TB640 Streamserver hardware guide

Document number
9010-00021-1B

October 2008

This page in intentionally left blank

Table of contents

This page in intentionally left blank

How to choose appropriate stream server hardware

While developing the TB640 adapter and the tbstreamserver software, we have performed several performance tests. According to the results of these tests, we have been able to determine guide-lines for choosing appropriate stream server hardware according to operating condition requirements.

The current document presents the results of our tests. However, we have not performed tests with a wide variety of hardware, thus we encourage our customers to follow the guide-lines presented in this document. We also encourage customers that wish to try other hardware solutions to run performance tests and share with us the results of those tests[1].

Typical performance bottle-necks

Available RAM

The tbstreamserver application requires some amount of **physical** RAM available on the host to cache audio data before it is being streamed.

- Total size of prompts: The tbstreamserver uses RAM to load prompts. Each prompt is loaded in RAM, and remains in RAM no matters if it is being played or not. It allows instantaneous play without causing a disk access. Prompts are generally used for short audio files that are frequently used.
- Playing files: The tbstreamserver uses double-buffering when playing files from disk. It plays one piece of the file while it loads the subsequent from disk. The amount of RAM will thus depend on the number of simultaneous playing files.
- Recording files: The tbstreamserver uses double-buffering when recording files to disk. It accumulates packets in a RAM buffer, while the other buffer is written to disk. The amount of RAM will thus depend on the number of simultaneous recording files.
- RAM Cache: The tbstreamserver will use any remaining RAM to implement RAM caching of the most recently played files. This reduces the number of disk access for frequently accessed files. The more RAM is available, the longer the file will remain in the RAM cache.

Available disk space

The tbstreamserver application saves audio files in aLaw or uLaw format on the host’s hard drive. This uses 8KB of disk space per second of audio stored. This allows about 2 years worth of audio on a single 500GB hard drive. The maximum number of hard drive per host depends on the server hardware.

The tbstreamserver application can also play/record compressed audio files to save disk space, and perform live compression/decompression. However this increases the CPU usage on the host. The only codec that offers reasonable performance is the VOX codec, which offers compression ratio of 2:1, and allows about 8,000 play or record channels on one Intel Core-2 Quad-core CPU.

Disk access performance

When playing audio files from disk, the tbstreamserver needs to regularly read data to disk for each playing channel. As each hard drive can provide a limited number of read access per second, this will limit either the number of simultaneously playing files, or the number of “calls per second”.

This bottle-neck can be solved by distributing files to play over multiple hard drives, using a good disk controller (on PCI-express or faster bus).

Note about using RAID

Please note that RAID should be used with caution with the tbstreamserver, as inappropriate configuration may cause much lower performance.

We recommend using RAID-Level-1 (mirroring) for redundancy, at the cost of 2x disk storage requirements.

If using RAID-Level-0 (striping), the size of the RAID stripes MUST exactly matche the size of the audio “chunks” used with the tbstreamserver (64KB, 128KB or 256KB), otherwise each disk access from tbstreamserver will require to move the two drives, thus reducing performance by 2x (no more faster than one drive). We don’t recommend using RAID-Level-0, as it provides no disk redundancy (on the opposite, if one of the drive fails, all data is lost!).

If using RAID-Level-5 (parity redundancy), the size of the RAID stripes MUST exactly matche the size of the audio “chunks” used with the tbstreamserver (64KB, 128KB or 256KB), otherwise each disk access from tbstreamserver will require to move multiple drives, thus reducing performance by 2x, 3x or even more.).

See below in this document for more information about RAID performance with tbstreamserver.

tbstreamserver audio database

The tbstreamserver application uses its own private database (stored on disk as one big file) to store audio files. This audio database has been optimized to avoid disk fragmentation, and provide much better write performance (about 2x) than writing to separate audio files on disk.

We recommend to play and record audio from this database, rather than from separate “wav” files on disk. However, this will make files not available from the file system. The tbstreamserver’s API will need to be used to export files from this database.

We recommend using dedicated disks to create the tbstreamserver’s audio database to avoid having the operating system or other applications also accessing the same disk and causing unpredictable performance drops

Ethernet card performance

We recommend running the tbstreamserver application on a host with a Gigabit Ethernet network card (and Ethernet switches). In fact, 100Mbps network cards will limit the number of simultaneous channel to a low number (around 1000 channels).

This bottle-neck can be solved by adding multiple Gigabit Ethernet cards in the same host, using separate subnets for each. Make sure that the total PCI bandwidth is not exceeded if they are connected to the same PCI bus. We recommend PCI Express Ethernet Cards or faster, as standard PCI bus does not have the necessary bandwidth.

Network performance

It is mandatory to design the Ethernet network architecture to make sure there exist no bottleneck between the tbstreamserver hosts and the TB640 adapters they play audio to. In fact, all Ethernet switches used should be Gigabit. If uplinks are required to inter-connect multiple Ethernet switches, the system architecture should be designed to avoid tbstreamserver hosts to stream audio to TB640 adapters that require going through the uplink. When this is impossible, the uplinks throughput should be high enough for the usage.

We recommend building a network redundant setup by doubling the number of Ethernet cards in the host server, and building two separate Ethernet networks to reach the two ports of the TB640 adapters to play/record audio with.

Host CPU

The host CPU is generally not the bottleneck for the tbstreamserver. Other bottlenecks will be reached before CPU capacity is exhausted (Ethernet bandwidth, Disk performance, PCI-express bus bandwidth, etc).

Please note, however, that the CPU usage will increase exponentially when getting close to the Ethernet card’s capacity, as the Kernel has to perform busy-waiting for writing more packets into the Ethernet card’s memory. Adding more CPU power to the host will not change anything in this case. So please keep that in mind to avoid getting to the wrong conclusion that the CPU is the bottleneck with the tbstreamserver, while the Ethernet card’s performance probably is the bottleneck.

Parameters that impact the bottlenecks

There are a few parameters that will affect the various performance bottlenecks of the tbstreamserver application.

tbstreamserver audio “chunk” size

The tbstreamserver reads files from disks in “chunks” of customizable size. The size of these “chunks” has an impact both on the RAM requirements, and on the disk access performance requirements.

Smaller audio “chunk” size will decrease the RAM requirements for playing files from disk, but will increase the number of disk access, thus increase the number of hard drives required to reach the required performance.

Bigger audio “chunk” size will do the opposite (increase RAM requirements; decrease the disk access performance requirements).

The tbstreamserver application allows the following audio “chunk” sizes: 64KB (8 seconds), 128KB (16 seconds) or 256KB (32 seconds). The ideal value will vary according to the host hardware and target application. For example, RAM vs. Hard drive cost may be taken in consideration. The maximum RAM and maximum number of hard drives in one system can also have an impact on this setting.

Packets duration

The tbstreamserver application supports packets duration of 20ms, 40ms, 80ms or 160ms.

However, 160ms packets should always be used. This will not increase the playback delay if the jitter-buffer on the TB640 adapter is configured to an appropriate value, such as 50ms.

All the performance measurements in this document are based on 160ms packets.

Identifying the bottlenecks

The requirement on the various bottlenecks of the tbstreamserver application should be calculated as follow.

Available RAM

Total RAM requirements for the host should be calculated as follow:

- Each audio prompt (playing or not) require **8KB of RAM per second**
- Each playing audio file requires **2 x "chunk size" KB** of RAM
- The tbstreamserver application requires about **32MB** for it's own usage
- The OS (Windows XP for example) requires about **256MB** for its own usage (512MB to 1GB for Windows Vista).
- tbstreamserver is available in 64 bits version for Linux and Solaris, if more than 2GB of RAM are required.

Available disk space

Total disk space requirements for the host should be calculated as follow:

- Each audio file requires **8KB per second**
- Nothing else should be stored on that disk to avoid unpredictable performance problems due to other applications or the operating system.

Disk access performance

Total disk access per second requirement for the host should be calculated as follow:

- While they are playing or recording, each channel requires one disk access every "chunk" duration (64KB chunks contain 8 seconds of audio, for example)[2]
- Here are typical numbers:
  - With 64KB audio chunks, each 7200 RPM hard drive is capable of about **150 accesses per second**, while high-end 15K RPM disks can provide more than 250 accesses per second.
  - With 256KB audio chunks, 7200 RPM drives reach about 100 accesses per second, while 15K RPM drives reach about 150 accesses per second.
- Thus this limits the number of play or record channels per drive to "**disk access per second" multiplied by "chunk duration in second".**

[2] If recording directly to disk (not using tbstreamserver's internal database), each chunk write actually requires two disk access, one for the chunk data write, one for the OS to update it's FAT.

Disk access performance with RAID-Level-1

If using RAID-Level-1 (mirroring), the read performance will be calculated using the total number of drives (as each of the two mirrored drives can simultaneously read a different file). However the write **performance will be calculated using the total number of drive pairs** (as each write must be done simultaneously on the two drives).

With RAID-Level-10 (combining multiple pairs of RAID-Level-1 drives), performance is multiplied by the number of drive pairs.

Disk access performance with RAID-Level-5

If using RAID-Level-5 (parity redundancy), the **read performance will be calculated using the total number of** drives (as each of the drives can simultaneously read a different data chunk). However the **write performance will be calculated using the total number of drive divided by two** (as parity must be updated for each chunk written).

Ethernet card performance

One typical Gigabit Ethernet card is capable of sending packets for about 8,000 audio channels (this includes a good comfort margin compared to maximum potential bandwidth).

Note: Increase the number of “transmit descriptors” for a given network adapter can reduce the host CPU usage. Typical default value number of “transmit descriptors” is 256. We recommend increasing to at least 1024. This is a configuration of the driver of the network adapter. Some network adapters don’t offer this configuration parameter.

Network performance

We recommend that all the TB640 which one tbstreamserver host is playing to are connected to the same Gigabit Ethernet switch. This avoids causing traffic on the uplink.

The network bandwidth requirements are about 67Kbps per channel, overhead included.

Method to identify the bottlenecks

As we just explained above in this document, there are multiple bottlenecks that can limit the tbstreamserver performance, and it is not trivial to choose the appropriate hardware configuration of a given application.

The current section of this document attempts to describe simple steps that allow identifying the main bottlenecks in order to choose the most appropriate hardware. Please not that these calculations are not exact and are somewhat based on statistical averages. We thus recommend to choose hardware that offers some performance margin above the numbers calculated with the method below (25-50% margin should be appropriate).

Do not forget to also take into account that performance will not scale infinitely. In fact, though two hard drives will probably be twice faster as one, 20 hard drives will definitely not be twice faster than 10, due to other bottle-necks in the system. This also holds for the number of Gigabit Ethernet ports.

The parameters you need to know

1. **TotalFilesSize**: The total size of audio files required on disk. This value may change a lot according to the application (ring-back-tone, music playback, voice mail, prompts playback).
2. **TotalPromptsSize**: The total size of prompts required. Prompts are generally used for short and frequently used files. They are kept in RAM all the time to avoid disk access when played. A typical usage of prompts is to join small audio files to build sentences such as “Monday May 19th, 2008…”).
3. **SimultPlayPrompt, SimultPlayFiles**, **SimultRecordFiles**: The total number of simultaneous channels required (prompts play, files play, and files record).
   The total being **SimultChannels**.
4. Disk RAID mode.

The bottlenecks calculation

1. Calculate the number of Gigabit Ethernet cards required, assuming one Ethernet card allows about 8,000 channels (play and record combined):
   NbEthernetCards = **SimultChannels** / 8,000 channels per card.
   Make sure that the host hardware can support that number of network cards (including PCI bottlenecks). Otherwise multiple hosts will be required.
2. Calculate the number of hard drives required to store these audio files, assuming 500GB drives:
   NbDrivesForStorage = (**TotalFilesSize** / 500GB)
   x 2 for RAID-1 setup
   x (N-1) for a N-drives RAID-5 setup.
3. Calculate the number of hard drives required for the required number of simultaneous channels (do the calculation for 64KB, 128KB and 256KB audio “chunks” size):
   a. **NbDiskAccessPerSecPlay** =
      (**SimultPlayFiles**/ **chunkDuration**).

   b. **NbDiskAccessPerSecRecord** = (**SimultRecordFiles**/**chunkDuration**).

   c. **NbDisks** = **(NbDiskAccessPerSecPlay** + (**NbDiskAccessPerSecRecord** x 2 if RAID)) / nb access per second per disk.

4. Calculate RAM requirements (for 64KB, 128KB and 256KB audio “chunks” size), assuming 256MB reserved for the operating system and 32MB for the tbstreamserver:
   RamUsage = 256MB (OS) + 32MB (tbstreamserver) + **TotalPromptsSize** + (2 x **(SimultPlayFiles+SimultRecordFiles)** x chunk size)

5. Determine the appropriate audio “chunk” size by choosing the most appropriate balance between RAM usage and number or disks required. Make sure that the host hardware can support that number of disks, or that amount of RAM.

Example 1

The following example describes a setup that would be suitable to build a voice mail application (messages being recorded and played back, with menu options implemented using short audio prompts in different languages).

Parameters:

1. TotalFileSize:	**2 TB** (over 8 years worth of audio, which represents by example, for a voicemail application, over 4 minutes of audio for more than 1 million users!).
2. TotalPromptSize:	**256MB** (about 9 hours worth of "instant" audio prompts)
3. SimultaneousChannels:	2,000 playing prompts 3,000 playing files 3,000 recording **8,000 total**
4. Raid-Level-5	Used for redundancy

Calculations:

1. NbEthernetCards:	8,000 channels / 8,000 channels per card = **1 Ethernet card** (or two if using network redundancy).
2. NbDrivesForStorage:	(2 TB / 500 GB) = 4 hard drives Plus one for RAID-5 = **5 dedicated hard drives**
3a. NbDiskAccessPerSecPlay:	Using 64KB audio "chunks": (3,000 channels / 8 sec per access) = 375 Using 128KB audio "chunks": (3,000 channels / 16 sec per access) = 188 Using 256KB audio "chunks": (3,000 channels / 32 sec per access) = 94
3b. NbDiskAccessPerSecRecord:	Using 64KB audio "chunks": (3,000 channels / 8 sec per access) = 375 Using 128KB audio "chunks": (3,000 channels / 16 sec per access) = 188 Using 256KB audio "chunks": (3,000 channels / 32 sec per access) = 94

3c. NbDisks: Using RAID, number of disk access per second for recording is multiplied by two.
7200 RPM drives are assumed, with estimate of 150, 125 and 100 access per second for 64,128 or 256KB chunks:
Using 64KB audio “chunks”:
(375 + 375x2) / 150 = **8 hard drives**
Using 128KB audio “chunks”:
(188 + 188x2) / 125 = **5 hard drives**
Using 256KB audio “chunks”:
(94 + 94x2) / 100 = **3 hard drives (but a minimum of 5 were required for total disk space)**

4. RamUsage: Using 64KB audio “chunks”:
256MB (OS) + 32MB (tbstreamserver) + 256MB (prompts) + (2 * (3,000+3,000) channels * 64KB) = **1,294MB**
Using 128KB audio “chunks”:
256MB (OS) + 32MB (tbstreamserver) + 256MB (prompts) + (2 * (3,000+3,000) channels * 128KB) = **2,044MB**
Using 256KB audio “chunks”:
256MB (OS) + 32MB (tbstreamserver) + 256MB (prompts) + (2 * (3,000+3,000) channels * 256KB) = **3,544MB**

5. We now have three disks vs. RAM configurations to choose from

- 64KB audio “chunks”: 1,294 MB RAM, 8 hard drives of about 250GB
- 128KB audio “chunks”: 2,044 MB RAM, 5 hard drives of about 500GB
- 256KB audio “chunks”: 3,544 MB RAM, 5 hard drives of about 500GB

In this example, we clearly see, comparing results for 128KB and 256KB audio “chunks”, that 256KB chunks almost doubles the amount of RAM required, and does not increase a lot the cost of disks. The best solution here would probably to use 128KB audio “chunks”.
That would require 2GB of RAM, though it may be wise to add more RAM so RAM caching mechanism of tbstreamserver reduces the number of disk access, and thus increase disks duration.

Final configuration:

The final configuration for this example would probably be:

- 3Ghz Pentium-4 CPU or faster (Intel Core-2 at 2Ghz or faster)
- 2GB RAM minimum (4GB recommended)
- 5 Hard drives, 500GB, configured in RAID-Level-5 mode, with 128KB RAID stripe size, plus one small hard drive to install the OS and other applications.
- 1 Gigabit Ethernet Cards (or 2 to use network redundancy across 2 distinct Ethernet networks).

Example 2

The following example describes a smaller setup that would be suitable for music playback, ring-back-tone or similar audio playback application.

Parameters:

1. TotalFileSize:	**100 GB**
2. TotalPromptSize:	**16 MB**
3. SimultaneousChannels:	500 playing prompts
	1500 playing files
	0 recording
	2,000 total
5. Raid:	Not used

Calculations:

1. NbEthernetCards:	2,000 channels / 10,000 channels per card = **1 Ethernet card**.
2. NbDrivesForStorage:	**1 hard drive**
3a. NbDiskAccessPerSecPlay:	Using 64KB audio “chunks”:
	(1,500 channels / 8 sec per access) = 188
	Using 128KB audio “chunks”:
	(1,500 channels / 16 sec per access) = 94
	Using 256KB audio “chunks”:
	(1,500 channels / 32 sec per access) = 47
3b. NbDiskAccessPerSecRecord:	0
3c. NbDisks:	7200 RPM drives are assumed, with estimate of 150, 125 and 100 access per second for 64,128 or 256KB chunks:
	Using 64KB audio “chunks”:
	188 / 150 = **2 hard drives**
	Using 128KB audio “chunks”:
	94 / 125 = **1 hard drive**
	Using 256KB audio “chunks”:
	47 / 100 = 1 **hard drive**

4. RamUsage: Using 64KB audio “chunks”:
256MB (OS) + 32MB (tbstreamserver) + 16MB (prompts) + (2 * 1,500 channels * 64KB) = **492MB**
Using 128KB audio “chunks”:
256MB (OS) + 32MB (tbstreamserver) + 16MB (prompts) + (2 * 1,500 channels * 128KB) = **679MB**
Using 256KB audio “chunks”:
256MB (OS) + 32MB (tbstreamserver) + 16MB (prompts) + (2 * 1,500 channels * 256KB) = **1054MB**

5. We now have three disks vs. RAM configurations to choose from

- 64KB audio “chunks”: 492 MB RAM, 2 hard drives of about 50GB
- 128KB audio “chunks”: 679 MB RAM, 1 hard drives of about 100GB
- 256KB audio “chunks”: 1054 MB RAM, 1 hard drives of about 100GB

In this example, 64KB chunks require more than one drive, while 256KB only increase RAM usage, which makes the 128KB chunks option the best solution.

Final configuration:

The final configuration for this example would probably be:

- Any Pentium-4 CPU or faster
- 1GB RAM.
- 1 Hard drive of 100GB, plus one small hard drive to install the OS and other applications.
- 1 Gigabit Ethernet Cards (or 2 to use network redundancy across 2 distinct Ethernet networks).

Hardware recommendations

RAM:

We recommend the fastest available RAM (we tested with DDR 333MHz RAM). Adding more than the minimum RAM may be wise, due to RAM cache effect on drive's life duration, and reduced risks of impact in case other application or services are running on the same host (which we don't recommend, of course!).

Hard drives:

We recommend using server grade 10,000 RPM drives that are rated to last for a long time under heavy load.

We recommend using **SAS or SCSI disks** over SATA disks because most SAS/SCSI controllers allow hardware queuing and reordering disk accesses to minimize seek time. SAS/SCSI disks are also generally built to last longer under heavy load conditions.

We recommend to store the stream server audio database on **dedicated disks** where NO OTHER FILE than the stream server's database are allowed to avoid disk fragmentation and maximize performance.

We do not recommend using RAID-level-0 due to lack of redundancy.

We recommend using RAID-level-5 for redundancy, **as long as the RAID stripe size matches the tbstreamserver's audio chunk size**.

Ethernet cards:

We recommend using **Gigabit Ethernet cards** (two if network redundancy is required, one otherwise) that are connected to the CPU through **PCI-Express** or faster BUS

PCI network adapters shoud NOT be used, because the PCI bandwidth is insufficient for Gigabit performance (lower than 1Gbps half-duplex, shared with all devices on the BUS). We DO NOT recommend using 100MBps network adapters.

We highly recommend using **1Gbps network switches** (two switches on separate subnets for network redundancy). The switch must provide throughput of 2Gbps per port (1Gbps Rx + 1Gbps Tx) and a backbone that supports this rate simultaneously for every port. The switch must handle at least 100,000 packets per second simultaneously on every port.

NEVER USE NETWORK HUBS, NEVER MIX DIFFERENT SWITCH SPEEDS (1Gbps connected to 100Mbps, for example).

If stream traffic has to be routed between separate Gigabit switches, the uplink between these switches MUST support the total aggregated bandwidth of all ports of the switches used for streaming.

CPU:

We recommend using one **3GHz Pentium-4** processor or faster. Intel Core-2 is even better, but should not be absolutely required.

IMPORTANT: On Windows, DO NOT CHANGE THE PRIORITY OF THE STREAM SERVER using Windows Task Manager (or other tool). When the priority is changed, the relative priority of the threads inside the application seems to be lost. The stream server will work properly only if relative priority of threads (relatively to each other) inside the stream server application are not modified.

IMPORTANT: On Solaris and Linux, it is mandatory that the user launching the application has the privileges to change threads scheduling to real-time at system-scope. Only the root user has these privileges by default.

Load balancing:

The stream server API easily allows an application to perform load balancing. Using the stream server API, any application can simultaneously query multiple stream servers for the current load, or availability of a file or prompt.

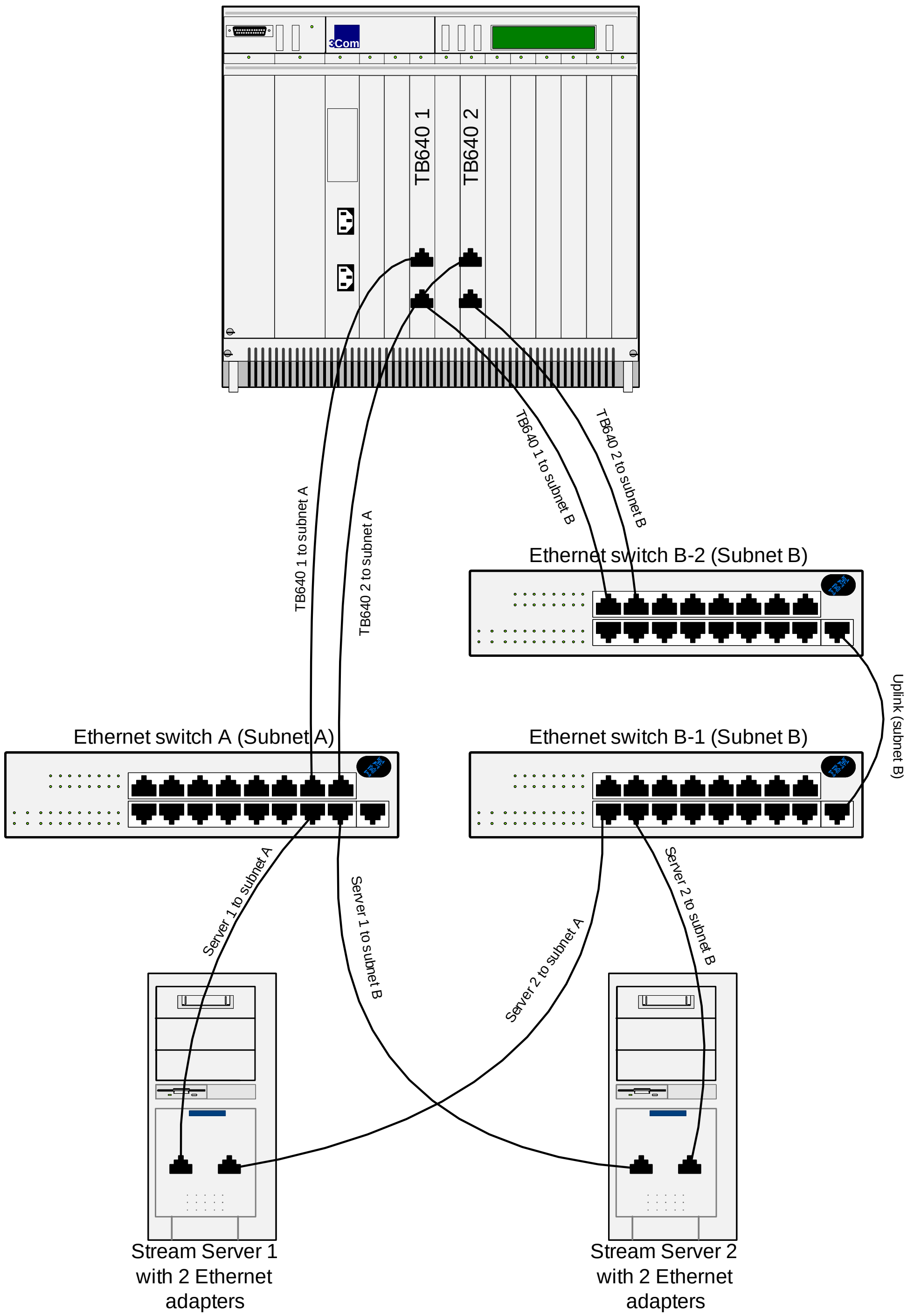
Using the information returned by every available stream server, the application can choose the stream server that has the least load to perform the next playback. The “amount of load” can be determined by looking at the number of active channels and the number of disk access pending, and comparing those values with the capacity of the server.

Because all the servers run on the same network and are all connected to every TB640 adapters through the Gigabit Ethernet switches, a properly designed network will allow any stream server to handle a play on any port of any TB640 adapter, thus allowing the controlling applications to easily implement load balancing.

The tbtoolpack software will automatically handle load balancing to multiple tbstreamservers.

Example network organization

When designing the network for a streaming application between stream servers and TB640 adapters, it is important to make sure the network has no bottleneck.
The following diagram shows an example network that allows two stream servers to feed two TB640 adapters with network redundancy.



End of the document

