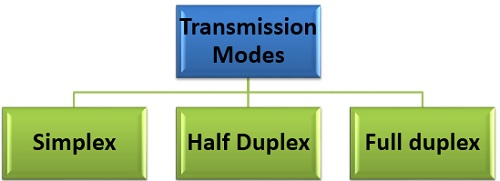
**Modes of Communication**



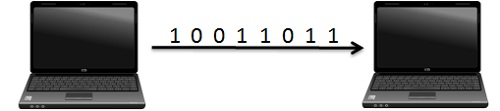
There are three modes of transmission simplex, half duplex, and full duplex. Transmission mode describes the direction, of flow of signal between two connected devices. The main difference between simplex, half duplex, and full duplex is that in a**simplex** mode of transmission the communication is unidirectional whereas, in the **half-duplex** mode of transmission the communication is two directional but the channel is alternately used by the both the connected device. On the other hand, in the **full duplex** mode of transmission, the communication is bi-directional, and the channel is used by both the connected device simultaneously.

| **BASIS FOR COMPARISON** | **SIMPLEX** | **HALF DUPLEX** | **FULL DUPLEX** |
| --- | --- | --- | --- |
| Direction of Communication | Communication is unidirectional. | Communication is two-directional but, one at a time. | Communication is two directional and done simultaneously. |
| Send/Receive | A sender can send data but, can not receive. | A sender can send as well as receive the data but one at a time. | A sender can send as well as receive the data simultaneously. |
| Performance | The half duplex and full duplex yields better performance than the Simplex. | The full duplex mode yields higher performance than half duplex. | Full duplex has better performance as it doubles the utilization of bandwidth. |
| Example | Keyboard and monitor. | Walkie-Talkies. | Telephone. |

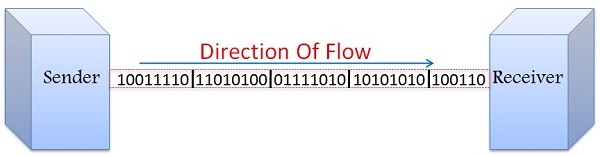
**Types of Data Communication/Transmission**

Serial Transmission

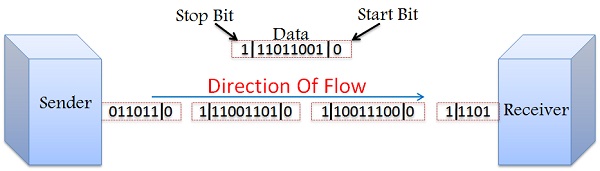
In Serial Transmission, data is sent bit by bit from one computer to another in bi-direction. Each bit has its clock pulse rate. Eight bits are transferred at a time having a start and stop bit (usually known as a Parity bit) i.e. 0 and 1 respectively. For transmitting data to a longer distance, data cables are used. It consists of D-shaped 9 pin cable that connects the data in series.



Serial Transmission has two subclasses **synchronous** and **asynchronous**. In asynchronous transmission, an extra bit is added to each byte so that the receiver is alert about the arrival of new data. Usually, 0 is a start bit, and 1 is the stop bit. In synchronous transmission, no extra bit is added rather the data transferred in the form of frames which contains multiple bytes.

In **Synchronous Transmission**, data flows in a full duplex mode in the form of blocks or frames. Synchronization between the sender and receiver is necessary so that the sender know where the new byte starts (since there is no gap between the data).  
  
Synchronous Transmission is efficient, reliable and is used for transferring a large amount of data. It provides real-time communication between connected devices. Chat Rooms, Video Conferencing, telephonic conversations, as well as face to face interactions, are some of the examples of Synchronous Transmission. In general, Processor memory buses are synchronous in nature. The typical speed is greater than 20Kbps.

In **Asynchronous Transmission,** data flows in a half duplex mode, 1 byte or a character at a time. It transmits the data in a continuous stream of bytes. In general, the size of a character sent is 8 bits to which a parity bit is added i.e. a start and a stop bit that gives the total of 10 bits. It does not require a clock for synchronization; rather it uses the parity bits to tell the receiver how to interpret the data.

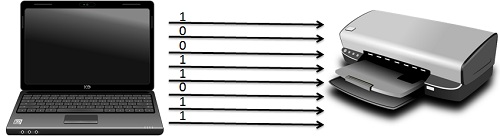
  
It is simple, fast, economical and does not require a 2-way communication. Letters, emails, forums, televisions and radios are some of the examples of Asynchronous Transmission. Generally I/O buses are asynchronous in nature. The data rate is typically less than 20kbps.

Synchronous Vs Asynchronous Transmission

1. In Synchronous Transmission data is transferred in the form of frames on the other hand in Asynchronous Transmission data is transmitted 1 byte at a time.
2. Synchronous Transmission requires a clock signal between the sender and receiver so as to inform the receiver about the new byte. Whereas, in Asynchronous Transmission sender and receiver does not require a clock signal as the data sent here has a parity bit attached to it which indicates the start of the new byte.
3. Data transfer rate of Asynchronous Transmission is slower than that of Synchronous Transmission.
4. Asynchronous Transmission is simple and economic whereas, Synchronous Transmission is complex and expensive.
5. Synchronous Transmission is efficient and has lower overhead as compared to the Asynchronous Transmission.

**Parallel Transmission**

In Parallel Transmission, various bits are sent together simultaneously with a single clock pulse. It is a fast way to transmit as it uses many input/output lines for transferring the data.



Parallel Transmission uses a 25 pin port having 17 signal lines and 8 ground lines. The 17 signal lines are further divided as

* 4 lines that initiates handshaking,
* 5 status lines used to communicate and notify errors and
* 8 to transfer data.

**Serial vs Parallel Transmission**

1. Serial transmission requires a single line to communicate and transfer data whereas, parallel transmission requires multiple lines.
2. Serial transmission used for long distance communication whereas, the parallel transmission used for shorter distance.
3. Error and noise are least in serial as compared to parallel transmission. Since one bit follows another in Serial Transmission whereas, in Parallel Transmission multiple bits are sent together.
4. Parallel transmission is faster as the data is transmitted using multiples lines whereas, in Serial transmission data flows through a single wire.
5. Serial Transmission is full duplex as the sender can send as well as receive the data whereas, Parallel Transmission is half duplex since the data is either sent or received.
6. Serial transmission cables are thinner, longer and economical in comparison with the Parallel Transmission cables.
7. Serial Transmission is reliable and straightforward whereas, Parallel Transmission is unreliable and complicated.