

OSPF 高效路由协议的设计与优化^{*}

王琦进 施 欢

(安徽新华学院)

【摘 要】针对大规模网络的路由特点,在分析 OSPF 动态路由协议基础上,提出 OSPF 高效路由协议的设计与优化方法.协议通过结合 SSH、MSTP、HSRP、DM-VPN 等技术,优化了路由的寻址与传输等性能.通过 IOU 仿真表明,优化后的路由在路由区域设定、路由路径走向、路由条目汇总、路由安全认证等方面都有较好的表现.

【关键词】OSPF; 路由协议; 网络优化; IOU

中图分类号: TP393.4 文献标识码: A 文章编号: 1000-5617(2016)05-0036-05

0 引言

随着网络规模的不断扩大,网络应用已遍布各个领域,路由作为网络技术的核心,一直是研究的热点. OSPF 路由协议作为主流有线路由协议有着广泛的市场应用,但使用 OSPF 路由协议时,如果不合理规划,可能会使网络不稳定,甚至不安全.因此在网络中如何正确部署 OSPF 协议,合理进行优化设计是非常重要的. OSPF 路由协议是经典有线路由协议,由 IETF 在 20 世纪 80 年代末期开发,最初由 RFC1131^[1] 文档定义,协议在稳定性和功能性方面有所不足.文档 RFC 1247 对 OSPF 协议的稳定性和功能性进行了改进,由于两个 OSPF 之间有实质性差别,所以文档 RFC1247^[2] 定义的 OSPF 称为 OSPFv2. 随着 OSPFv2 的不断演进,文档 RFC2328 又重新定义了 OSPFv2 的最新内容,但 OSPFv2 的最大不足之处是不支持 IPV6 环境. RFC2740^[3] 是 OSPFv3 的协议标准,协议在基本不改动 DR 选举、SPF 算法等前提下,修改了区域数据结构及部分报文内容,增加了 IPv4 向 IPv6 的过渡功能.

该文在 OSPFv2 协议基础上,以一个市区总

行到县支行的某银行网络为例,在路由表大小、路径走向、路由开销、网络安全等方面进行优化改进.通过仿真实验表明,优化后的 OSPF 路由传输效率更高、网络更稳定.

1 路由协议设计

1.1 OSPF 路由协议特性分析

路由协议根据自治系统区域特点分为内部网管和外部网关协议, RIP、OSPF 等是典型的内部网管路由协议. RIP 作为最早的内部网关路由协议,其特点在于 Metric 的大小由 Hop Count 计算,最大支持 16 跳,且路由表通过定时的方式更新全部路由信息,因此 RIP 协议不适用于大型网络. OSPF^[4] 是分层的链路状态路由协议,协议不受跳数的限制且通过触发方式更新有变化的路由表信息,因此 OSPF 协议能够更快速地响应网络拓扑情况的变化.另外该协议采用 Dijkstra 算法,支持无类 IP、VLSM、CIDR、手工汇总、验证等,计算出的路由信息最佳,因此 OSPF 路由协议更适用于大型网络.

1.2 网络环境搭建

该文实验使用 IOU 仿真软件,该软件是一款

收稿日期: 2016-04-09

^{*} 安徽省高校优秀青年人才支持计划重点项目(gxyqZD2016391); 安徽省高等学校自然科学研究重点项目(KJ2016A311)

基于 Linux 操作系统且具有图形化界面的网络仿真软件,能够对大型网络进行仿真,该软件在

实际仿真时设备的 CPU 和内存占用较小,该仿真拓扑图如图 1 所示。

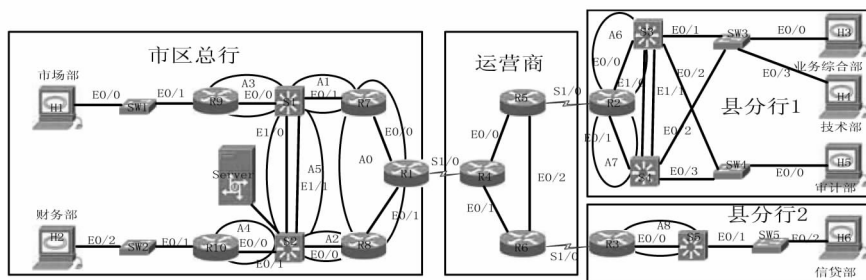


图1 网络拓扑图

其中, H1 - H6 代表各部门, R1 是总行的出网路由器, R7、R8 是内网的核心路由器, S1、S2 是总部的汇聚层交换, Server 是 web 服务器, SW1、SW2 是接入层交换, 主要连接用户, 划分 VLAN。其余两个分支设计与总部类似, 在此不做详细说明。考虑到设备性能, 仿真环境里的设备种类与数量未做过多扩展。

1.3 关键技术指标

(1) SSH(Secure Shell) 技术: SSH 为建立在应用层和传输层基础上, 专为远程登录会话和其他网络服务提供安全性的协议。利用 SSH 协议可以有效防止远程管理过程中的信息泄露问题。该方案三层设备采用基于密钥的安全验证方式控制用户登录设备, 这不仅可以对所有通信数据进行加密, 同时还可以成功避免“中间人”攻击。

(2) MSTP(Multiple Spanning Tree Protocol) 技术: MSTP^[5] 是基于 SDH 的多业务传送平台, 可同时实现 TDM、ATM、以太网等业务的接入、处理和传送, 提供统一网管的多业务节点。该方案针对银行内网划分 VLAN 过多, 产生大量生成树占用网络资源的问题, 通过 MSTP 技术, 将多个属性一样的 VLAN 虚拟成一个生成树, 以实现负载均衡。

(3) HSRP(Hot Standby Routing Protocol) 技术: HSRP^[6] 为热备份路由协议, 当使用的路由器或者三层交换机出现问题, 导致内部用户无法与其他用户正常通信时, 可以通过后补设备实现平滑的替换, HSRP 通常需要结合 Track 技术使用。该方案在 S3、S4 设备上通过设置虚拟地址实现热备。

(4) ACL(Access Control List) 技术: ACL^[7] 是保障网络系统安全的重要技术, 通过编写访问规则, 防止非法设备破坏系统安全, 获取系统数

据。该方案在财务部等部门使用 ACL, 通过控制端口进出数据包, 对流量进行控制, 以保证网络资源按照需求合理使用。

(5) DMVPN(Dynamic Multipoint VPN) 技术: 该方案中, 银行总部下有多个支行, 采用传统 IP-Sec VPN 需要编写多条静态路由, 且该方式不支持动态分支站点, 因此不利于网络管理。DMVPN^[8] 可解决 IP-Sec VPN 的问题, DMVPN 支持动态路由协议, 没有接口水平分割且不需要关闭下一跳便可以实现 GRE 流量互通。因此 DMVPN 技术与 OSPF 协议的结合更适合该实验网络的构建。

2 OSPF 路由技术对网络的优化

OSPF 协议特性复杂, 在构建网络时需要根据企业的需求合理设计。此次实验在确保构建网络安全稳定的基础上, 根据 OSPF 协议特性对网络进行优化, 并通过 IOU 仿真软件对优化前与优化后的网络进行对比分析。

(1) 未与骨干区域相连的普通区域路由表优化: 实验中首先启用 OSPF 进程, 对整个网络进行骨干区域和普通区域的划分, 以减少 LSA 网络泛洪的产生。图 1 中, 启用区域划分后, OSPF 区域间传递的 LSA 遵循一定原则, 即普通区域的路由不可以直接传递到另一个普通区域。当划分完成后, 没有和骨干区域相连的区域学不到其他区域的路由条目, 此时在区域间通过隧道方式, 可使该区域与骨干区域相连接。实验在 Area 1 上建立 tunnel 口, 并将该网段划分到 Area 3 中, 对于其他由于没有与骨干区域相连接的区域则采用虚链路技术。图 2 所示的是以路由器 R9 为例, 建立 tunnel 口前后所学路由的对比。

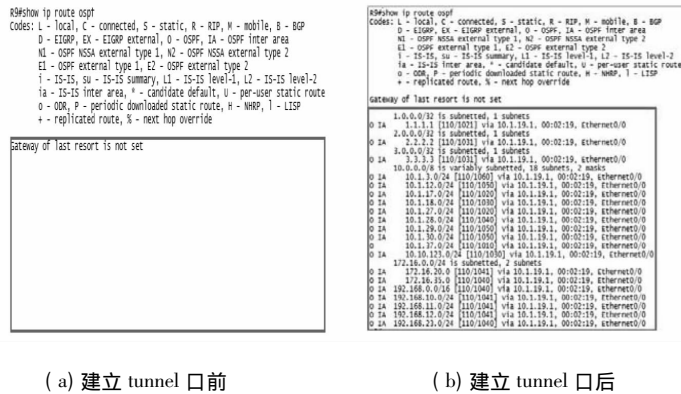


图 2 建立 tunnel 口前后 R9 路由表的对比

通过查看 R9 路由表可以看出,建立 tunnel 后 R9 的路由表学到了全网的路由条目。

(2) 特殊区域对路由表的优化: 实际网络环境中,某些区域的网络设备不需要接收 3 类以上的 LSA。对于该实验的网络拓扑规模,如果不进

行优化,会导致部分区域设备接收到很多无用的 LSA,占用设备资源。实验通过将部分区域设置成完全末梢区域,以解决无用 LSA 问题。图 3 所示,在 Area 3 区域中对路由器 R9 划分为完全 stub 区域,通过仿真对比路由划分前后的情况。

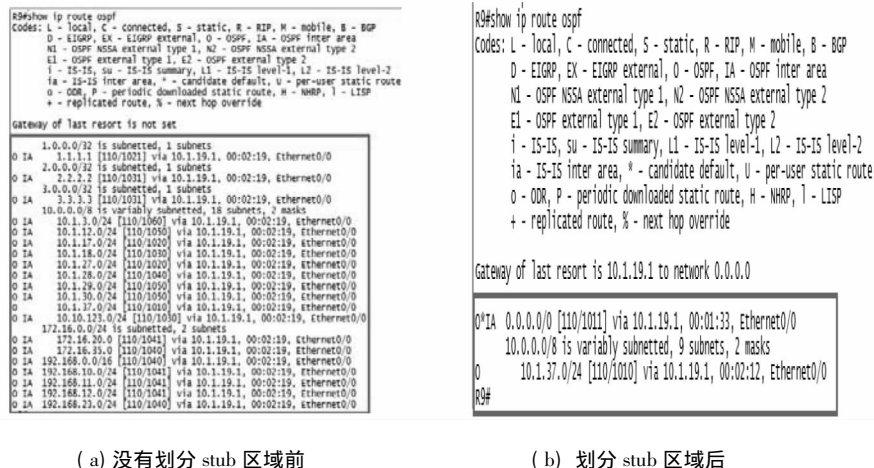


图 3 划分完全 stub 区域前后 R9 路由表的对比

通过对比可以看出,R9 路由表在划分完全 stub 区域后,路由表只有一条 3 类默认路由和一条通过 S1 学习到 tunnel 口路由。与没有划分前相比,路由条目数大大减少,降低了 R9 路由器内存和 CPU 的占用率。

(3) 合理的区域设置对路径走向的优化: 利

用 IOU 检测 H2 访问 H1 路径走向时,发现路径走向并不是最优的,H2 去往 H1 需经过路由器 R1,而不是直接经过 S1,这样会浪费通信时间及带宽。实验通过在 S1 和 S2 之间建立虚链路来改变路径走向,路径改变前后的仿真结果如图 4 所示。

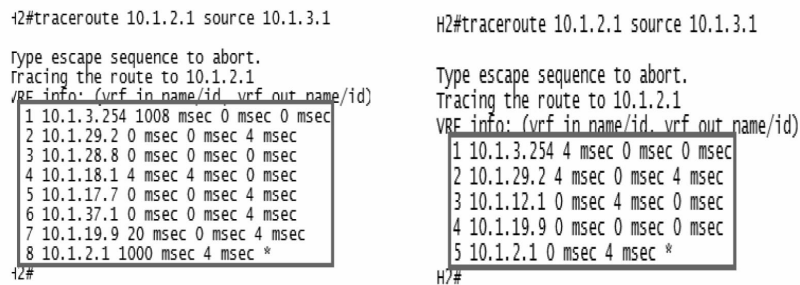


图 4 路径走向优化前后对比

仿真表明,在 S1 和 S2 之间没有建立虚链路时, H2 访问 H1 中间经过了 7 台设备,在 S1 和 S2 之间建立虚链路后, H2 访问 H1 中间只经过了 4 台设备. 路径的优化减少了路由转发的数目,减少了路由转发时间.

(4) 路由汇总对路由表的优化: 网络中骨干路由器的路由表数目非常庞大,其性能要求也非常高,为了降低对该类路由器的内存占用,可将某些区域的路由条目在区域边界路由器上进行汇总. 汇总后的路由表相比于之前的路由表规模变小,同时也使得整个网络更容易管理以方便进一步优化. 图 5 所示的是对 R1 路由表在 Area3

区域汇总前后的对比.

仿真结果显示,路由器 R1 在 Area3 区域汇总后其路由表的路由条目相对于汇总前减少了 5 条.

(5) 更改 Cost 值对路由条目的优化: 实验在 R7 和 S1 之间建立了 tunnel 口,通过查看 tunnel 口特性,该口带宽只有 100kbps,这样计算出的接口 Cost 值会很大,不符合实际网络构建的需求. 实验通过更改路由 Cost 值使整个网络能够高效的运用网络带宽. 图 6 所示,以 R8 路由器为例,更改 Cost 值前后的对比.

```
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
2.0.0.0/32 is subnetted, 1 subnets
O IA 2.2.2.2 [110/1001] via 10.10.123.2, 00:08:36, Tunnel0
O IA 3.0.0.0/32 is subnetted, 1 subnets
O IA 3.3.3.3 [110/1001] via 10.10.123.3, 00:08:36, Tunnel0
10.0.0.0/8 is variably subnetted, 17 subnets, 2 masks
D IA 10.1.2.0/24 [110/30] via 10.1.18.8, 01:08:00, Ethernet0/1
D IA 10.1.3.0/24 [110/40] via 10.1.18.8, 01:08:00, Ethernet0/1
D IA 10.1.4.0/24 [110/30] via 10.1.18.8, 00:09:21, Ethernet0/1
D IA 10.1.6.0/24 [110/50] via 10.1.18.8, 00:09:21, Ethernet0/1
O IA 10.1.12.0/24 [110/30] via 10.1.18.8, 01:08:00, Ethernet0/1
O IA 10.1.19.0/24 [110/40] via 10.1.18.8, 00:09:21, Ethernet0/1
O IA 10.1.27.0/24 [110/20] via 10.1.17.7, 01:08:05, Ethernet0/1
O IA 10.1.28.0/24 [110/20] via 10.1.18.8, 01:08:10, Ethernet0/1
O IA 10.1.29.0/24 [110/30] via 10.1.18.8, 01:08:00, Ethernet0/1
O IA 10.1.30.0/24 [110/30] via 10.1.18.8, 01:08:00, Ethernet0/1
O IA 10.1.37.0/24 [110/1010] via 10.1.17.7, 00:38:08, Ethernet0/0
172.16.0.0/24 is subnetted, 2 subnets
O IA 172.16.20.0 [110/1011] via 10.10.123.3, 00:08:36, Tunnel0
O IA 172.16.35.0 [110/1010] via 10.10.123.3, 00:08:36, Tunnel0
D IA 192.168.10.0/24 [110/1011] via 10.10.123.2, 00:08:36, Tunnel0
D IA 192.168.11.0/24 [110/1011] via 10.10.123.2, 00:08:36, Tunnel0
D IA 192.168.12.0/24 [110/1011] via 10.10.123.2, 00:08:36, Tunnel0
D IA 192.168.23.0/24 [110/1010] via 10.10.123.2, 00:08:36, Tunnel0
D IA 192.168.24.0/24 [110/1010] via 10.10.123.2, 00:00:34, Tunnel0
```

(a) Area 3 汇总前

```
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
2.0.0.0/32 is subnetted, 1 subnets
O IA 2.2.2.2 [110/1001] via 10.10.123.2, 00:46:18, Tunnel0
O IA 3.0.0.0/32 is subnetted, 1 subnets
O IA 3.3.3.3 [110/1001] via 10.10.123.3, 00:46:18, Tunnel0
10.0.0.0/8 is variably subnetted, 13 subnets, 3 masks
D IA 10.1.0.0/16 [110/20] via 10.1.17.7, 00:17:30, Ethernet0/0
D IA 10.1.3.0/24 [110/40] via 10.1.18.8, 01:45:42, Ethernet0/1
D IA 10.1.12.0/24 [110/30] via 10.1.18.8, 01:45:42, Ethernet0/1
D IA 10.1.27.0/24 [110/20] via 10.1.17.7, 01:45:47, Ethernet0/1
D IA 10.1.28.0/24 [110/20] via 10.1.18.8, 01:45:52, Ethernet0/1
D IA 10.1.29.0/24 [110/30] via 10.1.18.8, 01:45:42, Ethernet0/1
D IA 10.1.30.0/24 [110/30] via 10.1.18.8, 01:45:42, Ethernet0/1
172.16.0.0/24 is subnetted, 2 subnets
O IA 172.16.20.0 [110/1011] via 10.10.123.3, 00:46:18, Tunnel0
O IA 172.16.35.0 [110/1010] via 10.10.123.3, 00:46:18, Tunnel0
D IA 192.168.0.0/16 [110/1010] via 10.10.123.2, 00:00:15, Tunnel0
O IA 192.168.23.0/24 [110/1010] via 10.10.123.2, 00:26:51, Tunnel0
```

(b) Area 3 汇总后

图 5 Area 3 汇总前后 R1 路由表的对比

```
Gateway of last resort is not set
1.0.0.0/32 is subnetted, 1 subnets
O 1.1.1.1 [110/21] via 10.1.18.1, 01:51:49, Ethernet0/1
2.0.0.0/32 is subnetted, 1 subnets
O IA 2.2.2.2 [110/1011] via 10.1.18.1, 00:52:14, Ethernet0/1
3.0.0.0/32 is subnetted, 1 subnets
O IA 3.3.3.3 [110/1011] via 10.1.18.1, 00:52:14, Ethernet0/1
10.0.0.0/8 is variably subnetted, 12 subnets, 3 masks
D IA 10.1.0.0/16 [110/30] via 10.1.28.2, 00:40:18, Ethernet0/0
D IA 10.1.3.0/24 [110/30] via 10.1.18.1, 00:23:27, Ethernet0/1
D IA 10.1.12.0/24 [110/20] via 10.1.28.2, 01:51:44, Ethernet0/0
D IA 10.1.19.0/24 [110/20] via 10.1.28.2, 01:51:44, Ethernet0/0
D IA 10.1.27.0/24 [110/20] via 10.1.18.1, 01:51:49, Ethernet0/1
D IA 10.1.28.0/24 [110/30] via 10.1.28.2, 01:02:14, Ethernet0/0
D IA 10.1.29.0/24 [110/30] via 10.1.18.1, 01:51:39, Ethernet0/1
D IA 10.1.30.0/24 [110/20] via 10.1.28.2, 01:51:44, Ethernet0/0
D IA 10.1.37.0/24 [110/20] via 10.1.28.2, 01:51:44, Ethernet0/0
O IA 10.10.123.0/24 [110/1010] via 10.1.18.1, 00:52:14, Ethernet0/1
172.16.0.0/24 is subnetted, 2 subnets
D IA 172.16.20.0 [110/1011] via 10.1.18.1, 00:52:14, Ethernet0/1
D IA 172.16.35.0 [110/1020] via 10.1.18.1, 00:52:14, Ethernet0/1
D IA 192.168.0.0/16 [110/1020] via 10.1.18.1, 00:06:12, Ethernet0/1
D IA 192.168.23.0/24 [110/1020] via 10.1.18.1, 00:32:49, Ethernet0/1
```

(a) 更改 Cost 值前

```
Gateway of last resort is not set
1.0.0.0/32 is subnetted, 1 subnets
O 1.1.1.1 [110/11] via 10.1.18.1, 01:54:24, Ethernet0/1
2.0.0.0/32 is subnetted, 1 subnets
O IA 2.2.2.2 [110/21] via 10.1.18.1, 00:00:30, Ethernet0/1
3.0.0.0/32 is subnetted, 1 subnets
O IA 3.3.3.3 [110/21] via 10.1.18.1, 00:00:30, Ethernet0/1
10.0.0.0/8 is variably subnetted, 12 subnets, 3 masks
D IA 10.1.0.0/16 [110/30] via 10.1.28.2, 00:42:53, Ethernet0/0
D IA 10.1.3.0/24 [110/30] via 10.1.18.1, 00:26:02, Ethernet0/1
D IA 10.1.12.0/24 [110/20] via 10.1.28.2, 01:54:19, Ethernet0/0
D IA 10.1.19.0/24 [110/20] via 10.1.28.2, 01:54:19, Ethernet0/0
D IA 10.1.27.0/24 [110/20] via 10.1.18.1, 01:54:24, Ethernet0/1
D IA 10.1.28.0/24 [110/30] via 10.1.28.2, 01:04:49, Ethernet0/0
D IA 10.1.29.0/24 [110/30] via 10.1.18.1, 01:54:14, Ethernet0/1
D IA 10.1.30.0/24 [110/20] via 10.1.28.2, 01:54:19, Ethernet0/0
O IA 10.10.123.0/24 [110/20] via 10.1.18.1, 00:00:36, Ethernet0/1
172.16.0.0/24 is subnetted, 2 subnets
D IA 172.16.20.0 [110/31] via 10.1.18.1, 00:00:30, Ethernet0/1
D IA 172.16.35.0 [110/30] via 10.1.18.1, 00:00:30, Ethernet0/1
D IA 192.168.0.0/16 [110/30] via 10.1.18.1, 00:00:30, Ethernet0/1
D IA 192.168.23.0/24 [110/30] via 10.1.18.1, 00:00:30, Ethernet0/1
```

(b) 更改 Cost 值后

图 6 更改 Cost 值前后 R8 路由表的对比

仿真结果显示,没有修改 Cost 值前, R8 学习两个分部的路由条目以及 S1 和 R7 之间的 tunnel 路由, Cost 值很大(如划线处所示),将 tunnel 口的带宽更改为 10000kbps 后, R8 学习这些路由条目的 Cost 值降低,通过修改 Cost 值,避免了路由条目的学习延迟过大.

(6) MD5 认证对路由安全的优化: 为了确保零区域安全,实验在该区域上设置了 MD5 认证,在设置认证后,需对之前建立的虚链路开启认证,否则无法传输数据. 由于路由过程加入了安全认证,因此数据传输的安全性得到了进一步保证. 图 7 所示,在 R1 路由器上做零区域验证之

后,其邻居路由器的变化情况.

```
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri  State  Dead Time  Address  Interface
2.2.2.2      0  FULL/DR  00:00:19   10.1.18.8  Ethernet0/1
3.3.3.3      0  FULL/DR  00:00:19   10.10.123.3  OSPF_VL1
10.1.12.0/24 0  FULL/DR  00:00:19   10.1.12.2  Tunnel0
10.1.12.3.0/24 0  FULL/DR  00:00:19   10.1.12.3  Tunnel0
R1(config-router)#do sh ip os net
Neighbor ID  Pri
```

会 Up ,在路由传输中增加 MD5 认证 ,增强了网络数据传输的安全性 ,确保了网络邻居节点数据传输安全.

3 结束语

OSPF 是一种广泛使用的动态路由协议 ,协议的使用如不能合理的规划与设计 ,会出现路由路径走向杂乱 ,网络不稳定等现象. 该论文提出的 OSPF 高效路由优化方法对网络的路由条目、路由路径、路由安全等方面进行优化改进 ,通过仿真实验可以看出 ,优化后的网络路由传输效率更高、网络更加稳定安全 ,更能满足用户需求.

参 考 文 献

- [1] 康威. OSPF 路由协议安全性分析与研究[D]. 北京: 北京邮电大学 201 年.
- [2] 彭刚. OSPF 动态路由技术在大型企业内部信息网络中的应用[J]. 信息通信 2013(5) : 98 - 98.
- [3] 潘楠 ,王勇 ,陶晓玲. 基于 OSPF 协议的网络拓扑发现算法[J]. 计算机工程与设计 2011 32(5) : 1550 - 1553.
- [4] 梁洪泉 ,吴巍. 利用节点可信度的安全链路状态路由协议[J]. 西安电子科技大学学报: 自然科学版 2016 43(5) : 137 - 142.
- [5] 朱壮普. 多生成树协议 MSTP 在交换网络中的应用[J]. 太原城市职业技术学院学报 2012(11) : 148 - 149.
- [6] 吴刚. HSRP 协议在 IP 网络设计中的研究与应用[J]. 绵阳师范学院学报 2011 30(8) : 83 - 89.
- [7] 杨梅 ,杨平利 ,宫殿庆. ACL 技术研究及应用[J]. 计算机技术与发展 ,2011 21(6) : 145 - 149.
- [8] 梁玉柱. 基于 DMVPN 技术的广域网设计和实现[J]. 信息系统工程 2012(2) : 118 - 120.

Design and Optimization of OSPF Routing Protocol

Wang Qijin ,Shi Huan

(Anhui Xinhua University)

Abstract: According to the routing characteristics of large scale network ,based on the analysis of OSPF dynamic routing protocol , the design and optimization method of OSPF high efficient routing protocol is proposed in this paper. This protocol is combined with SSH , MSTP , HSRP , DMVPN and other technologies ,and the routing addressing and transmission performance are optimized. Using the IOU to simulate ,the simulation results show that the optimized routing has better performance in the aspects of routing area setting , route path , the summary of the routing entries , and the authentication of routing security.

Keywords: OSPF; Routing Protocol; Network Optimization , IOU

(责任编辑: 季春阳)