

Evolutionary Switched Serial Studio Interface: ESSSI

Submission based on IEEE 1355 to EBU/SMPTE Request for Technology on “Networking and Protocols for Audio, Video and Metadata Transfers”

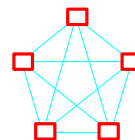
Paul Walker

4Links for technical help, paul@walker.demon.co.uk, +44 1908 566253

30 October 1997

Contents

1. INTRODUCTION.....	2
1.1 Benefits of evolution with 1355.....	2
2. BUILDING ON EXISTING STUDIO DIGITAL NETWORKS.....	3
2.1 Distinguishing between Crossbar and Packet-switched “TV lines”.....	5
3. IEEE 1355 PROTOCOLS.....	7
3.1 FIFO architecture and packet structure.....	7
3.2 Flexible embedding of other protocols.....	8
3.3 Routing packets through a network.....	9
3.4 Hardware interpretation of routing headers.....	10
3.5 Setting the routing tables.....	12
3.6 Out of band signalling.....	12
4. MAPPING 1355 DATA AND CONTROL CHARACTERS ONTO SDI.....	13
5. HIGHER LEVEL SOFTWARE FOR NETWORKING.....	14
6. NETWORK PERFORMANCE.....	14
7. CONCLUSIONS.....	18
8. APPENDIX A: PLUG AND PLAY.....	19
9. APPENDIX B: ANALOGY WITH MODERN MANUFACTURING.....	20



1. Introduction

There are two significant respects in which the existing digital standards used in studios, based on CCIR 601/656, exceed the performance of computer networks:

- The bandwidth-distance product of 270Mbits/s times 300 metres;
- The real-time response from the use of circuit-switched, rather than packet or cell switched, routing switches. And these packet/cell switches themselves far exceed the performance of one-dimensional topologies such as buses and rings.

It is suggested, therefore, that these benefits of the TV industry's existing technology and investment are retained and that a computer style network is built as an evolutionary extension to the existing networks.

Of the computer style networks, the one which offers most flexibility in terms of packet structure and network topology, and which has proven real-time response, is IEEE 1355 (see <http://www.1355-association.org>). This standard at present covers short distance connections (of order of 10 metres) at 100 Mbits/s and 1Gbit/s. The 1355 standard has no Physical layer defined for 270Mbits/s or 360Mbits/s and only achieves 300 metres at 125 MBaud with fibre.

So this submission suggests that the character level and packet structure of 1355 are merged with the physical layer, and with the circuit-switched protocols, of CCIR656/SMPTE 259M.

1.1 Benefits of evolution with 1355

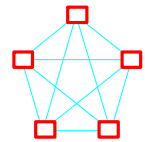
For **TV Studios**: No need to change infrastructure, no need to change all the equipment at once, incremental change as required, known and guaranteed QoS;

For **TV Studio Equipment** manufacturers: Incremental change to existing products instead of complete redesign, predictable performance;

For the **Computer Industry**: A network which meets real-time constraints;

For **Real-Time Control**: A computer network which meets real-time constraints;

For **IEEE 1355**: A physical layer for 270Mbit/s over 300 metres of copper, an Isochronous mechanism better than those of all other networks.



2. Building on existing studio digital networks

To show evolution from existing practice, and to offer the opportunity to correct the author's misconceptions, a simplified view of a studio is shown in Figure 1.

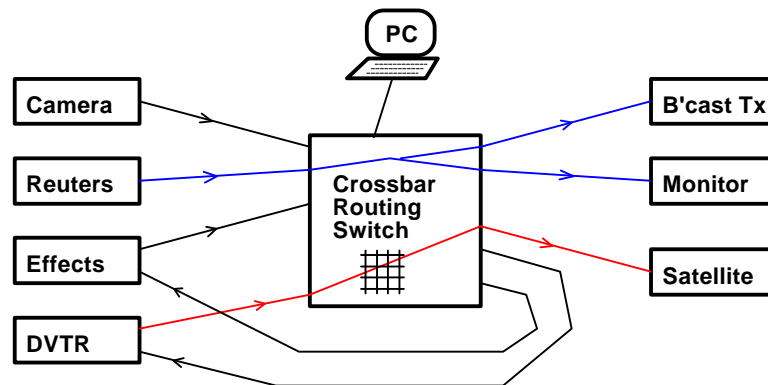


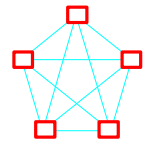
Figure 1: Simplified view of current Digital Studio Network

Figure 1 shows a number of sources of digital TV signals connected to a number of receivers of the signals via a Crossbar Routing Switch controlled by a PC (or computer in an editing control station). Signals can be routed from one source to several receivers, and several independent paths through the switch may be connected at any time.

Traffic in the network is essentially unidirectional, but a variety of equipment in the studio can either transmit or receive and so needs connections to both the inputs and outputs of the crossbar routing switch. For example, some pieces of equipment, such as servers and effects boxes, need to receive from the crossbar switch at the same time as they transmit to the switch. Other pieces of equipment, such as monitors, need only the outputs from the switch. A camera only needs to transmit to the switch, but if the camera is remote controlled, some channel is needed in the reverse direction to provide this control.

There is inevitably a short delay, of maybe nanoseconds but possibly milliseconds, between the PC changing the connections in the routing switch and the connections actually being made. But once made, the full bandwidth of the serial links is available even if every input to the switch and every output from the switch is in use at the same time. On a switch with 64 inputs and 64 outputs, this is a total throughput of 17 Gbits/s, with no delay while the data goes through the switch, and with inexpensive switches.

This combination of throughput, latency and cost is outstanding, and is unlikely to be matched by computer networks for the foreseeable future. So this proposal retains the existing infrastructure of cabling, the existing sources and receivers of digital signals, the existing SDT Digital Data Stream, and the existing crossbar switching. It adds to these a set of computer-network style protocols and an additional packet switching capability in parallel with the circuit-switched crossbar.



All that is necessary, therefore, to upgrade to the proposed new digital studio system is to replace the crossbar routing switch with a new switch capable of packet switching as well as circuit-switching. All the existing studio equipment will work with the new switch. As equipment is upgraded, it would incorporate the new protocols and be able to use them concurrently with the existing Digital Data Stream protocol.

Figure 2 is changed from Figure 1 to show a combined Crossbar and Packet switch, with the same traffic as in Figure 1 going through the crossbar and no traffic through the packet switch.

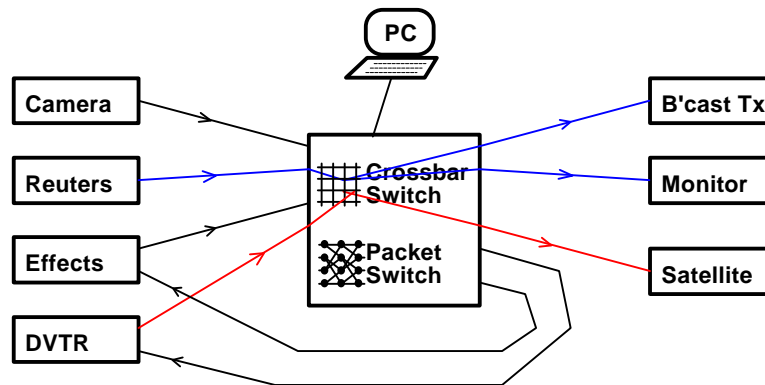


Figure 2: Studio Network with combined Crossbar and Packet switch

Figure 3 shows the same traffic as Figure 2, with the addition of a conversation between the PC and the effects box, via the packet switch, with the traffic in each direction shown as a different shade of green:

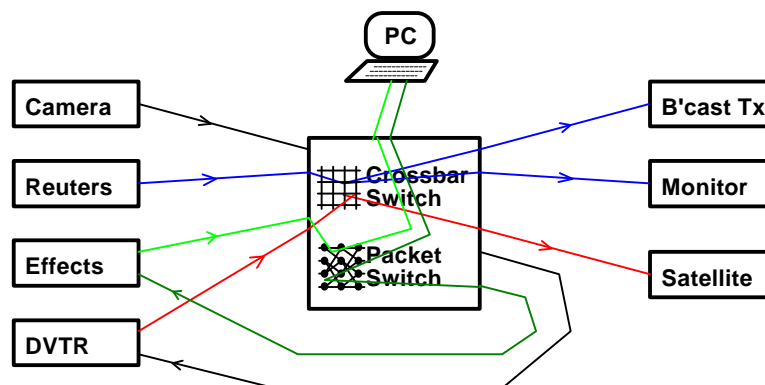


Figure 3: Use of Packet Switch in parallel with Crossbar Switch

Figure 3 shows some sources and receivers which do not have full-duplex connection with the crossbar switch, and these can be retained with the new switch. Indeed it may be more cost-effective for early versions of the new switches to have more crossbar ports than packet switched ports. Figure 4 shows an example of such a switch, with the full-duplex ports connected to both the crossbar switch and the packet switch, and the unidirectional ports connected just to the crossbar.

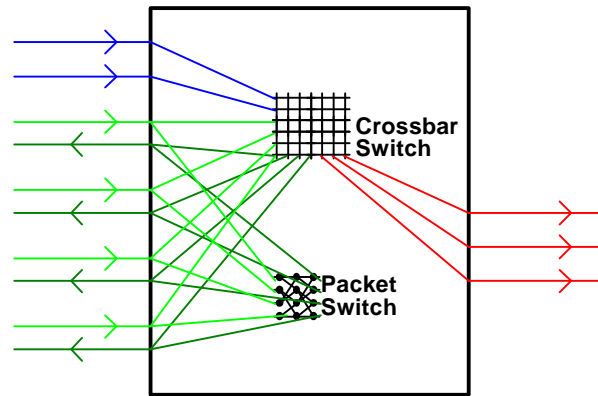
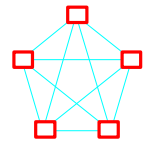


Figure 4: Unidirectional ports connected to Crossbar Switch, and full-duplex ports connected to both Crossbar Switch and Packet Switch

To make it possible for this combined switch to work, we need to define:

- a means of distinguishing between data for the crossbar switch and data for the packet switch, so they each take the correct path through the combined switch box.
- a packet switching protocol for the routing switch

To distinguish data for the crossbar-switch and for the packet-switch, we'll evolve the line header that follows the Timing Reference Signal (TRS). For the packet-switching protocol, we'll suggest the simplest protocol, that of IEEE 1355, whose simplicity should result in low-cost and good performance.

2.1 Distinguishing between Crossbar and Packet-switched "TV lines"

The format of the line header (from Tancock, Sony "Making the Right Connections, an overview of the serial digital interface") is currently:

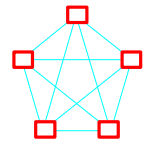
Timing Reference Signal	Line ID	Aux Data Flag	Data ID	Data Block No	Data Count	User Data 255 words maximum	Check sum
-------------------------	---------	---------------	---------	---------------	------------	--------------------------------	-----------

To preserve complete compatibility with current hardware, if a line header is received which meets the current definition of this for TV lines, the header and what follows until the next TRS is sent to the crossbar switch.

For several of these fields, a rather small number of values is currently defined, and alternative values would be used to generate the following Line Headers for packet-switched data.

Timing Reference Signal	"The line following this line header is packet-switched data and there is currently no packet in progress on this line"	Check sum
-------------------------	---	-----------

Timing Reference Signal	"The line following this line header is packet-switched data and this line header is inserted in the middle of a packet"	Check sum
-------------------------	--	-----------



Care must be taken in the allocation of the fields of these headers that it is very unlikely that an erroneous line header would be interpreted as a valid header of the wrong variety.

(Comment: This definition preserves compatibility, but does so by making a complete line either crossbar switched or packet-switched. In view of the trend to make the Serial Data Interface faster (360 or 540 Mbits/s), while it is used at 270, 177, and 143 Mbits/s, or at 50Mbits/s for compressed video, there may be value in interleaving crossbar and packet-switched data within a "Line". For example a 270Mbit/s crossbar-switched signal on a 360Mbit/s Interface would use three out of every four bytes, while the packet switched data would use the remaining one out of four. The result would be immediate response as at present for the crossbar-switched data, with any spare bandwidth available, with minimal added delay, for packet-switched data. If there is interest in this interleaving approach, it will be developed and defined in more detail in the next draft of this description.)

Figure 5 shows a possible arrangement of logic for separating the data for the crossbar-switch from the data for the packet-switch. Parallel data is examined by the Demux logic and the data path is either enabled to the crossbar switch or to the packet switch. Figure 5 actually shows the serial regenerated signal going to the packet switch, because it is much easier to make single-chip packet-switches with many ports if each port uses as few pins as possible, and the fewest pins result from the use of serial data. If the output from the packet-switch is also serial, it would need to be converted to parallel, perhaps with another STV1602A before being multiplexed with the data from the crossbar switch.

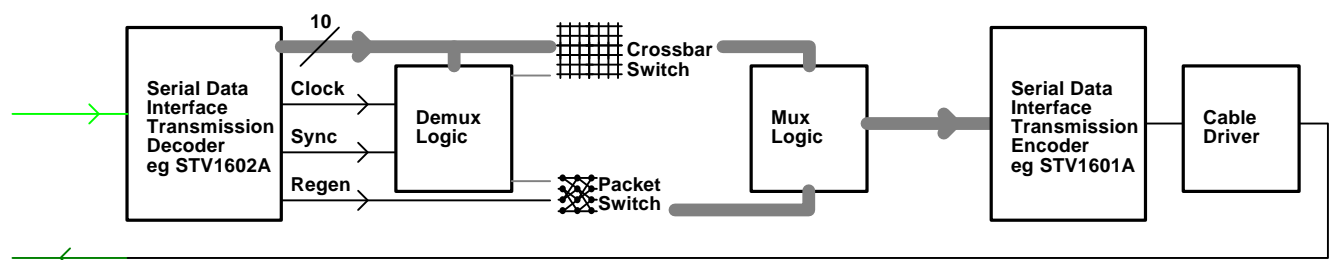
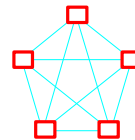


Figure 5: Logic arrangement for Demux and Mux between Crossbar Switch and Packet Switch

In fact the RCube router chip for 1355 is an eight-port full-duplex switch chip with each port capable of 1Gbit/s of data. Two of the RCube ports are byte-parallel on both input and output, and these byte-parallel ports could be connected almost directly to the parallel signals from the SDT decoder and encoder.



3. IEEE 1355 protocols

So far, we have described the combination of crossbar switching and packet switching, but not described the packet-switching protocols. It would, of course, be possible to use a variety of protocols, such as FibreChannel or IP or ATM, or the IP subset proposed for SDTI. But as IP switches are being currently built from ATM switches, and companies are designing ATM switches from 1355, and other companies are designing IP switches from 1355, there are advantages in using 1355 for the studio network directly.

Compared with IP, or FibreChannel, or ATM, 1355 is extremely simple. It just concerns itself with transferring packets through a network as simply, as cheaply, and as fast, as possible. Some readers may as a result feel that elements are left out of the protocols. They are.

(Comment, with reference to Sony SDTI proposals: Is it really necessary for every packet sent from one camera to one DVTR in the studio to have a header that can address every TV set (and toaster!) in the world? And is it really necessary for that toaster in Basingstoke to be able to reply to that camera in a studio in Atlanta?)

As a result of 1355's simplicity, it has been possible to build, in 1993 silicon technology, packet routing switches with 32 ports at 100Mbits/s or 8 ports at 1Gbit/s.

An appendix includes part of the characterisation done by CERN of these switches for use in real-time data acquisition.

3.1 FIFO architecture and packet structure

The easiest point at which to start describing the protocols of 1355 is the idea of a FIFO (First-In, First Out memory) between two chips. One chip can send the other as much data as it wants as fast or as slowly as it wants, and the other chip receives the data at its own speed. If the FIFO becomes full, the sender knows that it must not send any more until space becomes available in the FIFO. If the FIFO becomes empty, the receiver knows that it must wait for more data.

Of course data needs to go in both directions between the two chips, so there would be two FIFOs, one for each direction, as shown in Figure 6.

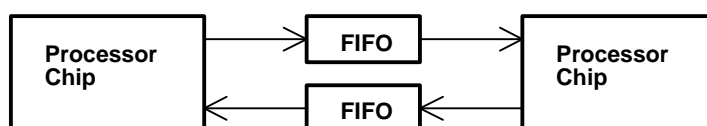
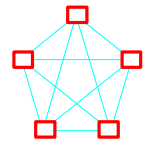


Figure 6: A FIFO interface between two processor chips

The architecture of 1355 uses exactly this model of the bi-directional FIFO, and indeed could use conventional parallel FIFOs. In practice, all the current



versions of 1355 use serial link interconnections, with the FIFOs distributed between the two chips as shown in Figure 7. The serial links are full-duplex to provide a return path that indicates the FIFO has space for more data.

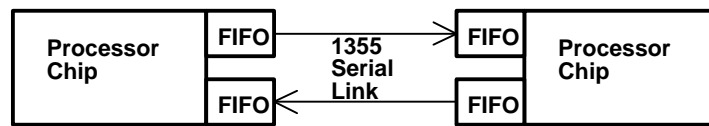


Figure 7: Distributing the FIFO between the chips and connecting them with a serial link

(Comment: The return path might be seen as an inefficiency for a TV studio where most of the traffic is unidirectional. Many items of studio equipment, however, such as DVTRs and effects boxes, already have input ports and output ports, so these would fit the full-duplex connection very easily. The advantage of the feedback flow-control is that data can be sent as fast as possible, and if either the network or the receiver is unable to keep up, signals are effectively sent back to the transmitter to slow down, without any data being lost. This is in sharp contrast to the current case where, for example, a transfer from DVTR to Disk File Server has to be limited to the speed of the server when it is at its busiest, even if the server is doing nothing else but serve that DVTR. With flow-control the transfer goes as fast as the devices can handle at the time. 1355's flow-control is also in sharp contrast with other protocols which "push" and hope that the buffering at the other end of the wire does not overflow: this results in large buffers at every point in the network, and a long control loop delay resulting in sluggish response.)

1355 provides for building networks, and so assembles the data to be sent into packets, with a header and an End of Packet (EOP) terminator as shown in Figure 8. (In this figure and subsequent figures, the packet is shown with front at the left, as on a 'scope trace, to match the EBU/SMPTE diagrams for the Timing Reference Signal and subsequent header. In other 1355 literature, the packets are usually drawn with the header starting at the right.)

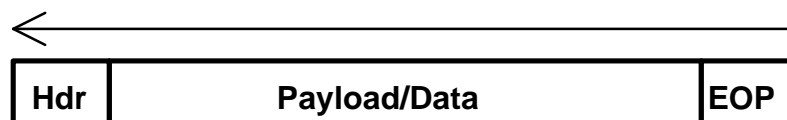
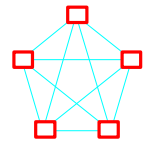


Figure 8: The simple and flexible packet structure of IEEE 1355

3.2 Flexible embedding of other protocols

Beyond the definition of the link forming a distributed FIFO, and the packet format shown in Figure 8, there is a huge freedom in the sort of packets that can be used in 1355 as shown in Figure 8.

For example the header can be a byte or two for addressing a destination, or a particular software process at a destination processor, or it might be a 32-bit or 64-bit memory address, or it might be the five byte header of an ATM cell.



Similarly there is no constraint imposed by 1355 on the packet length, which could be a 4 Mbyte uncompressed image or a 4 Kbytes disk sector or a 188 byte MPEG frame or a 48 byte ATM payload or a single byte command to say "please switch on the lights". Some examples are shown in Figure 9.

In an even more extreme case, the packet could be infinitely long, simply by not appending the EOP terminator. (Comment: This would in fact provide the circuit switching required by EBU/SMPTE, but that would mean changing all the studio equipment to use this protocol, so there are advantages in retaining the existing crossbar circuit-switching standards.)

There may be performance reasons for imposing limits on the length of a packet, but the flexibility of 1355 makes it easy to embed a wide variety of other protocols within the basic structure.

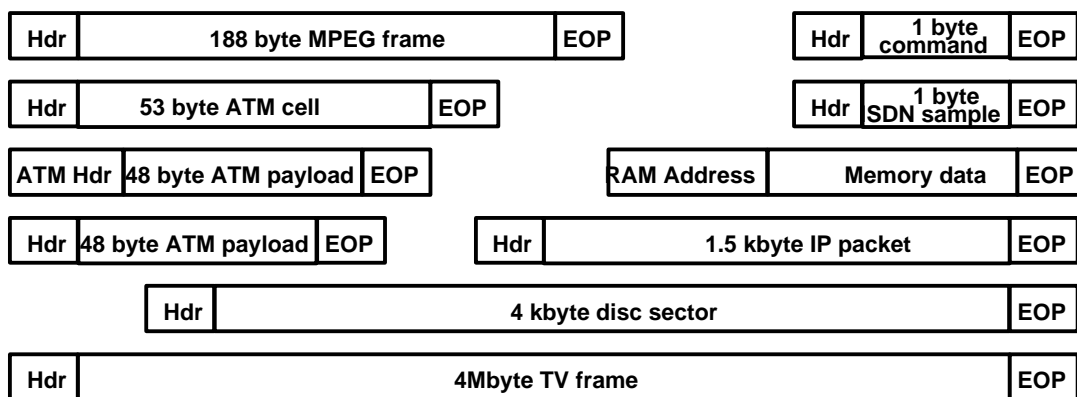


Figure 9: Some examples of possible packets carried on an IEEE 1355 network

3.3 Routing packets through a network

If the header and its handling are simple enough, the through-routing function can be performed by a hardware routing switch, as shown in Figure 10. The headers of 1355 are simple enough and make it possible to build routing switches with a large number of ports. For example the STC104 from SGS-THOMSON has 32 full-duplex ports plus a couple of control ports.

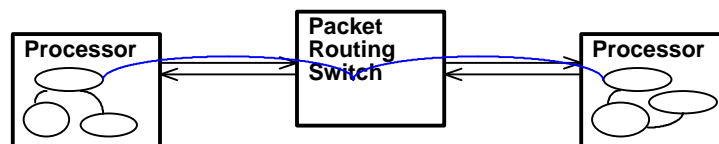
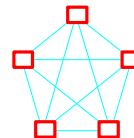


Figure 10: Channel connected to remote processor via a routing switch chip

In terms of the TV studio and the Bit-streams activity, the Header and EOP terminator of the 1355 packet are a "wrapper" for any form of content which may itself be packaged within another layer of wrapping as defined for bit-streams. The Header and EOP "wrapper" are simply there to guide the packet through the network. They carry no excess baggage to make the routing switches more expensive or slower than necessary.



3.4 Hardware interpretation of routing headers

The simplest use of the routing header is that it addresses a particular output port of the switch. So a Header Value of five would be output on output port five, and a header value of 42 would be output on output port 42. Having been used in this way, the Header Value is of no more use and so the byte is stripped.

Figure 11 shows the physical addressing and header stripping.

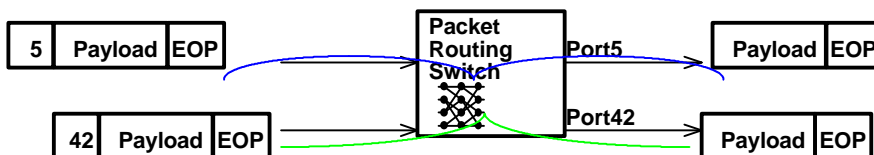


Figure 11: Use of header byte to address output port directly

Physical addressing as described here is used on the ICR C416 chip, and in logic designs of 1355 switches from IC-Routing Ltd and 4Links.

In the TV station, there may be a routing crossbar in Sports and another in News, and the two need to be connected together, at least temporarily, for some broadcasts. The header stripping shown in Figure 11 offers a convenient way to do this. The switch in Sports strips its header, but leaves another header byte which serves to route the packet through the News switch, as shown in Figure 12.

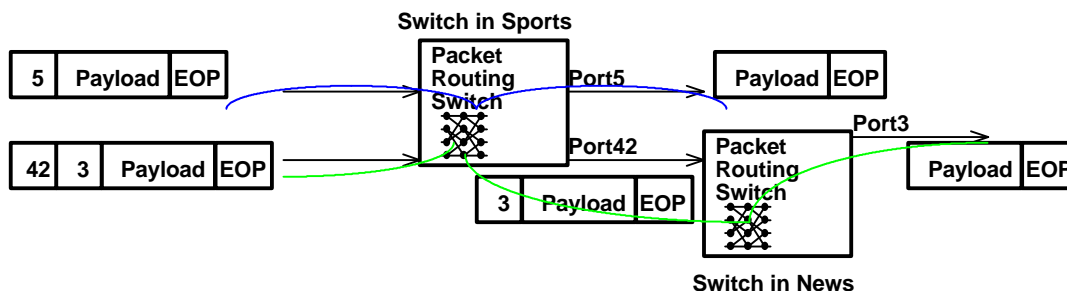


Figure 12: Different header bytes used in different switches

Of course using a byte for the header gives enough bits to address 256 ports, and there may not be any switches with so many ports. Even with switches in Sports and News and Features and Drama, etc., there may be no more than 256 switch ports on the site. So some efficiency can be gained by using a single header to trace a path through several switches. An example is shown in Figure 13.

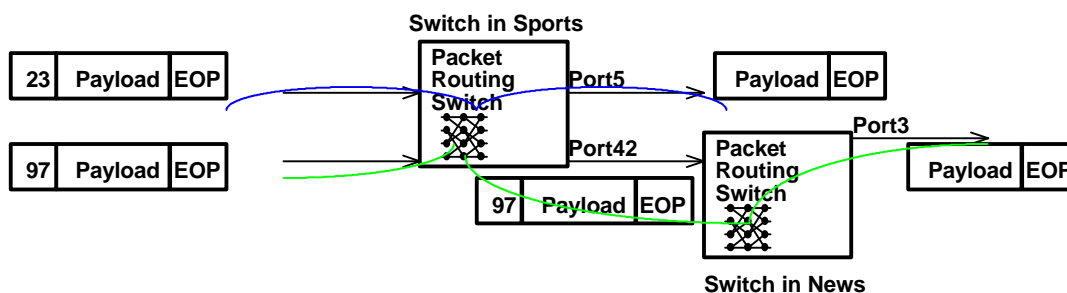


Figure 13: Reducing the header overhead, by using the same header in different switches

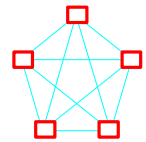


Figure 13 shows the same connections as Figure 12, but with a single byte header. The first switch is set to output a packet with a Header Value of 23 on Port 5, and to output a packet with a Header Value of 97 on port 42. The second switch is set to output a packet with Header Value 97 to Port 3. When the routing header has completed its usefulness, the header is stripped as in Figures 11 and 12.

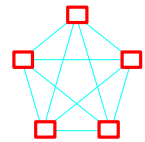
The existing 1355 routing switches, both at 100Mbits/s and at 1Gbit/s use this efficient technique, described as “interval labelling” to route packets through any number of switches in a network.

One of the requirements (or at least requests) for the studio is that bit-streams can be sent from a single source to multiple destinations, so called “multicast”. The existing 1355 routing switch products do not have this capability, but it can be added simply and protocols were defined in 1996 to provide it. Since then a switch has been designed for the protocols and the design could be adapted to the needs of the studio network.. The mechanism used by this design has routing tables in RAM with a word for each permissible Header Value. Each of these words has one bit for each output and an extra bit to say whether the header should be stripped in this switch. Figure 14 shows a small such table, with values 1 to 8 showing physical routing, values 9 to 13 showing interval labelling, and the remaining values showing multicast.

	Bit 0, Output Port 0	Bit 1, Output Port 1	Bit 2, Output Port 2	Bit 3, Output Port 3	Bit 4, Output Port 4	Bit 5, Output Port 5	Bit 6, Output Port 6	Bit 7, Output Port 7	Bit X Don't Strip Header
Header byte = 1	1	0	0	0	0	0	0	0	0
Header byte = 2	0	1	0	0	0	0	0	0	0
Header byte = 3	0	0	1	0	0	0	0	0	0
Header byte = 4	0	0	0	1	0	0	0	0	0
Header byte = 5	0	0	0	0	1	0	0	0	0
Header byte = 6	0	0	0	0	0	1	0	0	0
Header byte = 7	0	0	0	0	0	0	1	0	0
Header byte = 8	0	0	0	0	0	0	0	1	0
Header byte = 9	1	0	0	0	0	0	0	0	1
Header byte = 10	1	0	0	0	0	0	0	0	0
Header byte = 11	1	0	0	0	0	0	0	0	1
Header byte = 12	0	1	0	0	0	0	0	0	1
Header byte = 13	0	1	0	0	0	0	0	0	1
Header byte = 14	1	1	0	0	1	0	0	1	1
Header byte = 15	0	1	0	0	0	0	1	0	1
Header byte = 16	0	0	1	1	1	0	0	1	1
Header byte = 17	1	0	0	1	0	0	0	0	1
Header byte = 18	1	1	1	1	1	1	1	1	1

Figure 14: Example of Routing look-up table for header values

(Some of the capabilities and characteristics shown in Figure 14 are covered in a patent application from 4Links with Keele University.)



The 1355 routing switches have a number of other features to improve the performance of network, for example by providing alternative paths when another path is blocked by other traffic. These techniques can be described, if necessary, in later versions of this description. They are possible at reasonable cost because the 1355 protocols concentrate on what is necessary for optimising routing switch chips, omitting any unnecessary function which adds cost and complexity.

3.5 Setting the routing tables

The connections in the existing routing crossbar switches are controlled by a computer/PC connected to the switch.

The tables in the packet routing switch are similarly controlled by a computer/PC, or they may indeed be initialised to a simple set physical routes. As the header determines at the time of the packet what the path actually is through the switch, the computer can set all the routes statically, so the computer's job can be easier than its job for the crossbar switches. In some case, for example security, it may be preferable to set up only currently authorised routes. In this case the computer's job is very similar to the job it performs for the crossbar switch, but it has additionally to tell the sources what headers to use.

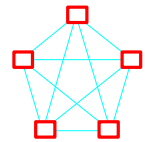
3.6 Out of band signalling

The existing CCIR 656/SMPTE 259M includes The TRS Timing Reference Signal which is a unique data pattern that can not be generated by normal TV data.

Similarly, the 1355 protocols need a small number of control characters that are unique from the header and payload of the packets. These special characters are used at two levels of the protocol, privately between the hardware at two ends of a full-duplex connection, and at the packet layer for which the control characters must be passed on transparently by the switches.

The control characters used privately between the two ends of a cable are:

- **FCC**, Flow Control Character, to indicate that the receive FIFO has space for a further "so many" characters. The 1355 standard allows choice of the actual amount of buffer space indicated by the FCC, depending on the actual physical layer used. For a physical layer based on 656/259M, the round-trip delay at 300 metres is 3 microseconds, which at 270Mbits/s is 81 bytes. So setting "so many" characters or the Flow-control-unit (Flit) to 32 bytes and a recommended minimum receive FIFO size of 128 bytes should ensure that the transmitter is never slowed down by cable delays.
- **NUL**, if there is no data to send, or if the receiver can accept no more, the cable must still be driven with something so that the AC-coupled circuits continue operating.



Some of the alternative 1355 physical layers also include other characters for initialisation. In this case, with the circuits being driven by the TV signals, such additional characters may not be necessary. More likely is that a character or sequence of characters will be required for error recovery. A suitable set is the characters used by the 1355-HS-Links (1Gbit/s).

The control characters which are transmitted transparently through switches are:

- **EOP**, End of Packet.
- **EEOP**, Exceptional End of Packet. This is used by a routing switch to indicate that an error occurred during the packet and so the data and/or packet length may be invalid.

Note that there is no marker of Start of Packet. After an EOP or EEOP the next data character is the first byte of the header of the next packet.

4. Mapping 1355 Data and Control Characters onto SDT

In the simplest form of IEEE 1355, the control characters are indicated by a flag bit to indicate whether the following bits are control or data. In the other forms of 1355, the data is block coded, and spare codes in the block codes are used for control characters.

For this ESSSI proposal, the most important aspect of the encoding of control characters is that it should be extremely improbable for bit errors to result in control turning into data or data turning into control.

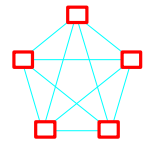
As both NRZI and the scrambler can multiply errors, it is worth-while to use two transmitted ten-bit words of SDT for each control character, and to ensure that there is a reasonable hamming distance between the different codes used.

Suggested coding of control characters is therefore:

NUL	Idle character	33, followed by CC
FCC	Flow Control	55, followed by AA
EOP	End of Packet	66, followed by 99
EEOP	Exceptional EOP	0F, followed by F0

There seem to be several different means of allocating bytes to the ten bits. If the author understands correctly, the 8-bit TV signals are sent on D[9:2], with D[1,0] set to zero. The SDTI proposal includes a nine bit value in D[8:0], with D9 the inverse of D8. Some manufacturers use D8:1 for the data, with D0 set to One to distinguish it from 8-bit TV, and D9 set to zero.

The choice of which of these to use is left open. There would be a small advantage in using D[8:1], because the spatial separation of Bits 0 and 9 is less liable to be corrupted in such a way as to form a valid but incorrect character as a result of error multiplication than the adjacent bits D[9,8] of the SDTI alternative.



Using D8:1 for data, the most and least significant bits would be used as:

8-bit non-TV Data	Zero	7	6	5	4	3	2	1	0	One
8-bit non-TV Control	One	7	6	5	4	3	2	1	0	Zero

(Comment, just in case the thought occurs that a control character can be FF and hence be seen as a possible start of a TRS. We defined the control codes to have four ones and four zeros in each byte, so the value FF is not possible.)

5. Higher level software for networking

As was shown above, it is straightforward to map other protocols onto 1355, and use of TCP/IP would preserve compatibility with much existing software. TCP/IP has been implemented on a 1355 network, initially under Windows NT, and this could be extended for more general use.

For particular applications within the studio, considerable performance benefits may be realised by using the raw protocols of 1355, perhaps between Java programs running on different machines. 1355 offers complete flexibility here, delivering packets transparently with contents that need be understood only by the sender and receiver of those packets.

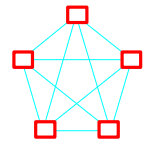
6. Network performance

The crossbar routing switches used with SDT have outstanding performance compared with any of the computer networks. Their job is admittedly much simpler than that of the computer networks. But the switch chip has 16 ports, each capable of 270Mbits/s, to give a total throughput of 4.3 Gbits/s. There are precious few computer network chips with such throughput.

When several of these switch chips are combined to make a larger switch box, the box can have as many as 64 ports, to give a total throughput of 17Gbits/s. Again there are few computer switch boxes with such throughput.

The topology you use in the studio is a star from the switch box, although there can be connections between switch boxes in different studios. On the other hand, some of the networks presented to the Task Force are based on bus and ring topologies, which have rather different characteristics.

It might be extreme, but suppose a 64-port switch was replaced by a single bus. Then the bus would need a bandwidth of 13Gbits/s. And that bandwidth is needed at every node on the bus, although each node only uses 270Mbits/s. And it assumes that each node has immediate access, but that is impossible because all the nodes are competing and they have to wait for each other. These queuing delays mean that the actual utilisation is considerably less than the theoretical peak. So if the queuing delays are reasonable, perhaps a bus bandwidth of



20Gbits/s might be adequate, although it could need far more, even 50Gbits/s, just to achieve the same performance as you achieve at present.

Whether 13Gbits/s, 20Gbits/s or 50Gbits/s, all of these are impossible at reasonable cost, so the bus needs to be broken into sections, with bridges or routing switches between the sections, and perhaps 1Gbit connections at each node. Even with as “low” a speed as 1Gbit/s, the connections are limited in distance to much less than the 300 metres of SDT.

If each node on a 1Gbit section needs 270Mbits/s, the maximum number of nodes on each section is three, and even this may be too many for the queuing delays caused by contention. But assuming three nodes per section is acceptable, a 1Gbit section might be as shown in Figure 15.

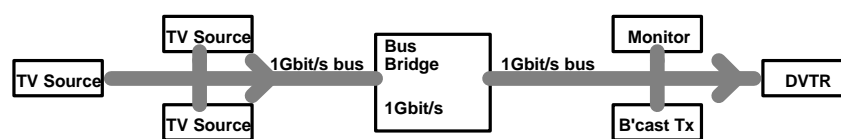


Figure 15: Sections of a 1Gbit bus, connected via a bridge

The problem comes when more nodes are needed and so bridges have to connect to bridges. Figure 16 shows a set of four bus bridges, which appear to allow twice as many nodes. But not really, because the bandwidth between the bus bridges is the same bus speed of 1Gbit/s, so it is still only possible for three sources to transfer to three destinations at the same time, and it is never possible for all six sources to be active at the same time.

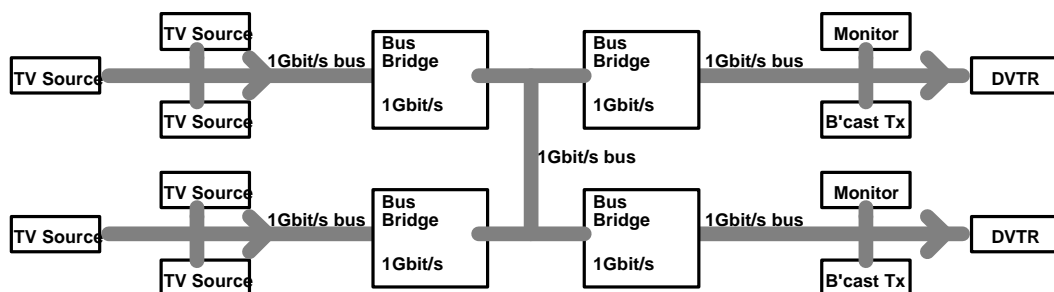


Figure 16: Multiple bus bridges provide no higher system throughput

To provide bandwidth enough for all the sources, either the central “backbone” bus can be increased in bandwidth, say to 10Gbits/s, or the bridges can be replaced by a switch.

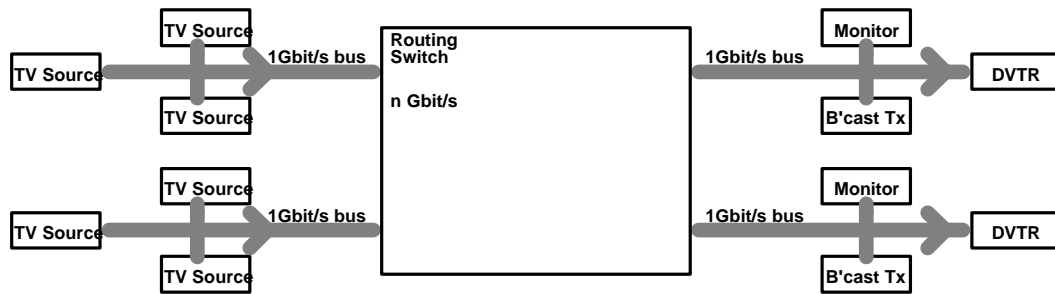
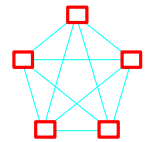


Figure 17: Routing switch, if available, can provide system throughput

The problem with the 10Gbit/s backbone is that none exists. The problem with the Routing Switch is that very few exist for 1Gbit/s ports. And it can be difficult to build routing switches for a bus protocol that was not designed for routing.

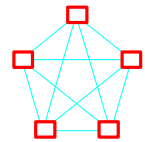
One Gbit/s switch that does exist is the RCube for IEEE 1355. It has eight full-duplex ports, an internal throughput of 5Gbits/s, and a latency from input to output of 180ns if the output port is available. Given reasonably simple protocols on the Gbit bus, the RCube could be used as the switch, with interfacing logic between each bus and the RCube.

For the mainly unidirectional traffic in the TV studio, the RCube might be regarded as a switch with four inputs and four outputs. But as each port can carry at least two multiplexed channels, the effect of the switch for the studio is more like 8 inputs and eight outputs. As with the crossbar switches in the existing equipment, multiple chips can be combined to produce switches with more ports. The multiplexing might be done on a Gbit bus as in the figures, but it might equally be done between SDT channels inside the switch box. Alternatively, as suggested for ESSSI, a new switch is developed for the SDT ports.

The RCube switch is fairly recent and has yet to be fully characterised in networks which use several switches. Extensive characterisation and simulation has been carried out on the lower speed version of IEEE 1355, which runs full duplex at 100Mbits/s. This link speed is slower than the studio needs, but the results of characterisation show achieved performance of 4Gbits/s and 13 Gbits/s, which are about the “peak” throughputs of the SDT crossbars.

The results given here are from a talk given by CERN in April 1997. (Grateful thanks to Stefan Haas and his co-authors. The title of the paper is “The Macramé 1024 Node Switching Network”, and the full presentation is at <http://www.cern.ch/HIS/dshs>)

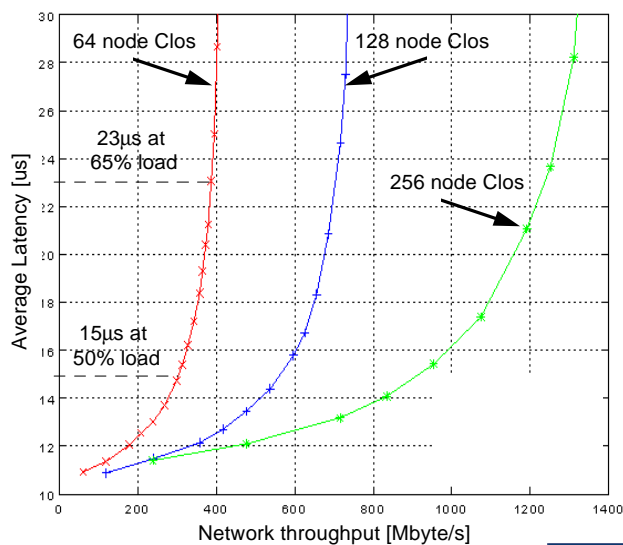
The slide shows latency and total throughput of composite switches made from multiple switch chips, from 64 ports which saturates at 400MBytes/s (4Gbits/s) to 256 ports which saturates around 1.3GBytes/s (13Gbits/s). The random traffic means that there is contention for the destination nodes, which can not be eliminated. But the overall throughput scales remarkably well.



(The way the 32-port switches are connected together to provide 64 ports, 128 ports, or 256 ports, is a so-called Clos network. This is a way of connecting multiple switch chips that has been found particularly effective in telephone exchanges, and it gives better performance for 1355 than other connection schemes that have been tested or simulated. It can be expensive if made with switch chips that have only a few ports, but is cost effective with the comparatively large number of ports of 1355 switches.)

Latency for Clos Networks

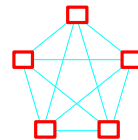
- Average latency versus network throughput
 - » Network load is varied
 - » Throughput and latency are measured
 - » for 64, 128 and 256 node Clos networks
 - » Random traffic
 - » 64 byte packets
- Latency increases exponentially as the network approaches saturation



Since this slide was prepared, the Clos network has been built to 512 nodes, and a simpler grid network to 1024 nodes. Throughput scales very well, even to these large networks. As with the smaller networks, the Clos gives substantially better performance than other topologies.

With these figures showing that the bandwidth needs of the studio can be fully met, even by 100Mbit/s links, extension of the 1355 protocols and switching technology to 270 or 360 Mbits/s will clearly provide the throughput. As the 100Mbit/s 1355 switch has 32 ports, and the Gbit/s 1355 switch has eight ports, a logical goal for a 270/360Mbit/s 1355 switch is 16 ports, which is comparable with the SDT crossbar switch chips.

Several of the computer networks have been designed for throughput rather than latency, and can have quite long latencies even on an idle network. Some computer networks do achieve latencies of a few microseconds when the network is idle. When the network becomes busy, the latencies on any network can become excessive. It would be interesting to hear of figures such as are shown in the CERN slide for the other networks under consideration by the EBU/SMPTE task force.



Apart from the average latency increasing sharply as the network approaches saturation, the worst-case latency increases even more. For example in the 64-node network, at 70% utilisation of every node, there is a 0.5% probability that the latency will exceed 100 microseconds. Again it would be interesting to see comparable figures for other networks. These long delays are accepted in computer networks but might be unacceptable for real-time TV signals

Much has been learned from the CERN results and others, and it is probable that a new switch design might improve on the CERN figures. But there will still be excessive latency at some load, and there are two ways to eliminate these long delays altogether:

- Retaining the existing crossbar-switch routing of live signals.
- Keeping completely within the 1355 protocols, if the packet is not terminated by the EOP character, the path stays open exactly as in the existing studio crossbar switches.

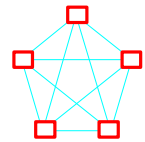
The advantage of the first is that the existing studio equipment will work completely transparently with a new switch. The advantage of the second is that no additional mechanism is needed to the 1355 hardware. Both advantages are worth having, and both guarantee the real-time response.

So building a network with SDT physical layer and with 1355 packet switching thus offers guaranteed response and guaranteed scaleable performance.

7. Conclusions

Building a network with SDT physical layer and with 1355 packet switching offers guaranteed response and guaranteed scaleable performance. With a switch chip designed to implement the 1355 protocols with the SDT physical layer it should be possible to make this chip with 16 full-duplex ports, comparable with the existing SDT crossbar switches. Building these chips to operate in parallel with existing crossbar switch chips, or including the crossbar function on the chips, creates the evolutionary switched serial studio interface proposed in this ESSSI submission. For the TV studios, it retains the existing infrastructure, and the existing equipment, while allowing new equipment to use the packet-switching capability. It allows larger networks, and so simplifies communication between different studios.

But the guaranteed response and guaranteed scaleable performance could be of great benefit to the computer industry, to industrial real-time control, and to data acquisition as well.



8. Appendix A: Plug and Play

While the hardware of 1355 is fully capable of live plug and unplug, there are as yet no defined protocols for software plug and play with 1355.

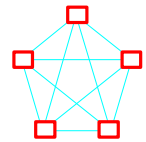
There is as wide a variety of P&P schemes as there are networking standards to choose from. Most of them are based on defined memory maps with the occasional possibility of a device driver for a particular processor instruction set and operating system.

Much the most flexible P&P scheme for plug-in boards is OpenBoot from SUN, which is a FORTH program that can be interpreted by any processor or operating system.

There is no doubt that if SUN were to invent OpenBoot now, they would use Java instead of FORTH, and gain the benefits of security and of a language defined for distributed systems. There are further advantages in using the JavaPP class libraries which make concurrent Java much easier to program and which overcome the race hazards which cause deadlock and starvation. (The race hazard was pointed out by Ted Lewis, Binary Critic, IEEE Computer, Mar. 1997, pp. 136, 133-135. JavaPP was described in a letter to IEEE Computer, published in July 1997, and further information on JavaPP can be found at <http://www.cs.bris.ac.uk/~alan/javapp.html>)

It is not suggested that an OpenBoot based on JavaPP be necessarily developed specifically for the TV studio network, but that in time there will be sufficient dissatisfaction with the existing closed and inflexible P&P mechanisms that a move in the direction of OpenBoot/JavaPP is inevitable.

On the other hand the synergy between the TV industry and the animations possible with Java may offer the right environment for developing a flexible distributed plug and play mechanism based on JavaPP.



9. Appendix B: Analogy with modern manufacturing

The job of moving data round a network, with the objectives of maximum throughput, minimum delay, and minimum cost, is very similar to the job of moving components and subassemblies round a factory. Manufacturing has been through a revolution over the last 50 years, and modern manufacturing uses concepts and techniques that would have been thought crazy 50 years ago. Even with recent advances, computer networks are still more like the old production lines. So if 1355 seems crazy now, this analogy might help to explain it.

Old production line	Modern manufacturing	IEEE 1355
Bottlenecks	Spare or flexible capacity to eliminate bottlenecks	Multiple paths and adaptive routing to eliminate bottlenecks, provide fault tolerance
Rigid production lines	Flexible manufacturing	Flexible encapsulation of other protocols, flexible network topologies
Quality doesn't matter/repair if faulty	Quality is paramount, Zero defects	Reliable links, buffers never overflow
Don't change it	Continuous improvement	Adding nodes and links continuously improves the network throughput
Lead time is irrelevant	Just-In-Time because delay reduces quality, reduces customer service, and reduces throughput	Worm-hole routing and flow-control minimise delay
Long conveyor belt	Many small, independent, cells	Many independent links
Large batch sizes	Small batches, in trays, preferably one unit per tray	Small packets, appropriate for the application
"Push" components onto the line	When a tray is used, the tray goes back as an order to "pull" more components	Feedback flow-control "pulls" data when there is space for it
Long change-over time	Very fast tool-change	Fast, widely distributed, independent, arbitrations; fast changeover between different packets and protocols
Environment does not matter	Waste of any kind is damaging to the environment (as well as profit)	Eliminate wasted data from buffer overflow, eliminate wasted power driving unnecessary wires, eliminate unnecessary RF pollution
Inventory is asset	Inventory is liability	Minimise data buffers
Make 50% profit margin once per year	Make 20% profit margin 10 times per year	1/5th the link speed in a switched network can offer 20 times the overall throughput of a bus or ring