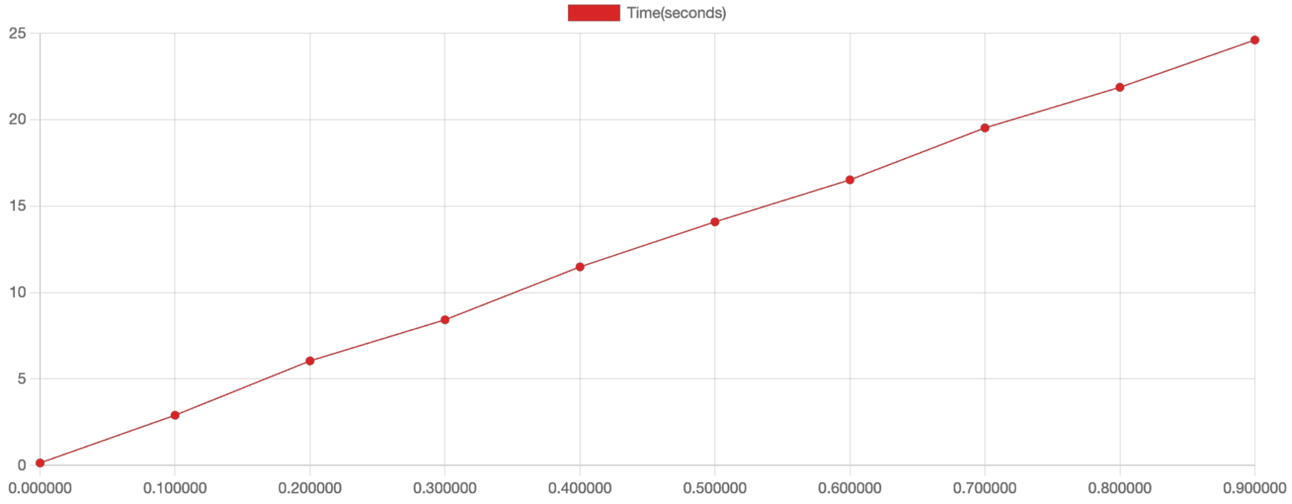
Computer Networks

Assignment 4
Group 12

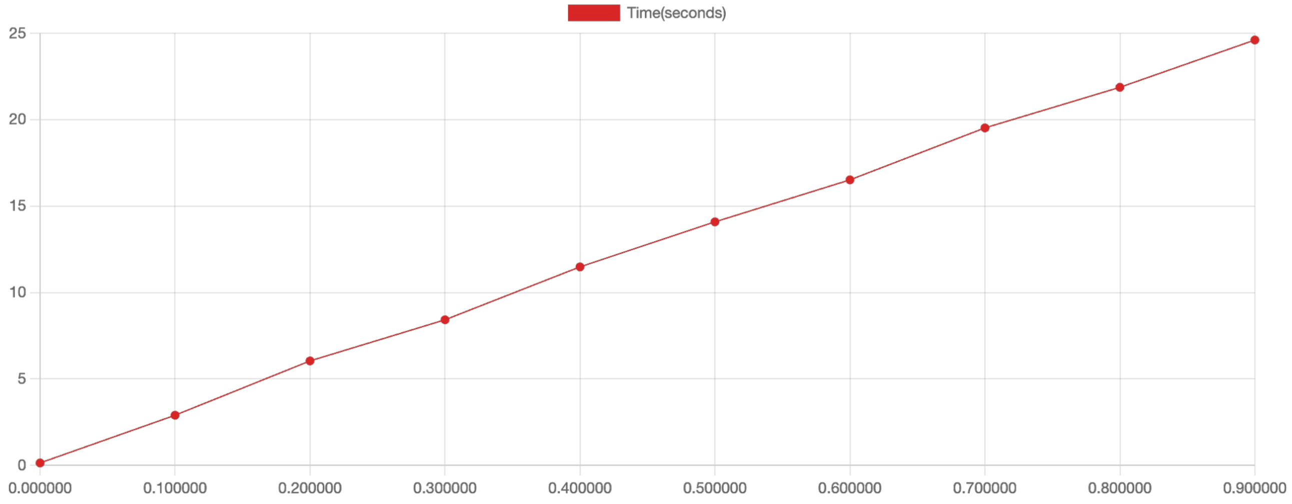
Ishan Arif Mardani (21CS01023)
Vishnu Tirth Bysani (21CS01032)
Akshit Dudeja (21CS01026)

Stop and Wait (Time in Seconds):

SELECT



POLL

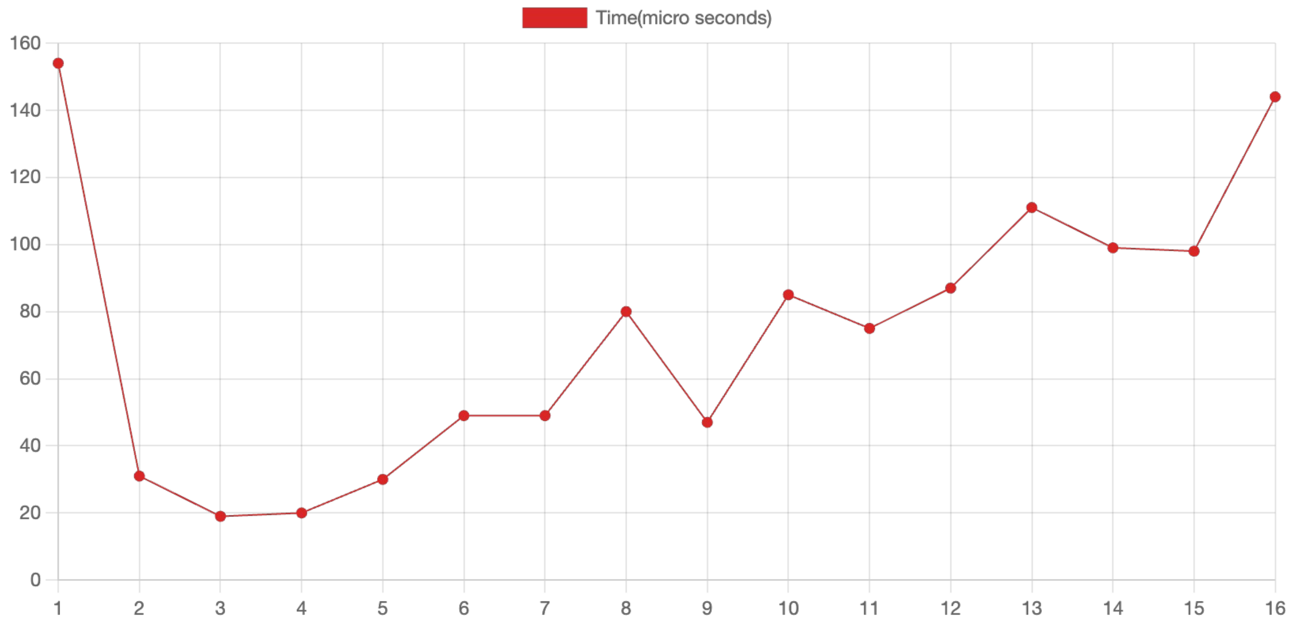


Explanation:

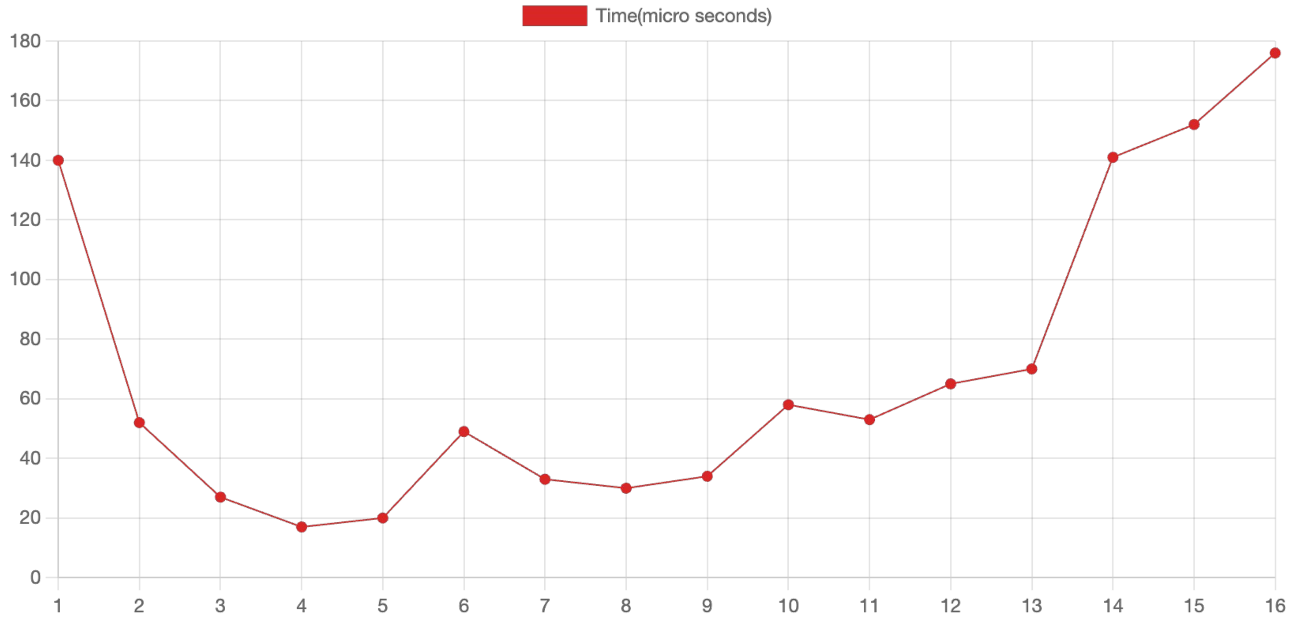
The trend we observe is that the time required for the file to be successfully transferred increases as the probability of packets being lost increases.
This is beacuse as probability of packet loss increases, the sender's timeout mechanism triggers a retransmission of the lost frame, until the ACK for that packet has been received. There can be multiple retransmissions occuring due to higher packet loss probability which can lead to further delays.

Go Back N (Time in Micro Seconds):

SELECT



POLL



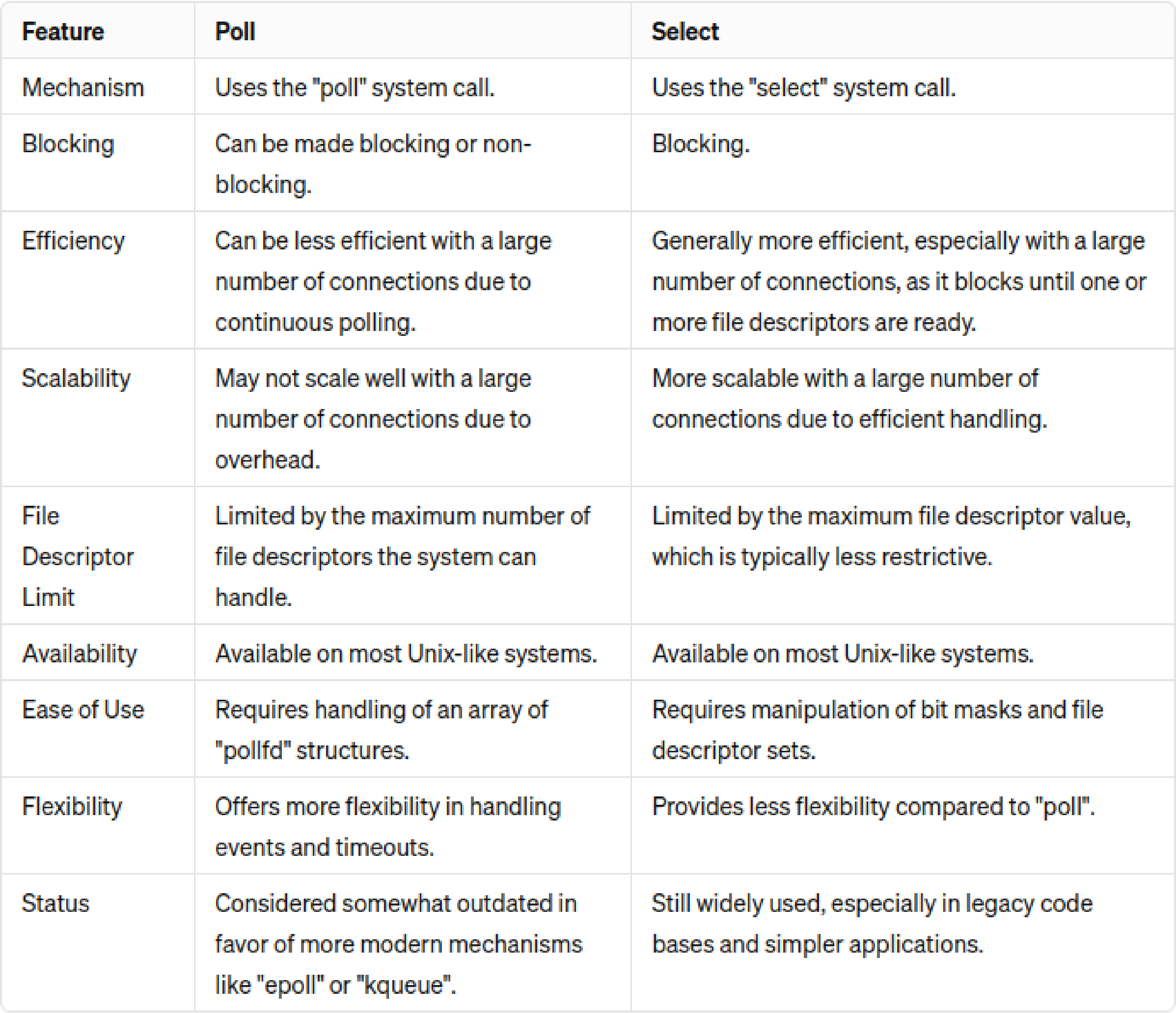
Explanation:

The trend we observe is that as the value of n increases, the time required for the file to be successfully transferred first decreases then increases.

As the window size (n) increases, more packets can be sent without waiting for acknowledgments. During this phase, the sender can transmit multiple packets without needing to wait for individual acknowledgments. This concurrent transmission of packets can reduce the overall time taken to transmit a large number of packets since acknowledgments for multiple packets can be received in parallel, thus reducing the time for file transfer.

However, as the window size continues to increase beyond a certain threshold, the probability of packet loss also increases. If any packet within the window is lost or damaged and the receiver will discard the out-of-order packet. As the sender has not received an acknowledgement it must retransmit all the packets in the current window. As window size increases the chance of packet loss increases and the sender may need to retransmit a significant number of packets, which can increase the overall time taken for successful transmission.

SELECT VS POLL:



Feature	Poll	Select
Mechanism	Uses the "poll" system call.	Uses the "select" system call.
Blocking	Can be made blocking or non-blocking.	Blocking.
Efficiency	Can be less efficient with a large number of connections due to continuous polling.	Generally more efficient, especially with a large number of connections, as it blocks until one or more file descriptors are ready.
Scalability	May not scale well with a large number of connections due to overhead.	More scalable with a large number of connections due to efficient handling.
File Descriptor Limit	Limited by the maximum number of file descriptors the system can handle.	Limited by the maximum file descriptor value, which is typically less restrictive.
Availability	Available on most Unix-like systems.	Available on most Unix-like systems.
Ease of Use	Requires handling of an array of "pollfd" structures.	Requires manipulation of bit masks and file descriptor sets.
Flexibility	Offers more flexibility in handling events and timeouts.	Provides less flexibility compared to "poll".
Status	Considered somewhat outdated in favor of more modern mechanisms like "epoll" or "kqueue".	Still widely used, especially in legacy code bases and simpler applications.