Experiment-6

**Date**-May 20, 2021.

**AIM-** Implementation and study of Go back-N and Selective repeat Protocols.

**Go-Back-N Protocol**

Go-Back-N protocol, also called Go-Back-N Automatic Repeat request, is a data link layer protocol that uses a sliding window method for reliable and sequential delivery of data frames. It is a case of sliding window protocol having to send window size of N and receiving window size of 1.

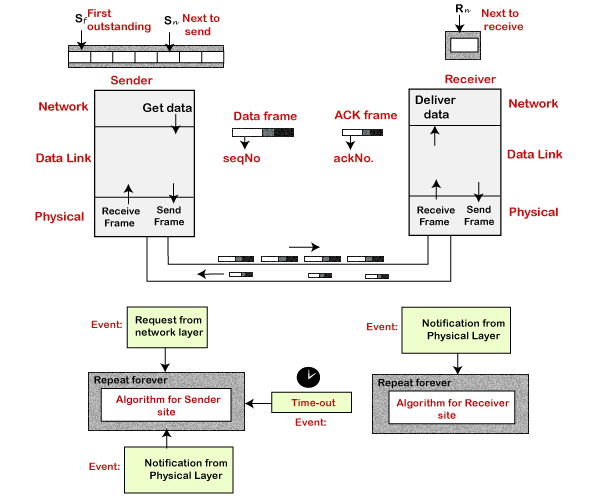
## **Working Principle**

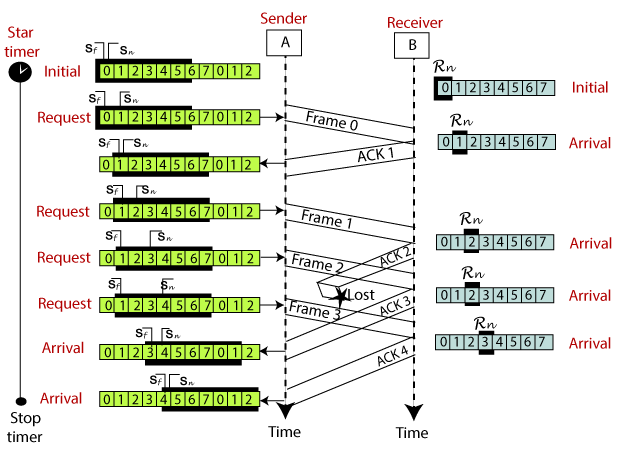
Go – Back – N ARQ provides for sending multiple frames before receiving the acknowledgment for the first frame. The frames are sequentially numbered and a finite number of frames. The maximum number of frames that can be sent depends upon the size of the sending window. If the acknowledgment of a frame is not received within an agreed upon time period, all frames starting from that frame are retransmitted.

The size of the sending window determines the sequence number of the outbound frames. If the sequence number of the frames is an n-bit field, then the range of sequence numbers that can be assigned is 0 to 2n−1. Consequently, the size of the sending window is 2n−1. Thus in order to accommodate a sending window size of 2n−1, a n-bit sequence number is chosen.

The sequence numbers are numbered as modulo-n. For example, if the sending window size is 4, then the sequence numbers will be 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, and so on. The number of bits in the sequence number is 2 to generate the binary sequence 00, 01, 10, 11.

The size of the receiving window is 1.





**Selective Repeat Protocol**

Selective repeat protocol, also called Selective Repeat ARQ (Automatic Repeat request), is a data link layer protocol that uses sliding window method for reliable delivery of data frames. Here, only the erroneous or lost frames are retransmitted, while the good frames are received and buffered.

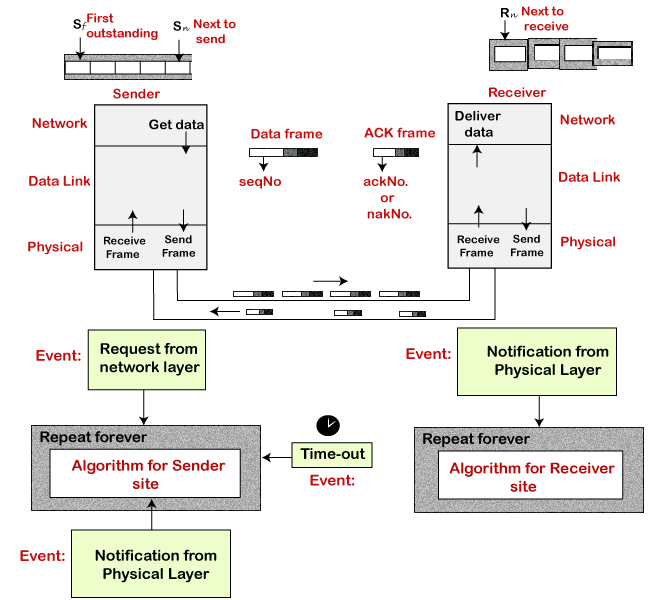
It uses two windows of equal size: a sending window that stores the frames to be sent and a receiving window that stores the frames receive by the receiver. The size is half the maximum sequence number of the frame. For example, if the sequence number is from 0 – 15, the window size will be 8.

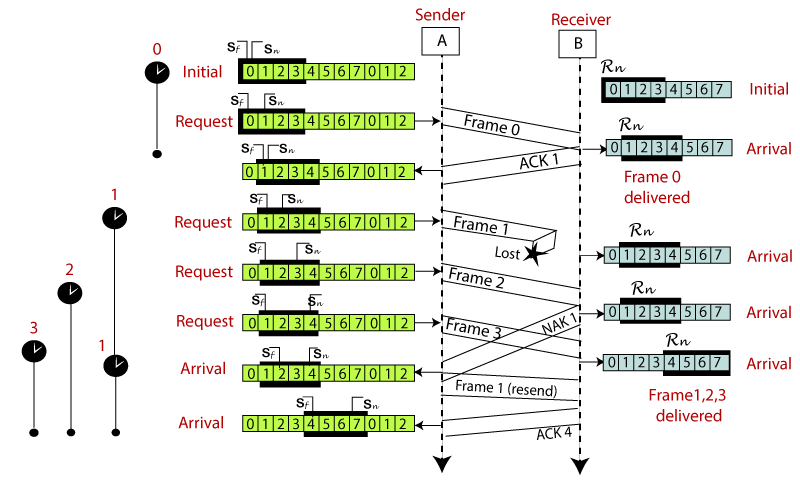
## **Working Principle**

Selective Repeat protocol provides for sending multiple frames depending upon the availability of frames in the sending window, even if it does not receive acknowledgement for any frame in the interim. The maximum number of frames that can be sent depends upon the size of the sending window.

The receiver records the sequence number of the earliest incorrect or un-received frame. It then fills the receiving window with the subsequent frames that it has received. It sends the sequence number of the missing frame along with every acknowledgement frame.

The sender continues to send frames that are in its sending window. Once, it has sent all the frames in the window, it retransmits the frame whose sequence number is given by the acknowledgements. It then continues sending the other frames.





**Difference between Go Back-N and selective repeat protocol**

|  |  |  |
| --- | --- | --- |
|  | Go-Back-N Protocol | Selective Repeat Protocol |
| 1. | In Go-Back-N Protocol, if the sent frame are find suspected then all the frames are re-transmitted from the lost packet to the last packet transmitted. | In selective Repeat protocol, only those frames are re-transmitted which are found suspected. |
| 2. | Sender window size of Go-Back-N Protocol is N. | Sender window size of selective Repeat protocol is also N. |
| 3. | Receiver window size of Go-Back-N Protocol is 1. | Receiver window size of selective Repeat protocol is N. |
| 4. | Go-Back-N Protocol is less complex. | Selective Repeat protocol is more complex. |
| 5. | In Go-Back-N Protocol, neither sender nor at receiver need sorting. | In selective Repeat protocol, receiver side needs sorting to sort the frames. |
| 6. | In Go-Back-N Protocol, type of Acknowledgement is cumulative. | In selective Repeat protocol, type of Acknowledgement is individual. |
| 7. | In Go-Back-N Protocol, Out-of-Order packets are NOT Accepted (discarded) and the entire window is re-transmitted. | In selective Repeat protocol, Out-of-Order packets are Accepted. |
| 8. | In Go-Back-N Protocol, if Receives receives a corrupt packet, then also, the entire window is re-transmitted. | In selective Repeat protocol, if Receives receives a corrupt packet, it immediately sends a negative acknowledgement and hence only the selective packet is retransmitted. |
| 9. | Efficiency of Go-Back-N Protocol is  N/(1+2\*a) | Efficiency of selective Repeat protocol is also  N/(1+2\*a) |

**PROGRAM (C++)- Go Back-N protocol**

#include <iostream>

using namespace std;

bool sender(int frame\_number, int total\_frames, int time\_taken, int max\_time, int size, int \*last\_sent\_frame\_number, int \*total\_transmissions){

if(time\_taken <= max\_time){

cout<<"Acknowledgement Received for Frame number "<<frame\_number<<endl;

if(\*last\_sent\_frame\_number + 1 < total\_frames){

\*last\_sent\_frame\_number += 1;

cout<<"Frame number "<<\*last\_sent\_frame\_number<<" --- Sent"<<endl;

\*total\_transmissions += 1;

}

return true;

}

else{

cout<<"No Acknowledgement Received"<<endl;

return false;

}

}

bool sender1(int frame\_number, int total\_frames, int time\_taken, int max\_time, int size, int \*last\_sent\_frame\_number, int \*total\_transmissions){

cout<<"Information sent from Sender's side\n";

int temp\_frame\_number = frame\_number;

for(int i=0; i<size; i++){

if(temp\_frame\_number < total\_frames){

cout<<"Frame number "<<temp\_frame\_number<<" --- Sent"<<endl;

\*last\_sent\_frame\_number = temp\_frame\_number;

temp\_frame\_number++;

\*total\_transmissions += 1;

}

else{

break;

}

}cout<<endl;

if(time\_taken <= max\_time){

cout<<"Acknowledgement Received for Frame number "<<frame\_number<<endl;

if(\*last\_sent\_frame\_number + 1 < total\_frames){

\*last\_sent\_frame\_number += 1;

cout<<"Frame number "<<\*last\_sent\_frame\_number<<" --- Sent"<<endl;

\*total\_transmissions += 1;

}

return true;

}

else{

cout<<"No Acknowledgement Received "<<endl;

return false;

}

}

int main() {

cout<<"Enter total number of frames ";

int n; cin>>n; cout<<endl;

cout<<"Enter window size ";

int size; cin>>size; cout<<endl;

cout<<"Enter maximum time it will take to receive the Acknowledgement (in sec)";

int max\_time; cin>>max\_time; cout<<endl<<endl;

bool tag=false;

int frame\_number=0;

int last\_sent\_frame\_number = -1;

int total\_transmissions=0;

while(frame\_number < n){

cout<<"Frame number "<<frame\_number<<endl;

cout<<"Enter time it will take to receive the Acknowledgement (in sec) ";

int time\_taken; cin>>time\_taken; cout<<endl<<endl;

if(tag == false){

tag = sender1(frame\_number, n, time\_taken, max\_time, size, &last\_sent\_frame\_number, &total\_transmissions);

}

else{

tag = sender(frame\_number, n, time\_taken, max\_time, size, &last\_sent\_frame\_number, &total\_transmissions);

}

if(tag == true){

frame\_number++;

}

cout<<"=============================================\n\n";

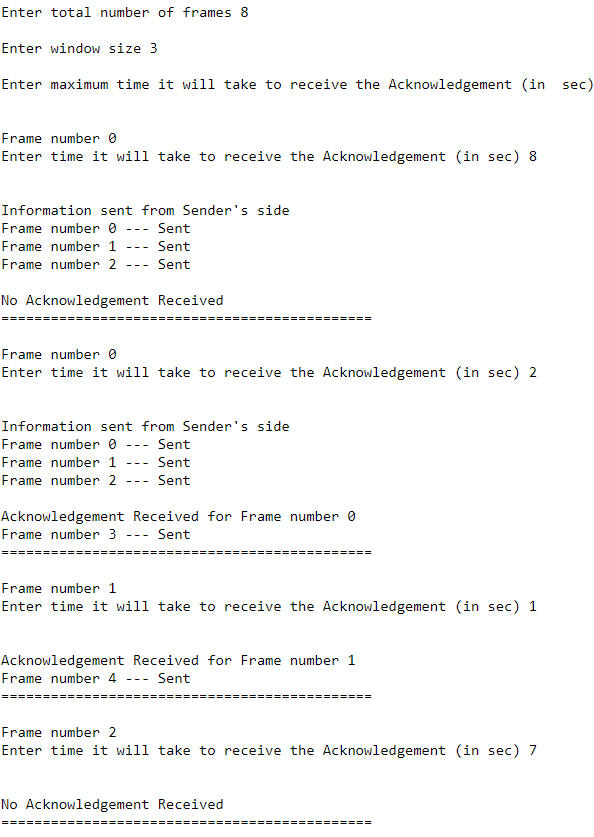
}

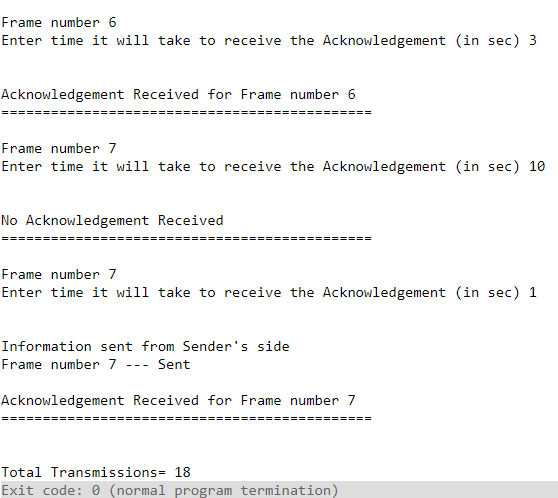
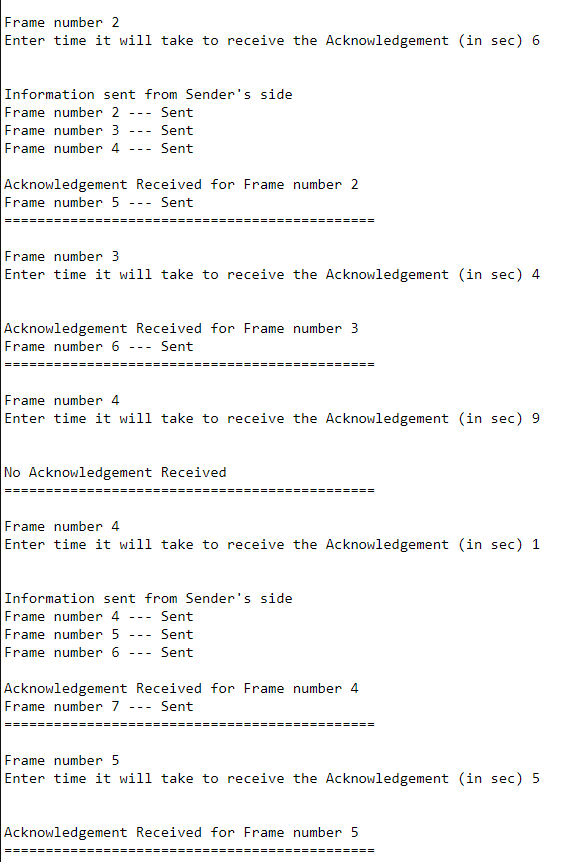
cout<<endl<<"Total Transmissions= "<<total\_transmissions;

return 0;

}

**OUTPUT- Go Back-N protocol**



****

**PROGRAM (C++)- Selective Repeat protocol**

#include <iostream>

using namespace std;

bool sender(int frame\_number, int total\_frames, int time\_taken, int max\_time, int size, int \*last\_sent\_frame\_number, int \*total\_transmissions){

if(time\_taken <= max\_time){

cout<<"Acknowledgement Received for Frame number "<<frame\_number<<endl;

if(\*last\_sent\_frame\_number + 1 < total\_frames){

\*last\_sent\_frame\_number += 1;

cout<<"Frame number "<<\*last\_sent\_frame\_number<<" --- Sent"<<endl;

\*total\_transmissions += 1;

}

return true;

}

else{

cout<<"No Acknowledgement Received"<<endl;

return false;

}

}

bool sender1(int frame\_number, int total\_frames, int time\_taken, int max\_time, int size, int \*last\_sent\_frame\_number, int \*total\_transmissions){

cout<<"Information sent from Sender's side\n";

int temp\_frame\_number = frame\_number;

for(int i=0; i<size; i++){

if(temp\_frame\_number < total\_frames){

cout<<"Frame number "<<temp\_frame\_number<<" --- Sent"<<endl;

\*last\_sent\_frame\_number = temp\_frame\_number;

temp\_frame\_number++;

\*total\_transmissions += 1;

}

else{

break;

}

}cout<<endl;

if(time\_taken <= max\_time){

cout<<"Acknowledgement Received for Frame number "<<frame\_number<<endl;

if(\*last\_sent\_frame\_number + 1 < total\_frames){

\*last\_sent\_frame\_number += 1;

cout<<"Frame number "<<\*last\_sent\_frame\_number<<" --- Sent"<<endl;

\*total\_transmissions += 1;

}

return true;

}

else{

cout<<"No Acknowledgement Received "<<endl;

return false;

}

}

bool sender2(int frame\_number, int total\_frames, int time\_taken, int max\_time, int size, int \*last\_sent\_frame\_number, int \*total\_transmissions){

cout<<"Information sent from Sender's side\n";

int temp\_frame\_number = frame\_number;

cout<<"Frame number "<<temp\_frame\_number<<" --- Sent"<<endl;

\*total\_transmissions += 1;

cout<<endl;

if(time\_taken <= max\_time){

cout<<"Acknowledgement Received for Frame number "<<frame\_number<<endl;

return true;

}

else{

cout<<"No Acknowledgement Received"<<endl;

return false;

}

}

bool sender3(int frame\_number, int total\_frames, int time\_taken, int max\_time, int size, int \*last\_sent\_frame\_number, int \*total\_transmissions){

if(time\_taken <= max\_time){

cout<<"Acknowledgement Received for Frame number "<<frame\_number<<endl;

return true;

}

else{

cout<<"No Acknowledgement Received"<<endl;

return false;

}

}

int main() {

cout<<"Enter total number of frames ";

int n; cin>>n; cout<<endl;

cout<<"Enter window size ";

int size; cin>>size; cout<<endl;

cout<<"Enter maximum time it will take to receive the Acknowledgement (in sec) ";

int max\_time; cin>>max\_time; cout<<endl<<endl;

bool tag=false;

int flag = 0;

int frame\_number=0;

int last\_sent\_frame\_number = -1;

int total\_transmissions=0;

while(frame\_number < n){

cout<<"Frame number "<<frame\_number<<endl;

cout<<"Enter time it will take to receive the Acknowledgement (in sec) ";

int time\_taken; cin>>time\_taken; cout<<endl<<endl;

if(flag == 0){

if(tag == false){

tag = sender1(frame\_number, n, time\_taken, max\_time, size, &last\_sent\_frame\_number, &total\_transmissions);

}

else{

tag = sender(frame\_number, n, time\_taken, max\_time, size, &last\_sent\_frame\_number, &total\_transmissions);

}

}

else{

if(tag == false){

tag = sender2(frame\_number, n, time\_taken, max\_time, size, &last\_sent\_frame\_number, &total\_transmissions);

}

else{

tag = sender3(frame\_number, n, time\_taken, max\_time, size, &last\_sent\_frame\_number, &total\_transmissions);

}

}

if(tag == false || flag == 1){

flag = 1;

}

if(tag == true){

frame\_number++;

if(last\_sent\_frame\_number < frame\_number){

flag = 0;

tag = false;

}

}

cout<<"=============================================\n\n";

}

cout<<endl<<"Total Transmissions= "<<total\_transmissions;

return 0;

}

**OUTPUT- Selective Repeat protocol**

