Previously

- Theoretical basis of Communication
- Fourier Analysis
- Maximum Data Rate

Elementary Protocols

- Assumptions & Declarations
- Unrestricted Simplex Protocol
- Simplex Stop and Wait protocol
- Simplex Protocol for a noisy channel

Assumptions

- Physical, Data link & network are all _____
  _____
- Service being provided is _____
- Data is always available _____
- Machines do not crash _____
- Treat all data as pure data although _____
- Assume the existence of the Physical layer including
  - to_physical_layer _____
  - from_physical_layer _____
  - Computation of the checksum

More assumptions

- wait_for_event _____
  - Normally this would be done with _____
- Frame arrival will either cause
  - event = frame_arrival _____
  - event = cksum_err _____
  - If everything is OK _____

Data Types


```
#define MAX_PKT 1024                        /* determines packet size in bytes */

typedef enum {false, true} boolean;         /* boolean type */
typedef unsigned int seq_nr;                /* sequence or ack numbers */
typedef struct {unsigned char data[MAX_PKT];} packet;/*   packet definition */
typedef enum {data, ack, nak} frame_kind;   /* frame_kind definition */

typedef struct {                            /* frames are transported in this layer */
  frame_kind kind;                          /* what kind of a frame is it? */
  seq_nr seq;                               /* sequence number */
  seq_nr ack;                               /* acknowledgement number */
  packet info;                              /* the network layer packet */
} frame;
```


Function definitions


```
/* Wait for an event to happen; return its type in event. */
void wait_for_event(event_type *event);

/* Fetch a packet from the network layer for transmission on the channel. */
void from_network_layer(packet *p);

/* Deliver information from an inbound frame to the network layer. */
void to_network_layer(packet *p);

/* Go get an inbound frame from the physical layer and copy it to r. */
void from_physical_layer(frame *r);

/* Pass the frame to the physical layer for transmission. */
void to_physical_layer(frame *s);

/* Start the clock running and enable the timeout event. */
void start_timer(seq_nr k);

/* Stop the clock and disable the timeout event. */
void stop_timer(seq_nr k);

/* Start an auxiliary timer and enable the ack_timeout event. */
void start_ack_timer(void);

/* Stop the auxiliary timer and disable the ack_timeout event. */
void stop_ack_timer(void);

/* Allow the network layer to cause a network_layer_ready event. */
void enable_network_layer(void);

/* Forbid the network layer from causing a network_layer_ready event. */
void disable_network_layer(void);

/* Macro inc is expanded in-line: Increment k circularly. */
#define inc(k) if (k < MAX_SEQ) k = k + 1; else k = 0
```

Unrestricted Simplex Protocol

- Data transmission _____
- Communication channel _____
- Processing time _____
- Buffer Space _____

Sender

```
void sender1(void){
  frame s;
  packet buffer;
  while (true) {
     from_network_layer(&buffer);
     s.info = buffer;
     to_physical_layer(&s);
  }
}
```

Receiver


```
void receiver1(void){
  frame r;
  event_type event;
  while (true) {
     wait_for_event(&event);
     from_physical_layer(&r);
     to_network_layer(&r.info);
  }
}
```


Stop–and–Wait Protocol

- Data transmission _____
- Communication channel _____
- Processing time _____
- Buffer Space _____

- The receiver can be flooded by the sender. To prevent this we can…
  - Insert delays _____
  - Give feedback to sender _____
  - Stop-and-Wait _____
    - Half Duplex _____

Sender


```
void sender2(void){
  frame s;
  packet buffer;
  event_type event;
  while (true) {
        from_network_layer(&buffer);
        s.info = buffer;
        to_physical_layer(&s);
        wait_for_event(&event);
      }
}
```


Receiver


```
void receiver2(void){
  frame r, s;
  event_type event;
  while (true) {
        wait_for_event(&event);
        from_physical_layer(&r);
        to_network_layer(&r.info);
        to_physical_layer(&s);
  }
}
```

Results

Simulating Protocol 2
Events 100000
Timoute 30
pkt_loss 0
pkt_cksum 0

	Process 0	Process 1
Total data frames sent	12580	0
Data frames lost	0	0
Data frames not lost	12580	0
Frames retransmitted	0	0
Good ack frames rec'd	12579	0
Bad ack frames rec'd	0	0
Good data frames rec'd	0	12580
Bad data frames rec'd	0	0
Payloads accepted	0	12580
Total ack frames sent	0	12580
Ack frames lost	0	0
Ack frames not lost	0	12580
Timeouts	0	0
Ack timeouts	0	0

Protocol for a Noisy Channel

- Data transmission _____
- Communication channel _____
  - Corrupt Frames _____
  - Missing Frames _____
- Solution
  - Use a timer: _____
  - Problem: _____
  - Use Sequence Numbers: _____
  - PAR/ARQ: _____

Sender


```
void sender3(void){
  frame s;                    packet buffer;
  event_type event;           seq_nr next_frame_to_send = 0;
  from_network_layer(&buffer);
  while (true) {
      s.info = buffer;        s.seq = next_frame_to_send;
      to_physical_layer(&s);
      start_timer(s.seq);
      wait_for_event(&event);
      if (event == frame_arrival) {
          from_physical_layer(&s);
          if (s.ack == next_frame_to_send) {
              from_network_layer(&buffer);
              inc(next_frame_to_send);
          }
      }
```


Receiver


```
void receiver3(void){
  frame r, s;             event_type event;
  seq_nr frame_expected = 0;
  while (true) {
      wait_for_event(&event);
      if (event == frame_arrival) {
          from_physical_layer(&r);
          if (r.seq == frame_expected) {
              to_network_layer(&r.info);
              inc(frame_expected);
          }
          s.ack = 1 - frame_expected;
          to_physical_layer(&s);
      }
  }
}
```

Results

Simulating Protocol 3

Events	100000
Timeout	30
pct_loss	20
pct_cksum	15

	Process 0	Process 1
Total data frames sent	5994	0
Data frames lost	1231	0
Data frames not lost	4763	0
Frames retransmitted	2806	0
Good ack frames rec'd	2694	0
Bad ack frames rec'd	493	0
Good data frames rec'd	0	4037
Bad data frames rec'd	0	726
Payloads accepted	0	2686
Total ack frames sent	0	4037
Ack frames lost	0	850
Ack frames not lost	0	3187
Timeouts	2806	0
Ack timeouts	0	0

Efficiency (payloads accepted/data pkts sent) = 44%
End of simulation. Time=100000