环形拓扑上快速生成树协议的简单协议增强

M. Marchese, M. Mongelli, G. Portomauro

DIST-通信，计算机和系统科学系
CNIT-意大利国家电信联盟
热那亚大学，Via Opera Pia, 16145，热那亚，意大利
{ mario.marchese, maurizio.mongelli, giancarlo.portomauro }@cnit.it

摘要: 本文讨论了使用快速生成树协议（RSTP）解决在以太网网络上的弹性问题。由于几乎无法在文献上找到关于 RSTP 实际性能的数据，因此这个主题是了一个开放的辩论问题。实际上，复杂的协议结构使得分析变得更加错综复杂，不适合一概而论。此外还存在其他的弹性算法，其机制和规则专为网络弹性设计，用于解决超出 RSTP 算法应用范围的问题。尽管这些算法可能比 RSTP 算法更高效，但是他们的开销大得多。从这个角度看，本文的目的就是在批判性的评估 RSTP 固有局限性的同时，提出一些简单协议的修改来提高 RSTP 的性能。通过分析验证了在快速反应和带宽的开销间进行权衡来提高可实现的性能的可行性。

关键词：网络弹性，RSTP，环形拓扑

I. 引言：

以太网被认为是最好的部署城域网（MANS）的方案。其中最重要的问题之一是弹性，即在网络故障时维持用户流量的能力。生成树协议及快速生成树协议[1] (RSTP, [1]，第 17 章)是以太网设备的逻辑链路控制(LLC)模块, 也称为桥(Brs), 他们用于:1)在 Brs 之间生成逻辑树形拓扑;2)支持弹性。它们使用专用的控制数据包，称为网桥协议数据单元(BPDUs)。动作 1)是需要的，以避免冗余以太网网络路径上的流量循环。就问题 2)而言，设置 BPDU 的拓扑变化(TC)标志是为了在检测到网络的某些变化时(例如在故障之后)发出 TC 信号。

我们工作的理论依据是目前的 RSTP 算法的真实性能仍存在一些争议。在这个问题上存在未解决的争论。部分文献的结果概述了良好的性能（例如[2,3]），其他的文献则没有[4,5]。尽管最近的行业报告显示了 RSTP 的良好性能[7,8]，但是网络工程师对 RSTP 算法的普遍看法是负面的[6]。从这个角度出发，本文研究了如何通过引入轻量级协议修改来实现最佳性能。由于环形拓扑的广泛使用以及为了简化分析，我们将其纳入考虑的范围。我们关注的性能指标是在环形网络中发生（链路或节点）故障后恢复用户流量所需的时间。理想的目标（本文中用T_c^* 表示）为$T_c^* = \sum_{l=1}^{n} d_l$，其中 n 为 Brs 的环网中 Brs 的数量，d_l为当与链路l的相邻端口产生 BPDUs，第l条链路的延迟加上 Br 的处理延迟。

为了保证以太网网络的弹性，还设计了大量其他专门的协议:例如环形 RSTP (RRSTP)、Viking、以太网自动保护交换(EAPS)、弹性分组环(RPR)、具有 Epochs 的 RSTP[2]，以及最近的 ITU-T G.8032 标准，也包括 GMPLS[3]。专门的协议理论上会有上诉的理想性能（T_c^*）。然而，缺点之一，它们是在城域网核心中实现的，而常规 STP 在任何情况下都存在于局域网(LANS)上。出于这个原因，建议在 LAN 和 MAN 上共同采用一种独特的协议(RSTP)。这将降

低成本，简化网络管理，从而消除不同协议之间的互通需求。

本文的其余部分组织如下。下一节将概述 RSTP 协议的特性。在第三节中，详细介绍了弹性的固有限制和提议的协议增强。第四节调查了性能评估，第五节最后概述了结论和未来的工作。

II. RSTP OVER RING：协议特点

在下文中，术语 *常规工作状态* 定义了一段没有发生故障的时间，配置 BPDUs 发送时没有任何 TC 指示. 配置 BPDUs 维护由 Brs 使用 STP 或 RSTP 计算出的虚拟树连通性。树的第一个元素叫做根 Br。下文中它对协议有重要的作用。端口的基本状态有:*转发*(即端口允许转发流量帧)和*丢弃*(即端口不转发流量帧)。Br 的学习过程是通过“嗅探”转发帧来记忆内部 *转发数据库*(FDB)中的用户地址。当一个帧到达一个端口时，FDB 告诉通过哪个端口可以到达目的地。当目的端不存在于 FDB 中时，报文会在所有处于转发状态的端口上“泛洪”。由 STP 产生的虚拟树拓扑结构确保在泛洪的情况下不会产生环路。

在 RSTP 中，定义了一些新的端口角色:*备用* (作为到达根的替换方式)和 *备份*(作为给定局域网连接的另一种方式)。如果检测到 TC 时，具有这些角色的端口在会立即被唤醒。如果接收到的 BPDU 链路包与路由的端口角色不一致，或由物理层直接发出故障信号时，Br 会推断出 TC。出现 TC 后，如果角色为“*备份*”，则不需要对网络的树配置进行其他更改。如果角色是“*备用*”，则会触发一个建议-协议机制，让 Brs 构建新的树状配置。

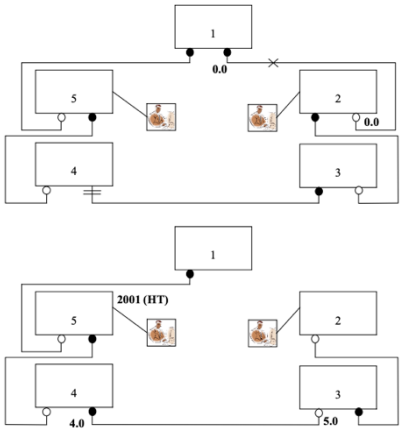

图 1. 5 个节点环上的 RSTP; (a)故障前-上，(b)故障后-下。

5 节点环上的 RSTP 端口角色在图 1(图 1(a))的顶部表示(图 1 的 *粗体* 值稍后将在后面的 III.C 小节中解释)。为了简单起见，在本文的所有图片中，MAC 地址被可视化为 Brs 的数字标识符。开始时(当网络“打开”时)，每个 Br 都将自己作为根结点;然后比较来自其他 Br 的消息的优先级，并将其 MAC 地址与其他 Brs 的地址进行比较，最后发现网络中具有较高优先级(值最低)的 Br 为根。对于环形情况，端口角色起始收敛时间T_t=$(n/2)+1$(n 为 br 个数)。T_t 后，根方向的端口变为“根端口”(按照标准符号，为空圈)，其余端口均为“指定端口”(转发流量的端口，用填充圈表示)，只有底部左下角的 Br 端口变为“备用”(图 1(a)中的 Br 4)。

A. 故障发生后的端口角色握手情况

图 1(b)为 br1 和 br2 链路故障后端口的新状态。这是收敛的最坏情况(即失去与根的连接)[8]。故障发生后需要重新配置 Brs 的部分端口角色;这个动作称为端口角色握手，并按如下方式进行。当 Br 2 检测到它的右端口被禁用时，它将声明自己为根节点;这个信息通过其他桥传播，直到到达 Br4，然后 Br4 通知原来的根仍然存在，并通过确认机制(该机制类似于用于初始化树的机制)在环的右侧重新分配端口角色，端口角

色握手的细节在[8]中。最终结果如图1(b)所示。本文中端口角色握手的时间(即故障后设置所有新端口角色的收敛时间)用T_c表示。

B. 清除 FDB 操作

值得注意的是，T_c并不是故障发生后正确转发流量前所需的所有操作完成的时间。T_c周期过后，RSTP 拓扑状态变化机(Tc State Machine，[1]中的 17.31 节)变为“*活跃*”状态(即进入“*传播*”状态)，并通知端口传输状态机转发常规 BPDU，将 Tc 标志位设置为 1。Br 接受此标志并产生刷新或者清楚 FDB 操作，其结果是删除 Br 在以前的操作中学习到的所有端口地址。这保证了在故障发生后流量将被正确转发(在 III.C 小节中说明了一个示例)。

III. 环状 RSTP：协议增强

该协议的基本定时器有:HelloTime (HT)、Forward Delay (FD)和 Max Age (MA)。HT 定义了 BPDU 报文的周期性重传，缺省范围为[1,10]s，缺省值为 2s。每个 Br 从它的端口周期性地向它的邻居发送 BPDU，以保持树的连通性。如果 TC SM 处于激活状态，则设置 TC 标志的 BPDU 间隔 HT 从根端口向邻居端口传播，直到 TC SM 过渡到非激活状态。FD 是强加给一些端口的状态转换延迟，默认范围是[4,30]s，默认值是 15s;MA 是端口接收 BPDU 到进入新状态的最大可接受时间;缺省值为[6,40]s，默认值是 20s。在提出的 RSTP 变体中，协议的基本计时器保留各自的默认值。这种选择的基本原理依赖于以下情况:1)细化定时器值对 STP 的收敛性有影响，而不是对 RSTP 的收敛性有影响;2)RSTP 对环型网络性能限制不是由于这些定时器[5]的特定值;3)在处理 RSTP 协议修改时，这些计时器通常不动[2]。回顾一下 RSTP 操作的时间粒度是 1s 是很有用的。这意味着 RSTP 协议每 1 秒唤醒一次，并在该时刻激活固有的 SMes。Br 的唤醒协议动作被称为$\mathrm{stpm_one_second}(\cdot)$ [9]。以下几个小节中提到的研究来自通过“BridgeSim”模拟器[9]进行的广泛的研究。从这个角度来看，一些模拟将显示协议最关键的特性。

A. TxHoldCount 效果

根据标准，TxHoldCount (TxHC)是 BPDU 的“端口传输状态机用来限制最大传输速率的值”;缺省值为[1,10]秒，默认值为 6s。当一个 BPDU 准备好传输，但上一秒的前次传输数量已经达到 TxHC 时，当前的传输将延迟 1s。一些例子有助于解释它对 Tc 的影响。

在不丧失一般性的情况下，现在考虑 17 个 Brs 的情况，以突出所有 TxHC 的含义。Br 的标识号和端口角色与图 1 类似。表 1 报告了 Tt(开始时的端口角色收敛时间，此时网络可以被认为是“接通”的)，以及最后一次注册清除 FDB 的时间(即在此时间之后，所有 Brs 的 TC SM 变得不活跃)作为 TxHC 的函数(表 1 中没有产生故障)。在 TC SM 不活跃之前引入的每个故障(表 2 和表 3)使得 TC 性能对 TxHC 非常敏感。请注意，如前所述，在故障发生后，TC 的活动期也是必要的。因此，TC SM 活动期间的每一次连续故障(或网元恢复)都会导致 Tc 的值比没有 TC SM 活动的故障更高。从这个角度来看，表 2-4 中的数值例子表明，在 TC SM 活动期间，BPDU 的传播数量随着时间的推移而减少，TxHC 的负面影响也随着时间的推移而减少。

TxHC		T_t[ms]		最近清除 FDB[s]
2		8001		40
3		7001		38
4		5002		38
5		5001		36
6		4001		36
7		3001		34

8		1002		34
9		9		32
10		9		32
Disabled		9		32

表 1　TxHC 效果- 5 节点环:Tt 性能和最后一个 clearFDB（在树结构完成之后）。

TxHC		Tc[ms]
2		5009
3		5009
4		3009
5		3001
6		2009
7		1009
8		1009
9		18
10		18
Disabled		18

表 2　TxHC 效应- 5 节点环:Tc 的性能(4s 时发生故障)。

TxHC		Tc[ms]
2		2009
3		2008
4		1009
5		18
6		18
7		18
Disabled		18

表 3　TxHC 效应- 5 节点环: Tc 的性能(8s 时发生故障)。

TxHC		Tc[ms]
2		1003
3		18
4		18
7		18
Disaabled		18

表 4　TxHC 效应- 5 节点环:Tc 性能（200s 时发生故障)。

根据分析结果，首先需要修改的协议是在连接环的端口上禁用 TxHC。通过对端口传输状态机的少量修改来禁用 TxHC 增量。在 SM 的“Transmit_config”、“Transmit_TCN”、“Transmit_RSTP”状态中取消“txCount+=1”指令(见[1]第 172 页图 17-17)即可。

B. message_age 效应

对于实施标准：“[…]每个 BPDU 包含一个信息年龄和最大年龄。信息年龄随着传播递增，如果超过最大年龄则丢弃信息。因此，信息可以通过的桥的数量是有限的。”这是必要的，因为可以避免旧的 BPDU 在冗余路径中无休止地循环，并影响新 BPDU 的传播。在实践中，设定 HT、FD 和 MA 定时器的默认值，并在每个 Br 处增加 message_age(为 1s)，将环的维度限制在不超过 18 Br。由于 message_age 上的一致性检查方程([1]的 17.21.23 小节)及其基本原理相当复杂(它们保持与 STP 的兼容性)，在此不做详细介绍。然而，为了避免 message_age 限制，只需对端口信息状态机进行简单的修改就足够了。这意味着只需要修改该状态机中的内部计时器“rcvdInfoWhile”。rcvdInfoWhile 是在一个端口上没有接收到其他 BPDU 情况下，该端口上保存的信息过期前的剩余时间。直观地说，如果一个端口上的 rcvdInfoWhile=HT，该端口就不能与相邻 Br 的其他端口保持一致的关系(即，在 rcvdInfoWhile 定时器到期之前，BPDU 没有从相邻 Br 到达);这种情况发生在给定的环的尺寸上(下方的“Rlimit”阈值)。更新端口信息状态机中的 rcvdInfoWhile 方程可

以稍作修改，如图 2 所示，设置环的大小。

```
//original version
        [...]
          if (eff_age < 1) {
            eff_age = 1;
          }
          eff_age += port->portTimes.MessageAge;
//modified version
        [...]
          if (eff_age < 1) {
            eff_age = 1;
          }
          eff_age += port->portTimes.MessageAge - (R - Rlimit);
```

图 2 UpdtRcvdInfoWhile()函数设置最大环大小

R 的值是预期的环的尺寸，Rlimit 是环尺寸极限，它取决于 HT(当 HT=2 时 Rlimit=18，当 HT=1 时 Rlimit=19)。这样的修改保证了当环尺寸为 R 时，rcvdInfoWhile(其整个更新方程没有在图 2 中显示)始终高于 HT。

C. ：clearFDB 效应

如上所述，在故障发生后检测到拓扑结构的变化，TC SM 开始传输 BPDU，并设置了 TC 标志，从而触发整个网络的 clearFDB 操作。只有所有 Brs 执行上述操作，流量才会被正确转发。如图 1 所示，其中 clearFDB 的时间(单位为 ms)以粗体表示。请注意，Br 5 在 2001ms(一个 HT，图 1 的底部)后才刷新它的端口。这种行为的基本原理依赖于 TC SM 的时间粒度，而时间粒度依赖于 HT。因此，clearFDB 的指示是以这种粒度在整个网络中传播。如图 1 所示，同样的时间段被视为丢失帧的时间段，因为 br5 在故障发生后需要两秒钟(一个 HT)才能停止在错误的方向(左端口)上转发流量。另一方面，由于传播网络右侧在 5ms 内收敛，如果位于 Br 2 的应用程序在 Tc 之后向正确方向转发流量，则 Br 5 立即被训练向正确方向转发流量。总而言之，流量丢失的时间取决于应用程序的实际行为。同样的概念适用于图 3 中的 11 Brs 的例子：即使 Br 11 在 7001ms 后接收到 clearFDB 指示，右侧也可以在 12ms 内正确转发流量。

一个简单的解决方案是使用更细的时间粒度激活协议(通过其主要函数，该函数调节内部 SMes，称为“*stpm_one_second(·)*”)。这意味着通过大幅降低 RSTP 的时间粒度，让 clearFDB 传播速度比常规情况快得多。在这种情况下定义Δ_{st} 为 stpm_one_second(·)随时间变化的重复频率。对于 RSTP 协议，HT 参数定义了端口在发送 BPDU 前等待Δ_{st}的时间。当 HT=1 时，表示 BPDU 每Δ_{st}发送一次。注意，此解决方案在某种程度上回顾了 EAPS 或 ITU-T G.8032 使用的“心跳”连续性检查(以高频率发送控制数据包，例如每 1 毫秒发送一次)，以监督环连通性并对故障做出反应。

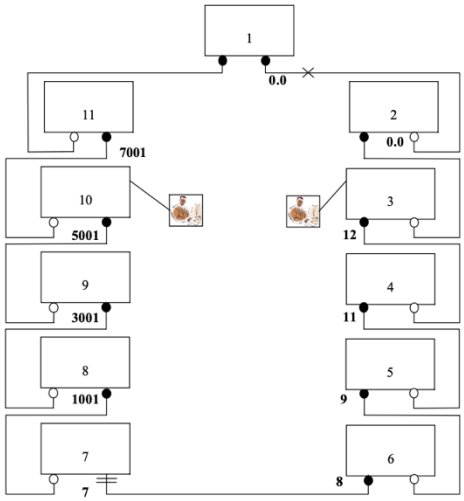

图 3. 11 个节点环上的 RSTP ：右侧是 clearFDB 的操作

作为加速 RSTP 的一个缺点，端口角色握手会产生大量 BPDU，这些 BPDU 的带宽开销需要精确评估，以保证 RSTP 优先于常规数据包或不太关键的指令(例如，以太网 GVRP、GARP、GMRP 指令)。

D. 最终性能

所有的协议修改在这里被定义为 RSTP over Ring (RSTPoR)。要优化的最终性能是 Tc(端口角色握手的持续时间)，加上在网络的所有 Brs 中执行 clearFDB

操作所需的时间。更具体的说，需要使得T_{cFDB}的时间等于最后一个 Br 在故障发生后执行 clearFDB 的时间;此 Br 是在链路故障的另一侧与根相邻的 Br(图 1 中的 Br 5，图 3 中的 Br 11)。T_{cFDB}是对应于在应用层上看到的流量丢失持续时间的关键时间段。

IV. 性能评估与讨论

表 5 概述了 RSTPoR 的性能与Δ_{ST}和环大小的关系。所有链路的延迟均为 1ms。HT 的值设置为 1。考虑到需要提高 RSTP 的性能到T_C^*，表 5 的结果可以总结为：如果$\Delta_{ST}=5ms$,T_{cFDB} 的效率上限为 2.4*T_C^*，如果$\Delta_{ST}=1ms$，在 40 个 Brs 的情况下，$T_{cFDB} \cong T_C^*$，并且在最坏的情况下（70 Brs），T_C^*应该加 15ms 来得到T_{cFDB}，最终得到$T_{cFDB} \cong 1.2 * T_C^*$。

图 4 和图 5 将 BPDU 速率纳入考虑范围，定义了驱动网络管理器以优化 RSTPoR 的解决方案的实用指南。图 5 显示了由 stpm_one_second(·)在所选Δ_{ST}值下的加速产生的 BPDU 速率。速率计算公式为$(1/T_C) * (maxBPDU * DimFrame)$，其中' DimFrame '是承载 BPDU 的以太网帧的尺寸，' maxBPDU '是环端口在 Tc 过程中产生的 BPDU 的最大数量;实际上，Tc 是与端口角色握手相对应的产生 BPDU 数量最多的时间水平的过程。图 5 中的数值对应于在特定 Br 的特定端口上登记的最高速率，因为并非所有 Br 在握手过程中产生相同数量的 BPDU。因此，RSTPoR 性能必须与信令开销一起考虑。首先有必要找出与环路大小相关的理想故障恢复时间。可以找到Δ_{ST}值来匹配所需的性能（图 4)。其次，Δ_{ST}的固有信令开销如图 5 所示。当$\Delta_{ST}=1.0ms$，带宽的开销相当于在 CC (continuity check)速率基础上增加 18%，即每 1.0ms 发送一个 BPDU，相当于 424kbps 的速率。其中 CC 速率与 RSTPoR、EAPS 和 G.8032 通用。这种有限的开销导致考虑 RSTPoR 解决方案适用于广泛的实际情况，其中 CC 也被应用。

V.结论与未来工作

本文研究了快速生成树协议（RSTP）在环形拓扑上的故障恢复性能。该分析突出了协议的特征，并解释了如何从应用程序层面来加速流量恢复。未来的工作主要涉及 RSTPoR 对网状拓扑结构的推广，以及其他调整技术和 RSTP ‘epochs’ 理论。

参考文献

[1] IEEE Std 802.1DTM-2004, IEEE Standard for Local and metropolitan area networks, Media Access Control (MAC) Bridges.

[2] K. Elmeleegy, A.n L. Cox, T. S. Eugene Ng, “On Count-to-Infinity Induced Forwarding Loops in Ethernet Networks,” Proc. of IEEE Infocom 2006, vol. 25, no. 1, 23-29 Apr. 2006, pp. 1699-1711.

[3] S. Ilyas, A. Nazir, F. S. Bokhari, Z .A. Uzmi, A. Farrel, “A Simulation Study of GELS for Ethernet over WAN,” Proc. of IEEE Globecom 2007, 26- 30 Nov. 2007, pp. 2617 - 2622.

[4] J. Madsen, D. Tebben, A. Dwivedi, P. Harshavardhana, W. Turner, “Cross Layer Optimization in Assured Connectivity Tactical Mesh Networks,” Proceedings of the IEEE Military Communication Conference, Milcom 2008, 600 San Diego (CA), 17-19 Nov. 2008.

[5] A. Myers, T. S. Eugene Ng, H. Zhang, “Rethinking the Service Model: 500 Scaling Ethernet to a Million Nodes,”

Proceedings of the 3rd HotNets Conference,Nov.2004.

[6]T. Gimpelson, “Metro vendors question Spanning Tree standard,” Network World White paper 2001,http://www.networkworld.com/archive/2001/123588_08-06-2001.html.

[7] D. DesRuisseaux, “Use of RSTP to Cost Effectively Address Ring Recovery Applications in Industrial Ethernet Networks,” Proceedings of the 13th ODVA Network Conference, Howey-in-the-Hills, FL, USA, Feb. 2009.

[8] M. Galea, “Rapid Spanning Tree in Industrial Networks,” RuggedCom Inc. - Industrial Strength Networks, white paper, 2004, http://www.ruggedcom.com/pdfs/white_papers/rapid_spanning_tree_in_industrial_networks.pdf.

[9] BridgeSim: C++ Rapid Spanning Tree Protocol (RSTP) simulator, http://www.cs.cmu.edu/~acm/bridgesim/index.html.

致谢

作者对 Selex Communications 表示感谢，包括 M. Carniglia, A. Civardi, G. Fontana, A. Sogliani 和 L. Spinacci，他们在工作的发展过程中提供了有益的和建设性的建议。

Ring size Δ_{ST} [ms]	10	20	30	40	50	60	70
1000	2001	8001	12001	18001	23001	27001	32001
500	1001	4001	6001	9001	11501	13501	16001
200	401	1601	2401	3601	4601	5401	6401
100	101	801	1201	1801	2301	2701	3201
50	101	401	601	901	1151	1351	1601
10	21	81	131	191	251	301	361
5	16	46	76	106	136	166	196
1	10	21	34	47	61	73	85

表 5 所示。T_{cFDB} [ms]为ΔST 和环尺寸的函数

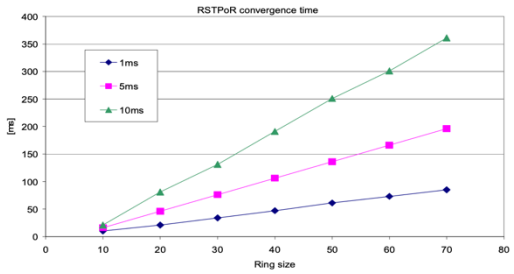

图 4 RSTPoR: T_{cFDB} 作为Δ_{ST}和环尺寸的函数。

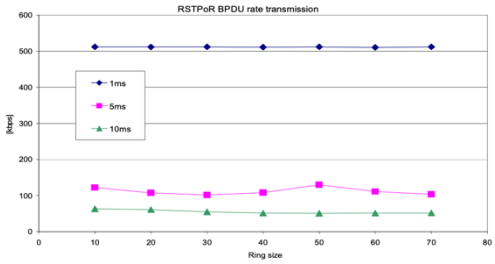

图 5。RSTPoR:信号开销作为环大小和Δ_{ST}的函数。