

概述

带宽限制在应用性能中一直扮演着一个非常重要的角色。多数网络分为两类。第一类为诸如拨号和帧中继等慢速网络。这些网络通常速率为 56 kbps 到 2 Mbps，并有许多节点。然而这种网络无法适用于拥有大型办事处的企业，这种数据速率通常适用于小型分支机构之间的网络连接。第二类（广域网）为高速网络，通常用于企业大型办事机构之间的数据复制与通讯。与小型办公网络不同，这些网络通常速率为从 45 Mbps 到 622 Mbps 之间。

尽管随着时间的推移，网络已经不断变化，但应用业务已经以令人惊讶的速度增长。可节约带宽的客户端服务器应用已被带宽需求型 web 应用所替换。从前，客户端服务器交易包括数千字节的数据，如今，基于丰富 Web 的门户应用在每次进行交易时，也能传输数千节的数据。电子邮件附件、可访问的远程文件共享等应用已大量增长。由于要求复制的数据的数量增加，即使是在专用高速链路中进行数据复制，也会面临带宽上的挑战。

对于慢速和高速网络而言，提供额外带宽以满足不断增加的需求通常会造价昂贵。带宽的价格尚未像预期中的那样下降，网络尚无法跟上应用需求发展的步伐。

挑战

网络压缩

此前尝试在网络级别进行压缩已被证明不太成功。多年来，路由器号称具备压缩能力，然而很少有哪家企业能够实现这一功能。原因在于，这种压缩通常会使得路由器自身增加额外负载进而增加开销，此外，由于路由器需花费时间压缩每个数据包，因此还会导致额外的延迟。

部署一套最佳压缩方案要求不仅有一个好的算法，而且架构设计还应实现最大效率和最佳性能。压缩效率中的一项关键要素是，数据如何呈现给压缩程序。所有压缩例程在处理同类数据时将获得更大的压缩比。在处理异质数据时（例如，多种协议的大量数据包），压缩比率会大大降低。

基于数据包压缩应用的主要问题是压缩时它将多种数据类型混合在一起。基于数据包的压缩系统会存在其它问题。压缩数据包时，这些系统必须在网络中编写小数据包，并进行其它工作以集合并封装多个数据包。每一操作都会达到最佳效果。在网络中编写小数据包会增加 TCP/IP 标头的开销。另外，集合并封装数据包会为该流增加封装标头。

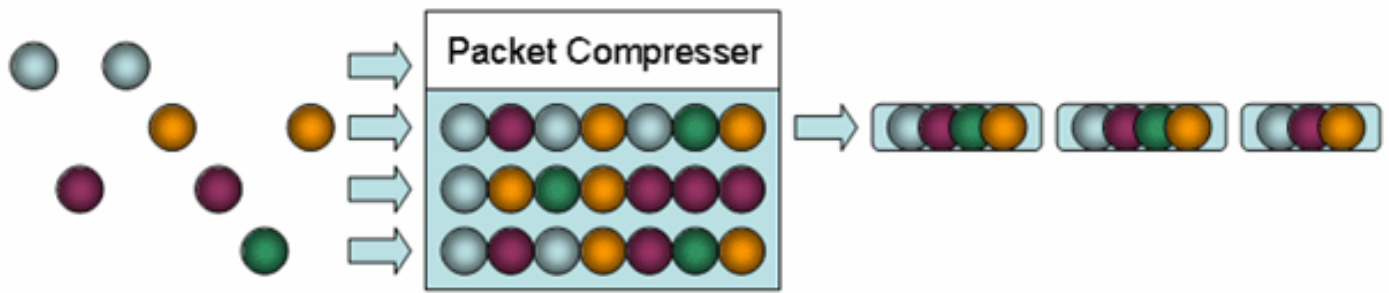


图 1：基于数据包的压缩

与以前的压缩解决方案不同，WANJet 在会话层中运行。这会支持 WANJet 在处理所有应用类型时能够在完全同类的数据之间进行压缩。WANJet 与同档基于数据包的系统相比压缩比率会更高。

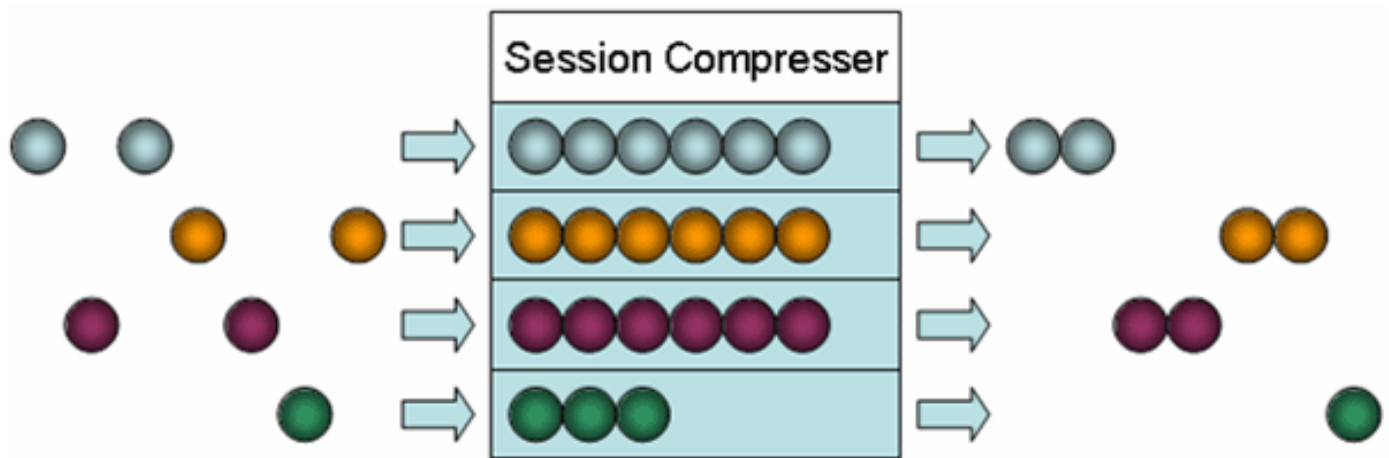


图 2：基于会话的压缩

此外，通过在会话层、数据包边界位置进行操作，重新分组的问题就不会出现。这能够使 WANJet 在数据流中轻松找到匹配的数据，在第三层中这些数据可能是许多分开的字节，但在第五层中可能就是邻近的字节。由于取消了封装阶段，因此在会话层中执行压缩时系统的吞吐率会增加。

获得较高的压缩比仅仅是求解性能困局的其中一项措施。为了提高性能，压缩器必须能增加网络的吞吐率。这就要求压缩器获得比线速吞吐率更大的性能。试以如下两个理论中的压缩设备为例。压缩器 A 能够实现 75% 的数据压缩，每秒可读取 20 Mbps 压缩数据。压缩器 B 的速度是 A 的两倍，但只能读取 50% 的压缩数据。

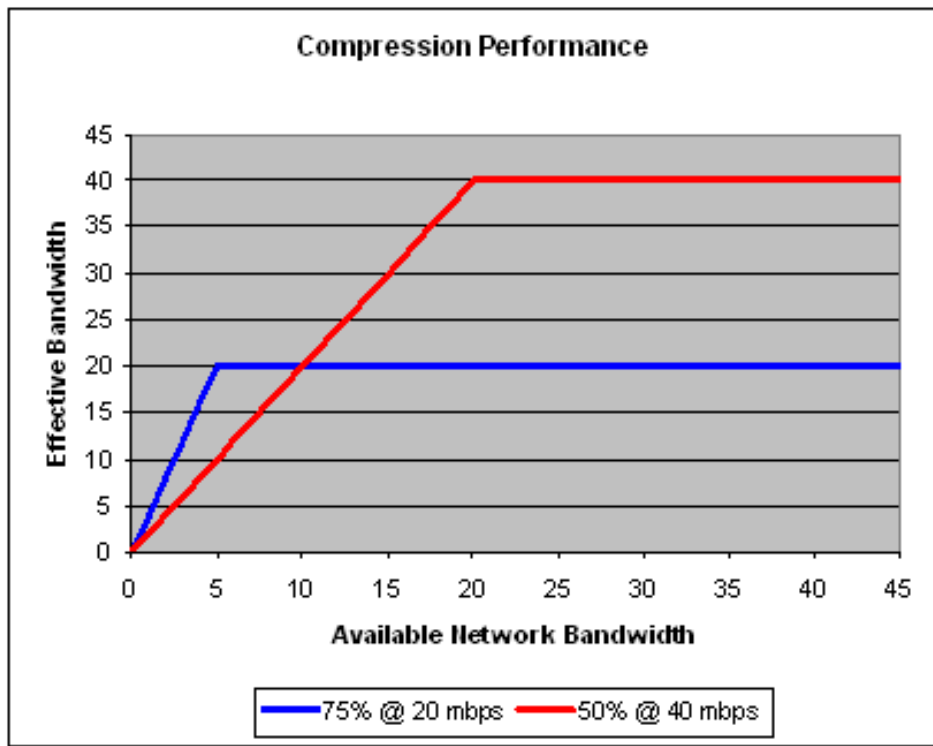


图 3: 压缩性能图

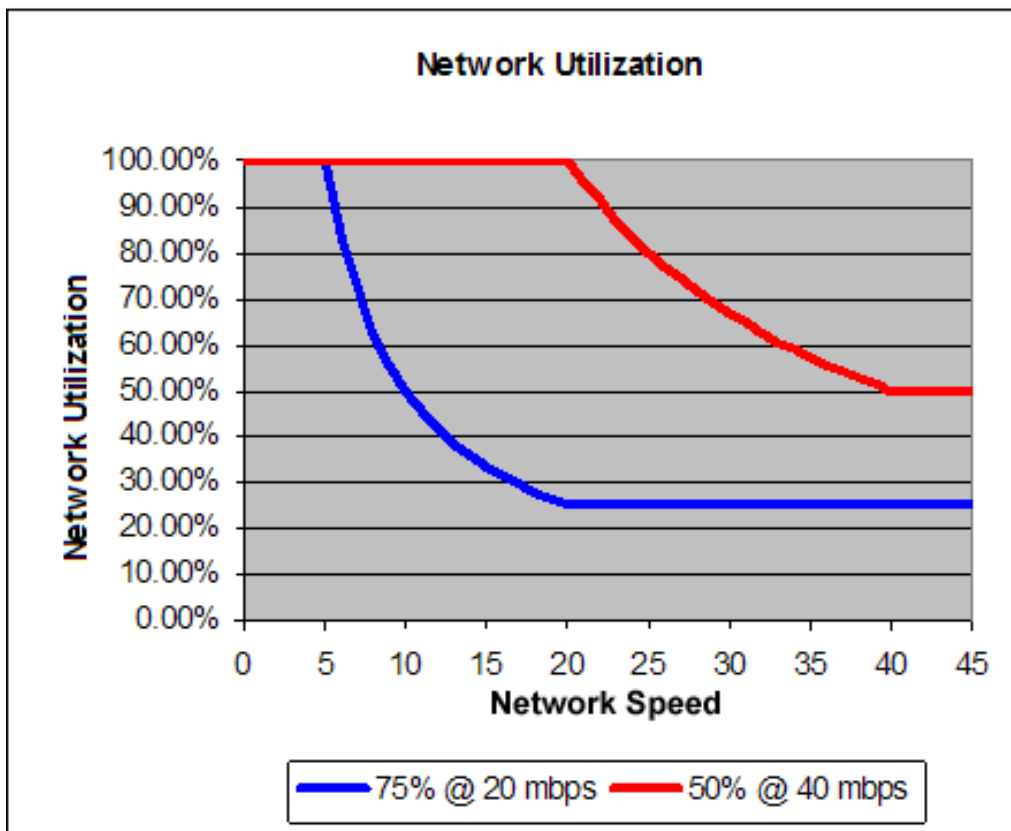


图 4: 网络利用率

初看，压缩器 A 似乎是一台更加高效的压缩设备，而实际上，压缩器 B 能够获得更出色的网络性能，其链路速率大于 10 Mbps。原因即在于，当网络速度增加时，压缩器 A 不能充分利用可用的带宽。

解决方案

透明数据压缩

F5 WANJet 产品充分利用一种称之为透明数据压缩 (TDR) 的技术来迎接带宽挑战。与以前的压缩形式不同, TDR 充分利用两阶段压缩流程实现带宽的节省, 同时还能降低处理过程中的延迟。流程的第一阶段 (TDR-2) 用于检测传输数据, 确定其中任意部分之前是否已发送。如果已发送, 相关内容会替换掉先前传输的区域。通过采用基于字母的压缩以及高级编码方案, 流程的第二阶段 (TDR-1) 可对数据进行进一步压缩。



图 5: 透明数据压缩

TDR-2

TDR-2 所采用的数据压缩程序, 可用于识别并删除所有 WAN 中所有重复的数据模式。当数据流通过两个 WANJet 装置时, WANJet 会记录字节模式并构建同步字典。如果同样的字节模式再次通过 WAN, 发送方旁边的 WANJet 就会用参考模式来替换字节模式, 并发送至字典中的副本。当该参考模式到达远程 WANJet 时, 参考模式就会被字典中的初始数据替换, 最终获得的数据流与最初发送的数据流将相同。

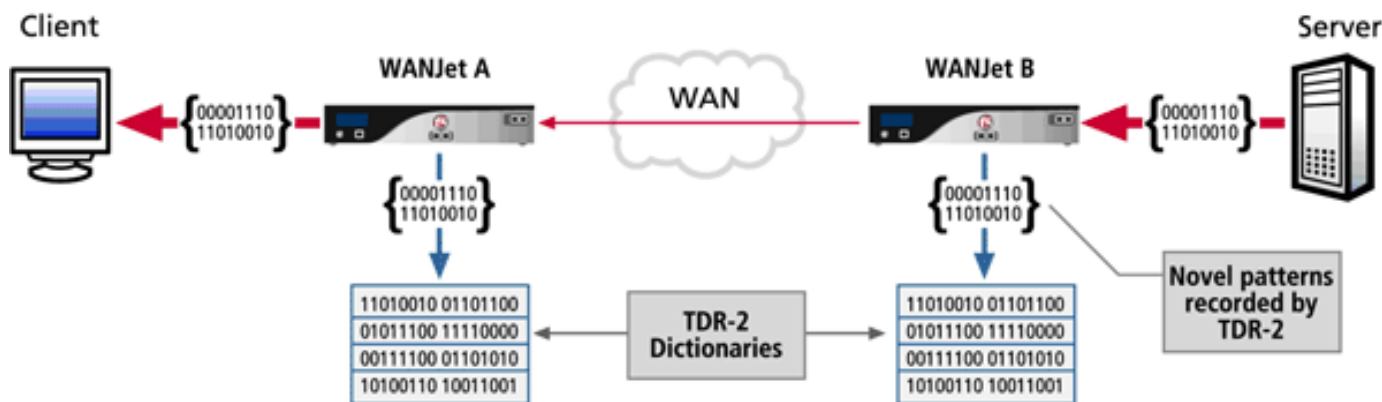


图 6: TDR-2

存储于 TDR-2 字典中的数据, 存储于应用协议和传输独立格式的地址当中。这意味着, 如果某字节模式在一个协议中首次

出现，之后在另一个协议中再次出现，第二次传输将从上一次传输中完全受益。即使第一次传输为下载、第二次传输为上传（例如，用户下载邮件中的附件，然后上传到 Windows 文件共享系统上），上述情况亦有可能发生。由于 TDR-2 可识别字节模式，并且不受不同协议和传输地址的影响，因此可充分利用初始的电子邮件 (MAPI) 传输，提高后来 Windows 文件共享传输的效率。

文件高速缓存

- 可能失效的数据
- 访问控制/安全受到威胁
- 特殊协议（如 HTTP）
- 文件名称依赖性
- 文件编辑传输整个文件

TDR-2

- 保证获得精确的数据
- 访问控制/安全受到威胁
- 与协议无关
- 文件名独立性
- 文件编辑仅传输新数据

由于 TDR-2 可搜索数据中的重复模式，因此，改进重复发送同样数据的文件传输（如 CIFS、电子邮件附件、FTP 以及应用协议）是一种非常理想的方法。与高速缓存技术不同，TDR-2 能够确保服务器收到全部交易并完全进行处理。这就使现有的安全检查能够顺利实施，客户端和服务端能够正常运行。

TDR-2 与另一种模式下的高速缓存技术不同。有了 TDR-2，就不会存在数据失效问题。借助传统的基于时间的高速缓存技术，对象会按一段预定义的时间存储。如果在此之前，服务器上的对象变更消失，高速缓存就会存储失效的数据。借助 TDR-2，所有的交易将由服务器来完成，所存储的数据仅用于减少 WAN 中传输字节的数量。如果服务器端数据变更，新的数据会在达到客户端期间记录至字典中。

对于曾做过修订的文件而言，这一方法可实现部分加速功能。借助一流的高速缓存技术，任何对文件的更新要求下载全部新的版本。由于 TDR-2 基于字节而非文件，因此只有文件新变化的部分 (novel portions) 才会被传输。

如前所述，仅靠获得较高的压缩比并不足以提升应用的性能。压缩系统还必须要有较大的吞吐率。由于支持 400 Mbps 以上的吞吐率，WANJet 400 能够确保数据的压缩转变了实际的性能增益，而不仅仅是空的 WAN。

TDR-1

在 TDR-2 已删除所有先前传输的字节模式之后，WANJet 会采用次一级的数据压缩程序（该程序被称为 TDR-1）。同时，可对 TDR-2 进行优化以提升重复传输的性能，通过采用高级编码技术，以及针对非常小的重复模式进行过优化的字典技术，TDR-1 可改进首次传输的性能。

压缩技术可部署于全部为开或关模式的路由器和其它网络设备中。为了减少拥塞，网络管理员必须向所有流量添加额外的处理延迟。对于诸如 HTTP 等协议，采用针对一定拥塞级别的压缩行为至关重要。与多数协议不同，HTTP 具有交互以及批量数据传输特征。当用户与 web 应用交互时，多数 HTTP 数据交换包含大量的小型数据的传输。然而在下载文件时，只有单一的传输发生，数据传输量通常为 5 MB 或更多数据。网络行为的这一变化给我们提出了一个非常有趣的挑战。在进行网页浏览时，至关重要的一点是，压缩程序采用最少数量的额外延迟。即使向每次交易增加几毫秒，也会降低一些应用的性能。同时，通过 HTTP 进行的较大文件的传输（如文件下载），通常会从压缩技术中受益颇多，因为这种传输时间上的改进会使额外处理上的延迟变得毫无意义。

通过适应不断变化的网络环境与应用需求，TDR-1 可解决这一问题。在具有较高拥塞率的时段，TDR-1 可提高压缩级别、降低拥塞和网络队列的延迟。在具有较低拥塞率的时段，TDR-1 可降低压缩级别、使压缩导致的延迟降至最低。借助 TDR-1 的适应性特性，可确保采用经过优化的压缩战略，并支持网络管理员对压缩进行部署，而不必担心应用性能的下降。

除了可提高应用的性能之外，TDR-1 也可对配置进行简化。与其它系统要求深刻了解压缩吞吐率和延迟特征不同，TDR-1 能够自动选择与网络环境相符的适当战略，并可实时更新战略。借助其智能特性，TDR-1 能够从较低速率的 64 Kbps 帧延迟网络，扩容至 155 Mbps OC3 网络，同时可优化范围广泛的各类协议。

结论

通过将 F5 WANJet 中的 TDR 部署与适应性压缩及数据压缩技术相结合，这一基于特定目的构建的架构可实现应用的最大性能。TDR 通过两个流程步骤实现这一优化。首先，以小参考模式识别并替换冗余模式至远程 WANJet 数据存储位置。其次，余下的网络数据可通过智能网络和应用程序（这些程序能够以尽可能少的字节对剩余数据进行最优编码）进行压缩。

所有上述举措最终将导致下列情形：解决方案于 WAN 之上，提供像 LAN 一样的应用性能，文件传输、电子邮件、客户端服务器应用、数据复制，以及其它应用的速度将加快，同时还能为广域网用户提供可预期的快速性能。