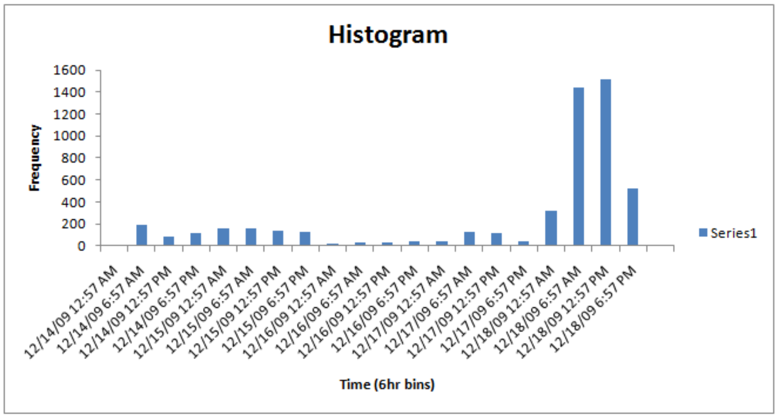
## Results

We ran our system for approximately one week. Part of this period included stability testing for the client; however, we were receiving experimental data throughout. By Friday, December 19, 2009 at approximately 9:00pm, we had received 259 data updates from 17 unique hosts. During the same time period, we had over 5000 “master activity events”, which involved clients performing checkin with the master, reporting data being sent, or reporting errors back to the master.



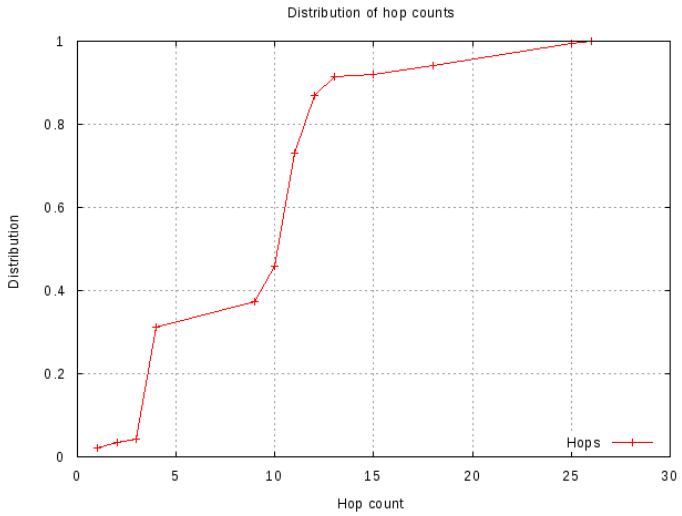
The frequency of client checkins rose dramatically for two reasons. First, we completed the Windows version of our client script, as well as the distributable packages for Mac and Linux systems, on the evening of December 17th. Secondly, we made our software publically available on December 18th, adding real-time statistics to our website as well as posting our distributable packages there. We also began advertising our project via Twitter and Facebook. Prior to this, we only distributed our client script to selected volunteers for testing; the bulk of the pre-December 17 checkins are actually from machines that we personally controlled. All checkins and data prior to this date were from Linux or Mac-based clients.

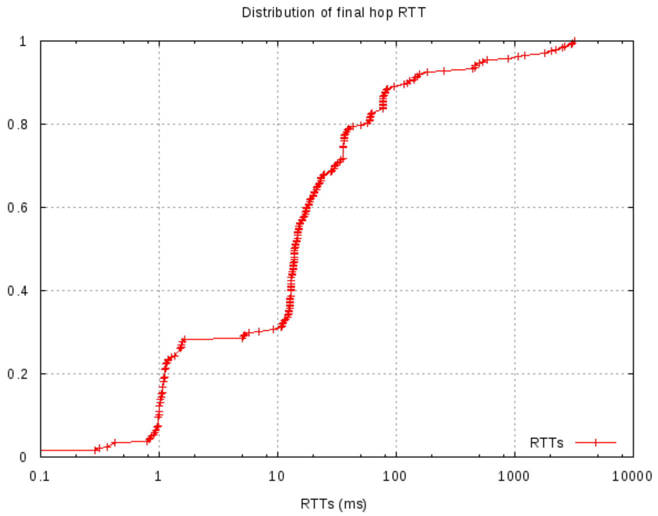
The large discrepancy between master checkins versus data actually collected is attributable to problems with our client. We were not able to finish testing of the client script, so there are likely bugs in the code we wrote. Secondly, and most seriously, the Windows version of pathload2 we were given does not run reliably, frequently crashing, generating numerous error messages without returning any data.

Despite these setbacks, we were able to gather data from a variety of hosts over the course of the week. Most of the 17 hosts that did report experimental data to our system did so at least five times, with several hosts reporting data over 20 times. This data is used in our analysis. Each of these systems reported their IP address, the IP address of the pathload server to which they ran their experiment, the available bandwidth both to and from that server, as well as the route to that server. Each hop in the route contained the RTT to the server at that hop, as well.

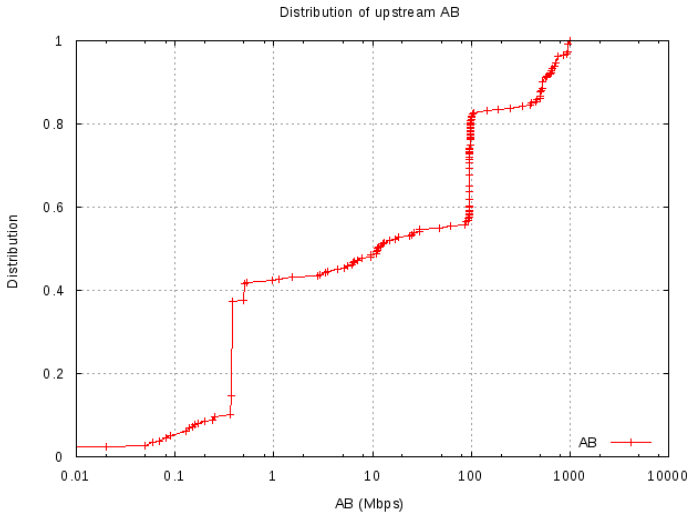
## Analysis

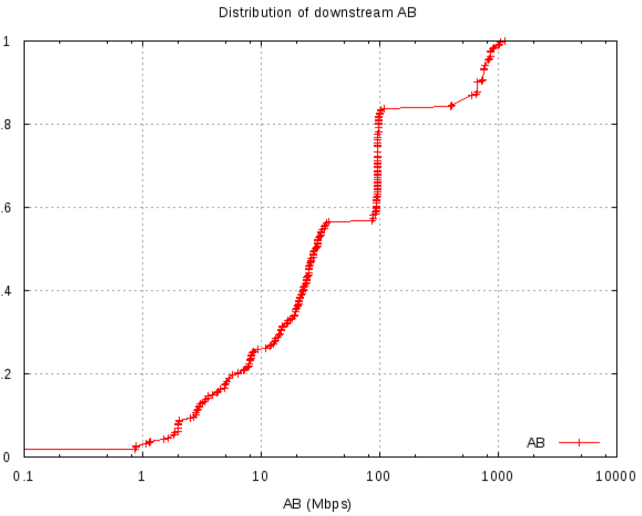
While we initially targeted clients in the Triangle area, the nature of our advertising attracted participants from a wide cross-section of the internet. While just under 50% of our clients were ten hops away from their pathload server, we saw clients up to 26 hops away. Approximately 30-40% of our clients reported from less than 5 hops away from their pathload server. The high proportion of very close clients is due to the clients we ran on machines inside the department, which were on the same network as our five pathload servers.





The distribution of “final-hop RTT” followed a similar pattern to the total number of hops per path. Because RTT is calculated to every host on the path between client and server, the RTT of the final hop of the route is the RTT between the client and the server. The high proportion of RTT’s in the 1ms range is again likely due to the several clients we had running inside the department. Our average RTT was 134.7ms, with a standard deviation of 481ms.



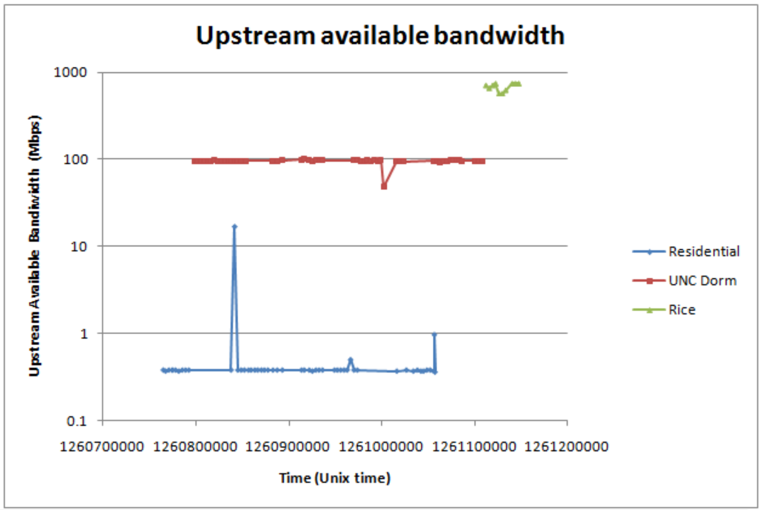


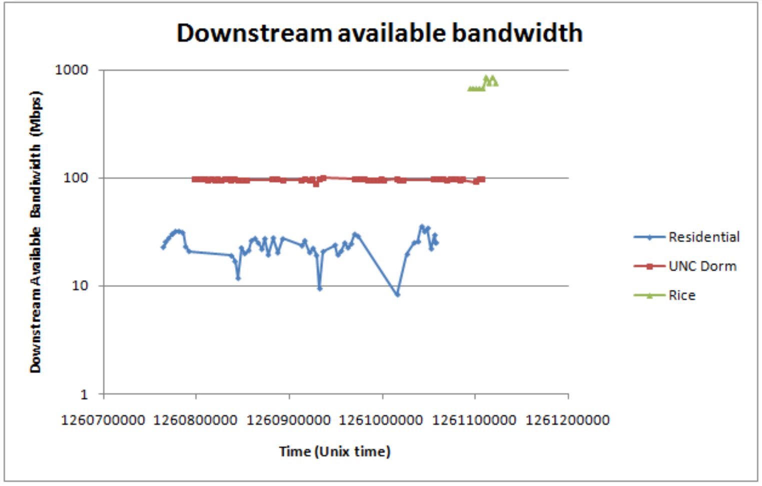
The distribution of available bandwidth across all our experiments reveals several interesting properties. First, available bandwidth in the downstream direction is generally greater than that of the upstream direction. Whereas over 40% of paths have less than 1Mbps of available bandwidth in the upstream direction, 40% of paths have over 20Mbps of available bandwidth in the downstream direction. This asymmetry may be due to the several clients we had behind residential internet connections, which typically undergo traffic shaping to enforce a highly asymmetric ratio of upstream to downstream bandwidth.

Additionally, the available bandwidth distribution has a very marked peak near 100Mbps. This, of course, is the maximum bandwidth of a link operating over a Fast Ethernet physical layer. Our data shows that the bottleneck available bandwidth for many paths is limited by a connection with approximately 100Mbps of available bandwidth. This would indicate a 100Mbps Ethernet connection links two segments of a path with at least (very likely more than) 100Mbps of available bandwidth. More likely, this jump represents a bottleneck *access link*, such as would be seen by a client machine connected directly to the UNC campus network via a 100Mbps Ethernet switch (we did appear to have at least one such client). Another, smaller peak can be seen close to the 1Gbps boundary, indicating some paths may be limited by their underlying Gigabit Ethernet network.

More data from a greater number of clients would give us much richer data in this regard. We were unable to recruit many wireless clients that reported data for more than a few experimental cycles (this makes sense, given the transient nature of wireless clients). We expect that a peak similar to that seen at 100Mbps would be seen close to the limits of the Wifi physical layer, though potentially less dramatic due to the variability in quality of a wireless connection.

We next consider three specific hosts with very different performance profiles. The first of these is connected via a residential internet connection in the Chapel Hill area. The second of these comes from a client connecting from a dormitory on the UNC campus. Finally, we include a set of data reports from a client connecting from the computer science department at Rice University in Houston, TX. We compare upstream and downstream available bandwidth, and latency, for each of these clients.

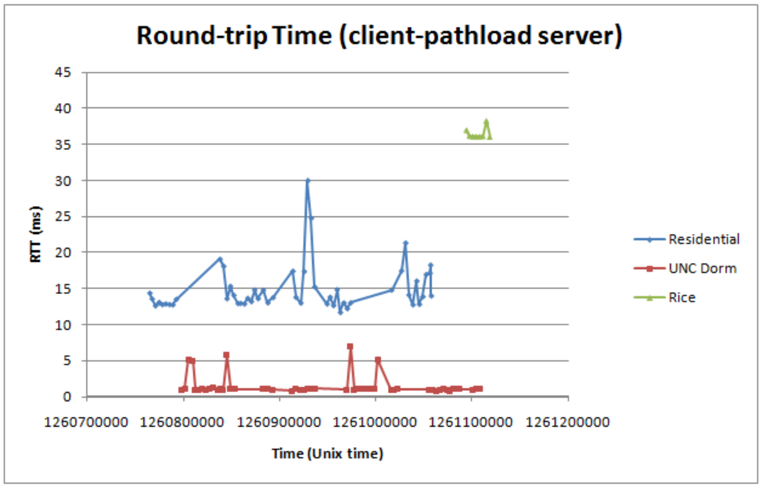




These charts show characteristic signatures of each of these network types. The residential connection shows a very uniform – and limited – amount of available upstream bandwidth, likely due to restrictions placed on that connection by the ISP. Downstream available bandwidth sees a similar "upper bound", albeit a significantly higher one. Nonetheless, the available bandwidth for this client is highly variable in the downstream direction. This variability makes intuitive sense to be seen by a residential customer, who could be sharing an oversold connection with multiple customers of their ISP. Similar variability is also seen in RTT from this client, which provides some evidence that the quality of the connection provided by the ISP is similarly variable, perhaps due to queuing delays where customer connections are terminated inside the

ISP.

Next, consider the UNC dorm connection. Unlike the (off-campus) residential connection, paths from this client show remarkable uniformity as well as a strong upper bound of 100Mbps available bandwidth. A computer in a dormitory would be connected directly to the UNC campus network via a 100Mbps Ethernet connection, which would serve as the limiting factor here, as the destination servers are directly attached to the UNC campus' gigabit backbone network. Latency also shows very little variability; this is less surprising, however, given that the client is essentially on the same network as the servers.



Slightly different than both of these is the client connecting from Rice University. While this client was only sending data for a few hours, its path to our servers – and the UNC campus network – was over Internet2. As a result, it was able to report available bandwidth in the range of 500-700Mbps, exceeding all other clients we saw save for those connected directly to the computer science department's network. Latency, likewise, stayed quite steady, approximately twice that of the residential client, despite being orders of magnitude farther from the servers physically.

Finally, we collected all our experimental data into a visualization of all paths between clients and servers that we observed, as well as our best estimate of the lower bound on the available bandwidth of a given link. This lower bound is the *maximum* of the available bandwidth measurements seen for paths traversing that link; if we see a certain level of available bandwidth on a link, we know that link could provide *at least* that much available bandwidth given the proper network conditions. This visualization is quite large, so we do not include it here. However, it is accessible at http://wwwx.cs.unc.edu/~gavaletz/mapnet/plots/pretest2.png.

## Conclusions

Our experiments demonstrated that a distributed system for measuring absolute bandwidth can yield interesting, non-obvious insights about the nature of network links. We identified distinct trends among our clients depending on what type of connection the client was behind. In addition to our measurements of available bandwidth, we were able to reconstruct the topology of the router links between our clients and servers, as well as approximate progressively tighter bounds on the potential available bandwidth of those links. Despite these results, we recognized the importance of having a multitude of diverse clients to avoid dramatically underestimating the capacity of a backbone link due to an available bandwidth bottleneck at the client's access link.

We believe our results validate the architecture of our distributed monitoring system. Despite an inability to communicate with a large number of clients, we were able to dynamically update our experimental configuration and continue receiving updates from stable clients. We believe the problems that prevented us from communicating reliably with the other clients will be resolved with further testing of our client scripts, particularly with regard to integration with the pathload client and to installation and deployment processes across the three supported platforms.