

一种基于分段共享的预测式错误恢复机制^{*}

徐洁 郭星廷 夏梦芹

(电子科技大学计算机学院 成都610054)

A Prognostic Segment-Based Sharing Recovery Mechanism

XU Jie GUO Xing-Ting XIA Meng-Qin

(College of Computer Science and Engineering, UEST of China, Chengdu 610054)

Abstract In MPLS/GMPLS networks, a range of restoration schemes will be required to support trade-offs between service interruption time and network resource utilization. Although, the tradition protection-based and reservation-based recovery schemes exist some merits, but they can not provide a good trade-offs. In this paper, we propose a Prognostic Segment-based Sharing Recovery Mechanism (PSSRM) for restoration of connection over shared bandwidth in a fully distributed GMPLS architecture. PSSRM can use QoS constraints such as bandwidth and recovery time to obtain the good trade-offs of recovery time and resource efficiency.

Keywords MPLS, GMPLS, RSVP-TE, Restoration

1 引言

路径错误恢复的目的在于自动地帮助网络管理者检测并且从错误状态中恢复,而无需人为的干涉,因此,不会引起网络用户的使用不便。通过应用一些错误恢复策略,服务提供者就能缩短错误的恢复时间和提高错误正确恢复的比率等,也就保证了服务质量协议(Service Level Agreement)。在网络发生问题后,服务流量的快速错误恢复不仅在当前是至关重要的,而且在将来的IP和数据传输网络更为必需和必备。现在,在基于MPLS(Multi-Protocol Label Switch)和GMPLS(Multi-Protocol Label Switch)的传输网络中,有关错误恢复机制的研究正在展开。

目前,典型的错误恢复机制有两大类:1+1/1:1/1:N/M:N保护式和预留型。保护式的错误恢复机制在错误发生前就建立了备份路径,一旦错误发生,在工作路径上的流量就能迅速地切换到备份路径上,其主要特点就是错误恢复时间短,但缺点是事先计算并分配资源,从而导致资源严重浪费;预留型的错误恢复机制在错误发生后才开始动态计算备份路径并分配资源,其主要特点就是资源利用率高,但缺点是错误发生后才动态计算路径,导致错误恢复时间长,错误恢复率不高。

鉴于此,本文提出了一种基于分段共享的预测式错误恢复机制(Prognostic Segment-based Sharing Recovery Mechanism—PSSRM),它克服了前述缺点,在最大程度上做到了错误恢复时间和资源利用率的平衡。它需要事先对网络拓扑进行分段,根据分段粒度因子(Segment Granularity Factor—SGF)为每个分段计算备份路径(这个过程可以由网络管理员手动配置也可以使用算法自动由每个Cross Connect——XC计算)。在错误发生后,选择距错误发生点最近的分段备份路径,分配资源并切换工作路径上的流量。

2 基于分段共享的预测式错误恢复机制

这是一种新型的错误恢复机制,在某些方面它的恢复机

制类似于保护式和预留式错误恢复,但是它融合了两者的优点,在最大程度上克服了两者的缺点,如低资源利用率和长错误恢复时间。

在PSSRM错误恢复机制中,错误恢复步骤包括:1)建立工作路径,同时可选地建立备份路径;2)错误发现;3)错误通告;4)错误恢复;5)正常化。

2.1 PSSRM的路径建立

2.1.1 PSSRM的资源分配 在路径(LSP)建立过程中,PSSRM使用RSVP-TE协议。与传统的事先分配资源的保护式错误恢复机制不同的是,PSSRM在建立工作路径之前,需要根据分段粒度因子对整个网络拓扑进行分段,并为其中的预定的链接建立备份路径。因此从某种意义上说,PSSRM的备份路径与工作路径的建立没有什么逻辑上的联系,它们是两个独立的过程。值得注意的是,PSSRM在建立备份LSP时,并不申请分配资源,它需要作的仅仅是为备份路径上的每个LSR(Label Switch Router)标记一下备份路径,也即分配一个标记,从而标示一条备份LSP的存在。下面通过图1来描述整个过程:

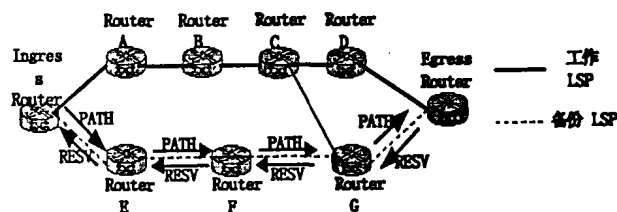


图1 路径建立

当RESV消息沿着备份LSP通过每个LSR时,它将动态刷新每个LSR的软状态,把连接超时计时器清零,同时它申明资源分配为零。这样周而复始的运转保证备份LSP的存在。当错误发生后,PSSRM只需要在RESV中携带要分配的资源,为备份LSP分配资源,等待从工作LSP上切换流量到

^{*}电子科技大学青年科技基金资助项目,编号:YF020802。徐洁 副教授,从事计算机网络、网络体系结构和协议研究。郭星廷 研究生,从事MPLS/GMPLS研究。夏梦芹 博士生,从事计算机网络和通信研究。

备份 LSP。

2.1.2 PSSRM 的资源共享 PSSRM 具有资源共享的能力。因为在 PSSRM 中,工作 LSP 和备份 LSP 的建立过程是两个独立的过程,所以多个工作 LSP 可以共享同一个备份 LSP。一条备份 LSP 可以被共享的数目可以手动配置到 PSSRM 选路算法中。

需要特别指出的是:当一条备份 LSP 的原有申请的带宽无法满足新的工作 LSP 的切换流量带宽时,PSSRM 对这个问题提供了一个很好的解决方案。首先 PSSRM 会试图根据新的切换流量带宽在最近的备份 LSP 上申请新增带宽,同时由于 PSSRM 具有 QoS 机制,它的 Hold-On-Time 约束将对整个错误恢复时间约束,如果申请带宽超时将放弃这条备份 LSP,在次优的备份 LSP 进行考察。如果次优备份 LSP 尚未申请带宽,则申请需要的带宽,如果已经申请带宽并且大于需要的带宽,则切换流量,如果小于需要的带宽,则申请新增带宽,如果还不能在时限内分配资源,则需要到次优的备份 LSP 进行同样考察,依次类推,直到 Hold-On-Time 超时,从而导致此次错误恢复失败,但是从仿真试验看,这种最坏情况发生的几率非常低,因此对 PSSRM 的性能影响甚小。

最后,PSSRM 在多切换流量共享同一个备份 LSP 时采用类似 CSMA-CD 机制,保证每个切换流量能在该备份 LSP 上传输。

2.1.3 PSSRM 的备份 LSP 建立 PSSRM 的备份 LSP 建立在时间上也有其特点,如表1所示。

表1

Category	Calculate Path	Resource Assignment	Cross-Connect
1	Before	Before	Before
2	Before	Before	After
3	Before	After	After
4	After	After	After
1:protection 3:PSSRM	2:pre-planned 4:protection restoration Before: before failure occur After: After failure occur		

从表1中可以看到 PSSRM 是在错误发生之前就已经计算备份 LSP,因此在总的错误恢复时间里减少了备份 LSP 计算的时延。同时 PSSRM 的资源分配在错误发生之后进行,所以能够从总体上增加资源利用率。

PSSRM 的备份 LSP 建立的最大特点就是完全边缘无关(Edge-Disjoint),也就是说,工作 LSP 与备份 LSP 除了入口 LSR 和出口 LSR 相同之外,其他连接均不在同一个物理链接中,如图2所示。

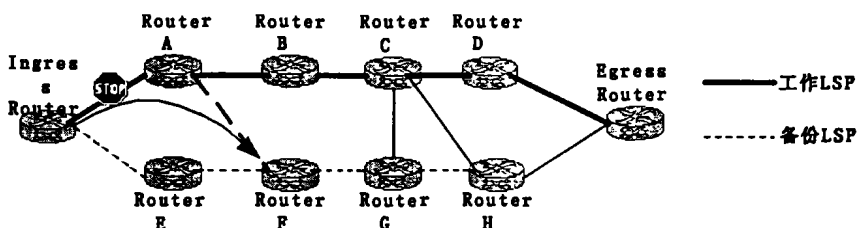


图2 备份 LSP 建立

图2中,工作 LSP 是(Ingress, A, B, C, D, Egress),备份 LSP 是(Ingress, E, F, G, H, Egress),备份 LSP 和工作 LSP 就是完全的 ED,传统的错误恢复机制,如:1+1/1:1/M:N 保护或预留式机制都可能把备份 LSP 选择为(Ingress, A, F, G, H, Egress),一旦 Ingress LSR 到 LSR A 之间的链接出现问题,则备份 LSP 和工作 LSP 都无法工作,因此就降低了错误恢复率。从仿真试验中也可以清楚看到,PSSRM 采用的完全 ED 具有很高的错误恢复率。因此,PSSRM 可增加网络的传输安全性。

2.1.4 PSSRM 的分段机制 在 PSSRM 错误恢复策略中,PSSRM 没有使用端到端(End-to-End)的流量切换机制,相反它采用子路径的机制对一个工作 LSP 进行分段。End-to-End 的方式会增大 LSP 的粒度,增大错误通知时间,同时增加从错误发生时到流量切换之间时段内无法恢复的流量数量,更重要的是浪费中间 LSR 之间正常链接的已分配资源。当错误发生后,PSSRM 不会立即选择从 Ingress LSR 到 Egress LSR 的备份路径,而是采用最短最近备份路径优先的算法。由于在建立工作 LSP 之前已经根据分段粒度因子对整个网络拓扑进行了预测性分段,这正是 PSSRM 中 Prognostic 和 Segment-based 的由来。

2.2 PSSRM 的错误发现与定位

在错误发现过程中,假设 RSVP-TE 协议中两点之间使用 Hello 机制来进行信息交互(Cisco Ltd. 的解决方案之一)。

任一个 LSR 周期性地发送 Liveness 消息到其相邻 LSR 并且等待相邻 LSR 的确认信息。因此一旦一个 LSR 在最大连接时间内没有收到其相邻 LSR 的确认信息,就断定该 LSR 或到该 LSR 之间的链接出现问题。当错误发生后,传输平台将触发控制平台使用 GMPLS 的信令来控制后继操作。错误定位需要节点之间的相互通信从而断定错误的确切位置。PSSRM 错误恢复策略中使用捆绑的泛洪信息传播,当一段物理链路或某个 LSR 出现错误,最近的 LSR 不必逐个对通过错误地点的工作 LSP 的 LSR 发送错误通知信令。它可以使用捆绑的泛洪信息传播方式对错误通知信息进行捆绑,把多个错误通知信息捆绑在同一个泛洪信息中,每到达一个 LSR 时就解复用一次,直到所有的捆绑信息都解复用了,这个捆绑的泛洪信息就结束传输。

2.3 PSSRM 的错误通告

错误通告用来通知一条故障 LSP 上的中间 LSR 上的错误决策实体,在某处发生了错误并且该错误已经被检测到,因此在该 LSP 上的相应业务流量不能继续传输。错误通告信息通过 GMPLS 的控制平台相互传输,GMPLS 的控制平台不一定会采用与故障 LSP 相同的链接。一个发现了错误的 LMP(Label Manage Protocol)节点将会给它的上游节点发送一个 LMP 消息通告这个错误。当一个上游节点接受到这个错误通告信息,它就能够把错误与相应的输入端口关联起来,接下来就可以初始化错误的链接或者启动 LSP 错误恢复程

3. PSSRM 的仿真

我们建立了一个模拟测试环境,其网络拓扑如图4所示。该拓扑包括75个LSR和89条边。仿真试验的目的在于评估资源利用率和错误恢复时间的综合平衡。假设任意地产生LSP建立请求,每条备份LSP的带宽可以申请从10unit到100unit。每条LSP都被设置了QoS约束条件,例如:带宽限制,备份路径的最大被共享数目和最大Hold-On-Time上限时间。工作LSP使用Dijkstra的最短路径优先算法来建立。每条物理链接被分配了100-1000unit的带宽资源。每个链接的传输时延为1ms,流量切换时延为5~10ms,错误发现时间为5ms,这里的假设数据来源于文[15]中的测试数据。

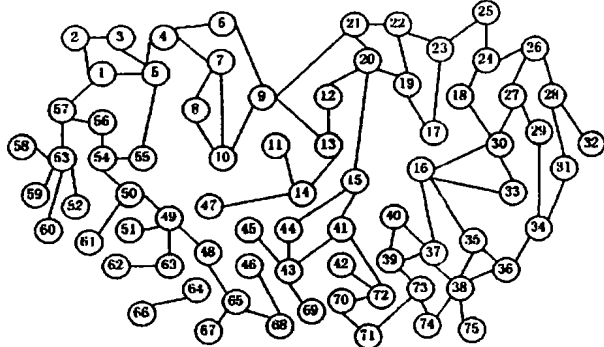


图4 仿真拓扑图

3.1 资源利用率评估

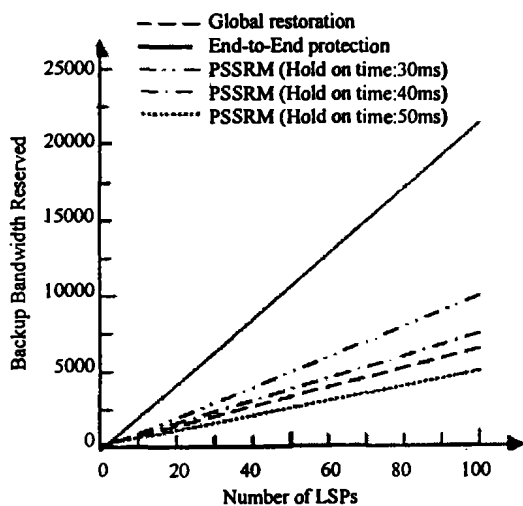


图5 带宽请求比较

从图5中可以看出End-to-End的保护式策略必须在错误发生前分配备份路径的资源,因此它的资源预留最高。对于预留式的错误恢复策略,它浪费了正常的中间节点间已分配的资源,因此其资源预留比采用Hold-On-Time值为50ms的PSSRM高一些,但是比Hold-On-Time值为30和40ms的PSSRM低一些。对于图6,随着Hold-On-Time值的增加,备份路径的预留资源值明显下降,这是因为错误流量可以等待足够时间来共享已经存在的备份路径,从而不需要重新申请新的资源。但是不能说Hold-On-Time值越大越好。这是因为Hold-On-Time的增大是以整个错误恢复时间的增大为代价的,如果Hold-On-Time值达到一定程度,PSSRM就会退化得比纯预留式的错误恢复策略还低效,从而失去算法的先进性。

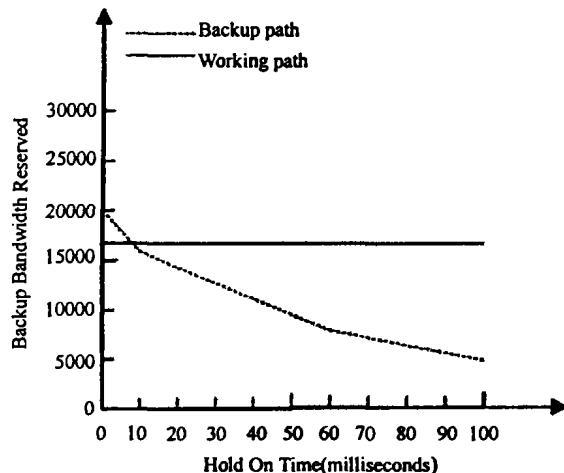


图6 Hold-On-Time 与备份带宽预留

3.2 错误恢复速度和恢复率评估

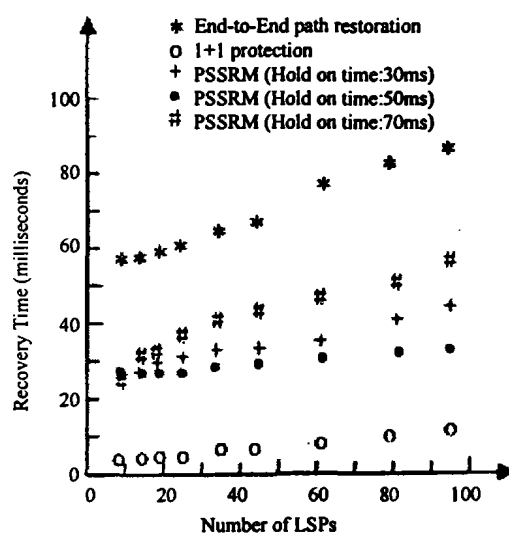


图7 平均错误恢复时间

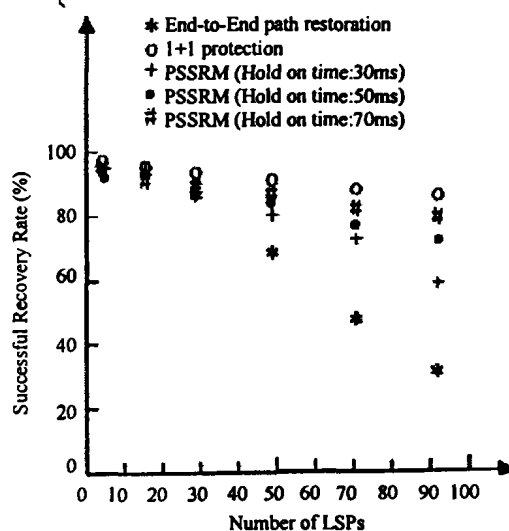


图8 错误恢复率

从图7可以看出1+1保护式的错误恢复策略拥有最短的恢复时间,而End-to-End的预留式错误恢复策略的恢复时间最长。这是因为前者在错误发生前已经计算了备份路径和分

配资源,所以其恢复时间最短;而后者正好相反,在错误发生之后才计算备份路径和分配资源,所以错误恢复时间最长。对于 PSSRM, Hold-On-Time 值越大,错误恢复时间就越大。如果 Hold-On-Time 值大到一定程度, PSSRM 的恢复时间可能比 End-to-End 的预留式恢复策略的恢复时间还要长,在图4中未表示出来。对于错误恢复率,理所当然的是保护式的错误恢复策略最高。随着 Hold-On-Time 值的增大, PSSRM 的错误恢复率将会增加,并且从总体上都高于预留式的错误恢复策略。

结束语 从仿真的实验结果,可以得出结论:PSSRM 的资源利用率、恢复时间和错误恢复率的综合平衡明显优于其他的策略。在本文中,算法使用了如 Hold-On-Time 和带宽等简单的 QoS 约束条件来优化网络的性能。PSSRM 同样可以使用其他的 QoS 约束条件来优化网络。

参考文献

- 1 Banerjee A, et al. Generalized Multi-Protocol Label Switching: An Overview of Signaling Enhancements and Recovery Techniques. IEEE Communication Mag., 2001, 39(7): 144~151
- 2 Mannie E, Papadimitriou D, et al. Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS). work in progress, Internet draft, draft-ietf-ccamp-gmpls-recovery-terminology-02.txt, May 2003
- 3 Mannie E, et al. Generalized Multi-Protocol Label Switching (GMPLS) Architecture. work in progress, Internet draft, draft-ietf-ccamp-gmpls-architecture-02.txt, March 2002
- 4 Sharma V, et al. Framework for MPLS-based Recovery. Internet Draft, draft-ietf-mpls-recovery-framework-03.txt, Feb. 2002
- 5 Shimano K, et al. RSVP extensions for GMPLS restoration signaling. Internet Draft, draft-shimano-imajuku-gmpls-restoration-00.txt, Feb. 2003
- 6 Liu, et al. OSPF-TE Extensions in Support of Shared Mesh Restoration. Internet Draft, draft-liu-gmpls-ospf-restoration-00.txt, Oct. 2002
- 7 Berger L. Generalized MPLS Signaling-RSVP-TE Extension. RFC3473, Jan. 2003
- 8 Papadimitriou D, Mannie E. Analysis of Generalized MPLS-based Recovery Mechanisms (Including Protection and Restoration). Internet Draft, work in progress, draft-ietf-ccamp-gmpls-recovery-analysis-01.txt, Nov. 2003
- 9 Mannie E. Recovery (Protection and Restoration) Terminology for generalized Multi-Protocol Label Switching (GMPLS). Internet Draft, work in progress, draft-ietf-ccamp-gmpls-recovery-terminology-02.txt, Nov. 2003
- 10 Lang J P. Generalized MPLS Recovery Functional Specification. Internet Draft, work in progress, draft-ietf-ccamp-gmpls-recovery-functional-00.txt, July 2003
- 11 Papadimitriou D, et al. Shared Risk Link Groups Encoding and Processing. Internet Draft, draft-papadimitriou-ccamp-srlg-processing-01.txt, May 2003
- 12 Awduche D, Rekhter W. Multiprotocol Lambda Switching: Combining MPLS Traffic Engineering Control with Optical Crossconnects. IEEE Comm. Mag., March 2001
- 13 Czezowski P. Optical Network Failure Recovery Requirements. Internet Draft, draft-czezowski-optical-recovery-reqs-00.txt, April 2003
- 14 Lang J P, et al. RSVP-TE Extensions in Support of End-to-End GMPLS-based Recovery. Internet Draft, work in Progress, draft-lang-ccamp-gmpls-recovery-e2e-signaling-00.txt, Aug. 2003
- 15 Doshi B T, Dravida S, Harshavardhana P, et al. Optical Network Design and Restoration. Bell Labs Technical Journal, vol. January-March, 1999. 58~84

(上接第33页)

存储解决方案,具有高数据传输率、高可用性和可扩展性。VI 结构作为对存储区域网(SANs)体系结构的发展,与传统的结构相比,它在带宽、延迟以及主机负担的处理上更有效。

VI 结构已经越来越被业界所接受。应用以及 Internet 网络的进一步发展将会使这项技术受益。VI 结构不要求 TCP/IP 广域的、分布式的能力就能给所有的应用带来性能与效率上的提高。从理论上来说,所有的拥有数据中心的网络骨干通信都可以从 VI 的实现中受益。

VI 结构是今天高效、高性能的互联技术技术的第一代技术,在接下来的几年里, InfiniBand 结构(IBA)会取代基于总线的互联,而成为主导的互联技术。IBA 是基于光纤的互联技术,几乎具有无限的可扩展性,在高带宽、低延迟的环境下允许多个设备并行通信。IBA 具有 VI 结构的许多特征,如 VI 结构的虚拟通道、远程访问能力等。然而,与 VI 结构不同的是, IBA 互联技术并不依赖于服务器的速度较慢的 PCI 总线结构。

参考文献

- 1 Compaq, Intel and Microsoft Corporations. Virtual Interface Architecture Specification. Version 1. 0, 1997. <http://www.viarch.org>
- 2 Dunning D, et al. The Virtual Interface Architecture. IEEE Micro, Mar. -Apr. 1998. 66~76
- 3 Buonadonna P, Geweke A, Culler D. An Implementation and Analysis of the Virtual Interface Architecture. In: Proc. of SuperComputing Conf. Nov. 1998
- 4 Giganet Inc., cLAN Performance. <http://www.giganet.com/products/performance.htm>
- 5 Dubnicki C, Bilas A, Li K, Philbin J F. Design and Implementation of Virtual Memory-Mapped Communication on Myrinet. In: Proc. of the 11th Intl. Parallel Processing Symposium, April 1997
- 6 Eicken T V, Basu A, Buch V, Vogels W. U-Net: A User-level Network Interface for Parallel and Distributed Computing. In: Proc. of the 15th Symposium on Operating Systems Principles, 1995. 40~53
- 7 National Energy Research Scientific Computing Center(NERSC), M-VIA: A High Performance Modular VIA for Linux. <http://www.nersc.gov/research/FTG/via>
- 8 孙伟平,周敬利,余胜生. 用户层通信接口 VIA 的研究(一). 计算机科学, 2003, 30(11): 125~128