Chapter 8

Results

8.1 Experimental Setup

The system comprises of 22 server node processes organized into the topology shown in the Figure 8.1.1. This set up is used so that the effects of queuing delays at higher publish rates, message sizes and matching rates are magnified. True topologies for GES systems are the one depicted in figure 8.1.1. Each server node process is hosted on 1 physical Sun SPARC Ultra-5 machine, with no SPARC Ultra-5 machine hosting two or more server node processes. The run-time environment for the server node processes is JDK-1.2. For the purpose of gathering performance numbers we have 1 publisher in the system and 200 client node processes with 5 client nodes attached to every server node within the system. The 100 client node processes reside on a SPARC Ultra-60 machine. The publisher is responsible for issuing events, while the subscribers are responsible for registering their interest in receiving events. The publisher and the *measuring* subscriber reside on another SPARC Ultra-5 machine.

8.2 Factors to be measured

Once the publisher starts issuing events the factor that we are most interested in is the *latency* in the reception of events. This latency corresponds to the response times experienced at each of the clients. We measure the latencies at the client under varying conditions of *publish rates*, *message sizes* and *matching rates*. Publish rates and message sizes correspond to the rate at which messages are being issued by the publisher and the size of these individual messages respectively. Matching rate is the percentage of events that are actually supposed to be receieved at a client. In most publish subscribe systems, at any given time for a certain number of events being present in the system, any given client is generally interested in a very small subset of these events. Varying the matching rates allows us to simulate such a scenario, and perform measurements under conditions of varying selectivity. For a sample of messages received at a client we calculate the *mean latency* for sample of received messages, the *variance* in the sample of these messages and the *system throughput* measured in terms of the number of messages received per second at the measuring subscriber. We also measure the highest and lowest message latencies within the sample of messages that have been received. Another very important factor that needs to be measured is the change in latencies as the connectivity between the nodes in a server network is increased. This increase in connectivity has the effect of reducing the number of server hops that an event has to take prior to being received at a client. The effects of change in latencies with decreasing server hops is discussed in section 8.4.

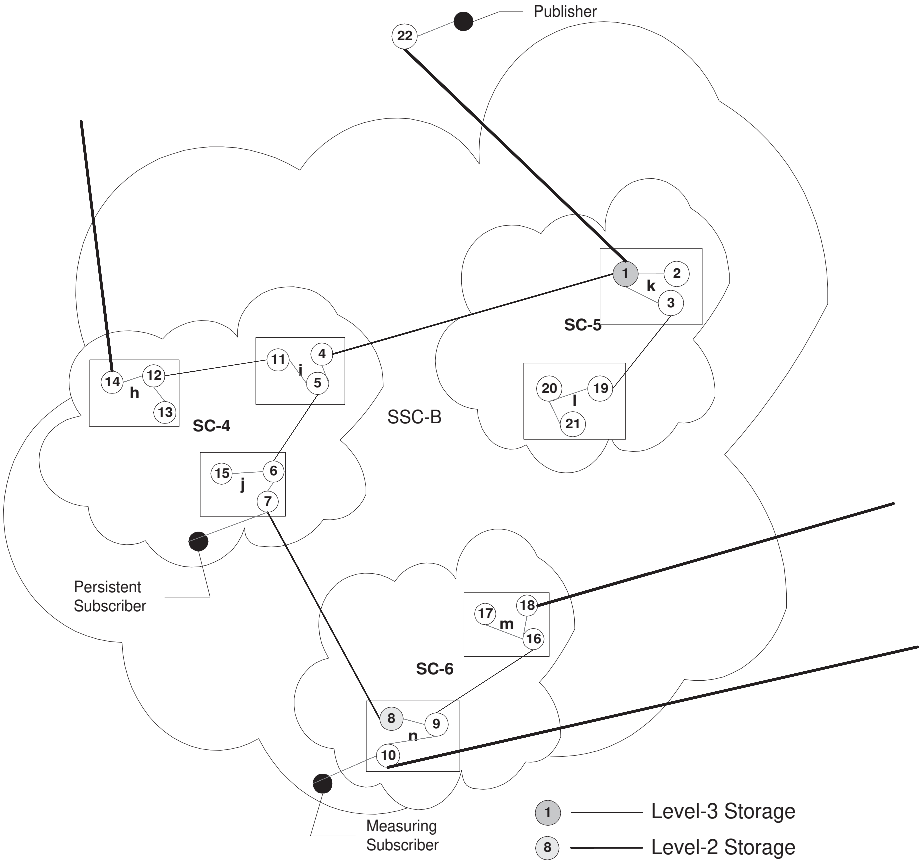

Figure 8.1.1: Testing Topology

8.2.1 Measuring the factors

For events published by the publisher the number of tag-value pairs contained in every event is 6, with the matching being determined by varying the value contained in the fourth tag. The profile for all the clients in the system, thus have their first 3 <*tag=value*> pairs identical to the first 3 pairs contained in every published event. This scheme also ensures that for every event for which destinations are being computed there is some amount of processing being done. Clients attached to different server nodes specify an interest in the type of events that they are interested in. This matching rate is controlled by the publisher, which publishes events with different footprints. Since we are aware of the footprints for the messages published by the publisher, we can accordingly specify profiles, which will allow us to control the dissemination within the system. When we vary the matching rate we are varying the percentage of events published by the publisher that are actually being received by clients within the system. Thus when we say that the matching rate is set at 50%, any given client will receive only 50% of the events published by the publisher. To vary the publish rates, we control the *sleep time* associated with the publisher thread, and also the number of messages that it publishes at a time once the publisher thread *wakes up*. This requires some preliminary tuning, once the values for the *sleep time* and the number of messages that are published at a time have been fixed for the publisher and the corresponding server node in question, we proceed to compute the real publish rates for the sample of messages that we send. This is the publish rate that we report in our results.

For each matching rate we vary the size of the messages from 30 to 500 bytes, and vary the publish rates at the publisher from 1 to 1000 Messages per second. For each of these cases we measure the latencies in the reception of events. To compute latencies we have the publishing client and one the *measuring*subscriber residing on the same machine. Event's issued by the publisher are *time-stamped* and when they are received at the subscribing client the difference between the *present time* and the *time-stamp* contained in the received message constitutes the latency in the dissemination of the event at the subscriber via the server network. In case the publisher and the subscriber were on two different machines, with acess to different underlying system clocks we would need to synchronize the clocks and also account for the drift in clock rates prior to computing the latencies in message reception. Having the publisher and one of the subscribers on the same physical machine with access to the same underlying clock, obviates this need for clock synchronization and also accounts for clock drifts. It should be noted that though the publisher and the *measuring* subscriber are on the same machine, they are connected to two different server nodes within the server network, as depicted in figure 8.1.1. In fact it takes 10 hops for a message issued by the publisher to be received at the measuring subscriber.

8.3 Discussion of Results

In this section we discuss the latencies gathered for varying values of publish rates, message sizes and matching rates. We then proceed to include a small discussion on system throughputs at the clients, and another discussion that outlines the trends in variance in latencies of messages received at a client. The results also discuss the latencies involved in the delivery of events to persistent clients in units with different replication schemes.

8.3.1 Latencies for the routing of events to clients

At high publish rates and increasing message sizes, the effects of queuing delays come into the picture. This queuing delay is a result of the messages being added to the queue faster than they can be processed. In general, the mean latency associated with the delivery of messages to a client is directly proportional to the size of the messages and the rate at which these messages were published. The latencies are the lowest for smaller messages issued at low publish rates. The mean latency is further influenced by the matching rates for events issued by the publisher. The results clearly demonstrate the effects of flooding/queuing that take place at high publish rates and high message sizes and high matching rates at a client. It is clear that as the matching rate reduces the latencies involved also reduce, this effect is more pronounced for cases involving messages of a larger size at higher publish rates.

Figures 8.3.1 through 8.3.4 depict the pattern of decreasing latencies with decreasing matching rates. The latencies vary from 391.85 *mSecs* to 52.0 *mSecs* with the <*publish rate, message size*> varying from <952 *messages/Sec* , 450 *Bytes*> for a matching rate of 100% to <952 *messages/Sec*, 400 *Bytes*> for a matching rate of 10%. This reduction in the latencies for decreasing matching rates, is a result of the routing algorithms that we have in place. These routing algorithms ensure that events are routed only to those parts of the system where there are clients which are interested in the receipt of those events. Thus events are queued only at those server nodes which

- Have attached clients interested in those events
- Are en route to server nodes which are interested in these events. These server nodes generally fall in the shortest path to reach the destination node.

In the flooding approach, all events would still have been routed to all clients irrespective of the matching rates.

Figure 8.3.1 depicts the case for matching rates of 100%. In this case the mean latency for the sample of messages varies from 15.54 *mSec* for <1 *message/Sec*, 50 *Bytes*> at a throughput of 1 message/Sec to 391.85 *mSec* for <952 *messages/Sec*, 450 *Bytes*> with a throughput of 78 *messages/Sec* at the client. The variance in the sample of messages varies from 2.3684 $mSec^2$ to 69,713.93 $mSec^2$ for the 2

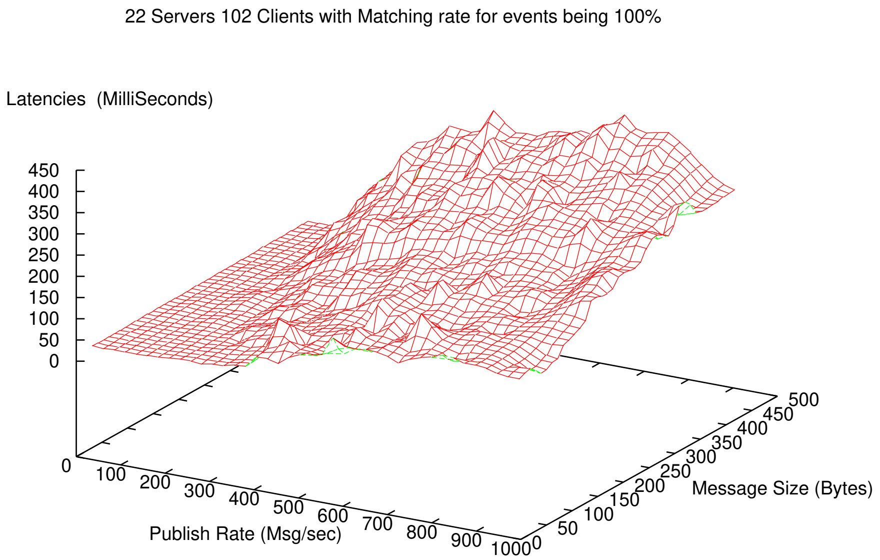

Figure 8.3.1: Match Rates of 100

cases respectively. The maximum throughput achieved was 480.76 *messages/Sec* at publish rates of 492 *messages/Sec* with messages of size 75 bytes.

Figure 8.3.2 depicts the case for matching rates of 50%. In this case the mean latency for the sample of messages varies from 13.02 *mSec* for <20 *messages/Sec*, 50 *Bytes*> to 178.66 *mSec* for <952 *messages/Sec*, 350 *Bytes*>. The variance in the sample of messages varies from 56.8196 $mSec^2$ to 14,634 $mSec^2$ for the 2 cases respectively.

Figure 8.3.3 depicts the case for matching rates of 25%. In this case the mean latency for the sample of messages varies from 14.40 *mSec* for <20 *messages/Sec*, 50 *Bytes*> to 66.6 *mSec* for <961 *messages/Sec*, 400 *Bytes*>. The variance in the sample of messages varies from 0.24 $mSec^2$ to 587.04 $mSec^2$ for the 2 cases respectively.

Figure 8.3.4 depicts the case for matching rates of 10%. In this case the mean latency for the sample of messages varies from 14.40 *mSec* for <20 *messages/Sec*, 50 *Bytes*> to 52.0 *mSec* for <952 *messages/Sec*, 400 *Bytes*>. The variance in the sample of messages varies from 0.44 $mSec^2$ to 103 $mSec^2$ for the 2 cases respectively.

8.3.2 System Throughput

We also depict the system throughputs at the client under conditions of varying message sizes and publish rates. We choose to depict the system throughputs at a Matching rate of 100% since at other matching rates only the relevant events are being routed to the clients, and thus does not reveal the true throughputs that can be achieved at a client. Figure 8.3.5 depicts the system throughputs achieved at a client under conditions of different publish rates and message sizes. The maximum throughput achieved was 480.76 *messages/Sec* at publish rates of 492 *messages/Sec* with messages of size 75 bytes.

Figure 8.3.2: Match Rates of 50

Figure 8.3.3: Match Rates of 25

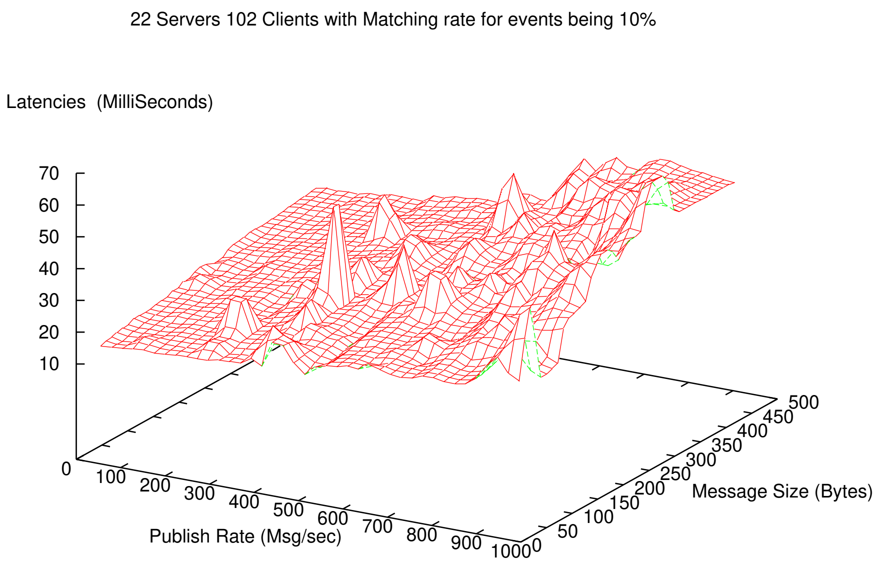

Figure 8.3.4: Match Rates of 10

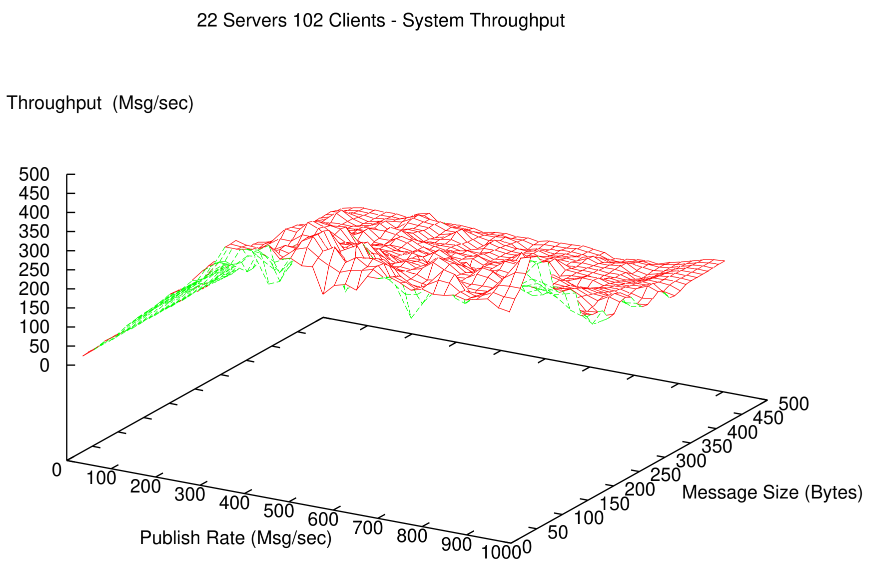

Figure 8.3.5: System Throughput

8.3.3 Variance

Variance for the sample of received messages at a client, demonstrate how queueing delays can add up to increase the mean latency, and also how this mean latency has high deviations from the highest and lowest latencies contained in the sample of latencies for messages received at a client. The variance in the sample of messages varies from 69713 $mSec^2$ to 133.76 $mSec^2$ for <952 *messages/Sec* , 450 *Bytes*> at matching rates of 100% to <877 *messages/Sec*, 450 *Bytes*> at matching rates of 5%. Thus variance in the sample of messages for higher message sizes at higher publish rates also reduces with decreasing matching rates for the published events.

8.3.4 Persistent Clients

In figure 8.1.1 we have also outlined the replication scheme that exists in the system. When an event arrives at node **1** the event is first stored to the level-3 stable store so that it has an epoch associated with it. The event is then forwarded for dissemination within the unit. Clients attached to node in super-cluster **SC-6** have a replication granularity of r_2, thus when the events issued by the publisher in the test topology is being disseminated when clients attached to nodes in **SC-6** receive that event, the event would have been replicated twice. For testing purposes we set up another *measuring* subscriber at node **7** in addition to the subscriber that we would set up at node **10**. When an event is received by the subscriber attached to node **7** the event would have been replicated only once, at node **1**. These *measuring* subscribers allow us to measure the response times involved for singular and double replications experienced at clients attached to nodes **7** and **10** respectively. Every node in the system has 5 persistent clients attached to it, for a total of 102 persistent clients. The publisher and the 2 *measuring* subscribers are all hosted on the same machine for reasons discussed earlier. Figures 8.3.6 and 8.3.7 depict the latencies in delivery of events at persistent clients, with singular and double replications.

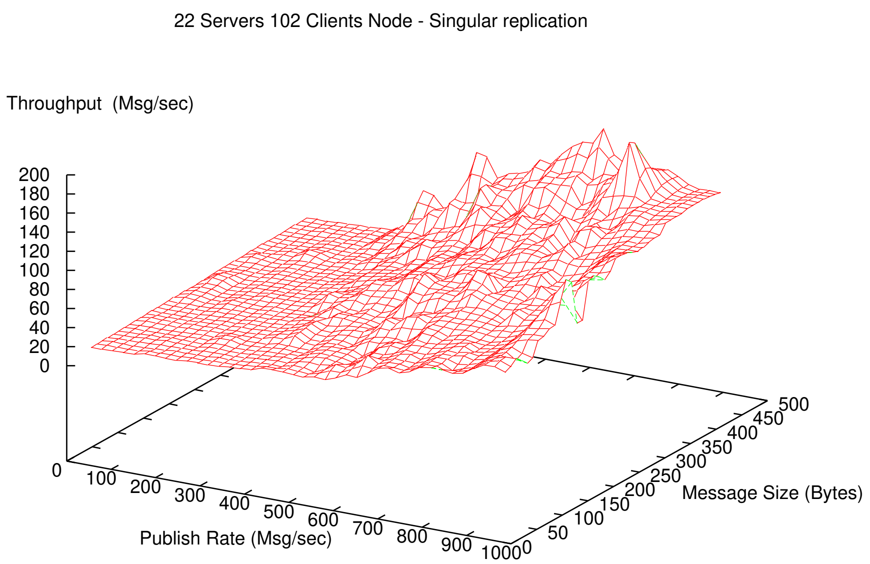

Figure 8.3.6: Match Rates of 50% - Persistent Client (singular replication)

Figure 8.3.7: Match Rates of 50% - Persistent Client (double replication)

8.4 Pathlengths and Latencies

The topology in figure 8.1.1 allows us to magnify the latencies which occur by having the queuing delays at individual server hops add up. In that topology the number of server hops taken by an event prior to delivery at the measuring subscriber is 9. We now proceed with testing for the topology outlined in figure 8.4.1. The layout of the server nodes is essentially identical to the earlier one, with the addition of links between nodes resulting in a strongly connected network. We have 5 subscribing clients at each of the server nodes. The mapping of server nodes and subscribing client nodes to the physical machines is also identical to the earlier topology. As can be seen the addition of super-cluster link between super-clusters **SC-5** and **SC-6**, and level-0 links between nodes **8** and **10** in cluster **SC-6.n** reduces the number of server hops for the shortest path from the publisher to the measuring subscriber at node **10** from 9 to 4.

In this setting we are interested in the changes in latencies as the number of server hops vary. We measure the latencies at three different locations, the measuring subscriber at node **10** has a server hop of 4 while the ones at nodes **1** and **22** have server hops of 2 and 1 respectively for events published by the publisher at node **22**.

In general as the number of server hops reduce the latencies also reduce. The patterns for changes in latency as the message size and publish rates increase is however similar to our earlier cases. We depict our results, gathered at the three measuring subscribers for matching rates of 50% and 10%. The pattern of decreasing latencies with a decrease in the number of server hops is clear by looking at figures 8.4.2 through 8.4.7. We had also made measurements for a matching rate of 25%, and the pattern is the same in those results too. We have however not included the figures for this case.

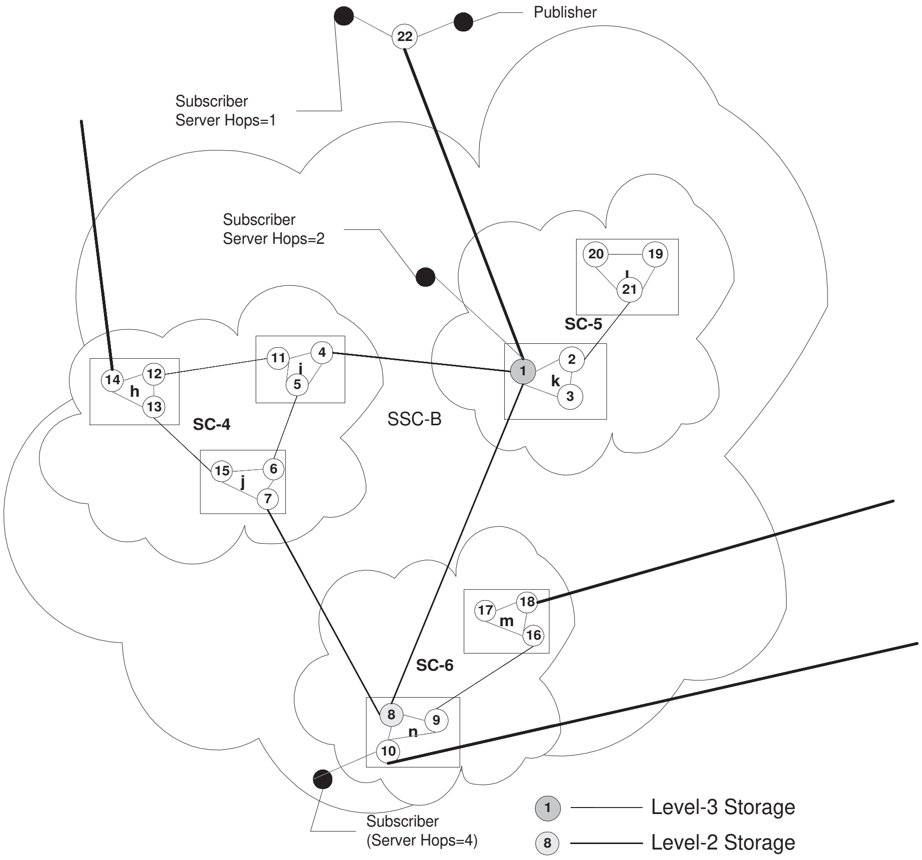

Figure 8.4.1: Testing Topology - Latencies versus server hops

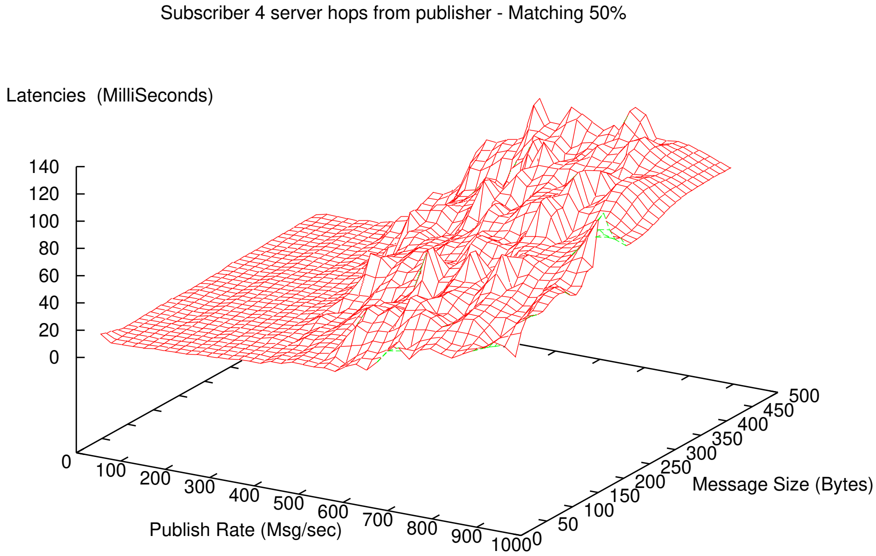

Figure 8.4.2: Match Rates of 50% - Server Hop of 4

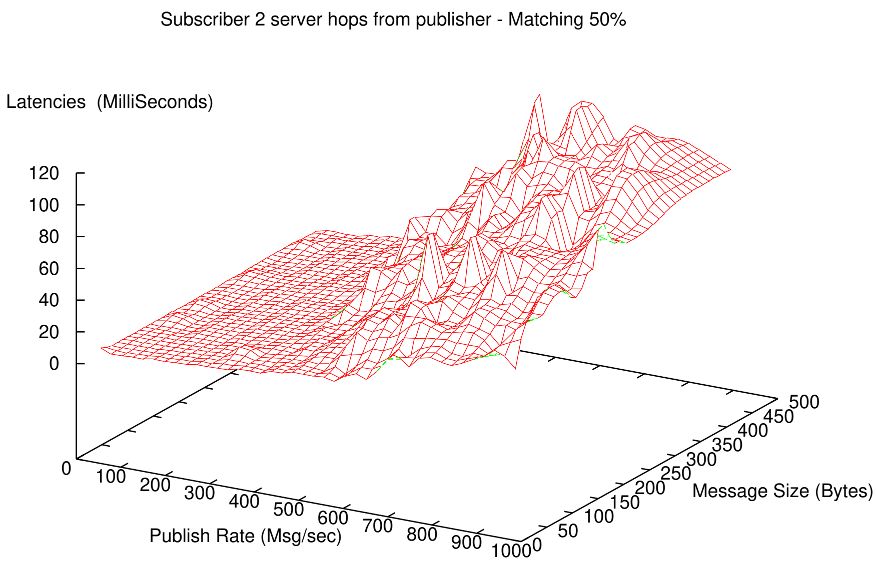

Figure 8.4.3: Match Rates of 50% - Server Hop of 2

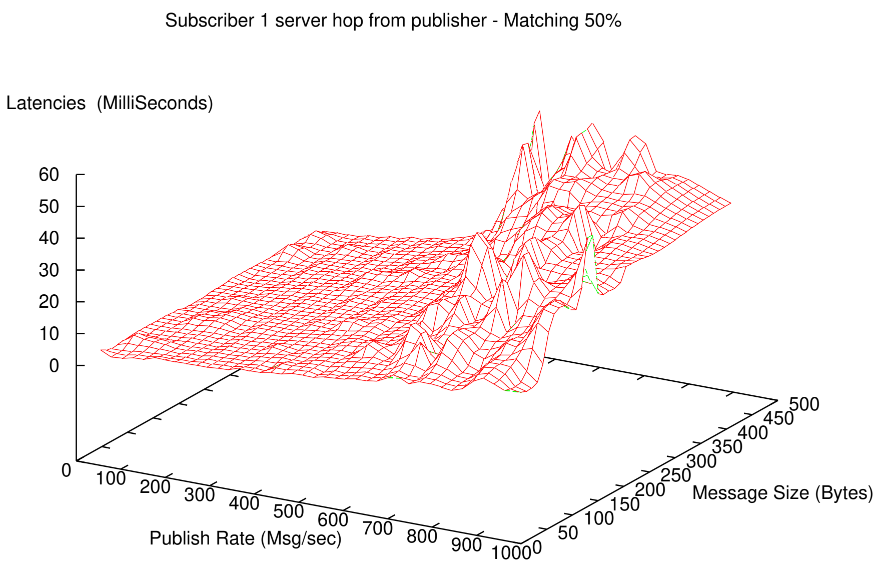

Figure 8.4.4: Match Rates of 50% - Server Hop of 1

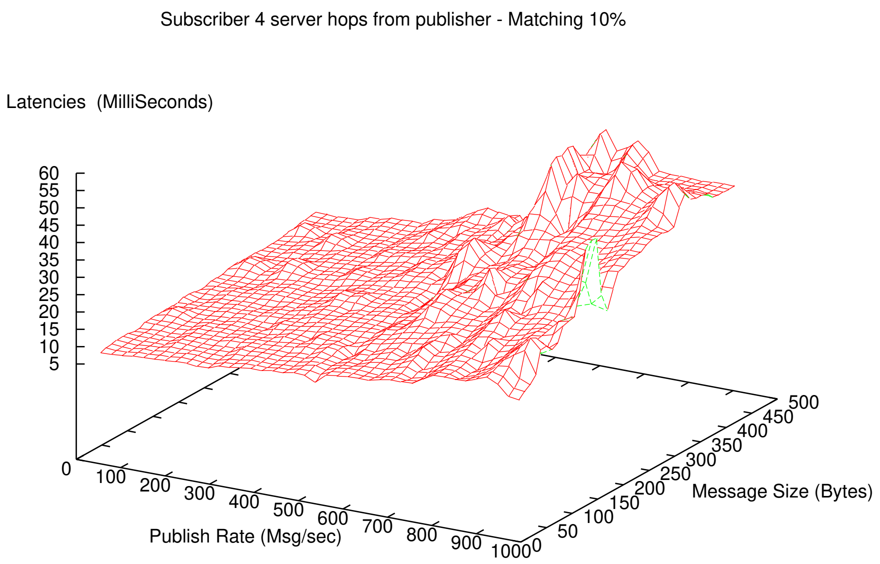

Figure 8.4.5: Match Rates of 10% - Server Hop of 4

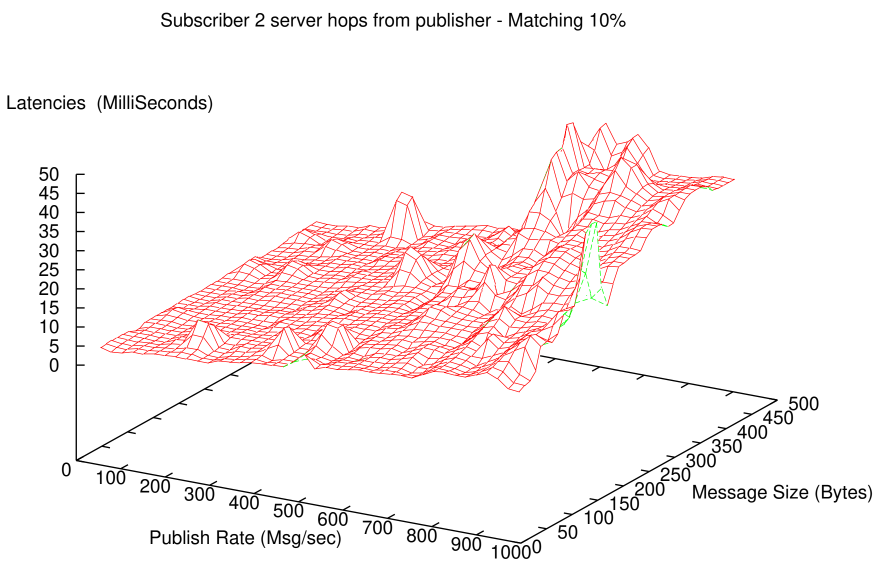

Figure 8.4.6: Match Rates of 10% - Server Hop of 2

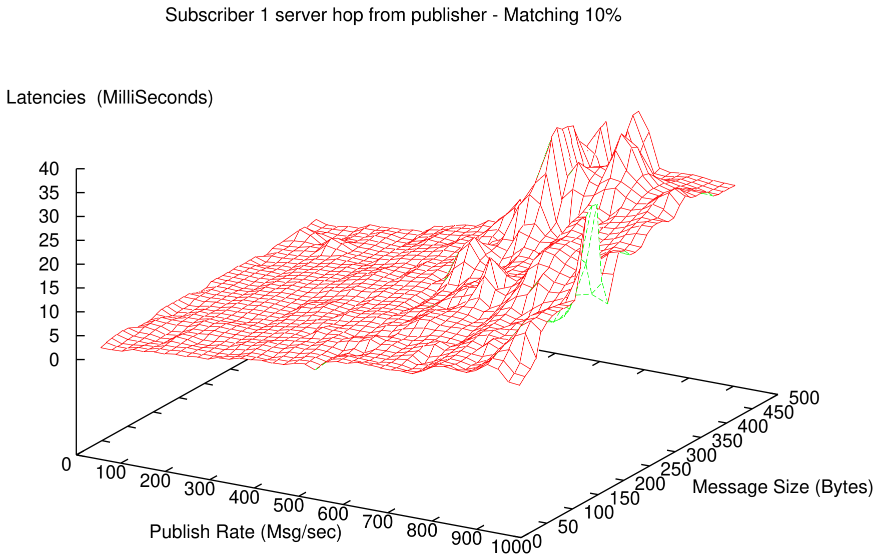

Figure 8.4.7: Match Rates of 10% - Server Hop of 1