ZScore Web Server Testing

Testing Scope

The purpose of this testing was to verify ZScore's web server behaviour under the typical use case scenario and find system performance limits by increasing the web traffic load until the system breaking point is found (i.e system load and stress tests). ZScore's web front end is currently used to render score views on audience's mobile devices. Under the typical load conditions the system is expected to serve up to 100 connected web interface users without any network packet loss and any adverse impact on the internal network score distribution. The maximum allowed web page load time in these tests was set to 5 seconds while the maximum server state data push latency was set to 1 second. The tests included client connectivity over the wired (Ethernet) and wireless (WiFi) connections for both HTTP and Websocket protocols. Furthermore, a set of tests was devised to assess the impact of the firewall filtering on the system throughput. The testing scope did not include wireless access point range tests and Open Sound Control UDP messaging to musicians' front ends.

Testing Environment

ZScore web server is currently integrated into the monolithic ZScore application written in Java (1.8). The server utilises Undertow library (v2.0.1) to provide HTTP, Websockets and SSE service. In this test the ZScore application was run on 16" MacBook Pro laptop connected to the Mikrotik *hAP ac* router via Ethernet cable, as illustrated in Figure 1. The test clients (custom Java and Apache jmeter) were run on a separate MacBook Pro laptop (Figure 1). The testing client laptop was either wired or connected wirelessly to the router, depending on a test scenario.

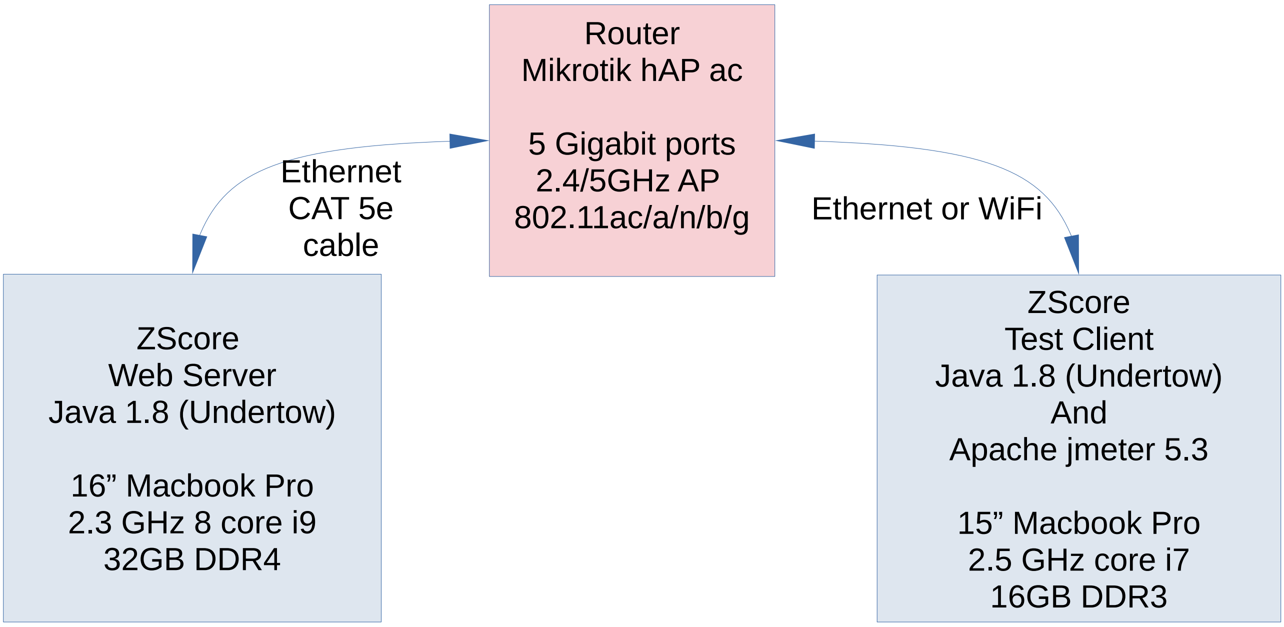
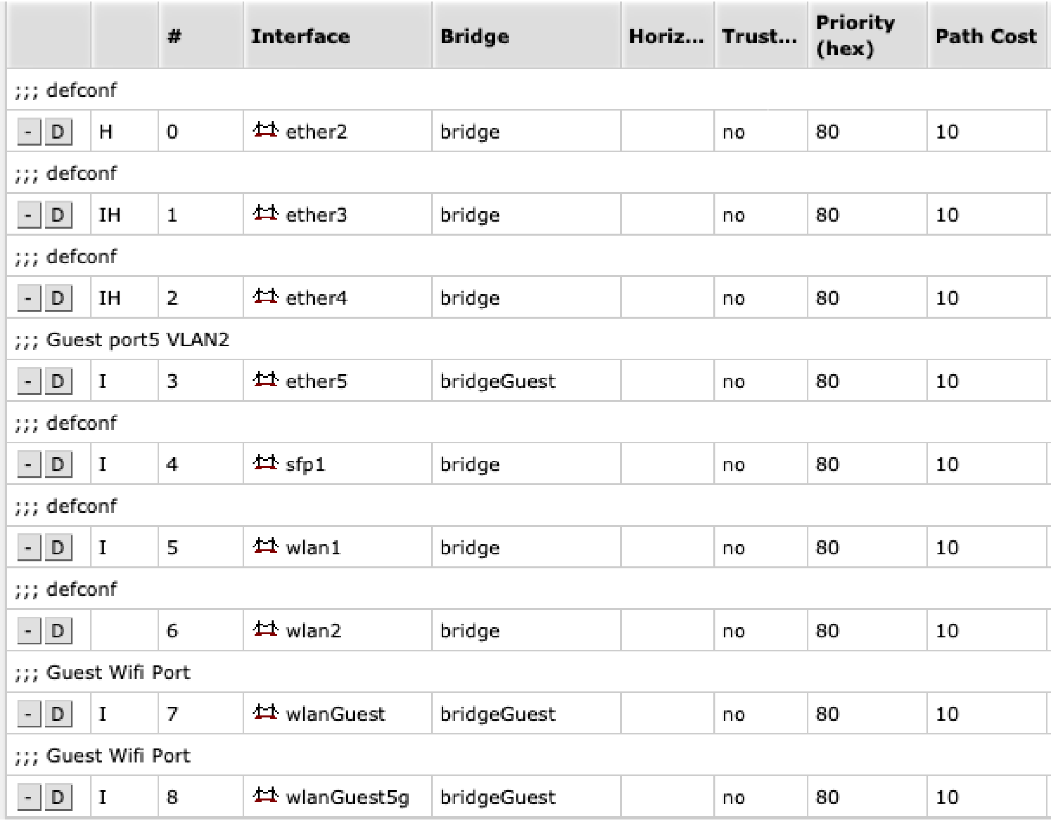


Figure 1: Test Environment

Router setup

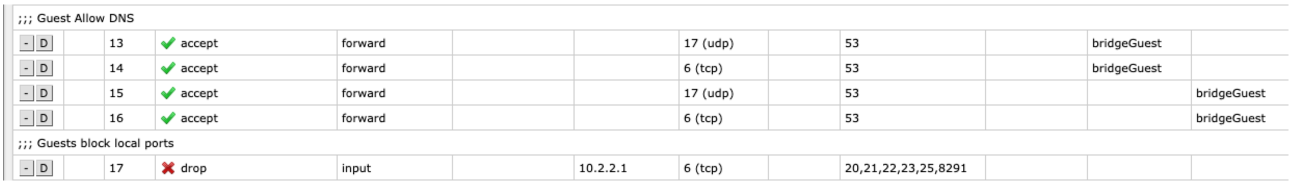
In a public performance situation audience's mobile device access to ZScore's internal network presents a potential security risk. A network firewall is an industry standard protection mechanism which isolates internal network and limits external users' access. In order to assess impact of the router's firewall on both wired and wireless connections a separate network bridge (guestBridge) with its own IP address subnet range was created on the router. One of the ethernet ports on the router (port 5) and both virtual wireless Guest WLANs (wlanGuest and wlanGuest5g) were assigned to the created guest bridge (Figure 2).



		#	Interface	Bridge	Horiz...	Trust...	Priority (hex)	Path Cost
;;; defconf								
- D	H	0	ether2	bridge		no	80	10
;;; defconf								
- D	IH	1	ether3	bridge		no	80	10
;;; defconf								
- D	IH	2	ether4	bridge		no	80	10
;;; Guest port5 VLAN2								
- D	I	3	ether5	bridgeGuest		no	80	10
;;; defconf								
- D	I	4	sfp1	bridge		no	80	10
;;; defconf								
- D	I	5	wlan1	bridge		no	80	10
;;; defconf								
- D		6	wlan2	bridge		no	80	10
;;; Guest Wifi Port								
- D	I	7	wlanGuest	bridgeGuest		no	80	10
;;; Guest Wifi Port								
- D	I	8	wlanGuest5g	bridgeGuest		no	80	10

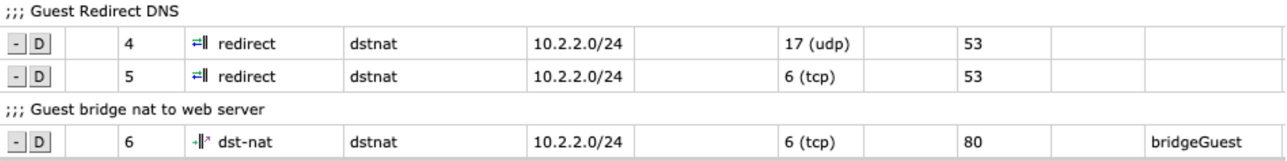
Figure 2: Router bridge ports assignment

A set of firewall rules was then added to prevent guest bridge access to all standard ports on the internal network except DNS (port 53) and HTTP (port 80). Additionally, a network address translation (NAT) rule was added to route all HTTP requests coming from the guest bridge subnet to the ZScore server (Figures 3 and 4).



;;; Guest Allow DNS												
- D		13	✓ accept	forward			17 (udp)		53		bridgeGuest	
- D		14	✓ accept	forward			6 (tcp)		53		bridgeGuest	
- D		15	✓ accept	forward			17 (udp)		53			bridgeGuest
- D		16	✓ accept	forward			6 (tcp)		53			bridgeGuest
;;; Guests block local ports												
- D		17	✗ drop	input		10.2.2.1	6 (tcp)		20,21,22,23,25,8291			

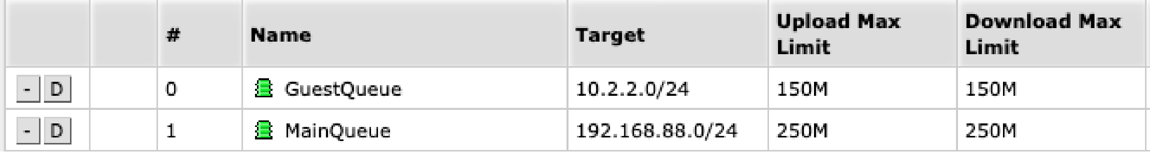
Figure 3: Guest bridge firewall rules



;;; Guest Redirect DNS											
- D		4	redirect	dstnat	10.2.2.0/24		17 (udp)		53		
- D		5	redirect	dstnat	10.2.2.0/24		6 (tcp)		53		
;;; Guest bridge nat to web server											
- D		6	dst-nat	dstnat	10.2.2.0/24		6 (tcp)		80		bridgeGuest

Figure 4: Guest bridge NAT rules

In order to limit the bandwidth available to guest users and prevent the congestion on the network a couple of router queues (one for the internal and one for the guest connections) were created (Figure 5). Both queues were setup to use SFQ (Stochastic Fairness Queuing) algorithm which ensures fair round-robin distribution to all sub-streams. Guest users were limited to 200Mbps overall bandwidth after the initial testing and calculations showed that it was a sufficient limit for the typical use case scenario.



		#	Name	Target	Upload Max Limit	Download Max Limit
- D		0	GuestQueue	10.2.2.0/24	150M	150M
- D		1	MainQueue	192.168.88.0/24	250M	250M

Figure 5: Router Queue setup

Wifi setup

Mikrotik hAP ac router provides 2.4GHz b/g/n and 5GHz a/n modes. In this test both WLANs (2.4 and 5GHz wireless LANs) were set to the same SSID, so the test client's wireless adapter was left in charge of the best connection mode. In a real-life performance scenario audience mobile devices will be allowed to select wireless connection mode in order to simplify user connection process.

The router's channel selection setting for both 2.4 and 5GHz WLANs was set to 'auto' (router's algorithm automatically selects the least congested channel). 2.4GHz WLAN channel width was set to 20MHz, while 5GHz WLAN channel width was set to 20/40MHz XX, where the XX mode allows for the automated control channel frequency selection. According to the *Mikrotik* product specifications, maximum theoretical *hAP ac* throughput for *2.4GHz* mode is 450 Mbit/s and for 5GHz it is 1300 Mbit/s. The actual maximum throughput achieved during testing was around 120 Mb/s. This aligned with the expectations as the maximum theoretical throughput for a single wireless stream is

150Mbps. *Mikrotik* AP provides multiple streams (MIMO) so when multiple WiFi clients are connected the overall throughput should increase up to five times (2x at 2.4 and 3x at 5GHz).

It is worth pointing out that there was a significant congestion and interference from the neighbouring wireless access points during testing. 34 access points with better than -75 dBm wireless signal strength were scanned from the testing site. It is unlikely that any performance site would have much worse wireless congestion and interference problems.

Testing Procedure

A number of testing scenarios were created to cover all permutations of the testing scope. These permutations included wired and wireless test client connections on both guest and internal bridge. Each scenario was run five times in order to get average results.

Union Rose score was loaded into the server as a typical example of the scores written for ZScore system. The initial HTTP load for this score was approximately 1.9MB of data. This included required HTML, Javascript, CSS and MP3 files. The average server state data update size was approximately 18KB.

The firewall contained 20 filter rules set up as described above. In the tests where the client was connected through the guest bridge all data had to pass through the firewall. If the test client was connected through the internal bridge the data bypassed the firewall. Wired connections on the internal bridge had the additional advantage of the hardware switch optimisation.

For the wireless tests, the client host was positioned approximately 2 meters from the access point. As described above, the scope did not include wireless range testing.

Two types of testing clients were used: custom Java (Undertow) and Apache *jmeter*.

Jmeter tests

ZScore’s web client uses HTTP polling mechanism when neither SSE nor Websocket transports are available on the client side. Jmeter tests cover this worst case scenario. The client’s behaviour was first recorded through the Jmeter proxy, starting with the index page retrieval which included HTML, Javascript, CSS and MP3 content (1.9MB overall). The captured scenario then issued 23 HTTP polling requests for the server state data in 500 millisecond intervals. Each server data state update size was around 18KB. The ramp-up time to reach desired client number was set to 10 sec. After the ramp-up time all clients simultaneously issued recorded requests. Jmeter tests were primarily used to stress test ZScore server and find out the system limits.

Custom Undertow client tests

The custom Java tests included the typical HTTP client content retrieval and Websocket server data push scenarios.

HTTP tests first created the specified number of HTTP clients and then issued index HTML page request for each client which downloaded the initial content (HTML, Javascript, CSS and MP3 files; overall size ~ 1.9MB). The index page requests were staggered to mimic real-life scenario. A random number of clients (between 1 and 5) requested the index page at the same time. After a random time interval (between 0 and 1000 seconds) a next group of clients was randomly selected to send their requests. This process was repeated until all test clients submitted index page requests. For 200 HTTP clients the average duration of the test was around 50 seconds.

Likewise, Websocket tests first initialised the required number of clients and then listened for the server state updates. For these tests ZScore server replayed server side events for the first 16 beats of *Union Rose.* This included 8 server state data updates, each of approximately 18KB size. A single test lasted approximately 12 seconds.

Findings and Analysis

Generally, the test results confirmed expected performance patterns. The most important findings can be outlined as follows:

- In its current form, **ZScore server** can cope with a significantly higher load than the typical use case scenario. Test results show that network data back pressure and packet loss start at around **1500 concurrent users**.
- Router's **switch** can successfully serve **200** concurrent **wired connections** going through the **firewall**, without any packet loss and the latency below the test criteria (HTTP: average 1.03, 90th percentile 2.1, max 4.4 sec; Websocket: average 0.054, 90th percentile 0.2, max 1.3 sec).
- Router's **wireless AP** can successfully serve up to **100** wireless **users** connected through the **firewall**. The maximum number of Wifi connections that did not suffer any packet loss was 125. However, the max latency for 100 wireless connections reached 9 sec (average: 1.6 sec, 90th percentile: 3.3 sec).
- Firewall filtering significantly impacted wired clients throughput, however, for the **wireless** connections the benefits of a secure **firewall** outweigh a negligible performance difference between open and firewalled connections.
- **Wireless** tests latency **Variance** was almost **4 times higher** then the wired tests variance. The most likely cause for such a large difference was the significant interference from the neighbouring wireless access points during testing, as described above.

The most significant difference between a real-life performance situation and these tests is that, in the real world, there would be multiple hosts (mobile devices) making a single connection to the server. This difference could have multiple consequences. The AP would have to manage and route a number of connections which might impede its performance. On the other hand, router's multiple wireless streams (2 for 2.4GXz and 3 for 5GHz) would be better utilised when multiple client adapters are connected. This could theoretically increase available bandwidth fivefold. The only way to measure this impact would be to have multiple (50+) mobile devices connected to the router's AP in a controlled environment.

In wired HTTP tests it was not possible to connect more than 235 clients and in wireless tests the maximum was 150 clients. The initial suspicion was that this was down to the test laptop's operating system file descriptor limits. However, even after increasing the number of OSX file descriptors the tests failed at the same point. As Jmeter tests managed to create successfully 2000 connections it is likely that this could be a limitation of the Undertow client implementation (multiple non-blocking threads and OS resource utilisation).

ZScore server memory heap size never went over 250MB during the testing, indicating a fairly light memory usage and no significant memory leaks when dealing with the web traffic. The maximum JVM CPU load during testing was 1.5% which, again, indicates fairly low server side load.

The Mikrotik Queue size reached the peak at around 110Mbps for 100 connected users. This aligns with expectations as each connection had to download 1.9MB of initial files and the limitation of the single wireless stream is 150 Mbps. The queue size needs to be reviewed whenever the maximum throughput or a number of connected users change.

In a real-life situation, WiFi user's experience will depend heavily on the performance site's wireless channel congestion, however, it is unlikely that the throughput will be much worse then the testing data numbers, due to the reasons described above.

Scalability Ramifications And Possible Solutions

From the ZScore web user's perspective, two factors can impact the system's performance the most: a size of data to push through the network and a number of connected users. If it is assumed that *Union Rose* is a typical score and the data load used in ZScore system, then the number of connected users becomes critical for system scalability planning. The system design discussion could be framed by the number of expected users (audience members) as follows.

System design for up to 100 WiFi users

Based on the test results, current system architecture should cope well with 100 users connected over Wifi and going through the firewall.

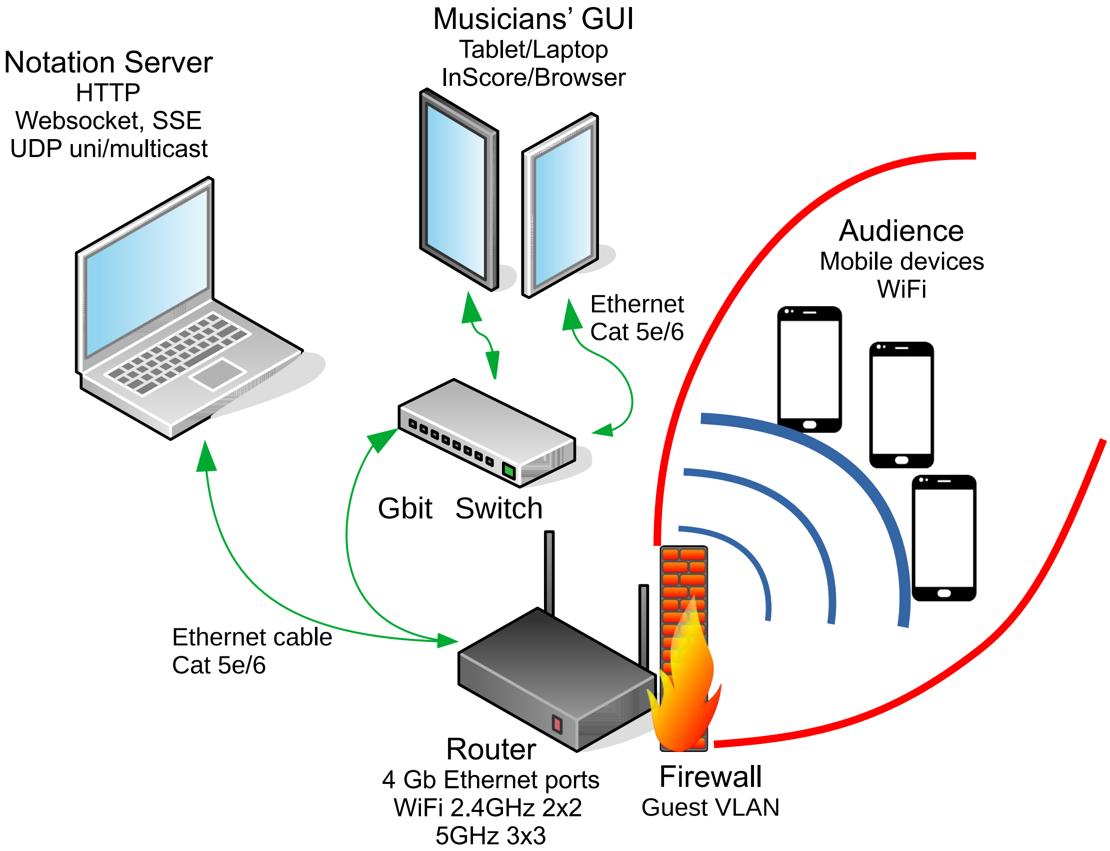


Figure 6: System architecture for up to 100 WiFi users

System design for 200 to ~500 WiFi users

For 200 connected WiFi users the typical required throughput would increase to around 400Mps. Although *Mikrotik's* theoretical wireless throughput is much higher, the tests have shown that additional wireless access points might be required. Based on the testing results, it could be extrapolated that for each additional 100 users one more wireless AP of the similar specifications to the test AP is required. Alternatively, a more powerful MU

MIMO AP (such as Unifi UAP AC HD) could be used to cater for more than 100 users. This expansion could be done until the router's switch becomes a bottleneck. By extrapolating test results it can be concluded that the current router could serve up to 500 users, which would require 4 additional MIMO or 2-3 MU MIMO access points. A system design suggestion for up to 200 connected WiFi users is shown in Figure 7.

Any additional AP needs to be connected to the router via Ethernet cable for stability and sufficient throughput. A shielded CAT 6e Ethernet cable would be a good choice for this purpose as it provides reduced interference and high throughput (up to 10Gb). Furthermore, CAT 6a can feed power over Ethernet (PoE) to AP if required. PoE would simplify system installation as it replaces conventional power cabling.

Additional APs should have similar specifications to *hAP ac* AP (MIMO 2x2 at 2,4GHz and 3x3 at 5Ghz) or better in order to provide required throughput (e.g. Unifi AP AC PRO). The additional AP should be set up with the same guest SSID as the router's AP to simplify user connection procedure. In this design, there are no guarantees where mobile devices are going to connect to, however, most modern WiFi adapters have built-in algorithms to choose an AP with the strongest signal.

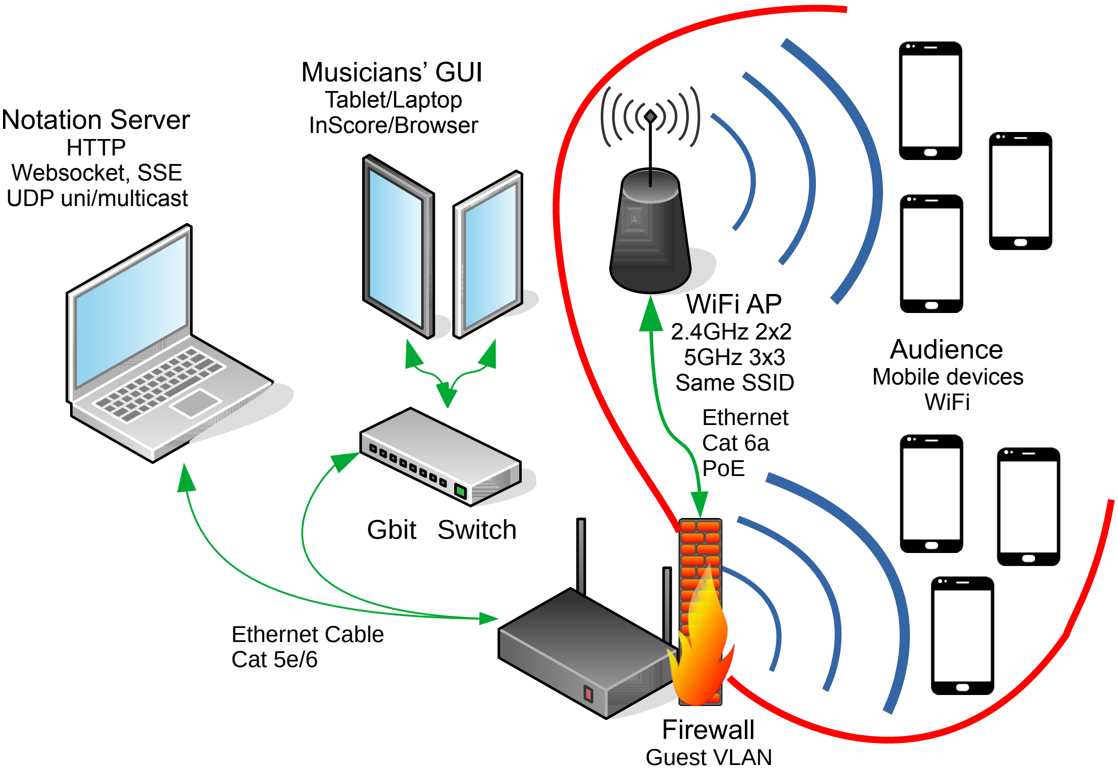


Figure 7: System design for up to 200 WiFi users

A potential issue could arise for audience members positioned in the middle between two access points with similar signal quality. In this case a mobile device WiFi adapter could switch intermittently between two access points. The switching process might take several seconds during which users' interaction with the network would be interrupted. This would create issues for any continuous streaming (audio/video). Websocket and SSE clients should be able to reconnect on mobile device rejoining the network, therefore, ZScore server would synchronise mobile device state on reconnection. Depending on a current content in mobile device's web browser users might not notice AP switching at all. However, the web front end and server state update logic should be designed with this problem in mind.

Wireless mesh vs multiple APs

Another option for the scenario above would be to deploy a wireless mesh solution. Mesh systems automatically negotiate client connectivity between wireless nodes so they provide a superior network client switching logic. However, as all network traffic over a mesh system is wireless, it would create a throughput bottleneck on the router's entry point.

On the other hand, network traffic coming from multiple wired APs can be split through hardware switches thus allowing for much higher aggregated bandwidth. Therefore, multiple APs are a recommended solution where high network throughput is required.

System design for ~500 to ~1500 WiFi users

Assuming that the required bandwidth requirement for 100 users is 200Mbps then 1Gbit router would not be able to cope with the typical network traffic as the number of connected WiFi users approaches 500. An enterprise grade 10Gbit router or a router providing independent Gigabit Ethernet ports (such as Mikrotik CCR1009-7G-1C-1S+PC) is required in this case. Ordinarily, enterprise grade routers do not provide any extra functionality, such as a built-in wireless AP, so the complexity and the cost of the system design would increase.

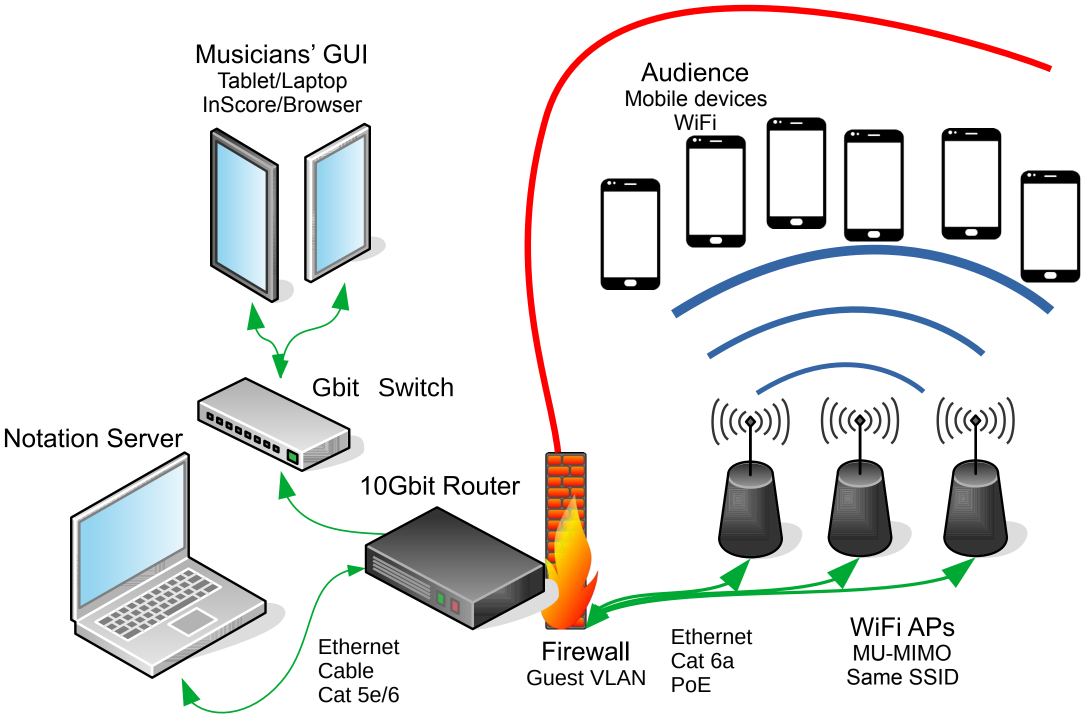


Figure 8: System design for 500 WiFi users

System design for 1500+ WiFi users

Tests have shown that ZScore server can successfully deal with up to 1500 connected users. Once the user number goes over this limit ZScore server becomes the bottleneck. In order to deal with such a load, ZScore server design needs to be modified. Web server needs to be hosted externally and possibly on multiple hosts. A load balancing proxy sitting in front of the web server instances could be utilised to marshal clients connections. Websocket functionality needs to be validated if any connection proxying is used.

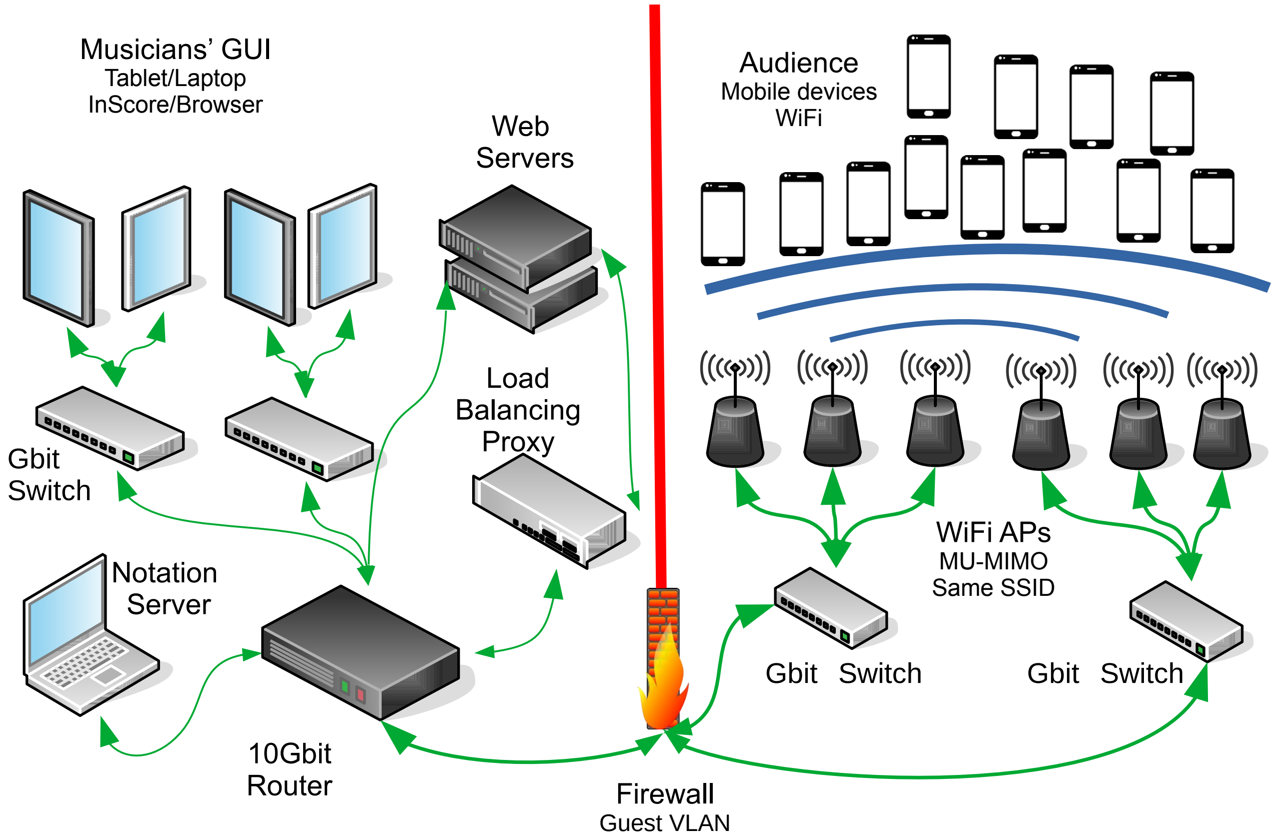


Figure 9: System design for 1000+ WiFi users

If the number of users exceeds 2000 then one 10Gbit router would not be able to cope with the load and a more comprehensive solution with multiple routers connected via optical cables is required. At that point the cost and complexity of the system deployment increases significantly.

Detailed Test Results

Test Case 1: Undertow Websocket, Wired, No Firewall, 200 Clients

					Latency			Percentile		
Test Id	**Client Type**	**Connection type**	**Firewall**	**Client No**	**Avg**	**Min**	**Max**	**90th**	**95th**	**99th**
1	Websocket	Wired	No	200	10	0	476	19	40	420
1	Websocket	Wired	No	200	9	0	471	15	39	316
1	Websocket	Wired	No	200	8	0	471	29	39	188
1	Websocket	Wired	No	200	7	0	464	13	37	195
1	Websocket	Wired	No	200	8	0	574	13	35	145
				AVG	8	0	491	18	38	253

Test Case 2: Undertow Websocket, Wired, No Firewall, 100 Clients

					Latency			Percentile		
Test Id	**Client Type**	**Connection type**	**Firewall**	**Client No**	**Avg**	**Min**	**Max**	**90th**	**95th**	**99th**
2	Websocket	Wired	No	100	8	0	461	6	14	308
2	Websocket	Wired	No	100	11	0	462	6	105	308
2	Websocket	Wired	No	100	8	0	472	4	13	408
2	Websocket	Wired	No	100	15	0	464	5	154	408
2	Websocket	Wired	No	100	6	0	458	2	11	158
				AVG	10	0	463	5	59	318

Test Case 3: Undertow Websocket, Wired, With Firewall, 200 Clients

					Latency			Percentile		
Test Id	**Client Type**	**Connection type**	**Firewall**	**Client No**	**Avg**	**Min**	**Max**	**90th**	**95th**	**99th**
3	Websocket	Wired	Yes	200	48	0	1044	260	401	726
3	Websocket	Wired	Yes	200	39	0	841	141	365	460
3	Websocket	Wired	Yes	200	39	0	960	50	361	570
3	Websocket	Wired	Yes	200	39	0	791	41	363	692
3	Websocket	Wired	Yes	200	35	0	702	77	339	508
				AVG	40	0	868	114	366	591

Test Case 4: Undertow Websocket, Wired, With Firewall, 100 Clients

					Latency			Percentile		
Test Id	**Client Type**	**Connection type**	**Firewall**	**Client No**	**Avg**	**Min**	**Max**	**90th**	**95th**	**99th**
4	Websocket	Wired	Yes	100	30	0	626	24	305	391
4	Websocket	Wired	Yes	100	24	0	390	26	268	355
4	Websocket	Wired	Yes	100	30	0	676	27	303	370
4	Websocket	Wired	Yes	100	29	0	518	63	269	379
4	Websocket	Wired	Yes	100	26	0	390	20	293	377
				AVG	28	0	520	32	288	374

Test Case 5: Undertow Websocket, Wireless, With Firewall, 200 Clients

					Latency			Percentile		
Test Id	**Client Type**	**Connection type**	**Firewall**	**Client No**	**Avg**	**Min**	**Max**	**90th**	**95th**	**99th**
5	Websocket	Wireless	Yes	200	54	0	979	162	527	727
5	Websocket	Wireless	Yes	200	60	0	1755	352	493	864
5	Websocket	Wireless	Yes	200	54	0	1610	176	512	734
5	Websocket	Wireless	Yes	200	49	0	1285	164	443	711
5	Websocket	Wireless	Yes	200	55	0	1232	181	481	836
				AVG	54	0	1372	207	491	774

Test Case 6: Undertow Websocket, Wireless, With Firewall, 100 Clients

					Latency			Percentile		
Test Id	**Client Type**	**Connection type**	**Firewall**	**Client No**	**Avg**	**Min**	**Max**	**90th**	**95th**	**99th**
6	Websocket	Wireless	Yes	100	46	0	745	133	413	582
6	Websocket	Wireless	Yes	100	36	0	621	139	348	472
6	Websocket	Wireless	Yes	100	41	0	823	135	431	478
6	Websocket	Wireless	Yes	100	33	0	678	104	338	436
6	Websocket	Wireless	Yes	100	41	0	514	193	388	482
				AVG	39	0	676	141	384	490

Test Case 7: Undertow Websocket, Wireless, No Firewall, 200 Clients

					Latency			Percentile		
Test Id	**Client Type**	**Connection type**	**Firewall**	**Client No**	**Avg**	**Min**	**Max**	**90th**	**95th**	**99th**
7	Websocket	Wireless	No	200	55	0	1666	154	539	844
7	Websocket	Wireless	No	200	56	0	1808	131	520	946
7	Websocket	Wireless	No	200	64	0	1310	166	559	963
7	Websocket	Wireless	No	200	60	0	1387	201	506	1002
7	Websocket	Wireless	No	200	64	0	1653	204	520	953
				AVG	60	0	1565	171	529	942

Test Case 8: Undertow Websocket, Wireless, No Firewall, 100 Clients

					Latency			Percentile		
Test Id	**Client Type**	**Connection type**	**Firewall**	**Client No**	**Avg**	**Min**	**Max**	**90th**	**95th**	**99th**
8	Websocket	Wireless	No	100	47	0	962	134	451	804
8	Websocket	Wireless	No	100	37	0	1014	94	336	603
8	Websocket	Wireless	No	100	46	0	881	137	464	868
8	Websocket	Wireless	No	100	37	0	882	72	424	502
8	Websocket	Wireless	No	100	29	0	831	117	315	461
				AVG	39	0	914	111	398	648

Test Case 9: Undertow HTTP, Wired, No Firewall, 200 Clients

					Latency			Percentile		
Test Id	**Client Type**	**Connection type**	**Firewall**	**Client No**	**Avg**	**Min**	**Max**	**90th**	**95th**	**99th**
9	HTTP	Wired	No	200	109	69	233	164	186	189
9	HTTP	Wired	No	200	109	69	229	167	196	229
9	HTTP	Wired	No	200	102	60	234	141	168	232
9	HTTP	Wired	No	200	118	63	459	148	268	311
9	HTTP	Wired	No	200	112	64	260	175	188	259
				AVG	110	65	283	159	201	244

Test Case 10: Undertow HTTP, Wired, With Firewall, 200 Clients

					Latency			Percentile		
Test Id	**Client Type**	**Connection type**	**Firewall**	**Client No**	**Avg**	**Min**	**Max**	**90th**	**95th**	**99th**
10	HTTP	Wired	Yes	200	1025	182	4722	2095	3045	4675
10	HTTP	Wired	Yes	200	974	201	3678	2062	2788	3359
10	HTTP	Wired	Yes	200	982	190	3786	2173	2514	3592
10	HTTP	Wired	Yes	200	1271	187	6506	2541	4390	5993
10	HTTP	Wired	Yes	200	942	182	3553	2008	2841	3549
				AVG	1039	188	4449	2176	3116	4234

Test Case 11: Undertow HTTP, Wireless, With Firewall, 100 Clients

					Latency			Percentile		
Test Id	**Client Type**	**Connection type**	**Firewall**	**Client No**	**Avg**	**Min**	**Max**	**90th**	**95th**	**99th**
11	HTTP	Wireless	Yes	100	1335	273	4739	2657	3335	4739
11	HTTP	Wireless	Yes	100	2357	270	6209	4650	5151	6209
11	HTTP	Wireless	Yes	100	1134	251	3372	2155	2730	3372
11	HTTP	Wireless	Yes	100	1465	271	4492	2808	3108	4492
11	HTTP	Wireless	Yes	100	2073	322	9220	4521	6420	9220
				AVG	1673	277	5606	3358	4149	5606

Test Case 12: Undertow HTTP, Wireless, With Firewall, 50 Clients

					Latency			Percentile		
Test Id	**Client Type**	**Connection type**	**Firewall**	**Client No**	**Avg**	**Min**	**Max**	**90th**	**95th**	**99th**
12	HTTP	Wireless	Yes	50	2190	318	5719	3935	5054	5719
12	HTTP	Wireless	Yes	50	895	274	2543	1662	1770	2543
12	HTTP	Wireless	Yes	50	1070	280	2129	1882	1919	2129
12	HTTP	Wireless	Yes	50	1570	304	5810	2952	3328	5810
12	HTTP	Wireless	Yes	50	3105	270	7162	5600	6292	7162
				AVG	1766	289	4673	3206	3673	4673

Test Case 13: Undertow HTTP, Wireless, No Firewall, 100 Clients

					Latency			Percentile		
Test Id	**Client Type**	**Connection type**	**Firewall**	**Client No**	**Avg**	**Min**	**Max**	**90th**	**95th**	**99th**
13	HTTP	Wireless	No	100	2637	382	6299	4714	5348	6299
13	HTTP	Wireless	No	100	1000	241	3901	2091	2805	3901
13	HTTP	Wireless	No	100	4279	606	8522	6970	8154	8522
13	HTTP	Wireless	No	100	3051	258	8278	5514	6535	8278
13	HTTP	Wireless	No	100	2751	284	8623	4785	5553	8623
				AVG	2744	354	7125	4815	5679	7125

Test Case 14: Undertow HTTP, Wireless, No Firewall, 100 Clients

					Latency			Percentile		
Test Id	**Client Type**	**Connection type**	**Firewall**	**Client No**	**Avg**	**Min**	**Max**	**90th**	**95th**	**99th**
14	HTTP	Wireless	No	50	2112	257	5598	4869	4987	5598
14	HTTP	Wireless	No	50	1018	321	3014	1857	2338	3014
14	HTTP	Wireless	No	50	701	244	2987	1372	1576	2987
14	HTTP	Wireless	No	50	684	255	2892	1215	1321	2892
14	HTTP	Wireless	No	50	1014	271	5114	1814	1821	5114
				AVG	1106	270	3921	2225	2409	3921

Test Case 15: Jmeter, Wired, No Firewall, 100 Clients

100 Users, Wired, No Firewall				Latency milliseconds				Percentile		
Label	**Sample No**	**KO**	**Error %**	**Average**	**Min**	**Max**	**Median**	**90th**	**95th**	**99th**
/htp-112	100	0	0.00%	3.46	2	6	3	4	4.95	5.99
/htp-113	100	0	0.00%	3.36	2	5	3	4	4	5
/htp-114	100	0	0.00%	3.32	2	5	3	4	4	4.99
/htp-115	100	0	0.00%	3.22	2	9	3	4	4	8.99
/htp-116	100	0	0.00%	2.97	2	4	3	4	4	4
/htp-117	100	0	0.00%	2.95	2	4	3	4	4	4
/htp-118	100	0	0.00%	2.93	2	5	3	4	4	4.99
/htp-119	100	0	0.00%	2.81	2	5	3	3	4	5
/htp-120	100	0	0.00%	2.76	2	5	3	4	4	4.99
/htp-121	100	0	0.00%	2.91	2	27	3	3	4	26.77
/htp-122	100	0	0.00%	2.71	2	8	3	3	4	7.96
/htp-123	100	0	0.00%	2.69	2	4	3	3	3	4
/htp-124	100	0	0.00%	2.63	2	4	3	3	4	4
/htp-125	100	0	0.00%	2.54	1	5	3	3	3	4.99
/htp-126	100	0	0.00%	2.63	1	12	2.5	3	3.95	11.93
/htp-127	100	0	0.00%	2.51	1	7	2	3	4	6.98
/htp-128	100	0	0.00%	2.54	1	6	2	3	3	6
/htp-129	100	0	0.00%	2.45	1	7	2	3	3	6.97
/htp-130	100	0	0.00%	2.51	1	4	2	3	3.95	4
/htp-131	100	0	0.00%	2.45	2	4	2	3	3	4
/htp-132	100	0	0.00%	2.46	2	3	2	3	3	3
/htp-133	100	0	0.00%	2.48	2	6	2	3	3.95	5.99
/htp-134	100	0	0.00%	2.4	2	5	2	3	3	5
Polling Avg	2300	0	0	2.77	2	7	3	3	4	7
/test.html-103	100	0	0.00%	19.52	14	62	19	23	24	62

Test Case 16: Jmeter, Wired, No Firewall, 500 Clients

500 Users, Wired, No Firewall				Latency milliseconds				Percentile		
Label	**Sample No**	**KO**	**Error %**	**Average**	**Min**	**Max**	**Median**	**90th**	**95th**	**99th**
/htp-112	500	0	0.00%	6.85	1	133	3	14	28.9	66.97
/htp-113	500	0	0.00%	6.64	1	142	3	13	24.95	69.9
/htp-114	500	0	0.00%	6.16	1	78	4	12	19.95	59.99
/htp-115	500	0	0.00%	6.41	2	211	3	12	17	52.95
/htp-116	500	0	0.00%	5.91	1	148	3	11	15.95	59.89
/htp-117	500	0	0.00%	5.87	1	156	3	11	19.95	49.96
/htp-118	500	0	0.00%	6.4	2	272	3	11	19	68.86
/htp-119	500	0	0.00%	5.86	1	167	3	12	17	48
/htp-120	500	0	0.00%	6.55	1	272	3	11	19.95	65.86
/htp-121	500	0	0.00%	7.39	1	316	3	13	26.9	68.89
/htp-122	500	0	0.00%	6.88	1	273	3	12	19.95	67.94
/htp-123	500	0	0.00%	6.8	1	321	3	12	22	61.84
/htp-124	500	0	0.00%	6.23	1	263	3	11.9	20.95	55.99
/htp-125	500	0	0.00%	5.54	1	79	3	13	18	41.99
/htp-126	500	0	0.00%	5.71	1	83	3	13	21	50.85
/htp-127	500	0	0.00%	5.64	1	139	3	12	17	56.97
/htp-128	500	0	0.00%	5.59	1	202	3	10	15	45.98
/htp-129	500	0	0.00%	5.58	1	108	3	11	18	44
/htp-130	500	0	0.00%	5.92	1	87	3	12	21.95	65.96
/htp-131	500	0	0.00%	6.29	1	136	3	11	21	62.99
/htp-132	500	0	0.00%	6.51	1	194	3	10.9	20.95	72.94
/htp-133	500	0	0.00%	6.58	1	302	3	10	21	51.98
/htp-134	500	0	0.00%	7.29	2	204	3	11	23.95	104.96
Polling Avg	11500	0	0	6.29	1	186	3	12	20	61
/test.html-103	500	0	0.00%	30.4	14	421	18	50	88	268

Test Case 17: Jmeter, Wired, No Firewall, 1000 Clients

1000 Users, Wired, No Firewall				Latency milliseconds				Percentile		
Label	**Sample No**	**KO**	**Error %**	**Average**	**Min**	**Max**	**Median**	**90th**	**95th**	**99th**
/htp-112	1000	0	0.00%	173.1	11	557	161	264.5	335.9	452.96
/htp-113	1000	0	0.00%	163.69	4	625	154	239	283.95	436.86
/htp-114	1000	0	0.00%	161.56	20	848	150	233	286.8	411.94
/htp-115	1000	0	0.00%	157.86	18	580	148	228	274.85	420.93
/htp-116	1000	0	0.00%	155.36	37	575	144	229.9	275.9	403.97
/htp-117	1000	0	0.00%	152.35	26	574	145	223	250.95	405.99
/htp-118	1000	0	0.00%	154.8	34	611	145	227.9	267.95	409.97
/htp-119	1000	0	0.00%	152.19	4	542	145	223	268.85	399.99
/htp-120	1000	0	0.00%	154.82	5	621	147	227.9	289.5	440.88
/htp-121	1000	0	0.00%	150.88	6	576	146	223	255.9	388.97
/htp-122	1000	0	0.00%	153.04	5	576	148	220	265.75	410.96
/htp-123	1000	0	0.00%	152.84	29	554	148	221.9	253.9	408.94
/htp-124	1000	0	0.00%	151.98	14	633	147	220	270.95	418.8
/htp-125	1000	0	0.00%	153.16	17	519	151	217	260.95	410.91
/htp-126	1000	0	0.00%	147.35	3	598	146	211.9	245.95	395.95
/htp-127	1000	0	0.00%	148.74	20	737	143.5	223	259.75	447.94
/htp-128	1000	0	0.00%	141.63	5	555	141	209	241.9	384.99
/htp-129	1000	0	0.00%	139.2	3	598	139	208	236	378.84
/htp-130	1000	0	0.00%	138.32	3	438	141	207.8	236.85	351.92
/htp-131	1000	0	0.00%	135.42	2	488	141	205.9	222.95	314.99
/htp-132	1000	0	0.00%	140.73	2	512	144.5	214.9	249.9	402.98
/htp-133	1000	0	0.00%	141.6	3	586	144	219.9	261.95	417.94
/htp-134	1000	0	0.00%	141.19	2	605	144.5	218.9	253.9	400.99
Polling Avg	23000	0	0	150.51	12	587	146	222	263	405
/test.html-103	1000	0	0.00%	2584.36	76	5338	2382	4982	5041	5246

Test Case 18: Jmeter, Wired, No Firewall, 1500 Clients

1500 Users, Wired, No Firewall				Latency milliseconds				Percentile		
Label	**Sample No**	**KO**	**Error %**	**Average**	**Min**	**Max**	**Median**	**90th**	**95th**	**99th**
/htp-112	1500	11	0.73%	430.89	3	17031	186.5	361.7	583.45	8433.16
/htp-113	1500	0	0.00%	275.61	14	16491	182	307	355.95	1316.99
/htp-114	1500	0	0.00%	213.16	4	1697	171	323	374	1309.99
/htp-115	1500	0	0.00%	213.58	4	1514	165	331.8	397	1307.99
/htp-116	1500	0	0.00%	197.58	3	1711	155	302	378.7	1233.98
/htp-117	1500	0	0.00%	170.9	4	1534	142	266.9	328	573.95
/htp-118	1500	0	0.00%	172.51	3	1697	138	249	325.95	1232.99
/htp-119	1500	0	0.00%	172.29	4	1710	133	251.8	324.95	1308
/htp-120	1500	0	0.00%	153.7	3	1704	127	228	282	1219.98
/htp-121	1500	0	0.00%	141.66	3	1315	124	219.9	274.9	407.9
/htp-122	1500	0	0.00%	141.12	3	1322	122	211.9	264.85	453.34
/htp-123	1500	0	0.00%	134.42	3	1518	120	201	248	339.95
/htp-124	1500	0	0.00%	134.92	4	1309	121	201	242.95	369.97
/htp-125	1500	0	0.00%	134.79	4	1307	123	197	230	396.82
/htp-126	1500	0	0.00%	134.91	4	1307	124	202.9	236.95	354
/htp-127	1500	0	0.00%	135.34	4	554	126	202.9	229.95	309
/htp-128	1500	0	0.00%	139.91	5	1221	132	213	233	322.98
/htp-129	1500	0	0.00%	144.49	3	1234	137	217	248.9	367.97
/htp-130	1500	0	0.00%	151.26	3	719	143	223	258	470.99
/htp-131	1500	0	0.00%	156.76	3	721	150	227	263	562.93
/htp-132	1500	0	0.00%	166.42	2	803	155	251	306.95	636.97
/htp-133	1500	0	0.00%	170.18	2	848	158	258	328.95	608.96
/htp-134	1500	0	0.00%	173.64	2	847	162	268	342.9	607.99
Polling Avg	34500	11	0	176.52	4	2614	143	248	307	1050
/test.html-103	1500	43	2.87%	7226.27	1	18070	7418	13530	16493	17008

Test Case 19: Jmeter, Wireless, With Firewall, 100 Clients

100 Users, Wireless, With Firewall				Latency milliseconds				Percentile		
Label	**Sample No**	**KO**	**Error %**	**Average**	**Min**	**Max**	**Median**	**90th**	**95th**	**99th**
/htp-112	100	0	0.00%	25.36	6	453	13	51.5	64.7	450.3
/htp-113	100	0	0.00%	53.17	6	445	18	154.7	227.45	444.92
/htp-114	100	0	0.00%	53.6	9	474	19	129.5	334.55	473.8
/htp-115	100	0	0.00%	53.67	11	555	19	125.7	285.25	554.29
/htp-116	100	0	0.00%	50.73	8	501	20	153	212.55	499.55
/htp-117	100	0	0.00%	38.35	7	452	20	53	82.75	451.21
/htp-118	100	0	0.00%	33.38	7	207	21	68	143.85	207
/htp-119	100	0	0.00%	44.72	6	467	20	82.5	193.8	466.79
/htp-120	100	0	0.00%	53.61	8	446	23.5	144.5	181	445.38
/htp-121	100	0	0.00%	57.02	9	705	21.5	128.4	305	702.96
/htp-122	100	0	0.00%	38.16	8	683	22	57.5	101.6	678.71
/htp-123	100	0	0.00%	35.37	8	390	21	56.6	114.1	388.1
/htp-124	100	0	0.00%	41.18	9	374	21	94.5	147.6	372.22
/htp-125	100	0	0.00%	61.4	9	718	25	113.5	355.5	715.42
/htp-126	100	0	0.00%	42.33	7	372	30	67.9	103.75	371.26
/htp-127	100	0	0.00%	44.04	7	459	22	60.6	120.65	458.74
/htp-128	100	0	0.00%	51.71	9	622	20	102.9	236.25	620.59
/htp-129	100	0	0.00%	50.17	6	517	21	88.9	219.4	516.51
/htp-130	100	0	0.00%	50.38	8	675	23	74.9	213.95	673.05
/htp-131	100	0	0.00%	45.14	8	505	20	58.9	191.35	504.96
/htp-132	100	0	0.00%	39.39	7	637	20	49.7	88.8	635.11
/htp-133	100	0	0.00%	33.98	5	333	20	56.6	99.15	331.42
/htp-134	100	0	0.00%	49.71	7	640	19.5	120.6	246.3	637.48
Polling Avg	2300	0	0	45.50	8	506	21	91	186	504
/test.html-103	100	0	0.00%	573.08	121	3254	331	1157	2059	3252

Test Case 20: Jmeter, Wireless, With Firewall, 200 Clients

200 Users, Wireless, With Firewall				Latency milliseconds				Percentile		
Label	**Sample No**	**KO**	**Error %**	**Average**	**Min**	**Max**	**Median**	**90th**	**95th**	**99th**
/htp-112	200	0	0.00%	366.58	15	2502	193	723.9	1388.05	2311.04
/htp-113	200	0	0.00%	393.04	15	4551	176	1008.1	1282.05	3128.61
/htp-114	200	0	0.00%	364.12	18	1900	192.5	932	1122.5	1880.18
/htp-115	200	0	0.00%	349.76	13	1766	203.5	756.6	1191.25	1669.01
/htp-116	200	0	0.00%	366.02	13	1873	199.5	836.8	1171.4	1823.57
/htp-117	200	0	0.00%	368.83	14	3254	171	951.4	1301.45	2351.16
/htp-118	200	0	0.00%	377.87	12	2783	172.5	1019.8	1304.55	2761.02
/htp-119	200	0	0.00%	350.49	12	2446	174	858.4	1003.95	2393.43
/htp-120	200	0	0.00%	377.49	12	2074	208.5	960	1271.75	1864.42
/htp-121	200	0	0.00%	362.61	10	3235	196.5	742.9	1102.3	2105.69
/htp-122	200	0	0.00%	379.71	11	4271	193	878.1	1102.65	2955.25
/htp-123	200	0	0.00%	359.36	10	2338	171	910.8	1259.1	2100.58
/htp-124	200	0	0.00%	316.94	15	1716	159.5	683.9	1290.55	1696.66
/htp-125	200	0	0.00%	366.63	11	2406	169.5	973.1	1204.3	2255.53
/htp-126	200	0	0.00%	289.68	13	1879	148.5	646.3	1011.45	1646.47
/htp-127	200	0	0.00%	283.97	21	1763	148.5	666.5	916.9	1221.24
/htp-128	200	0	0.00%	277.5	15	1680	147.5	658.2	866.9	1510.51
/htp-129	200	0	0.00%	250.61	13	1787	132.5	625.3	841.65	1733.67
/htp-130	200	0	0.00%	239.68	12	1256	141.5	630.7	909.9	1089.9
/htp-131	200	0	0.00%	280.4	12	3121	135.5	603.8	1023	2139.75
/htp-132	200	0	0.00%	247.69	11	2349	124	655	848.15	1850.74
/htp-133	200	0	0.00%	260.35	11	2066	126.5	618.9	1158.3	1802.99
/htp-134	200	0	0.00%	245.23	13	1681	117.5	627.5	815.35	1660.39
Polling Avg	4600	0	0	324.98	13	2378	165	781	1104	1998
/test.html-103	200	0	0.00%	7712.39	460	18398	8300	13253	14035	16356

Test Case 21: Jmeter, Wireless, No Firewall, 100 Clients

100 Users, Wireless, No Firewall				Latency milliseconds				Percentile		
Label	**Sample No**	**KO**	**Error %**	**Average**	**Min**	**Max**	**Median**	**90th**	**95th**	**99th**
/htp-112	100	0	0.00%	48.96	8	549	22	122	231.6	548.61
/htp-113	100	0	0.00%	91.41	8	1536	29	215.2	397.4	1528.32
/htp-114	100	0	0.00%	92.35	7	1329	36	224.7	474.5	1321.85
/htp-115	100	0	0.00%	80.63	7	959	38	203.8	518.95	955.29
/htp-116	100	0	0.00%	64.08	7	699	32	91.5	429.05	697.83
/htp-117	100	0	0.00%	64.47	7	871	33	113.2	246.55	869.02
/htp-118	100	0	0.00%	100.63	7	1196	36	409.8	568.4	1192.94
/htp-119	100	0	0.00%	71.46	7	1045	32	99.2	497.4	1042.65
/htp-120	100	0	0.00%	81.24	7	968	33.5	192.1	527.7	965.69
/htp-121	100	0	0.00%	60.7	6	544	32.5	75	374.45	543.82
/htp-122	100	0	0.00%	75.33	7	1050	34.5	127	502.85	1046.66
/htp-123	100	0	0.00%	80.61	5	984	34	188.7	524.55	979.95
/htp-124	100	0	0.00%	74.83	4	729	33	168	505.7	727.84
/htp-125	100	0	0.00%	94.36	4	681	36	371.4	551.9	680.16
/htp-126	100	0	0.00%	52.98	4	534	34.5	89.2	185.55	532.9
/htp-127	100	0	0.00%	76.04	4	967	36	145.1	366.8	965.39
/htp-128	100	0	0.00%	59.86	4	968	35	83.7	148.85	963.87
/htp-129	100	0	0.00%	56.99	5	559	33	71.2	220.1	558.95
/htp-130	100	0	0.00%	79.95	4	1535	33	147.4	443.95	1530.34
/htp-131	100	0	0.00%	92.06	5	967	33	190.2	531.75	966.94
/htp-132	100	0	0.00%	66.5	4	636	34	119.7	380.15	635.08
/htp-133	100	0	0.00%	57.79	5	541	33	87.4	368.75	540.65
/htp-134	100	0	0.00%	65.35	5	571	32.5	131.4	424.85	570.47
Polling Avg	2300	0	0	73.42	6	888	33	159	410	885
/test.html-103	100	0	0.00%	807	240	2196	661	1460	1590	2196

Test Case 22: Jmeter, Wireless, No Firewall, 200 Clients

200 Users, Wireless, No Firewall				Latency milliseconds				Percentile		
Label	**Sample No**	**KO**	**Error %**	**Average**	**Min**	**Max**	**Median**	**90th**	**95th**	**99th**
/htp-112	200	0	0.00%	346.25	9	2225	103	1049.7	1411.45	2184.22
/htp-113	200	0	0.00%	382.01	15	4290	119.5	965.9	1479.4	3460.38
/htp-114	200	0	0.00%	378.89	18	2912	116.5	1098.5	1614.95	2704.52
/htp-115	200	0	0.00%	335.24	19	3958	105	984	1353.2	2619.35
/htp-116	200	0	0.00%	327.52	16	2820	105.5	1001.5	1275.55	2384.91
/htp-117	200	0	0.00%	354.98	15	4894	100	1017.4	1227.05	2774.12
/htp-118	200	0	0.00%	331.85	22	3192	122.5	938.1	1244.6	2116.89
/htp-119	200	0	0.00%	438.9	14	4717	110	1178.1	1751.5	4664.63
/htp-120	200	0	0.00%	352.19	15	3592	106.5	959	1212.8	2518.44
/htp-121	200	0	0.00%	333.76	19	4177	103.5	963.8	1202.85	3108.28
/htp-122	200	0	0.00%	308.24	13	3629	114	715.8	1101.8	3391.06
/htp-123	200	0	0.00%	401.42	9	5894	109.5	1051.6	1667.7	5000.55
/htp-124	200	0	0.00%	316.66	8	4482	95.5	793.6	1366.65	3611.46
/htp-125	200	0	0.00%	288.8	7	2158	98	793.4	1119.95	1822.7
/htp-126	200	0	0.00%	319	6	5891	102	878.2	1123.25	2247.85
/htp-127	200	0	0.00%	290.3	8	2341	92	711.1	1309.4	2159.25
/htp-128	200	0	0.00%	259.7	9	3941	91	591.5	1098.3	3381.09
/htp-129	200	0	0.00%	299.42	6	3170	100.5	977.7	1173.4	2329.67
/htp-130	200	0	0.00%	272.29	5	3056	94.5	696	1094.95	2174.54
/htp-131	200	0	0.00%	265.19	6	3016	84	678.9	1249.4	2524.31
/htp-132	200	0	0.00%	243.72	5	3769	86.5	716.8	1046.65	2057.21
/htp-133	200	0	0.00%	247.42	5	2898	89.5	611.5	1087.75	2843.65
/htp-134	200	0	0.00%	186.91	4	3111	81	557.8	582.8	1956.43
Polling Avg	4600	0	0	316.55	11	3658	101	867	1252	2784
/test.html-103	200	0	0.00%	5697	267	14238	4675	11099	12383	13315