

---

**94462**  
**COMPUTER COMMUNICATIONS**

**LABORATORY 2**  
**COMPARISON BETWEEN CSMA/CD AND**  
**TOKEN RING MAC ALGORITHMS AND**  
**BRIDGED LANs**

**SEPTEMBER 1998**

**©COPYRIGHT BY MATTHIAS FALKNER 1998**

*(All rights reserved: No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author)*

## 1 PURPOSE

The purpose of this lab session is twofold: first of all, to compare the CSMA/CD and Token Ring Medium Access Control (MAC) algorithms. Secondly, to study bridges in LAN environments. MAC algorithms address an important problem, which arises mainly in LANs: how do you share a *common* medium among many workstations? A LAN is characterized by a large number of stations located geographically within a few kilometers of each other, which are connected by the LAN. Because of the large number of workstations, LANs use a *single* channel to connect the workstations together (can you imagine the complexity of connecting each pair of workstations?). So the MAC algorithms determine at what point in time and how long a single station is allowed to transmit its data on the LAN.

Most of the LANs in industry are based on the two MAC algorithms studied in this lab. CSMA/CD was developed as an extension to the Aloha protocol by the Xerox corporation. The token ring algorithm was developed by IBM as an alternative solution to networking. CSMA/CD is based on a *random access* scheme, where each station is allowed to transmit data as soon as it arrives at the station (the arrival of messages is assumed to be random). Token Ring is a *deterministic* algorithm, where the access to the transmission medium is regulated by a token – a circulating permission to transmit data. Of course, these descriptions of the CSMA/CD and Token Ring algorithms are much simplified. We will refine the description in the theoretical part of this lab.

At the end of this lab, you should be able to

- describe the CSMA/CD and Token Ring MAC algorithms
- identify the main differences between CSMA/CD and Token Ring
- justify the necessity for Bridges
- point out difficulties arising in bridged LANs
- build a COMNET III model from a given description.

In this lab we will not be concerned with the related topic of flow control. We will not discuss details such as

- the framing characteristics of CSMA/CD or Token Ring standards
- token management functions
- other MAC algorithms, such as Token Bus or Aloha
- physical aspects of LANs, such as topologies or cabling details
- the IEEE LAN standards (i.e. the different 802.x standard values) in detail.

This lab session builds on the theoretical material on MAC algorithms and LAN bridges. To successfully complete this lab session, you are expected to be familiar with the COMNET III network simulator. To execute this lab, please make sure that COMNET III is installed on your machine. This lab will also focus much more on building network simulation models in COMNET III. You will not be given any models, but instead will be asked to build the models yourself from scratch. Of course, a detailed description of the required models will be given to you.

## 2 THEORETICAL BACKGROUND

In this section, we will briefly review the CSMA/CD and Token Ring MAC algorithms and provide a short description of their main differences. In particular, we will focus on the maximum medium access time and the channel utilization as performance criteria. We will then describe the function of bridges in LANs.

### 2.1 CSMA/CD

CSMA/CD stands for Carrier Sense Multiple Access with Collision Detection. Dissecting this description gives you a good idea on what the main features of this MAC algorithm are. First of all, CSMA/CD is a *multiple access* protocol, meaning that many stations on the LAN *share* a single link. All stations on the LAN have access to the same channel. Secondly, each of the stations on the LAN *senses the carrier* (i.e. the link). CSMA/CD defines a set of rules on how this carrier sensing is done and used. Finally, since it is a multiple access protocol, there is always the risk of two stations transmitting data at the same time, thus causing a collision. CSMA/CD has a mechanism for each station to *detect such collisions* and to remedy such situations by stopping their transmissions.

To describe the basic operation of CSMA/CD, let us take the point of view of a single station on the LAN. This station either has data to transmit or not. In the latter case, the station just sits idle on the LAN and lets the other stations use the channel for their transmissions. If data arrives at the station for transmission, the station first of all senses the carrier, i.e. it checks that no other station is currently in transmission. If the link is busy, the station has to back off, i.e. it waits until it finds the channel idle. Note that this involves a continuous sensing of the link, because the station has no idea when the link is going to be free. So the station just ‘listens’ to the link. It observes the frequency transmitted on the link and thus determines when the link is idle. As soon as the link is available, the station starts to transmit its data. It transmits a single data frame, and then introduces a small delay, called the interframe gap, to allow other stations to detect the end of the data frame. After a short period of time into the transmission of a data frame (called the ‘collision window’), all the other stations on the LAN realize that data is being transmitted. If any of the other stations has data to transmit, they have to wait until the link is idle again.

Unfortunately, there is still a chance of two or more stations starting to transmit at the same time. The reason for this is the collision window. For example, if two stations have data to transmit and they are located geographically apart, then they may both find the link idle even though the other station has already started its transmission. This is because the electrical signal takes time to propagate through the link. Thus one station may sense that the link is idle and start transmitting data. This data only arrives at the opposite end of the link after the propagation delay. During this interval, another station may wish to transmit data, and since the electrical signal has not yet propagated fully across the LAN, the station also find the link idle and starts its transmission. Consequently, a collision occurs. For this reason, the CSMA/CD algorithm requires each station to sense the carrier during

its transmission. The station basically compares the signal it transmits to the signal it senses, and if the two are identical, the station knows that no collision has occurred. If, however, the transmitted signal differs from the observed signal, the station realizes that a collision has occurred and it stops its transmission. It then waits for a random amount of time, determined by the IEEE binary exponential backoff algorithm, and re-attempts the transmission after this interval.

Figure 1 shows you the CSMA/CD protocol stack. The Institute of Electrical and Electronic Engineers (IEEE) has standardized this algorithm as 802.3, also known as ethernet. Note that the CSMA/CD protocol operates at layer 2 of the ISO-OSI reference model. Data packets arrive from the network layer. They are then passed through the logical link control (LLC) layer, a common sublayer within all IEEE 802.x protocols, which constructs an LLC frame to perform error control, frame multiplexing and possibly flow control. The packets are then passed down to the CSMA/CD 802.3 MAC layer, which is responsible for segmenting the LLC-frames into MAC-frames and accessing the channel. The typical bandwidth capacity of CSMA/CD links is 10 Mbps. Recent versions of this protocol use bandwidths of 100 Mbps.

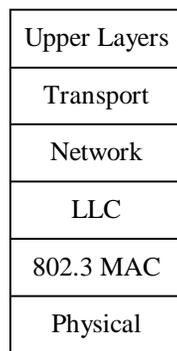


Figure 1: CSMA/CD protocol stack

Laboratories 1 and 2 study particular aspects of this algorithm: the effect of the propagation delay on channel utilization (and thus how collisions occur) and the IEEE binary exponential backoff algorithm. For this reason, we will not discuss this algorithm in more detail here. You should recall though that the utilization for this MAC algorithms is approximately given by

$$\eta_{CSMA/CD} \approx \frac{1}{1 + 5 \frac{T_{prop}}{T_{trans}}}$$

Here, we are also going to look at the maximum medium access time. Recall that the IEEE binary exponential backoff algorithm forces a station to back off for ever longer time intervals. The worst case backoff time is 1024 slot times. This worst case backoff time may arise between 10 and 16 re-transmissions. After the 16<sup>th</sup> re-transmission

attempt, the CSMA/CD algorithm fails and sends a message to the higher layer protocol. Because of this, the algorithm makes no guarantee that the data is being transmitted. The maximum medium access time, defined as the worst-case time that it takes the protocol to access the transmission medium, is therefore infinity for CSMA/CD. We write

$$MMAT_{CSMA/CD} \rightarrow \infty$$

You can justify this answer intuitively, by assuming that an infinite number of stations are attached to the LAN, all wishing to transmit data. Without a guarantee of access, a station may continuously be involved in collisions, and never get exclusive access of the channel!

### 2.2 TOKEN RING

Token Ring LANs differ significantly from CSMA/CD LANs. First of all, they use a ring-based topology, as shown in figure 2. Whether the stations are connected to a single ring, as shown in figure 2, or whether they are connected to each other by point-to-point links is actually irrelevant for our purposes. What is important in this MAC algorithm is that each station has only two neighbors, and that they are connected as a ring, at least logically.

The MAC algorithm is based on a single circulating token. A token is a special data frame, a unique bit sequence (the definition of the token) which is the permission to transmit data. This token is passed around all the stations on the LAN in a round-robin fashion. If a station receives the token and it has no data for transmission, it simply passes the token on to its downstream neighbor. If the station has data to transmit, it removes the token and replaces it with a start-of-frame identifier. The station then appends the data it wants to transmit, and sends the frame to the downstream neighbor.

If a station realizes that it is the destination for the data on the ring, it copies the frame into memory, and still passes the frame on. Eventually, the frame will arrive back at the sender, where it is removed from the ring. The sender then sends a token frame to its downstream neighbor, who either passes it on or appends its own data for transmission. If a station is not the destination of the frame, it simply copies the frame to its respective downstream neighbor.

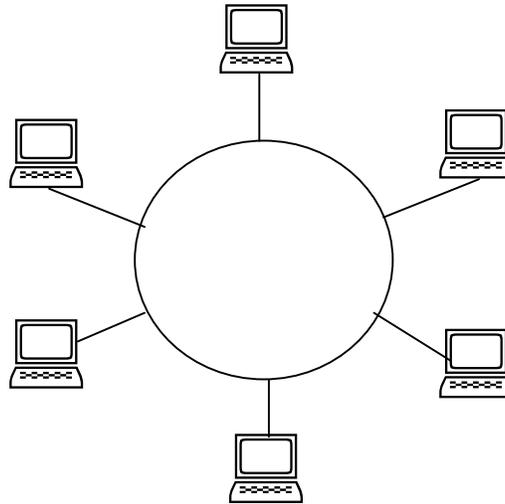


Figure 2: Token Ring Topology

This algorithm basically ensures that each station is allowed to transmit data. Note that a station is *guaranteed* admission to the ring (assuming the networking hardware does not fail). Since there is only one token, only a single station can transmit data at any time, so no collisions occur in a token ring network. For this reason, the token ring MAC algorithm is described as a deterministic access algorithm.

Figure 3 shows you the token ring protocol stack. The IEEE has standardized this algorithm as 802.5. Note that the token ring protocol also operates at layer 2 of the ISO-OSI reference model. Data packets arrive from the network layer. Like with CSMA/CD, they are then passed through the LLC layer for error control, frame multiplexing and possible flow control. The LLC-frames are then passed down to the token ring 802.5 MAC layer, which is responsible for segmenting the LLC-frames into MAC-frames and accessing the channel. Token ring networks typically operate at 4 Mbps or at 16 Mbps.

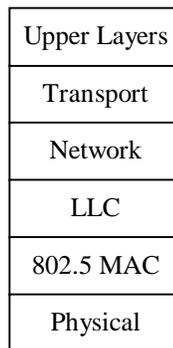


Figure 3: The token ring protocol stack

There are two further important concepts which you need to be aware of in token ring networks: the concept of a token holding time (THT) and the concept of early token release, also known as ‘release after transmission’.

### **2.2.1 RELEASE AFTER TRANSMISSION (RAT)**

The algorithm as described above is quite inefficient for medium access. If a station has data to transmit, the data frame is passed once around the entire ring. If a data frame is small compared to the length of the token ring LAN, it physically only occupies a small portion. In this case you may argue that the efficiency of the algorithm could be increased by allowing other stations to transmit at the same time, i.e. by having more than one token float around the ring. As long as the tokens follow each other and leave sufficient ‘space’ to insert data frames, such a scheme should improve the ring’s efficiency.

This is the idea behind the concept of release after transmission (RAT). In this version of the token ring MAC algorithm, a station re-generates the token immediately after it has transmitted the data frame, rather than waiting for the data frame to return and only re-generating the token then. So effectively, the token is appended at the end of the data frame. This allows multiple data frames to be passed around the ring simultaneously. For example, if the downstream node also has data to transmit, it first receives the data frame, indicated by the start-of-frame field. The station then passes this frame along, but encounters the token at the end of this data frame. Since the token is a permission to transmit, the station may insert its own start-of-frame field and transmit its data, again appending the token at the end. The removal of the data frames from the token operates like above: the destination copies the data frame, but still passes it on to the next station on the ring. Eventually, the data frame returns to the sender, who removes the start-of-frame delimiter and the data. Appended to the data is the token, which in this case is simply passed along (assuming that the sender has no more data to transmit).

### **2.2.2 TOKEN HOLDING TIME**

Another modification to the above description is the incorporation of a token holding time. As described above, a station is allowed to transmit all its data onto the ring. If the station has a lot of data for transmission, the medium access of the other stations on the LAN would be severely delayed. For this reason, any station is only allowed to hold the token for a limited period of time – the token holding time (THT). A station can thus only send as many frames as allowed by the THT, and is then forced by design to pass the token on to the downstream node. The only limitation imposed on the stations is that the transmission of the last data frame within a THT has to be completed. In other words, the frame transmission is only started if there is sufficient time left in the THT to complete the transmission. This effectively limits the transmission time of a single station and guarantees fairness of access among all the stations on the LAN.

### **2.2.3 PERFORMANCE ANALYSIS**

The efficiency of the token ring protocol and the maximum medium access time can easily be determined by a graph as shown in figure 4. This figure depicts the timing of events for one circulation of the token around the ring. We make a heavy traffic assumption, i.e. we assume that each station has data to transmit. Each station is thus allowed to hold the token for the same amount of time, the THT. Each station can transmit as many frames as it is possible to transmit in a single THT episode (note that the transmission of a frame has to be complete – no partial frames may be transmitted). These transmission episodes of the stations are indicated by THT, followed by the station number in figure 4. Following the transmission of data frames, each station then transmits the token. This is indicated by TK in the diagram. Following the token transmission time the signal requires some time to propagate to the downstream node, as shown by  $T_{prop}^{(X,Y)}$  in the diagram. Here, the first integer represents the transmitting station, whereas the second integer represents its downstream neighbor. The sum of all propagation delays equals the full propagation delay around the ring, and is thus dependent on the physical length of the ring.

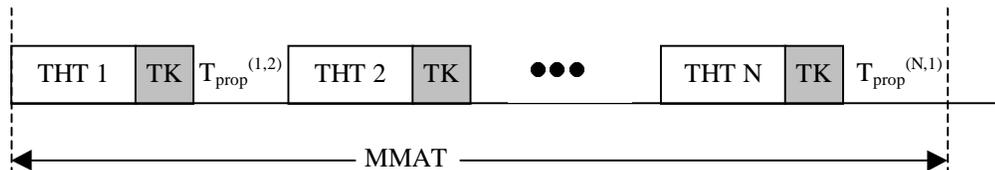


Figure 4: Timing diagram for the token ring MAC algorithm

Recall that the utilization of the algorithm is typically computed as the time in a full cycle that is spent on the transmission of useful frames, i.e. data frames. If we assume that there are  $N$  stations on the ring, and that each station always has data to transmit, then the length of a cycle, call it  $S$ , can be computed as

$$S = N(THT + TK) + T_{prop}$$

where (assuming that the  $N+1^{\text{st}}$  station is the first station)

$$T_{prop} = \sum_{X=1}^N T_{prop}^{(X, X+1)}$$

Therefore, the efficiency of the token ring algorithm is given by

$$\eta_{TR} = \frac{N * THT}{S}$$

To determine the MMAT, we again have to consider the worst case. The worst time for a data frame to arrive at a station is the time when the station has just started to transmit the last data frame that it is allowed to send in its THT. If the station were to transmitted another data frame, it would violate its THT. In this case, the station has to transmit this last data frame, followed by the token, and then has to wait for the  $(N-1)$  stations to finish

their respective transmission episodes. You can again use figure 4 to verify that the MMAT is given by

$$MMAT_{TR} = T_{trans} + T_{prop} + N * TK + (N - 1)THT$$

where  $T_{trans}$  indicates the time to transmit a data frame. Note that in contrast to the CSMA/CD algorithm, this performance value is *finite*.

### 2.3 BRIDGES

Bridges are used to connect LANs of different types together. This may sound more trivial than it actually is. To achieve such connectivity, bridges have to handle a variety of functions, such as different bandwidth capacities, frame sizes, protocol functions or address formats. Let us discuss these issues briefly.

CSMA/CD and token ring networks operate at different link speeds. The former protocol typically uses a bandwidth capacity of 10 Mbps, whereas the latter protocol often operates at speeds of 16 Mbps. A data stream arriving from a CSMA/CD network can be forwarded onto a token ring network at a higher speed. However, a data stream arriving from a token ring network can only be transmitted at 10 Mbps. Even if we assume that the bridge has exclusive access to the CSMA/CD LAN at the time the message arrives, the arrival rate still exceeds the transmission rate. To provide a reliable service, the bridge has to be able to buffer data to accommodate the different link speeds.

Secondly, the two protocols use different frame sizes. The data frames used by CSMA/CD and token ring are not identical. This implies that the bridge has to restore the LLC-frames and then re-segment into the required frame format used by the downstream link. This operation takes time, of course, and thus introduces additional delays.

Furthermore, the two MAC-protocols also offer different services. For example, under the token ring algorithm, a frame may be given a priority. On the other hand, no priorities exist under CSMA/CD. For our purposes it is enough that you are aware of this point, but we will not explain the detailed priority mechanism for token ring networks. It is sufficient to point out that the bridge has to handle the situation of eliminating / introducing priorities when transmitting traffic across dissimilar LANs.

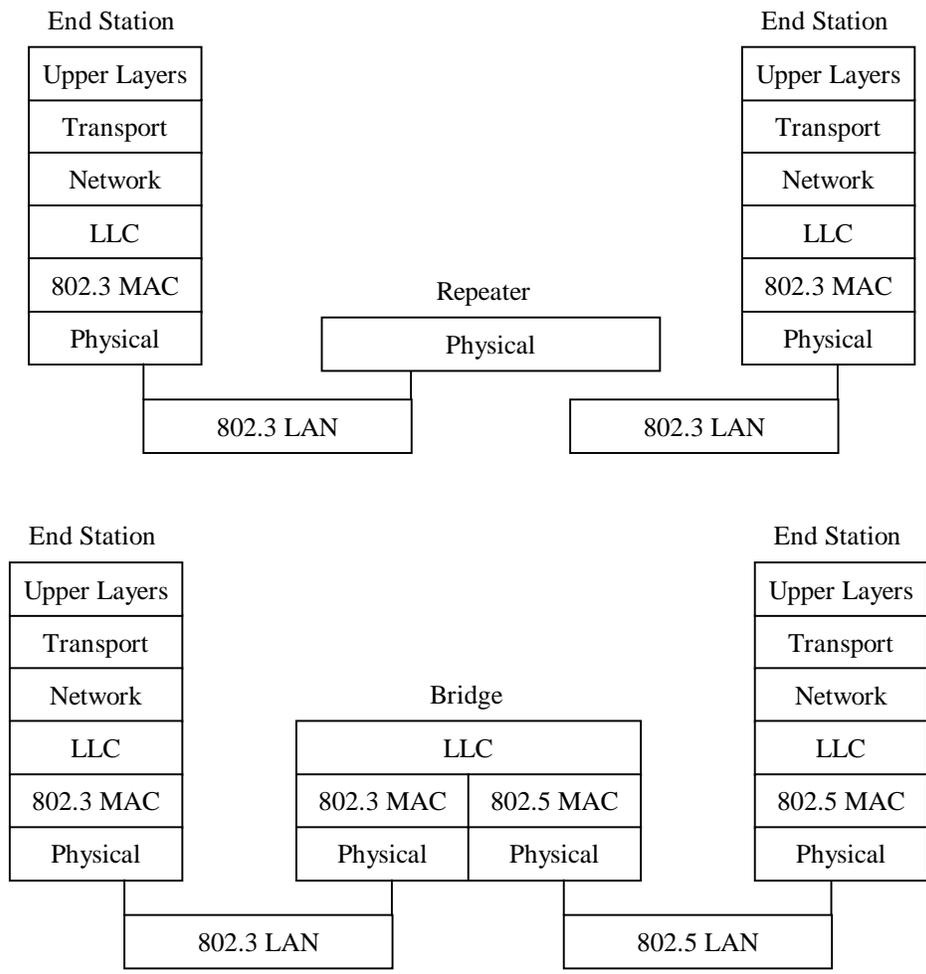
Interconnecting multiple LANs with bridges has tremendous advantages, though. It effectively increases the throughput on the LAN. Consider the following simple example:

Example 1: 100 stations are connected to a single CSMA/CD LAN at 10 Mbps. The large number of stations means that many collisions will occur on the LAN. If you introduce a bridge and create two LANs, each with 10 Mbps and 50 workstations attached, you effectively double the link capacity. At any point in time, two CSMA/CD frames may be in transmission. Furthermore,

since only 50 stations are now contending for the link, the number of collisions will decrease.

Another advantage of bridging LANs together is the geographical coverage. All LANs have a limit on the maximum length of the cable, also called a segment. Connecting two LAN segments by a bridge doubles the geographical distance that can be covered by the LAN.

Figure 5 shows you a graphical representation of a bridge, and compares it to a router and a repeater. A repeater is the simplest device to increase the range of a LAN. It operates at the physical layer of the ISO-OSI reference model. Repeaters can only be used for LANs of the same type. It takes any signal on either segment, and re-transmits it on the other segment. Note that this does not reduce the number of collisions when two CSMA/CD LANs are connected. The repeater effectively only extends the physical range, and the LAN behaves as if all stations were connected to a single link. A router, on the other hand, operates on ISO-OSI layer 3. The original packets are re-assembled, and then a routing decision is made. Routers thus introduce even more delays than bridges, since the data link layer functions have to be executed twice. They have the advantage of being able to connect networks with different data link protocols, e.g. connecting a token ring LAN to an X.25 WAN.



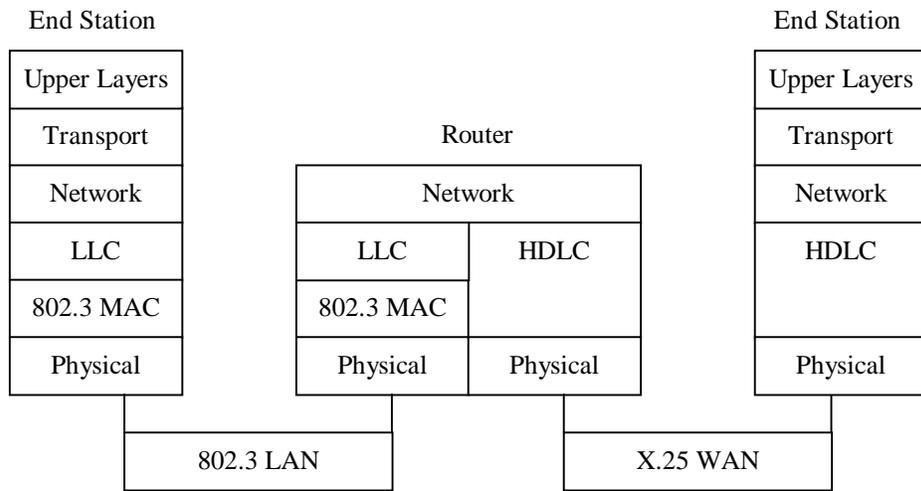


Figure 5: Layered representation of a repeater (a), bridge (b) and router (c)

### 3 EXPERIMENT 1

In the first experiment of this lab, you will be asked to build two small model of a CSMA/CD LAN. Here is the description for this experiment:

The company Network Solutions Inc. has 10 employees each with their own workstation. The IS manager is now considering to network these workstations together. In particular, the IS manager is considering a 10 Mbps 802.3 CSMA/CD Ethernet and a 16 Mbps 803.5 Token Ring network to obtain connectivity. He is asking you to demonstrate the technical differences between the two technologies using the simulator COMNET III. In particular, the IS manager wants to know the performance difference that arises from the differences between the two medium access schemes, as opposed to the differences arising from the different bandwidth standards.

The traffic profile per user is as follows:

- Each user generates constant 37500 byte messages.
- The inter-arrival time of the messages is exponentially distributed with a mean of 1 second.
- The messages use a generic transport protocol with a maximum of 1000 data bytes per packet.
- The messages are sent to any other employee in the company with equal probability, (but obviously not to the employee's own workstation).

#### Part 1:

- Build a COMNET III model of this network, using a computer group node to model 9 of the 10 stations and a processing node (computer & communications node) to model the 10th station. Model the LAN using the *default* CSMA/CD link type. Model the messages as message sources.
- Make sure that the Generic transport protocol has a maximum of 1000 data bytes per packet by clicking on 'Define / Protocols / Edit / Edit / Basic Protocol'. Change the packet size to 1000 bytes if necessary. Then check that your message generators use this transport protocol by bringing up the dialog box for the message source and clicking on the '..' next to 'Trans Protocol'. Then click on 'Edit' under the box 'Transport Protocols' (not Library Selections), and finally click on 'Basic Protocol'. You should have a value for 'data bytes' of 1000 under the upcoming dialog box.
- Select the following reports:
  - Nodes: Received Message Counts
  - Links: Channel Utilization
  - Links: Collision Stats
  - Message & Response Sources: Message Delay
  - Message & Response Sources: Packet Delay
- Simulate the network for 60 seconds without warmup. Note: the simulation will run a lot faster if you switch the animation off.

Part II:

- Modify your model from Part I by changing the link from an IEEE 802.3 LAN to a 10 Mbps (note the bandwidth!) token passing IEEE 802.5 LAN (use the *default* parameter set and modify the appropriate parameters).
- Replace the report 'Links: Collision Stats' selected above with the following:
  - Links: Token Ring Stats
- Simulate the network for 60 seconds without warmup. Note: the simulation will run a lot faster if you switch the animation off.

To Do: Provide a timing diagram for token ring under the assumption that the token is only re-generated after the data frame has completely circulated around the ring (i.e. without the RAT assumption).

Provide a brief report to the IS manager (max. 2 pages, type-written), interpreting the differences in the results between the CSMA/CD and the token ring MAC under the current setting.

## 4 EXPERIMENT 2

In this second experiment, you will now be asked to build a model with a bridge. Here is the description for this experiment:

A bridge connects LAN's of different types. Build a model which connects a 10 Mbps IEEE 802.3 LAN with a 16Mbps IEEE 802.5 LAN. Each LAN has 10 stations attached, which generate local traffic as in the last experiment, Part I. In addition, the single station on the IEEE 802.3 LAN (call it A) transfers a file of size 0.5 MB to the single station on the IEEE 802.5 LAN (call it B) with an exponentially distributed inter-arrival time of 10 seconds on average. Similarly, B transfers a file to A with the same file size and inter-arrival parameters.

- Use the models from Part I and Part II as a basis to build a model of this network. Model the bridge as a processing node which is connected to both LANs, but make sure that the bridge is not a final destination for any of the messages. The file transfers are again modeled by message sources following the generic transport protocol.
- In addition to the reports generated from above, select the following report for the bridge only:
  - Nodes: Buffers by node: Input and Output buffer totals
- Simulate the model for 60 seconds
- Provide a brief report to the manager (1 page, type-written), interpreting the results. Concentrate on the file transfer and buffer statistics as well as on the message delays.

## 5 DELIVERABLES:

### Experiment 1, Part I:

- Export your model to a file, using 'File / Export / External Model' (make sure that the names you have given to the model elements are unique!). *Provide only a print-out of the following sections of this file:*
  - Node Params
  - Link Params
  - Nodes
  - Links
  - Sources
- Print-out the reports generated by the simulation.
- Provide a screen-shot of your model.

### Experiment 1, Part II:

- Print-out of the exported model file (c3e format) - as above.
- Print-out of the reports.
- Provide a screen-shot of your model.
- Report to the manager.

### Experiment 2:

- Print-out of the exported model file (c3e format) - as above.
- Print-out of the reports.
- Provide a screen-shot of your model.
- Report to the manager.

Note: All reports have to be in letter format. All interpretations have to be typed.

### **Marking Scheme:**

Interpretations: 50%

Models: 35%

Format: 15% (this includes completeness, clarity, form)

---

**END OF LABORATORY 2**