# Patterns of Protocols

Presented by Keep It Simple Advice: Acquiring Knowledge Simply

![Keep It Simple Advice](image)

---

## Table of Contents

If you're viewing this document online, you can click any of the topics below to link directly to that section.

1. Introduction .......................................................... 2
2. User-Provider-Agent ............................................... 3
3. Route-Resolve-Relay ............................................... 4
4. Feedback .............................................................. 6
Section 1. Introduction

What are Patterns of Protocols

Patterns help you understand things, Protocols are styles of communication, so Patterns of Protocols are ways of understanding styles of communication.

I divide them into Architectural patterns and Operational Patterns.

Architectural Patterns

Here lie architectural patterns.

Operational Patterns

Here lie operational patterns.
Section 2. User-Provider-Agent

Communicating in a stack

The User writes as message and the Provider inputs it. The User will also provide the address of the provider peer to which this message should be sent.

The provider has to frame the message into sizes that its agent will be able to handle, and then format the message into a packet in accordance with the grammar defined by the protocol.

Then the provider will decide if the packet is intended for itself or if it should be forwarded to another provider.

A packet that is delivered by an agent and received by the provider is already formatted properly, and the provider merely needs to give it to the forwarding decider.

Now the address of the peer-provider will be used to decide which provider is the next hop, and an agent and agent-peer-address can deliver to that hop will be determined, and the packet will then will be sent to the appropriate agent, which will accept it for delivery, which will accept it for delivery.

At this point we move down the stack: the original user is forgotten, the provider is now the user and the agent is now the provider, and will determine the appropriate agent as above.
Section 3. Route-Resolve-Relay

Connecting the stack and the network

When a provider is about to send a packet that it has properly formatted to an agent, it first has to decide which agent to use, and tell the agent which peer-agent it should contact.

Routing

Routing might be done by looking in a routing table that is kept on the machine, as in the case of an IP provider.

Routing might be done by asking an outside server for the next hop, as in the case of an SMTP provider which needs to look up MX records.

Resolving

Resolving means finding out which peer-agent can deliver the packet to the desired peer-provider.

* In the case of IP this resolution is done by an ARP protocol, which (at least for the typical Ethernet agent) broadcasts a request over that network giving the the agent address, the the provider address, and the desired peer-provider address. It expects a response to that agent address giving the peer-agent address that claims to be able to deliver to the desired peer-provider address.

* In the case of mail and MX records the typical DNS server knows that you are going to want a resolution, and often sends along the IP address of the SMTP peer-provider as well as a so-called glue record. If you are wondering if this violates layering because it is giving an IP address and not a TCP address the answer is "yes but", and probably it just adds TCP port 25 to the IP address to form the full TCP address.

Relaying

Relaying means actually handing over the provider-packet to be sent, which for most protocols contains the provider-address and the peer-provider-address formatted in accordance with the protocol specification. It also means handing over the desired peer-agent address. As far as the agent-address some protocols allow or even insist that the agent itself decides which of its addresses to use, and some allow it to be specified.

In either case we now move "down the stack" and the initial provider is forgotten, the
agent now becomes the provider, and begins itself to search for an agent!
Section 4. Feedback

Feedback

Please send feedback on this. The only way patterns can be developed is if individuals try them in various settings, and communicate about what they have found.

Colophon

This tutorial was written entirely in XML, using a modified version of the developerWorks Toot-O-Matic tutorial generator. The Toot-O-Matic tool is an XSLT stylesheet and several XSLT extension functions that convert an XML file into a number of HTML pages, a zip file, JPEG heading graphics, and two PDF files. Our ability to generate multiple text and binary formats from a single source file illustrates the power and flexibility of XML. (It also saves our production team a great deal of time and effort.)