



Seller's Reference Manual Merkato Version 2.3

October 2002

©Copyright 2002 InvisibleHand Networks

All Rights Reserved.

This document contains information, which is protected by copyright. Reproduction, adaptation, or translation without prior permission is prohibited, except as allowed under the copyright laws.

The following are trademarks, registered trademarks, or service marks of InvisibleHand Networks: Merkato

Java is a trade mark of Sun Microsystems

Disclaimer:

The information contained in this document is subject to change without notice.

DOCUMENTATION IS PROVIDED “AS IS” AND INVISIBLEHAND NETWORKS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

InvisibleHand Networks, Inc. shall not be liable for errors contained herein, or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Document Number: MBR-2.3-1.00

Contents

What is Merkato?.....	1
Buyers, Sellers, and Agents in the Auction Market	2
Your Options as a Merkato Seller.....	3
Defining the Bandwidth Product	3
Defining the Bandwidth Marketplace	4
Seller Type-of-Service Options.....	6
Flow Identification.....	6
Traffic Direction.....	7
Bandwidth Limit Method.....	7
Seller Market Control Options	8
Maximum/Current Bandwidth Available per Market.....	8
Reservation Price Table.....	9
Percentage Fee.....	10
Spot Market Floor Price.....	11
Spot Market Selling Strategy	11
Spot Market Buy-Back Agents	12
Reservation Market Buy-Back Agents.....	12
The Seller Agent Interface	13
Choice of Interface.....	14
Java on Your Browser	15
Automatic Download	15
Active Download	15

Desktop Interface	16
Arranging Agent Windows.....	16
Detached From the Merkato Desktop.....	16
Tiled on Merkato Desktop	17
Cascading on Merkato Desktop.....	17
Upload Function.....	18
Connect/Start/Stop.....	19
Bumping	20
Saving Your Configuration	20
Exiting Without Saving.....	21
Exiting the Merkato Desktop.....	21
Spot Market vs. Reservation Market	22
Spot Market Agent Window.....	23
Resource Window	23
Units Window	24
Seller Valuation.....	25
Seller Strategy	27
News Windows	28
Allocation Window	29
Auction Graph	30
Auction Table.....	31
Saving, Uploading, and Exiting.....	32
Reservation Parameters	34
The Merkato Portal.....	35
Express Agent View	37
Express Agent Interface	37
Recent Allocations Table	39

Traffic View	40
Charge Reports	41
Transaction Data.....	45
Merkato Auction Mechanism: The Progressive Second Price Auction.....	46
Advanced Sellers Guide.....	50
Controls That React To Market Conditions.....	50
Valuation for Buy-Back Agents	50
Example	51
Considerations About Buy-Back Agents	51
On-Demand Controls.....	52
Settings Controlled by Merkato Administrator	52
Bid Fees.....	53
Use of the Bid Fee	54
Valuation Formulas.....	55
Budget Valuation.....	56
Budget-with-limits Valuation	56
Linear Valuation.....	56
Square Root Valuation.....	56
Logarithmic Valuation.....	57
Parabolic Valuation.....	57
Glossary.....	58

What is Merkato?

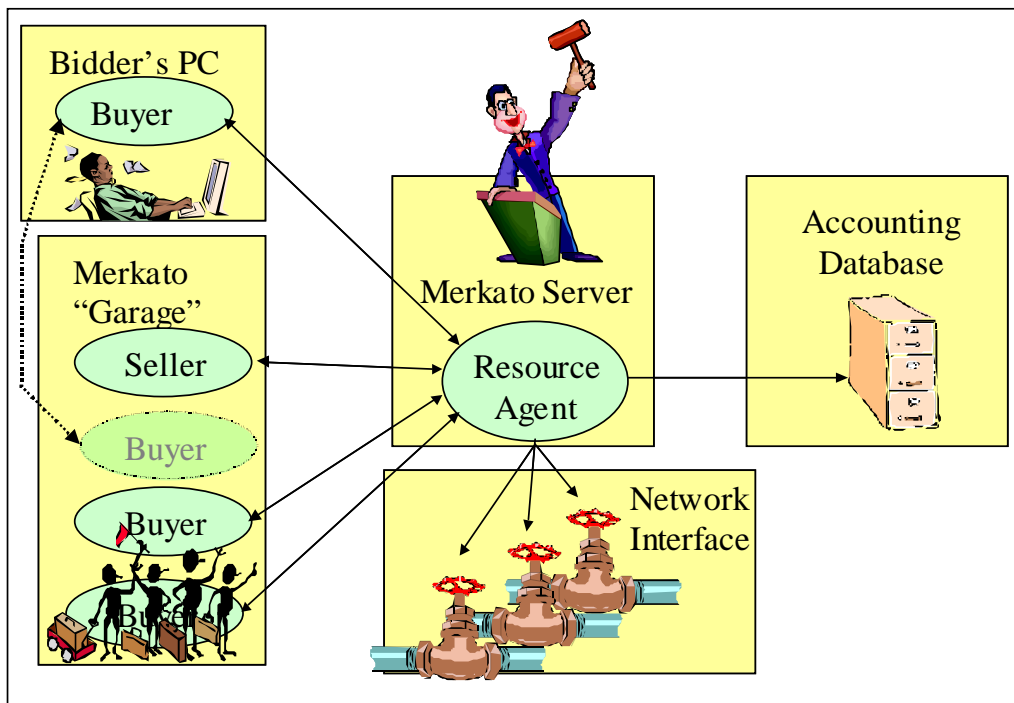
Merkato is a software platform that supports a unique way to obtain bandwidth, on demand, at fair and optimal prices.

There are two ways to purchase bandwidth in a Merkato system.

- You can get a fixed amount of bandwidth, at a fixed price, for a fixed term
- You can create a bidding strategy and contend for bandwidth with other buyers, establishing a *fair market price* based on supply and demand.

The first method is called the Reservation market. The second method is called the Spot market.

The spot market provides a true balanced supply-and-demand marketplace. The Merkato spot market uses the patent-pending Progressive Second Price Auction mechanism to collect bids from multiple buyers, analyze the bids to determine the natural market price for offered bandwidth, and then automatically apportion that bandwidth to bidders at the market price. The components of the Merkato system are illustrated below.



Buyers, Sellers, and Agents in the Auction Market

Agents are programs within Merkato that interact on behalf of buyers and sellers. Buyers can acquire bandwidth by configuring their agents with the price they are willing to pay for a range of available quantity, or with the reservation they want. Sellers configure their agents with a quantity of bandwidth for sale and a minimum price for that quantity. Buyer agents then express what they are willing to pay for bandwidth in the form of bids, consisting of a unit price and a quantity.

Sellers create a price floor by offering to purchase all the available bandwidth at their lowest acceptable price.

The Merkato market responds to bids with proposed allocations and pricing. The buyer and Seller agents can either accept the proposed allocation (by not bidding further), or reject the proposed allocation (by submitting another bid). When there is no more bidding, the auction closes, and Merkato makes bandwidth allocations.

The auction process typically takes no more than five minutes. The auction is completely open and fair, with all bidders seeing what other bidders have offered. The Progressive Second Price auction mechanism ensures that all bidders receive their allocations at the fair market price, which is equal to or less than the price they offer. (See “Merkato Auction Mechanism: The Progressive Second Price Auction” on page 46.)

Your Options as a Merkato Seller

As a seller, you decide the kind of marketplace you are creating, and the Merkato administrator configures the system accordingly.

Merkato controls an interface of a router or QoS-enabled switch. The capacity of this interface, or a portion of it, is a product you are offering to buyers through the Merkato automated marketplace.

Merkato is a very flexible platform. You can select among many configurable options that define the bandwidth product you are offering to customers.

Defining the Bandwidth Product

Answer the following questions to define the bandwidth product.

- **How will buyer traffic be identified?** Merkato must be able to distinguish between data belonging to each buyer so that bandwidth can be apportioned correctly. The most common identifier is an IP subnet—identified by an IP address and subnet mask. Traffic can also be identified by the physical port or logical interface to which a buyer is connected, by VLAN tags on the buyers' data streams, or by the MAC address of the next-hop router from which a buyer's traffic flows. (All buyers within the same marketplace should use a common method for identification of their traffic to avoid unnecessary complexity.)
- **In what direction will bandwidth be apportioned?** Although buyer traffic nearly always flows both in and out of any interface, it is often unnecessary to apportion bandwidth in both directions. Generally, there is an obvious direction that corresponds to the application being addressed. For example, if you use Merkato to apportion bandwidth among content providers, the direction to control would be from the source of content to the receivers of that content. If there is no obvious direction, Merkato can be used to apportion bandwidth in both directions simultaneously. It is also possible to use Merkato to create separate markets for incoming and outgoing traffic, but you should be careful that, if you grant bandwidth in one direction, you allow sufficient bandwidth for protocol acknowledgements in the other direction.
- **How strictly do you want to limit bandwidth?** The most common method of enforcing bandwidth limits is the least forgiving—to discard any traffic that exceeds the amount of bandwidth purchased. In some cases, however, you may wish to allow excess traffic to be sent if there is sufficient unused capacity. We only recommend allowing this permissive behavior in unique applications. When taken to extremes, any buyer could use the whole amount of bandwidth for sale during periods when no other buyers happen to be sending data. The result is that your buyers will not be purchasing bandwidth, but rather be purchasing bandwidth

insurance—guarantees of a minimum amount of bandwidth during periods of congestion. Buyers may discover that they need to purchase only the absolute minimum amount of bandwidth except during times of network congestion, so the market for bandwidth tends to be very limited unless congestion is very common.

Defining the Bandwidth Marketplace

Answer the following questions to define the Merkato bandwidth marketplace.

- **How much bandwidth do you want to release to the market?** The price of bandwidth sold in the Merkato system varies according to the level of demand. You can stimulate demand by reducing amount of bandwidth you make available. Of course, you sell less total bandwidth this way. At the other extreme, if available bandwidth always exceeds demand, the market price remains at your floor price. You can specify the amount of bandwidth to be sold on the Spot market and on the Reservation market separately. In the Spot market, you can adjust the capacity at any time via an Agent interface. Changing the capacity of the Reservation market requires contacting your Merkato administrator.
- **What pricing levels do you want to create for the Reservation market?** Your administrator sets the prices on the Reservation market using a rate table you dictate. You can set pricing based on the amount of bandwidth the buyer requests and on the length of time for which the buyer requests it. The table uses price points that you define. By default, Merkato 2.3 supports four time levels and five quantity levels (a total of 20 price points), but this can be increased. You also specify a cancellation fee. The cancellation fee is a percentage of the remaining value of the contract, charged if the buyer terminates a reservation prematurely.
- **What floor price do you want to set for the Spot market?** In the Spot market, bidding for bandwidth always begins at the floor price you set. This price is configured dynamically via an agent you control. Changes take effect at the start of the next round of bidding.
- **Do you want to create buy-back agents to automatically control the supply of bandwidth?** Buy-back agents have the effect of dynamically releasing bandwidth to the marketplace at fixed price points. These artificial buyers bid at certain price points and quantity levels and “win” the bandwidth until they are outbid by true buyers. (You should reveal this to the real customers so that they do not feel you are unfairly manipulating the market.)
- **Do you wish buyers to access Spot and Reservation markets through a single agent or through separate agents?** Normally, users use a single username and password to access both the Spot and Reservation agents. The buyer selects the agent to download from either the Spot garage or the Reservation garage. Alternatively, both agents can be kept in the same garage, but must be accessed via different usernames. (These usernames should be chosen to indicate which market they apply to.) The convenience of logging in with a single username may outweigh the slight inconvenience of selecting the proper garage—this is your choice.

- **How often would you like spot market auctions to be run?** By default, spot market auctions run as often as possible—every five minutes or so. We recommend running auctions as often as possible to provide buyers with the fastest possible response time to their changes. If, for some reason, you want to run auctions less often, your Merkato administrator can increase the pause time between auctions. By default, the pause time is just long enough to allow allocations to be configured in the network before the next auction starts. Pause time can be increased to lengthen the time between auctions. When you lengthen the time between the auctions, the auctions themselves last for the same duration.

Configuring many of these options is discussed in “Merkato Auction Mechanism: The Progressive Second Price Auction” on page 46.

Seller Type-of-Service Options

This section discusses seller options related to the qualities of the bandwidth being sold. These options include identification of buyer flows, in which direction bandwidth is controlled, and how capacity utilization is limited.

Flow Identification

Your router or QoS-capable switch must be able to distinguish data streams belonging to customers on a shared link. The Merkato administrator configures the detection scheme you choose based on information you provide about the buyers. Identification options include:

- IP address plus subnet mask
- Physical Port (or logical interface)
- VLAN tag
- MAC address of the next-hop router

These options are enabled by the use of Access Control List (ACL) filters on the router or a switch that enforces the Merkato appropriations.

Use the IP address plus the subnet mask when customer traffic is distinguished by either the source or destination IP address within the packets, or by a range of source or destination IP addresses.

Use the physical port (or logical sub-interface) to distinguish customers by the router port (or logical sub-port, also called a sub-interface) through which the traffic flows.

In some cases, customer traffic can only be differentiated within a device other than the router doing the bandwidth apportioning. In this case, the device doing the differentiating should label packets from different buyers with unique VLAN tags. The control router uses the tags to identify which data streams belong to each customer.

In some cases, customer traffic can be differentiated by the device sending the data to the control router (or receiving data from it). To accommodate this application, we allow use of the MAC (physical port) address of the next-hop device to distinguish customer traffic. To accommodate device replacement, we configure ARP (address resolution protocol) on the control router. ARP lets the router learn the MAC address of the next hop by asking this adjacent device for the information.

Traffic Direction

In nearly all Merkato applications, bandwidth is apportioned in one direction only. For example, if the buyers' traffic consists of locally sourced data streams, Merkato would typically be used to apportion bandwidth from the data source hosts to the distribution network. If the dominant traffic direction is not uniform among buyers, Merkato can be configured to apportion bandwidth in both directions simultaneously. Sellers must discuss their network structures with their Merkato administrator and determine the best setting.

Should you wish to sell bandwidth in both directions independently, consider that data traffic seldom flows one direction only. Most protocols need at least a limited amount of bandwidth in the opposite direction to accommodate protocol acknowledgements. Generally, a reverse capacity of one tenth of the bandwidth in the primary direction is sufficient. Merkato has the ability to maintain a "best effort" data path for non-bidders and for bidders who receive no allocation. This best effort feature should be used to supply the necessary bandwidth for the reverse paths.

Bandwidth Limit Method

For most applications, Merkato uses the control router's ability to apply rate limits to traffic streams to enforce bandwidth allocations. In this mode, allocations are dedicated to buyers whether they use them or not.

Merkato also supports a less restrictive method of limiting bandwidth, which amounts to "bandwidth insurance." Bandwidth insurance is implemented using weighted fair queue settings on the control router. In this mode, bandwidth allocations only guarantee a minimum amount of capacity that the buyer's traffic will receive. If buyers do not use their full allocations of bandwidth, the excess bandwidth is returned to a common pool that all buyers can use. In the limiting case, a buyer could potentially use all the available bandwidth while purchasing a very small amount, assuming that all other buyers are not using their allocation during that time period.

We typically do not recommend using the "bandwidth insurance" mode; it tends to make the market for bandwidth collapse except during times of congestion. Only in cases where demand exceeds supply frequently, but supply cannot be increased, have we seen value in this mode of operation.

Seller Market Control Options

As the seller, Merkato gives you many options for manipulating the quantity of bandwidth based on current demand and market price. The Merkato administrator configures some of these options during initial installation—these should not be changed. You can control some options on-demand by changing settings via a Seller agent. These changes take effect at the start of the next auction. Some options can be pre-configured to act dynamically during an auction, based on changing market conditions.

Maximum/Current Bandwidth Available per Market

Your Merkato administrator must configure the maximum bandwidth capacity for a link in the NSC (Network Service Controller). The administrator can then divide that bandwidth among one or more Merkato markets, or keep it on unused until demand in a market increases.

You can divide a single link among any number of Spot or Reservation markets. In general, the only reason to create multiple markets of the same type (for example, multiple spot markets) is when there is some feature that distinguishes bandwidth in each market, and this feature affects the value buyers place on that bandwidth. In general, it is best to have as many buyers as possible in a single market to create a more competitive environment.

For each Reservation market, you must enter the total capacity available into the NSC and the capacity currently available into the Resource agent. (The capacity currently available must be less-than or equal to total capacity.)

The Merkato administrator sets these values and can increase or decrease them as desired. Of course, the administrator should never reduce the capacity below the amount currently under contract (and in use) by buyers.

The Spot market has multiple ways to control total capacity. The maximum is set at both the NSC and the Resource agent. The “current” capacity is controlled on-demand by sellers via their agents. Changes take effect at the beginning of the next auction round.

Reservation Price Table

Reservation pricing is set using a pricing table. Only your Merkato administrator can make changes to this table. The reservation price table is shown in Figure 1.

The following values are configurable:

- Each cell, specifying a unit price for the adjacent time period shown in the column heading and capacity amount shown in the row heading.
- Values for each time column heading
- Units for each time column heading
- Values for each quantity row heading
- Units for each quantity row heading
- The fee charged for cancellation of a reservation (see “Percentage Fee,” below.)

In general, sellers offer the highest price for short reservations and for small quantities of bandwidth. They offer the lowest price for large quantities of bandwidth over long periods of time.

When a buyer requests a reservation, Merkato calculates a price based on the requested amount of bandwidth and the duration of the reservation. The reservation prices are automatically scaled between price points to prevent abrupt jumps when buyers select quantities just above or below the price point.

Your Merkato administrator can increase the number of rows or columns in the rate sheet to accommodate more price points.

The highest capacity rate and the longest time period shown in the table are the upper limits for reservations.

Improved Table Pricer

Percentage Fee

<input type="text" value="1.0"/>	<input type="text" value="h"/>	<input type="text" value="24.0"/>	<input type="text" value="h"/>	<input type="text" value="7.0"/>	<input type="text" value="day"/>	<input type="text" value="1.0"/>	<input type="text" value="month"/>
<input type="text" value="1.0"/>	<input type="text" value="300.0"/>	<input type="text" value="276.04"/>	<input type="text" value="253.72"/>	<input type="text" value="231.7"/>			
<input type="text" value="5.0"/>	<input type="text" value="304.85"/>	<input type="text" value="278.57"/>	<input type="text" value="254.08"/>	<input type="text" value="229.93"/>			
<input type="text" value="10.0"/>	<input type="text" value="309.24"/>	<input type="text" value="280.86"/>	<input type="text" value="254.41"/>	<input type="text" value="228.32"/>			
<input type="text" value="50.0"/>	<input type="text" value="361.24"/>	<input type="text" value="307.94"/>	<input type="text" value="258.28"/>	<input type="text" value="209.29"/>			
<input type="text" value="100.0"/>	<input type="text" value="405.62"/>	<input type="text" value="331.05"/>	<input type="text" value="261.58"/>	<input type="text" value="193.05"/>			
<input type="text" value="Mbps"/>	<input type="text" value="\$"/>	<input type="text" value="/"/>	<input type="text" value="month"/>	<input type="text" value="/"/>	<input type="text" value="Mbps"/>		

Figure 1. The Reservation Price Table

Percentage Fee

In addition to the unit price for reserved bandwidth, you can specify a cancellation fee. This fee is expressed as a percentage: the fee is that percentage times the remaining time on the reservation. If the reservation is cancelled, this percentage of the value of the cancelled portion of the reservation is charged to the buyer. For example, if the cancellation fee is 50% and the buyer cancels the last quarter of the contract, the buyer owes 25% of the original contract price (the remaining portion of the contract) times 50% (the cancellation fee), or 12.5% of the original contract price. Buyers are informed of this fee in a pop-up confirmation window when they ask to cancel a reservation.

Spot Market Floor Price

You can control the spot market floor price—the lowest price at which you are willing to sell bandwidth. This price is important as it is not only the minimum price, but also the unit price that is charged to all buyers if all the bandwidth in the market is not sold. This is because Merkato charges all buyers the price of the last unit of bandwidth sold. If all the bandwidth is not sold, the floor price is the basis for pricing.

You set the floor price in the seller valuation fields in the Express interface, shown in Figure 2.

Any changes you make take effect at the start of the next auction.

The price in the “Budget/Max Value” field represents the floor price for the entire Max Quantity. Therefore, the unit price is the “Budget/Max Value” divided by the “Max Quantity.”

Note that if you change the “Max Qty” to raise or lower the amount of bandwidth available in the marketplace, you must proportionally raise or lower the “Budget/Max Value” to maintain the same *unit* floor price.

Current Valuation	Seller Valuation	
Valuation View	Seller Valuation ▾	
Valuation Parameters	Min Qty	<input type="text"/> Mbps
	Max Qty	<input type="text" value="0.49"/> Mbps
	Budget/Max Value	<input type="text" value="450"/> \$/month
	<input type="checkbox"/> Max Qty = All	

Figure 2. Seller Valuation Fields in the Express Agent

Spot Market Selling Strategy

Merkato allows you to select between two strategies for pricing bandwidth on the spot market: static and dynamic.

The **static strategy** is Merkato’s basic mode of operation. This strategy is based on the second price auction mechanism—the successful bidder obtains the item being auctioned at a price based on the amount offered by the highest unsuccessful bidder. In the case of bandwidth, this is the price offered by the highest losing bids for the same amount of bandwidth the buyer receives. If the total amount of bandwidth requested by the losing bidder is less than that of the successful bidder, the difference is made up using the floor price. (See “Merkato Auction Mechanism: The Progressive Second Price Auction” on page 46.)

The disadvantage of this selling strategy is that, because buyers are allowed to bid many times during an auction cycle, until they are satisfied with their allocation, there are rarely any unsuccessful bidders. Using this method, virtually every auction would close at the floor price. Merkato solves this problem by offering the option to base pricing on the lowest successful bid rather than the highest unsuccessful ones. This is the **dynamic** strategy.

This alternative pricing paradigm is created without changing the auction rules used under the static selling strategy. Under the dynamic selling strategy, an agent you create bids for all the bandwidth at a price high enough to just barely lose. Since the highest losing bid is the price everyone pays, this agent

creates the basis for the pricing for all bidders in the “second price” auction. Since the dynamic seller bids barely below the price offered for the last unit of bandwidth, it never purchases bandwidth itself. (It will, however, purchase bandwidth to maintain the floor price, if necessary.) The action of the dynamic seller can be monitored in the auction graph and auction table in the desktop interface. (See “Auction Graph” on page 30 and “Auction Table” on page 31.)

Spot Market Buy-Back Agents

Some sellers may want to create a more complex floor price strategy. For example, you may want to release additional bandwidth to the marketplace at a certain price point. You can do this by increasing the amount of bandwidth available in the marketplace and simultaneously using *your own* Buyer agent to bid for that same amount of bandwidth at the desired price point. To obtain that additional bandwidth, a real buyer would have to out-bid this “buy-back agent.” You can duplicate this same method to release bandwidth at various price points with multiple buy-back agents. You can also create a buy-back agent that uses a more complex bidding strategy—one that gradually releases bandwidth to the marketplace as prices rise.

We recommend that you reveal your buy-back agents to your legitimate buyers so that they can factor this behavior into their bidding strategy. Static buy-back agents are “self revealing,”—easy to interpret by buyers using the Auction Table display. If you intend to change settings of your buy-back agents often, your buyers should understand your intentions. You should teach them to interpret the buy-back agent’s behavior on the Auction Table.

Reservation Market Buy-Back Agents

The Merkato administrator configures the maximum amount of bandwidth available in the Reservation market via settings in the Resource agent. Using a buy-back agent it is possible to adjust the bandwidth more dynamically. You can have your Merkato administrator create your own reservation Buyer agent and use it to obtain bandwidth on your behalf. This bandwidth is off the market for the duration of the reservation and returned to the market at the end of the reservation—or sooner if you cancel the reservation. Ordinarily the administrator would change the quantity offered for sale. Freeing up bandwidth by having the buy-back agent cancel reservations lets you offer more bandwidth instantly without the appearance of changing the quantity.

The Seller Agent Interface

Your first point of entry to the Merkato System is through the Portal, a web-based application that provides accounting and status information, as well as access to the Merkato agent. Your version of the portal may be slightly different than that the one below, but the functionality is the same.

Note: Merkato has been tested on Internet Explorer version 6.0 and later versions. While Merkato may work on other browsers, we have not verified this.

Enter a valid user name and password and click **Login**.

After you log in the Express Agent interface appears in the Merkato Portal, shown in Figure 3.

From this top-level agent page you can get to the various screens that make up the portal, and to the desktop interface.

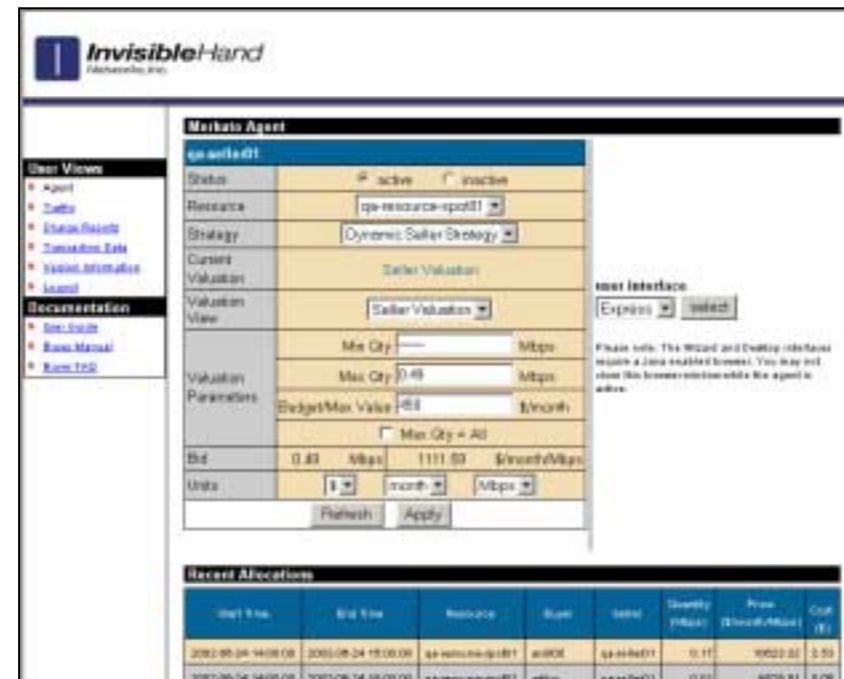


Figure 3. The Opening Merkato Portal Screen

Choice of Interface

You can choose from among interfaces to set preferences and get information. The User Interface menu gives you three choices.



- The **Desktop** interface allows you to sell bandwidth and monitor activity through a series of user-selectable windows. Your agent is brought to your PC and bids on your behalf from there.
- The **Express** interface lets you obtain status information from, and send configuration information to, an agent in the garage. The Express interface is displayed on the Portal page upon successful login. The Express interface is considered advanced because the fields are less interactive than the Java-based Merkato Desktop interface.

(The **Wizard** does not currently support seller configuration. If you select the Wizard an error message appears.)

When you are not interacting with your desktop agent you should upload it to the garage. This is critical for Seller agents on the desktop because if the Seller agent ceases to bid from the desktop the auction stops. (See "Upload Function" in *Seller's Guide to Using Merkato*.)

Java on Your Browser

If your browser is not configured for Java version 1.4.1 you must download it before you can run Merkato on your desktop. You can allow Merkato to do this automatically, or you can download it directly.

There are later versions of Java, but we do not recommend using them.

Automatic Download

If your browser doesn't have Java version 1.4.1, the first time you attempt to download a Desktop agent you are guided through the installation of the Java plug-in.

Reply "Yes" to all security permission requests.

Active Download

If you encounter problems, you can download Java free from Sun Microsystems. The URL is:

<http://java.sun.com/products/plugin/>

Note: Currently, to get the plug-in, you must download the entire Java Run Time Environment, which is greater than 5 MB. Netscape version 6.1 for Windows appears to support the Merkato Java applications, but it is not fully tested for Merkato version 2.3.

Desktop Interface

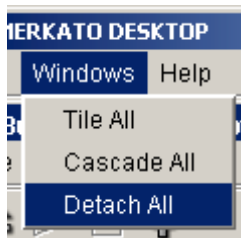
The Desktop interface allows you to control the selling agent and observe the auction in progress. Your agent is brought to your PC and bids on your behalf from the Merkato desktop.

Arranging Agent Windows

When more than one agent is displayed, you may want to arrange the agents on your screen.

Detached From the Merkato Desktop

By default, all agent windows appear within the Merkato Desktop window. If you want the agent window to appear as an independent window on your Windows screen, select *Detach All* from the Windows pull-down menu, as shown below:



When you select “Detach All” a check appears beside the name in the menu-bar. The agent can then be viewed outside of the Merkato desktop, as shown in Figure 9.

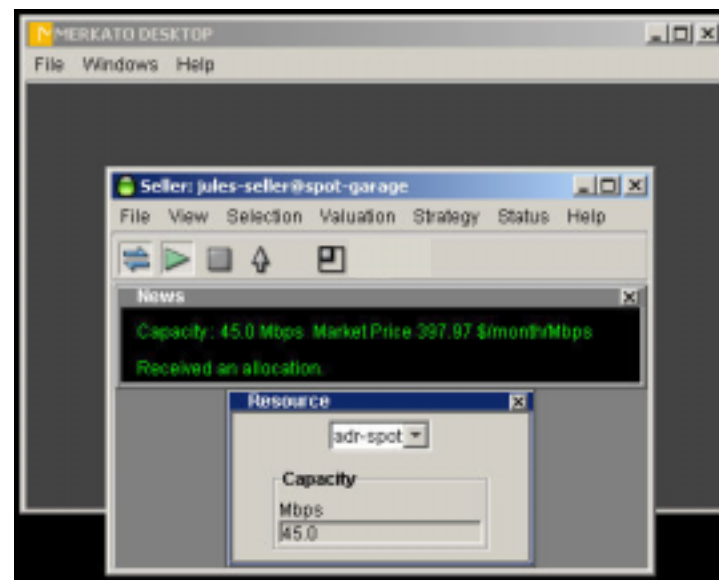


Figure 4: The Merkato Desktop Agent Detached From the Desktop

Tiled on Merkato Desktop

Whether the agent windows are inside or outside the Desktop, they can be automatically arranged relative to each other. If you select Tile All from the Windows pull-down menu, the windows arrange themselves in a tight, non-overlapping pattern. (See Figure 5.)

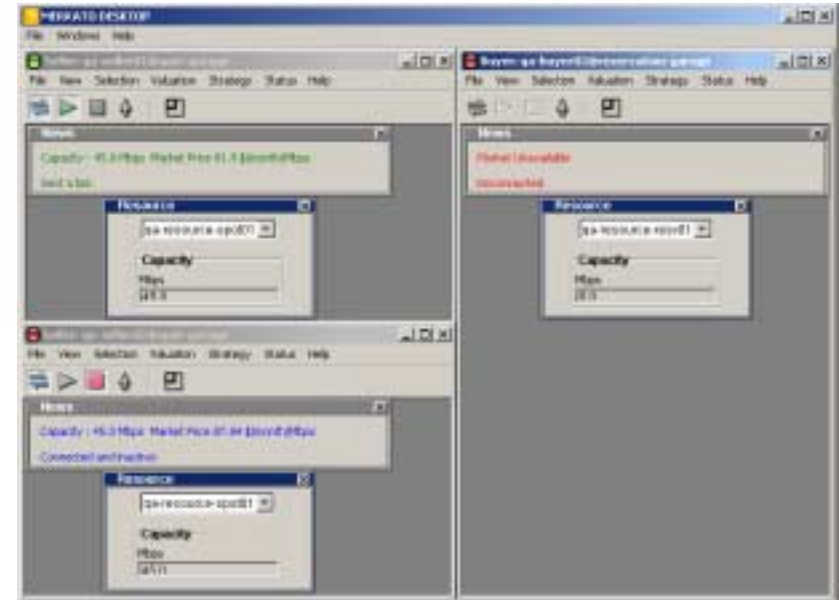


Figure 5: The Merkato Desktop with Multiple Agents Tiled

Cascading on Merkato Desktop

If you select Cascade All from the Windows pull-down menu, the windows are all offset from each other and overlap, as shown below.

The agent window consists of smaller sub-windows, each of which can be displayed or closed as necessary. By default, windows are stacked vertically in the agent window, as shown in Figure 6.

Each time a window is selected for viewing it appears beneath the previously activated window. To close a window, either click the close-box tab at the upper right hand corner of the window (X), or de-select the window in the pull-down menu-bar lists. The window closes and the other windows fill in the spot it occupied.

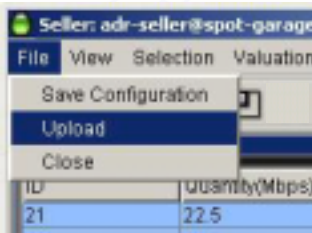
To resize the agent window to exactly match the windows displayed, click the Pack icon (☐) or select Pack from the View pull-down menu.




Figure 6: The Merkato Desktop With Multiple Agents Tiled

Upload Function

Your agent auctions bandwidth for you by processing bids from a Resource agent. You may have your agent work from the “Garage,” which normally runs on the same server as the Resource Agent. This is known as “uploading” your agent. When you upload your agent, all configuration changes are saved and the agent on your PC is closed.



You can upload either by pressing the up-arrow on the icon bar () or by selecting Upload from the File pull-down menu, as shown below:


Note: As a seller you must always upload your agent before exiting; otherwise the auction stops.

Connect/Start/Stop


The three icons at the upper left of the icon bar indicate “Connect,” “Start,” and “Stop” (from left to right).




When you first download a desktop agent, before you configure it to actively bid, it is disconnected from the auction. Your agent does not bid or allow you to view auction status when it is disconnected, although

you can change its configuration. When the connect symbol on the icon bar ( on the far left) is un-depressed and un-colored the agent is disconnected.

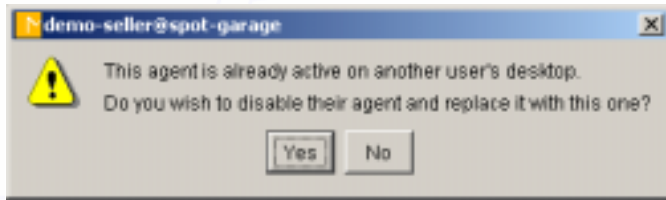
When you connect to the auction in “Stopped” mode you can observe the auction but you are not participating in it. The Auction Graph and Auction Table show you the current auction in progress and allow you to determine whether you wish to join it. (See *Merkato Seller’s Reference Manual*, “Monitoring Sales Via the Desktop Agent.”)

When you want to start the auction, either click the Start icon ( second from left on the icon bar) or select “Start” from the Selection menu. The auction starts immediately with the current active settings.

To stop selling, click the **Stop** icon ( third from left on icon bar) or select Stop from the Selection menu. Your agent stops selling and disconnects. To begin selling again you must reconnect. (Stopping during an auction disrupts the auction.)

Note that if more than one user has access to a Merkato username and password, it is possible that one bidder will be previewing an auction and be observing his own agent bidding on another user’s desktop. (If an auction is in progress when you, as a seller, access the desktop, this must be the case.) There is no way to know whether your agent is already active on another desktop when you preview an auction (although once you become adept at reading the auction table and auction canvas, it is possible to detect bidders who appear to have your valuation settings).

If you download an agent using a login that is already active on a desktop, you will be able to connect and view the auction in progress. But if you click **Start**, a message like the one below appears.



If you click **Yes**, your agent takes control of the bidding process. The other agent, on another user's desktop, is disconnected.

If two users attempt to activate the same login ID, the last user to log in successfully "wins" control of the agent. The previous user receives an error message, shown below:



Bumping

Bumping occurs when a user logs in using the ID and password of an active agent. The user who is logging-in sees a message that an agent with this login ID is already active. If the new user proceeds, the old user is disconnected. The agent uses the saved configuration of the agent, not necessarily the configuration of the previously active agent.

Bumping of an active Seller agent is very disruptive to the auction. You should not do this without good reason.

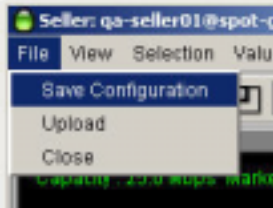
The agent that takes control of an existing agent by bumping it starts with the profile stored in the garage. Any changes made to the profile of the bumped agent that are not saved are lost.

Saving Your Configuration

Saving your configuration preserves changes to your agent's profile in the garage while you continue to keep you agent on your desktop. It also saves the arrangement of windows in the desktop agent so they re-appear the same way the next time you download the agent. While the agent is active on your desktop

there is a non-active profile of the agent's parameters stored in the garage, which should be kept up to date.

If you want to save the way your agent is configured, but not upload your agent to the garage, from the File menu, select Save Configuration:



A confirmation box appears (Figure 7), showing which garage your configuration will be saved to.

Click **OK** to confirm saving your configuration.

Exiting Without Saving

In rare instances, you will want to close your desktop window without updating your agent's profile in the garage. Do this by selecting "Close" rather than "Upload" in the menu-bar list. When you do this, the following confirmation screen appears (Figure 8).

If you click **Yes**, your agent is uploaded to the garage before it is closed on the desktop.

If you click **No**, your agent stops and closes without any changes saved.

If you click **Cancel**, you return to your Desktop agent; the agent is neither uploaded nor closed.

Exiting the Merkato Desktop

To exit the Merkato desktop, click the close-box tab at the upper right hand corner of the window (☒), or select "Exit Merkato" from the File pull-down menu.

You can leave the portal page in the browser open to monitor your agent status (via the Express agent), or click Logout in the list on the left side of the window.

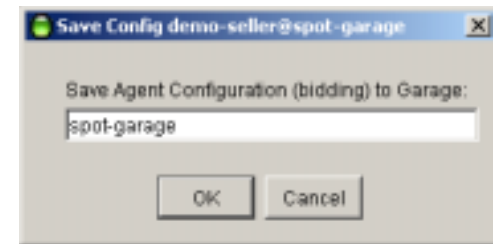


Figure 7: The Save Configuration Confirmation Box

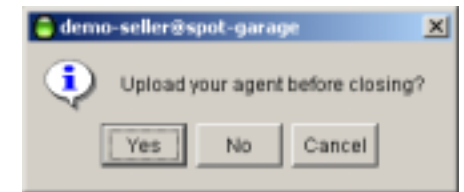


Figure 8: The Exit Confirmation Box

Spot Market vs. Reservation Market

There are two ways to buy and sell bandwidth in a Merkato system:

- Buyers can request and receive a fixed amount of bandwidth, at a fixed price, for a fixed term
- Buyers can create a bidding strategy and contend for bandwidth with other buyers at frequent intervals

The first method is called the Reservation market and uses a reservation agent. The second method is called the Spot market and uses a spot agent.

There is no Reservation agent for sellers (although you may wish to have your Merkato administrator create a buy-back reservation agent for you, as a convenient way of adding bandwidth to the marketplace). All reservation settings are pre-configured in the Merkato server.

Spot Market Agent Window

The agent window consists of multiple sub-windows that may be displayed, or not displayed, as you wish. The sections below describe the windows, in roughly the order a first-time user might access them.

Resource Window

The Resource window includes a pull-down menu of the resources available to the agent. It shows the amount of bandwidth available at the resource selected.

Open the Resource window by selecting it from the Selection pull-down menu. (See Figure 9.) If the agent is connected to a resource, the Capacity display indicates the total amount of bandwidth available for sale (as set by this Seller agent) at that resource.

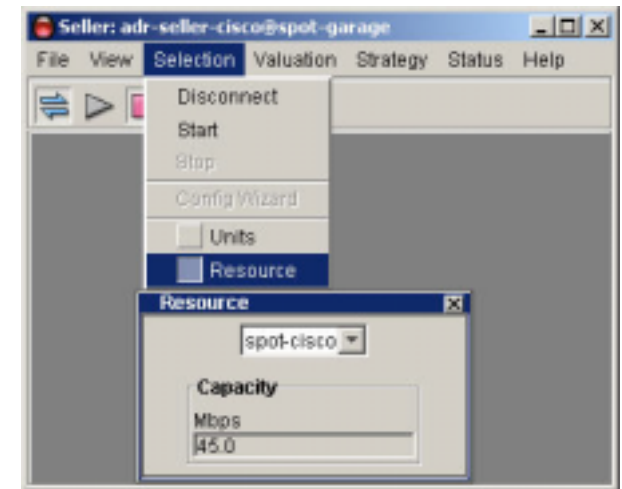


Figure 9. The Resource Window

Units Window

The Units window lets you select units in which currency, bandwidth, and time are displayed. (See Figure 10.) Units you select appear in the labels of windows as well as in pop-up message screens.

Currency units supported are dollars (“\$”) and cents (“c”).

Quantity units supported are kilobits per second (“Kbps”), Megabits per second (“Mbps”), and gigabits per second (“Gbps”).

Time units supported are minutes (“min”), hours (“h”), days (“day”), and months (“month”).

Changes you make to any units take effect immediately. All values in all other windows are automatically scaled to reflect the new units without changing the actual values.

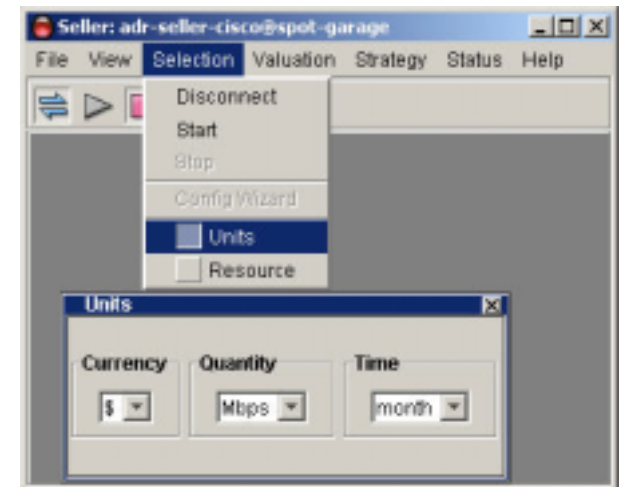


Figure 10. The Units Window

Seller Valuation

The seller valuation sets the quantity of bandwidth available in the marketplace and the minimum price at which it will be sold. (See Figure 11.) In an auction, your agent uses this valuation information to determine what quantity can be sold and the floor price.

Select the “Current Valuation” radio button label at the top of the window. The valuation must be active for the Seller valuation to take effect. The green arrow in the menu indicates the valuation that is currently active.

Two configurable parameters determine the auction parameters.

- “Qty” sets the current quantity for sale in the marketplace. Your agent offers all this bandwidth to the buyers.
- “Value” sets the minimum amount of money you are willing to accept for the entire “Qty” of bandwidth—the floor price.

Changes to either one of these values take effect immediately.

Floor Price

The floor price is usually referred to as a price per unit of bandwidth. The per-unit floor price is the “Value” divided by the total “Qty.” In the illustration (Figure 11), the floor price is 45 Mbps/\$4,500 per month, or \$100 per Mbps per month.

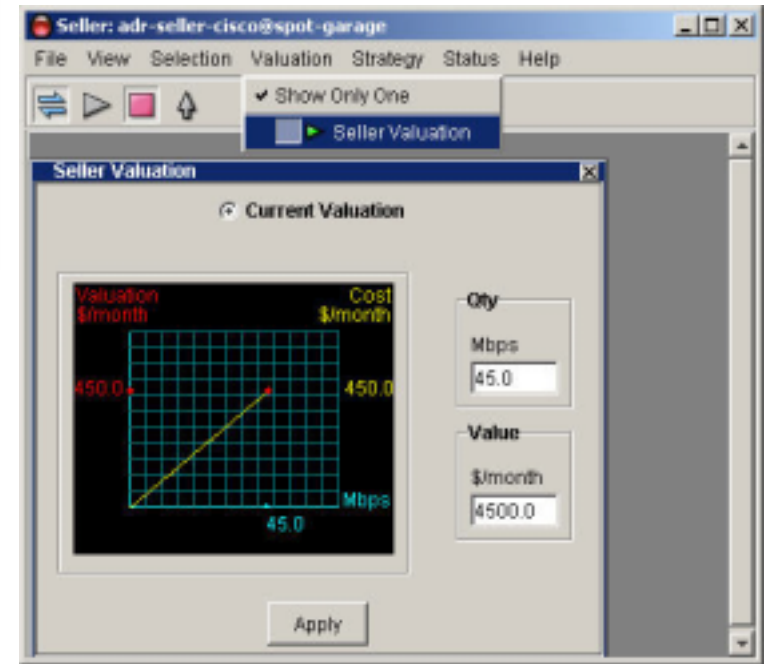


Figure 11. The Seller Valuation Window

Under certain conditions you may not sell all the bandwidth you are offering. These conditions include the following:

- You have bought back bandwidth because of insufficient demand at the floor price using your own buy-back agent.
- All buyers have specified a maximum amount of bandwidth that they wish to buy, independent of the price, and the total amount is less than the amount you are offering.

If the first condition occurs, you should consider whether your floor price is too high. If the second condition occurs you need to find more buyers. If demand on the Reservation market is high and the second condition occurs, you might consider having your Merkato administrator transfer bandwidth from the spot market to the Reservation market.

Seller Strategy

Once you have configured your valuation you must configure how your agent uses that information. There are two strategy options you can use in the Spot market: static and dynamic. (See “Spot Market Selling Strategy” on page 11.) A green arrow in the menu-bar list indicates the currently active strategy. (See Figure 12.)

Static Seller Strategy

Note: We do not recommend using the Static Seller strategy.

Under static seller pricing, the unit price for all bidders becomes the unit price of the last unit of bandwidth sold. Merkato sets this price via a Seller bidder that always loses at the highest possible price (and for the highest possible quantity) under the Progressive Second Price auction rules. (See “Merkato Auction Mechanism: The Progressive Second Price Auction” on page 46.)

With a Static strategy, the basis of all bidders’ pricing is nearly always the floor price as offered by the seller’s buy-back agent.

Dynamic Seller Strategy

Under Dynamic seller pricing, the unit price for all bidders is the highest losing bid. Merkato gets this bid via a Seller bidder that always loses at the highest possible price (bid for the highest possible quantity) under the Progressive Second Price auction rules.

With a Static strategy, the basis of all bidders’ pricing is nearly always the floor price, as offered by the seller’s buy-back agent. In the Dynamic strategy, the seller receives the price offered by the lowest successful bidder times the quantity of bandwidth for sale. (See Figure 13.)

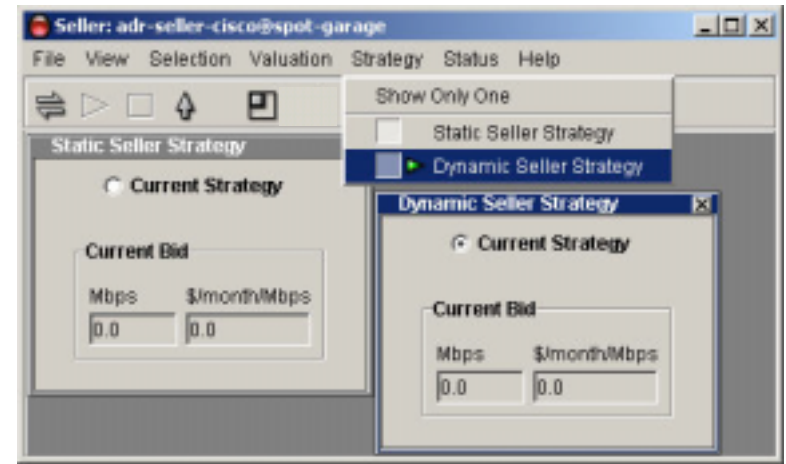


Figure 12. The Strategy Menu

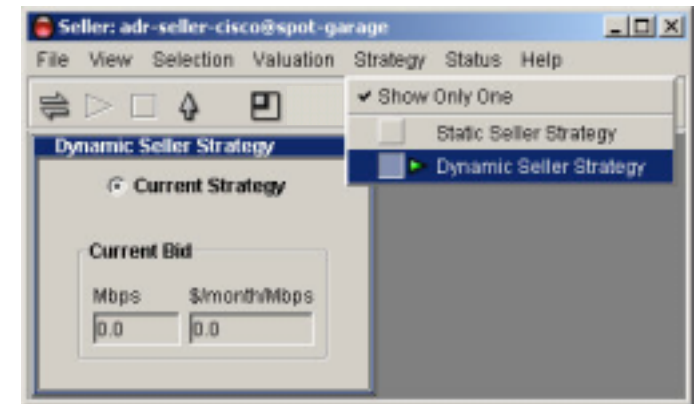


Figure 13. The Dynamic Strategy Window

News Windows

The News window provides real-time status of your Merkato agent. It is useful when you want to get a quick summary of marketplace conditions and confirm the state of your agent.

When your agent is disconnected, you see a News display like the one shown in Figure 14.

When you are connected but not bidding the News window looks like the one shown below. It indicates the bandwidth capacity you are offered the current market price of the auction as it is proceeding.



During an auction, the News window looks like the examples below.

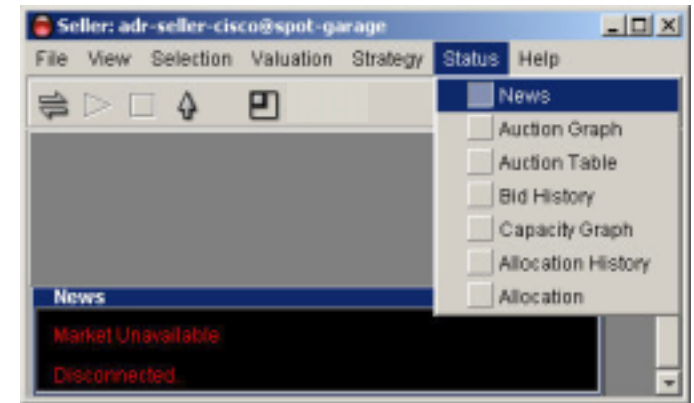
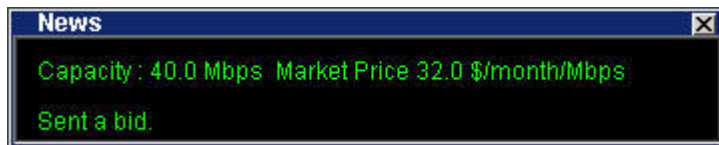


Figure 14. The News Window

Allocation Window

For buyers, the Allocation window provides information as to the quantity of bandwidth received and the price paid for that bandwidth at the close of each auction. For sellers (who are not allocated bandwidth) the allocation display is always zero.

The allocation window also provides a time indicator for the current auction. Each time a buyer places a bid the timer is set back. (The administrator can change this time setting.) When all bidders cease bidding (either because they are satisfied with their proposed allocation or have dropped out of the bidding at the current market price), the timer counts down to zero. At that point the Resource agent turns the proposed allocations into real allocations. This occurs during a “pause” period between auction rounds. The pause period is indicated by several changes to the time display: the timer label changes from “Time Left” to “Time Since,” the numbers are shown in red, and the counter counts up to the pause duration limit.

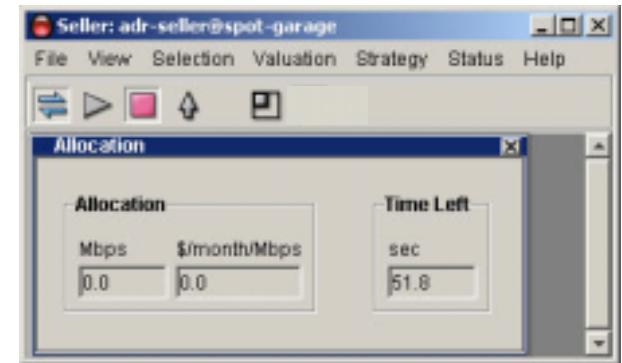


Figure 15. The Allocation Window

Auction Graph

The Auction graph indicates the state of the Merkato auction in progress. Dots indicate current bids. A Buyer agent displays a graph like the one in Figure 16.

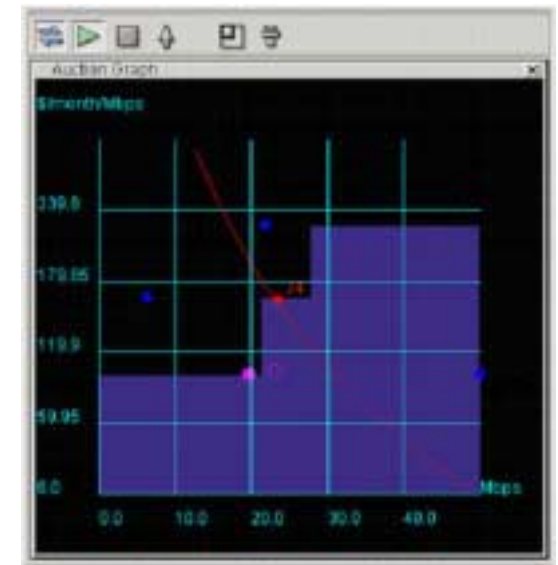


Figure 16. The Action Graph Seen by a Buyer

For Seller agents, the Auction graph displays fewer elements because allocations are not being made to the seller agent. The display looks like the one shown in Figure 17.

The graph elements are:

- Y-axis, indicating unit price (the axis scales automatically to show all current bids).
- X-axis, indicating quantity (the axis scales automatically to show available quantity).
- Blue dots representing the last bids received from all bidders (excluding a bid by the dynamic seller agent, which is shown as a red dot).
- A red dot, with ID number, indicating the last bid from the agent currently logged in.
- A red line, representing your agent's valuation curve, as determined by settings in your active valuation window.
- A blue shaded area, representing the allocation of bandwidth to successful bidders, except for any bandwidth allocated to your agent. This area is a series of blocks, where the height represents the unit price paid and the width represents the quantity allocated. (The bandwidth allocated to your agent is the amount of blue under and to the left of the red dot.)

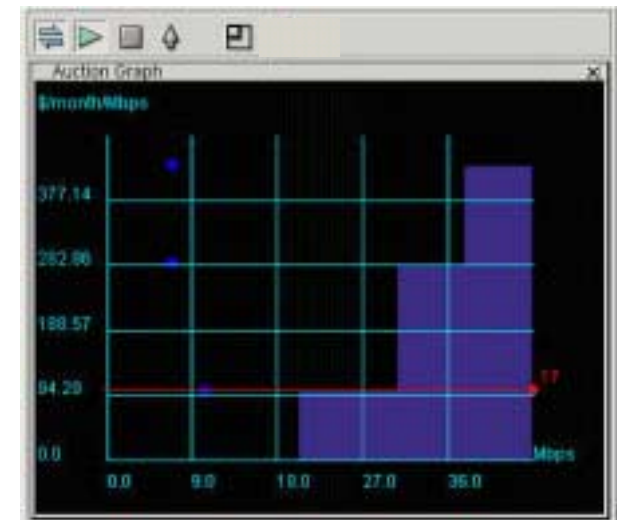


Figure 17. The Auction Graph Seen by a Seller

Auction Table

The Auction Table gives you a quantitative, real-time view of the auction in progress. (See Figure 18.)

The columns of the Auction Table may be resized (by clicking and dragging column boundaries in the header) or reordered (by clicking and dragging the column headers themselves).

The columns in the auction table are as follows:

- ID** The ID of the bidder. A bidder agent receives a new ID whenever it is uploaded to or downloaded from the garage.
- Quantity** The quantity requested in the last bid placed (note that this may be split into multiple rows, as explained below).
- Price** The price (unit price) requested in the last bid placed.
- Rate** The rate represented by the bid. This is the quantity times the price. It is the maximum cost represented by that bid, but not necessarily the price that will be paid by the bidder.

The rows in the Auction Table can be identified as follows:

Your Dynamic Seller agent's last bid is shown in red. The last bids of all other agents are shown in black (in the blue shaded area) or in yellow (in the black area).

Any bidders that receive an allocation of bandwidth (possibly including your Dynamic Seller agent) are shown in the blue shaded area. Any bidder who does not receive an allocation is shown in the black area. Bidders that receive only a portion of an allocation are shown in both the blue and black areas, with their quantities split accordingly.

Bidders currently unsuccessful in the auction round are shown in the black area. The price offered by unsuccessful bidders forms the basis for the actual cost paid by your agent if it receives an allocation. (See "Merkato Auction Mechanism: The Progressive Second Price Auction" on page 46.) If your Dynamic Seller agent is currently winning an auction round, the rates shown in the yellow-shaded area are the rates offered by bidders whose combined rate offers your offer has exceeded. These bidders are thus pushed out by your agent's bid.

The columns of the Auction Table may be resized (by clicking and dragging column boundaries in the header) or reordered (by clicking and dragging the column headers themselves).

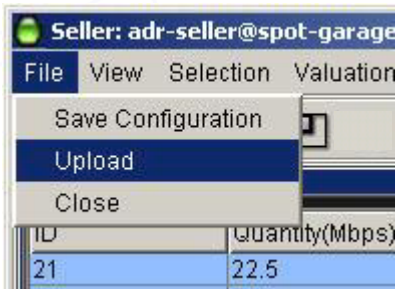
ID	Quantity(Mbps)	Price(\$/month/...	Rate(\$/month)
19	7.0	428.57	3000.0
21	7.0	285.71	2000.0
12	31.0	23.33	723.33
12	14.0	23.33	326.67
17	45.0	23.33	1050.0
			0.0


Figure 18. The Auction Table

Saving, Uploading, and Exiting

When you are done monitoring your agent on your desktop, you can upload your agent to the garage—the Merkato server where, using an Automatic bidding strategy, it can continue bidding on your behalf.

Upload and Exit



Use either the up-arrow in the icon bar () , or the “Upload” menu-bar item in the “File” category to access this function. You will be presented with a confirmation query message, as shown in **Error!**

Reference source not found..

The confirmation message indicates the garage into which your agent will move. If you click **Cancel** you are returned to the Desktop agent and no upload occurs.

The confirmation message indicates whether your agent is bidding. If your agent is stopped when you upload it, the confirmation message looks like the one in Figure 19. (Notice the red circle in the upper left.)

Make sure the bidding state is Active if you wish to continue bidding while your agent is in the garage.

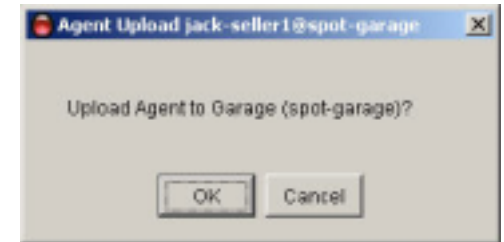
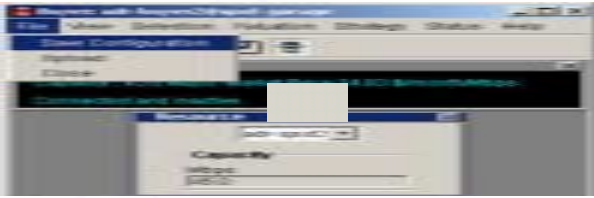


Figure 19. The Upload Confirmation Box When The Agent is Not Active

Saving Your Configuration



Saving your configuration preserves changes to your agent's profile in the garage while you continue to keep you agent on your desktop. While the agent is active on your desktop there is a non-active profile of the agent's parameters stored in the garage, which should be kept up do date.

If you don't keep this configuration up to date you could lose the changes you made to your agent while bidding on your desktop if your desktop is rebooted or crashes before you can upload.

For procedures, see "Saving Your Configuration" in *Seller's Guide to Using Merkatobid*.

Reservation Parameters

The Merkato administrator configures your reservation parameters. This section describes the reservation Buyer agent. As a seller, if you buy reservations from your own resources, you are removing bandwidth from the Reservation market. (Ask your Merkato Administrator to set up a Reservation Buyer agent for you.)

Buying a reservation gives you guaranteed bandwidth for a set period of time at a fixed price. The pricing is set by the seller and depends on both the quantity of bandwidth desired and the duration of the reservation. There is no additional fee.

Reservations can start and end at any time in the future, as long as you (the seller) provide capacity to the Reservation market.

Buyers can cancel reservations any time prior to expiration, but there may be a cancellation fee based on a percentage of the remaining value of the reservation. The fee may be anywhere between 0% (no penalty for cancellation) to 100% (no refund on cancellation). Merkato quotes the cancellation percentage to you when you make the reservation and again when the cancellation is confirmed. (See “Percentage Fee” on page 10.)

For complete details about reserving bandwidth, see *User Reference for Buyers*, “Reservation Agent.”

The Merkato Portal

The portal is the access point to the agent applications, as well as to obtain historical allocation and billing information. When you open the Merkato web site, the screen looks similar to the one show below:

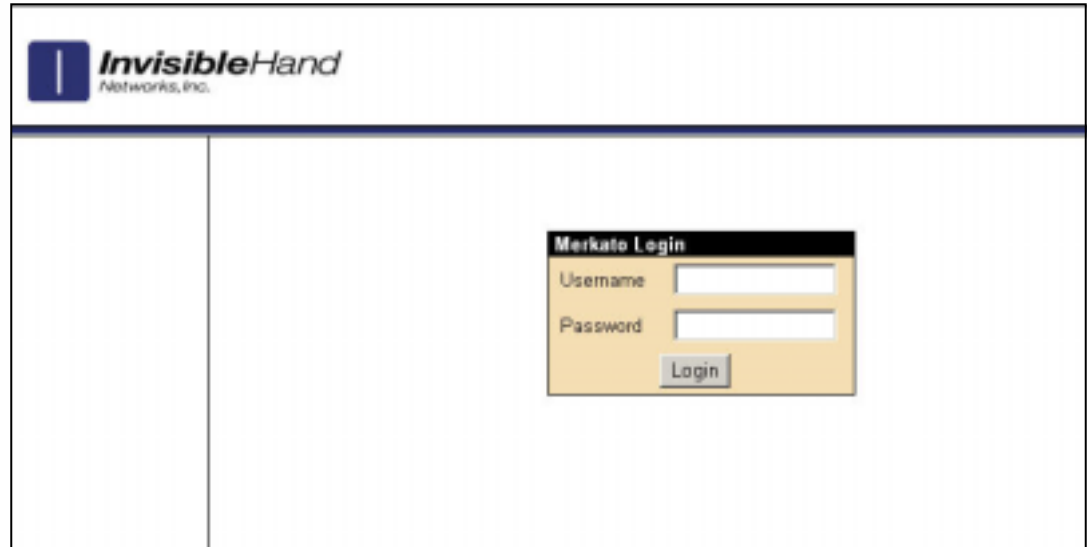


Figure 20: The Merkato Login Screen

If you enter a valid username and password, and click **Login**, the top-level agent screen appears (Figure 21).

The central portion of the window shows the Merkato agent. This is the Express agent interface. Below this area, visible when you scroll down, is the Recent Allocations table.

To the right, the pull-down user interface menu allows you to select the Wizard, Express, or Desktop interface.

The left portion of the page contains a navigation bar, which provides access to all pages of the portal:

- **Agent** – (The page shown) contains the Express agent interface, the Agent selection pull-down, the allocation table, and market price graphs.
- **Traffic** – Contains Monthly, Weekly, and Daily graphs of traffic associated with the logged in user.
- **Charge Reports** – Provides a billing interface where you can create customer queries into the Merkato database, and automatically provides account balance information.
- **Transaction Data** – Opens a billing interface that allows you to create queries into the Merkato database, providing a history of transactions for your account. The data is downloaded as a spreadsheet file.
- **Version Information** – For the Merkato and the database it uses, provides the version, build date, and installation date. Use this information to determine what version of Merkato you are using.
- **Logout** – Returns you to the login screen.

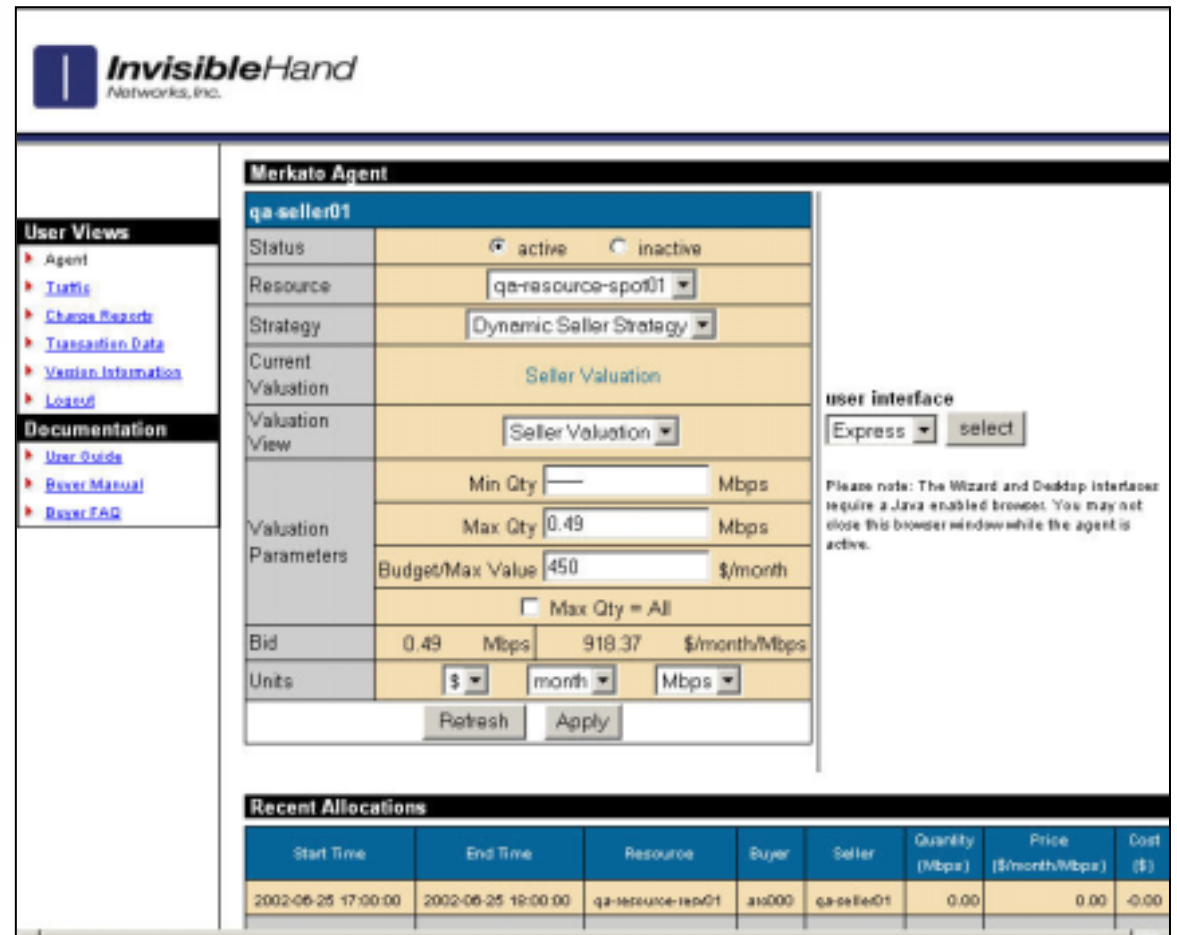


Figure 21. The Express Interface in the Merkato Portal

Express Agent View

The Interface pull-down menu on the left of the portal provides access to the Wizard and Desktop agent applications, described in their own sections of this document. The other areas of the Merkato Portal are described below.

Express Agent Interface

The Express agent interface is an HTML version of the Desktop interface, with some significant differences:

- It leaves your agent in the Merkato garage (on the Merkato server) and controls it remotely. Closing the browser window does not affect your agent in the garage.
- Your screen is not updated in real time. You must click **Refresh** to update information in status displays.
- Reservations and certain spot market settings are not supported in this interface (significant for Buyer agents, but not for Seller agents).

The Express agent interface is meant to support advanced users who know what the fields mean and want a quick way to check agent status or make configuration changes.

The Express agent is in the initial Portal browser window when you open it. If your agent was active and bidding from the garage, when you open the Merkato Portal, you see a window like the one in Figure 22.

The two buttons at the bottom of the window control your communication with the Resource agent.

- **Refresh** – Updates the read-only status fields in the windows without sending configuration changes to the Resource agent.
- **Apply** – Sends configuration changes to the Resource agent.

The fields in this display are as follows (identified by their row titles):

- **Status** – Controls whether the agent is bidding or not. You must click **Apply** for this change to take effect.
- **Resource** – Selection list for the resource for which you are bidding. After you select a resource from this list you must click **Apply** for this selection to take effect.

Merkato Agent	
qa-seller01	
Status	<input checked="" type="radio"/> active <input type="radio"/> inactive
Resource	qa-resource-spot01
Strategy	Dynamic Seller Strategy
Current Valuation	Seller Valuation
Valuation View	Seller Valuation
Valuation Parameters	Min Qty <input type="text"/> Mbps
	Max Qty <input type="text" value="0.49"/> Mbps
	Budget/Max Value <input type="text" value="450"/> \$/month
	<input type="checkbox"/> Max Qty = All
Bid	0.49 Mbps 1111.59 \$/month/Mbps
Units	<input type="text" value="\$"/> <input type="text" value="month"/> <input type="text" value="Mbps"/>
<input type="button" value="Refresh"/> <input type="button" value="Apply"/>	

Figure 22. The Express Agent Interface

- **Strategy** – Dynamic Seller Strategy is the only strategy supported by the Express interface. You can select Static strategy but there are no bid entry fields. Creation and cancellation of reservations are not currently supported (although you may view confirmed reservations in the allocation list below the Express interface in the Portal).
- **Current Valuation** –Displays the valuation currently in use. (This field is not editable.)
- **Valuation View** – Seller Valuation is the only valuation supported for Seller agents.
- **Valuation Parameters** – This set of fields contains the settings for the Seller valuation. You must click **Apply** after you make changes for them to take effect. A horizontal bar appears in the Min Qty field, indicating that the field is not used.
- **Bid** – A read-only display indicating the latest bid submitted. You must click **Refresh** to update this display. If this display is missing your agent is inactive and not bidding.
- **Allocation** – A read-only display indicating the allocation received in the last round. You must click **Refresh** to update this display. If this display is missing, either your agent is inactive or it did not receive an allocation in the previous auction round.
- **Units** – The selections made in this window change the units displayed for all other windows. All values are scaled to reflect the change in units. You need not click **Apply** for this change to take effect.

When your agent is inactive and there is no auction in progress, the Bid row is missing, as shown in Figure 23.

qa-seller01	
Status	<input type="radio"/> active <input checked="" type="radio"/> inactive
Resource	qa-resource-spot01
Strategy	Dynamic Seller Strategy
Current Valuation	Seller Valuation
Valuation View	Seller Valuation
Valuation Parameters	Min Qty <input type="text"/> Mbps
	Max Qty <input type="text" value="40"/> Mbps
	Budget/Max Value <input type="text" value="600"/> \$/month
	<input type="checkbox"/> Max Qty = All
Units	\$ month Mbps
<input type="button" value="Refresh"/> <input type="button" value="Apply"/>	

Figure 23. The Express Interface in the Inactive State

Recent Allocations Table

When you log in as a Seller, the Recent Allocations table indicates recent reservations and allocations made to buyers of the bandwidth you have offered. (See Figure 24.) Each Spot market allocation entry contains a summary of information for an hour, which generally includes many auction rounds.

The Recent Allocations table gives you an indication that Merkato auctions have been providing the desired results. You may obtain information for longer timeframes in the Charge Reports page, via custom queries.

Recent Allocations							
Start Time	End Time	Resource	Buyer	Seller	Quantity (Mbps)	Price (\$/month/Mbps)	Cost (\$)
2002-07-01 21:00:00	2002-07-01 22:00:00	qa-resource-spot01	aro000	qa-seller01	0.02	64.28	0.00
2002-07-01 21:00:00	2002-07-01 22:00:00	qa-resource-spot01	aburdin	qa-seller01	0.02	64.28	0.00
2002-07-01 21:00:00	2002-07-01 22:00:00	qa-resource-spot01	akim00	qa-seller01	0.02	64.28	0.00
2002-07-01 21:00:00	2002-07-01 22:00:00	qa-resource-spot01	arthur	qa-seller01	0.22	34.04	0.01
2002-07-01 21:00:00	2002-07-01 22:00:00	qa-resource-spot01	andi00	qa-seller01	0.02	64.28	0.00
2002-07-01 20:00:00	2002-07-01 21:00:00	qa-resource-spot01	aro000	qa-seller01	0.39	54.29	0.03
2002-07-01 20:00:00	2002-07-01 21:00:00	qa-resource-spot01	arthur	qa-seller01	0.47	53.74	0.04
2002-07-01 20:00:00	2002-07-01 21:00:00	qa-resource-spot01	aburdin	qa-seller01	0.39	54.29	0.03
2002-07-01 20:00:00	2002-07-01 21:00:00	qa-resource-spot01	akim00	qa-seller01	0.39	54.29	0.03
2002-07-01 20:00:00	2002-07-01 21:00:00	qa-resource-spot01	andi00	qa-seller01	0.39	54.29	0.03

Figure 24. The Recent Allocations Table

Traffic View

The Traffic screen shows the amount of data transferred through the Merkato network control point by the data stream associated with this user. (See Figure 25.) Three graphs are provided to represent three time scales:

- Daily graph showing 5 minute granularity
- Weekly graph showing 30 minute granularity
- Monthly graph showing 2 hour granularity

All three graphs are based on five-minute sample intervals. Merkato gathers traffic statistics in both the input and output direction every five minutes. Each five-minute sample represents an average over that five-minute period. In other words, peaks lasting less than five minutes are not accurately represented in the graphs. These five-minute samples are used directly in the daily graph. These values are then combined into longer averages for the weekly and monthly graphs.

The **green**-shaded portion of the graph indicates traffic in the incoming direction.

Traffic indicated by the **blue** line is in the outgoing direction.

The **purple** and **pink** lines show data associated with specific resources.

The caption below each chart indicates the IP address of the network control point for which traffic is measured. For each flow, the caption lists the average traffic level and maximum measured traffic level for the traffic measured in the time window represented by the graph (daily, weekly, or monthly).

If you are unclear which direction corresponds to “in” and “out” with respect to your application, contact your Merkato administrator.

Use the traffic graphs to help you decide how much bandwidth to purchase on the Spot and reservations markets. Often, the best buying strategy is to buy what you normally use on the Reservation market, then purchase additional bandwidth during periods of burst demand on the Spot market. The traffic graphs can also help you select the maximum desired quantity for your Spot market valuation settings. There is little performance improvement, but potentially significant cost impact, when you purchase more bandwidth than you need.

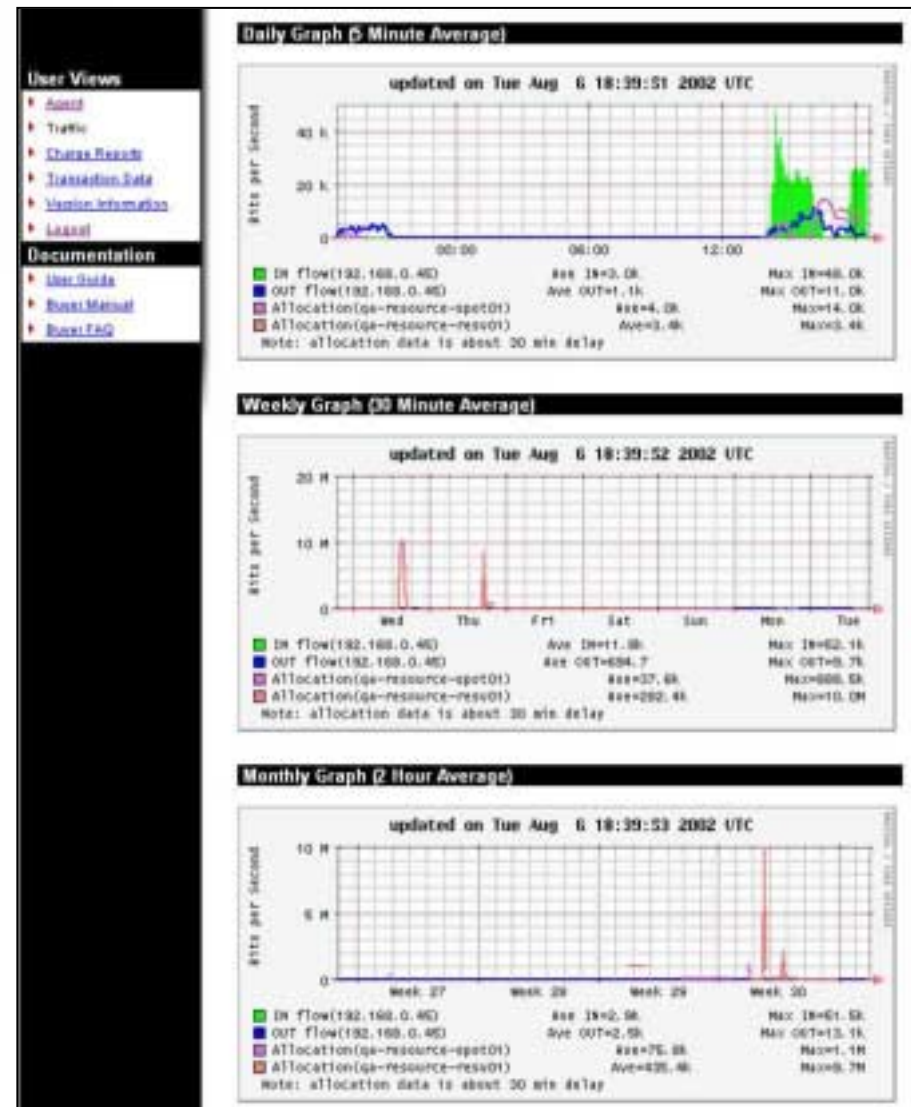


Figure 25. The Traffic View

Charge Reports

Merkato's Charge Reports feature lets you see details about bandwidth allocations you have made. (See Figure 26.) The reports it generates provide all the details about each allocation, but they are also summarized.

Figure 27 is an example of a charge report for one day:

The following fields appear in the Charge Reports screen:

From (mm/dd/yyyy): Enter the date you want the charge report to begin on.

To (mm/dd/yyyy): Enter the date you want the charge report to end on.

Resource: (optional) Enter the resource (from whom you have received allocations) whose charges to you will appear in the report.

Ordered By Day: A report ordered by day lists charge information from all resources grouped by day.

Ordered By Resource: A report ordered by resource lists charge information from each individually. If you select Detailed, within the listing for each resource, charges are ordered by day.


Summary: All charges are grouped and totaled.

Detailed: Charges are displayed separately by individual resource.

Figure 26: The Charge Report Selection Form

Charge Report - jbossom							
Generate New Report							
Details for 2002/06/13							
Resource	Seller	Buyer	Date	Quantity (Mbps)	Price (\$/month/Mbps)	Allocation Time (hours)	Cost (\$)
qa-resource-spa01	qa-seller01	jbossom	2002/06/13	0.00	0.00	24.00	0.00
Summary for 2002/06/13							
Resource	Seller	Buyer	Date	Quantity (Mbps)	Price (\$/month/Mbps)	Allocation Time (hours)	Cost (\$)
qa-resource-spa01	qa-seller01	jbossom	2002/06/13	0.00	0.00	24.00	0.00

Figure 27: Sample Charge Report

Rather than enter start and end dates manually, you can click on the small calendar icon () to the right of the field. A pop-up calendar (Figure 28) appears from which you can select your start and end dates.

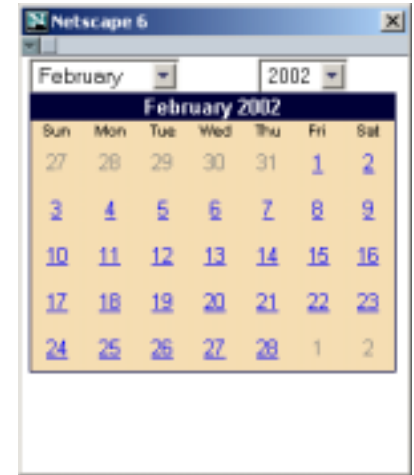


Figure 28: The Pop-Up Calendar

Similarly, when you click the pie-chart icon () , a list box (Figure 29) appears from which you can select a resource on which to do billing queries.



Figure 29: The Pop-Up Resource Selection Window

For example, if you select **Ordered by Day** and **Summary** and click **Generate Report** the view looks like Figure 30.

Details for 2002/06/14							
Resource	Seller	Buyer	Date	Quantity (Mbps)	Price (\$/month/Mbps)	Allocation Time (hours)	Cost (\$)
qa-resource-spa01	qa-seller01	jbossom	2002/06/14	0.00	2759.93	24.00	0.01

Summary for 2002/06/14					
Buyer	Date	Quantity (Mbps)	Price (\$/month/Mbps)	Allocation Time (hours)	Cost (\$)
jbossom	2002/06/14	0.00	2759.93	24.00	0.01

Details for 2002/06/15							
Resource	Seller	Buyer	Date	Quantity (Mbps)	Price (\$/month/Mbps)	Allocation Time (hours)	Cost (\$)
qa-resource-spa01	qa-seller01	jbossom	2002/06/15	0.01	3488.55	24.00	1.10

Summary for 2002/06/15					
Buyer	Date	Quantity (Mbps)	Price (\$/month/Mbps)	Allocation Time (hours)	Cost (\$)
jbossom	2002/06/15	0.01	3488.55	24.00	1.10

Figure 30: Sample Charge Report Set to Summary, Ordered By Day

If you select Ordered by Resource and Detailed and click **Generate Report**, the view looks like the report shown in Figure 31:

Transaction Data

The Transaction Data view allows you to query the accounting database. (See Figure 32.) When you click **Generate Data**, Merkato lets you download a file to your PC. You can read this file in a spreadsheet program or as plain text.

The following fields appear in the Transaction Data screen:

From (mm/dd/yyyy): Enter the date you want the charge report to begin on.

Resource: (optional) Enter the resource (from whom you have received allocations) whose charges to you will appear in the report.

Ordered By Time: A report ordered by day lists charge information from all resources separated by each day.

Ordered By Resource: A report ordered by resource lists charge information from each individually.

The seven columns are the following:

- Start time of the allocation (mm/dd/yyyy hh:mm)
- End time of the allocation (mm/dd/yyyy hh:mm)
- Buyer Agent
- Resource (the source of the bandwidth that was sold)
- Quantity
- Price (in dollars per month per Mbps)
- Cost of allocation (in dollars—the amount billed)

Details for qa-resource-resv01							
Resource	Seller	Buyer	Date	Quantity (Mbps)	Price (\$/month/Mbps)	Allocation Time (hours)	Cost (\$)
qa-resource-resv01	qa-seller01	ard000	2002/06/11	0.00	300.00	24.00	0.02
qa-resource-resv01	qa-seller01	julia0	2002/06/11	0.00	300.00	24.00	0.01
qa-resource-resv01	qa-seller01	ard000	2002/06/12	0.00	299.93	24.00	0.01
qa-resource-resv01	qa-seller01	ard000	2002/06/15	0.02	272.80	24.00	0.14
qa-resource-resv01	qa-seller01	julia0	2002/06/15	0.00	299.95	24.00	0.03
qa-resource-resv01	qa-seller01	ard000	2002/06/16	0.10	282.99	24.00	0.88
qa-resource-resv01	qa-seller01	jboss00	2002/06/16	0.04	299.09	24.00	0.38
qa-resource-resv01	qa-seller01	ard000	2002/06/17	0.09	282.99	24.00	0.78
qa-resource-resv01	qa-seller01	ard000	2002/06/18	0.01	239.36	24.00	0.11
qa-resource-resv01	qa-seller01	julia0	2002/06/18	0.02	297.16	24.00	0.18
qa-resource-resv01	qa-seller01	ard000	2002/06/19	0.06	239.36	24.00	0.47
qa-resource-resv01	qa-seller01	julia0	2002/06/19	0.01	271.61	24.00	0.07
qa-resource-resv01	qa-seller01	ard000	2002/06/20	0.00	300.53	24.00	0.00
qa-resource-resv01	qa-seller01	julia0	2002/06/20	0.00	274.13	24.00	0.01
qa-resource-resv01	qa-seller01	ard000	2002/06/21	0.02	278.10	24.00	0.21
qa-resource-resv01	qa-seller01	ard000	2002/06/22	0.12	275.93	24.00	1.09
qa-resource-resv01	qa-seller01	ard000	2002/06/23	0.28	275.93	24.00	2.61
qa-resource-resv01	qa-seller01	ard000	2002/06/24	0.00	0.00	24.00	-0.00
qa-resource-resv01	qa-seller01	ard000	2002/06/25	0.00	0.00	24.00	-0.00

Summary for qa-resource-resv01					
Resource	Seller	Quantity (Mbps)	Price (\$/month/Mbps)	Allocation Time (hours)	Cost (\$)
qa-resource-resv01	qa-seller01	0.04	249.94	456.00	8.97

Figure 31: Sample Charge Report Set to Detailed, Ordered By Resource

Merkato Auction Mechanism: The Progressive Second Price Auction

The patent-pending Progressive Second Price (PSP) auction mechanism creates the rules by which the Merkato marketplace operates. PSP provides the following:

- The ability to divide resources among many bidders
- The ability to react quickly to changes in a buyer's valuation of the resource for sale
- The ability to support any number of bidders who may enter or leave the bidding at will
- Establishment of market mechanisms that create a fair market price for bandwidth, benefiting both buyers and sellers
- Creation of market rules that encourage bidders to be maximally truthful about their desires and willingness to pay, allowing auctions to converge quickly

These things are true of the PSP auction mechanism generally. The Merkato implementation of PSP provides the following additional features:

- A lightweight protocol between bidders and the Resource agent that does not contribute significantly to bandwidth utilization
- Support for multiple buyers and one seller, as well as multiple sellers and one buyer
- Ability of customers to create automated bidding decisions based on mix-and-match combinations of valuation, budget, and strategy
- Inclusion of sellers as bidders, allowing them to take bandwidth off the market if they do not feel the price is sufficiently high

To understand the Progressive Second Price auction mechanism, it helps to consider the rules and motivations behind a non-progressive simple second price auction.

In a simple auction, where the highest bidder pays the price bid, bidders raise their bids incrementally in successive bidding rounds. Each bidder's goal is to obtain the item at the lowest price, which is the price just above what the second-place bidder is willing to pay.

In an auction where many buyers are bidding for a single item, a second price mechanism is a way to quickly decide the highest bidder without the normal incremental bidding process.

The *second price rule* does away with the incremental bidding—but achieves the same result—by asking

each bidder for a single highest bid. The highest bidder wins the item, but pays the amount offered by the second highest bidder. The highest bidder is guaranteed not to overpay for the item by revealing the maximum he or she is willing to pay. The seller gets essentially the same price for the item had the bidding price been raised incrementally until the second place buyer dropped out.

Applying this principle to a divisible resource such as bandwidth requires that the second price auction become “progressive.” In a progressive second price auction, each bid consists of a unit price and the quantity desired at that price. The bidders are ranked by unit price and then are tentatively awarded the amount they requested until the supply is exhausted. Unsuccessful bidders may re-bid in an attempt to replace successful bidders in the ranking (hence the term “progressive”). The price every successful bidder pays is the price offered by the lowest *successful* bidder. (In a strict interpretation of second price rules, it would be the highest unsuccessful bidder(s) who sets the price. However, the “progressive” nature of the bidding tends to turn all bidders into winners, which leaves no basis from which to set the price.)

Eventually, all bidders will have been awarded a share of the bandwidth or dropped out of the bidding because the market price is higher than they will pay for any amount of bandwidth.

Example:

Assume that the seller has 10 Mbps to sell and that six bidders are interested. Bids indicate the amount of bandwidth desired, as well as the unit price they are willing to pay. Let’s assume that the initial bids are these:

Bidder A: 3 Mbps desired at \$100 unit price (\$300 total)

Bidder B: 5 Mbps desired at \$80 unit price (\$400 total)

Bidder C: 3 Mbps desired at \$50 unit price (\$150 total)

Bidder D: 4 Mbps desired at \$30 unit price (\$120 total)

Bidder E: 6 Mbps desired at \$20 unit price (\$120 total)

Bidder F: 3 Mbps desired at \$10 unit price (\$30 total)

Bidders would be ranked according to their offers, as shown in Figure 33:

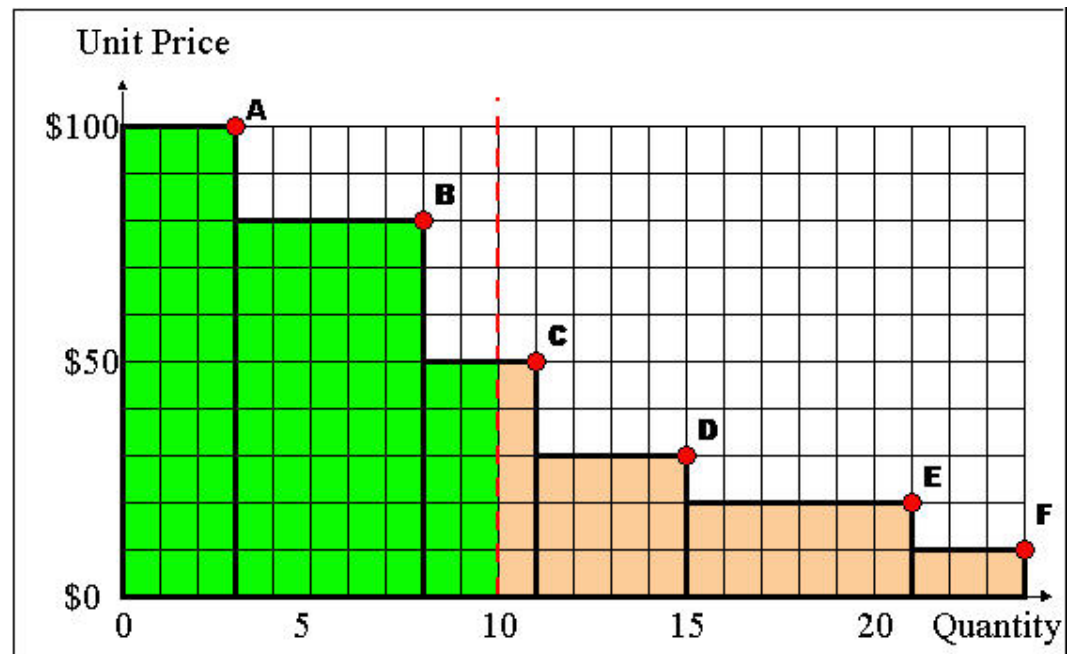


Figure 33: Stacked Bids Received for Portions of 10 Mbps of Bandwidth Offered by Seller

The shaded area represents the amount of money the seller is being offered (unit price times quantity). The green part of the shaded area at the left represents the amount offered by *successful*

bidders.

It might be instructive to determine what the seller would receive had the seller simply taken the highest initial offers. With 10 Mbps for sale, the highest two bidders (“A” and “B”) would receive what they want, and the third highest (“C”) would receive two of the three Mbps bid for. If the seller simply accepted the offers, the seller would receive $\$300 + \$400 + \$100 = \800 for the bandwidth offered from the successful bidders. This represents \$80 per Mbps average price for the 10 Mbps offered.

Now let’s look at how the bandwidth would be sold in a Merkato Progressive Second Price auction. To recap, the rules of the second price auction are as follows:

- All bidders may make multiple sequential offers.
- After each offer is received, the Resource agent (that runs the auction) ranks the bids according to price offered and apportions bandwidth to the highest bidders.
- The price at which the lowest successful bidder receives bandwidth is the “market price,” that every successful bidder pays.

The bid and re-bid cycle repeats until all bidders are satisfied with the amount of bandwidth received, and their price, or have dropped out of the bidding.

Let’s assume that the initial offers from all bidders are as before. All bidders are informed that the current market price is \$50 and the successful and unsuccessful bidders are informed of their status and allocation (if any). If the second price auction stopped here, the seller would receive \$50 for each unit of bandwidth or a total of \$500 for the 10 Mbps of bandwidth, which is \$300 less than he would have received by simply accepting the initial offers. Fortunately for the seller, the process does not end here!

The unsuccessful bidders will bid again, if they can afford to. Bidders “stay in the game” by increasing the unit price offered for bandwidth, above that of the lowest successful bidder.

How can bidders afford to do this if they revealed what they are willing to pay in their first bid? Well, assume that the bidder’s initial offer was determined by a total cost limit as opposed to a desired unit price. In this case, the bidder would desire to re-bid as long as a consistent total cost was maintained—by lowering the requested amount of bandwidth proportionally to any increase in offered unit price. (In Merkato, this price- versus -quantity information is carried by the valuation settings.) Let’s assume, for simplicity, that all the bidders follow this same strategy and that their initial bids represent their total budget for bandwidth.

Bidder “D” was the highest unsuccessful bidder, offering a unit price of \$30 for 4 Mbps of bandwidth. This represents a budget of \$120. Bidder “D” sees he can become a successful bidder this round by bidding above a unit price of \$50. His next offer could be \$60 per unit for 2 Mbps of bandwidth and stay within his budget limits. Each bidder can do this in turn if they find themselves pushed out of the successful bidder’s column.

Although all the examples so far have involved increments of 1 Mbps of bandwidth, bidders are under no such restriction. Even bidder “F,” with a \$30 budget, may become a successful bidder by asking for a fractional amount of bandwidth.

In such a simulation, when the bidding ends, the final bids would look like this:

Bidder A: 2.7 Mbps desired at \$112 unit price (\$300 total)

Bidder B: 3.6 Mbps desired at \$112 unit price (\$400 total)

Bidder C: 1.3 Mbps desired at \$112 unit price (\$150 total)

Bidder D: 1.1 Mbps desired at \$112 unit price (\$120 total)

Bidder E: 1.1 Mbps desired at \$112 unit price (\$120 total)

Bidder F: 0.2 Mbps desired at \$112 unit price (\$30 total)

Figure 34 shows the bids graphically, as before:

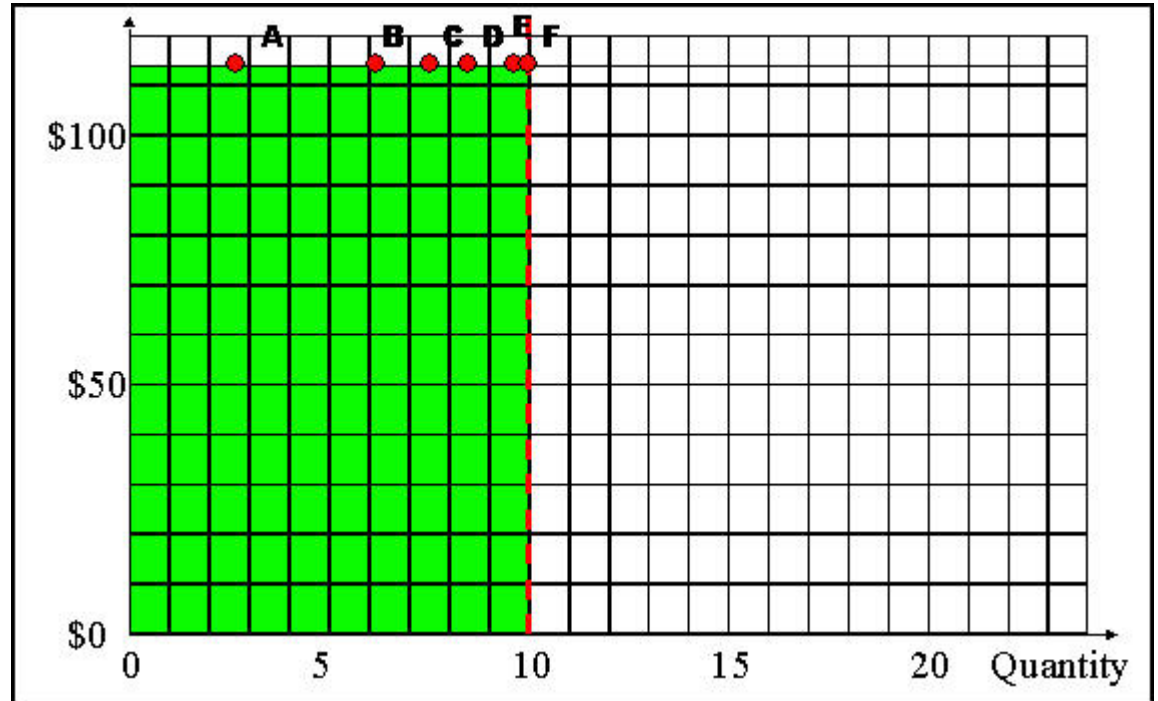


Figure 34: Auction Results Under Progressive Second Price Mechanism

The market price of \$112.00 represents the amount all bidders would have to offer in order that all are awarded bandwidth, but stay within their initial budget. The seller would receive \$1,120.00 for the 10 items.

All bidders are “winners” and receive an amount of bandwidth proportional to their willingness to pay and consistent with the bidding policies they established for their Merkato agents.

Advanced Sellers Guide

As a seller, we assume your goal is to maximize the revenue received from bandwidth sales. From Merkato's point of view, you will be maximizing the quantity of bandwidth sold times the unit price received for that bandwidth. Merkato provides the seller with several revenue-optimizing controls. We break these down into controls that come into play during an auction, controls that can be altered on-demand via the Seller agent, and controls that the Merkato administrator alters (longer term, strategic controls).

All these controls have one thing in common: they either control the amount of bandwidth available or the price at which additional bandwidth is made available.

Controls That React To Market Conditions

At the start of each Spot Market auction, the seller floor price and quantity available for sale are determined by settings in the Seller agent's valuation window. The seller can only release more bandwidth or remove bandwidth during an auction, in response to changing market conditions, via Buyer agents configured by the seller. Essentially, these Buyer agents contend with actual buyers for bandwidth. Because "real" buyers must out-bid these Buyer agents to obtain the bandwidth they hold, the end result is that bandwidth is withheld from the marketplace until a desired price point is reached.

Valuation for Buy-Back Agents

Although any valuation can be used for buy-back agents, we recommend that you use the Linear valuation, for two reasons:

- It is very easy to determine the amount of bandwidth "held back" by Linear valuation Buyer agents and the price at which they will release it. Use of Buyer agents with other valuations requires extensive modeling to ensure that they release the desired amount of bandwidth at the desired price points.
- It is relatively easy for "real" buyers to detect the presence and determine the settings for Linear buyers. One of the important features of Merkato is its openness—the ability of buyers to view what all other buyers have offered for bandwidth, letting them optimize their strategy. If buyers can easily detect the price points at which more bandwidth is released to the market, they can incorporate that knowledge into their buying strategies. Also, the transparency of the Linear bidder's settings allows the seller to change the price-point or quantities for these buyers notifying all buyers of the change.

Example

To illustrate how buy-back agents work, consider a case where the seller has a floor price of \$100 per Mbps for the first 25 Mbps of bandwidth released to the market, and is willing to release three more 25 Mbps blocks of bandwidth if the price reaches \$125, \$150, and \$175 respectively. This would be accomplished with the following seller/inelastic valuation settings for the Seller agent and three buy-back agents:

	Seller Agent	Buy-back Agent 1	Buy-back Agent 2	Buy-back Agent 3
Max Quantity Setting	100 Mbps	25 Mbps	25 Mbps	25 Mbps
Max Value Setting	\$10,000	\$3,125	\$3,750	\$4,375
Resulting unit price	\$100 per Mbps	\$125 per Mbps	\$150 per Mbps	\$175 per Mbps

Considerations About Buy-Back Agents

A few cautionary notes are in order:

- (1) The additional bandwidth will be released, at each price point, only when the market price for all buyers exceeds the unit price represented by the buy-back agent settings. There is no way to penalize only the buyer who desires an unusual amount of bandwidth for a short amount of time.
- (2) You cannot use this method to reduce the price for bandwidth as demand goes up (that is, to provide a quantity discount). This kind of pricing is possible in the Reservation market, however.
- (3) Buy-back agents tend to hold the market price at their level until demand is sufficient to reach the next level. This is because, until demand is sufficient to take all their bandwidth away, these bidders act as the lowest successful bidder and set the market price. There is nothing wrong with this, but the “stickiness” of the market price tends to surprise some first-time observers.
- (4) Changes in the settings of buy-back agents take effect as soon as you click **Apply**. The changes affect the auction in progress.

Although it reduces market “openness,” a single buy-back agent using the Parabolic valuation, whose bid “curve” is a line, may in some cases be used to replace many Linear buy-back agents. This also eliminates the price “stickiness” that results when Linear buy-back agents are used.

On-Demand Controls

Merkato has many features that are controlled by desktop agents, and, in the case of the spot market, take effect at the beginning of the next auction. These include the following:

- (1) Quantity available in the spot market (set via the seller valuation)
- (2) Price floor in the spot market (set via the seller valuation)
- (3) Quantity available in the Reservation market (controlled via a buy-back reservation agent controlled by the seller)

Items (1) and (2) are described fully in the seller valuation section of this manual. (See “Seller Valuation” on page 25.)

Item (3) is the reservation equivalent of a spot market buy-back agent. For example, suppose a seller has a maximum of 300 Mbps and would like to sell them on the Reservation market. Currently the seller wants to release only 100 Mbps. The seller could do so by using a reservation agent to reserve the additional 200 Mbps. Any time the seller wishes to release additional bandwidth to the Reservation market, the seller cancels the reservation. If required, the seller can use multiple reservation agents to release bandwidth to the market more gradually, depending on market conditions.

Settings Controlled by Merkato Administrator

Settings controlled by the Merkato administrator require advanced notice to change. They should be considered strategic rather than tactical controls. Often the Merkato server must be rebooted for these changes take effect. These controls include:

- The duration of the spot market auctions
- The absolute limit to the amount that can be sold in the spot market
- The maximum account balance
- The reservation price graph
- The reservation cancellation fee
- Bid fee controls (used to limit the length of auctions—can potentially be used as broker fees)

Bid Fees

Bid fees serve two purposes:

1. They limit the length of auctions and prevent auction instability
2. They provide a revenue stream for pure bandwidth exchange “brokers”

Bid fees work as follows.

The Resource agent has pre-configured settings for the following values:

- initial bid fee increment
- number of bids to double fee increment (MaxNbids)
- maximum bid fee

The fee for submitting a bid increases continuously throughout the auction. As the bid fee increases, it influences an agent’s behavior in two stages.

The first effect is that the agent does not submit a new bid if it appears that the bid fee outweighs the financial benefit resulting from the bid. This can be considered a “soft limit” imposed by the bid fee.

If the auction continues past this point, there is also a hard limit involved—as soon as the bid fee reaches the maximum bid fee value, the auction is stopped and allocations are made based on the last bids received.

The Resource agent keeps track of the bid fee for each bidder. Each time it gets a bid from any bidder the Resource agent increments the bid fee and inform all bidders what the new value is. Every time the number of total bids reaches the configured “MaxNbids,” the bid fee increment is doubled.

Example

If the initial bid fee increment is 1, and MaxNbids is 100, then the bid fee after the first bid is 1, after the second bid it is 2, and so on, until 100 bids are received. Thereafter the bid fee increment doubles, so that, following the next bid, the fee is 102, then 104, then 106, and so on. After 200 bids, the fee starts incrementing by 4 each bid, after 300 bids, by 8, and so on.

Use of the Bid Fee

In most Merkato installations, the bid fee is used to prevent unstable bidding patterns that keep auctions running forever. The fee is not actually charged to customers. This is largely because the agent behavior is automated and customers have little or no control over the frequency at which their agents bid.

In the future, customers who use Merkato to run bandwidth exchanges may elect to charge the bid fee as a fee for their services.

Valuation Formulas

The formulas from which valuations are derived are provided below. You may wish to enter them into a spreadsheet to analyze them. Remember, these formulas determine what an agent *bids*, but the player's actual cost is based on the market price, as determined by progressive second price auction rules.

The curves defined by these formulas vary according to settings in the valuation windows. To avoid confusion, the names of the settings have been given in the formulas as:

- **Budget** for budget settings in budget-based valuations
- **MaxQty** for either Max Qty in Budget-with-limits or the single Qty setting in the other valuation windows
- **MinQty** for the minimum quantity setting in the Budget-with-limits valuation
- **MaxValue** for the Value setting in the valuation windows

The variables in the equations are:

- **Price** for the unit price
- **Qty** for the quantity of bandwidth that will be requested at that price

Note that the units do not matter as long as you are consistent throughout:

- Budget is specified in currency-per-unit-time (such as \$/month).
- “Quantities” are quantities of bandwidth expressed in Gbps, Mbps, or Kbps.
- Price is specified in currency per unit bandwidth per unit time (such as \$/Mbps/month).
- MaxValue is given in terms of currency per unit time (such as \$/month).

To convert price to cost, multiply by the quantity (“Qty” in the equations).

Budget Valuation

$$\text{Price} = \text{Budget} / \text{Qty}$$

Budget-with-limits Valuation

For $\text{Qty} < \text{MinQty}$

$$\text{Price} = \text{Budget} / \text{MinQty}$$

For $\text{MinQty} \leq \text{Qty} \leq \text{MaxQty}$

$$\text{Price} = \text{Budget} / \text{Qty}$$

For $\text{Qty} > \text{MaxQty}$

$$\text{Price} = 0$$

Linear Valuation

For $\text{Qty} \leq \text{MaxQty}$

$$\text{Price} = \text{MaxValue} / \text{MaxQty}$$

For $\text{Qty} > \text{MaxQty}$

$$\text{Price} = 0$$

Square Root Valuation

Note: “SQRT” = Square Root of what follows in parentheses

For $\text{Qty} \leq \text{MaxQty}$

$$\text{Price} = \text{MaxValue} / (2 * \text{SQRT}(\text{MaxQty}) * \text{SQRT}(\text{Qty}))$$

For $\text{Qty} > \text{MaxQty}$

$$\text{Price} = 0$$

Logarithmic Valuation

Note: “Ln” = Natural logarithm of what follows in parentheses

For Qty \leq MaxQty

$$\text{Price} = (\text{MaxValue} / \text{MaxQty}) * \text{Ln}(\text{MaxQty} / \text{Qty})$$

For Qty $>$ MaxQty

$$\text{Price} = 0$$

Parabolic Valuation

For Qty \leq MaxQty

$$\text{Price} = 2 * \text{MaxValue} (1 - \text{Qty} / \text{MaxQty}) / \text{MaxQty}$$

For Qty $>$ MaxQty

$$\text{Price} = 0$$

- Maximum price bid (at Qty = 0) = $2 * \text{MaxValue} / \text{MaxQty}$
- Maximum cost offered (at half MaxQty) = $\text{MaxValue} / 2$

Glossary

agent: The program that interacts with the rest of Merkato on behalf of buyers and sellers. Buyers can acquire bandwidth by configuring their agents to offer the price they are willing to pay for a range of available quantity, or use their agent to request a quote for a fixed-price bandwidth reservation. Sellers configure their agents with a quantity of bandwidth for sale and a minimum price they are willing to accept for that quantity.

allocation: An amount of bandwidth available for your use. Depending on the type of service being offered through Merkato, this can represent the maximum bandwidth available to you or a minimum guarantee of bandwidth available to you.

bandwidth: The amount of data transmitted or received per unit of time. When we refer to acquiring or selling bandwidth, we mean the amount of information that can be sent over a connection at one time, at the allowed speed, without packet loss or excessive delay. Bandwidth is measured in bits-per-second.

Buyer agent: The program that interacts with the rest of Merkato on behalf of buyers. Buyers can acquire bandwidth by configuring their agents with the price they are willing to pay for a range of bandwidth (on the spot market), or with a time-frame and desired quantity (on the Reservation market). In a Merkato auction, Buyer agents signal what they are willing to pay for bandwidth in the form of bids, consisting of a unit price and a quantity.

capacity: In the Merkato interface, the term capacity refers to the total amount of bandwidth available from the Seller.

data transfer rate: The speed at which data is transferred from one port to another. See Kbps, Mbps, and Gbps.

Desktop agent: The Merkato application for buyers or sellers that runs on a Windows PC. Merkato desktop agents bid for bandwidth from the user's PC using a Java-based application. Alternatively, Merkato agents can run as an application on a server, called the "Garage." Users can interact with these remote agents through an HTML browser interface.

Express agent: The Merkato HTTP application for buyers or sellers that configures and obtains status from Merkato agents running from a remote location. Accessing the agent in this way is an alternative to bringing the agent to the user's desktop and using the Java-based interface. The Express agent is accessible through the Portal.

floor price: The lowest price the seller will accept for bandwidth. The seller can establish the floor price through the Seller agent.

garage: A feature on the Merkato server where player (Buyer and Seller) agent configuration profiles reside. The garage stores agent configurations whether the players are active or not.

Gbps: One of the ways of expressing units of bandwidth; Gigabits-per-second (1,000,000,000 bits-per-second).

Java: A programming language, useful for internet and networking applications. The environment used for the Merkato Desktop Interface.

kbps: One of the ways of expressing units of bandwidth—kilobits-per-second (1,000 bits-per-second).

market price: The price for something that buyers and sellers agree on. Merkato establishes a market price for bandwidth during each spot market auction round. There is a fixed amount for sale, so as demand increases, prices rise. Ordinarily, the market price is reached when the cumulative demand of all the buyers is equal to the amount of bandwidth being offered by the seller.

marketed service: The actual bandwidth that is offered for sale in Merkato. It contains Quality-of-Service attributes and is associated with a particular port on a particular router. Merkato resource agents request a portion of the bandwidth available through a service and sell that portion through their marketplace.

Mbps: One of the ways of expressing units of bandwidth—Megabits-per-second (1,000,000 bits-per-second).

Merkato desktop: The Merkato desktop is the first window that appears when the user begins to bring the Merkato buyer or seller agent to their desktop. It lets users specify which Merkato agent or agents they want to download to the desktop.

Merkato: InvisibleHand's software platform, which dynamically prices, sells, and allocates IP bandwidth in real time. Merkato means "market" in Esperanto.

Player: A buyer or seller in the Merkato auction.

Portal: The main access point to all Merkato applications. It is an HTML-based interface accessible from a standard web browser.

progressive second price auction: The unique market mechanism by which Merkato allocates bandwidth to potential buyers at an optimal market price. An auction is established, where each bid consists of a unit price and the quantity desired at that price. Bidders are ranked according to the unit price they offer. Bidders who get an allocation pay the unit price offered by the highest bidder who does not get an allocation. (This is the "second price" aspect of the auction.) Bidders who don't get an allocation can re-bid at a higher unit price, in an attempt to improve their ranking in the auction. (This is the "progressive"

aspect of the auction.) An auction round closes when all bidders are either successful, based on their last bid, or cannot match the price offered by the current lowest successful bidder.

PSP: See “progressive second price auction.”

Reservation market: The Merkato market mechanism by which a specified quantity of bandwidth, for a specific duration, is sold for a firm price specified by the seller and agreed to by the buyer. This is an automated process based on a rate sheet that the seller establishes in advance.

Resource agent: The program that manages the Merkato marketplaces, where bids are received and allocations issued (the spot market) or where price quotes for reservations are generated and accepted quotes for reservations are processed.

Seller agent: The program that interacts with the rest of Merkato on behalf of sellers. Sellers configure their agents with a quantity of bandwidth capacity for sale and a minimum price for that quantity. Buyer agents then express what they are willing to pay for bandwidth in the form of bids, consisting of a unit price and a quantity. Currently, there is no seller agent for the Reservation market, but one will be supported in a future release.

Seller strategy: A window in the Seller desktop interface, either static or dynamic. Dynamic Seller strategy sets prices based on the last unit of bandwidth sold. Static Seller strategy sets prices based on the difference between what the Seller would have received without each bid.

Spot market: The Merkato mechanism by which bandwidth is traded, in a progressive second price auction. An optimal fair market price is established and bandwidth is allocated to buyers, based on their bids relative to other buyers.

Spot Market agent: A Buyer agent or Seller agent connected to a spot market. There are also reservation agents (for buyers only).

valuation: The value a buyer or seller places on bandwidth. Setting a valuation is part of setting a purchasing strategy. Valuation settings within a buyer agent let buyers specify the amount they are willing to pay for varying amounts of bandwidth. This information is used by the agent to respond to changing market conditions during a Merkato progressive second price auction.

Wizard: In Merkato, an automated series of inquiry screens that walks you through the process of creating and configuring a Spot market Buyer agent.

