riverbed
Think fast.™

COMPETITIVE BRIEF

Riverbed vs. Expand Networks

Expand Networks Overview

Founded in 1998, Expand is a long-time vendor in the network compression and QoS market. In mid-2006, Expand acquired DiskSites, a small vendor of a file caching solution that had very few customers. Expand's WAN optimization product today is a hybrid device that combines memory-based compression and QoS component with a file caching component – two legacy approaches lacking in performance, scalability and ease-of-use when compared with the newer disk-based data reduction and non-caching single-copy acceleration used by Riverbed Steelhead products.

After 13 years of existence, Expand remains a private company, whose financial status, including balance sheet and income statement are all unknown, unverified, and unaudited. In contrast, Riverbed is a public company, with freely available financial statements showing profitability, growth, and a strong balance sheet.

Riverbed Advantages

- Single integrated product delivers disk-based bandwidth reduction and true application acceleration
- Avoids repetitive transfers of data in different files and objects – even when transferred days apart and/or through different application protocols
- Steelhead Mobile software client brings full acceleration benefits to mobile workers and micro-branches
- RiOS Services Platform enables the delivery of branch office services without the need to deploy a full-blown server
- Rich set of additional, application-specific optimizations including Lotus Notes, multiple MAPI versions, MS-SQL, and HTTPS
- Auto-discovery allows easy deployment – no tedious configuration of tunnels and file caches
- Leverages end-to-end file locking and data protection; no file-caching data integrity risks

Expand lacks disk-based data reduction – the foundation of WAN optimization

While the Expand product has disks, it only uses them for file and web caching, and not for compression or byte-level data reduction. The Expand device's only compression capability is via a memory-based mechanism. Of course, the amount of compression history that can be kept in RAM is orders of magnitude smaller than when using a disk-based dictionary; memory-based compression can therefore only suppress repetitions occurring across very short intervals.

However, many repetitions in enterprise traffic occur over long windows of days or weeks, and over separate application sessions. As an example, a user may receive a bulky 5 MB email attachment one day, save it to a CIFS share the next day, and send it as an email attachment the following week – with multiple gigabytes of additional, unrelated traffic traversing the WAN in between each of these operations. In such a scenario, while the same data traverses the WAN multiple times, the repeated appearances of those data patterns cannot be suppressed by a memory-based compression approach such as that used by Expand.

However, the repeated appearances of these data patterns will be recognized and suppressed by Riverbed's disk-based data reduction mechanism, which stores data persistently for long periods of time (days, weeks, or months) using a disk-based data store that holds hundreds of GB of data. Furthermore, should some variation of the original data be sent over the WAN (e.g., the file was edited before being re-sent), Riverbed's data reduction recognizes the changes and sends only portions that are different from the original data.

No auto-discovery: deploying Expand is a manual and configuration-intensive process

Unlike Riverbed Steelhead products, Expand devices do not have auto-discovery: they cannot automatically find each other in the course of normal client/server connection setup. Also unlike Riverbed, Expand devices also cannot automatically discover all servers. Due to the

lack of an autodiscovery mechanism, Expand devices require significant amounts of configuration, particularly when they are deployed in larger networks. IT personnel considering a non-trivial sized deployment with Expand can be sure that they will have to consume significant staff resources planning, configuring, and maintaining the deployment. Below are some examples of configuration tasks that are manual with Expand, but are entirely automatic with Riverbed.

- **Optimization peering relationships**: Each Expand device must be manually configured with the IP addresses of all other Expand devices that it will peer with to optimize traffic. The amount of available bandwidth to each peer must also be entered. For a large network with many sites, the amount of work and potential for error are significant, particularly for full-mesh MPLS-type networks, where n*(n-1) tunnel configurations must be manually performed.

- **Optimization tunnels**: Expand devices must be explicitly configured with all destination subnets for which they may optimize traffic. In theory, Expand devices can be configured in a way to learn this information automatically. But this requires participating in routing protocols, which is unacceptable to most network administrators because of the significant risks involved. As a result, entering subnets is usually a manual process – each and every routed subnet must be identified and manually configured in each Expand device. Many enterprise networks have dozens or hundreds of subnets, and these networks also change over time – the Expand devices must constantly be kept up-to-date after each network change.

- **File cache peering relationships**: Each Expand device that is serving as an edge file cache must be configured with the name of the core file caches with which it may communicate – even though peer IP addresses have already been entered to setup compression tunnels. File cache configuration must be done separately from the tunnel configuration steps.

- **File server identification**: Each origin file server containing files that may be accessed via the Expand devices must be explicitly entered into the core cache device fronting that server – adding yet another layer of complexity to any deployment with multiple file servers in multiple locations.

Expand uses a legacy file caching approach

For CIFS-based file access, Expand uses a legacy file caching approach. Any file caching approach ultimately depends on maintaining multiple separate copies of files and synchronizing those different file copies over the WAN. Inevitably, these synchronization techniques break down as the scale of the deployment increases, resulting in data coherency and integrity problems. In contrast, Riverbed's non-caching architecture means that all users are always operating on a single file copy on the server, ensuring consistency even in complex scenarios involving multiple users and various failure scenarios.

In addition to its potential for consistency and coherency problems, file caching is also less efficient at reducing WAN traffic and improving performance in many common situations. For example, a file cache cannot recognize and leverage data commonalities between files with different filenames, even if the files are identical or nearly-identical. Nor can caching recognize and leverage commonalities across files delivered using different application protocols, such as when a same file is received as an email attachment and subsequently saved to a CIFS share drive or uploaded to a document management system. Such transfers will get cold performance with Expand, while Riverbed's CIFS latency optimizations and disk-based data reduction will provide warm, LAN-like performance.

Expand's file caching functionality also suffers from a number of implementation-related limitations and drawbacks. The file cache is poorly integrated into the overall product, with certain identical configuration steps that must be performed repeatedly, once for memory-based compression, and a second time for file caching. Expand also imposes a rigid core/edge configuration choice for each device participating in file caching. This may be acceptable if all file servers are in one or a few data centers, but not for many enterprises where file servers are potentially hosted in a number of different sites.

Expand lacks application-specific support for key enterprise applications

Riverbed offers mature and proven application-specific optimization capabilities to address a broad portfolio of chatty application protocols and application-specific data formats. Today, these features are being successfully used by thousands of Riverbed customers in their production networks.

As described in the previous section, Expand has resorted to a file caching approach for addressing CIFS protocol chattiness issues, and a basic web cache for addressing protocol chattiness in HTTP-based applications. However, Expand has nothing to address the myriad of other application protocols that exhibit chatty protocol behavior. The following table compares application-specific optimization capabilities for Riverbed and Expand:

Riverbed Application-Specific Capabilities	Expand Application-Specific Capabilities
• CIFS (SMB version 1) • SMB version 2 • SMB Signed CIFS (for both SMB v1 and v2) • CIFS Print • Macintosh CIFS (OSX 10.5 and 10.6) • Exchange 2003/2007/2010 (MAPI) • Encrypted Exchange/MAPI • Outlook Anywhere (RPC over HTTPS) • Lotus Notes • HTTP • SSL (includes client-side certificate authentication) • Oracle Forms (both JInitiator and JRE) • Citrix ICA and CGP (XenApp and XenDesktop) • NFS • MS-SQL • FCIP • SRDF including auto-disable compression	• CIFS (file cache) • HTTP (basic web cache) *All other applications optimized only through generic memory-based compression and TCP optimization (no latency optimization and no disk-based data reduction)*

Expand's only application-specific functionality is CIFS file caching and basic HTTP web caching. Expand lacks application-specific optimizations for any other application. In contrast, Riverbed supports application-specific optimization for a wide range of different protocols (see table above). For applications other than CIFS and HTTP, the only benefit that Expand can offer is QoS and its basic memory-compression – in other words, slight bandwidth reduction and no acceleration. In contrast, Riverbed's application streamlining techniques include specific optimizations for each of the above traffic types, in addition to CIFS and HTTP.

Expand's marketing claims on ICA and RDP acceleration

Expand makes some claims about optimizing thin-client traffic flows such as Citrix ICA and RDP, as an attempt to distract from their weak WAN optimization capabilities. However, Expand's product capabilities to optimize ICA and RDP are by not unique; Riverbed and other vendors offer similar capabilities to perform memory-based compression and QoS on ICA and RDP traffic.

But unlike Expand, Riverbed offers the capability to dynamically disable the compression mechanism within the Citrix XenApp server (i.e., the ICA Presentation server). This feature is enabled in the Steelhead GUI management interface, and it does not require any configuration change in the XenApp server; once enabled, the Steelhead will dynamically communicate with the XenApp server to disable the default compression mechanism. This allows Riverbed's byte-level data reduction algorithms to be effectively applied on the clear-text Citrix ICA data traffic, yielding superior data reduction results compared to what previously would have been achieved through the default ICA compression mechanism alone. Because this Riverbed feature works on a per-flow basis, remote sites without a Steelhead appliance will continue to obtain benefits from the XenApp server's default compression. On the other hand when using Expand to optimize Citrix ICA traffic, the default compression mechanism must be manually disabled through a number of detailed configuration steps performed in the XenApp server. This has the drawback that all ICA traffic from that XenApp server will be uncompressed, including traffic sent to local users accessing from a LAN, as well as to remote sites not equipped with Expand devices.

Riverbed also offers advanced QoS enforcement capabilities, but with two features that Expand QoS lacks: hierarchical traffic classes, and Hierarchical Fair Service Curves (HFSC), an approach to implementing QoS priority queuing that decouples bandwidth and latency. HFSC allows latency-sensitive traffic class (such as ICA or RDP) to be given high priority without having to over-allocate bandwidth.

As a tunneling-based product, Expand also highlights its header compression functionality as a means to reduce bandwidth for thin client traffic. But for a TCP proxy-based solution such as the Riverbed Steelhead, header compression is a natural outcome of the traffic interception and optimization process itself. A TCP proxy already performs the equivalent of header compression on TCP-based traffic (including RDP and Citrix ICA) by coalescing small TCP segments into larger TCP segments. Note also that if applied too aggressively, header compression can add unneeded latency and jitter—exactly what should be avoided for real-time traffic in order to provide a good end-user experience..

Expand's promised software client is still not available

Expand has announced the forthcoming availability of a software-based client for mobile users with laptops or for small server-less branch offices. Despite multiple announcements over the past few years, Expand still has nothing to offer. In a press release dated September 3rd 2008, Expand stated that the product would be available in Q1 2009. Q1 of 2009 came and went without any product release from Expand. Then in a press release dated May 19th 2009, Expand announced availability of their software client, including "Hive" functionality for Q4 2009. But Q4 of 2009 came and went without any software client. Amusingly, as of April 2011 Expand's website continues to declare that their Mobile Accelerator Client (MACC) product will be "available 2010," despite the observation that we are now well into calendar year 2011.

Expand's past history of not delivering on their commitments creates uncertainty about its true capabilities if it ever becomes available. In contrast, Riverbed delivered the first version of the Steelhead Mobile client in September 2007, and has since released updates with multiple new features. Steelhead Mobile is a proven solution being used by hundreds of Riverbed customers, including a number of customers who have deployed thousands of Steelhead Mobile licenses in their environments.

Finally, it's important to note that Expand's software client—if it ever becomes available—is intended to interoperate with their existing appliance-based product. But interoperability requires that the future software client use the same core optimization technology employed by Expand's existing appliances, including memory-based compression, QoS, CIFS file caching, etc. Therefore, it would therefore be unrealistic for the future software product to out-perform Expand's existing appliance-based products. As already discussed in this document, Expand's existing appliance-based products have major weaknesses (i.e., weak or non-existent application-specific capabilities and memory-only byte-level data store) that cause them to significantly underperform Riverbed's Steelhead appliance and Steelhead Mobile software client offerings.

Expand's "Virtual Accelerator"

Expand has recently introduced a "Virtual Accelerator", which is essentially the Expand appliance software repackaged as a virtual machine. An important principle to keep in mind is that virtualized hostings of any software application are only as valuable as the original core software. Issues and shortcomings that may exist in the original software product don't go away just because that software is virtualized and hosted on a hypervisor such as ESX. This principle is as true for WAN optimization as it is for your typical software application. If your software application is poorly written or a headache for your employees to use, then merely hosting that software on a hypervisor isn't going to make things any better. On the other hand if you have very productive software that your business depends on, then virtualization provides you with additional options as far as deploying, scaling, and maintaining your tool. In light of this principle, it's important to note that Expand's "Virtual Accelerator" retains all of the key shortcomings discussed earlier in this document. This includes a lack of disk-based data reduction for byte-level data, data coherency and consistency risks when using the CIFS file cache, and a lack of layer-7 application-specific optimization capabilities for other types of application traffic.

Furthermore, note that one of the key drawbacks for any virtual-appliance WAN optimization product is that no existing hypervisor adequately exposes hardware resources, such as control of in-path bypass NIC interfaces to the hosted virtual machines. Expand circumvents this issue by creating a separate virtual machine to monitor the primary Expand optimization software machine. Although this separate virtual machine similarly lacks control of the hardware, it can execute a software bypass that forwards traffic directly between the in-path port interfaces should the main Expand optimization software machine stop responding, thus using software to simulate a fail-to-wire event.

But this is not as safe and straightforward as it may initially appear. In addition to its responsibilities to monitor the main Expand software machine, this separate virtual machine also has the complex task of buffering and routing potentially large amounts of network data to and from the main Expand optimization software. A software failure in Expand's virtual monitoring & forwarding machine, or any form of miscommunication with the primary Expand software machine (which might still respond to keepalive messages from the monitoring machine even if it is in a degraded state), can result in data being discarded and cascading failures that result in serious disruptions to the network. Since most studies show software failures to be more common than hardware failures, Expand's simulated fail-to-wire approach offers little real-world protection from a real failure.

SUMMARY

Expand's product is a combination of two separate legacy approaches that each have inherent limitations. File caching comes with consistency risks and cannot optimize many common use cases, such as the transfer of a renamed or slightly modified file. Memory-based compression is not effective for optimizing repetitive data that appears in large data flows or re-appears after long time intervals.

Expand appliances are significantly harder to install and deploy than Riverbed Steelheads, requiring each appliance to be explicitly configured with information for each peer Expand device it may communicate with. Furthermore, each server-side Expand appliance must to be configured with information for each file server to be optimized. Riverbed Steelheads automatically and dynamically discover each other and all origin file servers, without advance configuration by the administrator.

Expand capabilities for optimizing ICA or RDP are not unique; Riverbed is being used by many customers to optimize VDI traffic types including ICA and RDP. However, Expand lacks the ability to dynamically disable the default compression mechanism in the XenApp server. This is a capability offered by Riverbed, but not available from Expand.

The repackaged virtual machine form of Expand's appliance fails to address any of its shortcomings. Expand continues to lack a software client and continues to lack support for third-party services in the branch office.

Riverbed's Steelheads have a far broader feature set, including application-specific optimization of a number of key enterprise protocols which Expand can only optimize by means of its memory-based compression.

riverbed®
Think fast.™

Riverbed Technology, Inc.
199 Fremont Street
San Francisco, CA 94105
Tel: (415) 247-8800
www.riverbed.com

Riverbed Technology Ltd.
Farley Hall, London Rd. Level 2
Binfield
Bracknell. Berks RG42 4EU
United Kingdom
Tel: +44 1344 354910

Riverbed Technology Pte. Ltd.
391A Orchard Road #22-06/10
Ngee Ann City Tower A
Singapore 238873
Tel: +65 6508-7400

Riverbed Technology K.K.
Shiba-Koen Plaza Building 9F
3-6-9, Shiba, Minato-ku
Tokyo, Japan 105-0014
Tel: +81 3 5419 1990