Outline

The Communications Processor

1. Control Structures
   - 1.1. Program Modules
   - 1.2. Priority Structure
2. Data Structures
   - 2.1. Data Buffering
   - 2.2. Control Information
3. Performance
   - 3.1. Processor Issues
   - 3.2. Memory Issues
   - 3.3. I/O Issues
4. Example - IMP/TIP based on Honeywell 316
5. New Development - PLURIBUS
   multi-resource system with high performance and high reliability, based on a modular architecture

1.1. Program Modules

1. General Concepts

Two basic Kinds of control -
- Event-driven = "Interrupt" level
- Periodic tasks = clock interrupt, background loop

Two basic Kinds of events -
- Hardware - usually I/O - event
- Software event, e.g. resource requested or returned

Key ideas:
- Responsiveness to I/O events dictates interrupt structure
- Modules should interface at points where context is simple to save + restore
- Some modules must run at a guaranteed frequency
- Modules must not deadlock with each other, or get "stuck" or "hung", or loop indefinitely

2. Store and Forward

Modem Input ; TASK-S/F ; Modem Output
Timeout
Note: responsiveness to I/O ; bandwidth limits; straight line code

3 Source to Destination

Host Input ; TASK-data messages ; Host Output - data;
Control message generation ; TASK - control messages;
Host Output - control messages
Timeout
Note: bandwidth limits; code is serially reentrant and coroutined with clock and external events.

4. Terminal Control

Terminal In ; Data → Net ; Control → Net. Terminal Out; Data ← Net;
Control ← Net **Note:** control structure similar to 3; tighter limits

1.2 Priority Structure

1. General Concepts

Synchronous I/O needs high priority for high responsiveness
Modules must be able to invoke other modules,
both at higher and at lower levels
* Graceful degradation under load:
least important module stops running first

2. Example – ARPA Network IMP

Modem In
Modem Out
Host Out
Host In
Timeout
TASK
Background

2.1 Data Buffering

1. General Concepts
   Buffering related to performance + to lockup prevention
   Minimize the copying of information
   Maximize resource sharing
   Maximize the use of storage

2. Packet Buffers
   Fixed vs Dynamic Size Buffers
   Use for packets on lines and messages for Hosts
   Number of buffers – Performance
      I/O: Modems, Hosts
      Buffering: Source, Destination
   Shared / Dedicated / Overlapping Assignment
   Allocation Strategy
   Fairness

3. Device Buffers
   Dedicated vs Dynamic
   Use for characters for terminals, etc.
   Single / Double / Multiple – Input & Output
   Number of buffers – Performance
   Size of Buffers
   Allocation Strategy
   Fairness

4. Two Levels vs. One Level

   Should Packet Buffers ≡ Device Buffers?

2.2. Control Information

1. General Concepts

Two basic kinds of information:
- I/O control & status
- Algorithmic or procedural control

Two basic organization methods
- Parallel: table = all devices, one entry
- Serial: table = all entries, one device

Must have information on all:
- **Pending transmissions**
- **Previous transmissions (*)**
- **Pending replies**
- **Previous replies (*)**

(*) up to some age

Basic tradeoff between fixed and dynamic control tables - offer different efficiencies

2. Store and Forward

I/O: line up/down, queues, etc
Algorithm: ack state tables, routing tables

3. Source / Destination

I/O: Host state, queues, timers,
Algorithm:
- transaction block - source message
- reassembly block - destination mess.
- message control block - s/d conversation

4. Terminal Control

I/O: device ID, speed, etc; info on handling terminal data + commands
Algorithm: Host-Host protocol status, allocation status

3.1. Processor Issues

1. General Concepts

Depends on both processor speed and algorithm type
Two kinds of processing:
- length-dependent (e.g., per bit)
- length-independent (e.g., per packet)

and one or the other or both may be significant
There are many levels of length-independent processing overhead, so that one must consider
- # packets / message
- # messages / conversation
- # active conversations at a time, etc

in addition to # bits / packet

Two kinds of processor limits:
bandwidth (in bits/sec) based on processing speed
responsiveness - how many devices at what rates can be serviced

2. Store and Forward

Length-independent (per packet) dominates bandwidth.
Ack system critical - piggybacking important
Responsiveness important
* Packet size important

3. Source / Destination

Protocol somewhat important, but message size is main factor; per message processing significant

4. Terminal Control

Length-dependent code critical
Responsiveness important
May be 10-50 times processing/bit for S+F
Protocol issues make a big difference, eg.,
- echoing strategy, allocation strategy,
- transmission strategy, etc.

3.2. Memory Issues

1. General Concepts

Need to buffer data until receipt of appropriate delivery acknowledgement at "other end"
Need to buffer data on input
Need to buffer data for sequencing, reassembly, etc.
* Lack of sufficient storage ⇒ performance degrades

2. Store and Forward

Input: double buffer, to allow examination of acks and routing
Output: Buffering = bit rate * ack delay

3. Source / Destination

Buffer at source / destination / both
Buffering = bit rate * RFNM delay

4. Device Buffer

Input: double buffer to allow input of data + transmission of data
Output: double (multiple?) buffer to allow output of data and allocation of next buffer
Buffering: bit rate * ALLOCATE delay

3.3 I/O Issues

1. General Concepts

Speed of I/O hardware is most important
length-dependent factor
Buffering in I/O interfaces determines
the processor responsiveness criteria
Functional role of the I/O hardware is
flexible with that of the software
bandwidth, responsiveness, cost determine it.
How much intelligence to put in I/O interfaces
is another tradeoff
Interrupts vs Polling
Memory Channel vs Word-at-a-time

2. Store and Forward

line protocol + CRC in hardware
memory channel needed for bandwidth
some buffering needed for responsiveness
I/O must be fast

3. Source / Destination - Host:

line protocol in hardware
other issues similar to 2.

4. Terminal Control

hardware / software boundary flexible,
but critical for bandwidth + cost.
Cost per terminal should be low.
Processing per bit should be low.

4. Example - ARPA Net IMP / TIP

Honeywell 516 / 316

Designed ca. 1965
1.6 μs cycle
16-bit word
32K memory address
Simple instruction set: PC, AC, XR registers

16 interrupt levels
16 memory channels 3.2 μs/word
$\Rightarrow$ # Lines + # Hosts ≤ 7

Modem Interface

Finite State Machine + CRC / Mem Channel / Int.

Host Interface

Bit Serial, Handshake / Mem Channel / Interrupt

Clock
100 μs reading
25 ms interrupt

Task interrupt - program settable.

Cost: $50k
Performance: 600 Kbs Store-and-Forward
Reliability: 99.5%

Terminal Interface
Multi-Line Controller 64 devices
Asynchronous devices at several speeds
Memory Channel + Interrupt : MLC = concentrator