SLIDING WIN

```
#include <stdio.h>

int main() {

int w, f, i;

int frames[50];

printf("Enter window size: ");

scanf("%d", &w);

printf("Enter number of frames to transmit: ");

scanf("%d", &f);

printf("Enter %d frames: ", f);

for (i = 1; i <= f; i++)

scanf("%d", &frames[i]);

printf("\nWith Sliding Window Protocol, frames are sent as follows:\n");

printf("After sending %d frames, sender waits for acknowledgement.\n\n", w);

for (i = 1; i <= f; i++) {

printf("%d ", frames[i]);

if (i % w == 0) {

printf("\nAcknowledgement received for above frames.\n\n");

}

}

if (f % w != 0) {

printf("\nAcknowledgement received for above frames.\n");

}

return 0;

}
```

GO BACK N

```
#include <stdio.h>

int main() {

int windowSize, sent = 0, ack, i;

printf("Enter window size: ");

scanf("%d", &windowSize);

while (1) {

for (i = 0; i < windowSize; i++) {

printf("Frame %d has been transmitted.\n", sent);

sent++;

}

printf("\nEnter last acknowledgement received: ");

scanf("%d", &ack);

if (ack == sent)

break;

else

sent = ack;

printf("\n--- Sender retransmitting from frame %d ---\n", sent);

}

printf("\nAll frames successfully acknowledged.\n");

return 0;

}
```

CLIENT

```
#include <stdio.h> #include <string.h> #include <sys/socket.h> #include <arpa/inet.h> #include <unistd.h>

int main() { int socket_desc, client_sock, c; struct sockaddr_in server, client; char message[200]; socket_desc = socket(AF_INET, SOCK_STREAM, 0); server.sin_family = AF_INET; server.sin_addr.s_addr = INADDR_ANY; server.sin_port = htons(8080); bind(socket_desc, (struct sockaddr *)&server, sizeof(server)); listen(socket_desc, 3); c = sizeof(struct sockaddr_in); client_sock = accept(socket_desc, (struct sockaddr *)&client, (socklen_t *)&c); strcpy(message, "Hello Client"); write(client_sock, message, strlen(message)); close(client_sock); close(socket_desc); return 0; }
```

}

SERVER

```
#include <stdio.h>

#include <string.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <unistd.h>


int main() {

int sock;

struct sockaddr_in server;

char server_reply[200];

sock = socket(AF_INET,
SOCK_STREAM, 0);

server.sin_addr.s_addr =
inet_addr("127.0.0.1");

server.sin_family = AF_INET;

server.sin_port = htons(8080);

connect(sock, (struct
sockaddr *)&server,
sizeof(server));

recv(sock, server_reply,
sizeof(server_reply), 0);

printf("Server
Response: %s\n",
server_reply);

close(sock);

return 0;
```