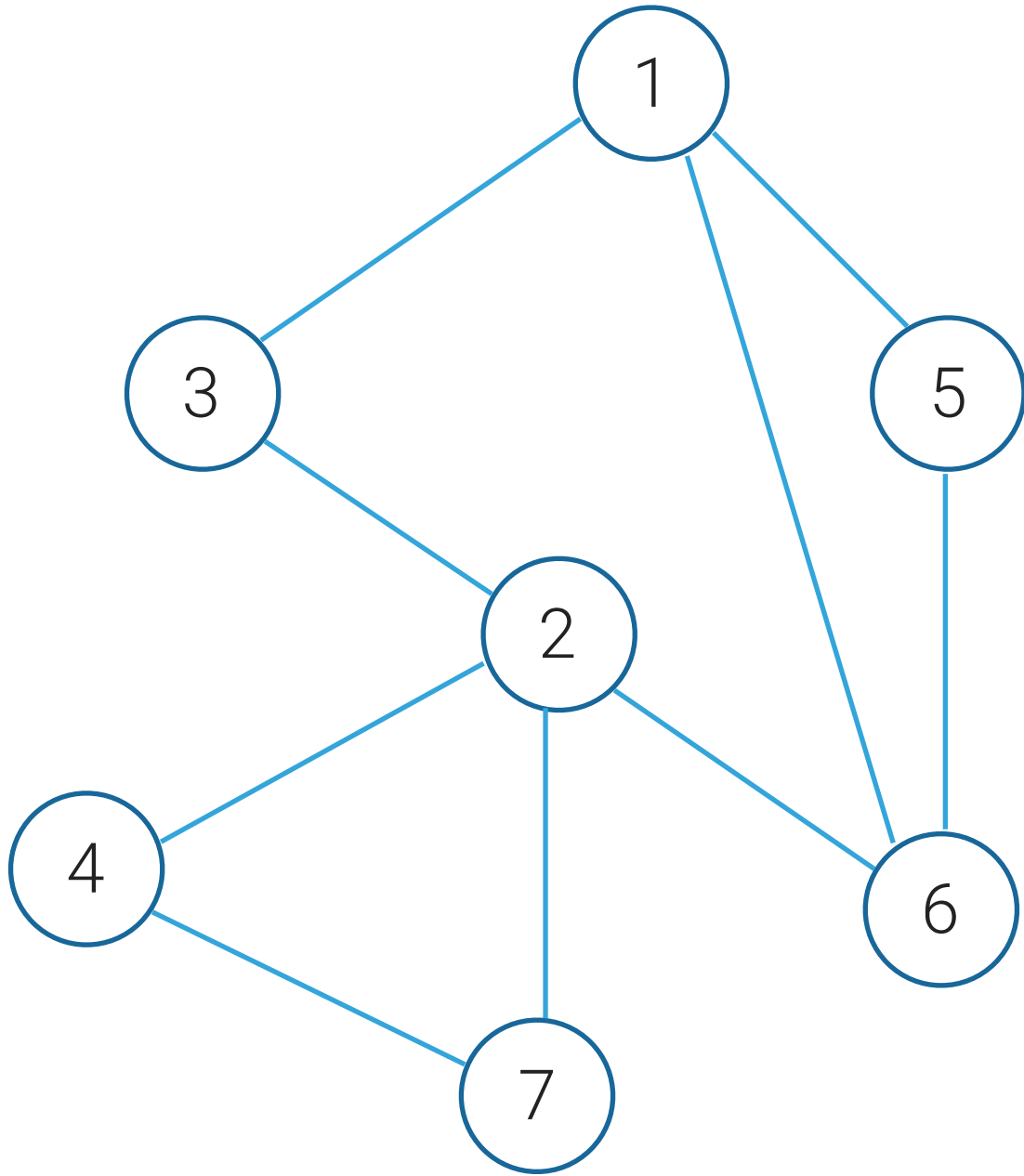


PASSOS DO ALGORITMO STP

Mensagens (Y, d, X)

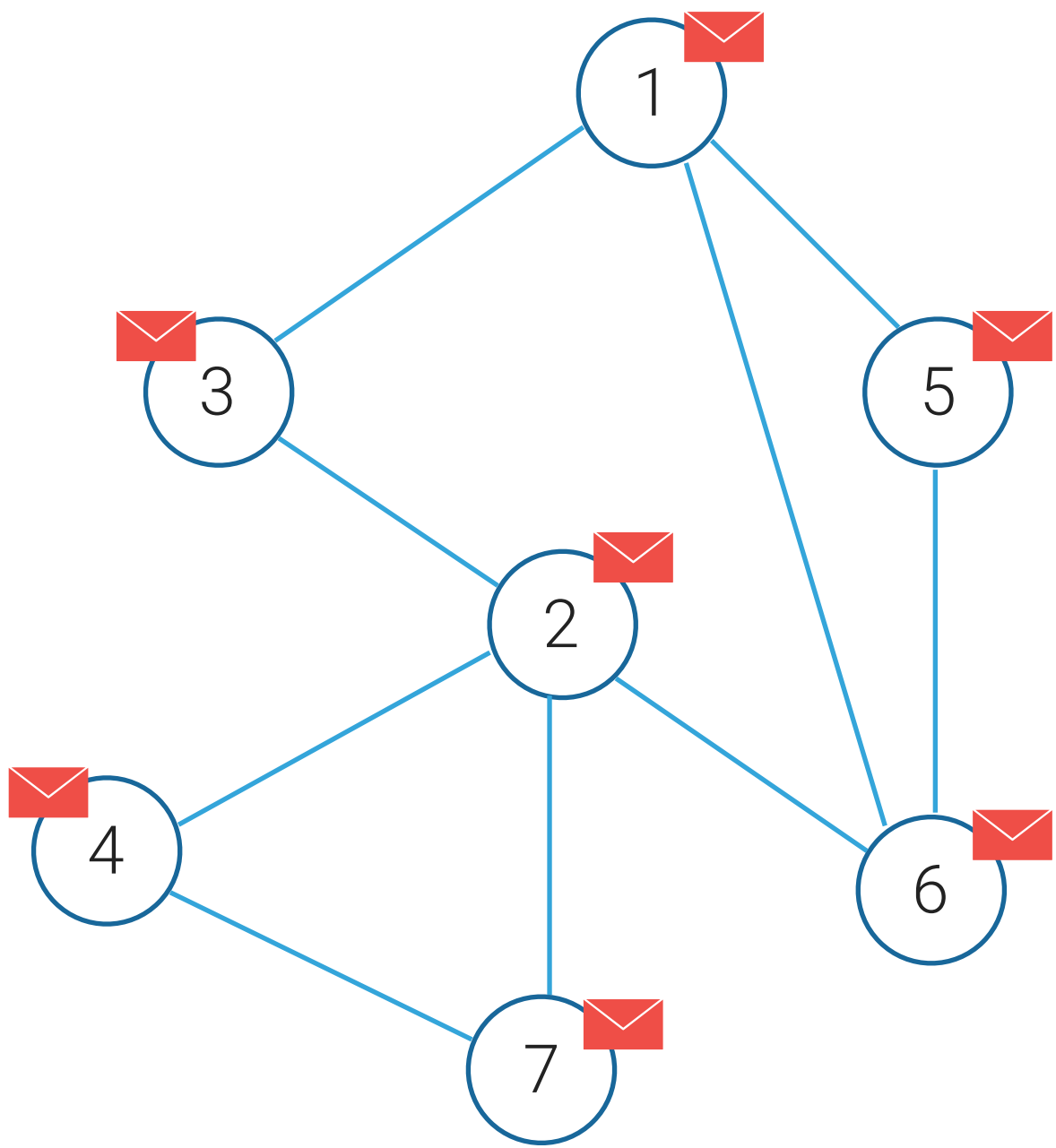
- ▶ Propondo *root* Y , a partir do nó X , anunciando distância d para Y
- ▶ Inicialmente cada *switch* se anuncia como *root*
 - ▶ ou seja, anuncia $(X, 0, X)$ para seus vizinhos
- ▶ *Switches* atualizam seus dados ao receber as mensagens
 - ▶ Se recebe um ID menor do que o do seu atual *root*, atualiza e define quem enviou como o próximo *hop*
- ▶ *Switches* computam a distância até o *root*
 - ▶ Adicionando 1 a distância recebida pelo vizinho
- ▶ Se o *root* mudou ou se o menor caminho para o *root* mudou:
 - ▶ Envia mensagem de atualização para os vizinhos: $(Y, d+1, X)$

ITERAÇÃO 1



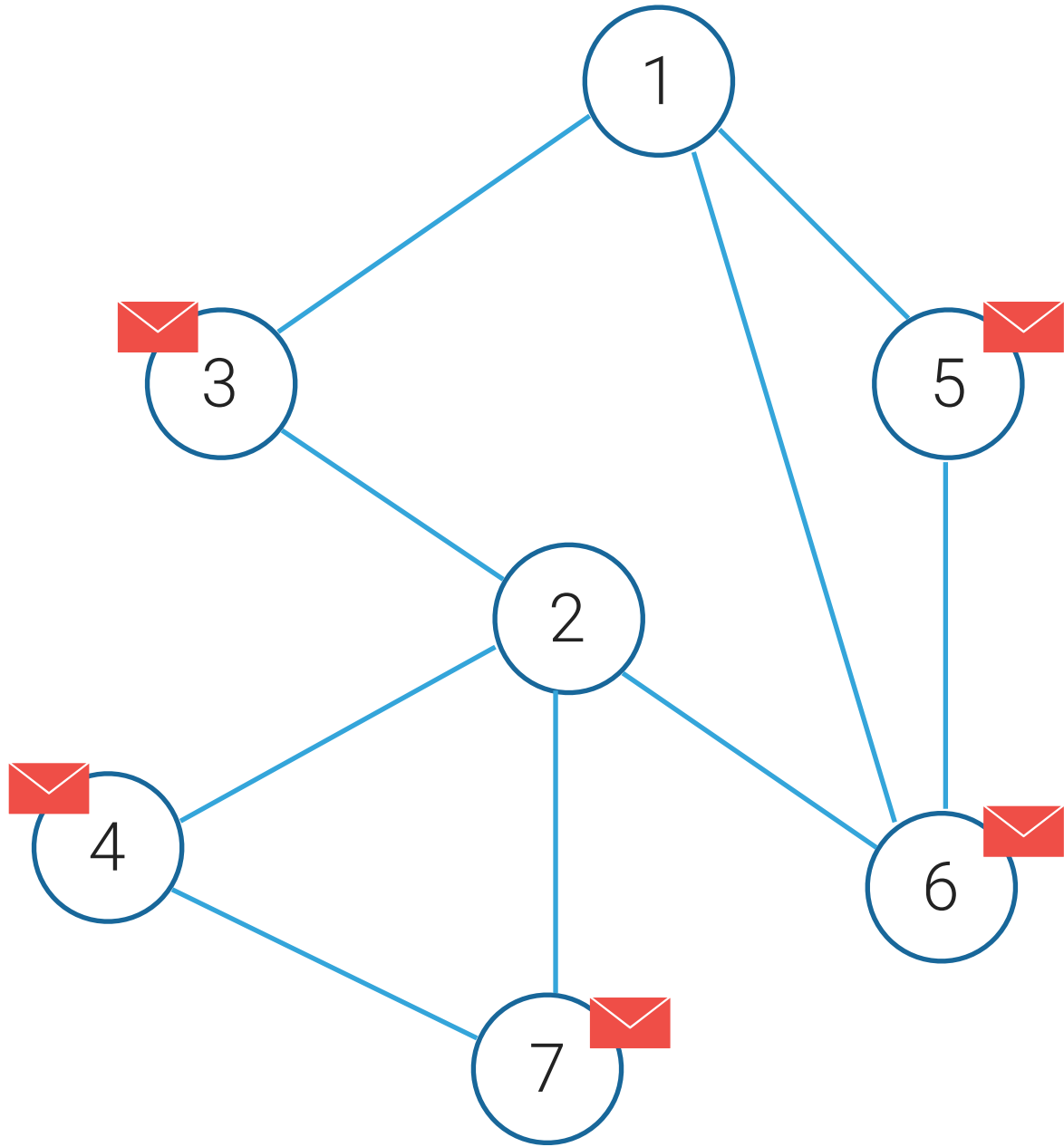
	RECEBE	ENVIA	PRÓXIMO HOP
1		(1, 0, 1)	1
2		(2, 0, 2)	2
3		(3, 0, 3)	3
4		(4, 0, 4)	4
5		(5, 0, 5)	5
6		(6, 0, 6)	6
7		(7, 0, 7)	7

ITERAÇÃO 2



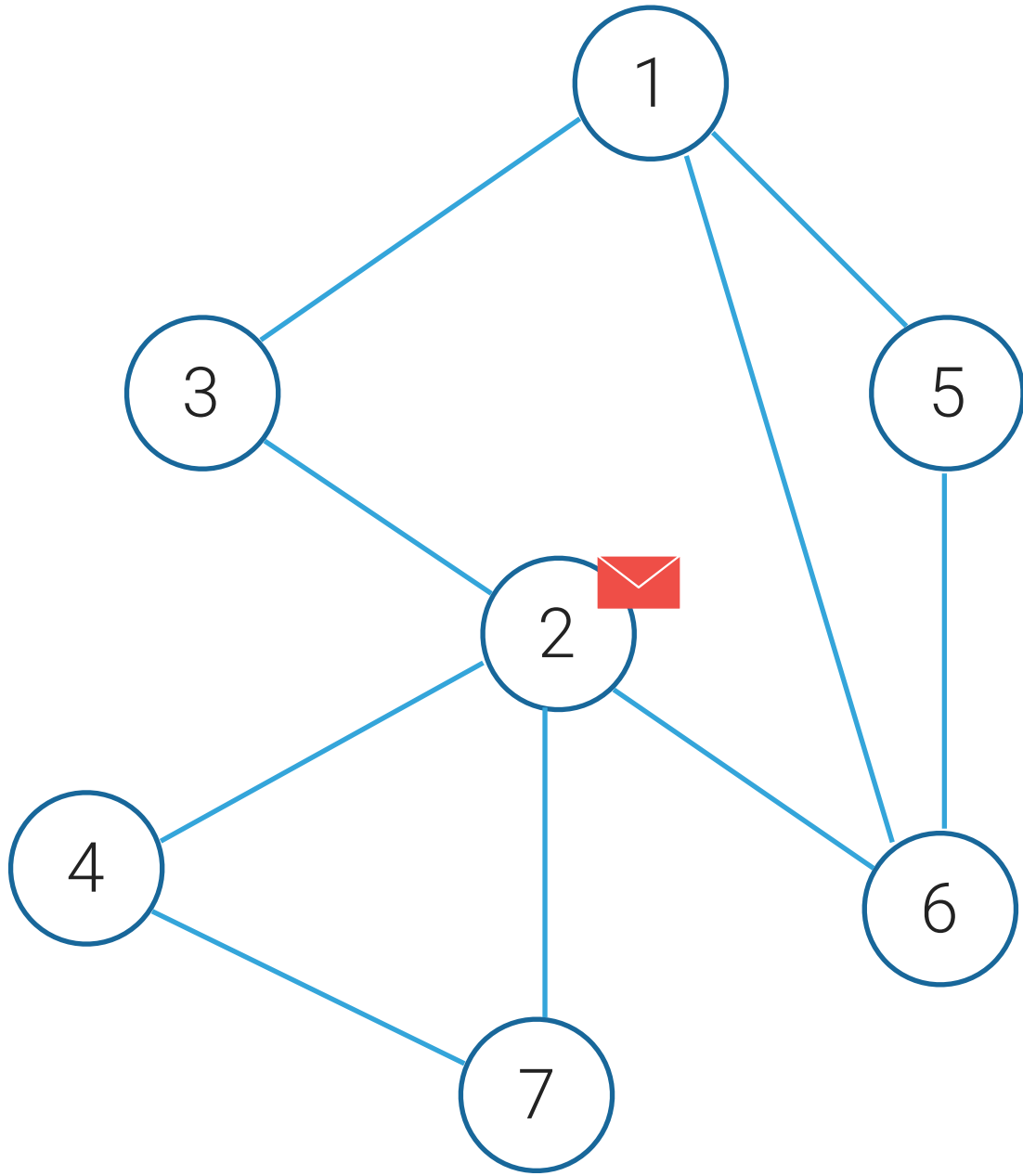
	RECEBE	ENVIA	PRÓXIMO HOP
1 (1, 0, 1)	(3, 0, 3), (5, 0, 5), (6, 0, 6)		1
2 (2, 0, 2)	(3, 0, 3), (4, 0, 4), (6, 0, 6), (7, 0, 7)		2
3 (3, 0, 3)	(1, 0, 1), (2, 0, 2)	(1, 1, 3)	1
4 (4, 0, 4)	(2, 0, 2), (7, 0, 7)	(2, 1, 4)	2
5 (5, 0, 5)	(1, 0, 1), (6, 0, 6)	(1, 1, 5)	1
6 (6, 0, 6)	(1, 0, 1), (2, 0, 2), (5, 0, 5)	(1, 1, 6)	1
7 (7, 0, 7)	(2, 0, 2), (4, 0, 4)	(2, 1, 7)	2

ITERAÇÃO 3



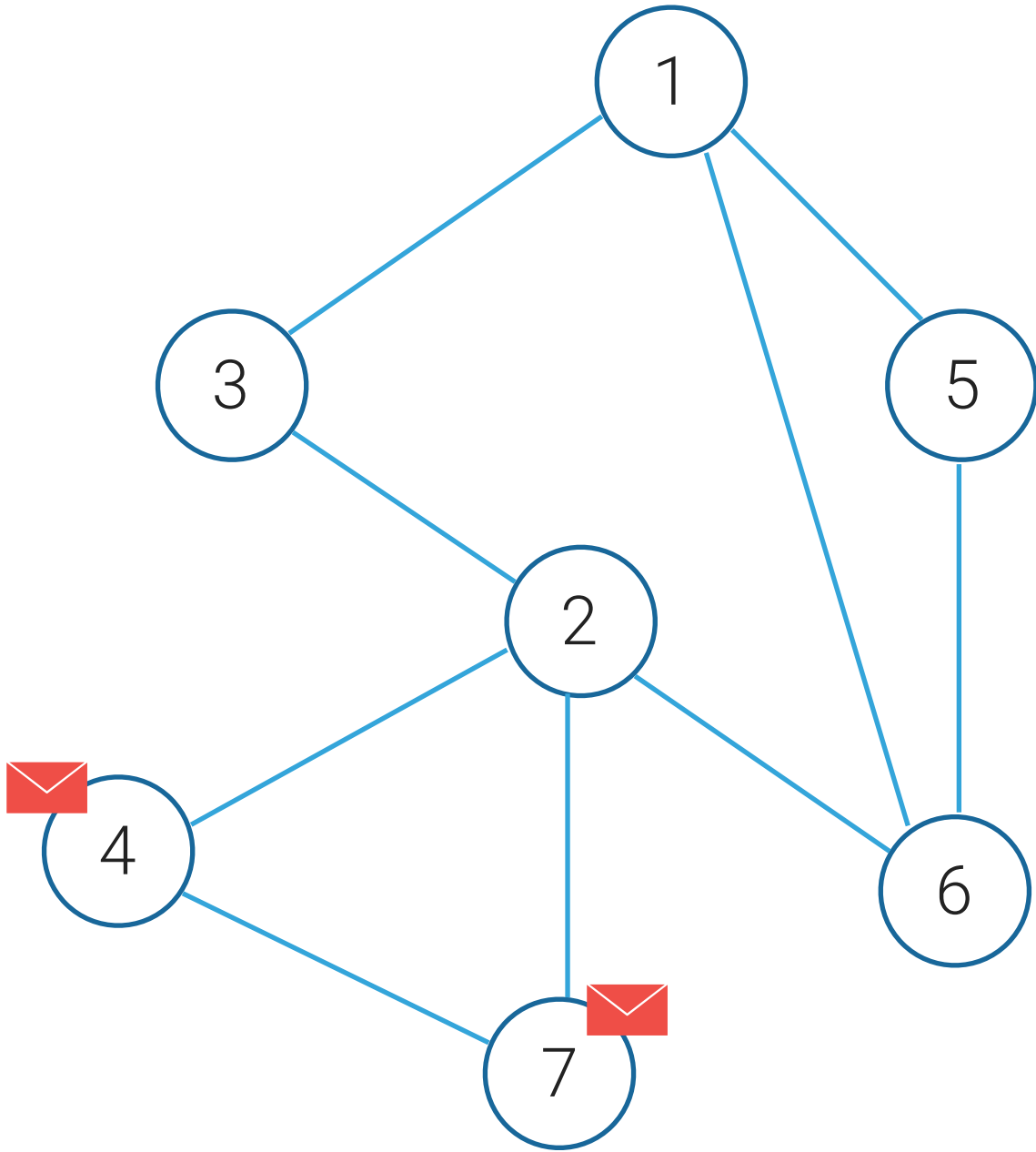
	RECEBE	ENVIA	PRÓXIMO HOP
1 (1, 0, 1)	(1, 1, 3), (1, 1, 5), (1, 1, 6)		1
2 (2, 0, 2)	(1, 1, 3), (2, 1, 4), (1, 1, 6), (2, 1, 7)	(1, 2, 2)	3 (ou 6)
3 (1, 1, 3)			1
4 (2, 1, 4)	(2, 1, 7)		2
5 (1, 1, 5)	(1, 1, 6)		1
6 (1, 1, 6)	(1, 1, 5)		1
7 (2, 1, 7)	(2, 1, 4)		2

ITERAÇÃO 4



	RECEBE	ENVIA	PRÓXIMO HOP
1 (1, 0, 1)			1
2 (1, 2, 2)			3
3 (1, 1, 3)	(1, 2, 2)		1
4 (2, 1, 4)	(1, 2, 2)	(1, 3, 4)	2
5 (1, 1, 5)			1
6 (1, 1, 6)	(1, 2, 2)		1
7 (2, 1, 7)	(1, 2, 2)	(1, 3, 7)	2

ITERAÇÃO 5



	RECEBE	ENVIA	PRÓXIMO HOP
1 (1, 0, 1)			1
2 (1, 2, 2)	(1, 3, 4), (1, 3, 7)		3
3 (1, 1, 3)			1
4 (1, 3, 4)	(1, 3, 7)		2
5 (1, 1, 5)			1
6 (1, 1, 6)			1
7 (1, 3, 7)	(1, 3, 4)		2

APÓS 5 ITERAÇÕES...

- ▶ 3-1
- ▶ 5-1
- ▶ 6-1
- ▶ 2-3
- ▶ 4-2
- ▶ 7-2

