

**University of Dhaka**

**Department of Computer Science and Engineering**

**CSE 3112- Data and Telecommunications Lab**

2nd Year – 2nd Semester

Session: 2016 - 2017

**Experiment No.: 02**

**Name of the Experiment:**

Emulation of Sync-TDM and Stat-TDM

**Submitted by:**

M. Tanjid Hasan Tonmoy, Roll: 09

Sakib Hasan, Roll: 63

Experiment Date: 27-07-2017

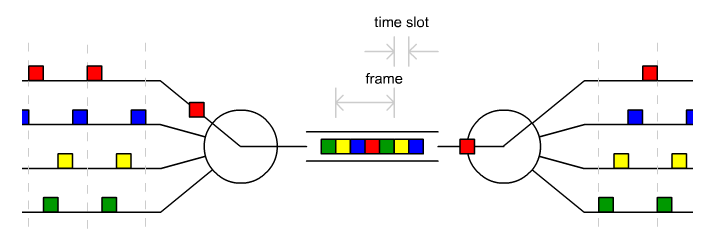
Submission Date: 20-08-2017

**Submitted to:**

1. Dr. Md. Rezaul Karim, Professor, Dept. of CSE, DU
2. Ms. Iffat Anjum, Lecturer, Dept. of CSE, DU

|  |
| --- |
| Problem definition: |

This experiment emulates Synchronous Time Division Multiplexing (Sync-TDM) and Statistical Time Division Multiplexing (Stat-TDM). Consider two different traffic models: *backlogged traffic model*, where data values are always available from n input files and *Random process model,* where the presence of traffic from a particular file is random.



Inputs & Outputs:

We considered that n = 5 (5 files with different sizes, should need more than five individual slots to send an entire file), slot size: 10 Byte. Each slot will have a specific format like below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Start Marker** | **Source Address** | **Destination Address** | **Data** | **End Marker** |

The frame format will be like below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Start Marker** | **Slot 01** | **Slot 02** | **Slot 03** | **Slot 04** | **Slot 05** | **End Marker** |

|  |
| --- |
| Equipment: |

* Desktop/Laptop
* Netbeans IDE
* JDK 8

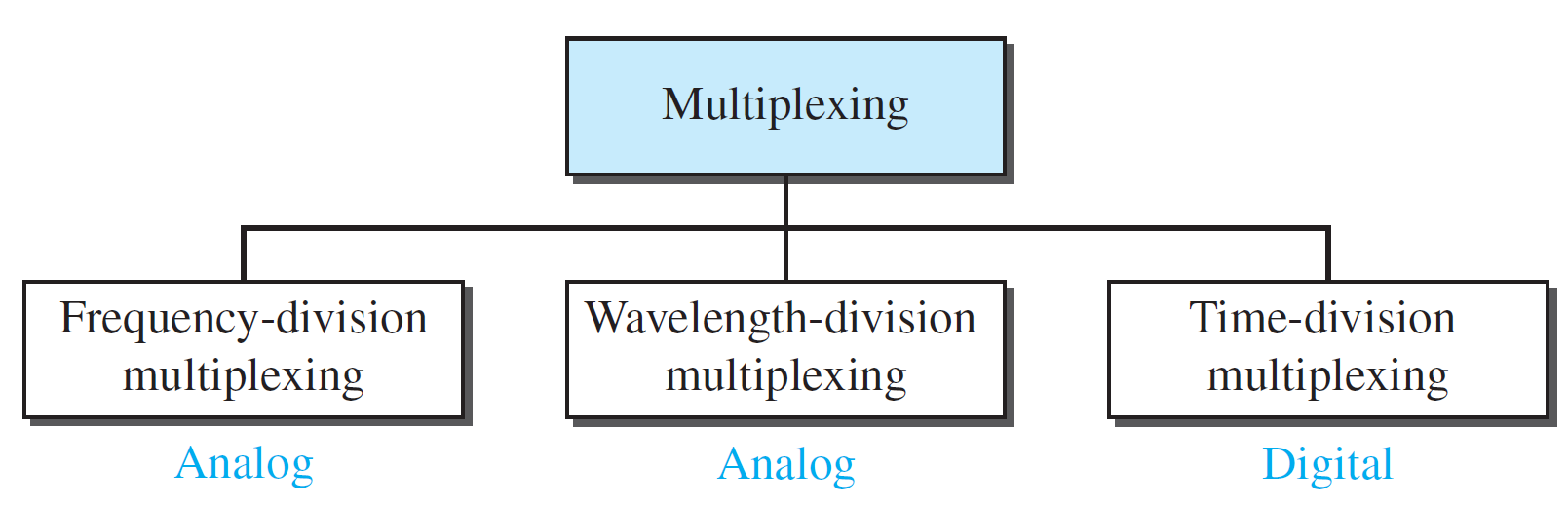
|  |
| --- |
| Theoretical background: |

Multiplexing: Multiplexing (or *muxing*) is a way of sending multiple signals or streams of information over a communications link at the same time in the form of a single, complex [signal](http://searchnetworking.techtarget.com/definition/signal). Multiplexing is the set of techniques that allow the simultaneous transmission of multiple signals across a single data link.

Networks use multiplexing for two reasons:

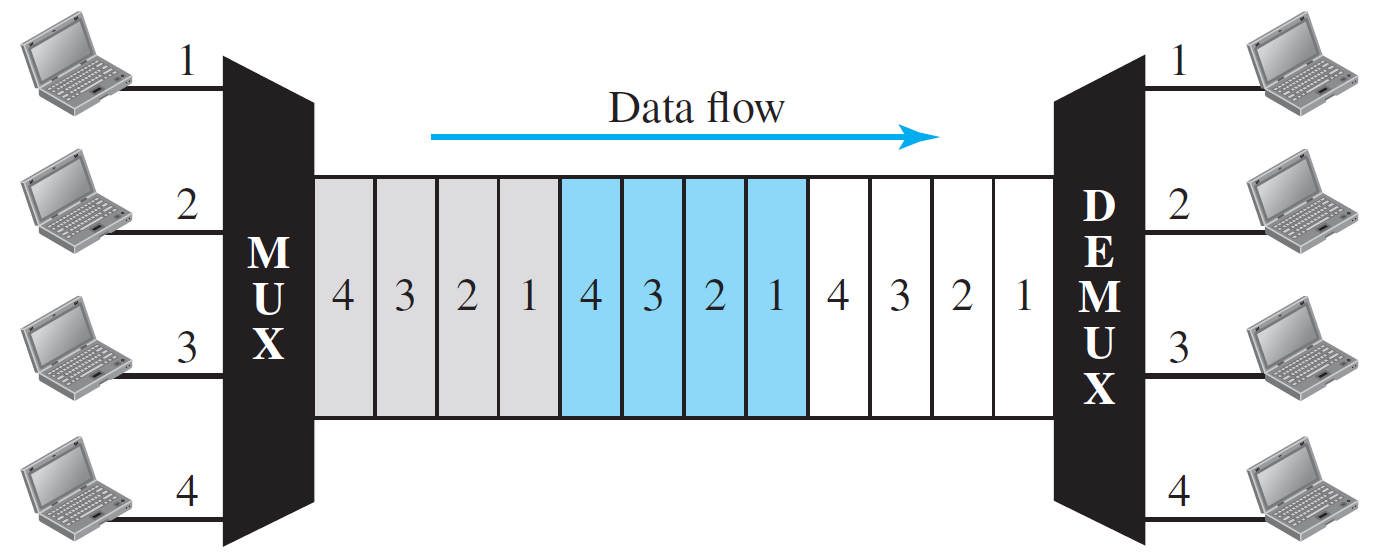
* To make it possible for any network device to talk to any other network device without having to dedicate a connection for each pair. This requires shared media.
* To make a scarce or expensive resource stretch further (to send many signals down each cable or fiber)

There are three basic multiplexing techniques: frequency-division multiplexing, wavelength-division multiplexing, and time-division multiplexing. The first two are techniques designed for analog signals, the third, for digital signals



**Time-Division Multiplexing:**

Time-division multiplexing (TDM)is a digital process that allows several connections to share the high bandwidth of a link. Instead of sharing a portion of the bandwidth as in FDM, time is shared. Each connection occupies a portion of time in the link. Following Figure gives a conceptual view of TDM. In the figure, portions of signals 1, 2, 3, and 4 occupy the link sequentially.



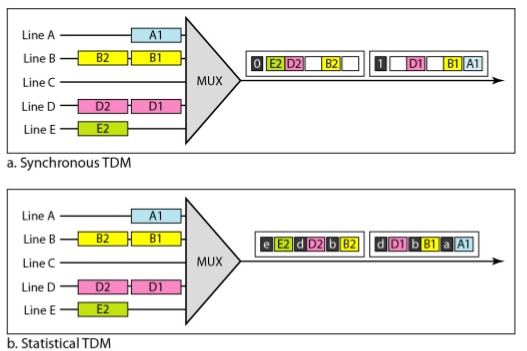
In this figure, all the data in a message from source 1 always go to one specific destination, be it 1, 2, 3, or 4.

We can divide **TDM** into two different schemes:

1. Synchronous TDM
2. Statistical TDM.

Synchronous TDM – In synchronous TDM, each input connection has an allotment in the output even if it is not sending data. In synchronous TDM, the data rate of the link is *n* times faster and the unit duration is *n* times shorter.

Statistical TDM – In statistical time-division multiplexing, slots are dynamically allocated to improve bandwidth efficiency. Only when an input line has a slot’s worth of data to send is it given a slot in the output frame. In statistical multiplexing, the number of slots in each frame is less than the number of input lines.



* Advantages of TDM over other multiplexing techniques: TDM provides greater flexibility and efficiency, by dynamically allocating more time periods to the signals that need more of the bandwidth, while reducing the time periods to those signals that do not need it. FDM/WDM lacks this type of flexibility, as it cannot dynamically change the width of the allocated frequency. FDM divides the channel into two or more frequency ranges / WDM divides the channel into two or more wavelength ranges that do not overlap, while TDM divides and allocates certain time periods to each channel in an alternating manner. Due to this fact, we can say that for TDM, each signal uses all of the bandwidth some of the time, while for FDM/WDM, each signal uses a small portion of the bandwidth all of the time.

|  |
| --- |
| Working principle: |

* 1. How did you accomplish your Lab05 task

**Pseudo code for our program:**

***Client:***

Procedure Main:

Initialize socket and connect to server through a port

Initialize input stream and output stream

Initialize FileReader for all the input files

Create BufferedReader with FileReader

Declare an Object of Random class

Declare the start marker, end markers and other delimiters

FOR no of iterations

Declare StringBuilder Object for creating frame

FOR no of input files

Generate random number

IF random number>50

Skip this file

READ 10 bytes with the Corresponding BufferedReader

Create a String with the bytes read

 Append the String to frame as a slot with delimiters

PRINT the frame

SEND the frame to server

Close the input, output streams

END Procedure main

***Server:***

Procedure Main:

Create ServerSocket with a port

Connect Client through socket

Initialize input stream, output stream

Initialize FileWriter for all the output files

Declare the start marker, end markers and other delimiters

WHILE InputStream available

Receive frame as a String

PRINT the frame

Get the slots from frame by tokenizing the frame with delimiters

FOR no of slots in the frame

Separate data from destination address

WRITE the data to destination file using FileWriter

END WHILE

Close the input and output streams

END Procedure main

**Main challenges and their solutions**

* Sync/Stat TDM is visualized and used through hardware. Making a simulation of this with given constraints in *JAVA* was quite a challenge.
* Reading a fixed no of bytes instead of the entire line – solution: BufferedReader.
* Reading from multiple files.
* For Statistical TDM creating different size of frames was not easy – solution: we used array of string to keep the data and made a method to create different sized frame using existing slots with data only

|  |
| --- |
| Discussion: |

**Applications of TDM:**

* [PCM](https://en.wikipedia.org/wiki/Pulse-code_modulation)(Pulse-code modulation) system, for digital transmission of several telephone calls over the same copper cable or fiber cable in the circuit switched digital telephone network
* The CDMA mobile phone service is based on time division multiplexing.
* The [synchronous digital hierarchy (SDH)/synchronous optical networking (SONET)](https://en.wikipedia.org/wiki/Synchronous_optical_networking) network transmission standards that have replaced PDH.

TDM can be further extended into the [time-division multiple access](https://en.wikipedia.org/wiki/Time-division_multiple_access) (TDMA) scheme, where several stations connected to the same physical medium, for example sharing the same [frequency](https://en.wikipedia.org/wiki/Frequency-division_multiplexing) channel, can communicate. It’s mostly use now in –

* The [GSM](https://en.wikipedia.org/wiki/Global_system_for_mobile_communications) telephone system
* The Tactical Data Links [Link 16](https://en.wikipedia.org/wiki/Link_16) and [Link 22](https://en.wikipedia.org/wiki/Link_22). Link 16, mostly used for military purpose (*aeronautical radio navigation* service and *radio navigation satellite* service)  is a [TDMA](https://en.wikipedia.org/wiki/Time_division_multiple_access)-based [secure](https://en.wikipedia.org/wiki/Security), jam-resistant, high-speed [digital](https://en.wikipedia.org/wiki/Digital_data) data link

**Real Life examples:**

Actually TDMA is the technology which is being used by  
telecommunication companies to make communication between mobile  
users, with GSM in it a time slot is allocated to every user in particular cell.  
Slots are allocated in TDMA when a user enters in new cell .Only major disadvantage is that a user cannot continue its call when he is entering in new cell and it’s all time slots are full.

Let’s suppose, there are approximately 1500 people in a colony and statistics say that 15 persons makes a call at most at the same time. So keeping 1500 wire connections for 1500 persons would be a waste as most of the connection will remain idle most of the time. So to reduce cost and complexity we just can have 100 wires for 1500 men supposing 15 men makes a call at the same time. By multiplexing/ using TDM we can solve our problem. Stat TDM will be more efficient in this case and save more cost.

1. Any hardware that support TDM, trainer boards (for example any chip, IC or device that perform TDM on given inputs, are those available in Bangladesh?
2. Comparative analysis
3. Different implementations and versions (other than sync and stat)

* Describe at least one of its variants.

**

*🡪 Why is stat-TDM performs better than sync-TDM?*

*🡪 Is there any situation where the reverse is true?*

**Answers:**

1. In synchronous TDM, all party’s clocks need to be aligned with some reference clock, which could be provided by one of the involved systems.

In statistical TDM, the timeslots are not fixed. They are assigned dynamically as needed. So, Stat-TDM performs better because slots are allocated only if there is data packets. This leads to better throughput.

1. Slots in Synchronous TDM carry data only and there is no need of addressing. Synchronization and pre assigned relationships between input and outputs that serve as an address.

Slots in Statistical TDM contain both data and address of the destination.

1. Synchronization bits are used at the beginning of each frame in sync TDM. In stat TDM, no synchronization bits are used.

So, stat-TDM performs better and faster than sync-TDM.

Reverse is so if it is guaranteed that input is available all the time. In this case the maximum utilization of bandwidth is possible and no address bits need to be sent reducing overhead.

**Throughput:** Throughput is a measure of how many units of information a system can process in a given amount of time. It is used to measure the performance of [hard drives](https://techterms.com/definition/harddrive) and [RAM](https://techterms.com/definition/ram), as well as Internet and network connections.

For example, a hard drive that has a maximum transfer rate of 100 [Mbps](https://techterms.com/definition/mbps) has twice the throughput of a drive that can only transfer data at 50 Mbps. Similarly, a 54 Mbps wireless connection has roughly 5 times as much throughput as an 11 Mbps connection. However, the actual data transfer speed may be limited by other factors such as the Internet connection speed and other network traffic. Therefore, it is good to remember that the maximum throughput of a device or network may be significantly higher than the actual throughput achieved in everyday use.

**Calculate Throughput:** A typical method of performing a measurement is to transfer a 'large' file from one system to another system and measure the time required to complete the transfer or copy of the file. The throughput is then calculated by dividing the file size by the time to get the throughput in [megabits](https://en.wikipedia.org/wiki/Megabit), [kilobits](https://en.wikipedia.org/wiki/Kilobit), or [bits](https://en.wikipedia.org/wiki/Bit) per second.

|  |
| --- |
| Graphs: |

Assumptions:

* Data rate: 100 Byte/ sec
* Number of Slots for Sync-TDM: 10

1. (y-axis: throughput of sync-TDM) vs (x-axis: Number of same sized files
   1. (y-axis: throughput of stat-TDM) vs (x-axis: Number of same sized files (vary from 3-10))
   2. (y-axis: throughput of sync-TDM) vs (x-axis: Number of different sized files (vary from 3-10))
   3. (y-axis: throughput of stat-TDM) vs (x-axis: Number of different sized files (vary from 3-10))