

# Ensuring QoS in Professional Audio Networks which use IEEE 1394b

Fred Otten and Richard Foss  
Department of Computer Science  
Rhodes University, Grahamstown, South Africa  
Tel: 046 6038291 Fax: 046 6361915  
g05o5894@campus.ru.ac.za, r.foss@ru.ac.za

**Abstract**—The calculation of the bandwidth available, the latency of audio and control data and making sure the correct number of audio channels can be transmitted is of paramount importance to maintaining Quality of Service (QoS) in an audio network. QoS is very important in professional audio networks where audio needs to be delivered accurately to various nodes on the network. Firewire (IEEE 1394) is one technology which is used in these networks. Due to the limitation of cable length in IEEE 1394a and the improvements in IEEE 1394b, IEEE 1394b needs to be used. By performing calculations to determine the relevant bandwidth and latencies in any given scenario, the audio engineer can ensure QoS in a professional audio network. This paper begins by giving a brief overview of firewire and then discusses bandwidth calculations for IEEE 1394a networks. It then examines the differences between IEEE 1394a and IEEE 1394b and presents a method which can be used to calculate bandwidth in an IEEE 1394b network. This method is evaluated using data obtained from a real network and shown to be accurate.

**Index Terms**—Modelling, Simulation, Bandwidth Calculation, Professional audio networks, IEEE 1394, Quality of Service

## I. INTRODUCTION

The use of digital multimedia networks for professional audio and video is increasing. The advantages of less cables, dynamic routing and flexible command and control make digital multimedia networks attractive. Professional audio networks are large audio networks used in convention centers, studios and stadiums. They consist of a number of nodes/devices, such as amplifiers, mixing desks and routers, which are connected together using Firewire (IEEE 1394a [9] or IEEE 1394b [6]) or Ethernet (CobraNet [5], Ethersound [3] and AVB [4]). The equipment used in these networks is expensive, so the networks need to be carefully planned by Audio Engineers before deploying them. To ensure QoS, they need to take into consideration the bandwidth available, the latency of audio and control data, and make sure the correct number of audio channels can be transmitted.

This paper begins by giving a brief overview of firewire and then discusses bandwidth calculations for IEEE 1394a networks. It then examines the differences between IEEE 1394a and IEEE 1394b and presents a method which can be used to calculate bandwidth in an IEEE 1394b network. Finally, this method is evaluated using data obtained from a real network.

## II. FIREWIRE NETWORKS

Firewire is a high speed serial bus based on CSR architecture - ISO/IEC 13213 (ANSI/IEEE 1212) "Control and Status registers architecture for microcomputer buses" [7]. It was

The work reported in this paper was undertaken in the Telkom Centre of Excellence in Distributed Multimedia at Rhodes University, with financial support from Telkom, Comverse, Tellabs, StorTech, Amatole Telecom Services, Bright Ideas 39, THRIP, DAAD and The Andrew Mellon Foundation

standardised by the IEEE as IEEE 1394 in 1995 [8]. Subsequent revisions occurred in 2000 (IEEE 1394a) [9], 2002 (IEEE 1394b) [6] and 2006 (IEEE 1394c) [10].

Firewire networks consist of a number of buses (up to 1024) which are joined together using bridges or routers. A bus can contain up to 64 nodes. These nodes are logical entities on firewire devices (which are also termed modules). Each node can in turn contain a number of ports which can be used to daisy chain devices. One of the attractions of the firewire bus is the dual transmission modes - Asynchronous and Isochronous. Asynchronous transmission allows devices to read and write from/to an addressed location periodically with guaranteed delivery, while isochronous transmission allows devices to perform real-time streaming of data by sending data every 125 $\mu$ s. Isochronous data can use up to 80 percent of the available bandwidth (100 $\mu$ s). Each device is able to transmit a number of sequences within isochronous streams, which are allocated channel numbers.

IEEE 1394 and IEEE 1394a both use a signalling technique called data/strobe signalling in the physical layer. Data is transferred on one pair, while a strobe is transmitted on the other pair. This means that data cannot be transferred in both directions at the same time. Bi-directional data transfers are therefore accomplished by using a half-duplex implementation.

When the network is powered up or a bus reset occurs, a self identification process (Self-ID phase) occurs and each node is assigned a Node ID. During this process, a tree is built and a root node (also called the cycle master) is identified. This node is responsible for sending out CYCLE\_START packets at regular intervals. This packet denotes the start of the isochronous interval during which the transmission of isochronous data occurs. The isochronous interval is ended by an idle gap which is called a sub action gap. These idle gaps also occur between asynchronous transmissions which are also termed sub actions. Certain nodes are also assigned management roles within the IEEE 1394 network. These are the Bus Manager and the Isochronous Resource Manager (IRM) (these roles can be assigned to a single node). The Bus Manager is responsible for power management, optimising bus traffic, and storing topology information and a speed map (which it determines during the Self-ID phase). The IRM keeps track of the bandwidth and channels which are available for the transmission of isochronous data. When a node wishes to send isochronous traffic, it attempts to modify the CHANNELS\_AVAILABLE and BANDWIDTH\_AVAILABLE registers in the IRM. The IRM, in this way, makes sure that the isochronous interval does not exceed 80 percent of the available bandwidth and ensures that bandwidth is available before allowing a device to transmit isochronous data.

To ensure that there are no collisions on the bus, an arbitra-

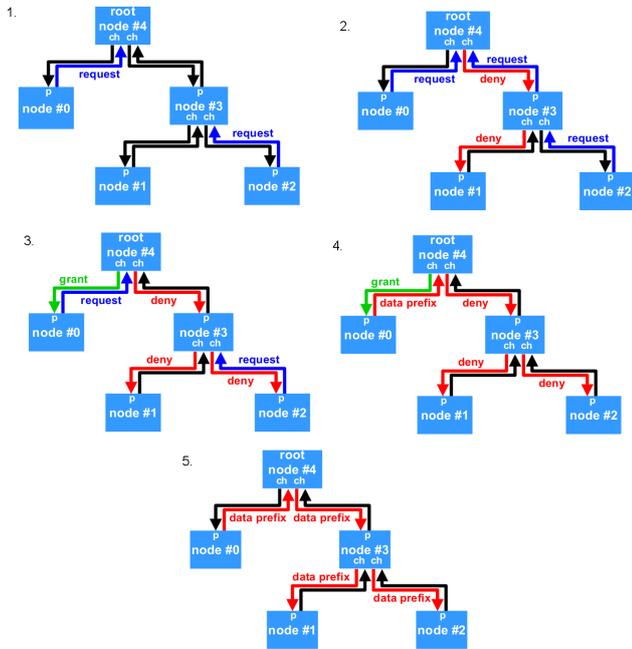


Figure 1: Legacy Arbitration [12]

tion mechanism needs to be used. This is of particular interest to bandwidth calculation since it influences the time when the nodes are able to transmit. It also means that additional gaps and communication might be necessary to obtain use of the bus.

Figure 1 shows the technique used for arbitration in IEEE 1394 and IEEE 1394a networks. Each node wishing to arbitrate sends a request to their parents. The parents forward the request and send a deny to the other children. Eventually the request gets to the root (which is responsible for handling arbitration) and it sends a grant to the node which wins arbitration. This node then sends out a DATA\_PREFIX packet which gets sent to all the nodes. Once the other nodes see this packet, their requests are withdrawn and all the nodes are ready to receive the data transmission. The other nodes need to then wait for a gap to arbitrate. This technique is designed for a bus in which bi-directional transmission is only possible using a half-duplex implementation. Idle gaps are also used in IEEE 1394 to mark the end of the self-ID phase, terminate the isochronous interval and separate asynchronous transmissions.

IEEE 1394a was created to clarify the initial specification, add additional features and performance improvements. IEEE 1394b defines further improvements and uses a new signalling method. This is described in Section IV.

### III. BANDWIDTH CALCULATION FOR IEEE 1394a FIREWIRE NETWORKS

Real-time audio is sent within each isochronous interval, so the isochronous interval is the focus from this point. In order to calculate the amount of bandwidth an IEEE 1394 node is able to use to transmit sequences, the amount of overhead due to arbitration, the network configuration (cable length and topology) and the packet headers need to be taken into account. These all effect how the BANDWIDTH\_AVAILABLE register content should be calculated.

As noted in the previous section, idle gaps are used to avoid contention and hence increase the overhead. This overhead is therefore completely dependent on the topology of the network, the length of the cables used, and settings such as the GAP\_COUNT (which is used to determine the size of the

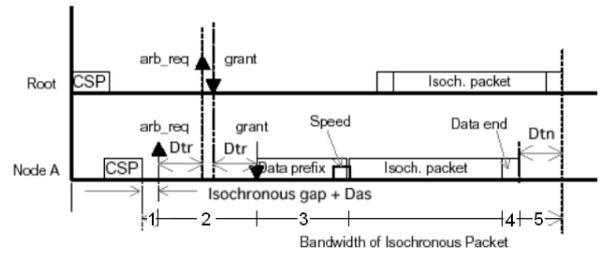


Figure 2: Firewire bus for root and node A over a given time period

sub action gap and arbitration reset gaps). Anderson [1] gives the following formulae for calculating the sub action gap and arbitration reset gap based on the GAP\_COUNT (where  $g$  is the GAP\_COUNT):

- Sub action Gap:  $\frac{29+16g}{98.304} \mu s$
- Arbitration Reset Gap:  $\frac{53+32g}{98.304} \mu s$

By modifying the GAP\_COUNT, the length of the sub action gap and arbitration reset gap can be minimised and hence more bandwidth can be made available for the transmission of data.

Yamaha [2] recommend the root should be forced to be the most central node in the network (this minimises the total time spent sending data to the root when sending arbitration requests). They also recommend that based on the maximum number of hops in the network, the GAP\_COUNT should be set to a minimal value for the given network configuration. This ensures that every device in the network will be able to detect the sub action gaps, while minimising their length.

When transmitting packet data in an IEEE 1394 network, a DATA\_PREFIX is first transmitted as a packet transmission signal (this includes an indication of the packet speed). This is followed by the transmission of the packet data and a DATA\_END symbol, which signals that packet transmission has completed. Once the packet transmission has completed, the bus enters an idle state. After the bus has been in an idle state for a period of time termed an isochronous gap ( $0.04\mu s$ ), arbitration begins. The next node then transmits its packet data in the manner described above. This process continues until all the nodes have transmitted. Once this has happened, the bus will be idle for a sub action gap and then asynchronous data will be transmitted until the end of the cycle.

Figure 2 shows the bus of an IEEE 1394a network over time from the point of view of the root and an arbitrary node - node A. This figure shows a CYCLE\_START packet, node A arbitrating for use of the bus, receiving a grant from the root and then transmitting an isochronous packet. This shows that apart from the data being sent, there are a number of additional factors which need to be taken into consideration when calculating the amount of bandwidth which is available for transmitting isochronous data. These factors include:

- The arbitration start delay (1)
- The arbitration delay (which takes into account the delay between the root node and node A - Dtr) (2)
- The time taken transmitting the data prefix (3)
- The time taken transmitting the data end (4)
- The delay for the packet to reach the next transmitting node (5)

These factors are labeled in Figure 2 using the numbers indicated in brackets.

Yamaha [2] take into account these factors and use a method to produce a table and calculate the amount of overhead in the network. The rest of this section describes this method.

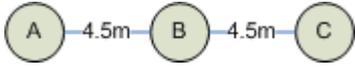


Figure 3: 3 node firewire network

	A	B	C	Total
Arbitration Start Delay	226	226	226	678
Arbitration Delay	0	333.45	666.9	1000.35
Data Prefix	140	140	140	420
Data End	260	260	260	780
Delay to Next	22.73	22.73	22.73	68.18
PHY Delay	144	144	144	432
Total Overhead (ns)	792.73	1126.18	1459.63	3378.53
Total Overhead (BWU)	38.95	55.34	71.73	166.02
Total Overhead ID				6
Overhead ID	2	2	3	7
Accumulated Overhead ID	2	3	6	
Adjusted Overhead ID	2	1	3	6

Table I: Total overhead for isochronous transmission

According to Yamaha [2], the PHY delay is 144ns. The delay to root and delay to next transmitting node is calculated by adding this to the cable delay, which according to Yamaha [2] is 5.05ns/m. Yamaha [2] also note that the time for the data prefix and data end are 140ns and 260ns respectively. Once the network is analysed and the overhead is calculated, the overhead is converted to bandwidth units (BWU). A BWU is defined as the time taken to transport 1 quadlet of data at 1600Mbps (i.e. 20.35ns) [2]. This means that the maximum amount bandwidth which can be used for isochronous transmission is 4915.20 BWU (100000ns). It is from this value that we need to subtract the overhead due to the factors described above. In the Yamaha scheme, each node is also assigned an Overhead ID where one Overhead ID is equivalent to 32 BWU. The Overhead ID is constrained to one byte and is used by the Yamaha devices when altering the BANDWIDTH\_AVAILABLE registers. The Overhead ID is rounded up for each node and hence can be too high. The Overhead ID is therefore adjusted so that the Overhead ID used by the device is not too high (which ensures the maximum amount of bandwidth can be used). This is done by observing the summed values of the Overhead IDs and the total amount of overhead at the time the device transmits.

Figure 3 shows an example three node network which is used for calculations to show the impact of these factors. Each node has a cable of 4.5m between them (which is the maximum length for a cable in an IEEE 1394a network). The root is Node A, so therefore the transmission order is Node A, then Node B and then Node C (i.e. in order of who is closest to the root). The maximum amount of hops in this network occurs between Node A and Node C (2 hops). This translates to an optimal GAP\_COUNT of 2, which leads to a sub action gap of 620.52ns (which is equivalent to 30.49 BWU).

Table I shows the calculation of the overhead for each node in the network. The arbitration delay and the delay to next are calculated using the cable length (A-B and B-C are both 4.5m which equates to 22.73 ns). The arbitration delay is the time to the root and back including PHY delay (A-B = 166.73ns so A-B-A = 333.45ns). In addition to these factors shown in the table, a sub action gap (before the start of asynchronous transmission) and the overhead for the headers of the isochronous channels need to be taken into account. The overhead for the sub action gap is 30.49 BWU (as described above), while the overhead for headers is 60 BWU when transmitting at 400 Mbps (3 channels - 1 per device, 5 quadlets at 400 Mbps). This leads to a total overhead of 282.49 BWU, which means that 4632.71 BWU out of the 4915.20 BWU are available for isochronous transmission.

This equates to 144.77 (i.e. 144) sequences at a sampling rate of 48 KHz and a transmission speed of 400 Mbps - each sequence requires 32 BWU (8 quadlets at 400 Mbps).

#### IV. IMPROVEMENTS IN IEEE 1394B NETWORKS

The IEEE 1394b bus is a backwards compatible bus created to support faster speeds than 400 Mbps (which is supported by IEEE 1394 and IEEE 1394a specifications). This bus supports speeds of 800 Mbps and was created to support up to 3.2 Gbps. It also supports a number of different mediums including: optical, coaxial and twisted-pair cabling. This section describes some of the improvements in the IEEE 1394b bus. It begins by describing the new Physical Layer - particularly discussing beta mode signalling (which uses the 8B/10B encoding). It then describes the BOSS arbitration technique which is introduced to remove the need for sub action gaps and take advantage of the full duplex nature of beta mode signalling.

##### A. IEEE1394b Physical Layer

With the IEEE 1394b project, the IEEE aimed to overcome the limitations of IEEE 1394a (short cable length and bandwidth wasted by the use of idle gaps) in order to broaden the scope of IEEE 1394 and make the interface more valuable to the end user. To enable the use of longer cables, IEEE 1394b uses a form of 8B/10B encoding [13], which was developed by IBM to enable transmission over longer distances, in beta mode signalling. The use of beta mode signalling rather than data strobe signalling also means that full duplex transmission is possible, since a strobe does not need to be transmitted on the second pair. IEEE 1394b still maintains the ability to do data-strobe signalling to ensure legacy compatibility.

##### B. BOSS Arbitration

A number of arbitration enhancements such as fly-by concatenation, ACK accelerated arbitration, multispeed concatenated packets and priorities are included in IEEE 1394a [1]. These arbitration techniques, however, still include sub action gaps and hence do not make optimal use of the available bandwidth. IEEE 1394b solves this problem by defining a new arbitration technique which takes advantage of the full duplex nature of beta mode signalling. This technique is called Bus Owner/Supervisor/Selector (BOSS) arbitration. In BOSS arbitration, the request signalling is overlapped with data transmission, which removes the need for idle gaps. By removing these idle gaps, extra bandwidth is made available for data transmission.

In legacy arbitration, each device sends a request to a fixed root which determines the winner of arbitration. This means that devices which are further away from the root have to wait longer for arbitration requests to be completed. The last node to transmit is usually in the proximity of the next node to transmit, so instead of having a fixed root which determines the winner of arbitration, BOSS arbitration uses a variable node. When a node begins transmission, it becomes the BOSS. It is the only node in the bus that can be receiving arbitration requests on every active port and is therefore in the best position to decide which node to issue a grant to. The BOSS is initially the root node.

Figure 4 shows an example of requests being sent to the BOSS on one channel while data is sent from the BOSS on the other channel. When a node begins transmitting a packet then it becomes the BOSS immediately. This means that the

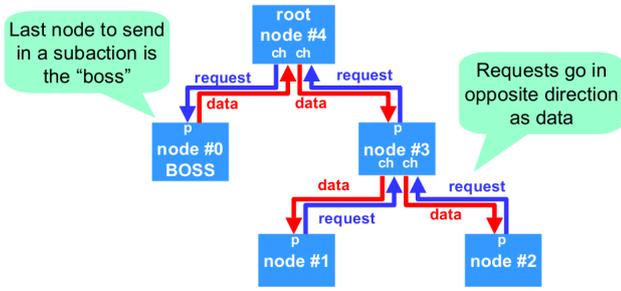


Figure 4: BOSS Arbitration with Data being sent from the BOSS and requests being sent to the BOSS [12]

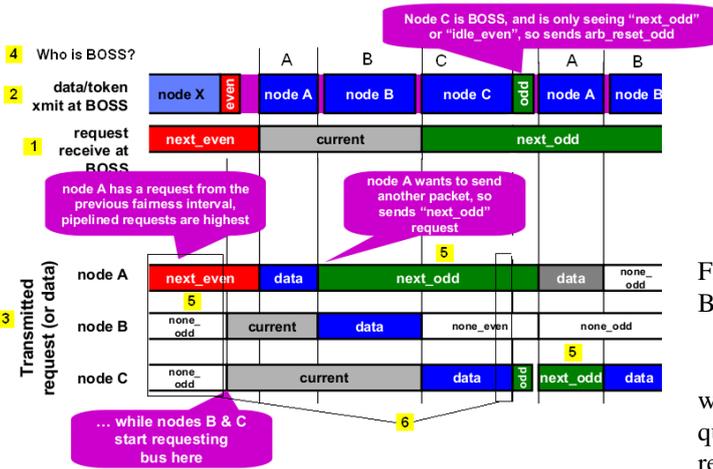
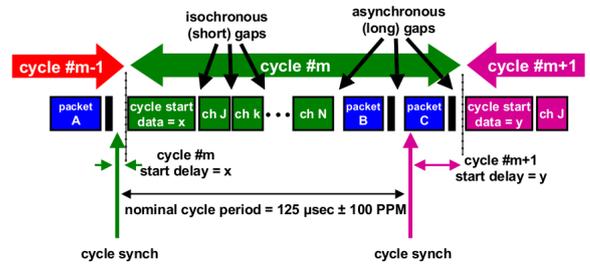


Figure 5: BOSS arbitration with 3 nodes [12]

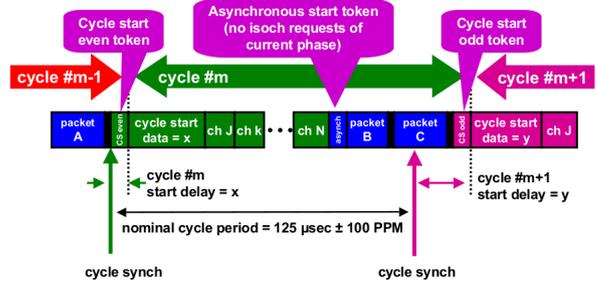
transmission of an acknowledgment packet or a PHY response packet establishes a node as BOSS. A node can also become BOSS by receiving a grant. There are two different types of grants which can be issued: an explicit grant and an implicit grant. An explicit grant occurs when a PHY receives a packet whose packet end consists of GRANT or GRANT\_ISOCH symbols or when these symbols have been sent to a node in response to an arbitration request to the BOSS. In this case, the BOSS is explicitly granting the node control of the bus and hence this is termed an explicit grant. An implicit grant occurs when a node independently determines that the sub action has concluded and that it is able to become BOSS. In this case it is implied that the node can transmit and hence it becomes BOSS when it begins transmission. After an extended period of inactivity, the root node assumes the role of BOSS.

BOSS arbitration also introduces the ability to pipeline requests. This is made possible by introducing the concept of a phase. Both the isochronous and asynchronous intervals have phases which are independent of each other. These phases oscillate between EVEN and ODD (In the case of a bus reset both phases become EVEN). In this way, the nodes are able to send arbitration requests for the current phase (ISOCH\_CURRENT) or the next phase (ISOCH\_NEXT\_EVEN/ODD). When a node has no more requests for the current phase or the next phase, it transmits a “none” request indicating the current phase (NONE\_EVEN/ODD). This informs the BOSS that the node has no more requests for the current phase.

Figure 5 shows an example of BOSS arbitration with three nodes. This figure illustrates the process described above. It shows the requests being received by the BOSS (1), the data/tokens being transmitted by the BOSS (2) and the three nodes (node A, B and C) transmitting data (3). It also shows



(a) Legacy Cycle



(b) Beta Cycle

Figure 6: Typical Cycle when using Legacy arbitration and BOSS arbitration [12]

who is the BOSS at various points (4), the use of pipelining requests (5) and the change of phase when there are only “none” requests and requests for the next phase being transmitted (6).

BOSS arbitration uses tokens to communicate with the devices on the bus. These tokens are transmitted for a minimum amount of time which ensures that everyone on the bus receives the token. These tokens include: CYCLE\_START\_EVEN/ODD and ASYNC\_EVEN/ODD which are used to indicate the start of the isochronous interval and asynchronous interval respectively. As mentioned, in IEEE 1394a the isochronous interval begins after a cycle start packet and ends when a sub action gap occurs (at which point asynchronous data can be transmitted). When using beta mode signalling, the Isochronous interval begins with a CYCLE\_START\_EVEN/ODD token and a cycle start packet and ends when the BOSS has no more in-phase isochronous requests pending (i.e. only “none” requests and requests for the next phase). When this is the case, an ASYNC\_EVEN/ODD token is transmitted to indicate the start of the asynchronous interval.

Figure 6 (a) shows a typical cycle using legacy arbitration, while Figure 6 (b) shows the cycle structure using BOSS arbitration. This shows the removal of the gaps between sub actions and isochronous packets. Figure 6 (b) also shows the use of CYCLE\_START\_EVEN/ODD tokens and ASYNC\_EVEN/ODD tokens to indicate the start of the isochronous interval and asynchronous interval respectively.

## V. PROPOSED METHOD TO CALCULATE BANDWIDTH IN IEEE 1394B NETWORK

A beta-only network is a network in which all the nodes use beta mode signalling and BOSS arbitration. When calculating bandwidth in a beta-only network, there is no need to include an arbitration time since arbitration is done in parallel with the sending of data. This section takes into account these changes and describes how to calculate bandwidth for an IEEE 1394b beta-only network.

When calculating bandwidth, a knowledge of the packet format is essential to determining the overhead. The packet

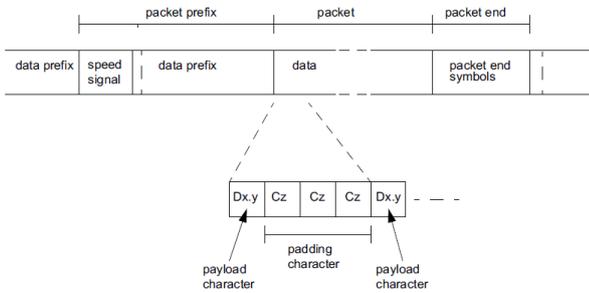


Figure 7: Packet Format

consists of 3 parts - the packet prefix, the packet itself and the packet end.

Figure 7 shows a depiction of the IEEE 1394b packet format. The packet prefix incorporates the speed signal followed by a number of DATA\_PREFIX tokens. The speed signal consists of a number of SPEEDa, SPEEDb and SPEEDc control tokens. These tokens are utilised to indicate the packet speed (relative to the port operating speed) to the receiving node. The use of SPEEDa and SPEEDb tokens also indicates whether the packet format is legacy or beta (which it is in this case). For any particular packet speed the speed signal has a constant duration.

In IEEE 1394b, the port operating speed remains constant for all packet speeds. In the case where the packet speed is less than port operating speed, other characters (padding characters or packet delimiters) are transmitted with the payload characters as indicated in Figure 7. These characters are SPEEDc symbols.

Following the speed signal, a number (p) of DATA\_PREFIX symbols are transmitted. In a beta-only network, p is the ratio of the port operating speed to the packet speed (for all packet speeds). A number (e) of DATA\_PREFIX symbols are also transmitted after the p DATA\_PREFIX symbols to compensate for clock frequency differences. These symbols are called deletable symbols. They are introduced such that the duration is at least 2 symbols at the packet speed (or the duration of 2 symbols at 400 Mbps for packet speeds faster than 400 Mbps).

The packet end consists of a stream of packet end symbols. These include: ARB\_CONTEXT, DATA\_END, DATA\_PREFIX, DATA\_NULL, GRANT, GRANT\_ISOCH. In a beta-only network, a number (n) of packet end symbols are transmitted at port operating speed where  $n=2*g$  (g is the ratio of the port operating speed to the operating speed of the port being transmitted to) if the port being being transmitted to has a port operating speed less than the transmitting port, otherwise  $n=2*m$  (m is the ratio of the port operating speed to the packet speed).

The duration of the packet prefix and packet end in a beta-only network is less than their duration in an IEEE 1394a network. This, along with the lack of idle gaps, are the reasons why there is more bandwidth available for transmitting data in a beta-only network than in an IEEE 1394a network.

Figure 8 shows the isochronous interval from the perspective of two devices - The BOSS and the next node to transmit.

As in the previous section, there are a number of overhead factors which need to be taken into account. These are as follows:

- The time taken to transmit the Speed signal (1)
- The time taken to transmit the Data Prefix (2)
- The time taken to transmit the Data End (3)
- The delay to next transmitting node (4)

These factors are labeled in Figure 8 using the numbers indicated in brackets.

	A	B	C	Total
Data Prefix	81.05	81.05	81.05	243.15
Data End	40.7	40.7	40.7	122.1
Delay to Next	22.73	22.73	22.73	68.18
PHY Delay	144	144	144	432
Total Overhead (ns)	288.48	288.48	288.48	865.43
Total Overhead (BWU)	14.18	14.18	14.18	42.53
Total Overhead ID				2
Overhead ID	1	1	1	3
Accumulated Overhead ID	1	1	2	
Adjusted Overhead ID	1	0	1	2

Table II: Overhead for an IEEE 1394b network

As mentioned, a speed signal is included in the packet prefix. The duration of this speed signal is  $\frac{8000}{x}$  ns where  $x$  is the packet speed in Mbps. There is no information on the PHY delay in the IEEE 1394b spec. It is, however, faster than the PHY delay in a IEEE 1394a node, and varies from manufacturer to manufacturer. For the purposes of this calculation, the same value is used as is used for IEEE 1394a (144ns) since we know that the PHY delay for IEEE 1394b is less than or equal to this value. The packet end consists of a ISOCH\_GRANT which is sent to the next node to transmit. Using the same methodology as the calculation presented in Section III, we can calculate the overhead in a beta-only network.

Table II shows the overhead calculated for a beta-only network using a 3 node network equivalent to the one used in Section III (which is shown in Figure 3). The port operating speed of all nodes is 400 Mbps (as is the packet speed) so the packet prefix consists of a speed signal (20ns) and 3 Data Prefix symbols transmitted at 400 Mbps (61.05ns). The packet end consists of two ISOCH\_GRANT or DATA\_END symbols transmitted at 400 Mbps (40.7ns). There is no sub action gap to factor in, however there is still the overhead for headers - which is 60 BWU (3 channels - 1 per device, 5 quadlets at 400 Mbps). This leads to a total overhead of 124 BWU, which means that there is more than 150 extra BWU available for the transmission of audio relative to IEEE 1394a. 4791.20 BWU out of the 4915.20 BWU are available for isochronous transmission. This equates to 149.73 (i.e. 149) sequences at a sampling rate of 48 KHz and a transmission speed of 400Mps - which is approximately 5 more sequences than when using IEEE 1394a.

## VI. EVALUATION OF THIS METHOD

In order to be able to use the calculations to advise an audio engineer on the current bandwidth available, we desire them to be accurate. To do this, a program (Bandwidth Allocation Program) was created which performed the bandwidth calculation described in the previous section. A further program (Firespy Analysis Program) was created to analyse Firespy (which is a device which can be used to capture traffic on a IEEE 1394 network) output from a real network and calculate the real bandwidth usage. The Bandwidth Allocation Program calculates the amount of overhead (in BWU) which is used by a given network while the Firespy Analysis program takes the output from the Firespy and calculates the amount of bandwidth for each cycle based on the time taken and the number of sequences which are sent in that cycle. To perform tests, an audio network was created that transmitted 128 sequences (using 10 channels - 6 of the channels containing 16 sequences and the other 4 containing 8 sequences) between 2 routers which were both using beta mode signalling and recorded this output using the Firespy. 3 devices on the network were sending 2 channels, each containing 16 sequences, while 2 were sending

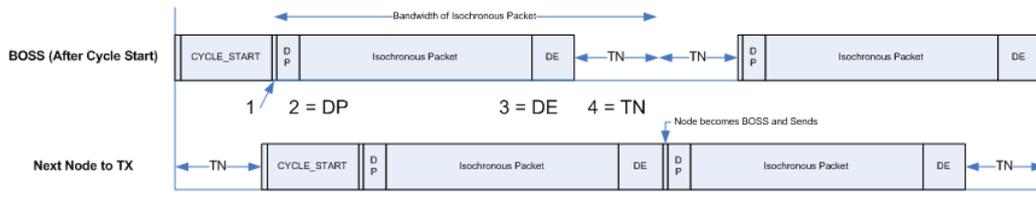


Figure 8: Isochronous Transmission with BOSS Arbitration

	A	B	Total
Data Prefix (ns)	81.05	81.05	162.1
Data End (ns)	40.7	40.7	81.4
Delay To Next Tx (ns)	20.2	20.2	40.4
PHY Delay (ns)	144	144	288
Total Overhead (ns)	285.95	285.95	571.9
Total Overhead (BWU)	14.05	14.05	28.1
Total Adjusted Overhead (BWU)			32

Table III: Bandwidth calculation between 2 routers

2 channels each containing 8 sequences to the first router which was connected to another router (with the transmission between the routers using beta mode signalling).

Table III shows the output from the Bandwidth Allocation Program given a network containing 2 nodes with 4 meter cable between them (which is approximately the length of the cable used between the 2 routers when conducting the tests).

The total BWU used for headers would be 200 BWU (10 channels, 5 quadlets at 400 Mbps). This means that there is a total overhead of 232 BWU. This results in 4683.2 BWU being available for isochronous transmission - which equates to 146.35 sequences at a sampling rate of 48 KHz and transmission speed of 400 Mbps.

The devices sending the audio used DICE II chips [11]. These chips only send data when the buffer is full (8 data blocks). This leads to 32 BWU being used for transmitting audio at a sampling rate of 48KHz. 32 BWU is enough to send audio at a sampling rate 64 KHz every cycle. This means that an empty packet is sent once every four cycles in these tests. This is observed in the Firespy output.

The Firespy Analysis program uses the data captured by the Firespy to calculate the number of sequences which can be transmitted in each cycle. Since the headers have been already been transmitted for the channels being used, the amount of sequences which can be sent in the time remaining needs to be added to the amount of sequences which have already been sent in the isochronous cycle to obtain the number of sequences which can be transmitted in each cycle. Transmitting a sequence at 48KHz requires 32 BWU which is equivalent to 651.2ns (1 BWU = 20.35ns [2]).

The following results are produced by the Firespy Analysis Program:

```

There are 5097 cycles
The maximum used is 73.92 percent of isoch
Based on largest cycle (5071) 152.04237 sequences
can be sent
Overall usage - 63.27 percent of isoch
Based on these results 146.40811 sequences can be sent
The least sequences possible in any cycle is 132.38927
in cycle 1956

```

This output shows 3 results: The overall usage during the captured period, the largest cycle (which contains the most sequences being sent in the shortest time) and the least sequences which can be sent in any cycle over the capture period. These results agree with the results given by the calculation presented in the previous section which calculated that 146 sequences, with a sampling rate of 48KHz, can be transmitted at 400

Mbps between the two routers. We can therefore conclude that the calculation presented in the previous section provides an accurate representation of the activity on the bus.

## VII. CONCLUSION

To ensure successful transmission of audio between devices, the audio engineer needs to be sure that there is enough bandwidth available on the network when making connections. Firewire is one technology which is used within professional audio networks. With the creation of IEEE 1394b, support is provided for longer distances due to a new type of signalling. It also offers full duplex transmission, faster transmission speeds and better use of bandwidth with a new arbitration technique called BOSS arbitration. This paper has described how the IEEE 1394b bus operates and presented a method which can be used to calculate bandwidth for a beta-only bus. It showed that in a simple network consisting of three nodes (connected together with two 4.5m cables), five additional sequences (a 3.4 percent increase) can be sent on a bus when using beta mode signalling and BOSS arbitration rather than data strobe signalling and legacy arbitration. This calculation was also shown to be accurate and agrees with results obtained when analysing the output from a Firespy capture of data transmitted on a real network transmitting 128 sequences on 10 channels.

## REFERENCES

- [1] Don Anderson. *Firewire System Architecture*. PC System Architecture Series. Mindshare, 2nd edition, October 2000.
- [2] Yamaha Corporation. *mLAN Guide for Bandwidth Efficiency*, 2002.
- [3] Digigram. *An Introduction to the ES-100 Technology Rev. 3.0b*, August 2006.
- [4] Philip J. Foulkes and Richard J. Foss. Providing Interoperability of, and Control over, Quality of Service Networks for Real-time Audio and Video Devices. In *South African Telecommunications and Networking Applications Conference*, 2009.
- [5] Kevin Gross. *CobraNet Technology Datasheet*. Technical report, Peak Audio, April 2001.
- [6] IEEE Society. *IEEE Standard for a Higher Performance Serial Bus - Amendment 2*, January 2002.
- [7] IEEE Computer Society. *ISO/IEC 13213 (ANSI/IEEE 1212) - Control and Status Registers (CSR) Architecture for microcomputer buses*, October 1994.
- [8] IEEE Computer Society. *IEEE Standard for a High Performance Serial Bus - Firewire*, January 1995.
- [9] IEEE Computer Society. *IEEE Standard for a High Performance Serial Bus - Amendment 1*, January 2000.
- [10] IEEE Computer Society. *IEEE Standard for a High Performance Serial Bus - Amendment 3*, January 2006.
- [11] TC Applied Technologies. *Digital Interface Communications Engine (TCD2210/2220) Users Manual - Revision 0.14*, February 2007.
- [12] Michael Johas Teener. *Technical Introduction to IEEE 1394*. Available Online: <http://www.teener.com/docs/files/NewTechIntroTo1394.pdf> (Last Accessed 10 February 2009), February 2009.
- [13] A. X. Widmer and P.A. Franaszek. A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code. *IBM Journal of Research and Development*, 27(5):700-707, May 1983.

**Fred Otten** holds a Master of Science Degree in Computer Science from Rhodes University. He is currently reading for a PhD in Computer Science at the same institution and has a keen interest in real-time multimedia and network simulation.