

文章编号:1001-9081(2006)02-0261-04

区分服务网络中基于主动队列管理的病态流控制

葛卫民,舒炎泰,张 亮,高德云
(天津大学 电子信息工程学院,天津 300072)
(gewm@tju.edu.cn)

摘 要:在区分服务(Differentiated Services, DiffServ)网络中,为了消除病态流对确定性转发(Assured Forwarding, AF)服务的不良影响,提出了一种基于RIO(RED IN and OUT)的主动队列管理机制RIO-SD。这种机制不需要在核心路由器维护每一个流的状态,而是通过观测虚拟队列的丢包历史记录来鉴别病态流,并通过虚拟队列的前置滤波器加大病态流的丢包率,实现对病态流的控制。仿真结果表明RIO-SD可以有效抑制病态流对带宽的占用,提高其他正常流的性能。

关键词:区分服务;确保转发;病态流;主动队列管理

中图分类号: TP393 **文献标识码:** A

Controlling ill-behaved flows with active queue management in DiffServ networks

GE Wei-min, SHU Yan-tai, ZHANG Liang, GAO De-yun
(School of Electronic and Information Engineering, Tianjin University, Tianjin 300072, China)

Abstract: A RIO-based active queue management mechanism, RIO-SD (RED IN and OUT with Selective Dropping), was proposed to remove the impact of ill-behaved flows on Assured Forwarding Services in DiffServ networks. Under this scheme, it is not required to maintain per-flow state in core routers, while the ill-behaved flows can be identified based on the drop history of the OUT-profile virtual queue. RIO-SD controls the monitored ill-behaved flows by increasing its drop probability using two pre-filters placed in front of the IN-profile and OUT-profile virtual queues respectively. Simulation results indicate that our approach can improve the performance of other normal flows.

Key words: DiffServ(Differentiated Services); AF(Assured Forwarding); ill-behaved flows; active queue management

0 引言

区分服务(DiffServ)^[1]网络是IETF为解决Internet中的QoS问题而提出的一种网络结构。其基本原理是,在网络边界点对进入网络的业务流进行分类和设置条件,并用区分服务编码点(Differentiated Services Code Point, DSCP)来表示;在网络的核心部分,数据包根据与DSCP相联系的逐跳行为(Per-Hop Behavior, PHB)进行转发,从而实现差别服务。目前,为两类PHBs提供了形式描述:确定性转发(AF)^[2]服务和优质转发(Expedited Forwarding, EF)^[3]服务。其中AF服务是目前讨论最为集中的一种典型服务类型。

在AF中,共有四个业务类型,每个业务类型又分为三个丢包等级(分别用绿、黄、红代表)^[2]。在网络发生拥塞时,低优先级数据包(红)将先于高优先级数据包(绿、黄)丢掉,从而实现多等级服务。由于缓存区管理机制涉及在缓存区满时如何丢弃数据包问题,因而成为影响AF性能的重要因素。虽然在AF机制中并未明确要求缓存区队列类型,但目前普遍采用的是随机早期检测(Random Early Detection, RED)^[4]或其改进算法,如RIO、FRED^[5,6]等。

近来通过对AF PHB的研究,发现了一些影响各业务类

间带宽公平分配的因素^[7,8]。这些因素导致了高负载网络中带宽的不公平分配及对业务的不公平抑制,其中一个关键的原因是存在UDP等非响应业务流与TCP等响应流同属一个AF类的情况。为此,DiffServ工作组建议通过给UDP流分配与TCP不同的丢包优先级来保护TCP流^[9~12]。但是,无论在轻载网络中还是在重载情况下,现有的这种通过将TCP和UDP业务分配到不同丢包优先级上的方法都无法保证它们之间的公平性。目前,在AF PHB中,关于UDP/TCP公平性问题的各种解决方法都集中于惩罚UDP业务流。然而,在同一个AF等级中,在需要保护响应的TCP业务流的同时,应当认识到UDP业务流也需要得到公平对待,因为许多多媒体应用需要使用UDP数据包。只有那些恶意发送大量数据包的高带宽流(我们称之为病态流)需要加重惩罚,这些病态流既包括TCP流也包括UDP流,因为如果一些恶意终端把UDP数据包伪装成TCP数据包,边界路由器是无法鉴别的。因此,在采用一些新的方法时需要综合考虑TCP和UDP的公平性。然而现有的队列管理机制无法有效地控制病态流,我们需要设计新的队列管理机制。

在DiffServ网络中,针对病态流对AF服务的影响,本文在RIO的基础上提出了一种新的主动队列管理机制RIO-SD

收稿日期:2005-08-10;修订日期:2005-10-26 基金项目:国家自然科学基金资助项目(60472078;90104015)

作者简介:葛卫民(1964-),男,内蒙古呼和浩特人,副教授,博士研究生,主要研究方向:计算机网络性能评价、分布式数据库;舒炎泰(1942-),男,江西于都人,教授,博士生导师,主要研究方向:计算机网络性能评价、CIMS、计算机实时控制;张亮(1979-),男,河北武安人,博士研究生,主要研究方向:无线网络的性能评价;高德云(1973-),男,博士,主要研究方向:计算机网络性能评价及仿真。

(在 IN 和 OUT 队列中采用选择性丢包的 RED 机制)。在这种新的缓冲区管理机制中,将核心路由器的 OUT 队列中的高带宽流看作病态流,通过丢包历史来鉴别这些病态流。然后通过提高它们在队列中的丢包概率来进行惩罚,实现对病态流的控制,提高正常流的吞吐量。

1 RIO-SD 的队列结构

在 AF 服务中,DiffServ 网络队列^[2]由 4 个物理队列构成,每个物理队列与一类业务相对应。每个物理队列又由 3 个虚拟队列构成,每个虚拟队列与一个丢包等级对应,如图 1 所示。每个业务类与丢包等级的组合与一个代表一定的服务级别的 DSCP 相对应。实际应用中可不必使用全部队列,而是根据实际情况进行配置。调度算法可采用 WRR (Weighted Round Robin), WIRR (Weighted Interleaved Round Robin), RR (Round Robin), PRI (Priority) 等;队列管理机制可以采用 RED 的各种变形,如 RIO, FRED 等。本文采用了文献[2]中的这种队列结构,但为了便于讨论问题,只设置了两个使用 WRR 调度策略的物理队列,每个物理队列包含两个使用 RIO 队列管理机制的虚拟队列。

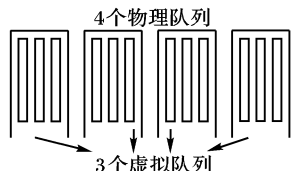


图 1 DiffServ 中的队列结构

RIO 队列管理机制通过设置两个虚队列来提供有差别的服务,一个为“IN”队列,一个为“OUT”队列,每个队列都使用 RED 机制,对应的参数分别为 $(\min_in, \max_in, Pmax_in)$ 和 $(\min_out, \max_out, Pmax_out)$ ^[5]。为了使“OUT”队列的数据包比“IN”队列的数据包更有可能丢弃,通过以下途径实现:

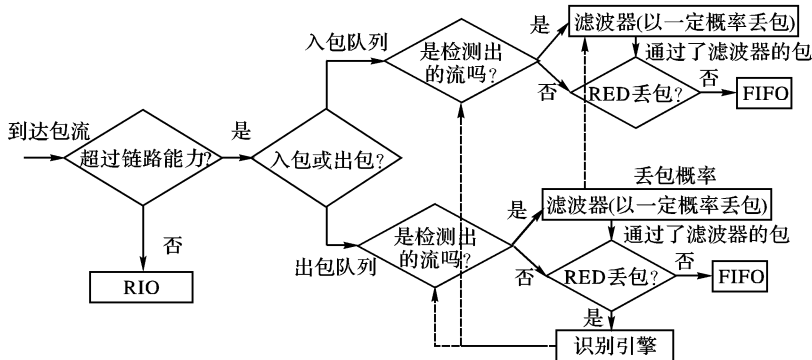


图 2 RIO-SD 控制机制

2.1 病态流的识别

如果路由器没有内存和 CPU 等方面的限制,则可以为每个业务流计算一段时间内的到达速率和丢包速率。这样,路由器可以通过直接测量到达速率的方法来辨识高带宽业务流。另外,路由器也可以测量丢包速率和到达速率,将它们代入到 TCP 吞吐量计算公式中,从而判断是否是病态业务流。然而,在路由器处维护每个流到达速率及丢包率的复杂信息是不必要的,实现也相当困难。Floyd Sally 的研究表明^[15], RED 丢包历史可用于估计流的到达速率和丢包速率。因此,在基于 RIO 队列管理机制的 AF 服务中,“OUT”虚拟队列的 RED 丢包历史即可用于识别病态流。为此,路由器需要维护

设置 \min_out 小于 \min_in ; 设置 $Pmax_out$ 大于 $Pmax_in$, 使得“OUT”队列的数据包的丢弃概率高于“IN”队列的数据包; 另外, 设置 \max_out 小于 \max_in , 使得“OUT”队列的数据包进入拥塞控制阶段要远远早于“IN”队列的数据包。

在 DiffServ 结构中, 每一个用户与一个服务级别协议 (Service Level Agreement, SLA) 关联, SLA 是用户与 ISP 之间关于用户能够获得的转发服务的一种合同, 也约定了用户的业务特征。在 RIO 机制中, 只考虑两类包: 入包 (IN packets) 和出包 (OUT packets)。入包代表了符合约定的包, 出包代表了不符合约定的包。如果一个业务流属于 SLA 所规定的业务, 边界路由器将该业务流的数据包标记为 IN 包, 并在“IN”队列中排队; 而将不属于 SLA 所规定的业务的包标记为 OUT 包, 并在“OUT”队列中排队。当链路发生拥塞时, 处于“OUT”队列中的高带宽流是引起拥塞的一个主要原因, 我们将其视为病态流。这些业务流在路由器损害了正常的业务流而耗用了大量的带宽, 如果能够对这些业务流提前进行识别并进行有效的控制, 将可以有效地提高其他正常流的性能, 保证各种业务流的带宽分配公平性。因此我们用基于 RIO 算法的队列管理机制 RIO-SD 来实现对病态流的控制。

2 RIO-SD 的实现

在 best-effort 网络中, 已有一些方法控制高带宽流来提高其他业务流的吞吐量^[13,14]。我们借鉴 Floyd Sally 的 RED-PD (使用优先级丢包的 RED)^[14] 方法, 提出了 RIO-SD 控制机制, 如图 2 所示。RIO-SD 分别在“OUT”、“IN”队列前设置滤波器, 一个识别引擎设置在“OUT”虚拟队列上用以识别病态流。所有被检测出的病态流都经过这些滤波器, 以一定的概率丢包, 然后放入输出队列, 未被检测出的流直接放入输出队列。

该机制包括 3 个关键部分: 病态流的识别、病态流的控制、前置滤波器的丢包概率设置依据和计算方法。

M 个表格 (我们称之为丢包历史记录表), 每个表格记录着一段时间范围内的业务流的丢包信息。需要说明的是, 每隔一段时间 T , 都将建立一个新的丢包历史记录表, 同时释放掉当前时间最远的表格。例如, 当开始进入 $(M+1)T$ 时间范围内, 就释放掉 $[0, T]$ 时间范围内的丢包历史记录表, 创建 $[(M+1)T, (M+2)T]$ 时间范围内的丢包历史记录表。

丢包是到达业务流的无偏差采样值。当一个流在“OUT”虚队列的总共 M 个丢包历史记录表中有 K 个含有丢包, 则认为这个流是病态流。如果这个流被识别为病态流, 则被标记为“检测出 (Monitored)”的流。这里 M, K, T 是可以事先确定的常量。

2.2 病态流的控制算法

在 RIO-SD 中,当数据包到达路由器时,处理步骤如下:

第一步:计算 RIO 队列的加权平均队长 avg_total 之和

$$avg_total \leftarrow (1 - w) * avg_total + w * q_inst$$

其中: q_inst 为队列实际长度, w 为权重,是一个事先确定的低通滤波器时间常数。在确定权值 w 时应仔细考虑其是否满足要求,这需要折衷。如果 w 值太大,平均过程将不能在网关滤掉瞬时出现的拥塞;如果 w 值太小,流的平均丢包数对实际的变化情况反映太慢,此时,路由器就不能检测到拥塞。根据实际的仿真实验来看,我们认为 w 值在 0.2 ~ 0.4 时能有效地计算出流的平均丢包数。

第二步:判断业务量是否超过链路能力

1) 如果 avg_total 小于预先设定的阈值 $Lmaxthresd$,则认为业务量未超过链路能力,该数据包进入队列,并按通常的 RIO 队列进行处理;

2) 否则,认为业务量超过链路能力。

第三步:识别病态流并进行控制

设 fid 为到达数据包的流号,判断流号为 fid 的流是否为被检测出的病态流,若不是被检测出的病态流,则该数据包进入队列,并按通常的 RIO 队列进行处理。

若是病态流,则分别进入“IN”和“OUT”队列前面的过滤控制器,以概率 P_a 丢弃(在不同虚队列前的过滤控制器其概率不同)。在过滤器中存活下来的数据包,进入队列,按通常的 RIO 队列进行处理。

其中如何确定“IN”和“OUT”队列前面的过滤控制器的丢包概率是实现 RIO-SD 算法的关键。下面将说明他的设置依据和计算方法。

2.3 过滤器中丢包概率设置依据

某个流在“IN”队列前的滤波控制器中的丢包概率由该流在“OUT”队列前的滤波器中的丢包概率决定,这是因为我们需要根据“OUT”队列的丢包信息判断病态业务流对网络的负面影响程度。我们设定两者之间的比率等于前者的丢包等级与总的丢包等级数的比率(当然,不仅仅限于 2 个虚队列)。例如:如果在同一业务类中有 3 个虚队列,我们首先计算最低丢包等级的丢包概率 P ,则中间级别的概率为 $\frac{2}{3} * P$,而最高丢包等级的丢包概率为 $\frac{1}{3} * P$,在我们的仿真实验中,由于只研究两个虚队列的情况,所以高丢包级别的丢包概率为低丢包级别的丢包概率的一半。这个比率可根据实际网络环境来设定,如果网络管理者想要加重惩罚病态流,则可以将高丢包级别的丢包概率设得大一些。

病态流丢包概率的每次增量由两个因素决定。第一个因素是队列的环境丢包率。所谓“环境丢包率”是指队列的丢包率,不包括滤波器处的丢包。当环境丢包率较高时,路由器应快速降低病态流的速率,故病态流的丢包概率增大的增量应该较大。第二个因素是该流的发送速率。对于发送速率较大的病态流,其丢包概率的增量应较大以使其快速回到正常范围。根据以上分析,病态流的丢包概率的增量应等于 $\frac{\text{丢包数}}{\text{平均丢包数}} * P_{total}$,“丢包数”是该流在检测周期($M * T$)丢包的数目,“平均丢包数”是在检测周期内所有被检测出病态流的丢包数的均值, P_{total} 是输出队列丢包率。如果一个已检测

出的流在最近的 M 个表中丢包为 0,则其丢包概率减半。同时我们为避免震荡也设置了单步最大减少量和最大增量。

RIO-SD 机制也保证了被检测出流之间的相对公平性。每个病态流的丢包概率都正比于其到达速率超过目标速率的部分,而这个量由“OUT”队列的丢包历史给出。

2.4 过滤器中丢包概率的计算方法

根据上述分析确定各个虚队列的过滤控制器的丢包概率 P_a 计算公式为:

$$P_a \leftarrow (Prec / Numprec) * P \quad (1)$$

其中 $Prec$ 为该数据包的丢包优先级号, $Numprec$ 为队列中丢包等级数目, P 为流 fid 的队列过滤控制器丢包概率。为便于说明,这里先介绍一下将要用到的量: M, K, T 为定义的常量; P 为业务流经过过滤控制器时的丢包概率; $max_decrease$ 为设置的常量,用来避免 P 一次变化太大; P_{min} 为丢包概率最小阈值; $dropf$ 为该流在当前的丢包历史记录表中丢包的个数, avg_drop_count 为 M 个丢包历史记录表中所有被检测出的病态流的平均丢包数, P_{total} 为输出队列的丢包率, P_{delta} 为丢包概率增量。

下面介绍流 fid 的队列过滤控制器丢包概率 P 的具体计算过程:

当数据包由于链路能力不足被丢包或 RIO 丢包(不包括上述过滤控制器丢包)时,更新当前的丢包历史记录表,将该流所对应的丢包数 $dropf$ 增加 1。

如果当前的丢包历史记录表的创建起始时间与当前时间之差已大于或等于 T ,则检查最近的 M 个历史记录表:

1) 对于每一个目前是被检测出的病态流,如果它在最近的 M 个历史记录表中都未出现丢包(只要某个丢包历史记录表格中记录的流丢包数大于等于 1,就认为对应的时间范围内出现丢包),则:

若 P 大于两倍的 $max_decrease$,则该流的过滤控制器丢包概率减小为:

$$P \leftarrow P - max_decrease \quad (2)$$

否则该流的过滤控制器丢包概率变减小为:

$$P \leftarrow P/2 \quad (3)$$

若该流的过滤控制器丢包概率小于 P_{min} ,则释放该流,即该流不再是被检测出的病态流。

2) 对于每一个在最近的 M 个丢包历史记录表中的 K 个或 K 个以上丢包历史记录表中都出现丢包的流:

如果该流不是被检测出的病态流,则 $P \leftarrow 0$ 。

$$\text{否则 } P_{delta} \leftarrow (dropf / avg_drop_count) * P_{total} \quad (4)$$

若 $P_{delta} > P + P_{total}$,则:

$$P_{delta} \leftarrow P + P_{total} \quad (5)$$

这样就可得到该流的过滤控制器丢包概率为:

$$P \leftarrow P + P_{delta} \quad (6)$$

3 性能评价

我们使用包含 DiffServ 模块的 NS-2^[15,16] 来实现仿真实验。为在 NS-2 中设计实现 DiffServ 结构,在类的层次结构中增加了 5 个模块:一个用于基本的 DiffServ 路由器的功能(dsRED),一个用于基于 RED 的排队机制,一个用于与策略有关的功能,另两个分别用于边界路由器及核心路由器。我们在基于 RED 的排队机制模块中实现了 RIO-SD 机制。

3.1 仿真实验的网络拓扑结构

图3是仿真实验所采用的网络拓扑。我们使用了3对源(S0, S1, S2)和目的(D0, D1, D2)节点,它们之间由DiffServ网络连接。DiffServ网络由两个边界节点(E1, E2)和一个核心节点(C)构成,除核心节点C与边界节点E2间的链路外,网络中各链路设置相同。

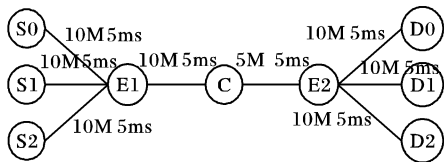


图3 仿真实验的网络拓扑结构

在源和目的对之间建立了多种交互。S0和D0之间由10条1类TCP流, S1和D1之间有另外10条2类TCP流,这些流的上层应用为FTP。此外,在S2和D2之间又一条使用UDP的2类CBR流代表病态流。CBR流的初始发送速率为1Mbps,在100s时变为5Mbps,在200s时又变成1Mbps。

3.2 性能指标和仿真结果评价

为分析仿真数据和性能评价,选择下列性能指标:

吞吐量:测量时间间隔内接收到的总字节数/测量时间。分别计算了与高优先级和低优先级对应的所有TCP流的总吞吐量。

丢包率:仿真期间属于CBR/UDP流的每秒丢包数。

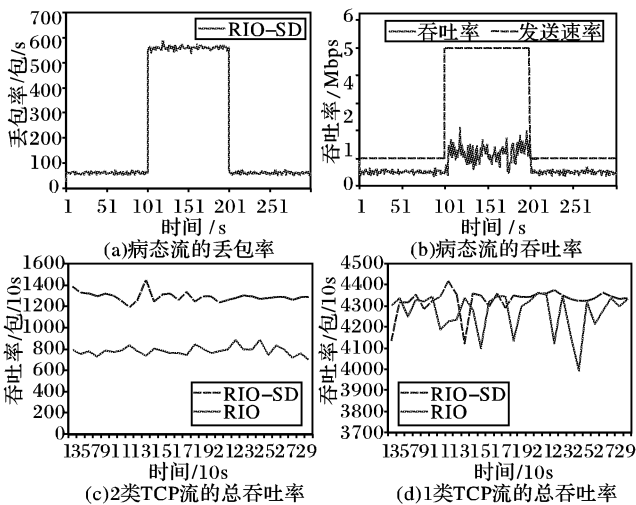


图4 仿真实验结果

通过图4的仿真结果对RIO-SD与RIO进行比较。在这些图中,“RIO-SD”代表我们提出的机制,“RIO”代表通常的RIO机制。从图4(a)中可以看出,当病态流的发送速率快速增大时,其丢包率也迅速增大。图4(b)表明,尽管病态流的发送速率增长很大,但是病态流的实际吞吐量增长却很小。因此,其他正常流的性能得到了保护。这说明我们的机制是有效的。

从图4(c)和图4(d)可以看出,其他数据流的总吞吐量改进了,特别是与病态流属于同一业务类(2类)的TCP流的吞吐量有了明显的改进。这主要归功于病态流得到了有效控制,病态流占用的资源得到了释放,其他正常流获得了更多的资源。

4 结语

病态流对DiffServ网络中AF服务有严重的影响。为了

减少这种影响,借鉴RED-PD的主动队列管理机制的思想,在DiffServ网络中的核心路由器上实现了一种基于RIO的缓冲区队列管理机制RIO-SD。使用该机制,不需要在核心路由器维护每个流的状态,而是根据“OUT”虚队列的丢包历史数据来鉴别病态流,同时在“IN”、“OUT”虚队列前分别设置两个滤波器,通过加大病态流的丢包率来控制鉴别出的病态流。仿真结果表明RIO-SD可以有效地控制病态流,提高其他正常业务流的性能。该机制实现简单且对网络业务具有一定的鲁棒性。

参考文献:

- [1] BLAKE S, BLACK D, CARLSON M, *et al.* An Architecture for Differentiated Services. IETF RFC 2475[S]. December 1998.
- [2] HEINANEN J, BAKER F, WEISS W, *et al.* Assured Forwarding PHB Group. IETF RFC 2597[S]. June 1999.
- [3] JACOBSON V, NICHOLS K, PODURI K. An Expedited Forwarding PHB. IETF RFC 2598[S]. June 1999.
- [4] FLOYD S, JACOBSON V. Random Early Detection Gateways for Congestion Avoidance[J]. IEEE/ACM Transactions on Networking, 1993, 1(4): 397-413.
- [5] CLARK D, FANG W. Explicit Allocation of Best-Effort Packet Delivery Service[J]. IEEE/ACM Transactions on Networking, 1998, 6(4): 362-373.
- [6] LIN D, MORRIS R. Dynamics of Random Early Detection[A]. Proceedings of ACM SIGCOMM[C]. New York: ACM Press. 1997, 27: 127-137.
- [7] SEDDIGH N, NANDY B, PIEDA P. Bandwidth Assurance Issues for TCP flows in a Differentiated Services Network[A]. IEEE GLOBECOM'99[C]. 1999, 1792-1798.
- [8] JAIN R, LIU C, GOYAL M, *et al.* Performance Analysis of Assured Forwarding. IETF DRAFT[S]. February 2000.
- [9] ELLOUMI O, CNODDER DS, PAURWELS K. Usefulness of Three Drop Precedence in Assured Forwarding Service. IETF DRAFT[S]. July 1999.
- [10] SEDDIGH N, NANDY B, PIEDA P. Study of TCP and UDP Interaction for the AF PHB. IETF DRAFT[S]. September 1999.
- [11] GOYAL M, DURRESI A, JAIN R. Effect of Number of Drop Precedence in Assured Forwarding. IETF Draft[S]. June 1999.
- [12] WOOD L, ANDRIKOPOULOS I, PAVLOU G. A fair traffic conditioner for the assured service in a differentiated services internet. Proceedings of ICC 2000[DB/OL]. <http://lib.nau.edu.ua/acm/disk2/db/conf/icc/icc2000-2.html>, 2002-01-16.
- [13] CLAFFY K, MILLER G, THOMPSON K. The Nature of the Beast: Recent Traffic Measurements from an Internet Backbone[EB/OL]. <http://www.caida.org/outreach/papers/1998/Inet98/Inet98.pdf>, 1998.
- [14] MAHAJAN R, FLOYD S, WETHERALL D. Controlling High Bandwidth Flows at the Congested Router[A]. Proceedings of ICNP'01[C]. Washington: IEEE Computer Society, 2001. 192-201.
- [15] Thenetworksimulatorns-2.26[EB/OL]. <http://www.isi.edu/nsnam/ns/>, 2003-02-28.
- [16] PIEDA P, ETHRIDGE J, BAINES M, *et al.* A Network Simulation: Differentiated Services Implementation[EB/OL]. <http://www7.nortel.com:8080/CTL/>, 2000-07-06.