Lecture 10

Communication Fundamentals in Computer Networks

M. Adnan Quaium

Assistant Professor
Department of Electrical and Electronic Engineering
Ahsanullah University of Science and Technology
Room – 4A07
Email – adnan.eee@aust.edu
URL- http://adnan.quaium.com/aust/cse4295

Communication Protocols

- The hardware of a network is made up of components whose tasks consists of transmitting bits from one computer to another.
- If one wanted to organize computer communication at this level alone, it would be comparable to programming a computer in a rudimentary machine language, i.e., with just zeros and ones.
- Doing things this way, it would be impossible to manage the effort and complexity of the tasks involved.
- Just as in computer programming, complex software systems have been created for the control and use of computer networks.

Protocol Families

It is necessary in communication that all communicating parties agree on a fixed set of rules for message exchange.

- This applies both to the language used as well as to the rules of conduct which enable efficient communication to even take place at all.
- These rules are summed up in technical language with the term *communication protocol* or simply protocol.

- In the case of communication in computer networks, the software that implements the network protocol at a computer is called protocol software.

Protocol Families

- The various sub-problems are handled by special protocols. Working together they must all mesh seamlessly.
- To ensure this interplay, the development of the network protocol software is viewed as a comprehensive task, the solution provided by the *protocol suite* or *protocol family*.

All of the individual protocols interact efficiently with one another and in this way solve the overall problem of network communication.

Source of Errors

When many computers communicate with each other over a shared communication network, numerous problems can arise. These all have to be handled by the network protocol software.

Just a few examples of these problems are:

- Hardware Failure
- Network Congestion
- Packet delay and Packet loss
- Data Corruption
- Duplicated Data packets and mixed up sequence

Layer Models

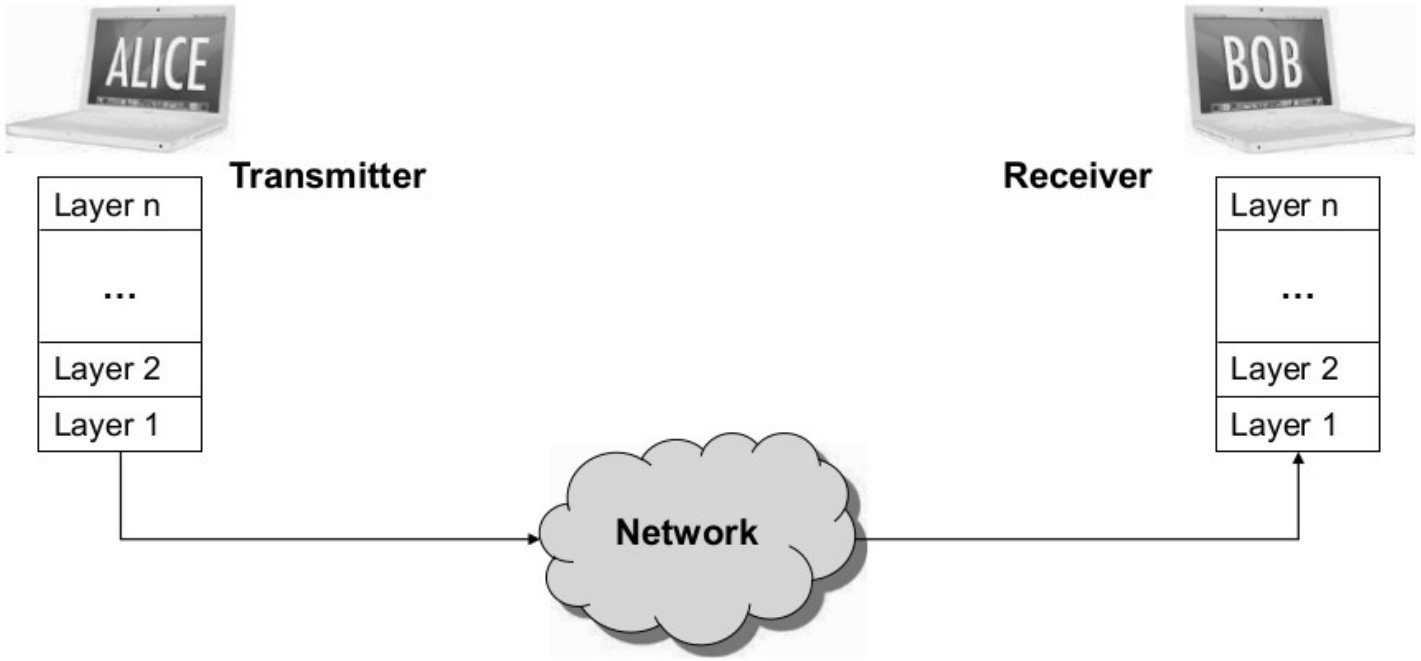
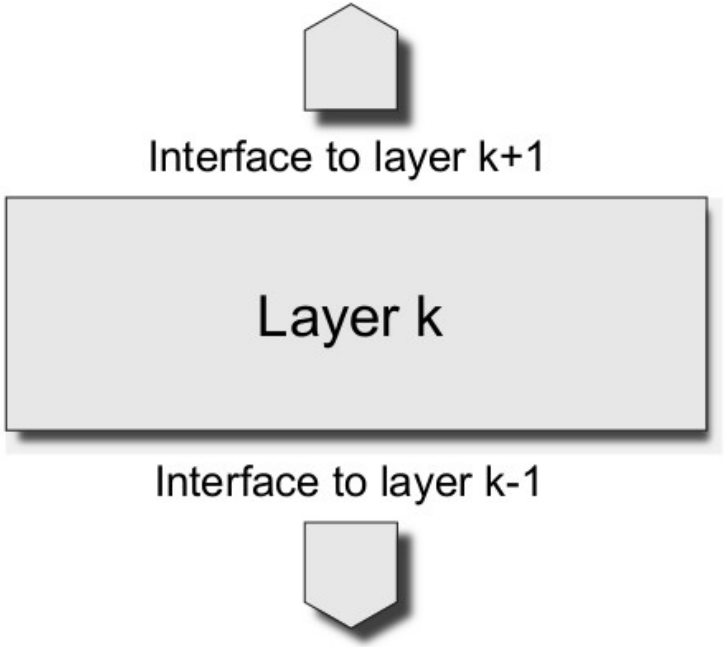


Fig. 3.16 Data transfer via a protocol stack.

- To support protocol designers in their work, tools and models have been developed that minutely break down the entire process of network communication and arrange it hierarchically.
- This enables the largely independent development and improvement of each of the network protocols settled on the layers and makes them as simple as possible.
- The most well-known of these models is the protocol stack (layering model).

Layer Models



- Here the entire network communication process is broken down into individual layers organized one on top of the other.
- Every layer addresses a specific sub-problem of the network communication and with each layer a new level of abstraction is added to the communication.
- Ideally, the designer constructs a protocol family – the so-called protocol stack – from these layers where each protocol addresses the task presented at a specific layer.

Layer Models

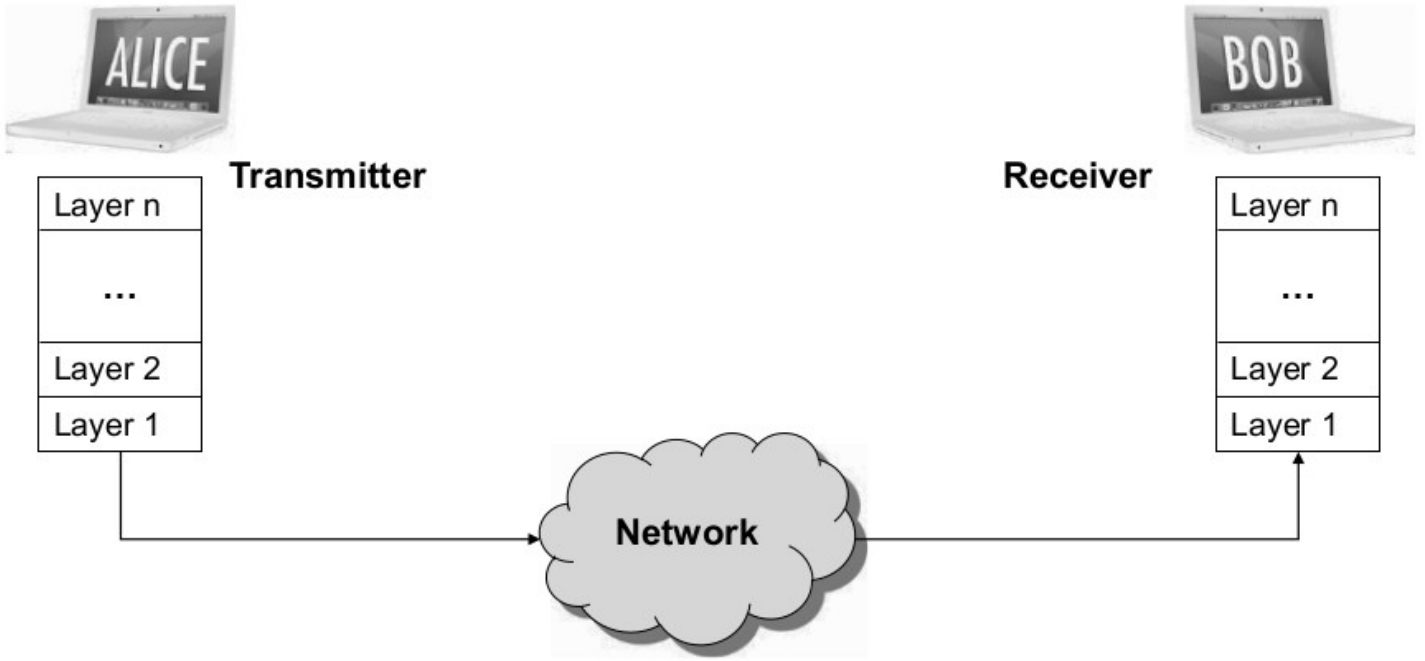
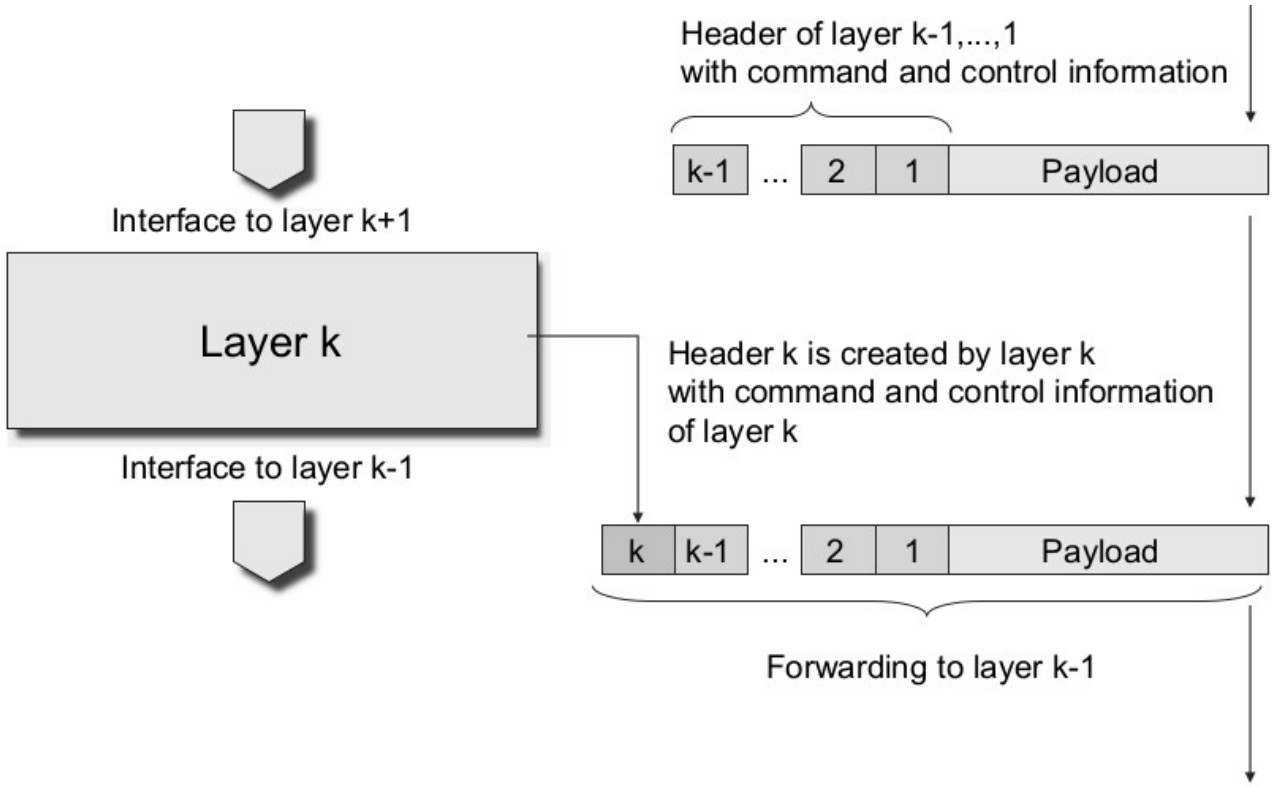


Fig. 3.16 Data transfer via a protocol stack.

- A message is transmitted from the application program of one computer to the application program of another computer.
- The message is passed from top to bottom via the various protocol layers on the source computer.
- It is then physically transported via the transmission medium.
- Subsequently, at the destination computer the same protocol layers are run through in the *reverse order* and the message transferred to the application.

Layer Models



- In the layer model, each layer is responsible for solving a certain part of the tasks in network communication.
- For this purpose, at the sending computer is added the necessary command and control information at every single layer of the protocol stack for solving the particular task.
- At the receiving computer, this extra information is read by the corresponding layer of the protocol software and processed further so that the transmitted data can be delivered correctly at the end.

Layer Models

- Based on the layer model of network communication, the protocol software of a specific layer k at the destination computer must receive exactly the message that was transmitted by the protocol software of layer k at the sending computer.

- Every change or adjustment applied to the data to be transmitted by the protocol of a certain layer has to be completely reversed at the receiver.

- If layer k adds an additional command and control header to the sent data, then layer k at the receiving computer must remove it again. When data encryption takes place at layer k on the receiver's side, the encrypted data at layer k must then be decrypted again

Layer Models

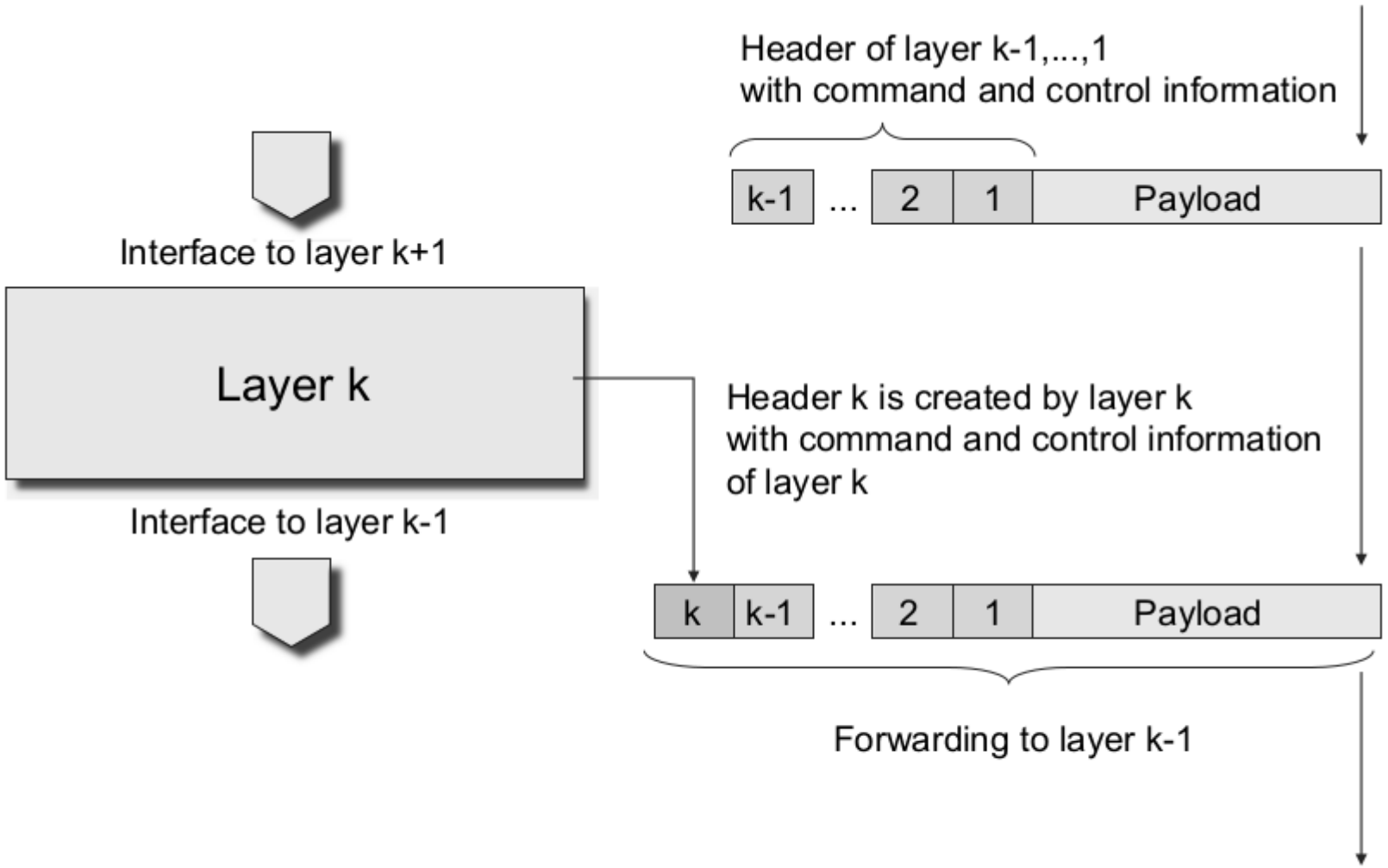


Fig. 3.18 Sending: Each layer of the protocol stack adds to the data to be forwarded its own header with command and control information.

Layer Models

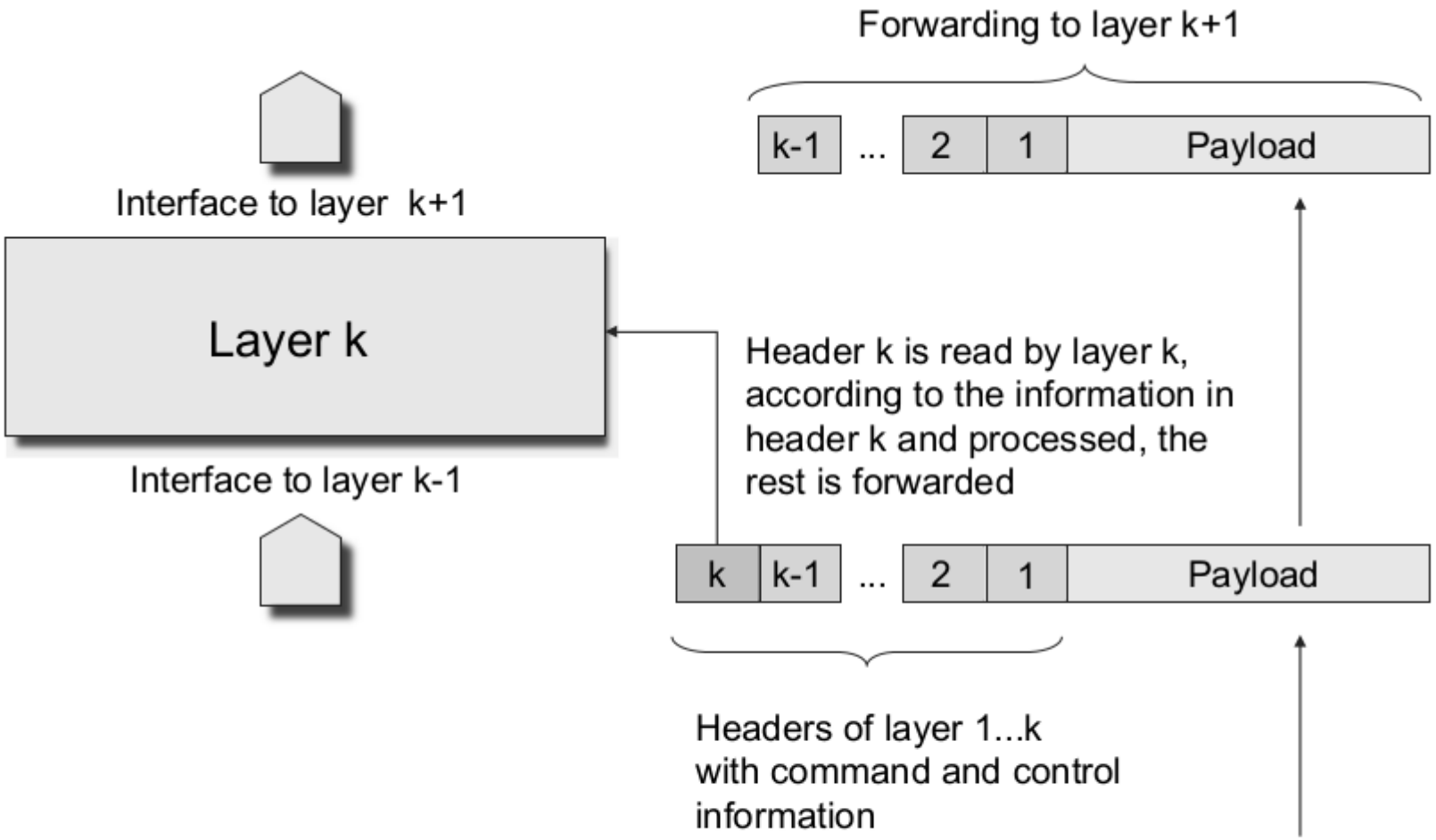


Fig. 3.19 Receiving: each layer of the protocol stack reads the header from the data received. It contains the necessary command and control information for processing at this layer.

Layer Models

- Actual communication is always carried out in a *vertical direction* in the protocol stack. When data is sent, each protocol layer adds its own command and control information.
- Most of the time this information is then prefixed to the layer above as a header – one speaks of the data packet as being *encapsulated*.
- The protocol software at the receiver's side, or in an intermediate system, receives the necessary command and control information from this additional data to ensure that the forwarded data is transmitted correctly and reliably.