**6.1 Set up Metrics**

On experiment to evaluate overhead, we use iMac 2015 version to construct a server agent, both client and ISP are built on Ubuntu 16.04 LTS Intel Core i7-6700 16GB Memory, and server also uses Ubuntu as operating system and uses nginx http server as our CP server. Test tool is wget and file size is 10 MB if measure the download time.

On experiment to evaluate scalability, we use same method to build a server agent, client and ISP, also server is constructed on Ubuntu as before, but now is a TCP server programmed by boost asio library. When testing end-to-end delay, we transport a single packet in network traffic between server and client. When testing download time, we transport a 10MB random data and measure its transport time.

**6.2 Overhead**

**6.2.1 In Data Plane**

Control plane contains hash matching process and will cause packet transmission slower than in traditional network protocol, such as TLS or TCP, even though packet transmitted in data plane not spend so much time, control plane transport are obviously slower as we can see in Table 1, nearly two times time-consuming than data plane, and data plane has to wait for control plane matching outcome even though a packet already be mirrored on ISP and stored there, that is the reason cause overhead on ZFree.

Figure 1-(a) shows a concrete example about data plane overhead when we use TLS and ZFree blocking and non-blocking versions to loading top 500 websites in Alexa, y axis shows cumulative probability that loading time no larger than corresponding x axis values. We can see directly that non-blocking mode and TLS have almost same loading time, which the average loading overhead of non-blocking mode to TLS is just 3.08 milliseconds in our experiment. Also, blocking mode of ZFREE has a small overhead since for each smaller x-axis value, the accumulative probability always smaller for ZFREE blocking mode, in which the average overhead is 73.11 milliseconds comparing to TLS. We can observe result that ZFree non-blocking mode has very close performance with TLS, but blocking mode has obviously overhead when comparing with TLS. Following evaluation will holds similar conclusion.

When encrypting messages in network channel, we use TLS as comparison and implement TLS on ZFree non-blocking and blocking modes. Also, we use unencrypted models on all three above, in which TLS is reduced as bare TCP connection. Figure 1-(b) presents evaluation outcomes on all six models above about the effect of bandwidth to end-to-end delay, as bandwidth becomes larger and larger, limitation in network channel is less and less thus end-to-end delay keeps decreasing. It is clearly non-blocking mode always has close performance with TLS when all messages are encrypted and also with TCP when we not use any encryption methods. However, no matter with encryption or not, ZFree blocking mode always has overhead no larger than 20% with TLS or TCP, that is also what the effect from control plane to data plane transmission.

Furthermore, we test the effect of machine capability by increasing the execution capability of CPU and run ZFREE to download a 10MB file, also limit bandwidth in 1MB and do measurement. Execution capability ratio can only be set no smaller than 40\% thus we begin from 40\% to 100\% and see the performance of encrypted blocking mode of ZFREE. As we can see in figure 1-(c), download time keeps obviously decreasing in seconds. This outcome is expectable since the more ratio of execution capability allocate to ZFREE, the quicker it performances to download a file. Also, we can see clearly download time decreases dramatically when execution capability starts increasing and not so high, when CPU is almost occupied by ZFREE, it is reasonable that download time

**6.2.2 In Control Plane**

ZFree is a CUP friendly and memory saving program on practical system, concrete codes are running on Ubuntu 16.04 LTS, holding 15.6 GiB memory and implemented with Intel Core i7-6700 CPU. Our outcome shows only 15% CPU occupancy rate and 1.4% memory occupancy, thus implementing and running ZFree is not an ordeal for system.

Traditional network traffic is only a single plane between client and server, all messages can be transported directly in this channel. ZFree has two planes, in which the data plane is what like traditional network mode and added with control plane, thus actually two planes of traffic in same channel and causes network overhead, but this overhead is not big because control plane uses hash function dealing with each packet, which will make network communication much more easier. In our experiment, an average of data plane size is 1514 Byte and control plane size is 74 Byte, this difference comes from hash process thus significantly saving space in network channel, overhead in network is due to the packet size in control plane, which ratio is 74/1514 = 4.888%, a very minor overhead for transmitted packets thus cannot cause obvious burden in network channel.

**6.3 ZFree Scalability**

Practical network communication not only limited in one client to one server communication, also not only limited in single TCP connection in network traffic, thus our ZFree should ensure valid in multi-connection or multi-client situation. Figure 3-(a) shows end-to-end delay in different number of TCP connection, we only transmit a single packet to compare different time for each model. We can see overhead is almost a constant between ZFree blocking mode and TCP no matter encrypted or unencrypted, if encrypted, TCP will become TLS connection. For each packet send from serve, the mirror, hashing, transmission and matching processed in control plane takes almost a constant time, thus overhead is invariant. Also, if we choose non-blocking mode in ZFree, which removes the control plane, transmitted packet only be got and transferred on each participator, this process will not cause obvious overhead thus we can see non-blocking mode has similar delay with TCP in unencrypted condition and with TLS in encrypted condition.

Similar issues happened in Figure 3-(b), in which an almost constant overhead exist between each two lines when we download a 10MB file, this graph is all about the effect comes from just number of connection, we not care about the client number. In Figure 3-(c), all six lines presents almost same download time when number of clients keeps increasing, here we use only a single TCP connection for each client, and also download a 10MB file in our experiment. Since overhead is very small, which is millisecond level as Figure 3-(a) shows, thus this minor overhead is too small to be obvious in graph thus all models show a very near download time.

**6.4 Security**

**6.4.1 URL Attack**

In general URL attack, attacking program firstly modify host field in HTTP request, making Gateway has error judge to believe this host field is a zero-rating one. What actually need to visit is the host analyzed by DNS, thus if user install a client agent which could change host field, user could visit any website with zero-rating service. However, this kind of attack is invalid under ZFree, since malicious proxy or malicious server does not have server agent, thus illegal data stream will be intercepted by ISP after error matching.

**6.4.2 Content Modifier Attack**

In general, unencrypted data stream can be modified by malicious proxy. More details, when a packet arrive to a malicious proxy, proxy can modify data section in this packet, thus data integrity will be broken. Under ZFree model, if we do this type of attack, no matter a packet be modified or not, it will be hashed by server agent and ISP separately, if packet has been modified during transportation between server and client traffic, server agent will have different hash outcome with ISP, which cause this packet be dropped by control plane. Therefore, matching process in control plane could protect data integrity.

**6.4.3 TCP Retransmission Attack**

TCP is a reliable transport protocol, in which when server or client receive packet from each other, they will send an ACK packet back to show they have successfully get this packet. One attack focus on ISP is construct a malicious server who always drop ACK packet, thus server will believe client has not receive packet and keep sending a big amount same packets, these packets called retransmission packet, all these packets will be counted into billing system of user. Under ZFree model, since there is no server agent by malicious server, thus hashed packet matching process will not be passed through in control plane, in this situation ISP will drop all retransmission packets thus protect billing of user.