Performance Analysis of an IEEE 1394 Network

Felix Bartholdy *MSLP Europe/1394 Application*

ABSTRACT

IEEE1394 is a high performance serial bus, which combines PC, PC peripherals, multimedia, and industrial applications together in one bus. Two of the most asked questions are: Is 1394 the right choice for my application and what kind of performance can I expect?

This report shows the general architecture, the different transactions down to the signals on the twisted pair lines, and an estimated performance of the bus. Furthermore the report shows how to achieve the best performance with the right gap count and gives some suggestions to optimize the bus.

Contents

Figures

Tables

1 Technical Glossary

Acknowledge packet: An 8-bit packet that may be transmitted in response to the receipt of a primary packet. *Syn:* acknowledge.

Arbitration: The process by which nodes compete for control of the bus. Upon completion of arbitration, the winning node is able to transmit a packet or initiate a short bus reset.

Arbitration reset gap: The minimum period of idle bus (longer than a normal subaction gap) that separates fair-ness intervals.

Arbitration signaling: A protocol for the exchange of bidirectional, unclocked signals between nodes during arbitration.

Asynchronous packet: A primary packet transmitted in accordance with asynchronous arbitration rules (out-side of the isochronous period).

Bus manager: The node that provides power management, sets the gap count in the cable environment, and publishes the topology of the bus and the maximum speed for data transmission between any two nodes on the bus. The bus manager node may also be the isochronous resource manager node.

Concatenated transaction: A split transaction comprised of concatenated subactions.

Cycle start packet: A primary packet sent by the cycle master that indicates the start of an isochronous period.

Fairness interval: A time period delimited by arbitration reset gaps. Within a fairness interval, the total number of asynchronous packets that may be transmitted by a node is limited. Each node's limit may be explicitly established by the bus manager or it may be implicit.

Gap: A period of idle bus.

Isochronous: Uniform in time (i.e., having equal duration) and recurring at regular intervals.

Isochronous gap: For an isochronous subaction, the period of idle bus that precedes arbitration.

Isochronous resource manager: A node that implements the BUS_MANAGER_ID, BANDWIDTH_AVAILABLE, CHANNELS_AVAILABLE, and BROADCAST_CHANNEL registers (some of which permit the cooperative allocation of isochronous resources). Subsequent to each bus reset, one isochronous resource manager is selected from all nodes capable of this function.

Isochronous subaction: Within the isochronous period, either a concatenated packet or a packet and the gap that preceded it.

Link layer: The serial bus protocol layer that provides confirmed and unconfirmed transmission or reception of primary packets.

Listener: An application at a node that receives a stream packet.

Node: A serial bus device that may be addressed independently of other nodes. A minimal node consists of only a physical layer (PHY) without an enabled link. If the link and other layers are present and enabled they are considered part of the node.

Packet: A sequence of bits transmitted on serial bus and delimited by DATA_PREFIX and DATA_END.

Payload: The portion of a primary packet that contains data defined by an application.

PHY packet: A 64-bit packet where the most significant 32 bits are the one's complement of the least significant 32 bits.

Physical layer (PHY): The serial bus protocol layer that translates the logical symbols used by the link layer into electrical signals on serial bus media. The physical layer is self-initializing. Physical layer arbitration guarantees that only one node at a time is sending data. The mechanical interface is defined as part of the physical layer.

Ping: A term used to describe the transmission of a physical layer (PHY) packet to a particular node in order to time the response packet(s) provoked.

Port: The part of the physical layer (PHY) that allows connection to one other node.

Quadlet: Four bytes, or 32 bits, of data.

Request: A primary packet (with optional data) sent by one node's link (the requester) to another node's link (the responder).

Response: A primary packet (with optional data) sent in response to a request subaction.

Self-ID packet: A physical layer (PHY) packet transmitted by a cable PHY during the self-ID phase or in response to a PHY ping packet.

Subaction: A complete link layer operation: optional arbitration, packet transmission, and optional acknowledgment.

Subaction gap: For an asynchronous subaction, the period of idle bus that precedes arbitration.

Talker: An application at a node that transmits a stream packet.

Transaction: A request and the optional, corresponding response.

2 Introduction

Physically an IEEE1394 network is made of point-to-point connections. By repeating the signals, seen on the inputs to all outputs on one node, 1394 becomes a logical bus. So all information can be seen at each cable segment. The topology is free and can be adjusted to the special application.

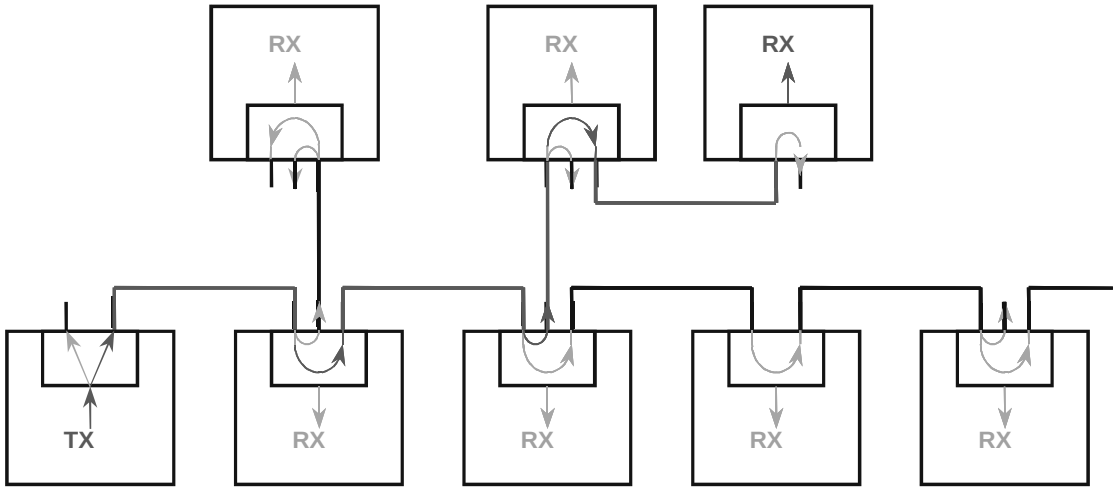

Figure 1. Concept of IEEE1394

Due to the repeater mechanism of each node on the bus, the signals are delayed on the next cable segment. Furthermore idle times are necessary to avoid data collisions. A fair arbitration protocol guaranties a fair access of every node on the bus. These timings, needed for a stable system, differ due to several factors such as the type of transfers (isochronous or asynchronous), number of nodes, and the actual topology. With the knowledge of these factors a performance analysis can be made and an optimized network built.

3 Asynchronous Data Transfers

The general data transmission type for asynchronous transfers is a kind of question and answer game. If one node wants to have information from another node it generates an inquiry (read request). The receiver acknowledges the reception of the packet request. If the receiver has collected the requested data, it sends a read response including the requested data back to the requesting node, which is again acknowledged by the requestor. In the meantime other nodes can use the bus.

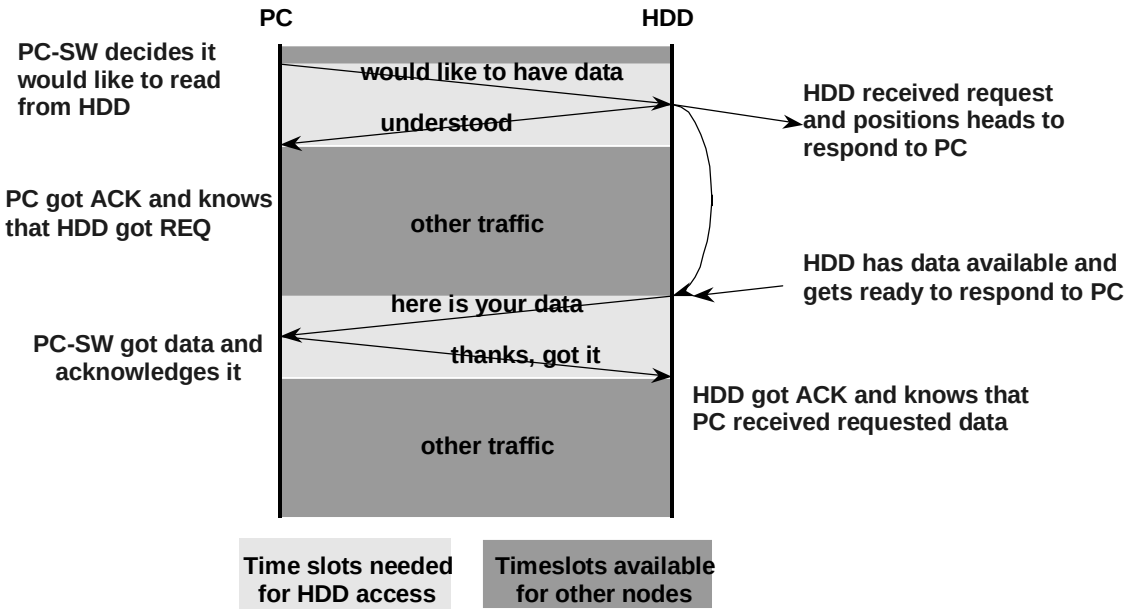

Figure 2. Asynchronous Transaction

The acknowledge packet contains information on the status of the receiver or an error message if the transmission is aborted. For example, the node has successfully accepted the packet (Ack_Complete), or the packet could not be accepted. The node may accept the packet on a retry (Ack_Busy_X). The size of the acknowledge is 8 bits (4 bit data, 4 bit parity).

The maximum packet size depends on the transmission speed and is 512 Bytes at 100 Mbit/s, 1024 Bytes at 200 Mbit/s and 2048 Bytes at 400 Mbit/s.

Table 1. Asynchronous Packet Types

Source	Header	Payload	Destination	Header	Payload
Read request for data quadlet	4Q	0	Read response for data quadlet	4	1
Read request for data block	5	0	Read response for data quadlet	6	.512
Write request for data quadlet	4	1	Write response	4	0
Write request for data block	6	.512	Write response	4	0
Lock request	6	2/4	Lock response	6	1/2

Table 1 shows asynchronous packet types as described in the standard split into header and payload quadlets (1 quadlet (q) = 32 Bit) for both request and response. Beside the listed packet types the standard defines special formats such as:

- Cycle start 4q Header + 1q cycle time data
- Asynchronous Streams 3q Header up to 512q Payload

- Acknowledge 4 Bit Ack 4 Bit parity
- Phy packets 1q Data + 1q Logical inverse
  - Self-ID
  - config
  - power down / resume

Some of these packets don't have a response packet. The cycle start packet, that is a broadcast packet, is received by each node to synchronize the cycle timer register.

4 Isochronous Data Transfer

For the transmission of isochronous data the talker (the node that wants to send the data) needs to allocate the necessary bandwidth and a channel number at the isochronous resource manager (IRM). The isochronous data are transmitted in a broadcast mode so that every node, who is allowed to, can listen. The maximum available bandwidth for isochronous data is restricted to 80 percent of the total bandwidth.

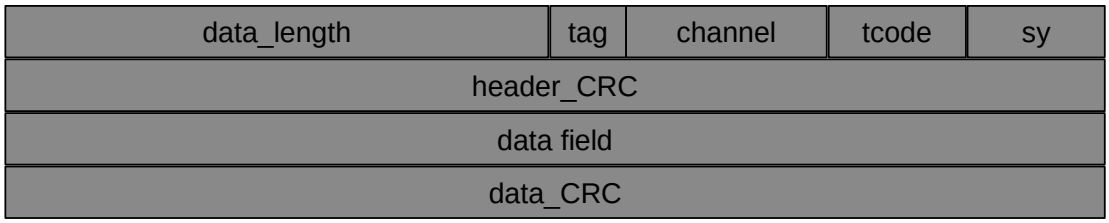

Figure 3. Isochronous Data Packet

Figure 3 shows the standard packet for isochronous transmission. The size of the data field can be up to 1200 quadlets. The header of the packet is just one quadlet. The total overhead including the two cyclic redundancy checks (CRC) is three quadlets.

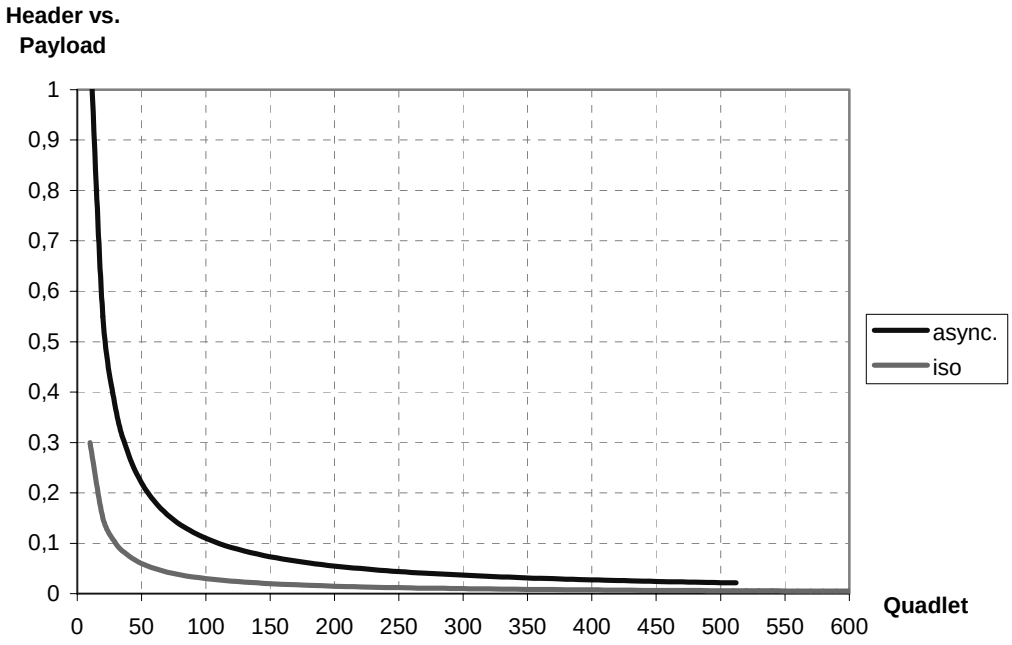

Figure 4. Protocol Overhead on Asynchronous and Isochronous Transactions

The graph shows the protocol overhead of the header vs the payload of a 1394 packet. The larger the payload of a packet, the better the relation between header and data payload.

Asynchronous Transaction:	Read request (5q Header / 0q Payload) + Read response (6q Header / 0 .. 512q Payload)
Isochronous Transacton:	3q Header / 0 .. ~1200q Payload

5 Packet Transmission Times

The complete 1394 packet contains four major parts (see Figure 5). It starts with the arbitration phase (arbitration time) that is followed by the Data_Prefix, which is superpositioned by the speed signaling. The next part is the data packet with the specific header. The Data_End completes the transmission. The time for the arbitration varies, depending on the position in the network of the requesting node to the root node. If both are next to each other (best case), the arbitration time is about 80 ns (2 x Min response time). This time increases with the number of repeaters with 128 ns each (2 x Min Arb_Response_Delay).

The Data_Prefix referring to the 1394a-2000 standard specifies a minimum time of 140 ns, whereas most of the physical layers (PHY's) need a longer time to synchronize the internal DLL to clock the data in. The Texas Instruments PHY devices of the AB family (TSB41ABx) use a Data_Prefix of approximately 700 ns for 100Mbit/s packets and 350 ns for 400Mbit/s. The root of the network can access the bus without arbitrating if the appropriate gap has occurred. The other members on the bus have to arbitrate and to wait until the root grants the access. The time for the arbitration depends on the position of the requestor to the root and is not defined exactly. The limiting factor is the response timeout of 1 ms.

The packet core consists of the data header and the data payload, which varies depending on the kind of transmission (see Asynchronous Data Transfers and Isochronous Data Transfer). The end of the packet is marked with the Data_End, which is equal in all packets. The minimum time is 240 ns and the maximum time is 260 ns.

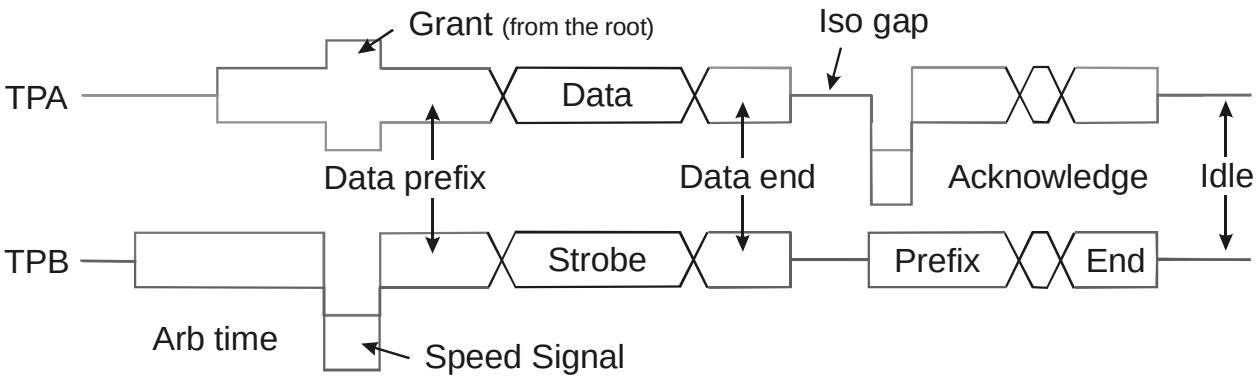

Figure 5. Twisted Pair Signal of a Data Packet

The data packet is followed by the acknowledge from the receiver, separated through the isochronous gap. The receiver does not need to arbitrate for that packet. It can send the acknowledge just after the isochronous gap has occurred. The acknowledge contains 8-bit data. The prefix and data end are the same as for the data packet (see Table 2).

Table 2. Signals on the Twisted Pair Lines

	Comment	Length
TPA /TPB	Twisted pair data lines from the requesting node	
Arbitation time		Depending on the position in the network
Grant	Signal from the root that indicates the won arbitration	Approximately 120ns
Speed signal	Speed indication from the transmitting node	120ns
Data prefix	Packet dependent	140ns – 700ns
Data	Packet / payload dependent	4q – 512q
Strobe	Derivate form data	4q – 512q
Data End	End of packet	240 – 260ns
Acknowledge	Packet confirmation from the destination	8 bit

An additional parameter for the estimation of actual payload on the bus is the duration of the gaps. A shorter or longer gap occurs after each data packet to avoid data collisions. The standard defines different kinds of gaps for different lengths.

Isochronous packets are separated through isochronous gaps, which are 40 ns long. The transition from the isochronous traffic to the asynchronous data phase is marked with the subaction gap. The length of these gaps is substantially determined by the gap count and lasts in best case 280 ns (gap count = 0) and can go up to 10 μs (gap count = 63).

If one node wants to transmit more than one asynchronous packet it has to wait till the arbitration reset gap has occurred. The Arb_Reset_Gap signals the end of a fairness interval. These gaps are substantially determined by the gap count and are in a range of 530 ns (gap count = 0) to 21 µs (gap count = 63).

The gap count affects the efficiency of the network to a high degree and an optimization of the gap count is recommended (see Performance Optimization Possibilities).

6 Payload Estimation

6.1 Point-to-Point Network (asynchronous only)

A point-to-point connection of two nodes is the easiest network topology. Assuming that these two nodes are accessing the bus in the same way, one can estimate the performance. For each transaction a request with an acknowledge and a response with an acknowledge is required (see Asynchronous Data Transfers). Between any transaction, a subaction gap has to occur, and if one node wants to transmit more than one packet, it has to wait for the arbitration reset gap.

Table 3. Packet Times

Read Request for Datablock	Signal	Comment	Length [µs]
	DATA_PREFIX	MIN_DATA_PREFIX	0,34
	HEADDER/CRC	5Q (Bit TIME 2.54ns @ S400)	0,4064
	DATA_END_TIME		0,24
	MIN_IDLE_TIME		0,04
	MIN_DATA_PREFIX		0,34
	1/4 QUADLET	8 Bit	0,02032
	DATA_END_TIME		0,24
Read Response for Datablock			
	DATA_PREFIX	MIN_DATA_PREFIX	0,34
	HEADDER/CRC	6Q (Bit TIME 2.54ns @ S400)	0,48768
	PAYLOAD	1 to 512	
	DATA_END_TIME		0,24
	MIN_IDLE_TIME		0,04
	MIN_DATA_PREFIX		0,34
	1/4 QUADLET	8 bit	0,02032
	DATA_END_TIME		0,24
Gap times			
	SUBACTION_GAP	Gap count 0	0,2848
	ARB_RESET_GAP	Gap count 0	0,5287
	Middle gap	Assuming that every other packet is a arb reset	0,4069
Isochronous			
	DATA_PREFIX	MIN_DATA_PREFIX	0,34
	HEADDER/CRC	3Q (Bit TIME 2.54ns @ S400)	0,24384
	PAYLOAD	1 to 1230q	
	DATA_END_TIME		0,24

Gap times			
	ISOCHROUNUS _GAP		0,04

The calculation of the graph is based on the read transaction, which includes one more header quadlet than the write transaction. For the calculation of the gap times alternating gaps (subaction /Arb_Reset_Gap) are used.

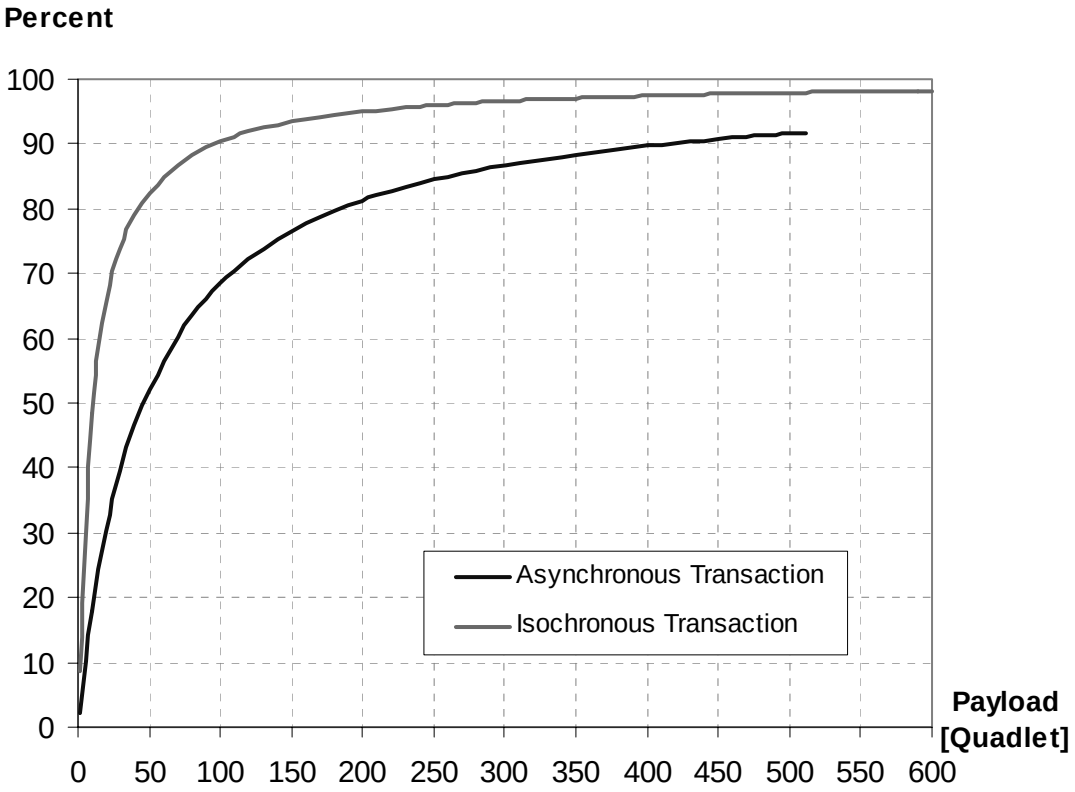

Figure 6. Useable Bandwidth at 400Mbit/s

One can see that an efficiency of 80 percent is reached at a payload size of 40 quadlets for isochronous transmission and 190 quadlets asynchronous. The maximum efficiency is reached at the maximum payload and is 96 percent for isochronous and 92 percent for asynchronous transmission.

6.2 Complex Networks

A detailed statement on the estimated payload of a complex network is nearly impossible. There are many aspects to consider.

The major factor, which influences the data throughput, is the kind of data transfers on the bus. If only small packets are transferred, the gap times, arbitration times, and the protocol overhead become a big part of the transmission time, and the efficiency goes down. On the other hand, the data transfer is very secure because of the CRC checks and the acknowledgements.

The number of hops, which a packet has to pass, is one key parameter on the calculation of the efficiency, because this determines the value of the gap count. Another factor is the position of the requestor to the root, which influences the length of the arbitration signal. The farther away the requestor is from the root, the longer the arbitration time. The time increases approximately 2 x 64 ns at each repeater. The cable length has to be taken into consideration for calculating the delay too. The 1394 standard cables are specified with a propagation time of 5 ns/meter.

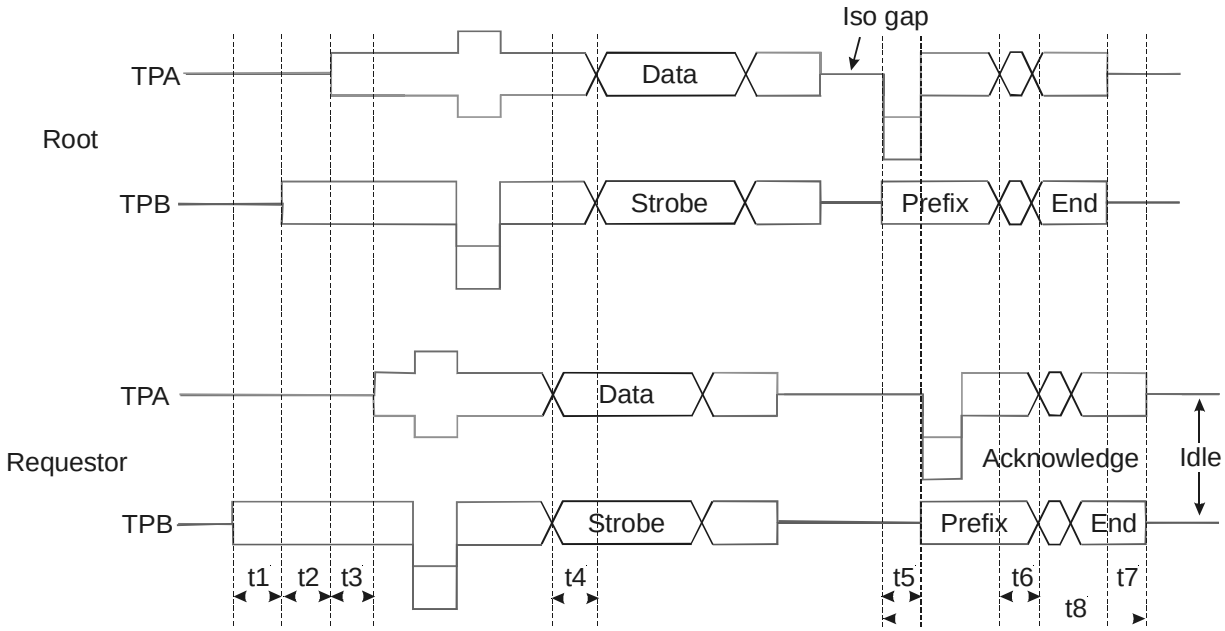

Figure 7. Twisted Pair Signal of a Data Packet Through a Repeater

Parameter	Description	MIN	MAX	Unit
t1, t3, t5	Arbitration response delay of repeater		80	Ns
t2	Time the root needs to detect arbitration und to generate the grant signal	64		Ns
t4, t6, t7	PHY repeater delay		144	Ns

Figure 7 shows how the signals on the twisted pair lines behave if they are transmitted through a repeater. You can see that the arbitration signal (t1) as well as the grant from the root (t3) are delayed through the Arb_Response_Delay of the physical layer. The data signals are added with the 144 ns repeater delay (t4). The cable propagation time of 5 ns/meter has to be added for all signals. The receiver generates the acknowledge if it noticed the Data_End and sends this packet after it has recognized the complete isochronous gap. The acknowledge is delayed by the repeater again through the Arb_Response_Delay (t5) as well as the data through the repeater delay (t6) so that the total delay of the Acknowledge packet is the repeater delay (t8).

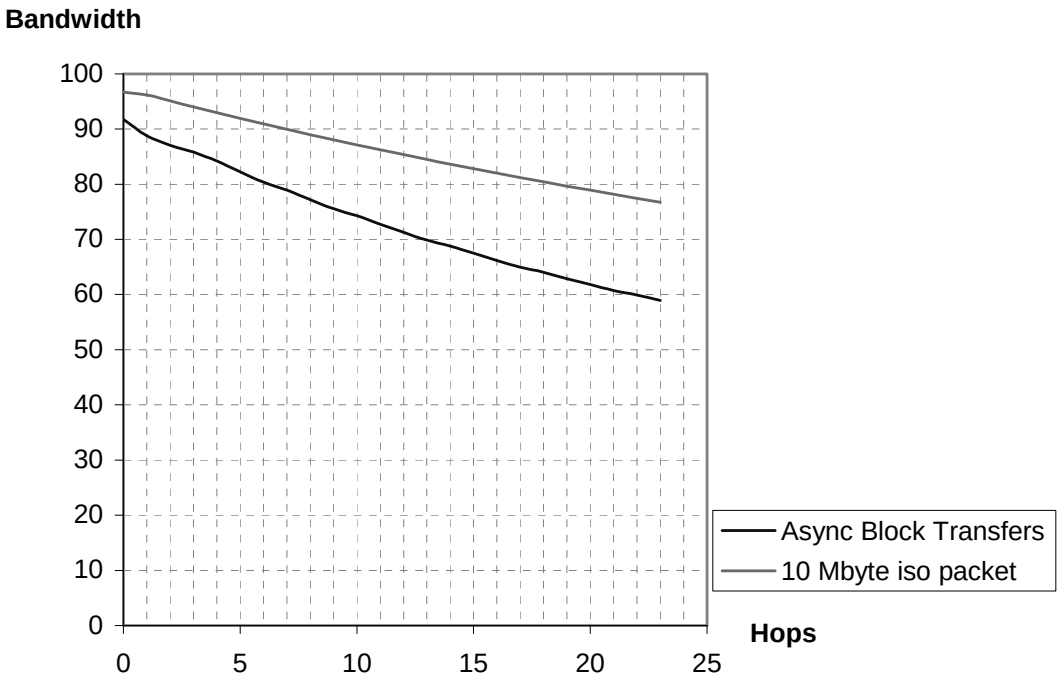

Figure 8. 2k Byte Block Transfers

The graph in Figure 8 displays the bandwidth as a function of hops. The calculation is based on a complete read transaction at 400 Mbit/s with a payload of 2k bytes including read request, read response, the acknowledges of these packets and the gap counts, (see Table E1 of IEEE1394-2000). As an example for the isochronous traffic a payload of 10 Mbyte is refined. This graph shows exclusively the data transfer of the two ends of the network, which is the worst-case scenario, because these two nodes have the longest arbitration times. Through all these assumptions, the graph can only be seen as an estimated calculation of the bandwidth with the influence of hops. In a normal network, the data traffic would be distributed, which results in a better efficiency.

The IEEE1394a-2000 Standard defines a few special features like packet concatenation, which affect the transmission. A node that gets a request can send the response to it directly, without waiting for the Arb_Reset_Gap to occur. Due to the complexity of the different possible configurations of the network, this kind of transmission is not included in this report, but it improves the performance of the bus.

7 Performance Optimization Possibilities

The following example shows the topology influence of the data payload on the bus. If a network consists of five nodes, one can build several different topologies in which the position of the root can differ as well. In the first topology example (Figure 9), the nodes are daisy-chained, the maximum distance of two nodes is through three repeaters between node 0 and 2. If node 0 wants to send a packet, it has to arbitrate for the bus to the root. The root always has the highest number in the network (in this case #4). The arbitration signal now needs to go through the node between #0 and #4 before the root can grant the access. Each hop delays the arbitration signal by 64 ns. After node 0 gets the grant it sends its data, which is delayed through each PHY by 144 ns (plus individual PHY delay). The maximum distance of two nodes in example two (Figure 10) is through one repeater and every node has a direct connection to the root. This reduces the arbitration time as well as the delay through the repeater to a minimum.

Figure 9. Example 1: Network of 5 Nodes (gap count = 8)

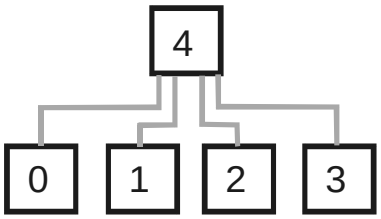

Figure 10. Example 2: Network of 5 Nodes (gap count = 5)

An optimization of the gap counts can be done through the methods described in the Annex E of the IEEE1394a-2000 standard. One possible method to optimize the bus through the bus manager is to count the maximum hops a signal needs to pass in the network. Therefore, a gap count of 8 in the first example and a gap count of 5 in the second can be set, which results in a 500 ns shorter sub action gap and a 1 us shorter arbitration reset gap of the second topology. More obvious are the numbers of example three (Figure 11) and four (Figure 12) where 11 nodes are chained with a minimum gap count of 26 and 37 nodes with a gap count of 8, which results in a 3 µs shorter sub action gap and a 6 µs shorter arbitration reset gap, although the network four contains 26 more nodes.

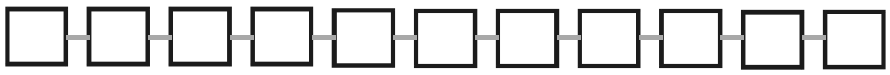

Figure 11. Example 3: Network of 11 Nodes (gap count = 26)

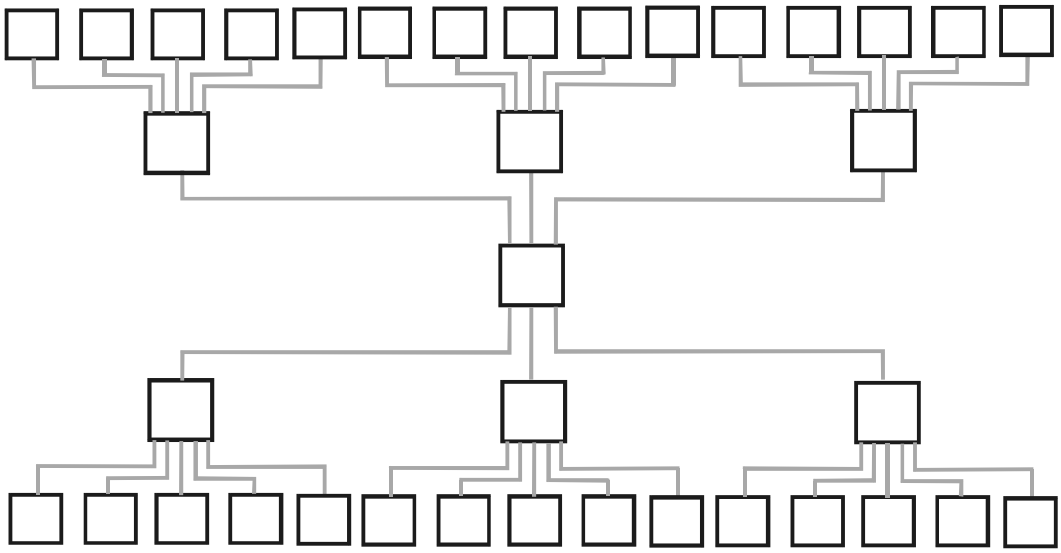

Figure 12. Example 3: Network of 37 Nodes (gap count = 8)

The figures in the table are based on 4.5 m between two nodes and a repeater delay of 144 ns.

A second method is the calculation of the round trip based on a ping packet, that is determined by the bus manager. Each PHY attached to the bus will generate a self-ID packet after it receives the ping packet. The bus manager now can measure the exact time delay between sending the ping packet and receiving the self-ID of each PHY and calculate the gap count. In this number the actual cable length and the individual PHY delays are included.

Another possibility to optimize the 1394 network is to analyze the self-ID packets through the bus manager, because the self-ID contains information on the number and status of the ports and the transmission speed of each node. If for example, a S100 node is located between two S400 nodes the speed through the S100 node can only be S100. The bus manager now can propose the user to reconfigure the network by putting the low speed device at the end of the network. The BM can as well propose a topology change in view of the gap count by using unconnected ports to optimize the round trip (see Figure 9 / Figure 10).

The more ports a node provides, the more efficient the network can be.

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, license, warranty or endorsement thereof.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations and notices. Representation or reproduction of this information with alteration voids all warranties provided for an associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Resale of TI's products or services with *statements different from or beyond the parameters* stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Also see: Standard Terms and Conditions of Sale for Semiconductor Products. www.ti.com/sc/docs/stdterms.htm

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265