Sequoia Routing Protocol

White Paper 2.0

treeqiu ▸ Tencent Network Lab ▸ 2013/2/6

2013 年 2 月 6 日开始撰写；
2013 年 2 月 7 日 15:37 发布第一版；
2013 年 2 月 7 日 21:55 修订；
2013 年 2 月 9 日 23:36 修订，添加后记部分图表 21；
2013 年 2 月 13 日 22:54 修订，添加 SRP 和 OSPF 层级对比、接收路由 Update 和发送路由 Update 的流程图；
2013 年 2 月 21 日 17:56 修订，添加 Grid 修改行、列，Border 对内对外路由方案说明；
2013 年 2 月 25 日 17:33 修订，添加参考资料；
2013 年 2 月 28 日 9:15 修订，修改页眉

Sequoia Routing Protocol

White Paper 2.0

在网络路由协议领域中，OSPF 无疑就是这个领域的名片。经过 20 多年的发展，OSPF 已经可以稳定地在任意网络拓扑中计算任意两点间的最短路径，甚至在网络拓扑发生变化时，也能迅速地重新计算最短路径，在数据中心网络领域，OSPF 的统治地位依然非常显著，那么在未来 IDC 网络中 OSPF 的统治地位是否依然可以延续？

1. 产生背景

路由协议的理论基础是“图论”中一个非常著名的命题——任意两点间的最短路径，对此与许多学者提出了很多算法，如 OSPF 所依据的 Dijkstra 算法，RIP/BGP 所依赖的 Bellman-Ford 算法等，这些算法有 2 个共同点：

1. 可以工作在任意结构的图中；
2. 假设图中有 N 个节点，对于任意 2 个节点间的最短路径计算复杂度开销为 $O(N^2)$。

IDC 网络体现了更多批量化建设和规模运营的思路，和传统网络相比也有两个非常明显的特点：

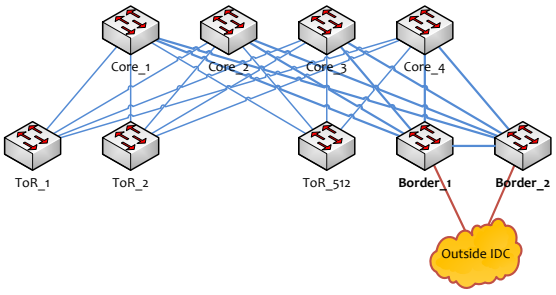
1. 网络拓扑是固定的、已知的，而且是简单的 CLOS 结构；
2. 为了降低单位 IDC 网络成本，网络节点 N 的数量是传统园区网络的 4 倍以上，甚至达到数 10 倍。

这就意味着不论是 Dijkstra 还是 Bellman Ford 在未来大规模 IDC 网络架构中并不适用，我们需要一种新的算法和路由协议。

2. 技术原理

在数据结构中对“树”和“图”对比，在“树”中寻找任意 2 点间的最短路径无疑要比“图”简单许多，那么 IDC 网络拓扑是 CLOS 类型，是否可以作为“树”型结构进行处理呢？

2.1. 理论依据——Multi-Root-Tree

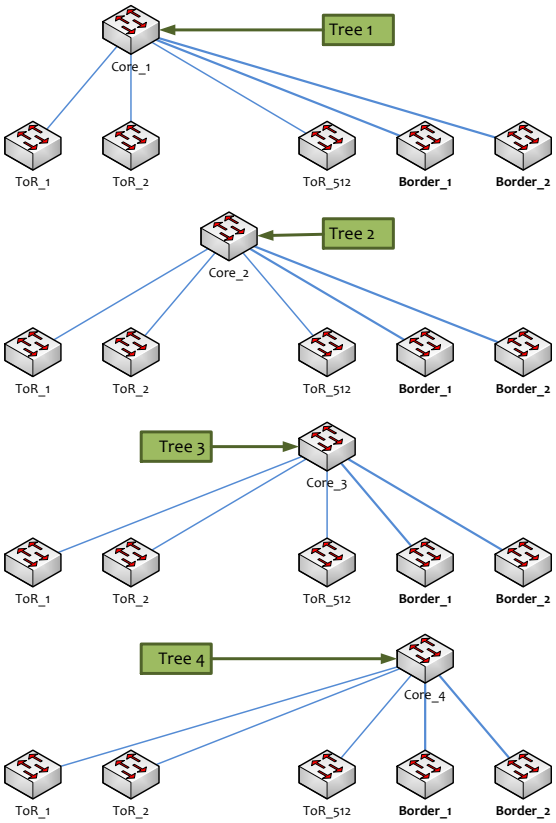


图表 1 数据中心网络拓扑

如图表 1 所示，这是一个典型 CLOS 状的 IDC 网络结构：

1. 图中有 512 个 ToR 用于连接大规模服务器，作为 CLOS 结构的 I/O Stage；
2. 有 2 至 4 个 Core 用于 ToR 之间互联，作为网络中的 CLOS Fabric；
3. 有 2 个 Border 用于访问 IDC 外部，和 ToR 一样，也是 I/O Stage。

这种结构中任意 2 台 ToR 或者 ToR 和 Border 之间的最短路径 Core 是必经之路，且 Core 的数量和最短路径 ECMP 数量一致。这种 CLOS 架构也被称为 Fat-Tree 模型，如果我们将 Fat Tree 模型看作多棵树的组合——称之为 Multi-Root-Tree，那么最短路径的计算和维护就变得非常简单。



图表 2 将 IDC 网络架构拆成 4 棵不同的树

如图表 2 所示，将 Fat-Tree 当成 Multi-Root-Tree 后，可以将 IDC 网络结构拆分成 4 棵独立的树：

1. 在每一棵树上计算最短路径非常简单，可以通过静态寻址方式计算任意 2 点间最短路径，复杂度为 0；
2. 将 4 棵树内计算的任意 2 点间最短路径进行叠加就是整个 IDC 最短路径计算结果；
3. ToR 作为多棵树的公共节点，需要隔离不同树的计算结果，不允许将从任意 Core 收到的流量再转发至任意 Core（这种方式也称之为“Vertical Split”——垂直分割），通过在 Core 上固定每个 IP 子网的可选 Next hop 方式实现。

Multi-Root-Tree 就是 Sequoia Routing Protocol 的理论依据。

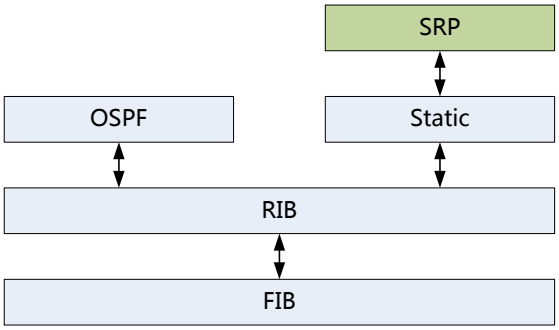
2.2. 实现分析

IDC 网络除了是已知的简单网络外，它还是一个基本固定的网络结构：

1. 网络规模相对固定，如设计 512 个 ToR 的规模；
2. 每个 ToR 有固定的 IP 子网用于连接服务器，并且 IP 子网根据 ToR 的编号呈等差递增关系，可以通过公式自动化计算，而且这些 IP 子网在运营过程中不会发生变化；
3. IDC 内 IP 子网可以汇聚成一条路由，IDC 内外访问均通过 Border，IDC 内网络类似于 OSPF 的 Total Stub 区域，由 Border 发布默认路由至 IDC 内。

这就意味着 IDC 网络内路由是相对固定的，变化的是每个子网的状态：

1. 某个 ToR 尚未开启或者关机，该 ToR 携带的 IP 子网处于 Down 状态；
2. 某个 ToR 已经开启，该 ToR 携带的 IP 子网处于 Up 状态；
3. 某个 ToR 至某个 Core 的连接中断，该 ToR 携带的 IP 子网在该 Core 为 Root 的树中处于 Down 状态。



图表 3 转发系统内 SRP 与 OSPF/BGP 等协议的关系

基于如上因素，如图表 3 所示，SRP 对 IDC 网络路由简化实现如下：

1. 每棵树的最短路径计算可以采用预设静态路由方式实现；

2. 对于预设静态路由范围之外的路由将会被 SRP 拒绝；

3. 设备之间运行 SRP 路由协议，建立并维护 Peer 关系；

4. ToR、Core、Border 各自监控本地路由表中预设 IP 子网或者外部路由状态，将预设 IP 子网、外部路由、SRP 静态路由状态通过 Peer 关系传递；

5. SRP 接收到 Peer 传递的消息或者 Peer 之间关系变化修改预设 SRP 静态路由状态。

2.3. 技术优势

SRP 的优势体现在：

1. 面向运营，路由协议运行预先规划的 IP 子网，实现规划到运营的闭环，而避免了 OSPF/BGP 有学习未规划 IP 子网的隐患，并且 SRP 可以对路由进行全方位的匹配——只查看失效的路由、只查看生效的路由、只查看 ToR_X 发布的 IP 子网、只查看业务 IP 子网等，这些都是在 OSPF 和 BGP 中无法实现或很难实现的；

2. SRP 在控制平面上动态地操作静态路由，能够快速地实现和方便地移植到不同交换机平台；

3. SRP 在每台设备上为每个 IP 子网固定了可选 Next hop，避免计算许多无效路由，而 OSPF 在局部网络故障时依然会计算大量无效路由，BGP 也会计算这些无效路由（再通过有效性检查对无效路由进行丢弃），SRP 是真正做到避免计算的路由协议；

4. SRP 保留了 OSPF/BGP 中的邻居概念，用于传递消息，SRP 的邻居关系维护和消息格式直接借鉴了 BGP 的部分实现原理和格式；

5. SRP 分工明确，并没有通过 SRP 来实现所有工作，稳态的 IDC 内 SRP 运行模式和 OSPF Total Stub 区/IS-IS 的 Level-1 区域，而 2 个 Border 则相当于 ABR 和 Level-1-2 路由器，对 IDC 内发布 SRP 默认路由（Border 可以同时监控 N 条外部路由，只有 N 条路由同时失效，SRP 默认路由才会失效），对 Outside IDC 重分布 IDC 内的汇总路由（并非通过 SRP 实现，而是人工配置一条 IDC 内的聚合静态路由，通过路由策略将这条聚合静态路由重分布至 Outside IDC）。

基于以上 5 点优势，SRP 更加贴近网络运营，精简并控制了路由计算，简化开发和实现，可以适应更大规模的网络，可以满足未来 IDC 网络的要求。

3. 协议介绍

3.1. 协议数据库——SRP Grid

Subnets	Core1	Core2	Core3	Core4	Description
10.1.1.64/26	1	1	1	1	T2_Production
10.1.1.128/26	1	1	1	1	T3_Production
……	…	…	…	…	
10.1.32.192/26	1	1	1	1	T128_Production
0.0.0.0/0	1	1	1	1	B_exit

ToR1_Grid

Subnets	ToR1	ToR2		ToR128	Border1	Border2	Description
10.1.1.0/26	1	*	*	*	*	*	T1_Production
10.1.33.0/26	1	*	*	*	*	*	T1_ILO
10.1.0.41/32	1	*	*	*	*	*	T1_Loopback
10.1.1.64/26	*	1	*	*	*	*	T2_Production
10.1.33.64/26	*	1	*	*	*	*	T2_ILO
10.1.0.42/32	*	1	*	*	*	*	T2_Loopback
……	…	…	…	…	…	…	…
10.1.32.192/26	*	*	*	1	*	*	T128_Production
10.1.64.192/26	*	*	*	1	*	*	T128_ILO
10.1.0.168/32	*	*	*	1	*	*	T128_Loopback
0.0.0.0/0	*	*	*	*	1	1	B_exit

Core1_Grid

Subnets	Core1	Core2	Core3	Core4	Border2	Description
10.1.1.0/26	1	1	1	1	3	T1_Production
10.1.33.0/26	1	1	1	1	3	T1_ILO
10.1.0.41/32	1	1	1	1	3	T1_Loopback
10.1.1.64/26	1	1	1	1	3	T2_Production
10.1.33.64/26	1	1	1	1	3	T2_ILO
10.1.0.42/32	1	1	1	1	3	T2_Loopback
……	…	…	…	…		
10.1.32.192/26	1	1	1	1	3	T128_Production
10.1.64.192/26	1	1	1	1	3	T128_ILO
10.1.0.168/32	1	1	1	1	3	T128_Loopback
10.1.0.1/32	1	*	*	*	3	C1_Loopback
10.1.0.2/32	*	1	*	*	3	C2_Loopback
10.1.0.3/32	*	*	1	*	3	C3_Loopback
10.1.0.4/32	*	*	*	1	3	C4_Loopback

Border1_Grid

图表 4 协议核心 SRP Grid

SRP 实现的核心围绕着 SRP Grid，它是 SRP 预设路由的数据库，每台设备均有独立的 SRP Grid，如图表 4 所示：

1. 根据网络架构中的设备类型 ToR、Core 和 Border 共分为 3 类 Grid；

2. Grid 是一种表格，行表示目的 IP 子网，列表示 SRP 邻居；

3. 行和列所确定的 Cell 即为 SRP 邻居作为目的 IP 子网的可达性状态，Cell 内容有“*”和整数，字符“*”表示永远不可达（是 SRP 初始化默认值之一，且永远不会改变），SRP 不会为该 Cell 产生静态路由，Cell 数值为奇数表示目前可达，偶数表示目前不可达（也是 SRP 初始化的默认值之一）；

4. ToR 和 Border Grid 比较类似，可以看到不同树的转发路径，如 Core1 列就表示以 Core1 为根的树，以此类推，图中使用不同底色表示不同的树；

5. Core 的 Grid 非常简单，可以发现除了默认路由外每一行只有 1 个 Cell 是整数值，其余皆为*，这体现了树形转发的特点，也是避免计算无效路由的法门；

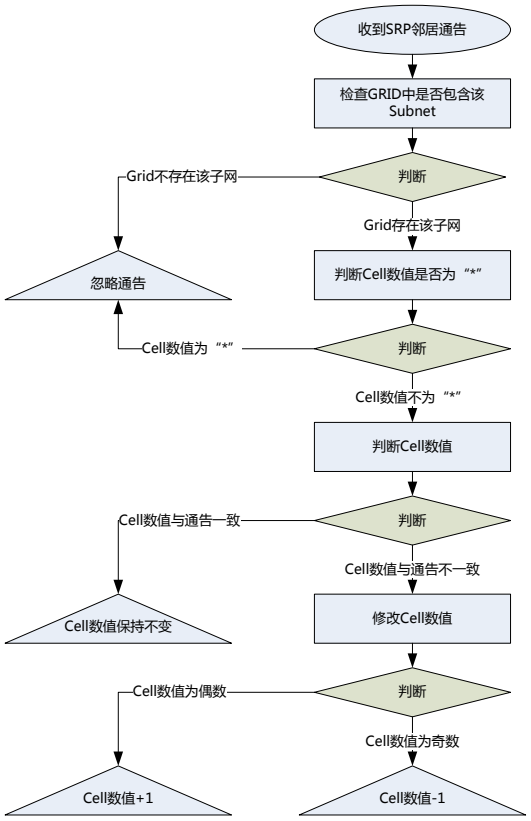
6. Border 的 Grid 可以看到有数值为 3 的 Cell，这种特殊的 Cell 只在 Border 之间存在，主要是为 IDC 外部访问 IDC 内部子网备份路径——当某一行中没有数值为 1 的 Cell 时（该 Border 至该 IP 子网已无优选 Next hop），该 Cell 才会生效（表示可以通过 Border 邻居作为 Next hop），不同数值的 Cell 通过设置静态路由的 Distance 以示区分。

SRP Grid 文件是 txt 格式的，为了使 Grid 更易于维护，白皮书中我们使用表格方式介绍。

3.2. Cell 数值变化的原则

不同的设备在初始化时加载各自的 Grid 文件，在运行过程中不会修改 Grid 的行和列，只会修改数值 Cell 的数值，修改 Cell 数值只有 2 种途径：

1. SRP 检测到邻居关系中断，将该邻居列所有奇数 Cell 数值减 1，如 1-1=0，3-1=2；



图表 5 收到邻居通告后的处理流程

2. 如图表 5 所示，若 SRP 邻居关系正常（包括从中断恢复或一直正常），从邻居收到消息做如下处理：

   a) 消息中某个 IP 子网状态变化，检查本地 Grid 中是否包含该 IP 子网，如果没有则丢弃该消息；

   b) 若存在该 IP 子网，则检查该 IP 子网所在行、该邻居所在列的 Cell 数值是否是“*”，如果是“*”则继续丢弃该消息；

   c) 如果不是“*”，则检查该 Cell 数值与消息通告状态是否一致，如果一致则保持数值不变；

   d) 如果状态不一致则，Cell 数值为偶数时加 1（如 0+1=1，2+1=3），Cell 数值为奇数时减 1（如 1-1=0，3-1=2）。

当 Cell 数值从偶数变奇数时会产生 1 条静态路由，从奇数变偶数时会撤销这条静态路由。

3.3. 增加、删除子网和邻居

SRP 的工作环境为规模化 IDC，这类 IDC 的地址规划和网络架构规划类似，非常的稳定，很少发生变化，因此子网、邻居都是可以根据规划提前制定的（并非按照建设进度制定，如某个 ToR 还没有上架，但是这个 ToR 是在规划内的，相关的子网和邻居却早已在 Grid 中制定），这个现象已经通过现有大规模 IDC 的长期运营验证。

加入网络因为规划的变动需要增加、删除子网和邻居，则需要重新设计更新后的 Grid，所有的 Core、ToR、Border 需要重新重新加载 Grid 文件：

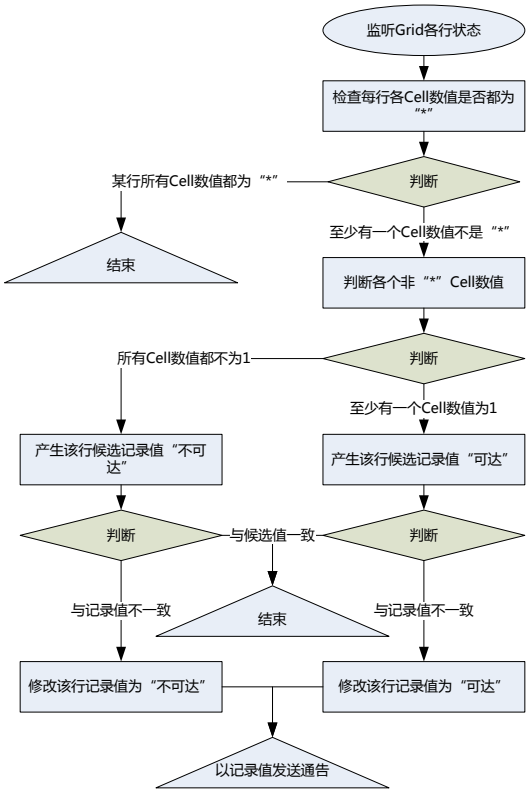
1. 当前实现的加载方式为静态加载，即需要重新启动 SRP 进程的方式加载；
2. 将来可以在实现方式上进行优化，实现动态加载，即只进行增量的添加或删除，对 Grid 中未曾发生变化的单元格不改变 RIB 和 FIB。

3.4. SRP 协议状态机和消息分类

Grid 是 SRP 工作的核心，而 SRP 的邻居关系是 Cell 数值变化的唯一途径：

1. SRP 邻居协议状态机和 BGP 完全一致，监听端口为 TCP 40079；
2. SRP 邻居协议消息格式与 BGP 基本一致，但只只有 3 种：Open、Keepalive 和 Update，其中 Update 有 2 种子类型，一种是失效 IP 子网列表，另外一种是生效 IP 子网列表；
3. SRP 默认情况下使用 Keepalive 消息来维持邻居关系，时间参数与 BGP 一致，也可以和 BFD 结合加速邻居关系的探测；
4. SRP 建立邻居也是和 BGP 类似的，除了需要指定邻居 IP 地址和建立邻居关系的 source interface 外，还需要指定邻居的类型（如 Core、ToR 或 Border）和编号（如 1、2、111 等）以进行身份验证，值得一提的是 SRP 作为 IGP，目前不建议建立 multi-hop peer。

3.5. 根据 Grid 变化发送 Update 通告



图表 6 收到邻居通告后的处理流程

SRP 作为 IGP，有两种方式产生路由 Update：

1. 被 SRP 所卷入的 Connect 路由发生变化；
2. SRP 所管理的路由状态发生变化。

第一种方式在所有 IGP 中都是通用的，而第二种则重点体现 IGP 的设计原理。

如图表 6 所示，邻居协议部分根据 Grid 发送 Update 消息：

1. 监听 Grid 各行（IP 子网）状态，当一行之中皆无数值为 1 的 Cell 时（数值为 3 的 Cell 不作为可达性考察范围），表示该 IP 子网不可达，需要向邻居发送 Update 消息通告该 IP 子网失效；

2. 若某行由 2 个数值为 1 的 Cell 变成 1 个或者 3 个，都不需要发送 Update 消息，因为该 IP 子网一直处于可达状态；

3. 若某行由 0 个数值为 1 的 Cell 变成至少 1 个，那么就需要向邻居发送 Update 消息通告该 IP 子网生效。

3.6. 监听 Connect 路由和 Outside 路由发送 Update 通告

SRP 作为一种 IGP，其核心职责就是要将 Connect 路由或者 Outside 路由状态通过邻居协议发布出去。SRP 和 BGP 一样，使用 network 命令来监控 Connect 路由或者 Outside 路由状态：

1. ToR、Core 都需要监听 Connect 路由状态（根据 network 命令监听指定 IP 子网），当指定 IP 子网 Connect 路由从不存在变成存在时，即需要向所有邻居发送 Update 通告该 IP 子网生效，反之发送 Update 通告该 IP 子网失效；

2. Border 则需要 network 多条 Outside 路由（IDC 外部路由，有可能是 OSPF/BGP/IS-IS 类型），将这些路由状态取或后与发布的默认路由状态绑定——只有当所有 Outside 路由都失效时，该 Border 才会发布 Update 消息通告所有邻居默认路由失效，否则默认路由都是生效的，因此在 Border 上选择外部路由条目时比较关键，尽量选择哪些必须使用、稳定使用的 Outside 路由，如集中网管服务器网段等，此外将来还可以通过 IP-Detection 结果与默认路由状态绑定方式加以实现。

4. Study Case

4.1. SRP 初始化时各个 Grid 的状态

如图表 1 所示，IDC 内运行 SRP，Border 和 Outside IDC 运行其余路由协议

Subnets	Core1	Core2	Core3	Core4	Description
10.1.1.64/26	0	0	0	0	T2_Production
10.1.1.128/26	0	0	0	0	T3_Production
……	…	…	…	…	
10.1.32.192/26	0	0	0	0	T128_Production

ToR1_Grid

Subnets	ToR1	ToR2		ToR128	Border1	Border2	Description
10.1.1.0/26	0	*	*	*	*	*	T1_Production
10.1.33.0/26	0	*	*	*	*	*	T1_ILO
10.1.0.41/32	0	*	*	*	*	*	T1_Loopback
10.1.1.64/26	*		0	*	*	*	T2_Production
10.1.33.64/26	*		0	*	*	*	T2_ILO
10.1.0.42/32	*		0	*	*	*	T2_Loopback
……	…	…	…	…	…	…	
10.1.32.192/26	*	*	*	0	*	*	T128_Production
10.1.64.192/26	*	*	*	0	*	*	T128_ILO
10.1.0.168/32	*	*	*	0	*	*	T128_Loopback

Core1_Grid

Subnets	Core1	Core2	Core3	Core4	Border2	Description
10.1.1.0/26	0	0	0	0	2	T1_Production
10.1.33.0/26	0	0	0	0	2	T1_ILO
10.1.0.41/32	0	0	0	0	2	T1_Loopback
10.1.1.64/26	0	0	0	0	2	T2_Production
10.1.33.64/26	0	0	0	0	2	T2_ILO
10.1.0.42/32	0	0	0	0	2	T2_Loopback
……	…	…	…	…		
10.1.32.192/26	0	0	0	0	2	T128_Production
10.1.64.192/26	0	0	0	0	2	T128_ILO
10.1.0.168/32	0	0	0	0	2	T128_Loopback
10.1.0.1/32	0	*	*	*	2	C1_Loopback
10.1.0.2/32	*	0	*	*	2	C2_Loopback
10.1.0.3/32	*	*	0	*	2	C3_Loopback

Border1_Grid

图表 7 初始化运行时 ToR、Core、Border 的 Grid 状态

如图表 7 所示，初始状态下除了“*”Cell 外，其余 Cell 均为偶数值 Cell，此时假设 ToR2 的 SRP 监控到路由表中存在 3 条 Connect 路由 10.1.1.64/26、10.1.33.64/26、10.1.0.42/32，则立刻发送 Update 给所有邻居（Core1~Core4），Core1_Grid 如图表 8 所示：

Subnets	ToR1	ToR2		ToR128	Border1	Border2	Description
10.1.1.0/26	0	*	*	*	*	*	T1_Production
10.1.33.0/26	0	*	*	*	*	*	T1_ILO
10.1.0.41/32	0	*	*	*	*	*	T1_Loopback
10.1.1.64/26	*		1	*	*	*	T2_Production
10.1.33.64/26	*		1	*	*	*	T2_ILO
10.1.0.42/32	*		1	*	*	*	T2_Loopback
……	…	…	…	…	…	…	
10.1.32.192/26	*	*	*	0	*	*	T128_Production
10.1.64.192/26	*	*	*	0	*	*	T128_ILO
10.1.0.168/32	*	*	*	0	*	*	T128_Loopback

Core1_Grid

图表 8 当 ToR2 发布 Update 后 Core1 Grid 的状态

当 Core1 检测到这 3 个 IP 子网所在行有数值为 1 的 Cell 后，根据“Vertical Split”原则，发送 Update 给同行中“*”值邻居（除 ToR2 外的 ToR 和 Border），如图表 9、图表 10 所示：

Subnets	Core1	Core2	Core3	Core4	Description
10.1.1.64/26	1	0	0	0	T2_Production
10.1.1.128/26	0	0	0	0	T3_Production
……	…	…	…	…	
10.1.32.192/26	0	0	0	0	T128_Production

ToR1_Grid

图表 9 当 Core1 发布 Update 后 ToR1 Grid 的状态

Subnets	Core1	Core2	Core3	Core4	Border2	Description
10.1.1.0/26	0	0	0	0	2	T1_Production
10.1.33.0/26	0	0	0	0	2	T1_ILO
10.1.0.41/32	0	0	0	0	2	T1_Loopback
10.1.1.64/26	1	0	0	0	2	T2_Production
10.1.33.64/26	1	0	0	0	2	T2_ILO
10.1.0.42/32	1	0	0	0	2	T2_Loopback
……	…	…	…	…		
10.1.32.192/26	0	0	0	0	2	T128_Production
10.1.64.192/26	0	0	0	0	2	T128_ILO
10.1.0.168/32	0	0	0	0	2	T128_Loopback
10.1.0.1/32	0	*	*	*	2	C1_Loopback
10.1.0.2/32	*	0	*	*	2	C2_Loopback
10.1.0.3/32	*	*	0	*	2	C3_Loopback

Border1_Grid

图表 10 当 Core1 发布 Update 后 Border1 Grid 的状态

同时 Core1 发送的 Update 也会通知到 Border2，Border2 的 Grid 和 Border1 类似，Border2 检查到 3 个 IP 子网所在行也有数值为 1 的 Cell，也会向邻居 Border1 发送 Update 修改数值为 2 的 Cell，如图表 11 所示：

Subnets	Core1	Core2	Core3	Core4	Border2	Description
10.1.1.0/26	0	0	0	0	2	T1_Production
10.1.33.0/26	0	0	0	0	2	T1_ILO
10.1.0.41/32	0	0	0	0	2	T1_Loopback
10.1.1.64/26	1	0	0	0	3	T2_Production
10.1.33.64/26	1	0	0	0	3	T2_ILO
10.1.0.42/32	1	0	0	0	3	T2_Loopback
……	…	…	…	…		
10.1.32.192/26	0	0	0	0	2	T128_Production
10.1.64.192/26	0	0	0	0	2	T128_ILO
10.1.0.168/32	0	0	0	0	2	T128_Loopback
10.1.0.1/32	0	*	*	*	2	C1_Loopback
10.1.0.2/32	*	0	*	*	2	C2_Loopback
10.1.0.3/32	*	*	0	*	2	C3_Loopback

Border1_Grid

图表 11 当 Border2 发布 Update 后 Border1 Grid 的状态

4.2. 当 Border1 和 Core1 邻居关系中断

假设 Border1 和 Core1 在正常情况下 Cell 如图表 12 所示：

Subnets	Core1	Core2	Core3	Core4	Border2	Description
10.1.1.0/26	1	1	1	1	3	T1_Production
10.1.33.0/26	1	1	1	1	3	T1_ILO
10.1.0.41/32	1	1	1	1	3	T1_Loopback
10.1.1.64/26	1	1	1	1	3	T2_Production
10.1.33.64/26	1	1	1	1	3	T2_ILO
10.1.0.42/32	1	1	1	1	3	T2_Loopback
……	…	…	…	…		
10.1.32.192/26	1	1	1	1	3	T128_Production
10.1.64.192/26	1	1	1	1	3	T128_ILO
10.1.0.168/32	1	1	1	1	3	T128_Loopback
10.1.0.1/32	1	*	*	*	3	C1_Loopback
10.1.0.2/32	*	1	*	*	3	C2_Loopback
10.1.0.3/32	*	*	1	*	3	C3_Loopback
10.1.0.4/32	*	*	*	1	3	C4_Loopback

Border1_Grid

图表 12 正常情况下 Border1 的 Grid 状态

当 Border1 检测到和 Core1 关系中断后，Grid 会发生如图表 13 样变化：

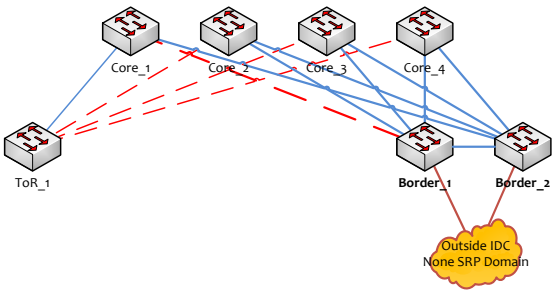
Subnets	Core1	Core2	Core3	Core4	Border2	Description
10.1.1.0/26	0	1	1	1	3	T1_Production
10.1.33.0/26	0	1	1	1	3	T1_ILO
10.1.0.41/32	0	1	1	1	3	T1_Loopback
10.1.1.64/26	0	1	1	1	3	T2_Production
10.1.33.64/26	0	1	1	1	3	T2_ILO
10.1.0.42/32	0	1	1	1	3	T2_Loopback
……	0	…	…	…		
10.1.32.192/26	0	1	1	1	3	T128_Production
10.1.64.192/26	0	1	1	1	3	T128_ILO
10.1.0.168/32	0	1	1	1	3	T128_Loopback
10.1.0.1/32	0	*	*	*	3	C1_Loopback
10.1.0.2/32	*	1	*	*	3	C2_Loopback
10.1.0.3/32	*	*	1	*	3	C3_Loopback
10.1.0.4/32	*	*	*	1	3	C4_Loopback

Border1_Grid

图表 13 当 Border1 和 Core1 邻居关系中断后的 Grid

4.3. 当 Core2~Core4 与 ToR1 邻居中断

在 4.2 节的基础（Border1 与 Core1 邻居中断）上，网络继续发生变化，Core2~Core4 与 ToR1 邻居相继中断，网络拓扑如图表 14 所示：



图表 14 在 Border1-Core1 中断基础上 Core2~Core4-ToR1 中断

Core2~Core4 各自发送 Update 消息至 Border1 和 Border2，此时 Border1 的 Grid 会如图表 15 所示：

Subnets	Core1	Core2	Core3	Core4	Border2	Description
10.1.1.0/26	0	0	0	0	3	T1_Production
10.1.33.0/26	0	0	0	0	3	T1_ILO
10.1.0.41/32	0	0	0	0	3	T1_Loopback
10.1.1.64/26	0	1	1	1	3	T2_Production
10.1.33.64/26	0	1	1	1	3	T2_ILO
10.1.0.42/32	0	1	1	1	3	T2_Loopback
……	0	…	…	…		
10.1.32.192/26	0	1	1	1	3	T128_Production
10.1.64.192/26	0	1	1	1	3	T128_ILO
10.1.0.168/32	0	1	1	1	3	T128_Loopback
10.1.0.1/32	0	*	*	*	3	C1_Loopback
10.1.0.2/32	*	1	*	*	3	C2_Loopback
10.1.0.3/32	*	*	1	*	3	C3_Loopback
10.1.0.4/32	*	*	*	1	3	C4_Loopback

Border1_Grid

图表 15 和 ToR1 相关的 IP 子网均无数值为 1 的 Cell——Border1 Grid

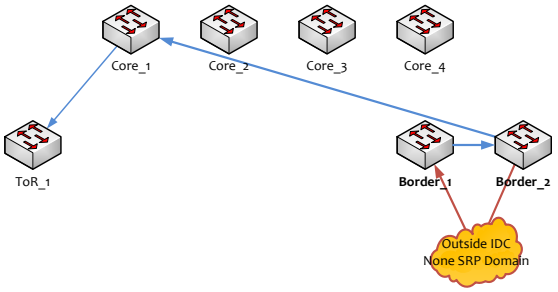
由于 Border2 也接收到 Core2~Core4 发送的 Update 消息，对应 Core2~Core4 列 ToR1 相关行的 Cell 也会变成 0，同时此时 Border1 会发送 Update 至 Border2，Border2 的 Grid 会如图表 16 所示，对比图表 15 可以发现 Border1 和 Border2 之间并未产生 ToR1 各子网的环路：

Subnets	Core1	Core2	Core3	Core4	Border1	Description
10.1.1.0/26	1	0	0	0	2	T1_Production
10.1.33.0/26	1	0	0	0	2	T1_ILO
10.1.0.41/32	1	0	0	0	2	T1_Loopback
10.1.1.64/26	1	1	1	1	3	T2_Production
10.1.33.64/26	1	1	1	1	3	T2_ILO
10.1.0.42/32	1	1	1	1	3	T2_Loopback
……	1	…	…	…		
10.1.32.192/26	1	1	1	1	3	T128_Production
10.1.64.192/26	1	1	1	1	3	T128_ILO
10.1.0.168/32	1	1	1	1	3	T128_Loopback
10.1.0.1/32	1	*	*	*	3	C1_Loopback
10.1.0.2/32	*	1	*	*	3	C2_Loopback
10.1.0.3/32	*	*	1	*	3	C3_Loopback
10.1.0.4/32	*	*	*	1	3	C4_Loopback

Border2_Grid

图表 16 在 Border2 上接收到 Core2~Core4、Border1 发送 Update 后

此时 Outside IDC 访问 ToR1 子网路径即会如图表 17 所示：



图表 17 当 Outside IDC 通过 Border1 访问 IDC 内 ToR1 时，路径经过 Border1 和 Border2 之间链路

通常 Border 与每个 Core 互联通常不会是单条链路，同时拥有 4 个 Core 的网络中，因此发生这种故障（Border1-Core1 中断，同时 Core2~Core4-ToR1 都中断）的概率极其小，但是在只有 2 个 Core 的网络中的概率相比而言会大一些（Border1-Core1 中断，同时 Core2-ToR1 中断），而此时备份 Cell 的作用就会凸显。

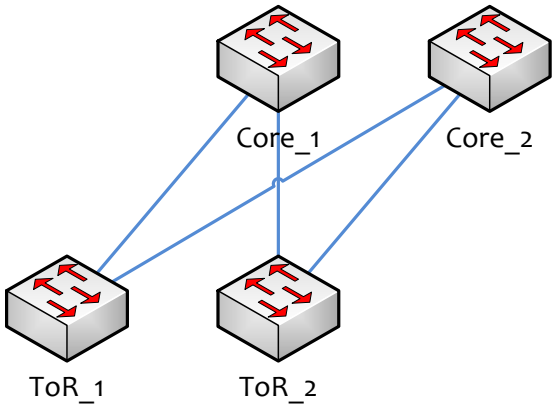
4.4. 引进控制器后可以取消备份 Cell

未来 SRP 的另外一种做法可以取消 Border 间互联链路：

1. 设置集中控制器 SRP Orchestrator 监控 Border1 和 Border2 的 Grid 状态；
2. 当 Border1 上对于某个的子网均无数值为 1 的 Cell，而 Border2 上相同子网存在数值为 1 的 Cell；
3. SRP Orchestrator 会在 Border2 上下发指令将该子网对应的 SRP 路由注入到 Border2 与 Outside IDC 互联的路由进程中；
4. 由于 Outside IDC 从 Border1 上只学习到汇聚路由，而从 Border2 却学到更为精确的 SRP 路由，因此 Outside IDC 访问该 IP 子网时会优选 Border2，从而解决 IDC 内路由汇聚后的高可用路径问题。

5. SRP 当前测试数据

5.1. 测试拓扑环境



图表 18 测试拓扑环境

如图表 18，测试拓扑分别运行 OSPF 和 SRP 后进行，操作为：

1. ToR1~ToR2 之间通过测试仪模拟稳定的双向流量；
2. 在 ToR2 上 Shutdown 和 Core1 之间的链路；
3. 在 ToR2 上执行 NO shutdown 和 Core1 之间链路；
4. 分别观察 ToR1→ToR2 和 ToR2→ToR1 的流量中断数量。

从使用收敛时间 $= \frac{流量中断数量}{流量平均速率}$，对比 OSPF 和 SRP 的收敛时间，由于是在 ToR2 上执行 Shutdown 操作，所以 ToR2→ToR1 方向属于本地 FIB 更新时间，和路由协议无关，而 ToR1→ToR2 方向则经历了：

1. Core1 感受到与 ToR2 邻居中断；
2. Core1 向 ToR1 发布 Update；
3. ToR1 刷新 Grid；
4. ToR1 修改路由表；
5. ToR1 更新 FIB。

等 5 项动作，此收敛时间更具实际价值。

5.2. 测试结果

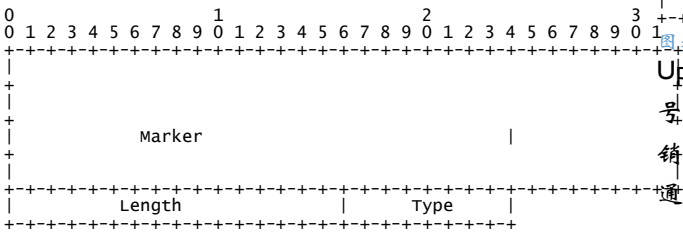
	ToR1→ToR2 协议收敛时间	*ToR2*→ToR1 本地刷新时间
OSPF 链路中断	0.55s	0.51s
OSPF 链路恢复	0.00s	0.08s
SRP 链路中断	0.87s	0.50s
SRP 链路恢复	0.00s	0.00s

图表 19 测试结果对比

从对比结果来看 OSPF 在链路中断场景中拥有更快的收敛时间，其余场景 SRP 均不弱于 OSPF。这主要是 SRP 目前的机制尚未优化至最佳，理论上可以和 BGP 相同，而 OSPF 在更大规模的测试结果表明其收敛时间恶化较多。

6. 附录

6.1. SRP 消息头



图表 20 消息格式——SRP 消息头

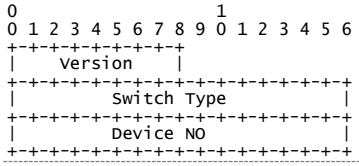
Marker（标记）：该标记在 SRP 中还可以用于邻居身份确认。

长度(Length)：包含消息头在内的 SRP 消息长度，单位是字节。

类型（Type）：一字节的无符号整数制定了消息类型编码。如下定义：

1. Open 消息——1；
2. Update 消息——2；
3. Keepalive 消息——4。

6.2. Open 消息格式



图表 21 消息格式——Open 消息

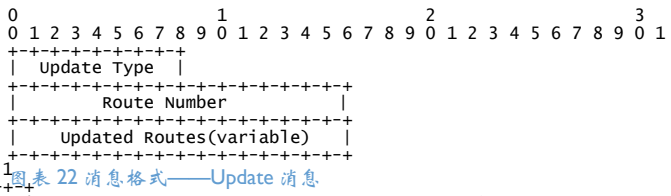
Version（版本）：1 字节整数，指示消息的协议版本号。当前的 SRP 版本号是 2。

Switch Type（交换机类型）：2 字节整数，指示当前设备的类型，目前有 3 种：

1. Core——0；
2. ToR——1；
3. Border——2。

Device NO（设备编号）：2 字节无符号整数指示当前设备的编号，从 1 开始。

6.3. Update 消息

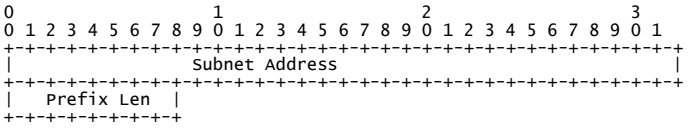


图表 22 消息格式——Update 消息

Update Type（Update 消息类型）：1 字节无符号整数指示了该消息是通告路由添加还是撤销，1 表示该消息通告添加路由，2 表示该消息通告撤销路由。

Route Number（路由条数）：2 字节无符号整数指示了该消息通告的路由更新条数。

Updated Routes(variable)（更新的路由，可变长度）：这是一个可变长字段，包括一系列的 IP 前缀将要更新的路由。每一个 IP 前缀编码为〈子网地址，子网掩码长度〉二元组，每个二元组有 5 字节，如图表 20 描述：



图表 23 消息格式——IP 子网信息

Subnet Address(子网地址)：4 字节指示了子网的 IP 地址。

Prefix Len（前缀长度）：1 字节指示了子网掩码的长度。

6.4. Keepalive 消息

SRP 的 Keepalive 消息和 BGP 一样，只包含消息头，并未采用特殊消息格式，Keeplive 的发送周期可以调节以实现不同的邻居关系检测速度。

6.5. 参考资料

1. Wikipedia. http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm.
2. Mohammad Al-Fares, Alexander Loukissas and Amin Vahdat. A Scalable, Commodity Data Center Network Architecture. ucsd_sigcom08_fattree.pdf.

后记

SRP 的未来

SRP 当前以 3000 行左右的核心代码实现了作为 IGP 的必要功能，还有接近 4000 行的代码初步实现了比较友好的人机接口。

Subnets	Core1	Core2	Core3	Core4	Description
10.1.1.64/26	50%	25%	25%	0	T2_Production
10.1.1.128/26	25%	25%	50%	0	T3_Production
……	…	…	…	…	
10.1.32.192/26	25%	50%	25%	0	T128_Production
0.0.0.0/0	25%	25%	25%	25%	B_exit

ToR1_Grid

图表 24 消息格式——IP 子网信息

SRP 的未来方向是要实现权重路由：ToR 和 Border 的 Cell 数值不再是整数，而是百分比，如图表 21 所示，表示流量在 ECMP 链路上的负载分担因子，而百分比是受 SRP Orchestrator 管理的。为了朝这方面发展，SRP 在未来的实现也许有可能和 OpenFlow 结合起来。