

Network Reading Group

End-To-End Arguments in System Design

by J. H. Saltzer, D.P. Reed and D.D. Clark

Lecture: Kameswari Chebrolu

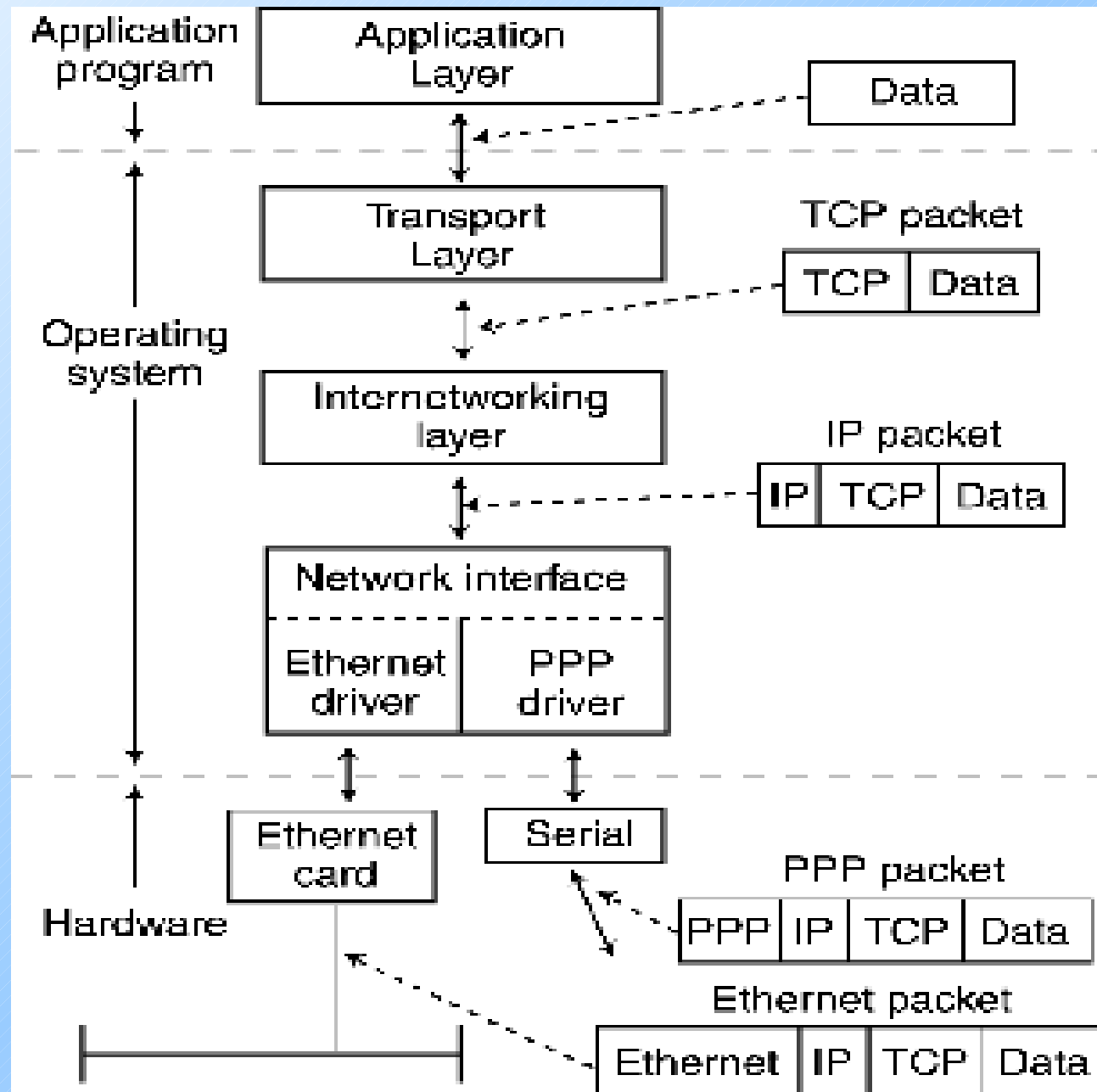
Thursday, 15 March 2005

<http://home.iitk.ac.in/~chebrolu/net-read.html>

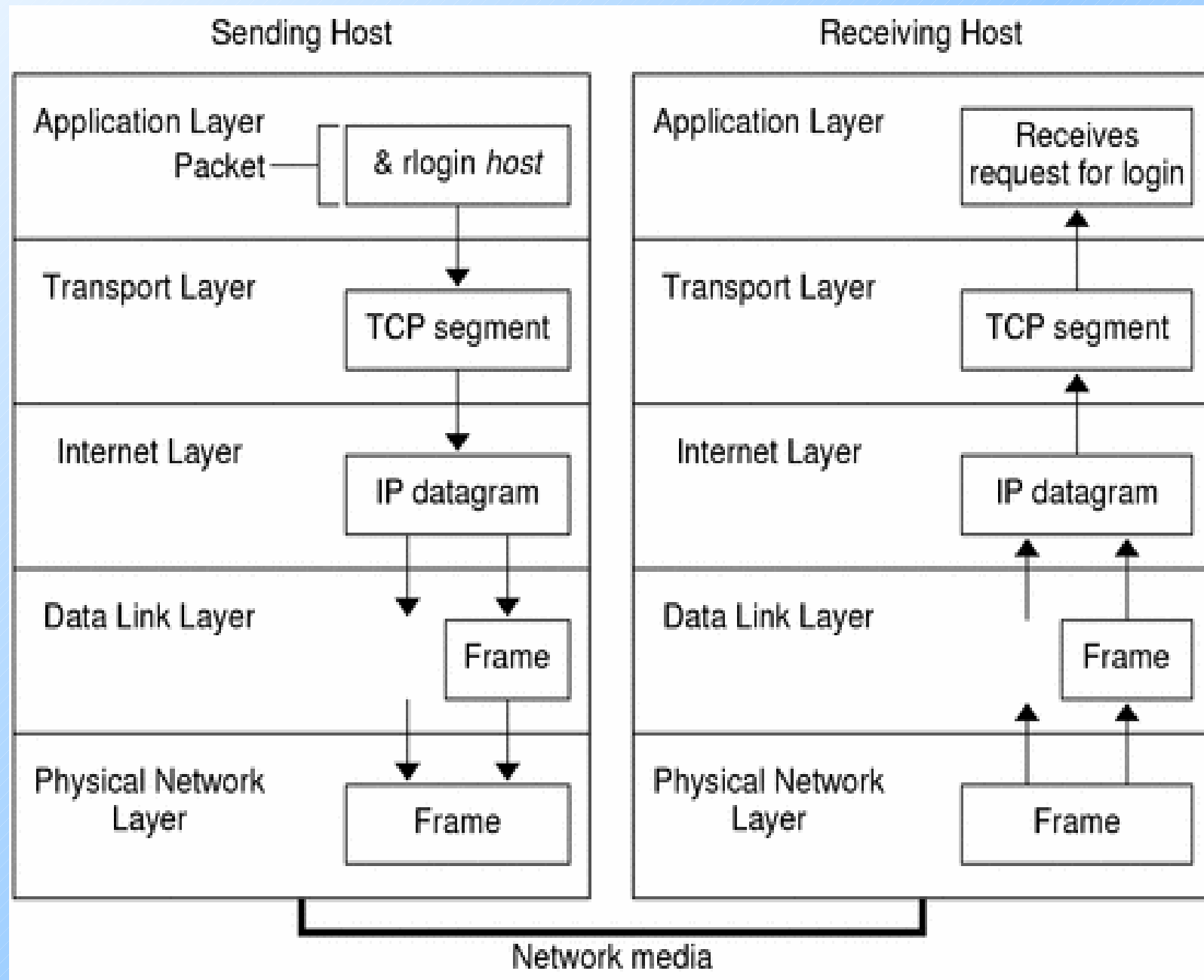
Recap

- To interconnect systems from different manufactures, one needs standard rules
- Underlying Concept: Layering
- OSI Protocol Stack: 7 layers
- Information exchange via “Encapsulation”

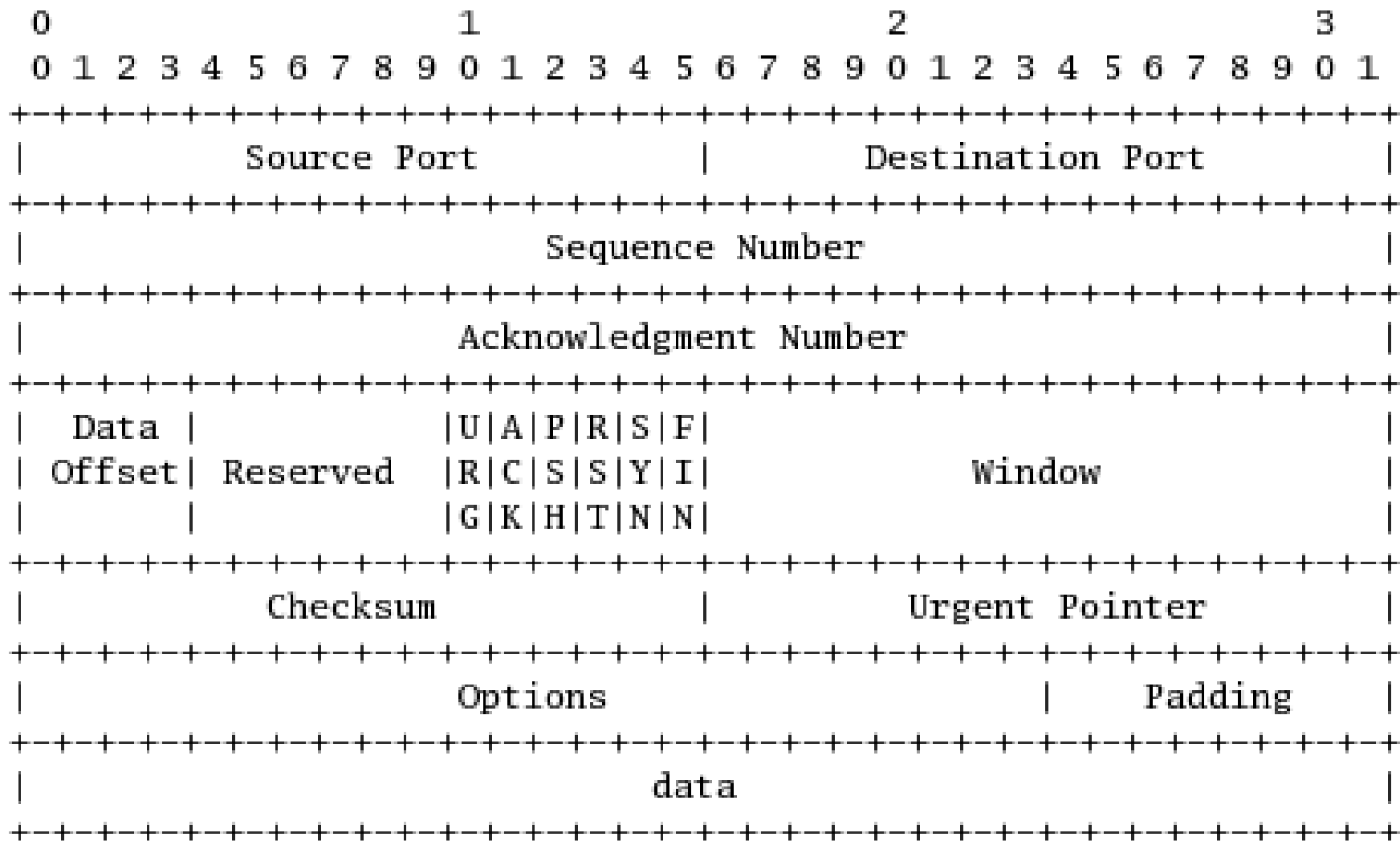
Packet Travel down the stack



Packet Travel down the stack



TCP Header



TCP Header Format

IP Header

0	4	8	16	19	31
Version	IHL	Type of Service	Total Length		
Identification			Flags	Fragment Offset	
Time To Live	Protocol		Header Checksum		
Source IP Address					
Destination IP Address					
Options					Padding

Objective

- Presents a design principle - “End-to-End Argument”
- Principle guides placement of functions among modules of a distributed system
- *“The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system.”*

Example: File Transfer

- 1) Host A: Read file from disk and pass it to FTP in fixed size blocks
- 2) Host A: FTP asks data communication system to transmit the file
- 3) Data communication system moves file from Host A to Host B
- 4) Host B: Communication system hands data to the FTP program at B
- 5) Host B: FTP asks file system to write the received data on the disk at B

Threats to Transaction

- 1) Hardware faults in disk storage systems
- 2) FTP or the communication system may make mistakes buffering/copying data
- 3) Processor/Memory might have a transient error while buffering/copying
- 4) Communication system might drop, change bits in a packets or duplicate it.
- 5) The hosts may crash halfway through the transaction

Solutions

- Introduce redundancy
 - duplicate copies, timeouts and retry
 - if threats are low probability, this approach is uneconomical
- End-to-end check and retry
 - If failures rare, this works well

Conclusions

- For data systems to provide extraordinary reliability does not reduce the burden on the application program to ensure reliability
- Tradeoffs exist between performance and reliability
- Lower levels common to many applications, which may not need the functions implemented there.
- Lower levels may not have as much information as higher levels

Other Examples

- Secure transmission of data
- Duplicate message suppression
- Datagrams

Next Meeting

Congestion Avoidance and Control

V. Jacobson and M.J. Karels

March 22th, 2005

(Tue: 5.30pm- 6.30pm)