# Deterministic Service Guarantees in 802.12 Networks, Part II: The Cascaded Network Case

Peter Kim
Network Technology Department
HP Laboratories Bristol
HPL-97-148
December, 1997

quality of service,
LAN,
deterministic
service

In part I [1] of this paper, we studied the problem of allocating resources in shared single hub 802.12 networks. We described the packet scheduling model and defined the admission control conditions which enable the network to provide deterministic service guarantees. In this paper, we analyse cascaded (multi-hub) 802. 12 topologies. We derive the relevant network parameters and show that the admission control conditions defined in part I also apply to cascaded topologies when used with these parameters. Experimental results were achieved with a UNIX kernel based implementation in cascaded test networks. These confirm the theoretical results received for network parameters, throughput and end-to-end delay.

# Deterministic Service Guarantees in 802.12 Networks, Part II: the Cascaded Network Case

*Peter Kim*

pk@hplb.hpl.hp.com

HP Technical Report HPL-97-148, April 1997.

Hewlett-Packard Laboratories, Filton Rd, Bristol, U.K.

## Abstract

*In part I [1] of this paper, we studied the problem of allocating resources in shared single hub 802.12 networks. We described the packet scheduling model and defined the admission control conditions which enable the network to provide deterministic service guarantees. In this paper, we analyse cascaded (multi-hub) 802.12 topologies. We derive the relevant network parameters and show that the admission control conditions defined in part I also apply to cascaded topologies when used with these parameters. Experimental results were achieved with a UNIX kernel based implementation in cascaded test networks. These confirm the theoretical results received for network parameters, throughput and end-to-end delay.*

## 1 Introduction

To support end-to-end service guarantees through the Internet, mechanisms which enable these guarantees must also be introduced in switched/bridged LANs. There is however no standard mechanism for providing service guarantees across existing LANs such as 802.3 Ethernet, 802.5 Token Ring, or 802.12 Demand Priority. This is because the medium access mechanisms of these technologies differ. Another factor to be considered is the bridged LAN topology which can include shared, half-duplex- or full-duplex switched links. The packet scheduling and the admission control conditions which are required for supporting advanced services will thus typically be technology specific, sometimes even topology dependent, and must be defined separately for each LAN technology.

The IETF Integrated Services over Specific Link Layers (ISSLL) working group was chartered with the purpose of exploring the mechanisms required for various link layer technologies. References [2] and [3] describe the framework and the priority mapping required for supporting Integrated Services e.g. the Controlled Load [4] or the Guaranteed [5] service across shared and switched IEEE 802 LAN technologies. Our work fits into this framework.

This paper investigates the allocation of resources in cascaded 802.12 networks. It focuses on the support of deterministic service guarantees as required for a Guaranteed service. IEEE 802.12 [6] is a standard for a shared 100 Mbit/s LAN. Data packets are transmitted using either 802.3 or 802.5 frame formats. The MAC protocol is called Demand Priority. Its main characteristics in respect to Quality of Service (QoS) are the support of two service priority levels (normal- and high priority) and the service order: data packets from all network nodes are served using a simple round robin algorithm.

In part I of this paper, we analysed the single hub (repeater) 802.12 network topology. The medium access is centralized and controlled by the hub. In contrast, cascaded networks consist of a number of hubs which are connected in a rooted tree like network structure. Each hub in the shared network can have many links which either connect to a lower level hub or to an end-node. Cascaded topologies may thus potentially incorporate hundreds of network nodes and may have a physical extension of many hundred meters.

The remainder of this paper is organized as follows. Section 2 briefly reviews the scheduling model and the admission control conditions derived in part I. In section 3, we summarize the operation of the Demand Priority protocol in cascaded topologies and discuss whether this affects the service properties. The details of the protocol operation are described in Appendix A.1 and A.2 when we analyse the worst case Demand Priority per-packet overhead and the time it takes to pre-empt the normal priority service in cascaded topologies. In section 4, we discuss the worst case network performance and show how this is adapted by the admission control conditions. We further compare the theoretical results achieved for the data throughput and the end-to-end delay with the results measured in our test network. Section 4 further discusses resource utilization issues and describes how network resources are partitioned. In section 5, we briefly evaluate our allocation scheme in respect to costs and performance. Section 6 discusses related work. Our conclusions are presented in section 7.

## 2 Scheduling Model and Admission Control Conditions - a Review of Part I

### 2.1 The Scheduling Model

In 802.12 networks, each node maintains two link level output queues: one for normal- and one for high-priority traffic. In our system we added link level rate regulators to control the access to the high priority queue on a per-flow basis on each network node. Figure 1 shows the structure of the system for a single hub network.

All traffic is classified at the link layer e.g. in the 802.12 device driver. Best-effort data packets are immediately passed to the normal priority output queue. High priority traffic is rate controlled and passed into the high priority queue. The shared medium access in 802.12 networks is deterministic. All output queues are served in round-robin order. This is based on the exchange of link control signals between network nodes and hubs. The 802.12 priority mechanism ensures that, after the normal priority service is pre-empted, all high priority packets have strict priority over packets with normal priority.

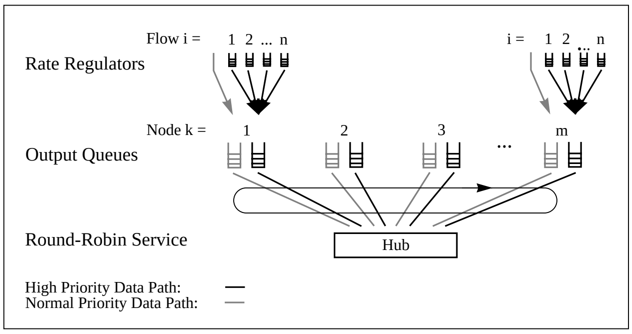


**Figure 1. The Packet Scheduling Model.**

The allocation scheme proposed in part I is based on a time frame concept and is built on top of the 802.12 high priority access mechanism. Admission control is applied to provide deterministic service guarantees. Key design constrains were the variable throughput in 802.12 networks and the fact that hubs are not able to identify and isolate single flows. The variable data throughput is caused by the Demand Priority signalling overhead. This overhead has a significant impact on the performance if small sized data packets are used for the data transmission and, depending on the packet size and the network topology, substantially reduces the data throughput on the network. In part I, we identified two parameters which can be used to describe this overhead: (1) the normal priority service interrupt time $D_{it}$, and (2) the per-packet overhead $D_{pp}$. $D_{it}$ represents the minimum network resources that must be left unallocated at the beginning of each allocation time-frame. The per-packet overhead $D_{pp}$ reflects the available network capacity. It describes the worst case signalling overhead which is required for the transmission of a single data packet across the network. Both parameters, $D_{it}$ and $D_{pp}$ are fixed for a given 802.12 network topology.

The admission control consists of two parts: a bandwidth test and a delay bound test. The bandwidth test defined in Theorem 1 proves that the network has sufficient spare bandwidth to support the new request. The theorem checks that all data from all end-nodes can be transmitted within the time frame. The delay bound test is defined in Theorem 2. It takes advantage of the round-robin service policy, which allows us to calculate a delay bound for each individual node that can be considerably lower than the overall time frame.

In Theorem 1 and Theorem 2, we use the traffic constraint function $b^i(\Delta t)$ defined in part I for a flow $i$, where:

$$b^i(\Delta t) \le \delta^i + r^i \Delta t + r^i T \qquad (2.1.1)$$

$\delta^i$ and $r^i$ are the token bucket depth and the token generation rate of flow $i$ ‘s rate regulator - which is an implementation of a leaky bucket filter [7]. $b^i(\Delta t)$ describes the amount of data which can leave the rate regulator within the time interval $\Delta t$. In both Theorems, $b^i$ is equivalent to $b^i(TF)$, where $TF$ is the length of the time frame.

The parameter $T$ in equation 2.1.1 denotes the timer granularity of the rate regulator. It was considered because the clock used in our implementation is granular (1ms).

For each flow $i$, we further use the packet count $pcnt^i$. It denotes the maximum *number of packets* which are sent by flow $i$ within the time frame $TF$. The packet counts are used to bound the total Demand Priority per-packet overhead within $TF$. In part I, we proposed a simple Time Window measurement algorithm for estimating $pcnt^i$. It is based on the assumption that the application’s packetization process does not change over time. For the applications tested e.g. *vic*, *vat*, *nv*, *MMC* [8], [9] and the *OptiVision MPEG Communication System* [25], we found that the algorithm was able to find an accurate upper bound without impairing the deterministic service guarantees.

In the following, we further use the term *real-time flow* for flows using the 802.12 high priority access mechanism.

## 2.2 The Bandwidth Test

**Theorem 1** *Consider an 802.12 network with $m$ nodes, where each node $k$ has $n$ real-time flows, which are already admitted. Assume a time frame of $TF$, a link speed of $C_l$ and that the packet count for flow $i$ on node $k$ is $pcnt_k^i$. Further let $P_{min}$ be the minimum link packet size and $D_{pp}$, $D_{it}$ be the topology specific worst-case per-packet overhead and normal priority service interrupt time, respectively. Assume also, that all flows are rate regulated and that the input traffic passed into the output queue obeys the traffic constraint function $b_k^i$ for all $i$ ‘s and all intervals $TF$. Sufficient bandwidth for the new flow $\nu$ with $b^\nu$, is available if*

$$b^\nu \le \frac{TF - D_{it} - \frac{1}{C_l}\sum_{k=1}^{m}\sum_{i=1}^{n} b_k^i - \sum_{k=1}^{m}\sum_{i=1}^{n} pcnt_k^i \cdot D_{pp}}{\frac{1}{C_l} + \frac{D_{pp}}{P_{min}}} \qquad (2.2.1)$$

The proof can be found in part I. The rather complicated structure of Theorem 1 is caused by considering the Demand Priority per-packet overhead $D_{pp}$.

## 2.3 The Delay Bound Test

**Theorem 2** *Consider an 802.12 network with* $m$ *nodes, where each node* $k$ *has* $n$ *real-time flows. Assume a link speed of* $C_l$ *and that the packet count for flow* $i$ *on node* $k$ *is* $pcnt_k^i$. *Further let* $P_{max}$ *be the maximum link packet size and* $D_{pp}$ , $D_{it}$ *be the topology specific worst-case per-packet overhead and normal priority service interrupt time, respectively. If Theorem 1 applies, and if all flows are rate regulated and the input traffic passed into the output queues obeys the traffic constraint function* $b_k^i$ *for all* $i$ *'s and all intervals* $TF$ *, then the queuing delay* $d_k$ *for node* $k$ *is bounded by:*

$$\sum_{j=1, j\neq k}^{m}\left(MIN\left(\sum_{i=1}^{n} pcnt_k^i, \sum_{i=1}^{n}\frac{b_j^i}{P_{max}}\right)\cdot\frac{P_{max}}{C_l} + MIN\left(\sum_{i=1}^{n} pcnt_k^i, \sum_{i=1}^{n} pcnt_j^i\right)\cdot D_{pp}\right) +$$

$$\frac{1}{C_l}\sum_{i=1}^{n} b_k^i + \sum_{i=1}^{n} pcnt_k^i \cdot D_{pp} + D_{it} \le d_k \le TF \qquad (2.3.1)$$

The proof of Theorem 2 can also be found in part I.

# 3 IEEE 802.12 in Cascaded Topologies

The cascading mechanism was introduced in 802.12 to allow enlargements of network size and extension. Figure 2 shows potential topologies. The *Root-*, or *Level-1* hub is located at the top of the topology tree. All hubs directly connected to the Root hub are called *Level-2* hubs. These may themselves have many links to end-nodes or lower level hubs, which are then denoted *Level-3* hubs, and so on for larger hierarchies. All hubs, except the Root hub, have a single link which connects them to the next upper hub in the hierarchy. This link is called the *Up Link* of the hub. Links connecting lower level hubs or end-nodes are called *Down Links*. Each hub may thus have many Down links but has never more than one Up link.

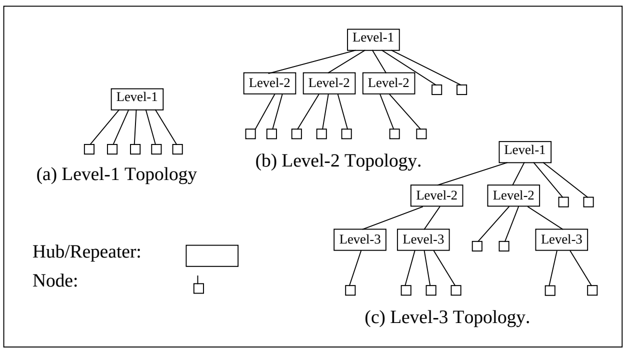


**Figure 2. Cascaded 802.12 Network Topologies.**

The cascading level is used to classify the resulting topologies. A *Level-N Cascaded Topology* consists of at least *N* hubs. It always includes one Level-1- and at least one Level-N hub, but never a *Level-(N+1)* hub. The single hub network discussed in part I of this paper can thus be classified as Level-1 cascaded topology. With a UTP physical layer, cascaded networks with topologies of up to Level-5 are supported by the standard. The maximum cable length between end-nodes and hubs is 200 m in these topologies. Networks with a high cascading level, e.g. Level-4 and Level-5 topologies, are however only required in cases when the physical extension of the network need to be enhanced[1]. Realistic network sizes can already be achieved using Level-2 or Level-3 topologies. A Level-2 topology consisting of 32 x 32 port hubs (1 Root-, and 31 Level-2 hubs) for example could incorporate a maximum of 31 x 31 = 961 end-nodes. The 32nd port of all Level-2 hubs is the Up-link. This should be sufficient to satisfy any requirement for a single shared network.

## 3.1 The Demand Priority Protocol Operation

In the single hub topology, the shared medium access is entirely controlled by the one hub in the network. Nodes wishing to transmit a data packets first signal a service request (or demand) to the hub. Each request is labelled with either normal- or high priority. The hub processes all requests in round-robin order. High priority requests are served first. When a node is selected for transmission, the hub acknowledges the request by sending a Grant signal. This permits the transmission of one packet. After detecting the Grant, the selected node starts sending its packet to the hub which then forwards the packet to its destination. Further details can be found in [10] or [1].

To control the shared medium access in large rooted tree topologies with many hubs, the basic Demand Priority protocol was extended. A mechanism was introduced to allow the distributed operation of the algorithm. As in the single hub topology, there is however always only one hub in control of the network. Using MAC signalling mechanisms, the network control is then passed from hub to hub in the network, such that all nodes are collectively served in a single shared round-robin domain.

The following basic algorithm is carried out: whenever the cascaded network is idle then the network control is at the Root hub. End-nodes wishing to transmit a data packet first signal their service request to the hub to which they are connected to (their local hub), just as described for the single hub case. To serve this request, the local hub must first acquire the network control. If the hub is not the Root hub, then the request is passed on through the Up link to the next upper hub, and so on until it reaches the Root hub.

Following the basic Demand Priority protocol, the Root hub serves all requests in round-robin order. It can distinguish whether a request was received from a directly con-

1 Fiber optic links could also be used when long distances are to be crossed.

nected end-node, or from a lower Level-1 hub. Whenever the service request from a lower Level-1 hub is granted then the Root hub passes the network control down to that hub. Having the network control enables the Level-1 hub to serve one request from all end-nodes connected to it. If required, then the network control is further passed down to a lower Level-2 hub, and so on, so that requests from end-nodes at the leaves of the topology tree can be served.

The network control is returned after a hub has once served a request from all downstream end-nodes and hubs. Note that the control is only passed down on request. It is never given to a lower level hub that does not have a pending service request.

The two priority level are also supported in cascaded topologies. If the Root hub receives a high-priority request while a lower level hub is in the process of servicing low-priority requests, then the Root hub can effectively interrupt the lower level hub in order to serve the high priority request first. This is based on the use of a special MAC control signal. After the network has processed all high priority requests, it resumes the normal priority service at the point in the network, at which it was interrupted. This ensures that fairness is maintained, even in large networks with many hubs.

### 3.2 Service Properties

The distributed operation of the Demand Priority protocol ensures that the service properties, in particular: the packet service order, the priority access mechanism and the fairness, which we observed in single hub networks are also maintained in cascaded topologies. This is most important for our allocation scheme since it enables us to use the same scheduling model and the same admission control conditions, as defined for the single hub network, also for higher cascaded topologies.

Networks with a different cascading level however differ in respect to the network performance. This is considered in the admission control by using cascading level specific values for the Demand Priority per-packet overhead and the normal priority service interrupt time. Both parameters significantly increase in higher cascaded topologies. Theorem 1 and Theorem 2 thus differ by the cascading level specific values to be used for the per-packet overhead and the interrupt time, when applied to different network topologies. Table 5 and Table 6 in Appendix A.1 and A.2 provide the numerical results for these parameters.

## 4 Resource Allocation in Cascaded Topologies

In this section, we compare theoretical and measured results achieved for the network throughput, the allocation limit and the end-to-end delay in cascaded network topologies. The theoretical results were computed using Theorem 1 and Theorem 2. The measured results were collected using the implementation described in [1]. The test network differed in respect to the number of computers and hubs used. We further varied the cable length according to the requirements and the constraints of each test.

### 4.1 The Allocation Limit

We first discuss the worst case network performance and show how this affects the resource allocation limit. For this, we measured the maximum throughput on standard cascaded 802.12 networks versus the packet size used for the data transmission. The same test was already performed for a single hub topology in part I. Here, we however used test networks with a Level-2, Level-3 and a Level-4 topology. This was to: (1) show the impact of the cascading level on the network performance, and (2) to experimental confirm the applicability of Theorem 1 for cascaded networks when combined with the results from Appendix A.1.

The performance of the Level-2 topology was measured first. The test network consisted of one Root hub and three Level-2 hubs as shown in Figure 2 (b). We used three computers which we called Traffic Clients to generate multicast traffic with a packet size ranging from 512 bits (64 bytes) to 12000 bits (1500 bytes). Another computer which we called the Controller was used to: (1) control the packet sizes used by the Traffic Clients, and (2) to measure the throughput. All computers were HP 9000/700 workstations which used the HP-UX 9.05 operating system and standard EISA 802.12 interface cards.

Each Traffic Client was connected to one of the three Level-2 hubs. This caused the maximum signalling overhead for this topology because the network control had to be passed on to another Level-2 hub for each data packet. All three Level-2 hubs in the test network were linked to the Root hub. The Controller measured the data throughput by periodically reading the MIB counters [11] from the managed Root hub. This used SNMP *Get-Request* messages [12]. The incremental step of the packet size was 4 bytes, the measurement time interval was 30 seconds. The links between the Traffic Clients and the hubs and between the hubs themselves consisted of 100 m Category 3 UTP cable. The Controller was directly connected to the Root hub via a 5 m cable of the same type.

We then repeated the experiment in a Level-3 and a Level-4 cascaded network. This used the same setup and the same UTP cabling. The Level-3 topology consisted of one Root hub, three Level-2- and three Level-3 hubs. Each Traffic Client was connected to a Level-3 hub, which itself had an Up link to one of the Level-2 hubs. All Level-2 hubs were connected to the Root hub creating a symmetric topology tree with the Root hub as the only branch point. The Level-4 cascaded topology differed from the Level-3 topology by three additional Level-4 hubs, which were inserted between the Traffic Clients and the Level-3 hubs. The measurement results for all three topologies are shown in

Figure 3. For comparison, we also added the result for the single hub network.

The results in Figure 3 show that the network throughput significantly decreases in higher cascaded topologies. This is caused by the extensive signalling which is required to control the shared medium access in high cascaded topologies. One can observe a maximum performance difference of over 30 Mbit/s in the graphs for the Level-1 and the Level-4 topology. These are the costs for having (1) a larger network size and (2) a wider physical network extension. The results in Figure 3 also suggests that cascaded networks should be built in rich, flat topologies.

The throughput further substantially decreases in all topologies when only small sized packets become used for the data transmission. This dependency was already observed in the single hub network and discussed in detail in part I of this paper. The maximum throughput measured in a Level-4 topology for e.g. data packets of 100 bytes is as low as 18 Mbit/s.

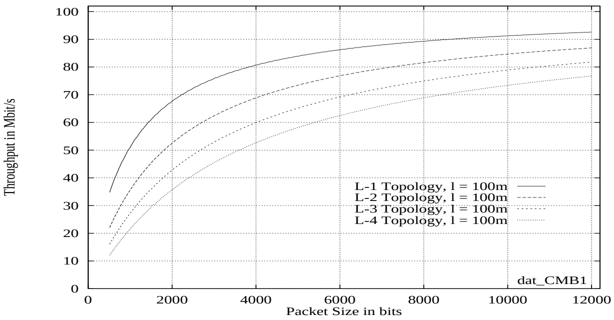


**Figure 3. Measured Worst Case Throughput in Cascaded 802.12 Networks using a UTP Physical Layer.**

Note that all results in Figure 3 were achieved in a setup that only included Traffic Clients located at the leaves of the topology tree. This maximized the signalling overhead. Data packets were further transmitted using multicast in compliance with the worst case transmission model discussed in Appendix A.1. In realistic networks however, unicast and multicast are used. Servers and bridges are directly connected to the Root hub. This reduces the overhead. Hubs can further serve requests from several end-nodes before passing on the network control, which further decreases the signalling requirements. In real networks, we will therefore on average observe a much higher network performance than shown e.g. in Figure 3 for the Level-4 topology.

Figure 4 shows the comparison between the measured throughput, the theoretical worst case throughput and the maximum allocation limit for a Level-2 cascaded network. The upper curve in Figure 4 is the measured throughput as shown in Figure 3 for this topology. The second curve is the computed worst case throughput. It was computed assuming: (1) there is only one active flow, (2) a time frame of $TF = 20\ \mathrm{ms}$, (3) a Level-2 cascaded topology with 100 m UTP cabling represented in a per-packet overhead of $D_{pp} = 21.45\mu\mathrm{s}$, and (4) a normal priority service interrupt time of $D_{it} = 0$. The third curve in Figure 4 is the allocation limit up to which resources can be allocated. It differs from the theoretical throughput such that the computation additionally considered the interrupt time for the Level-2 topology, where $D_{it} = 554.11\mu\mathrm{s}$. As in the single hub case, the computation of both graphs assumed a non-bursty flow and a timer granularity of $T = 0$ to show the accuracy of the admission control.

Two observations can be made in Figure 4: (1) the measured throughput is always higher than the computed worst case result, and (2) the computed and the measured results match closely. This was also found for the single hub network. The results in Figure 4 however do not match as accurately as the results received for the single hub network. The difference between the measured and the computed data throughput is caused by the worst case character of the per-packet overhead $D_{pp}$. This overhead is computed by adding up the worst case delay of all network elements along the data path. In reality however, simultaneous worst case conditions at all layers of the network stack e.g. at the MAC, PMI, PMD, and PHY are rarely met, so that data packets on average are forwarded faster than described by the worst case transmission model. For the 100 m Category 3 UTP cables used in our tests for example, we measured a propagation delay of about 480 ns using an oscilloscope. The standard however allows a maximum delay of 570 ns.

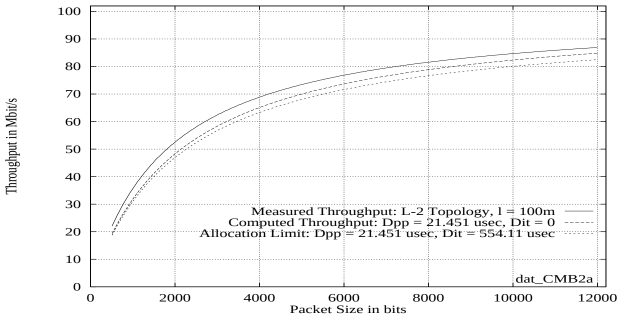


**Figure 4. Comparison: Measured Throughput and Computed Allocation Limit in a Level-2 Cascaded 802.12 Network using 100 m UTP Cabling.**

For the single hub network, we still receive most accurate results because the data path between any two nodes only included two UTP links and one repeating hub. Higher cascaded topologies however have a longer maximum data path. Our Level-2 test network for example connected any two end-nodes via a chain that included 4 links and 3 repeating hubs. The differences in the delay between the model and the reality add up along the longer data path and thus decrease the accuracy between the measured and the computed throughput in higher cascaded topologies.

The difference between the computed throughput and the allocation limit in Figure 4 has also become larger when compared with the results received for the single hub network. This is caused by the normal priority service interrupt time, which also increases in higher cascaded topologies. The difference further depends on the length of the time frame. Smaller time frames reduce the maximum delay bound, but also decrease the maximum allocation limit since the interrupt time must be left unallocated in each time frame. This however seems to be acceptable since a network manager will rarely allow a resource allocation up to the computed throughput. Instead, managers will restrict the use of the high priority access mechanism in order to guarantee that normal-priority traffic does not starve. This is discussed in section 3.4.

Figure 5 contains the results for the Level-3 cascaded network. This is to show how the gaps between the three curves further increases in the Level-3 topology. The measured throughput is identical to the graph shown in Figure 3 for this topology. The theoretical throughput and the allocation limit were computed using a per-packet overhead of $D_{pp} = 32.79\mu s$ and a normal priority service interrupt time of $D_{it} = 878.07\mu s$. The computation was made under the same basic assumptions as considered for the Level-1 and Level-2 topology.

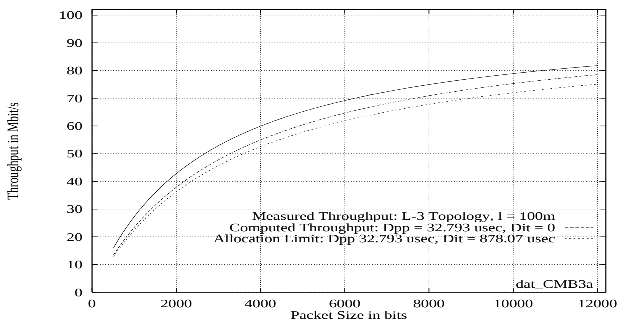


**Figure 5. Comparison: Measured Throughput and Computed Allocation Limit in a Level-3 Cascaded 802.12 Network using 100 m UTP Cabling.**

The results for the Level-4 topology further confirm the behaviour observed for the Level-2- and the Level-3 topology. They are however omitted in this paper.

The important result of this section is that Theorem 1 provides sufficiently accurate results for the minimum data throughput in all cascaded topologies such that the actually available network capacity is always slightly higher than the computed result. This confirms our network model and the results received for the Demand Priority per-packet overhead in Appendix A.1. The results further justify the need for an accurate analysis of the per-packet overhead. Less accurate bounds for the delay in basic network elements like repeating hubs or connecting links will have a large impact on the theoretical results for high cascaded topologies since these topologies have many hubs and links in the data path. Accuracy however ensures that the allocation system has enough resources to manage, such that a sufficient number of high priority flows can be admitted while also guaranteeing a certain resource share for the total best effort traffic.

## 4.2 Delay Issues

In this section we first discuss experimental results received for the interrupt time in cascaded 802.12 networks. These were measured in order to confirm the theoretical analysis performed in Appendix A.2. For several test applications, we then compare the worst case delay bound, which is provided by the admission control, with the maximum delay measured for these applications in our test network.

We however refer to part I for a discussion of the basic 802.12 delay characteristics caused by the priority access and the round robin service policy. Since the Demand Priority service properties are retained in cascaded networks, the general results received for the single hub network are also valid in cascaded networks, even though the exact numerical results would be different in higher cascaded topologies.

### The Measurement Setup

To measure the end-to-end delay in cascaded networks, we used the same measurement methodology as for the single hub case. We only changed the network topology and the cable length. Figure 6 shows the setup for the Level-2 test network.

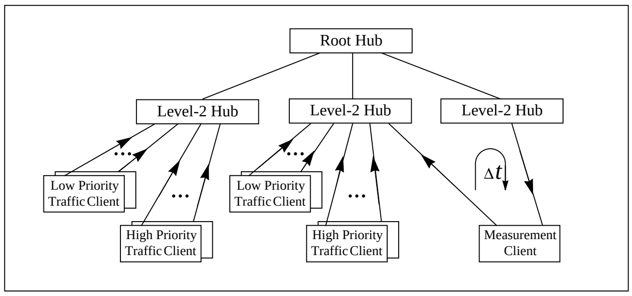


**Figure 6. Setup for Measuring Delay in a Level-2 Cascaded 802.12 Network.**

All measurements were taken by a computer which we called the Measurement Client. It had two 802.12 LAN adapter cards, each of them was connected via a separate UTP cable to the hub. One interface was exclusively used for sending data, the second one was used for receiving. All data packets generated by the Measurement Client were addressed to a pre-defined multicast group which was joined with the receive interface. By using the same computer for sending and receiving test packets, we could use the same clock for determining the start and finish time of each measurement. This avoided timing discrepancies that would have occurred if we had used two separate computers. The time was measured using PA-RISC register CR16 [13], which provides a 10 ns tick on a 100 MHz HP 700 workstation. This ensured a high accuracy of the time-stamps.

The measured delay $\Delta t$ is the link layer end-to-end delay. It includes: (1) the time for transferring the packet from kernel memory to the sending LAN adapter card, (2) the queueing and network transmission time, (3) the time for transferring the packet from the receiving LAN adapter card back into the kernel memory, and (4) the time caused by the interrupt processing and the context switch. We refer to part I for a discussion of the operating system overhead caused by the DMA-, the interrupt process and the context switch.

We further used Traffic Clients to impose high- and normal (low) priority cross traffic. We called these High- or Low-Priority Traffic Clients according to the priority of the traffic they generated. These are the same computers as used for the throughput tests described in the previous section. All packets generated had a length of 1500 bytes to show the worst case effect.

All High- and Low Priority Traffic Clients were connected to Level-2 hubs as shown in Figure 6. The setup for higher level cascaded topologies differed such that for each new level *N*, three additional *Level-N* hubs were inserted between all Clients and the *Level-N-1* hubs. The Measurement- and the Traffic Clients were thus always only connected to Level-N hubs. This enforced a maximum signalling overhead and a maximum data path for all test packets generated by the Measurement Client.

### The Interrupt Time in Cascaded 802.12 Networks

To allow a comparison of the results, the experiments for determining the interrupt time were equivalent to the test that was carried out for the single hub network. The high priority traffic was generated by the Measurement Client. It sent packets at a low mean rate - about 0.56 Mbit/s - corresponding to constant rate compressed video. We further used 10 Low Priority Traffic Clients which imposed normal priority multicast traffic at a total load ranging from 0 to 100 Mbit/s. The measurement interval for each sample was 1 minute which corresponds to about 3000 packets transmitted by the Measurement Client. The incremental step of the normal priority network load was 500kbit/s. In contrast to the setup in Figure 6, we did not use High Priority Traffic Clients in this experiment.

We measured the interrupt time in test networks with a Level-2, Level-3 and a Level-4 topology. The Measurement Client and the hubs were interconnected using 100 m Category 3 UTP cabling. All Traffic Clients however were linked to the hubs via 5 m cable of the same type, since we did not have a sufficient large number of 100 m cables available. This introduced a small difference between the measurement setup and the model in Appendix A.2. The difference is however not significant since the overhead plus propagation delay across a 5 m versus a 100 m UTP cable only differs by 0.541 $\mu s$.

Figure 7 shows the maximum- and the minimum end-to-end delays ($\Delta t$) observed by the Measurement Client. We only labelled the maximum delay curves and, for comparison, added the results achieved in the single hub network (the Level-1 cascaded topology). All measured delays are bounded. For each topology, the time difference between the corresponding maximum- and minimum delay is the time it takes to interrupt the normal priority data transmission within that topology ($D_{it}$).

The minimum delay in a single hub network is about 300 $\mu s$. This slightly increases in higher cascaded topologies due to the data transmission and signalling across a longer data path, which e.g. included 7 repeating hubs and 8 links in the Level-4 cascaded test network. We measured a minimum delay of about 335 $\mu s$ for the Level-4 topology. The maximum delay observed in the single hub network is about 570 $\mu s$. It is caused by the transmission of two normal priority data packets which become served before the normal priority service can be interrupted. For each higher cascading level, two additional normal priority data packets are served. This is analysed in detail in Appendix A.2.

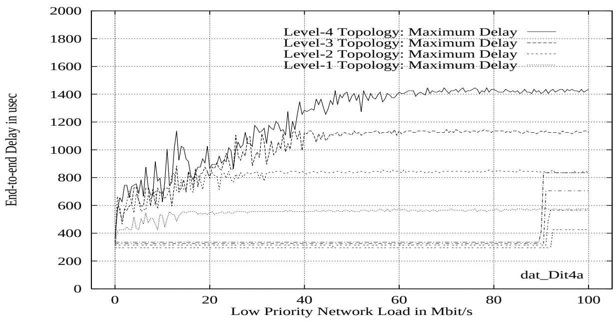


**Figure 7. The Normal Priority Service Interrupt Time in Cascaded 802.12 Networks.**

The maximum delay in Figure 7 thus increases with each cascading level by about 240 $\mu s$ plus the packet transmission overhead for two data packets, where 240 $\mu s$ is the time it takes to transmit two maximum sized packets. In our experiments, we measured a maximum delay of 855 $\mu s$, 1135 $\mu s$ and 1445 $\mu s$ for the Level-2, Level-3 and Level-4 topology, respectively. The resulting normal priority service interrupt times ($D_{it\_L2}$, $D_{it\_L3}$, $D_{it\_L4}$) are: 540 $\mu s$, 810 $\mu s$ and 1110 $\mu s$, respectively. These results confirm the theoretical results shown in Table 6 in Appendix A.2.

### The Quality of Service provided by the Network

In the second part of this section, we report results for the maximum-, the average-, and the minimum link-level end-to-end-delay, which we measured for several test applications in a Level-2 cascaded network with high- and normal priority cross traffic. The experiments had two goals: (1) to confirm that the delay bounds assigned by our allocation system are valid. This implies that all real-time data packets encountered a smaller delay than predicted by the admission control (Theorem 2). (2) To compare the measured maximum- and average delay with the computed worst case bound.

| Test | Appli-cation | Encoding scheme | Delay bound requested. | Per-flow link layer resources allocated | | | |
|---|---|---|---|---|---|---|---|
| | | | | Data rate | Burst size | Max. r-ctrl-er q-length | pcnt considered (TF = 10ms) |
| 1 | vat | PCM2 audio | 10 ms | 75 kbit/s | 1500 bytes | 3 pkts | 2 |
| 2 | vat | PCM2 audio | 10 ms | 75 kbit/s | 1500 bytes | 3 pkts | 2 |
| 3 | vat | PCM2 audio | 10 ms | 75 kbit/s | 1500 bytes | 3 pkts | 2 |
| 4 | vic | JPEG video | 10 ms | 1 Mbit/s | 1500 bytes | 16 pkts | 5 |
| 5 | vic | JPEG video | 10 ms | 1 Mbit/s | 1500 bytes | 16 pkts | 5 |
| 6 | vic | JPEG video | 10 ms | 1 Mbit/s | 1500 bytes | 16 pkts | 5 |
| 7 | OVision | MPEG-1 video | 10 ms | 1.8 Mbit/s | 1500 bytes | 137 pkts | 7 |
| 8 | OVision | MPEG-1 video | 10 ms | 1.8 Mbit/s | 1500 bytes | 137 pkts | 7 |
| 9 | OVision | MPEG-1 video | 10 ms | 1.8 Mbit/s | 1500 bytes | 137 pkts | 7 |
| 10 | MMC | JPEG video | 10 ms | 3 Mbit/s | 1500 bytes | 62 pkts | 8 |
| 11 | MMC | JPEG video | 10 ms | 3 Mbit/s | 1500 bytes | 62 pkts | 8 |
| 12 | MMC | JPEG video | 10 ms | 3 Mbit/s | 1500 bytes | 62 pkts | 8 |

**Table 1. Source and Token Bucket Parameters for the Delay Tests in the Level-2 Cascaded 802.12 Network.**

In the experiments, we used the multimedia test applications *vat*, *vic*, *OptiVision*[1] and *MMC*[2] [8], [25], [9] in the following configurations:

1. *vat* version v3.2: *vat* generated an audio data stream of about 75 kbit/s. The test used the default application setup for PCM2 audio encoding. The data source was the built-in audio device of the HP 9000/725 workstation. 75 kbit/s were allocated at the link layer.

2. *vic* version v2.7b2: *vic* generated a motion jpeg compressed video data stream with a rate of about 1 Mbit/s. Hardware support was given by a parallax card [14]. The data source was a video camera. We used the following *vic* specific parameters that can be adjusted by the user: normal size (resolution 368 x 276 pixel), ordered, jpeg, 22 frames/s. At the link layer, we allocated 1 Mbit/s for application data.

3. *OptiVision* version 1.2f: the OptiVision system generated an MPEG-1 encoded video stream with an average rate of about 1.2 Mbit/s. The video source was a video player playing the adventure movie *Jurassic Park*. The picture resolution was 704 x 480 pixel. 25 frames per second were generated. We allocated 1.8 Mbit/s at the link layer for each flow.

4. *MMC* version v4.0: *MMC* generated a motion jpeg compressed video data stream of about 3 Mbit/s. This was based on the same parallax card as used for *vic*. The size of the video was 720 x 540 pixel. About 11 frames per second were generated. We allocated 3 Mbit/s.

Table 1 summarizes the source- and token bucket parameters. Resources were allocated at the link layer by using the LLRMP signalling protocol [15], [16]. For the sake of simplicity, a delay bound of 10 ms was requested for all applications in all experiments. Columns 7 and 8 in Table 1 show the maximum length of the rate controller queue at the source node and the packet count (*pcnt*) considered for the flow in the admission control. Furthermore, all data packets were sent using IP multicast.

Not all of our computers however did have the audio or video hardware support required for our test applications. To overcome this, we recorded a 2 hour test trace for each of the four applications. This was performed using a traffic monitor which, for each data packet, stored an entry of the format: <*packet_arrival_time, packet_length*> in a trace file. In the delay tests, the traces were then passed to a UNIX kernel based traffic generator which generated an almost identical data stream to the original trace monitored on the network. The time stamps of the traffic monitor have an accuracy of 1 $\mu s$. The traffic generator used a rate controller of 1 ms.

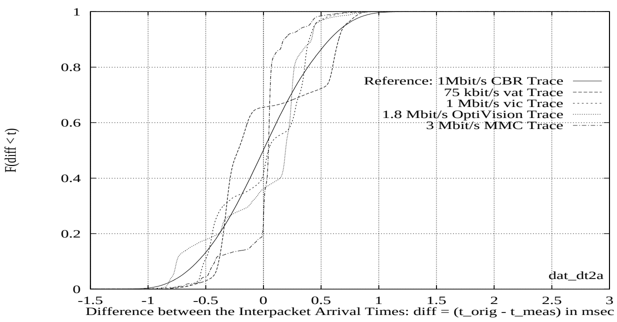


**Figure 8. Difference of the Interpacket Arrival Times between original and measured Data Traces.**

[1] OptiVision is an MPEG Communication system supporting MPEG-1 and MPEG-2 video transmissions. It can be used for conferencing or Video-on-Demand.

[2] MMC is a high quality conferencing system supporting voice, video and application sharing.

For all four traces, Figure 8 shows the *interpacket arrival time differences* between the original trace, which was sent into the network by the traffic generator, and the trace which was then measured with the traffic monitor. For reference, we also added the measurement result received for a 1 Mbit/s constant bit rate (CBR) trace which used random packet sizes between 64 bytes and 1500 bytes. In Figure 8, it can be observed that the differences between the interpacket arrival times are small. In our tests, they are mainly determined by the 1 ms timer granularity of the traffic generator. For the MPEG encoded Jurassic Park trace for example, 99 percent of all packet interarrival times differed by less than 0.85 ms.

During the delay tests, each data source then selected a uniformly distributed start-offset into the trace between 0 and 7200 seconds. On reaching the end of the trace, the sources wrapped around to the beginning. This method is similar to the one used in simulations in [26], to simulate data from several sources using one variable bit rate video trace.

The use of real traffic traces enabled us to generate audio and video data flows on test computers without specific hardware support. It further ensured realistic flow characteristics, in particular a realistic packet size distribution of all high priority data flows in the network. This was important for testing the Time Window measurement algorithm proposed in part I. Results achieved with this algorithm were used for the admission control in all experiments.

The test network was a Level-2 cascaded network with one Measurement Client and several High- and Low Priority Traffic Clients, as shown in Figure 6. It however included two additional Level-2 hubs. This created a topology with five Level-2 hubs and one Root hub. The wiring was the same as used in the previous experiments: the Low- and High Priority Traffic Clients were connected via 5 m Category 3 UTP cables. All other links in the network consisted of 100 m cable of the same type. In all experiments, we used 2 Low Priority Traffic Clients. They generated a total cross traffic load of 89 Mbit/s using a simple traffic generator. To enforce maximum normal priority service interrupt times, all Low Priority Clients generated data packets of maximum length.

The test applications generating *vat*, *vic*, *OptiVision* and *MMC* type traffic run on the Measurement Client and on the High Priority Traffic Clients. In each experiment, we admitted homogeneous applications e.g. only *vic* flows or only *MMC* flows until we reached the allocation limit. All High Priority Traffic Clients and the Measurement Client further always had an identical setup in respect to the type and the number of applications running. This simplified the measurement process since we did not have to measure the delay at High Priority Clients. Measurements were only taken for data packets generated on the Measurement Client. We can however assume that the basic results achieved for the Measurement Client e.g. the average delay are also valid for each High Priority Client since on average, they passed a similar traffic pattern into the shared network.

All measurement were carried out on a per-flow basis by measuring the end-to-end delay in the network for each data packet that was generated by the selected flow. Any delay introduced by the rate-controller at the source node was not considered because our investigations were focused on the actual network behaviour. The measurement interval was 30 minutes for each individual experiment.

Table 2 shows the measurement results. The first three columns of the table contain the test number, which corresponds to the number in Table 1, the application type and the total number of flows admitted in the test. The fourth column shows the deterministic delay bound provided by our allocation system for each flow after all flows had been admitted.

Topology information is given in columns 5 and 6. For each application type, we carried out three experiments, in which we varied the number of High Priority Cross Traffic Clients and the number of local flows.

| Test | Application | Number of flows admitted. | Delay bound provided | Topology information | | Measured parameters | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Number of nodes with reservations | Number of flows per node | High priority data rate | Minimum delay | Average delay | 90 % | 99 % | Maximum delay | Average packet Size |
| 1 | 75 kbit/s vat | 55 | 9.98 ms | 11 | 5 | 4.07 Mbit/s | 0.155 ms | 0.477 ms | 0.545 ms | 0.595 ms | 0.755 ms | 368 bytes |
| 2 | 75 kbit/s vat | 55 | 9.98 ms | 5 | 11 | 4.07 Mbit/s | 0.095 ms | 0.468 ms | 0.545 ms | 0.595 ms | 0.695 ms | 368 bytes |
| 3 | 75 kbit/s vat | 55 | 9.98 ms | 1 | 55 | 4.07 Mbit/s | 0.155 ms | 0.484 ms | 0.535 ms | 0.575 ms | 0.805 ms | 368 bytes |
| 4 | 1 Mbit/s vic | 26 | 9.34 ms | 13 | 2 | 23.89 Mbit/s | 0.105 ms | 0.611 ms | 0.715 ms | 0.915 ms | 1.685 ms | 934 bytes |
| 5 | 1 Mbit/s vic | 26 | 9.34 ms | 8 | 3, Mclient: 5 | 23.77 Mbit/s | 0.095 ms | 0.628 ms | 0.755 ms | 0.975 ms | 1.625 ms | 934 bytes |
| 6 | 1 Mbit/s vic | 26 | 9.34 ms | 2 | 13 | 23.91 Mbit/s | 0.105 ms | 0.628 ms | 0.735 ms | 0.955 ms | 1.725 ms | 934 bytes |
| 7 | 1.8 Mbit/s OVision | 18 | 8.98 ms | 9 | 2 | 21.49 Mbit/s | 0.235 ms | 0.734 ms | 0.845 ms | 1.045 ms | 1.965 ms | 1332 bytes |
| 8 | 1.8 Mbit/s OVision | 18 | 8.98 ms | 6 | 3 | 22.94 Mbit/s | 0.235 ms | 0.745 ms | 0.875 ms | 1.085 ms | 2.065 ms | 1332 bytes |
| 9 | 1.8 Mbit/s OVision | 18 | 8.98 ms | 1 | 18 | 22.77 Mbit/s | 0.235 ms | 0.757 ms | 0.885 ms | 1.385 ms | 2.225 ms | 1331 bytes |
| 10 | 3 Mbit/s MMC | 13 | 8.65 ms | 13 | 1 | 38.90 Mbit/s | 0.145 ms | 0.746 ms | 0.875 ms | 1.105 ms | 2.055 ms | 1356 bytes |
| 11 | 3 Mbit/s MMC | 13 | 8.65 ms | 6 | 2, Mclient: 3 | 38.90 Mbit/s | 0.135 ms | 0.752 ms | 0.875 ms | 1.255 ms | 2.445 ms | 1356 bytes |
| 12 | 3 Mbit/s MMC | 13 | 8.65 ms | 2 | 6, Mclient: 7 | 38.86 Mbit/s | 0.115 ms | 0.771 ms | 0.955 ms | 1.615 ms | 2.545 ms | 1356 bytes |

**Table 2. Comparison: Computed and Measured Network End-to-End Delay in a Level-2 Cascaded 802.12 Network.**

In Test 10 for example, we admitted a single 3 Mbit/s *MMC* flow on 13 computers (12 High Priority Clients plus one Measurement Client). In test 11, the network contained five High Priority Clients and one Measurement Client. Each High Priority Client injected two 3 Mbit/s *MMC* flows into the network, the Measurement Client generated three 3 Mbit/s *MMC* flows in this experiment. In test 12, six MMC flows were admitted at a single High Priority Client, seven at the Measurement Client, resulting in just two nodes with reservations in the network. The total number of flows admitted in each test was determined by the allocation limit, and implicitly, by the delay bound requested. A 14th *MMC* flow could thus not have been admitted.

The difference between the requested delay bound (10 ms) and the provided upper bound shown in Table 2 is mainly caused by the use of the Time Window algorithm and its initial pessimistic assumption that a new flow only uses minimum sized packets for the data transmission. This requires more free resources at call admission due to the additional per-packet overhead that must be considered. A 14th *MMC* flow is thus rejected even though sufficient resources for supporting the flow are actually available in the network. This is because the admission control does not yet know that *MMC* does not only use minimum sized data packets. In high loaded networks, applications requesting a higher data rate will thus have a lower probability to become accepted.

Column six in Table 2 shows the high priority data rate measured in all experiments over the measurement interval of 30 min. The results for *vat*, *vic* and *MMC* are close to their allocation limit. The total data rates observed in the *OptiVision*-tests are significantly lower since resources were over-allocated to avoid long maximum queuing delays in the source’s rate controller.

The next 5 columns (7 - 11) contain the main results of our experiments. They show the minimum-, average-, 90%-tile, 99%-tile and the maximum end-to-end delay measured for a single *vat*, *vic*, *OptiVision* or *MMC* flow. For the experiments 7, 8 and 9 (OptiVision tests), the delay density and the corresponding distribution functions are shown in Figure 9, Figure 10, Figure 11 and Figure 12.

We first observe that all results for the average- and for the maximum delay are significantly lower than the worst case upper bound computed with Theorem 2. This was expected since: (1) simultaneous worst case conditions in the network and at all Clients are rare, and (2) several High Priority Clients were connected to the same Level-2 hub in our test network. The latter reduced the average Demand Priority signalling overhead because all Level-2 hubs could sometimes subsequently serve data packets from several High Priority Clients. Since the available data rate in a Level-1 network may differ by more than 10 Mbit/s, some transmission requests were thus served much faster than assumed in the worst case for the Level-2 topology. This increased the total throughput and thus reduced the delay.

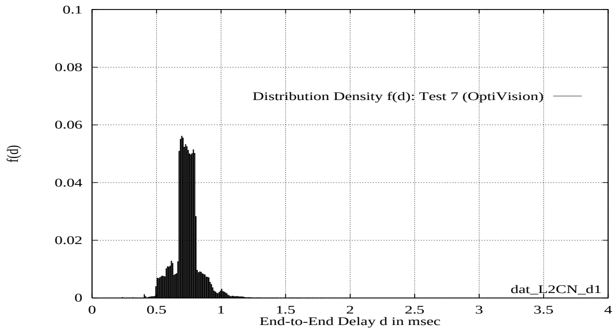


**Figure 9. The Delay Distribution (Density) for the Results of Test 7 in Table 2.**

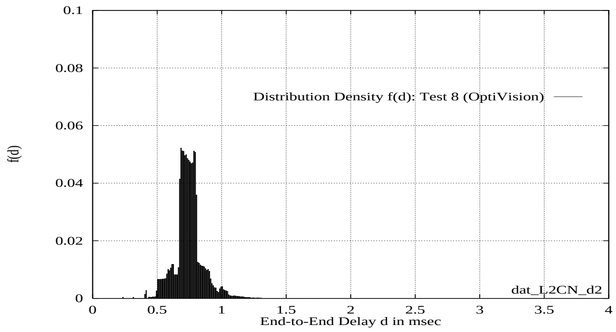


**Figure 10. The Delay Distribution (Density) for the Results of Test 8 in Table 2.**

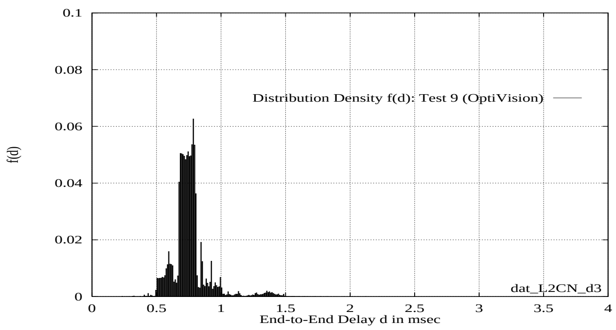


**Figure 11. The Delay Distribution (Density) for the Results of Test 9 in Table 2.**

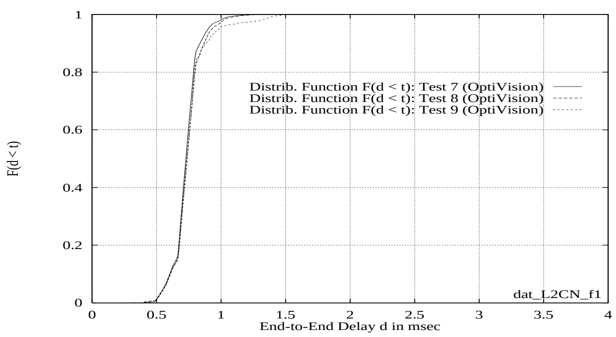


**Figure 12. The Distribution Function for the Results of Test 7, Test 8 and Test 9 in Table 2.**

We can further observe that varying the topology while keeping the total high priority load constant did not have any significant impact on the average delay. We assume that this is due to (1) the rather low high priority network load, and (2) the fairness of the round-robin packet service policy which enforces a sufficient sharing of network resources between different nodes.

Given the low high priority network load, the results for the average delay, especially in the *vat* and *OptiVision* tests, might, at a first glance, seem rather high when e.g. compared with results for the same load on a full-duplex 100 Mbit/s link. Figure 13 thus shows how the average delay received in Test 7 is composed. The figure contains the delay distribution function for four different experiments: (1) the Measurements Client as in Test 7 (with two 1.8 Mbit/s *OptiVision* high priority flows) but no other traffic on the LAN, (2) the Measurements Client as in Test 7 and unicast low priority cross traffic, (3) the Measurements Client as in Test 7 and multicast low priority cross traffic, and (4) the setup of Test 7: one Measurement Client, eight High Priority Cross Traffic Clients and multicast low priority cross traffic.

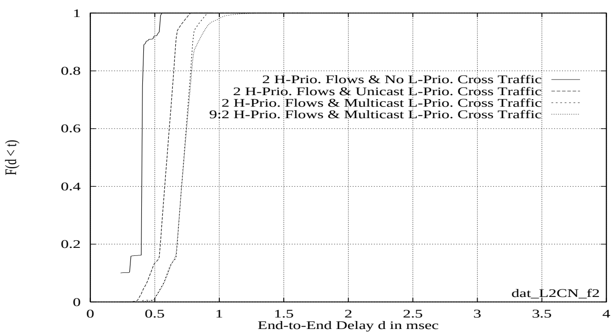


**Figure 13. The Impact of the Low Priority Service Interrupt Time on the Result for Test 7 (OptiVision) in Table 2.**

One can observe that the average delay for the no cross traffic case (1) in Figure 13 is low. We measured 396 $\mu s$. The average delay increases by about 190 $\mu s$ in experiment (2) when unicast cross traffic is added. As in all previous experiments, low priority cross traffic is generated at a rate close to the network capacity using fixed size packets of 1500 byte. The Measurement Client generates high priority traffic at a low data rate. For almost every high priority packet sent, the low priority service needs to be interrupted. While the corresponding link control signal (see Appendix A.2 for details) is travelling from the Measurement Client to the hub in control, several unicast packets are served by the network before the high priority request can be granted. The average delay further increases significantly when all low priority cross traffic is sent using multicast as shown by the results for experiment (3) in Figure 13. The interrupt time is however bounded as observed in Figure 7. Finally, the high priority cross traffic added in case (4) does not have any significant impact on the average delay of the flow measured by the Measurement Client. It only marginally changes the distribution as can be observed in Figure 13.

The results in Table 2 have shown that the network is capable of providing very small packet transmission delays. We believe that these are sufficient for supporting existing time critical applications. The use of the priority access combined with admission control guarantees that these delays remain very low when there is a high normal priority network load, or when the shared network incorporates many more hubs and nodes.

## 4.3 Maximum Resource Utilization

Table 3 shows the maximum number of *vat*, *nv*, *vic*, *OptiVision* and *MMC* flows which the allocation system could simultaneously admit in a Level-2 cascaded network. This is: (1) to compare these results with the results received for the same test in the single hub network, and (2) to show the impact of the reduced allocation limit on the total number of flows that can be supported in a Level-2 topology.

To allow an accurate comparison, the results in Table 3 were achieved with the same test applications and under the same assumptions as made for the single hub network. The application setup for *vat*, *vic*, *OptiVision* and *MMC* were the same as described in section 4.2 of this paper. *nv* generated a video data stream of about 128 kbit/s. Hardware support was provided by an HP A.B9.01.3A frame grabber card. The test used *nv* version v3.3beta and the default setup with a medium picture size (resolution 320 x 240 pixel with a frame rate of 1 - 3 frames/s). We allocated 128 kbit/s at the link layer.

All results are based on the use of the Time Window measurement algorithm proposed in part I. Following the worst-case model, each flow was first admitted assuming the use of only minimum sized packets, where $P_{min} = 64 byte$. For all existing flows $i$, the admission control used the packet counts ($pcnt^i$) measured by the Time Window algorithm. Note that flow arrival and lifetime statistics were not considered in this test since we focused on determining the highest utilization in a pre-defined setup.

Table 3 shows the maximum number of flows ($N_{max}$) that could be admitted for three different time frames: 10 ms, 20 ms and 40 ms. The admission control used the Level-2 topology specific parameters for the per-packet overhead ($D_{pp} = 21.45 \mu s$) and the interrupt time ($D_{it} = 554.11 \mu s$).

For the sake of simplicity, the queuing delay bound requested for all flows was always equal to the time frame. The timer granularity $T$ used in equation 2.1.1 during the admission control was 1 ms. The rate regulators allowed an initial burst of $\delta^i = 12000 \text{bits}$, which corresponds to one maximum size data packet. We further always admitted homogeneous flows.

Each row in Table 3 provides the result for one application in a given setup: e.g. for a time frame and delay bound of 20 ms a maximum of 40 1 Mbit/s *vic* flows could be admitted in the Level-2 network.

| Time frame TF | Delay Bound | Application | Data rate allocated per flow. | Max. number of flows admitted ($N_{max}$) | $pcnt$ measured | Bandwidth allocated (Mbit/s) | Maximum high priority network utilization (%) |
|---|---|---|---|---|---|---|---|
| 10 ms | 10 ms | vat | 75 kbit/s | 55 | 2 | 4.12 | 5.15 |
| | 10 ms | nv | 128 kbit/s | 47 | 3 | 6.02 | 7.51 |
| | 10 ms | vic | 1 Mbit/s | 26 | 5 | 26.00 | 32.45 |
| | 10 ms | OptiVision | 1.8 Mbit/s | 18 | 7 | 32.40 | 40.43 |
| | 10 ms | MMC | 3 Mbit/s | 13 | 8 | 39.00 | 48.67 |
| 20 ms | 20 ms | vat | 75 kbit/s | 87 | 4 | 6.53 | 7.91 |
| | 20 ms | nv | 128 kbit/s | 83 | 4 | 10.62 | 12.88 |
| | 20 ms | vic | 1 Mbit/s | 40 | 6 | 40.00 | 48.49 |
| | 20 ms | OptiVision | 1.8 Mbit/s | 26 | 9 | 46.80 | 56.74 |
| | 20 ms | MMC | 3 Mbit/s | 17 | 11 | 51.00 | 61.83 |
| 40 ms | 40 ms | vat | 75 kbit/s | 152 | 5 | 11.40 | 13.63 |
| | 40 ms | nv | 128 kbit/s | 130 | 6 | 16.64 | 19.89 |
| | 40 ms | vic | 1 Mbit/s | 50 | 10 | 50.00 | 59.77 |
| | 40 ms | OptiVision | 1.8 Mbit/s | 30 | 16 | 54.00 | 64.55 |
| | 40 ms | MMC | 3 Mbit/s | 20 | 17 | 60.00 | 71.72 |

**Table 3. High Priority Network Utilization in a Level-2 Cascaded 802.12 Network.**

The last column shows the maximum high priority network utilization which is computed by relating the allocated bandwidth to the maximum allocation limit. The maximum allocation limit is the maximum capacity that can be allocated when all data is sent with maximum sized packets. It is fixed for each topology and can thus be used as reference value for computing the network utilization. For our test topology and setup, the maximum allocation limit is 82.89 Mbit/s. We thus receive a maximum high priority resource utilization of 48.49% for the admitted 40 1 Mbit/s *vic* flows.

In Table 3, similar observations can be made as discussed for the single hub network: the maximum resource utilization is low when only low bitrate (vat) flows are admitted due to the allocation overhead and the reduced data throughput in the network. The utilization substantially increases for larger delay bounds, and when high bitrate flows using large packet sizes (vic, MMC) become admitted. A comparison with the results achieved for the single hub network shows that, as expected, less flows can be admitted for all applications in the Level-2 topology. The largest difference in the allocated bandwidth can be observed for MMC flows. Even though only 4 flows less became admitted in the Level-2 topology, the allocated bandwidth decreased by 12 Mbit/s. A sufficient number of flows can however still be admitted as can be observed in Table 3.

Similar consideration can be made for the Level-3 cascaded network. In this section, we however only focus on the Level-2 topology because we believe that this is the most widely used cascaded topology.

## 4.4 Resource Partitioning

To ensure that normal priority traffic does not starve, network resources must be partitioned. The availability of resources for normal priority traffic is guaranteed by restricting the access to the high priority service. This is controlled by admission control.

To increase the guaranteed resource share for normal priority traffic, the high priority allocation limit must be decreased. For this, we first define the *High Priority Utilization Factor f*, where $0 \le f \le 1$. $f$ defines the maximum resource share that can be allocated for high priority traffic. A utilization factor of $f = 1$ thus allows the allocation of all network resources available.

Since the allocation scheme is based on a time frame concept, the resource maximum corresponds to the total transmission time that is available within the time frame $TF$. We additionally define the minimum normal priority transmission time $LTT$. It represents the minimum resource share that is guaranteed to be available for normal priority traffic. The minimum for $LTT$ is given by the interrupt time $D_{it\_LN}$, where $N$ is the cascading level of the network topology. The resources represented by $D_{it\_LN}$ can not be allocated since they are required for pre-empting the normal priority service. The maximum for $LTT$ is the time frame itself. We thus have the relation: $D_{it\_LN} \le LTT \le TF$. If we consider the high priority utilization factor $f$, then we receive for the minimum normal priority transmission time:

$$LTT = MAX(D_{it\_LN}\ ;\ TF \cdot (1 - f)) \qquad (4.4.1)$$

where $D_{it\_LN} \le LTT \le TF$ is achieved for utilization factors of $0 \le f \le 1$. If we now replace the interrupt time $D_{it\_LN}$ in Theorem 1 with the minimum normal priority transmission time $LTT$ then we have:

$$b^{\nu} \le \frac{TF - LTT - \frac{1}{C_l}\sum_{k=1}^{m}\sum_{i=1}^{n} b_k^i - \sum_{k=1}^{m}\sum_{i=1}^{n} pcnt_k^i \cdot D_{pp}}{\frac{1}{C_l} + \frac{D_{pp}}{P_{min}}} \qquad (4.4.2)$$

To enable the network administrator to control the high priority allocation limit, the equations 4.4.2 and 4.4.1 are used for admission control. The allocation limit is changed by adjusting the utilization factor $f$. An example is given in Figure 14. It shows the allocation limit in a Level-2 cascaded network for $f = 0.6$ $(60\%)$.

The average network capacity that is available for normal priority traffic will however be higher than $1 - f$ because: (1) the allocation limit was determined based on worst case assumptions (the worst case computed throughput), and (2) any resources unused by high priority flows are immediately available for normal priority traffic.

Theorem 2 does not need to be updated to support resource partitioning since for all utilization factors, the normal priority data transmission is still pre-empted after $D_{it\_LN}$ time units. For low utilization factors, the delay bounds given by Theorem 2 are always significantly smaller than the time frame $TF$. This is due to the smaller total amount of resources allocated.

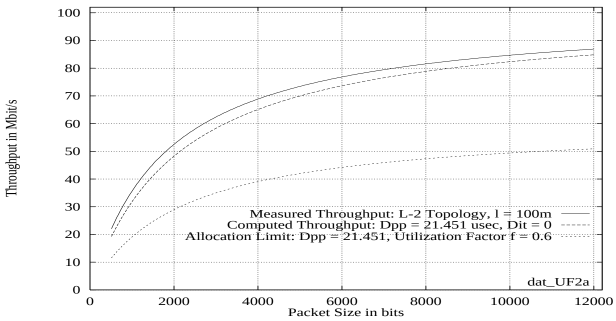


**Figure 14. The Allocation Limit in a Level-2 Network for a High Priority Utilization Factor of: f = 0.6.**

The partitioning mechanism described in this section provides a simple method for network administrators to set a basic policy required in Integrated Services networks: the minimum bandwidth available for normal- and high priority traffic. We believe that without any such control, an advanced service based on a static priority queueing system can not be deployed because of the starvation problem. This section however showed that such control can easily be integrated in our allocation system.

# 5 Costs and Performance Issues

The main advantages of the allocation scheme proposed in this paper and its first part [1] are its simplicity and its low costs. Hubs do not have to support per-flow classification or per-flow buffering and have only buffer space for a single maximum size data packet. The shared network may have a large size and extension, but deterministic service guarantees can still be provided. The solution further does not require any changes to the 802.12 standard and can be implemented in software. When deployed, then only network nodes which use the high priority access mechanism need to be updated.

The simplicity of the hubs and of the scheduling policy however result in a low resource utilization, especially for low bitrate flows. We believe that this is acceptable since any unused resources are not wasted, but can immediately be used by the network for serving normal priority service requests. A statistical multiplexing gain between real-time flows from different network nodes can not be exploited since all high-priority traffic is rate controlled at end-nodes and not within hubs. Our experiments however showed that the average delay across the network is sufficiently low for supporting existing time critical applications. The support for just two priority levels in 802.12 further limits the number of service classes that can be supported in the network. Other drawbacks are the general costs for the link level reservation setup mechanism, and for the classifier and the rate regulators in the device driver. These are however not specific to our solution, but will also occur in other reservation schemes with active admission control.

Assuming the current price differences between 100Mbit/s repeaters and bridges, shared 802.12 networks supporting quality of service seem to be a flexible and cost effective network solution for supporting applications with stringent time constraints. Bridges are required when the total network traffic exceeds the capacity of the shared system.

# 6 Related Work

In [18] the Target Transmission Time (TTT) technique was proposed for allocating resources on Demand Priority networks. The algorithm leads to bandwidth and delay guarantees, and supports a fixed delay bound for all real-time flows in the network. This delay bound is the TTT. The paper however only reports preliminary results. Admission control conditions and mechanisms for the reservation setup or the TTT negotiation were not provided. Apart from [18], we are not aware of any other scheme for allocating resources or controlling the high priority access in 802.12 networks.

The support of service guarantees over LANs has however been investigated for other technologies. In [19], [20] and [21], the real-time performance of the timed token protocol as used in FDDI has been studied. [19] and [20] analyse several schemes for allocating synchronous network capacity. [21] investigates performance parameters to maximize the throughput for best effort traffic, while meeting access delay bounds for real-time traffic. [22], [23] report the design and implementation of a software based timed-token protocol that provides performance guarantees on existing Ethernet hardware. The authors of [24] investigated the use of priorities in 802.5 token-ring networks. All these schemes are based on a time frame mechanism. Network capacity is allocated as a certain fraction of the time frame. The minimum delay bound guaranteed for all flows depends on the token rotation time.

Our allocation scheme also uses a time frame. The time frame however is not necessarily the minimum delay bound. The allocation scheme can guarantee much smaller delay bounds of the order of a few milliseconds. This makes the requirement for a mechanism to negotiate the time frame less important than on networks operating according to e.g. the timed-token protocol.

# 7 Conclusions

In this paper we showed that the resource allocation scheme proposed in part I of this paper can also be applied across cascaded 802.12 networks. This is based on the use of topology specific network parameters in the admission control conditions. In one part of this work, we analysed the network performance and derived results for: (1) the per-packet overhead and (2) the normal priority service interrupt time for cascaded topologies.

Experimental results received in standard cascaded test networks confirmed the analytical results for these parameters. We observed in our measurements that the network throughput substantially degrades in higher cascaded topologies. Network properties which we had already observed in the single hub case e.g. the strong dependency between network performance and used packet size, were also found in cascaded topologies. Our experiments further showed that the admission control conditions when used with the topology specific network parameters can accurately model the network performance.

We found that the scheme offers excellent delay characteristics. Very small delay bounds can be guaranteed in potentially large shared networks by using the 802.12 high priority access method with admission control. The results received in the analysis for 802.12 network parameters are further an essential condition for allocating resources in bridged/switched 802.12 networks since they will enable us to compute the available data rate for outgoing links in bridges and switches.

# Acknowledgements

We thank Aled Edwards, John Grinham, Greg Watson, David Cunningham, Michael Spratt, Costas Calamvokis and Chris Dalton for their comments and the fruitful discussions while this work was carried out. We further thank Jon Crowcroft for his comments and for encouraging us to write this paper.

# A Appendices

In the following two appendices, we derive the worst case Demand Priority per-packet overhead and the time it takes to interrupt the normal priority service in cascaded 802.12 networks. We consider topologies using UTP non-bundled cables as physical links.

We assume the reader to be familiar with Appendix A.3 and A.4 in part I [1]. These describe the details of the signalling across a single link and the delay components introduced in each layer of the 802.12 stack. In this paper, we make use of numerical results derived in these appendices for the maximum time it takes: (1) a data packet ($D_{Tx\_Data}$), (2) a *Grant* control signal ($D_{Signal\_Grant}$), (3) an *Incoming* control signal ($D_{Incom}$), and (4) a *Request* signal ($D_{Req\_H}$), to travel from one MAC (e.g. from a network node), across a single link to another MAC (e.g. a hub). Beside the propagation time across the physical medium, all results include the delay introduced in the sending- and the receiving PMD and PMI. Furthermore, the parameter $D_{MAC\_data}$ denotes the worst case delay which a data packet may encounter in the MAC of the hub. For a discussion of the 802.12 MAC timers, we also refer to part I. The timer values for the *IPG*- and *D_IPG* window, and the *I_BST* offset can be found in the standard [6] (see section 12.5.1).

## A.1 The Worst Case Signalling Overhead in Cascaded 802.12 Networks

In this appendix, we derive the worst case per-packet overhead. The worst case occurs under exactly the same conditions as in the single hub network. These are: (1) when two network nodes are switching between sending and receiving unicast data packets, or (2) when two or more nodes send data packets using multicast or broadcast.

Figure 15 shows a model for the packet transmission and the signalling that is required for transmitting four data packets across a Level-2 cascaded network. The model only shows the signalling details which are relevant for deriving the per-packet overhead in this topology and omits the high- or normal priority service request (*Req_H*, *Req_L*) signalling and the *IPG*, *D_IPG* and *I_BST* timer constraints discussed in detail in part I. The example topology consists of three hubs and two nodes. Each node is connected to a Level-2 hub creating a maximum data path between the two nodes. We further assume that both nodes have at least two data packet to send and request the same service priority.

The data flow in Figure 15 starts when Node 1 sends a data packet. This packet travels along the data path and traverses all three hubs in the network on its way towards Node 2. When the Root hub has finished repeating the packet, it hands the network control over to Hub 3. This is carried out with the Grant signal. Having the network control enables Hub 3 to serve the request from Node 2. For this, Hub 3 carries out the same procedure as a hub in a single hub network: it sends a Grant to Node 2 and, when it receives the data packet, forwards the packet towards the destination e.g. towards Node 1. After forwarding the last bit, Hub 3 passes the network control back to the Root hub by signalling Idle, as shown in Figure 15. The Demand Priority timing constraints ensure that the Root hub receives the network control before it has itself repeated the last bit of the data packet from Node 2.

After the packet processing is finished, the Root hub hands the network control over to Hub 2, so that the next request from Node 1 can be served.

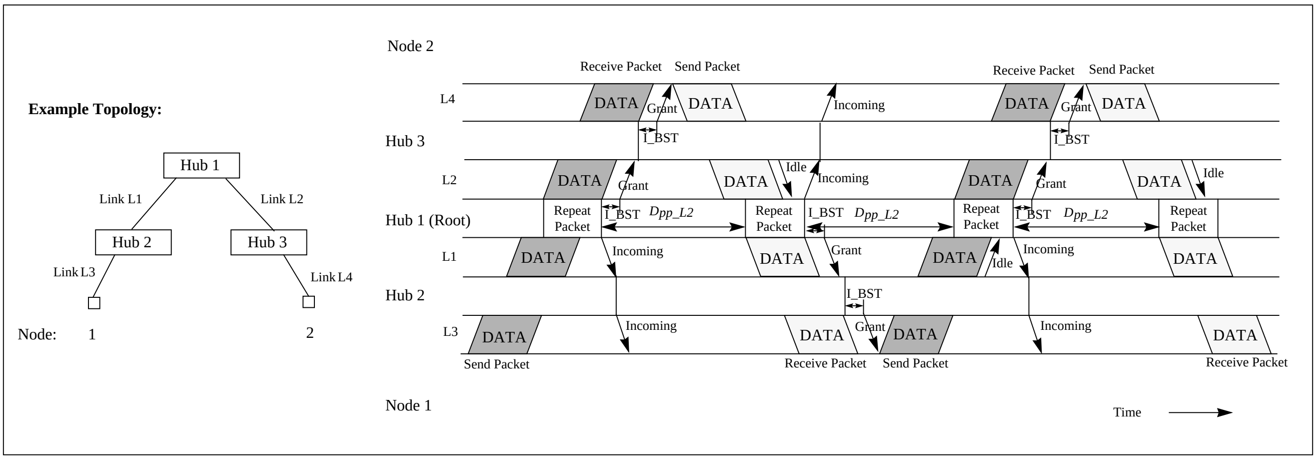


**Figure 15. Worst-Case Signalling and Data Transmission on a Level-2 Cascaded 802.12 Network.**

When this request is processed then the control is again given to Hub 3 and so on. The network control is thus passed between both Level-2 hubs for each service request in the network. This creates a maximum overhead without that the network runs idle.

As already observed for the single hub case, the Grant signalling in Figure 15 is always delayed by a preceding data packet. This increases the per-packet delay since $D_{Tx\_Data} > D_{Signal\_Grant}$ . The delay between the time when the Root hub decides to pass the network control to Hub 3 (by sending Grant) and the time when Node 2 detects the Grant signal is thus as long as: $D_{Tx\_Data} + D_{MAC\_data} + D_{Tx\_Data} + I\_BST$ . This follows from Figure 15 and the considerations made in Appendix A.3 in part I of this paper. When Node 2 starts the packet transmission, it takes a maximum of $D_{Tx\_Data} + D_{MAC\_data} + D_{Tx\_Data} + D_{MAC\_data}$ time units until the MAC of the Root hub passes the first bit of the data packet to the PMI of link L1. If we consider the constrain of the 802.12 standard that the gap between two subsequent data packets is at least as big as the Interpacket Gap: *IPG + D_IPG*, then we receive for the worst case per-packet overhead $D_{pp\_L2}$ in a Level-2 topology:

$$D_{pp\_L2} \le MAX\,((IPG + D\_IPG);\ (D_{Tx\_Data} + D_{MAC\_data} + D_{Tx\_Data} + I\_BST + D_{Tx\_Data} + D_{MAC\_data} + D_{Tx\_Data} + D_{MAC\_data})) \qquad \text{(A.1.1)}$$

The same consideration as for the Level-2 topology can also be made for higher cascaded networks. The results for this are omitted here. If we rearrange Equation A.1.1, then we have:

$$D_{pp\_L2} \le MAX\,((IPG + D\_IPG);\ (D_{Tx\_Data} + I\_BST + D_{Tx\_Data} + D_{MAC\_data} + 2 \cdot (D_{Tx\_Data} + D_{MAC\_data}))) \qquad \text{(A.1.2)}$$

A comparison of equation A.1.2 with the result received for the single hub case shows that both results only differ in the term: $2 \cdot (D_{Tx\_Data} + D_{MAC\_data})$ . This can be generalized since for each higher cascading level, the maximum data path always increases by two hubs and two links, which causes an additionally delay of $2 \cdot (D_{Tx\_Data} + D_{MAC\_data})$ for data packets travelling along this path. The worst case per-packet overhead $D_{pp\_LN}$ in a Level-N cascaded topology is thus given by:

$$D_{pp\_LN} \le MAX\,((IPG + D\_IPG);\ (D_{Tx\_Data} + I\_BST + D_{Tx\_Data} + D_{MAC\_data} + 2 \cdot (N-1)(D_{Tx\_Data} + D_{MAC\_data}))) \qquad \text{(A.1.3)}$$

Using Equation A.1.3 with the numerical results for the signalling delay across a single link, we computed the worst case per-packet overhead for Level-N cascaded topologies, where $1 \le N \le 5$ . The results for 5 m, 100 m and 200 m UTP cabling are shown in Table 5.

| UTP-Cable Length | Cascading Level $N$ | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| 5 m | 9.03 µs | 19.29 µs | 29.55 µs | 39.81 µs | 50.07 µs |
| 100 m | 10.11 µs | 21.45 µs | 32.79 µs | 44.14 µs | 55.48 µs |
| 200 m | 11.25 µs | 23.73 µs | 36.21 µs | 48.70 µs | 61.18 µs |

**Table 5. Worst-Case 802.12 Per-Packet Transmission Overhead $D_{pp\_LN}$ for different Cascading Level $N$ .**

## A.2 The Worst Case Normal Priority Service Interrupt Time in Cascaded Topologies

In this appendix, we derive the worst case time it takes to interrupt the normal priority service in cascaded 802.12 networks. We first describe the packet transmission model and compute the result for the Level-2 network. The results for the Level-1 and Level-2 topology are then generalized for higher cascaded topologies.

Figure 16 shows the time space model for the Level-2 cascaded topology. It illustrates the worst case signalling that is required for pre-empting the normal priority service and for transmitting a single high priority packet. The model only shows the signalling details which are required for deriving the interrupt time and omits the *IPG*, *D_IPG* and *I_BST* timer constraints. The example topology consists of three hubs and three nodes. All nodes are connected to Level-2 hubs at the leaves of the topology tree. We analyse the interrupt time in respect to Node 1 which is requesting the transmission of a high priority packet. The two other nodes in the setup, Node 2 and Node 3, only use the normal priority service. As in the single hub case, we assume that Node 2 and Node 3 send multicast or broadcast data packets while Node 1 is requesting the high priority service. This provides the worst case interrupt time, which is denoted with $D_{it\_L2}$ for the Level-2 cascaded topology.

When we compare the model in Figure 16 with the model shown in part I for the single hub case then we can observe that the interrupt time $D_{it\_L2}$ now includes the transmission times for four normal priority data packets. These are sent by Node 2 and Node 3. The worst case occurs when the high priority request (*Req_H* signal in Figure 16) only travels across a single UTP link before the signalling is delayed by a normal priority data packet. At the same time, the network control toggles between the Root hub and Hub 3.

In the worst case, it is passed to Hub 3 just before the *Req_H* signal from Node 1 reaches the Root hub. The Root hub must then first regain the network control before the high priority request from Node 1 can be granted. For this, the Root hub sends a special link control signal to Hub 3. This signal is called *Enable-High-Only* (*Ena_HO*). It is used for pre-empting the normal priority service. When Hub 3 detects the *Ena_HO* signal, it finishes the processing of the current normal priority packet and returns the network control. The Root hub then passes the control to Hub 2, so that the high priority request from Node 1 can be served.

In higher cascaded topologies, a hub receiving *Ena_HO* from a higher level hub might have to pass the signal on when the network control is currently at a hub that is located further down the topology tree. Further, if a hub is serving high priority requests while it receives *Ena_HO*, it may finish its current high priority service round before it returns the network control to the upper level hub.

The data flow in Figure 16 starts when Node 1 sends a data packet towards Node 2 and Node 3. We assume that at the same time, Node 2 has a service request for a normal priority data packet pending. Both data packets are served by the network in the same way as described for the first two data packets in Figure 15 in Appendix A.1.

The overhead associated with the data packet from Node 2 is the worst case delay $D_{pp\_L2}$ for this topology, as can be observed in Figure 15. We then assume that the MAC of Node 1 in Figure 16 runs idle.

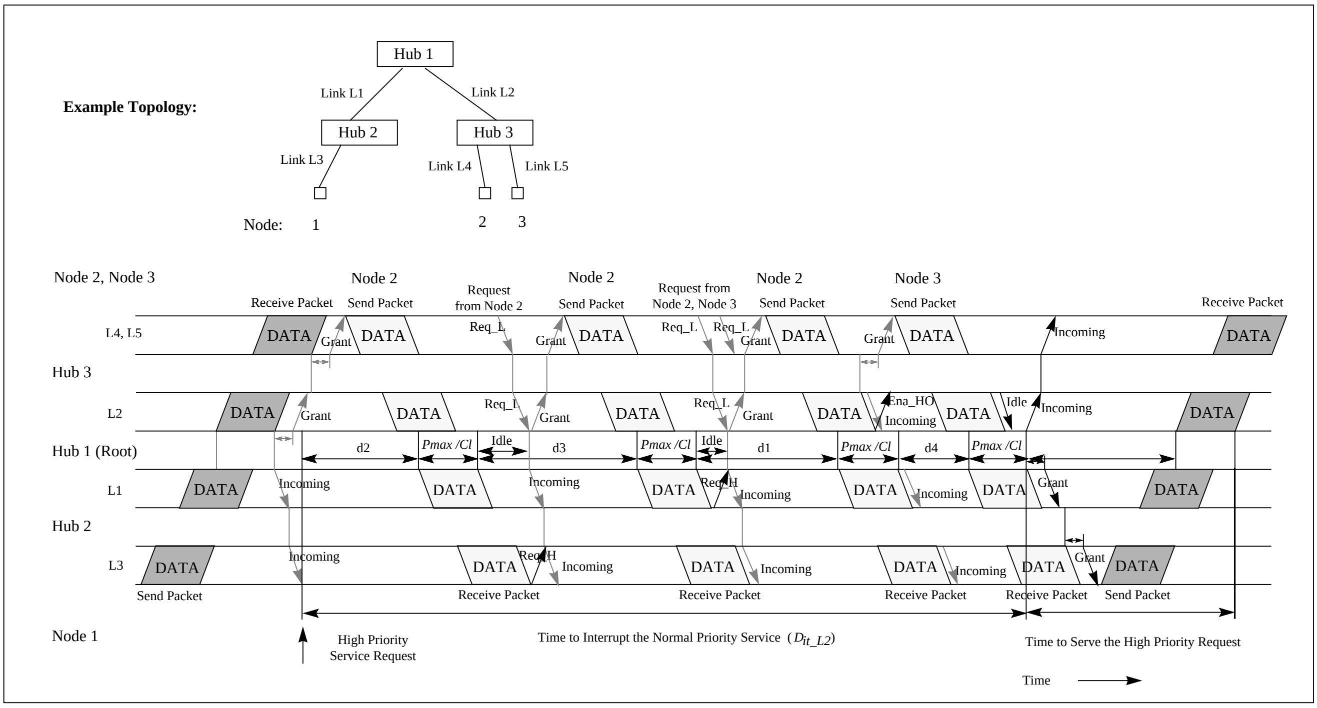


**Figure 16. Model for Computing the Worst-Case Interrupt Time on a Level-2 Cascaded 802.12 Network.**

As in the single hub case, the worst case condition for the interrupt time $D_{it\_L2}$ occurs when a new high priority request is made at Node 1 instantly after the *Incoming* signal was detected. Since the *Incoming* signal must travel across two links before it can arrive at Node 1, we receive for the overhead $d_2$ to be considered in $D_{it\_L2}$ for the first normal priority data packet:

$$d_2 = D_{pp\_L2} - 2 \cdot D_{Incom} \tag{A.2.1}$$

After the Root hub has forwarded the packet from Node 2, it runs idle until it receives the next normal priority service request from Node 2. Note that this request could also be from a different node in the network. The request is instantly granted as shown in Figure 16. For this, the Root hub hands the network control to Hub 3 and, at the same time, sends *Incoming* to Hub 2. The worst case in respect to $D_{it\_L2}$ occurs when the high priority request (*Req_H*) from Node 1 arrives at Hub 2 at the same time as the *Incoming* signal from the Root Hub. In this case, the UTP PMD of Hub 2 does not pass the request on to the Root hub since it must prepare itself for receiving the data packet from Node 2. If the *Incoming* signal had arrived later at Hub 2, then the *Req_H* would have travelled further across link L1 to the Root hub. The overhead to be considered for the second data packet from Node 2 is denoted with $d_3$ in Figure 16. It is larger than $D_{pp\_L2}$ since it also contains the time in which the Root hub runs idle. From Figure 16 we receive for $d_3$ by using the delay components derived in part I:

$$\begin{aligned} d_3 = {} & D_{Tx\_Data} + D_{MAC\_data} + D_{Tx\_Data} + \\ & D_{Req\_H} - D_{Incom} + 2 \cdot D_{Signal\_Grant} + \\ & D_{Tx\_Data} + D_{MAC\_data} + D_{Tx\_Data} + D_{MAC\_data} \end{aligned} \tag{A.2.2}$$

When the Root hub has forwarded the data packet from Node 2, it again runs idle. The idle time is equal to the idle time observed in the single hub case. Node 2 and Node 3 then request the transmission of a normal priority packet by signalling *Req_L* to Hub 3. As in the single hub case, the worst case occurs when the *Req_H* signal from Hub 2 arrives at the Root hub just after the normal priority request from Node 2 has been granted. The *Enable_High_Only* (*Ena_HO*) signal is not signalled across link L2 before the data packet from Node 2 has been fully received at the Root hub. From Figure 16 follows for the per-packet overhead $d_1$ to be considered for the third packet from Node 2:

$$\begin{aligned} d_1 = {} & D_{Tx\_Data} + D_{Req\_H} + 2 \cdot D_{Signal\_Grant} + \\ & D_{Tx\_Data} + D_{MAC\_data} + D_{Tx\_Data} + D_{MAC\_data} \end{aligned} \tag{A.2.3}$$

After Hub 3 has forwarded the data packet from Node 2, it keeps the network control and serves the normal priority request from Node 3. The *Enable_High_Only* signal from the Root hub always arrives at Hub 3 after this decision has been made. The network control is thus not returned until the data packet from Node 3 has been fully repeated. The corresponding per-packet overhead $d_4$ can be as long as the worst case delay in a single hub network: $D_{pp\_L1}$, since Node 3 did also have to receive the preceding multicast data packet from Node 2 (this however is not shown in Figure 16). This follows from our considerations in part I:

$$d_4 = D_{pp\_L1} \tag{A.2.4}$$

The normal priority packet transmission is pre-empted when the Root hub has regained the network control from Hub 3. The network then serves the high priority request from Node 1. The control can however not be handed over to Hub 2 before the Root hub has finished the forwarding of the multicast data packet from Node 3. The signalling and data transmission which is carried out for the high priority request from Node 1 is the same as discussed for the data packets in Figure 15. If we now assume that Node 2 and Node 3 sent data packets of maximum size $P_{max}$ then we receive from Figure 16 for the worst case interrupt time $D_{it\_L2}$ in a Level-2 cascaded network:

$$D_{it\_L2} \le 4 \cdot \frac{P_{max}}{C_l} + d_2 + d_3 + d_1 + d_4 \tag{A.2.5}$$

where $P_{max} / C_l$ is the transmission time for a data packet of maximum size, and $d_2$, $d_3$, $d_1$ and $d_4$ are the results received with the equations A.2.1 to A.2.4, respectively.

**Generalization**

We made the same considerations as in Figure 16 for the Level-3 and the Level-4 cascaded network. If we consider the cascading level in the results received for the Level-1, Level-2 and Level-3 topology, then we get for the interrupt times:

$$D_{it\_L1} \le 2 \cdot \frac{P_{max}}{C_l} + d_1(1) + d_2(1) \tag{A.2.6}$$

$$D_{it\_L2} \le 4 \cdot \frac{P_{max}}{C_l} + d_1(2) + d_2(2) + d_3(2) + d_4(2) \tag{A.2.7}$$

$$D_{it\_L3} \le 6 \cdot \frac{P_{max}}{C_l} + d_1(3) + d_2(3) + d_3(3) + d_4(3) + d_5(3) + d_6(3) \tag{A.2.8}$$

It can be observed that the maximum number of normal priority data packets which are served by the network before the normal priority service is pre-empted is equal to the number of UTP links in the data path. In a Level-5 cascaded topology, as many as ten normal priority data packets can thus be served by the Root hub before the high priority request is granted. The per-packet overheads in equations A.2.6, A.2.7 and A.2.8 are computed using the functions $d_i(N)$, where *N* is the cascading level and $i$ a packet index. The functions provide a generalized way to compute the per-packet overhead in all topologies. $d_2(2)$ and $d_4(2)$ for example provide the overhead of the first and fourth normal priority data packet in $D_{it\_L2}$, and are thus identical with the equations A.2.1 and A.2.4, respectively. If we generalize the equations A.2.6, A.2.7 and A.2.8 then we receive for the Level-N cascaded topology:

$$D_{it\_LN} \leq 2N \cdot \frac{P_{max}}{C_l} + \sum_{i=1}^{2N} d_i(N) \tag{A.2.9}$$

where $1 \leq N \leq 5$. The generalization of the per-packet overheads for packets with an even index $i$ is straightforward. Observing the results for the Level-1, Level-2, Level-3 and Level-4 topologies, we get for the corresponding functions $d_i(N)$ in equation A.2.9:

$$d_2(N) = D_{pp\_LN} - N \cdot D_{Incom} \tag{A.2.10}$$

$$d_4(N) = D_{pp\_L(N-1)} \tag{A.2.11}$$

$$d_6(N) = D_{pp\_L(N-2)} \tag{A.2.12}$$

$$d_8(N) = D_{pp\_L(N-3)} \tag{A.2.13}$$

where $D_{pp\_LN}$ and $D_{pp\_L(N-1)}$ for example are the worst case per-packet overhead in the Level-N and Level-N-1 cascaded topology, respectively. The results for the functions $d_i(N)$ with an odd index $i$ are more complicated since they also describe the idle times which we e.g. observed for the Root hub in Figure 16. We further made two worst case assumptions for all cascading level *N*: first that each Grant signal is delayed by an preceding idle burst (*I_BST*), and secondly that all per-packet overheads are at least as big as $D_{pp\_LN}$.

Both assumes that the receiver of the next Grant is always also a receiver of the last data packet. Since this assumption is however not always true as can be observed in Figure 16, this insignificantly increases the computed upper bound. It however enables a simple generalization of the results for all cascading level. By adding these two assumptions to the results received for the Level-1, Level-2, Level-3 and Level-4 topologies, we got for the functions with an odd index $i$ in equation A.2.9:

$$\begin{aligned} d_1(N) = \; & MAX((D_{pp\_LN});\\ & (D_{Tx\_Data} + D_{Req\_H} +\\ & N \cdot (I\_BST + D_{Signal\_Grant} + D_{Tx\_Data} + D_{MAC\_data}))) \end{aligned} \tag{A.2.14}$$

$$\begin{aligned} d_3(N) = \; & MAX((D_{pp\_LN});\\ & (2 \cdot D_{Tx\_Data} + D_{MAC\_data} + D_{Req\_H} - D_{Incom} +\\ & N \cdot (I\_BST + D_{Signal\_Grant} + D_{Tx\_Data} + D_{MAC\_data}))) \end{aligned} \tag{A.2.15}$$

$$\begin{aligned} d_5(N) = \; & MAX((D_{pp\_LN});\\ & (3 \cdot D_{Tx\_Data} + 2 \cdot D_{MAC\_data} + D_{Req\_H} - 2 \cdot D_{Incom} +\\ & N \cdot (I\_BST + D_{Signal\_Grant} + D_{Tx\_Data} + D_{MAC\_data}))) \end{aligned} \tag{A.2.16}$$

$$\begin{aligned} d_7(N) = \; & MAX((D_{pp\_LN});\\ & (4 \cdot D_{Tx\_Data} + 3 \cdot D_{MAC\_data} + D_{Req\_H} - 3 \cdot D_{Incom} +\\ & N \cdot (I\_BST + D_{Signal\_Grant} + D_{Tx\_Data} + D_{MAC\_data}))) \end{aligned} \tag{A.2.17}$$

The results for the functions $d_9(N)$ and $d_{10}(N)$ are straightforward to derive by observing the results for the lower cascaded topologies. This is however omitted here.

In Figure 16, one can observe that the idle times increase the interpacket gaps between subsequent normal priority data packets. We found that these idle times further increase in higher cascaded topologies. They however do not lead to a significant increase of the worst case interrupt time $D_{it}$. Using the numerical results computed for the Grant-, Incoming- and the Data signalling delay in part I of this paper, the impact is only in the order of a few microseconds. This is because the Grant signal, which is sent after each idle time, can travel about twice as fast as the Incoming- or the Data signal. The worst case per-packet overhead is thus not always achieved with a maximum idle time. Instead, the maximum interpacket gap often occurs when the normal priority request is instantly granted and the Grant signal is delayed by a preceding multicast data packet. In this case the per-packet overhead becomes $D_{pp\_LN}$, as we described for a Level-2 network in Appendix A.1.

Using equation A.2.9, the equations A.2.10 - A.2.17, and the delay components derived in part I of this paper, we computed the worst case interrupt time for all valid cascading level *N*. The results for 5 m, 100 m and 200 m UTP cabling are shown in Table 6. A comparison of these results with the results measured in our test network is provided in section 4.2.

| UTP-Cable Length | Cascading Level $N$ | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| 5 m | 259.22 µs | 545.45 µs | 861.34 µs | 1208.57 µs | 1586.58 µs |
| 100 m | 261.92 µs | 554.11 µs | 878.07 µs | 1236.06 µs | 1628.23 µs |
| 200 m | 264.77 µs | 563.23 µs | 895.74 µs | 1265.70 µs | 1673.11 µs |

**Table 6. Worst Case Normal Priority Service Interrupt Times in Cascaded 802.12 Networks using UTP Cabling.**

# References


[1] P.Kim, *Deterministic Service Guarantees in 802.12 Networks, Part I: the Single Hub Case*, HP Technical Report HPL-97-147, April 1997.

[2] A. Ghanwani, J. W. Pace, V. Srinivasan, *A Framework for Providing Integrated Services Over Shared and Switched LAN Technologies*, Internet Draft draft-ietf-issll-is802-framework-01.txt, April 1997.

[3] M. Seaman, A. Smith, E. Crawley, *Integrated Services over IEEE 802.1D/802.1p Networks*, Internet Draft draft-ietf-issll-802-01.txt, June 1997.

[4] J. Wroclawski, *Specification of the Controlled-Load Network Element Service*, Internet Draft draft-ietf-intserv-ctrl-load-svc-03.txt, August 1996.

[5] S. Shenker, G. Partridge, R. Guerin, *Specification of the Guaranteed Quality of Service*, Internet Draft draft-ietf-intserv-guaranteed-svc-06.txt, August 1996.

[6] *IEEE 802.12, IEEE Standard for Local and Metropolitan Area Networks: Demand-Priority Access Method, Physical Layer and Repeater Specification for 100Mb/s Operation*, IEEE, November 1995.

[7] R. L. Cruz, *A Calculus for Network Delay, Part I: Network Elements in Isolation*, IEEE Transactions on Information Theory, Vol. 37(1), pp. 114 - 131, Jan. 1991.

[8] Mbone Tools, Online Software: http://www-nrg.ee.lbl.gov/.

[9] N. Leymann, *Eine Videokomponente fuer das Videokonferenzsystem Multimedia Collaboration*, Diploma Thesis, in German, Technical University of Berlin, August 1996.

[10] G. Watson, A. Albrecht, J. Grinham, J. Curcio, D. Dove, S. Goody, M. Spratt, P. Thaler, *The Demand Priority MAC Protocol*, IEEE Network Vol. 9, No. 1, pp. 28 - 34, Jan. 1995.

[11] J. Flick, *Definitions of Managed Objects for IEEE 802.12 Repeater Devices*, Internet Draft, June 1995.

[12] J. Case, M. Fedor, M. Schoffstall, C. Davin, *Simple Network Management Protocol (SNMP)*, RFC 1157, May 1990.

[13] Hewlett-Packard, *PA-RISC 1.1 Architecture and Instruction Set*, Reference Manual, Manual Part No: 09740-90039, September 1992.

[14] Parallax Graphics, *PowerVideo700 Board*, hhttp://www.parallax.com/products/hp/xvideo700.html.

[15] P. Kim, *LLRMP: a Signalling Protocol for Reserving Resources in Bridged Networks*, in Proc. of OPENSIG '96, October 1996.

[16] P. Kim, *Link Level Resource Management Protocol (LLRMP), Protocol Specification - Version 1*, Internet Draft draft-kim-llrmp-01.ps, December 1996, (ftp://hplose.hpl.hp.com/pub/pk/draft-kim-llrmp-01.ps).

[17] M. Jacobs, *CASCADE Architecture High Performance LAN Cards - Hardware External Reference Specification*, Version 2.0, Hewlett-Packard, October 1994.

[18] J. Grinham, M. Spratt, *IEEE 801.12 Demand Priority and Multimedia*, in Proc. of 4th Intern. Workshop on Network and Operating Systems Support for Digital Audio and Video, pp. 75 - 86, 1993.

[19] G. Agrawal, B. Chen, W. Zhao, *Guaranteeing Synchronous Message Deadlines with the Timed Token Protocol*, in Proc. of IEEE Conference on Distributed Computing Systems, pp. 468 - 475, Yokohama 1992.

[20] G. Agrawal, B. Chen, W. Zhao, *Local Synchronous Capacity Allocation Schemes for Guaranteeing Message Deadlines with the timed Token Protocol*, in Proc. of INFOCOM '93, pp.186-193, 1993.

[21] K. Shin, Q. Zheng, *Mixed Time-Constrained and Non-Time-Constrained Communications in Local Area Networks*, IEEE Transaction on Communications, Vol. 41, No. 11, Nov. 1993.

[22] C. Venkatramani, T. Chiueh, *Design, Implementation, and Evaluation of a Software-based Real-Time Ethernet Protocol*, in Proc. of SIGCOMM '95, pp. 27 - 37, Aug. 1995.

[23] C. Venkatramani, *The Design, Implementation and Evaluation of RETHER: A Real-Time Ethernet Protocol*, PhD Dissertation, State University New York, January 1997.

[24] C. Bisdikian, B. Patel, F. Schaffa, M. Willebeek-LeMair, *The Use of Priorities on Token-Ring Networks for Multimedia Traffic*, IEEE Network, Vol. 9, No. 6, pp. 28 - 37, Dec. 1995.

[25] OptiVision Inc., *OptiVision Live MPEG Communication System*, User's Guide, Version 1.2 f, September 1996.

[26] M. Garret, W. Willinger, *Analysis, Modelling and Generation of Self-Similar VBR Video Traffic*, in Proc. of ACM SIGCOMM'94, pp. 269 - 279, London, September 1994.