

### SIP to PSTN Call Flow

In this scenario, Alice (sip:alice@a.example.com) is a SIP phone or other SIP-enabled device. Bob is reachable via the PSTN at global telephone number +19725552222. Alice places a call to Bob through a Proxy Server (Proxy 1) and a Network Gateway (NGW 1). Bob answers the call then Alice disconnects the call. Signaling between NGW 1 and Bob's telephone switch is ANSI ISUP.

 [http://www.eventhelix.com/RealtimeMantra/Telecom/SIP\\_PSTN\\_Call\\_Flow.pdf](http://www.eventhelix.com/RealtimeMantra/Telecom/SIP_PSTN_Call_Flow.pdf)

### SIP to ISDN PBX Call Flow

Alice is a SIP device while Carol is connected via a Gateway (GW 1) to a PBX. The PBX connection is via an ISDN trunk group.

 [http://www.eventhelix.com/RealtimeMantra/Telecom/SIP\\_ISDN\\_Call\\_Flow.pdf](http://www.eventhelix.com/RealtimeMantra/Telecom/SIP_ISDN_Call_Flow.pdf)

### H.323 Call Flow

The call flow diagram presents the flow of an H.323 call. The following steps are covered:

- H.225/Q.931 Call Setup
- H.245 Negotiation and Voice Path Setup
- RTP/RTCP Based Voice Communication

 [http://www.eventhelix.com/RealtimeMantra/Telecom/h323\\_call\\_flow.pdf](http://www.eventhelix.com/RealtimeMantra/Telecom/h323_call_flow.pdf)

 [http://www.eventhelix.com/RealtimeMantra/Telecom/h323\\_context\\_diagram.pdf](http://www.eventhelix.com/RealtimeMantra/Telecom/h323_context_diagram.pdf)

### Border Gateway Protocol (BGP) Startup Sequence Diagram

The Border Gateway Protocol (BGP) is an inter-autonomous system routing protocol. An autonomous system is a group of networks under common administrative control and routing policies. This sequence diagram describes the sequence of messages exchanged when a new BGP router is made operational. The steps involved are:

1. Establish TCP connections
2. Exchange BGP Open messages.
3. Start periodic exchange of Keepalive messages.
4. Exchange routing information with the BGP Update message.

For a detailed description of BGP, refer to RFC 1771.

 [http://www.eventhelix.com/RealtimeMantra/Networking/bgp\\_startup.pdf](http://www.eventhelix.com/RealtimeMantra/Networking/bgp_startup.pdf)

### Comparing C++ and C (Classes and Methods)

This two part series compares C and C++ by comparing the C++ code and its equivalent C code. This comparison should give you a better feel of the performance differences between C and C++. The first article covers:

- C++ Method Invocation
- Object Construction

- Object Destruction
- Static Access

 <http://www.eventhelix.com/RealtimeMantra/basics/ComparingCPPAndCPerformance.htm>

## Comparing C++ and C (Inheritance and Virtual Functions)

We have covered C++ and C performance in a [previous article](#). In this article we will dig deeper into C++ performance by analyzing the overhead of inheritance and virtual functions. The topics covered are:

- Memory Layout in Inheritance
- Object Construction and Destruction with Inheritance
- Virtual Function Invocation
- Memory Overheads of Inheritance

 <http://www.eventhelix.com/RealtimeMantra/basics/ComparingCPPAndCPerformance2.htm>

## Object Oriented Programming in C

Embedded software development is slowly moving towards object oriented analysis, design and programming. The introduction of object oriented technologies in some systems has not happened due to lack of C++ support on some platforms.

This article focuses on platforms where C++ compilers are not available. The developers working on these platforms should still be able to use object oriented analysis and design. When it comes to final code development they can use C.

The following sections cover an example of C++ code and its implementation in C.

- [C++ Source Code](#): C++ source files implementing TerminalManager and Terminal classes.
- [C Source Code](#): TerminalManager and Terminal implemented in C for a platform that does not support C++.

 [http://www.eventhelix.com/RealtimeMantra/Basics/object\\_oriented\\_programming\\_in\\_c.htm](http://www.eventhelix.com/RealtimeMantra/Basics/object_oriented_programming_in_c.htm)

## HTTP Post Sequence Diagram

HTTP Post is used by the browser to report form data when the user submits a form. This sequence diagram illustrates the flow of a typical HTTP Post. The sequence of actions is:

- The user fills a form and clicks on the submit button.
- The web browser establishes a TCP connection with the web server.
- The web browser sends an HTTP POST.
- The web server responds with "HTTP 200 OK" to indicate that the post has been accepted.
- The web browser displays the return page to the user.
- The user closes the browser.
- The browser releases the TCP connection.

 [http://www.eventhelix.com/RealtimeMantra/Networking/HTTP\\_Post.pdf](http://www.eventhelix.com/RealtimeMantra/Networking/HTTP_Post.pdf)

## Web Page Redirection Sequence Diagram

This sequence diagram shows the web page redirection using the HTTP 302 code. CNET's News.com redirect is taken as an example. The sequence is:

- The user enters http://www.news.com
- The web browser queries the DNS to locate www.news.com.
- The web browser sends an HTTP GET www.news.com
- news.com uses the HTTP code 302 to redirect to http://news.com.com
- The web browser automatically redirects and queries the DNS to locate news.com.com.
- The browser sends an HTTP get to news.com.com
- news.com.com replies with HTTP 200 OK, indicating success.

The collaboration diagram for web page redirect is also included.

 [http://www.eventhelix.com/RealtimeMantra/Networking/Web\\_Page\\_Redirection\\_Sequence\\_Diagram.pdf](http://www.eventhelix.com/RealtimeMantra/Networking/Web_Page_Redirection_Sequence_Diagram.pdf)

 [http://www.eventhelix.com/RealtimeMantra/Networking/Web\\_Page\\_Redirection\\_Collaboration\\_Diagram.pdf](http://www.eventhelix.com/RealtimeMantra/Networking/Web_Page_Redirection_Collaboration_Diagram.pdf)

## Web Page Caching and Refresh Sequence Diagram

This sequence diagram tutorial describes the web caching feature on the web browser side. HTTP primitives that facilitate the implementation of web browser level caching are shown using a live example.

Two scenarios are covered:

- The user revisits the EventHelix.com home page.
- The user issues a refresh command for the EventHelix.com home page.

 [http://www.eventhelix.com/RealtimeMantra/Networking/web\\_browser\\_caching\\_tutorial.pdf](http://www.eventhelix.com/RealtimeMantra/Networking/web_browser_caching_tutorial.pdf)

## EventStudio System Designer 2.5 User's Manual

An updated version of the EventStudio System Designer 2.5 user's manual is now available online.

 [http://www.eventhelix.com/EventStudio/EventStudio\\_System\\_Designer\\_Manual.pdf](http://www.eventhelix.com/EventStudio/EventStudio_System_Designer_Manual.pdf)