

Protokol OSPF

Protokol OSPF (Open Shortest Path First) patří mezi nejpoužívanější směrovací protokoly. Vychází z algoritmu SPF (Shortest Path First) vytvořeného holandským vědcem Dijkstrou a řadí se mezi algoritmy označované jako Link State. Tučně uvedený text specifikuje klíčové fáze protokolu OSPF, globálním cílem, společným pro všechny směrovací protokoly, je samozřejmě sestavení směrovací tabulky pro každý směrovač v síti. Ta obsahuje údaje, umožňující posílat data libovolnému směrovači v rámci dané sítě (resp. síti či podsíti) a to) optimální trasou. Pojem optimální ovšem v praxi představuje kombinaci několika dílčích kritérií (aj. přenosová rychlost, doba odezvy, chybovost aj.), kterým jsou přisouzeny různé váha. Pro zjednodušení je v popisovaném příkladu použito kritérium jediné a to administrativní údaj nazvaný cena cesty (případně vzdálenost).

Je vhodné připomenout existenci jiných přístupů, například protokolů využívajících algoritmus tzv. distančního vektoru, často užívaných v menších sítích (směrovací protokoly RIP, RIP2, IGRP aj.). Algoritmus distančního vektoru je charakterizován tím, že směrovač rozhlašuje svoje ceny cest ke všem dalším sítím všem sousedním směrovačům. Každý z nich pak přijaté údaje doplní o vlastní hodnoty (o dosud neuvedené sítě, zvýší cenu cesty ke sítím známým) a nejlepší ceny cest rozešle zbylým sousedům, přičemž jako autora všech údajů uvede sebe sama (ve skutečnosti se ještě aplikují opatření zabráňující vznikům smyček či opakované výměně dat).

Každý směrovač identifikuje své sousedy, zjišťuje cenu cesty ke každému z nich a port, prostřednictvím k nim může přistupovat

K tomuto účelu mohou posloužit pakety *Hello*, jejichž výměnou se směrovače vzájemně představují se svými sousedy, případně jsou vztahy se sousedy konfigurovány manuálně.

Každý směrovač sestaví svůj Link State Advertisement (LSA) a ten záplavou distribuuje všem dalším směrovačům v celé síti

Směrovač rozhlašuje cenu cesty ke každému ze svých sousedních směrovačů a ty dále všem dalším směrovačům v síti (záplava – vysílá se do všech rozhraní, odkud dosud daný paket nepřišel). Směrovač bude nové LSA vysílat periodicky. Dále bude směrovač rozhlašovat LSA v případě, že objeví nového souseda, jestliže se změní cena cesty k některému ze sousedů nebo pokud některý ze sousedů přestane být dostupným; v tom případě je záplava omezená.

Každý směrovač přijímá LSA od všech ostatních směrovačů

Musí být garantovány dvě věci a to doručení LSA od každého směrovače všem dalším směrovačům a zjištění, zda je či není konkrétní LSA tím nejaktuálnějším odeslaným tím či oním směrovačem. Každý směrovač v síti musí mít k dispozici kompletní a identickou množinu (databázi) LSA od všech dalších směrovačů v síti. Jestliže by množiny LSA nebyly identické, směrovače by mohly vypočítat nekonzistentní trasy. Pak by mohly vzniknout kruhové cesty a naopak některé cíle by nemusely být dostupné. Následující popis tedy předpokládá, že každý směrovač včas přijal po jednom LSA od každého z dalších směrovačů a dále že je schopen odlišit nové LSA od starých.

Každý směrovač používá pro výpočet nejkratší cesty ke každému cíli sadu LSA

Příklad popisuje algoritmus Link State a detailně rozebírá postup, jehož pomocí směrovač R1 vytváří směrovací tabulku.

Každý směrovač identifikuje své sousedy a zjišťuje příslušné ceny cest

V příkladu R1 identifikuje jako sousedy R2 a R5 s cenami cest 7 a 2. Ceny jsou přiřazeny rozhraním a to v R1.

Každý směrovač konstruuje svůj LSA, který se rozesílá všem dalším směrovačům v celé síti, nikoliv jen sousedním směrovačům.

LSA vysílaný směrovačem R1 (viz tabulka 1) říká, že bezprostředně sousedí se dvěma směrovači a to s R2 při ceně cesty 7 a dále s R5 při ceně cesty 1. Záznam v LSA má pro každý sousední směrovač tvar:

(Identifikátor_sousedního_směrovače, cena_cesty_k_němu, číslo_portu_k_němu).

R1	
Soused	Cena cesty
R2	7
R5	1

Tabulka 1 LSA směrovače R1

Každý směrovač přijímá LSA pakety od všech ostatních směrovačů

Následující tabulky znázorňují LSA všech dalších směrovačů

R2	
Soused	Cena cesty
R1	1
R3	1
R4	3
R6	1

R5	
Soused	Cena cesty
R1	3
R4	1

R3	
Soused	Cena cesty
R2	2

R6	
Soused	Cena cesty
R2	1

R4	
Soused	Cena cesty
R2	2
R5	1

Každý směrovač používá pro výpočet nejkratší cesty ke každému cíli identickou množinu LSA

Pomocí algoritmu SPF je postupně budována směrovací tabulka. Při jejím výpočtu jsou používány dvě databáze, PATH pro vlastní směrovací tabulku a TEMP pro právě zpracovávané údaje. Zbývající údaje jsou připraveny ke zpracování.

Krok 1: Kořenem je R1, takže cena cesty k němu je 0.

PATH: (R1,0,0)

TEMP: (-)

Krok 2: Prozkoumání LSA od směrovače R1. Tento LSA obsahuje záznamy o dvou uzlech, R2 a R5; tyto záznamy jsou přidány do TEMP.

PATH: (R1,0,0)

TEMP: (R2,7,1)(R5,1,2)

Krok 3: Záznam (R5,1,2) obsahuje nejnižší cenu cesty ze všech záznamů v TEMP, je proto přesunut do PATH a proces pokračuje návratem ke kroku 2.

PATH: (R1,0,0)(R5,1,2)

TEMP: (R2,7,1)

Krok 2: Prozkoumání LSA od směrovače R5. Zde se vyskytnou dva záznamy, přičemž jeden z nich uvádí R1 s cenou 3. Ovšem algoritmus již zná cestu k R1 s cenou 0 (R1 je kořenem), a proto lze tento záznam v LSA od R5 zrušit. Další záznam v tomto LSA se týká cesty k R4. Vzhledem k tomu, že žádná cesta k R4 dosud nebyla známa, je tato přidána do TEMP s vypočtenou cenou 2. Ta je dána součtem ceny cesty od R1 k R5 (1, je již uvedena v PATH) a ceny cesty od R5 k R4 (rovněž 1, příslušná hodnota byla získána z LSA).

PATH: (R1,0,0)(R5,1,2)

TEMP: (R2,7,1)(R4,2,2)

Krok 3: Záznam (R4,2,2) obsahuje nejnižší cenu cesty ze všech záznamů v TEMP, je proto umístěn do PATH a proces pokračuje krokem 2.

PATH: (R1,0,0)(R5,1,2)(R4,2,2)

TEMP: (R2,7,1)

Krok 2: Je prozkoumán LSA od směrovače R4, který obsahuje dva záznamy. První popisuje cestu od něj k R5 s cenou 1. Již dříve vypočtená cena cesty od kořene k R4 je 2 (je uložena v databázi PATH), tudíž by celková cena cesty od kořene k R5 přes R4 byla 3. Protože již byla nalezena cesta od kořene k R5 s cenou 1, je nová cesta přes R4 vyřazena. Dalším záznamem v LSA je záznam pro R2, přičemž cena cesty z R4 k němu činí 2. Kombinací této hodnoty s cenou cesty od kořene k R4 (tj. 2, opět viz databáze PATH) je stanovena celková cena cesty od kořene k R2, tj. 4. Ta je nižší a tedy lepší než cena prozatím uložená v databázi TEMP (má hodnotu 7). Proto je tento horší záznam pro R2 (s cenou 7) v databázi TEMP zrušen a místo něj je přidán nový záznam pro cestu vedoucí k R2 přes R4.

PATH: (R1,0,0)(R5,1,2)(R4,2,2)

TEMP: (R2,4,2)

Krok 3: Do PATH je přesunut záznam z TEMP s nejnižší cenou a proces se vrací zpět ke kroku 2.

PATH: (R1,0,0)(R5,1,2)(R4,2,2)(R2,4,2)

TEMP: prázdná

Krok 2: Nyní je prozkoumán LSA od R2, mající čtyři záznamy. Údaj o cestě k R1 je zahozen (cena 1), protože již existuje lepší s cenou 0. Stejně je naloženo s údajem o cestě k R4 (cena 3), která není lepší než existující cesta k R4 (cena cesty 2). Do databáze TEMP jsou přidány zbývající dva záznamy a to o R3 a R6, což jsou nové uzly. Celková cena cesty od kořene k R3 je vypočtena jako součet ceny cesty k R2 (4, je již v PATH) a ceny cesty od R2 k R3 (záznam z LSA od R2); tato celková cena cesty činí 5. Podobným způsobem vypočtena cena cesty k R6, činí rovněž 5.

PATH: (R1,0,0)(R5,1,2)(R4,2,2)(R2,4,2)

TEMP: (R3,5,2)(R6,5,2)

Krok 3: Záznam v TEMP s nejnižší uvedenou cenou je přesunut do PATH. Protože takovou hodnotu mají záznamy dva, lze vybrat kterýkoliv z nich. Proces se vrací zpět ke kroku 2.

PATH: (R1,0,0)(R5,1,2)(R4,2,2)(R2,4,2)(R3,5,2)

TEMP: (R6,5,2)

Krok 2: Je prozkoumán LSA od R3, který obsahuje jediný záznam a to pro R2. Vypočtená celková cena cesty od kořene k R2 prostřednictvím R3 samozřejmě je horší než již známá cesta k R2 a proto není provedena žádná akce.

PATH: (R1,0,0)(R5,1,2)(R4,2,2)(R2,4,2)(R3,5,2)

TEMP: (R6,5,2)

Krok 3: Uvedený jediný záznam je z TEMP přesunut do PATH a proces se vrací zpět ke kroku 2.

PATH: (R1,0,0)(R5,1,2)(R4,2,2)(R2,4,2)(R3,5,2)(R6,5,2)

TEMP: prázdná

Krok 2: Je prozkoumán LSA od R6 obsahující pouze jediný záznam a to od R2. Opět je vypočtena celková cena cesty od kořene k R2 přes R6, která je nutně horší než již známá cesta k R2 a proto není provedena žádná akce.

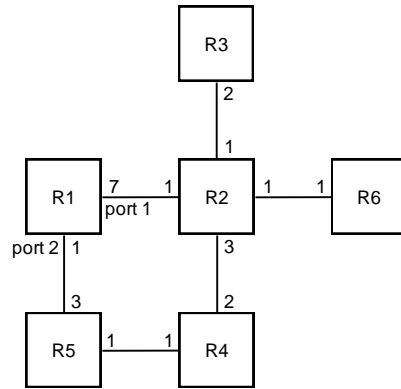
PATH: (R1,0,0)(R5,1,2)(R4,2,2)(R2,4,2)(R3,5,2)(R6,5,2)

TEMP: prázdná

Krok 3: Protože byly zpracovány LSA od všech směrovačů a ani v databázi TEMP se již nenalézá žádný záznam, je algoritmus u konce. Databáze PATH nyní obsahuje seznam nejkratších cest od směrovače R1 ke každému dalšímu směrovači v síti. Například druhý záznam v databázi PATH říká, že R5 je dosažitelný přes port 2 při ceně 1. Třetí záznam ukazuje, že k R4 vede cesta přes port 2 a cena je 2. Pomocí těchto informací lze snadno sestavit následující směrovací tabulku:

Cílový směrovač	Port	Následující skok
R2	2	R5
R3	2	R5
R4	2	R5
R5	2	-
R6	2	R5

Tabulka 2 Směrovací tabulka směrovače R1



Obr. 1 Schéma sítě pro demonstraci algoritmu Link State (pozor – toto schéma obecně není známo, i když by je bylo možno snadno odvodit; pracuje se se separátními údaji od každého ze směrovačů)