IP-GEOLOCATION MAPPING FOR MODERATELY CONNECTED INTERNET REGIONS

Abstract:

Most IP-geo location mapping schemes take delay-measurement approach, based on the assumption of a strong correlation between networking delay and geographical distance between the targeted client and the landmarks. In this paper, however, we investigate a large region of moderately connected Internet and find the delay-distance correlation is weak. But we discover a more probable rule—with high probability the shortest delay comes from the closest distance. Based on this closest-shortest rule, we develop a simple and novel IP-geo location mapping scheme for moderately connected Internet regions, called GeoGet. In GeoGet, we take a large number of web servers as passive landmarks and map a targeted client to the geo location of the landmark that has the shortest delay. We further use JavaScript at targeted clients to generate HTTP/Get probing for delay measurement. To control the measurement cost, we adopt a multistep probing method to refine the geo location of a targeted client, finally to city level. The evaluation results show that when probing about 100 landmarks, GeoGet correctly maps 35.4 percent clients to city level, which outperforms current schemes such as GeoLim [16] and GeoPing [14] by 270 and 239 percent, respectively, and the median error distance in GeoGet is around 120 km, outperforming GeoLim and GeoPing by 37 and 70 percent, respectively.

Existing system:

Traditional IP-geo location mapping schemes are primarily delay-measurement based. In these schemes, there are a number of landmarks with known geo locations. The delays from a targeted client to the landmarks are measured, and the targeted client is mapped to a geo location inferred from the measured delays. However, most of the schemes are based on the assumption of a linear correlation between networking delay and the physical distance between targeted client and landmark.
Disadvantages of existing system:

The strong correlation has been verified in some regions of the Internet, such as North America and Western Europe. But as pointed out in the literature, the Internet connectivity around the world is very complex, and such strong correlation may not hold for the Internet everywhere.

Proposed system:

In this paper, we investigate the delay-distance relationship in a particular large region of the Internet (China), where the Internet connectivity is moderate. The data set contains hundreds of thousands of (delay, distance) pairs collected from thousands of widely spread hosts. We have two observations from the data set. First, the linearity between the delay and distance in this region of Internet is positive but very weak. Second, with high probability the shortest delay comes from the closest distance, and we call this phenomenon the “closest-shortest” rule.

Based on the observations, we develop a simple yet novel IP-geo location mapping scheme for moderately connected Internet regions, called GeoGet. In GeoGet, we map the targeted client to the geo location of the landmark that has the shortest delay. We take a large number of web servers with wide coverage and known geo locations as passive landmarks, which eliminates the deploying cost of active landmarks. We further use JavaScript at targeted clients to generate HTTP/Get probing for delay measurement, eliminating the need to install client-side software. To control the measurement cost, we step-by-step refine the geo location of a targeted client, down to city level. In practice, GeoGet can be deployed in combination with a certain locality-aware application such that the application can easily obtain the geo locations of their clients.

Advantages of proposed system:

The contributions of this paper are twofold. First, by studying a large data set, we show that most of the traditional IP-Geo location mapping schemes cannot work well for moderately connected Internet regions, since the linear delay-distance correlation is weak in this kind of Internet regions. Second, based on the measurement results (MR), we develop and implement
GeoGet, which uses the closest-shortest rule and works much better than traditional schemes in moderately connected Internet regions. We acknowledge that we are not the first to apply the closest shortest rule and the mapping accuracy of GeoGet is still not very high. However, we go a large step toward developing a better IP-Geo location system for moderately connected Internet regions. We believe the accuracy will improve significantly if probing more landmarks.

System architecture:
Algorithm: landmark selection algorithm

Input: $T.IP$ (/24 IP prefix of the targeted client)
$T.AS$ (AS number of $T.IP$)

// Area-level landmark selection
01 CSET1 = center cities that $T.IP$ has not probed
02 LSET1 = $\emptyset$
03 for each city1 in CSET1
04 $LC1 =$ landmarks in city1 within $T.AS$
05 $LC0 =$ landmarks in city1 within ASes other than $T.AS$
06 if $|LC1| \geq M1$
07 $LC2 =$ randomly select $M1$ landmarks from $LC1$
08 $LSET1 = LSET1 \cup LC2$
09 else
10 $LC2 =$ randomly $(M1 - |LC1|)$ landmarks from $LC0$
11 $LSET1 = LSET1 \cup LC1 \cup LC2$
12 end if
13 end for
14 Assign $LSET1$ for targeted client to probe

// City-level landmark selection
15 CSET1 = P areas to enter city-level probing
16 CSET2 = cities in CSET1 that $T.IP$ has not probed
17 LSET2 = $\emptyset$
18 for each city2 in CSET2
19 $LC1 =$ landmarks in city2 within $T.AS$
20 $LC0 =$ landmarks in city2 within ASes other than $T.AS$
21 if $|LC1| \geq M2$
22 $LC2 =$ randomly select $M2$ landmarks from $LC1$
23 $LSET2 = LSET2 \cup LC2$
24 else
25 $LC2 =$ randomly select $(M2 - |LC1|)$ landmarks from $LC0$
26 $LSET2 = LSET2 \cup LC1 \cup LC2$
27 end if
28 end for
29 Assign $LSET2$ for targeted client to probe
System configuration:

Hardware configuration:

- Processor - Pentium –IV
- RAM - 256 MB(min)
- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA

Software configuration:

- Operating System: Windows XP
- Programming Language: JAVA
- Java Version: JDK 1.6 & above.

REFERENCE: