Tarantella Secure Global Desktop Enterprise Edition Software

Adaptive Internet Protocol

Abstract
Tarantella Secure Global Desktop Enterprise Edition provides secure, remote access to desktop applications and data to any user, anywhere, through a web browser interface. This White Paper outlines the Adaptive Internet Protocol, a patented technology that delivers optimum performance across any network to a wide range of client devices.

This white paper is for information only and is subject to change without notice.

Technical White Paper
February 2004

Contents

List of Figures

1 Introduction

Inherent to Network Computing is the need to find the optimal way to send data over the network. Typically, protocols are designed to accommodate a specific type of network connection, or to perform a particular function such as graphics rendering. The X Window System protocol, for example, was designed to optimally render graphics over a high-bandwidth network such as a fast LAN, whereas the ICA protocol, created by Citrix, was designed to work well over a low-bandwidth network, such as a modem, and so uses extensive compression techniques.

As Tarantella™ embraces a diverse range of clients and connection types, a protocol which adapts to the user's environment is necessary. Thus, the Adaptive Internet Protocol (AIP) was born. AIP is designed to provide optimal performance over complex network routes with varying bandwidths. It employs heuristics to determine the type of device and network connection used. These parameters are used to dynamically adapt and optimize performance.

With Tarantella, the server maintains a persistent session for the user. This enables users to disconnect their session and reconnect it later, even from a different location if wished. The reconnected session continues, using the same set of running applications. The user can also switch the type of client device used when reconnecting to a session. For example, a user could start off by reading mail at home on a Mac, then go to work and continue the session on a Windows® XP desktop. The AIP detects any change of client device or connection and automatically adapts to ensure optimal network performance.

The document provides a high level overview of the Adaptive Internet Protocol. It explains where AIP fits into the Tarantella framework, gives a basic overview of AIP and follows that up with a closer look at some of the heuristics involved in implementing the protocol specifically for graphical applications.

2 The Tarantella Secure Global Desktop Enterprise Edition

Three-tier Architecture

A basic understanding of the Tarantella framework and technology is necessary to gain an understanding of how the Adaptive Internet Protocol works.

Tarantella is based on a three-tier architecture (Fig. 1). These tiers are referred to throughout this document.

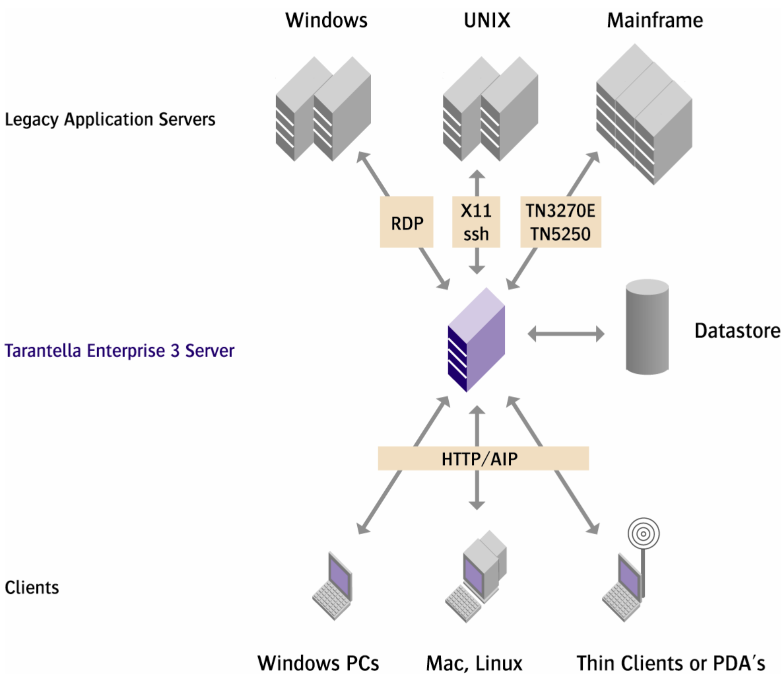

Fig. 1: The Tarantella Secure Global Desktop Enterprise Edition Three-tier Architecture.

The functions of the tiers are as follows:

Tier 1: The Client Devices
Tarantella supports client devices that run a Java™–enabled browser, such as Internet Explorer, or Tarantella Native Clients that run on a variety of devices ranging from Windows CE, Mac OSX through to Linux and Solaris systems. Client devices, therefore, will range from a full-blown Windows desktop to PDA devices.

Tier 2: The Tarantella Server
This enables the applications in Tier 3 to be viewed by the clients on Tier 1.

Tier 3: The Application Servers
This is where the applications reside—typically they will be either mainframe, UNIX® system or Windows applications.

The Tarantella Server talks the standard protocols to the Application Servers. For example, Tarantella speaks RDP to Windows Terminal Servers, X11/ssh to UNIX servers and IBM TN5250 and TN3270E protocols to mid-range and mainframe servers. The Tarantella Server communicates with the client using AIP.

Let's take a closer look at the Tarantella server (Tier 2) and the mechanism used to run applications, given Tarantella's three-tier architecture (Fig.2).

Built into Tarantella are Protocol and Display Engines. For each type of application that can be on Tier 3, there is an associated Protocol/Display Engine pair. For example, to run graphical applications there is a Graphics Protocol Engine and an associated Graphics Display Engine, and to run character applications there is a Terminal Protocol/Display Engine pair. The Protocol Engine runs on the server and does the bulk of the emulation by acting as a client to the application running on Tier 3. This Protocol Engine understands the standard protocols the application currently uses and then translates them into the Adaptive Internet Protocol that the Display Engine understands. Applications continue to run on the servers they are running on today, untouched and without any re-engineering or re-writes. These engines are implemented as native binaries to ensure optimum performance on the server.

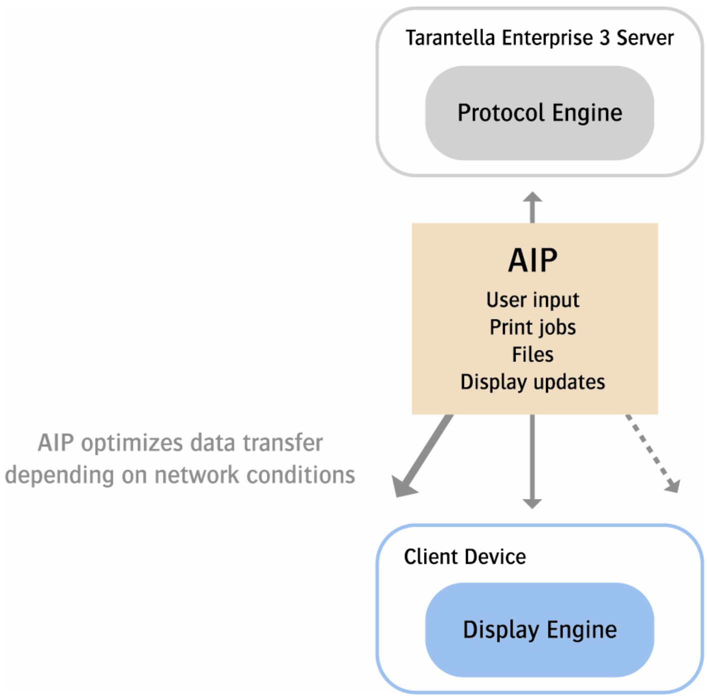

Fig. 2: AIP operates between the Tarantella client and server.

The Display Engines are Java applets that are downloaded on demand to the client device (or built into the Tarantella Native Clients). They have been designed to do just enough to render the application and allow input/output to the user. As a result they are quick to download, even over low-bandwidth networks, and know nothing about the state of the application.

3 Adaptive Internet Protocol (AIP)

The key word in AIP is *adaptive*. AIP's job is to constantly adapt itself to the client device capabilities, network bandwidth and network load. Consequently, if the client has little or no local processing power or if the level of network traffic is high, data will be sent over the wire in small, efficient packets, with much of the work being done on the Application Broker. On the other hand, if the client is fully capable and the network is fast, more of the processing will be done on the client and the data will just flow over the network. This adaptation is accomplished by constantly monitoring the network and client performance, adjusting internal models, and then acting accordingly.

AIP can best be described by looking closer at the some of the processing that happens between the Graphics Protocol and Display Engine.

4 AIP for the Graphics Protocol/Display Engine

For UNIX graphical applications, the Graphics Protocol Engine (XPE) enhances a basic X server to provide intelligent heuristics and adaptive capabilities. As the applications themselves are unchanged, they talk to the XPE just as they would to an X server. The XPE then massages the data and transmits the display information via AIP to the X Display Engine (XDE) on the client device.

Let's look a little closer at what is actually happening in the XPE (Fig.3).

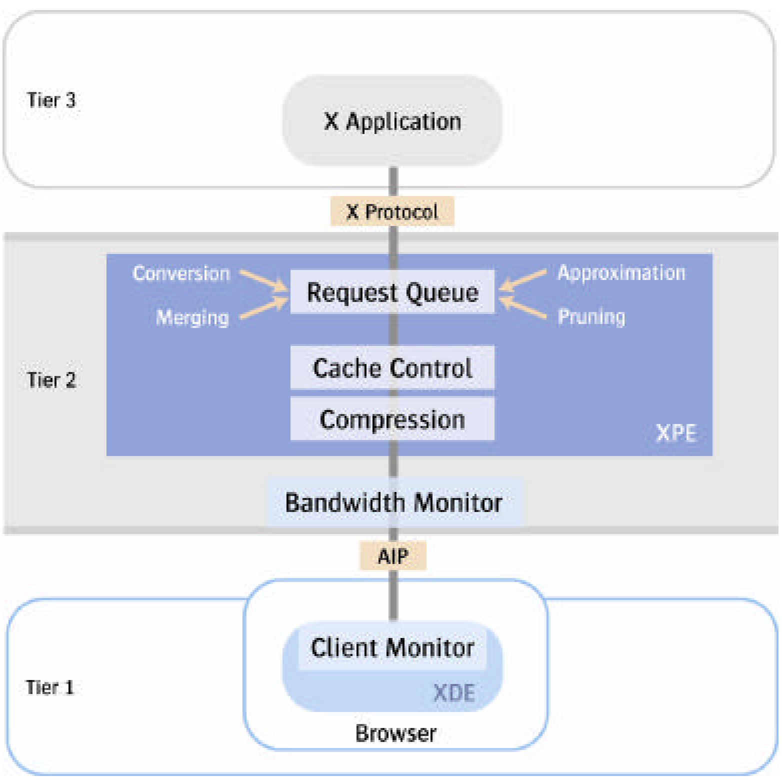

Fig. 3: AIP for the X Protocol/Display engine pair

The XPE takes requests from the X application and forms a Request Queue. It then performs a number of operations on the Request Queue to optimize it prior to sending the requests to the XDE. These operations are:

4.1 Request Pruning

Typically, X applications send a drawing request to draw one element and then immediately send another request that draws right on top of it. This is usually a result of the toolkit used by the authors of the application. Requests such as these are pruned out of the queue and discarded.

4.2 Request Merging

Instead of sending drawing requests one by one to the XDE, the requests are combined into a single image and that single image is sent to be displayed. Merging occurs on the server because the time it takes a Java applet to draw a single image is faster than if the requests were sent one by one.

4.3 Request Conversion

If the Display Engine is not capable of executing a particular request (or if the request executes slowly) the request is converted into another form. For example, if the Display Engine is not capable of drawing wide, dashed circles, those requests could be replaced with a series of polygon/rectangle fill operations.

4.4 Request Approximation

To improve performance, drawing requests can be replaced with requests that only approximate the original request. This approximation may produce slight variances in output (e.g. a different shade of color) but the actual update rate is much faster. If an approximation is done, a screen fix operation is automatically scheduled (for the near future) to update the screen contents to the correct state.

4.5 Update Scheduling

Some drawing requests have to be drawn into a frame buffer before being copied on the screen. Since screen updates are often slow operations, the XPE can look ahead in the request queue to see if there are other requests that will also need a draw update operation. If there are such requests, the screen update is delayed until after these operations have been completed.

4.6 Caching

The XPE examines the Request Queue for repetitions of particular elements. Fonts, stipples, files and pixmaps are often repeated, wasting valuable bandwidth. XPE instructs the XDE to cache these on the client device, so they don't need to be sent every time they're used.

4.7 *Variable Compression*

General compression and compression of images minimizes the amount of information sent over the network. The amount of compression done is variable since there is a trade-off between the time it takes to compress the data in the XPE and then uncompress the data in the XDE, versus just sending continuous output on the network.

4.8 *Bandwidth Monitor*

The feedback that comes from the bandwidth and client monitors determines whether some or all of these operations are performed. These monitors are constantly sending feedback to indicate the performance of the client device and the network latency and bandwidth. This feedback dictates exactly how much processing the Protocol Engine will do and how many operations will be left to the client device. If the client device can handle continuous output, it is usually better to send a continuous flow of data, otherwise there is a risk of the display appearing jerky. Also, client devices involved in user interaction require screen updates to be visible as soon as possible. Give this, the XPE classifies as either ‘continuous’, ‘user input’ or ‘normal’ and the level of optimization and priority is varied in terms of output and scheduled accordingly.

4.9 *AIP Configuration*

Tarantella Administrators can chose to enable or disable parts of AIP on a per-application basis, using the Tarantella administration tools.

5 Conclusion

Adaptive Internet Protocol (AIP) is a unique technological breakthrough for remote presentation. It adapts itself automatically to ensure data is delivered to the client at optimal performance levels.

By using the AIP, not only does MIS not need to know about the client device type, users themselves have the freedom to change client device at any time.

AIP uses intelligent heuristics to automatically adapt as the network traffic changes. It learns for itself and so requires no user input and is completely hidden.

6 Related Links

See the following resources for further information:

For the latest information about Tarantella Secure Global Desktop Enterprise Edition, see the Tarantella website at http://www.tarantella.com/products/e3

The AIP Patent is US Patent Number 6,104,392.

7 Contact Information

Corporate and Worldwide

Tarantella, Inc.
425 Encinal Street
Santa Cruz
CA 95060
United States

Tel: +1 831 427 7222
Fax: +1 831 427 5400

Europe

Tarantella Ltd
2nd Floor
Richmond House
Lawnswood Business Park
Redvers Close
Leeds
LS16 6RD
UK

Tel: +44 113 368 6000

Toll Free Sales Information

Corporate & Worldwide

+1 888 831 9700

www.tarantella.com
sales@tarantella.com

Europe

UK: 0800 0390134
France: 0800 913184
Germany: 0800 1802450
Italy: 0800 781920

Local Offices

USA Corporate Headquarters

Tarantella, Inc.
425 Encinal Street
Santa Cruz
CA 95060
USA

Tel: +1 831 427 7222
Fax: +1 831 427 5400

United Kingdom

Tarantella Ltd
2nd Floor
Richmond House
Lawnswood Business Park
Redvers Close
Leeds
LS16 6RD
UK

Tel: +44 113 368 6000

USA Corporate Headquarters

Tarantella, Inc.
425 Encinal Street
Santa Cruz
CA 95060
USA

Tel: +1 831 427 7222
Fax: +1 831 427 5400

Italy

Tarantella
Via Bergamo 6
20060 Cassina de' Pecchi
Milano
Italy

Tel: +39 029544941
Fax: +39 0295449434
info.italy@tarantella.com

Germany

Tarantella
Vertriebsniederlassung
Deutschland
Zimmersmühlenweg 68
61440 Oberursel
Germany

Tel: +49 (0) 6171 9879570
Fax: +49 (0) 6171 9879571
info.germany@tarantella.com

Italy

Tarantella
Via Bergamo 6
20060 Cassina de' Pecchi
Milano
Italy

Tel: +39 029544941
Fax: +39 0295449434
info.italy@tarantella.com

Spain

Tarantella
Calle Doctor Fleming 55, 3 dcha,
28036 Madrid
Spain

Tel: +34 913 590 478
Fax: +34 913 507 849
info.spain@tarantella.com

Spain

Tarantella, Tarantella Secure Global Desktop Enterprise Edition and the Tarantella logo are trademarks or registered trademarks of Tarantella, Inc. in the USA and other countries. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the USA and other countries. All other brand and product names are or may be trademarks of, and are used to identify products or services of, their respective owners. This document is provided "as is" and may include technical inaccuracies or typographical errors. Tarantella, Inc. reserves the right to add, delete, change or modify this document at any time without notice. This document is for information only, no express or implied representations or warranties are given in this document. Copyright © 2004

Tarantella, Inc. All Rights Reserved.
TWP_E3_AIP_Feb04.doc