CSMC 417

Computer Networks

Prof. Ashok K Agrawala

© 2003 Ashok Agrawala

Set 2A

Spring 2003

An Information-centric View of Communications

What information is needed where and why?

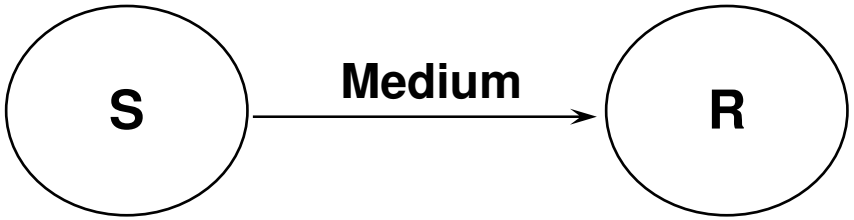
Who Communicates

- **Two autonomous entities**
  - **Capable of independent operations**
    - » **Timing**
    - » **What if not independent**
      - **Master-Slave operation**
      - **I/O Devices**
- **What do they communicate ?**
  - **Information**
  - **Convert to message – Has meaning only to the two entities**

Message

- Encoding and decoding known/agreed upon by the sender/receiver
- Representable in a form so that it can be manipulated by processors
  - Stored
  - Moved
  - (Not Necessarily Interpreted)

Communications Between Two Entities



- S and R need to communicate
- Sender S needs to send information I to receiver R and be assured that R has received it
- All assumptions have to be true by design or due to the physical properties

Entities

- **Sender**
  - Capable of autonomous action
  - Has access to information I
    - » How long does it take for it to access I ?
  - Has ability to send and receive signals to/from the medium
- **Receiver**
  - Capable of autonomous action
  - Has ability to send and receive signals to/from the medium
- **Medium**
  - Can move signals from S to R and R to S
    - » What additional properties must M have?

Who to Communicate with

- How does the sender know to communicate with the receiver?
  - Based on additional information
  - Hard Coded
  - Search engines
    - » Has to know the name/address of search engines
- What does sender know about the Receiver?
  - Name
  - Address
  - Some other properties

Expectations

- **Does the Sender expect the Receiver to take some specific action on receiving the message?**
  - **Based on additional information**
  - **How does it know that the action was taken by the Receiver?**
    - » **Confirmation**
    - » **Response**
    - » **…**
  - **How does it know that the receiver received the message intact?**

Receiver

- How does it know that Sender is sending a message?
- When should it listen?
  - Always listening
  - At some agreed upon times
  - Some other mechanism
    - » Signal
      - How does it know to listen for the signal
  - …

Message Movement

- **Message Representation**
  - **Bit String**
- **Movement**
  - **Medium**
  - **IPC**
    - » **On the same computer**
    - » **OS Mechanisms**
    - » **Shared memory**
  - **On different computers**
    - » **Medium permits movement of message encoded as bit string from one machine to the other**

Medium

- **Capabilities**
  - **Can move messages**
  - **What can we assume about the capabilities**
    - » **Functionality – storage, order, processing**
    - » **Performance**
- **Topology**
  - **Point to point – one sender/one receiver**
  - **Broadcast – one sender/multiple possible receivers**
- **Control**
  - **Active**
  - **Passive**

Medium

- Lowest level
  - Dedicated
    - » Point-to-point
  - Shared
    - » Multicast/Broadcast
    - » Control Mechanisms
      - Who
      - When
- Higher level
  - Additional capabilities
    - » Buffering
    - » Error handling
    - » Format Conversion
    - » Segmentation/ Packetizing
    - » ...

Timing

- **Passive Medium**
  - –There must be agreement about timing
    - »Predefined
    - »R Listening all the time
      - •Interrupt is a form of listening all the time
      - •Polling can be in hardware or in software
  - –Sender and receiver must synchronize to interpret the signals properly
  - –Buffers act as universal sender/receiver
- **Similar issues arise at higher levels**

Information Coding

- S and R must know the common coding for the information
- Interpretation of the information must be consistent

Naming

- S must know the identity of R
  - Explicitly
  - Implicitly
- ID must contain enough information for the medium to locate R
- Name / Address mapping issues
- Routing issues
- Is mapping static or dynamic
  - Mobile computing

Confirmation

- How does S know that R has received the information correctly?
  - True by design
  - Handshake
  - Explicit acknowledgement
- Error Detection and Handling
  - Who detects
  - What steps are taken
    - » Ignore error
    - » Inform sender
    - » ..

Building Block

- **Simple Medium**
- **Media as agent**
  - **Active**
    - » **Capable of acting as**
      - **sender**
      - **receiver**
      - **processor**
  - **State information**
  - **Storage Capabilities**

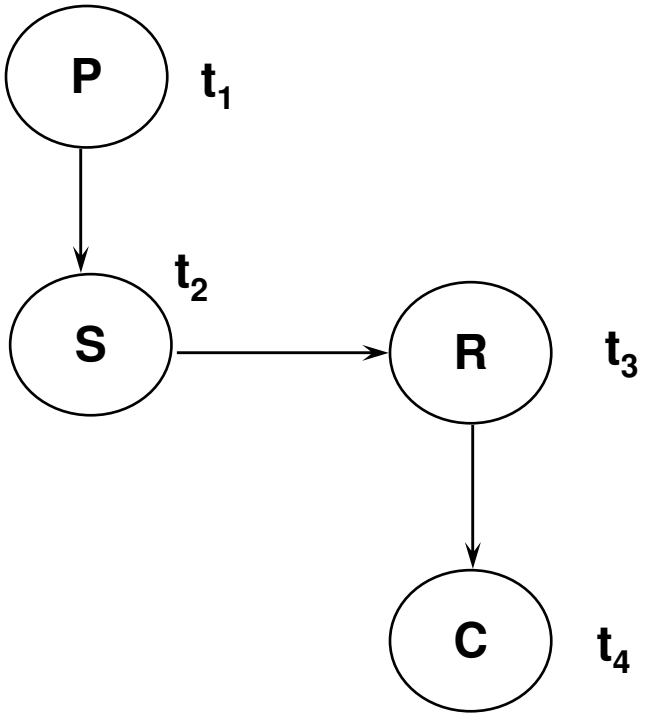
Organizing a network

- Hierarchical structure
  - Levels of abstraction
- Defined in terms of units of information
  - bit
  - byte
  - frame
  - packet
  - message
  - cell
- At one level
  - Sender
  - Receiver
  - Medium

Transfer Cycle

- Move one unit of information from S to R and sustain a delay
- Unacknowledged
- Sequencing of information
  - Explicit - sequence numbering
  - Implicit - Medium guarantees in order delivery

Producer Consumer Relationship



- No Storage => $t_2 = t_3$

  Cycle time = Max(t_1,t_4) + t_2

- Single Buffer

  Cycle Time = Max (t_1+t_2, t_2+t_3, t_3+t_4)

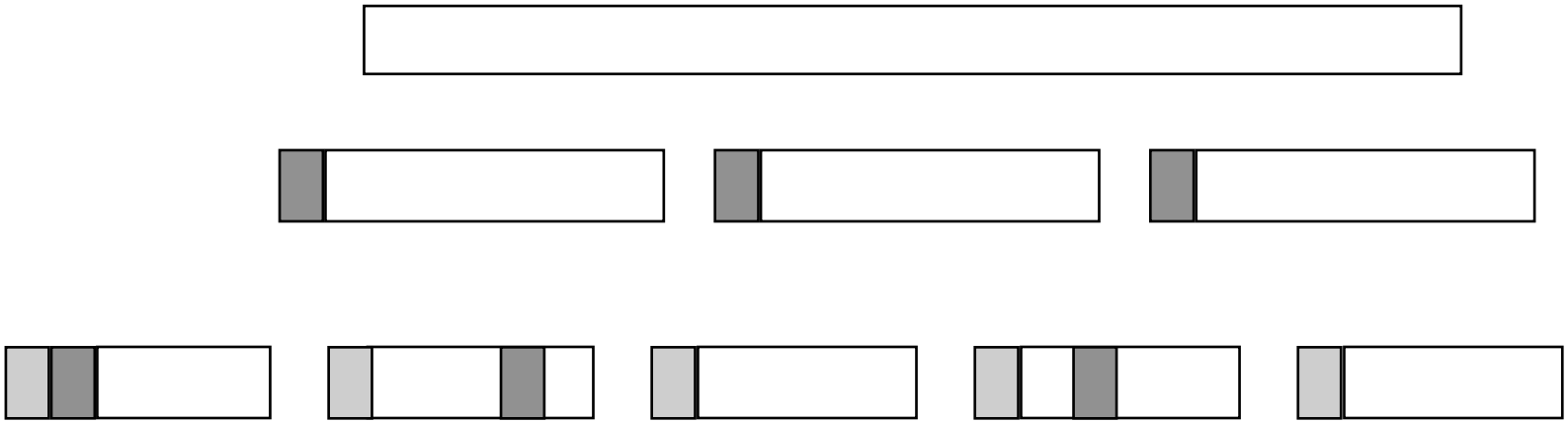
- Multiple Buffers ?

Resource Implications

- All actions require the use of resources
- Have to consider them in Resource/Time space
- Independent resources may be capable of autonomous actions
- Interdependencies/ sequencing of actions has to be explicitly identified

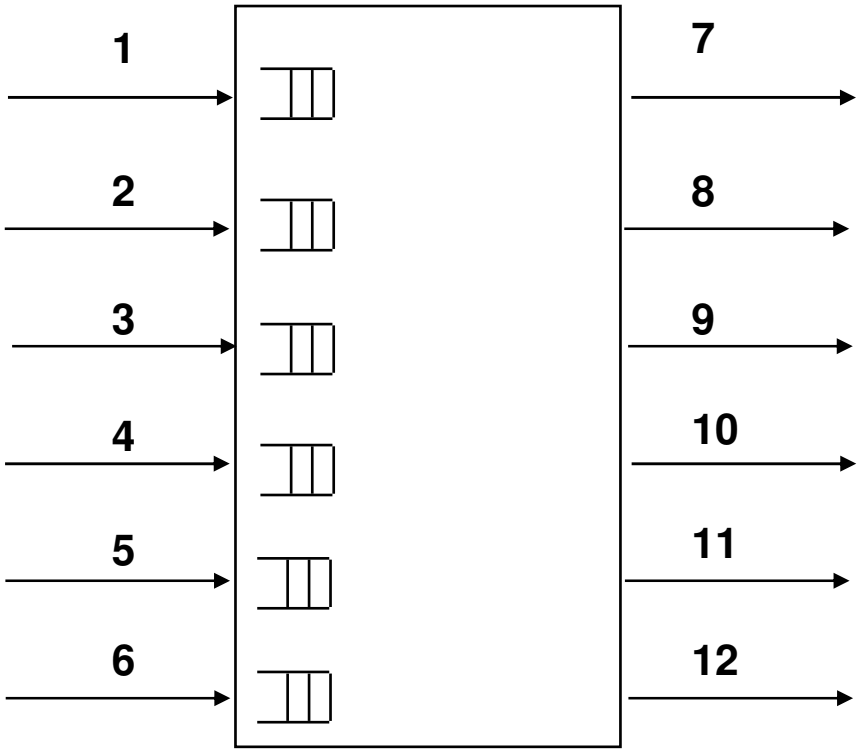
Basics of Packet Switching

Multiple Protocol Layers



Headers/trailers are required at each level

Model of a node



Processing Required

Move the incoming packet to an outgoing link

- **Packet must have enough information to permit determination of which link?**

Steps in Processing a Packet

1. **Packet is received in the buffer of the line receiver**
   - Checked for errors
   - New buffer assigned to the line receiver
2. **Header is examined**
   - What to do if error
3. **Outgoing link is determined**
   - What to do if error
4. **Packet is moved to the outgoing link buffer**
   - What to do if buffers are full?
5. **Packet is sent**

Detailed model is lot more complex

Resources Available

- **Receiver**
  - Has buffers
  - Can function autonomously receiving a packet once buffer is available
- **Processor**
  - Carries out processing and error handling
  - Buffer management
  - Priority management
- **Sender**
  - Has buffer
  - Can function autonomously sending a packet once the packet is in the buffer

Step 1 - Receiving the Packet

- t_1 -Time to receive the packet
  - Line speed R
  - Packet size N
- $t_1 = N/R$
- Example
  - R = 1 Mb/sec
  - N = 1024 Bytes
  - t_1 = 1024 *8/1 = 8 ms
- Example
  - R = 1 Gb/s
  - N = 50 Bytes
  - t_1 = 50*8/1 = 400 ns

Resource Used - Receiver

Steps 2,3,4 - Processing

- t_2 - Depends on many factors
  - Processor speed
  - Complexity of steps
  - Error conditions and handling
  - ...
- Resource Involved - Processor
  - One processor may be handling all the lines
- There may be queuing delays

Step 5 - Sending

- t_3 -Time to send the packet
  - Line speed R
  - Packet size N
- $t_3 = N/R$
- Example
  - R = 1 Mb/sec
  - N = 1024 Bytes
  - t_3 = 1024 *8/1 = 8 ms
- Example
  - R = 1 Gb/s
  - N = 50 Bytes
  - t_3 = 50*8/1 = 400 ns

Resource used - Sender

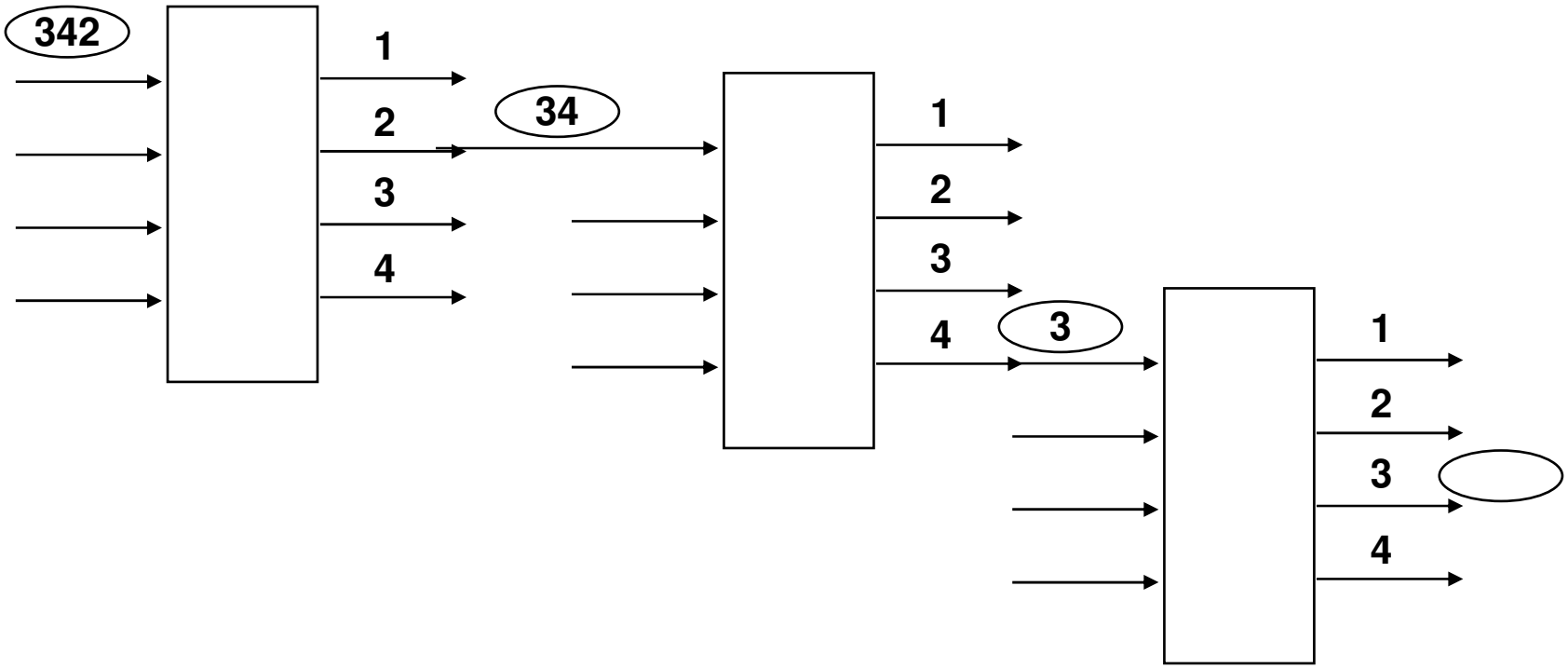
Timing Issues

- **Store and Forward**
  - **Can not overlap steps**
  - **Time to process**
    - » $t_1 + t_2 + t_3$
  - **For a series of packets**
    - » **May overlap the times when different resources are used**
- **Cut Through Forwarding**
  - **Start forwarding the packet as soon as the header has arrived and examined to determine outgoing link**
  - **What is the link is busy ?**
  - **Have to implement store and forward also**

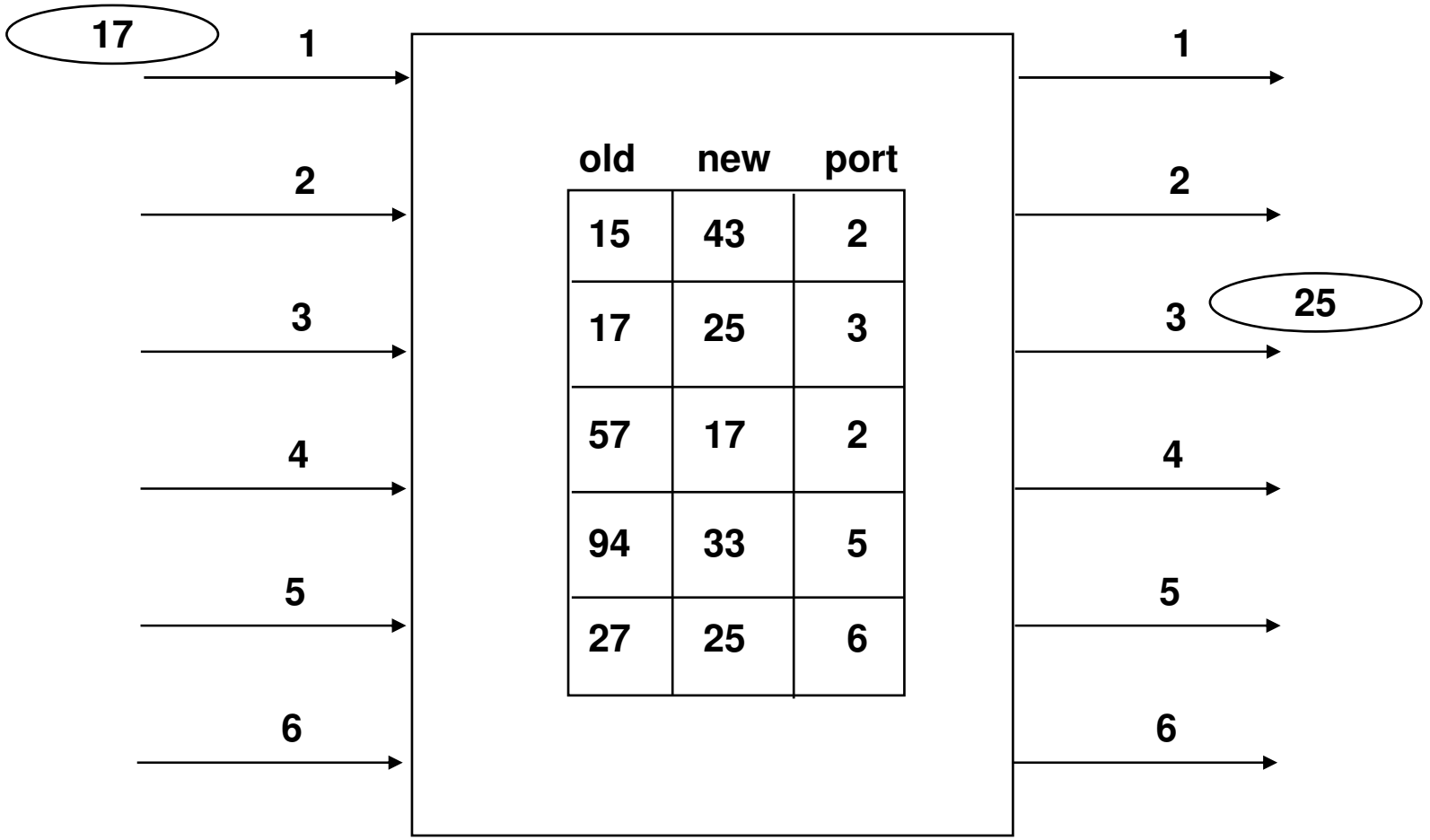
Routing

- **Source Routing**
  - The header contains the informaiton about each outgoing link
  - For long routes header may become large
- **Hop Routing**
  - Each node maintains a table
  - Header contains Old ID, New ID and Link(Port)
    - » Look up old ID in the table
    - » Replace with New ID in the packet
    - » Put it on the Link
- **Hierarchical**
  - Have a few levels

Source Routing



Hop Routing


17
1
2
3
4
5
6
old
new
port
15
43
2
17
25
3
57
17
2
94
33
5
27
25
6
1
2
3
25
4
5
6

Error Recovery

- Should error recovery be done at the lowest level?
- Sequence Numbering ?
- When cell lost what to do ?
  - Loose message
- Does Forward Error Correction help ?
- What are the tradeoffs ?