Chapter V

Teaching Protocols through Animation

Kenneth J. Turner, University of Stirling, UK

Abstract

Communication protocols are essential components of computer and data communication networks. Therefore, it is important that students grasp these concepts and become familiar with widely used protocols. Unfortunately, communication protocols can be complex and their behavior difficult to understand. In order to learn about protocols, a student therefore needs a more controlled and constrained environment. This chapter describes the development and use of a protocol animator for teaching and learning communication protocols.

Learning Objectives

After completing this chapter, you will be able to:

• List and describe the desirable features of a protocol animation tool.
• Describe the main features of the JASPER protocol animator.
• Explain how an animation tool can be used in the classroom to enhance teaching and learning communication protocols.
• Define the following key terms: animation, simulation, message, time-sequence diagram, medium, protocol, and protocol entity.
• Consider further enhancements to JASPER and the protocols it simulates.

Introduction

Communications protocols are a vital ingredient of all networks. It is essential for students to grasp their fundamental principles and to become familiar with widely used protocols. Unfortunately, communications protocols can be complex and difficult to understand. Even the simple Alternating Bit Protocol displays surprising depths of complexity. It is easy to understand static aspects of a protocol such as message formats. Where protocols become difficult is in their dynamic behavior.

The main behavior of a protocol is often straightforward. However that may account for only 20% of its functionality. Protocols may be used in different configurations such as client-server or peer-to-peer. They have to operate reliably over a communications link that may exhibit a variety of faults. The complications in protocol design usually arise in error recovery: dealing with message loss, misordering, duplication, or misdelivery. Protocols operate in real time, so tracing their behavior in practice can be difficult.

In order to learn about protocols, a student therefore needs a more controlled and constrained environment. This chapter explains how a protocol animator has been developed to meet these needs. The animator is platform-independent, being written in Java, and is available in source code form, precompiled, and on the Web.

Background and Motivation

Existing Work

The term protocol animation is used in this chapter in the sense of giving life to a protocol definition (from Latin anima meaning soul). Protocol animation has received surprisingly little attention. An overview of work on the topic can be found in Turner (2002).