User Behavior Prediction based Adaptive Policy Pre-fetching Scheme for Efficient Network Management

Yuanlong Cao[1], Jianfeng Guan[1], Wei Quan[1], Jia Zhao[3], Changqiao Xu[1,2], Hongke Zhang[1,3]

**[1] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
Beijing, 100876, China**

**[2] Institute of Sensing Technology and Business, Beijing University of Posts and Telecommunications
Wuxi, Jiangsu 214028, China**

**[3] National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University
Beijing 100044, China**

Abstract

In recent years, network management is commonly regarded as an essential and promising function for managing and improving the security of network infrastructures. However, as networks get faster and network centric applications get more complex, there is still significant ongoing work addressing many challenges of the network management. Traditional passive network censoring systems lack of adaptive policy pre-fetching scheme, as a result, preventing malicious behavior (such as hacker, malware etc.) is big challenging. In this paper, we propose a novel user behavior prediction based adaptive policy pre-fetching scheme for efficient network management. A newly Distributed web User Behavior Prediction model (DUBP) is introduced first to cognize and predict user behavior. It extends a distributed DHT network to fix the bottleneck in traditional Client-Server (C/S) architecture occurred by large-scale network service requesting and massive user log analyzing and calculating. Based on user behavior sensing and prediction provided by DUBP, a further Adaptive Policy Pre-fetching and Caching scheme (APPC) is addressed for fine-grained and efficient network management. Our Universal Network (UN) will employ DUBP and APPC scheme to justify its advantages in secure network service.

Keywords: *network management; user behavior analysis, policy pre-fetch; Universal Network*

1. Introduction

Network censoring is getting increasingly important due to the immense growth of Internet users and service providers, such as Internet Content Providers (ICPs), Internet Service Providers (ISPs) and so on. On the one side, data gathered based passive network censoring has been regarded as the common solution for advanced network censoring and security systems that require fine-grained performance measurements, such as Deep Packet Inspection (DPI) [1]. On the other side, adaptive policy pre-fetching scheme is the key feature for fine-grained and efficient network management.

As networks get faster and network centric applications get more complex, sensing malicious behavior then pre-fetching policy gets more difficult. To solve this problem, log records have been proven effective in detecting and combating these harmful behaviors [2]. As mentioned in a report on data breaches investigated by the Verizon Corp Business Risk team reported in 2008 that 66% of organizations investigated had “sufficient evidence available within their logs to discover the breach had they been more diligent in analyzing such resources” [3]. Actually, there are more and more researches focus on network log [4-5]. As addressed above, it is very clear that logs can play a more important role for security event detection, mitigation and prediction.

However, log server employed in current user behavior analysis and network censoring systems usually base on the traditional C/S architecture. For example, work [6] proposed a central log tracker to collect and analyze large scale of users’ viewing behavior on Video-on-Demand (VoD) streaming. But with more and more user behavior records arising, the traditional C/S tracker server will inevitably become a bottleneck during communicating, caching and analyzing required to process large-scale network service request.

Since balancing the load of file storage and transfer with fully distributed design, Peer-to-Peer (P2P) networks has been widely applied in distributed applications over internet in recent years, such as P2P file sharing [7], P2P Grid computing [8], P2P SIP transfer [9] and multimedia content delivery [10] As the typical structured P2P networks, Distributed Hash Table (DHT) [11] i.e., Chord [12] becomes a promising solution to avoid the flooding search by tightly coupling data or indices of data hereby mechanism that each node has an M-bit identifier by

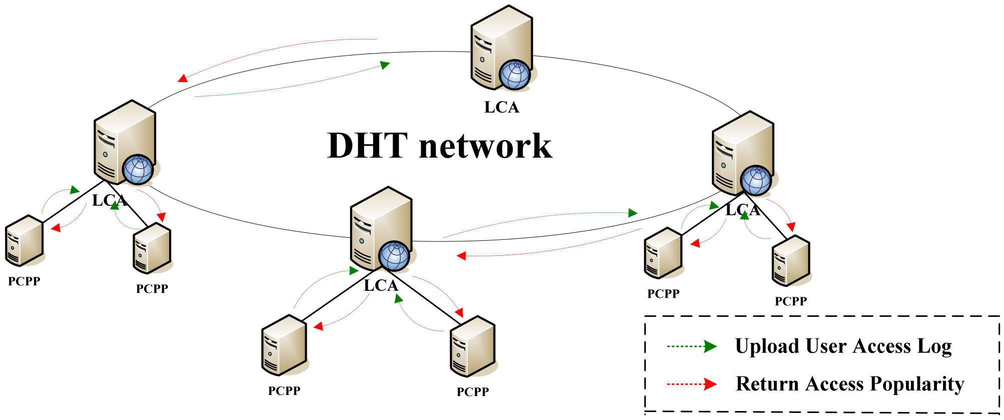


Figure 1. Architecture of the DUBP model

Table I. The description of DUBP model

Component	Description
DHT network	The DHT overlay network is constructed by Chord topology. It consists of LCA. Functions of DHT network include: 1) save user access service logs; 2) compute user behavior popularity; 3) disseminate $\Re$ to PCPP; and 4) self-update once LCA join or leave.
LCA	The LCA's functions involve: 1) receive user access service logs from the PCN; 2) analyze user access service logs and evaluate $\Re$ periodically; 3) share $\Re$ with other LCA in the DHT network; and 4) return $\Re$ to requested PCPP.
PCPP	The PCPP's functions aim to: 1) caputring packet from router or switch etc.; 2) extracting desired log information accordance with specified criteria; 3) uploading log information to its connected LCA. And 4) Request $\Re$ from it's connected LCA then prefetch correspoinding policy for p_i with higher $\Re$ from the PD.

hashing the IP address and other information using a base hash function such as SHA-1 [13].

In this paper, we propose a novel user behavior prediction based adaptive policy pre-fetching scheme for efficient network management. A newly Distributed web User Behavior Prediction model (DUBP) is introduced first to cognize and predict user behavior. It extends a distributed DHT network to fix the bottleneck in traditional Client-Server (C/S) architecture occurred by large-scale network service requesting and massive user log analyzing and calculating. Based on user behavior sensing and prediction provided by DUBP, a further Adaptive Policy Pre-fetching and Caching scheme (APPC) is addressed for fine-grained and efficient network management.

The organization of this paper as follows, Section 2 introduces the overview of DUBP model, as well as its functions design. Section 3 details the proposed adaptive policy pre-fetching and caching scheme for fine-grained and efficient network management. Section 4 introduces the design of test bed, which is based on our universal network architecture. A necessary conclusion and future work will be shown in Section 5.

2. DUBP Model Description

Fig. 1 shows the architecture of the proposed DUBP model. The goal of DUBP is to analyze user access network services log and estimate user behavior popularity by means of distributed manner, per predicting user behavior to improve the speed of censoring policy responding and the performance of overall network censoring systems. In the DUBP model, all Log Collection and Analysis node (LCA) are chosen to construct the Chord to storage, disseminate and analyze users' access network service log. Packet Capturing and Policy Pre-fetch node (PCPP) is in charge of capturing packet from network equipments (such as router, switch etc.), extract desired log information accordance with specified criteria, then upload those log information to its connected LCA, and pre-fetch policy accordance with accessed page popularity provided by the LCA. Comparing to the traditional C/S-based log tracker and analysis model, the proposed DUBP model can support high scalability for large number of users with efficiently and robustly.

The DUBP model consists of two structures. The upper layer is a DHT network which is consisted by LCA. The lower layer is a *C/S* structure that connects LCA with PCPP. Detailed functions of the DUBP's components are described in Table 1.

Assume 1: Each web user had registered his/her basic information (such as age), and got his/her own access identifier (such as IP address, user ID etc.) in *User Information Database* (UID). Besides, current researches omit that people in different classification (such as age, the used core network etc.) have different interesting, so we play attention to which core network user used as well to implement a more fine-grained censoring.

Definition 1: A website S_i consists of a set of webpage which can be represented by (p_1, p_2, … , p_n). And $p_i \in (p_1, p_{2,}\dots, p_n)$ has own identification (URL) and popularity $\Re$.

The overall workflow of the DUBP model can be briefed as follows: The PCPP capturing HTTP-based packet from router or switch etc., extracts and use *Source IP* to get ***C**ore **N**etwork ID* (CNID) from its ***IP R**egular **D**atabase* (IPRD), then *PCPP* upload the record consisted of four tuples <CNID, DIP, DPort, URL> (DIP denotes Destination IP, DPort is on behalf of Destination Port, URL is the webpage path user requested) to its connect LCA; Once a record arriving at the LCA, the LCA share the record and inquire desired count URLs of the accessed website which has higher $\Re$ over DHT network, then a *replying* consisted of desired count URLs and source IP will be returned to the PCPP, the PCPP pre-fetches related policies from ***P**olicy **D**atabase* (PD) to support real-time and adaptive user behavior censoring. The LCA also stores and analyzes the $\Re$ periodically.

This section details how DUBP performs the functions of web user log analysis and behavior prediction for network censoring systems. For convenience, we define some useful annotations as described in Table II.

Table II. Annotation description used in DUBP model

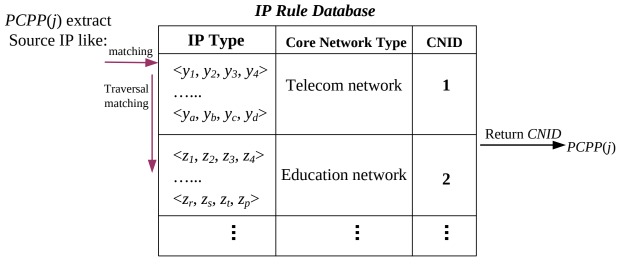
Annotation	Description
LCA(*i*)	The *Log Collection and Analysis* node *i*
PCPP(*i*)	The *Packet Capturing and Policy Pre-fetch* node *i*
LCAID(*i*)	The ID of the LCA *i*
SIP(*i*)	The Source IP *i*
DIP(*i*)	The accessing Destination IP *i*, namely the web user *i*
DPort(*i*)	The Port of Destination IP *i*
Hash()	The hash function of DHT network. For example, *Hash*(*CNID, DIP*(*i*), *DPort*(*i*)) = LCA(*i*) means that map tuple < *CNID*, *DIP*(*i*), *DPort*(*i*)> to the LCA whose ID is *LCAID*(*i*)

Below subsections address the major stages of DUBP model.

2.1 Web Access Records Uploading

As mentioned in Section II, when a web user initiates web access request, the PCPP will capture access logs then upload logs to its connected LCA, as well as receives $\Re$ information from the LCA. That information will be exchanged between LCAs over DHT network. How the designed process of uploading access logs work is illustrated as follows:

Step 1: Assume that a user *SIP*(*i*) request a website S_i, and PCPP(*p*) captured the *Get* packet from router/switcher, then the PCPP(*p*) extract *Source IP* and use it to get CNID from IPRD (CNID is a integration type such as 1,2,…, n which stands for the core network the user used such as the telecom network, the education network and so on respectively defined in the IPRD). Fig. 2 shows the example how to get CNID.

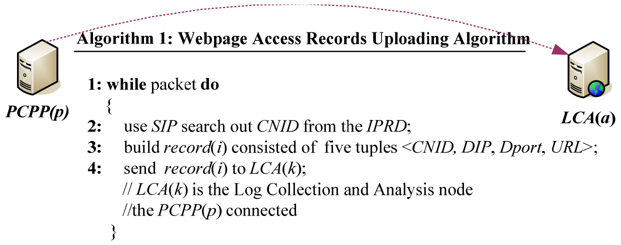


IP Rule Database

IP Type	**Core Network Type**	**CNID**
<y_1, y_2, y_3, y_4> …… <y_a, y_b, y_c, y_d>	Telecom network	**1**
<z_1, z_2, z_3, z_4> …… <z_r, z_s, z_t, z_p>	Education network	**2**
⋮	⋮	⋮

Figure 2. Diagram of an example for getting CNID

Step 2: The PCPP(*p*) use DPI [1] to extract requested page URL, then transmits those access record formed by <*CNID*, *DIP*, *DPort*, *URL*> to its connected LCA periodically. Fig. 3 shows how the PCCP uploads access records to its connected LCA.



Algorithm 1: Webpage Access Records Uploading Algorithm

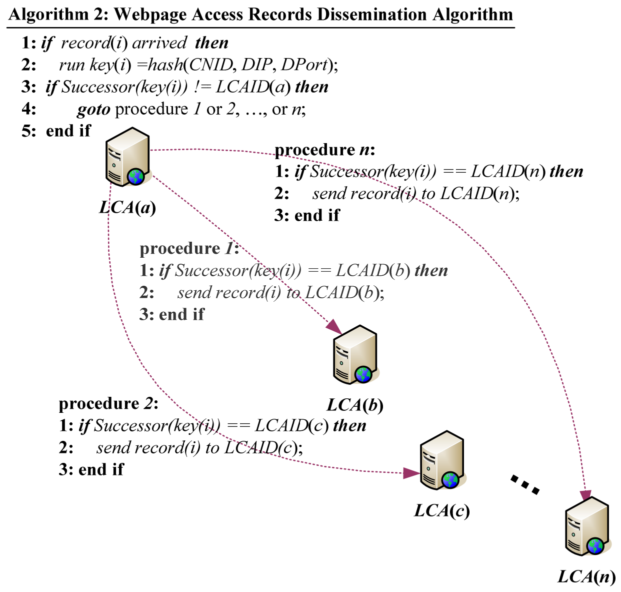
```
1: while packet do
    {
2:     use SIP search out CNID from the IPRD;
3:     build record(i) consisted of  five tuples <CNID, DIP, Dport, URL>;
4:     send  record(i) to LCA(k);
       // LCA(k) is the Log Collection and Analysis node
       //the PCPP(p) connected
     }
```

Figure 3. web user access records uploading algorithm

Step 3: When the *LCA* receives access records, it will run *hash* function as Eq.(1) to disseminate the record to others *LCA* over DHT network whose ID equals *Successor*(*key*). *Successor*(*key*) denotes the successor node of key value which has been detailed in [14].

$$key = hash(CNID + DIP + DPort) \qquad (1)$$

Fig. 4 illustrates the algorithm that how *LCA*(*a*) disseminates access record uploaded from *PCPP*.



Algorithm 2: Webpage Access Records Dissemination Algorithm

```
if record(i) arrived then
    run key(i) =hash(CNID, DIP, DPort);
    if Successor(key(i)) != LCAID(a) then
        goto procedure 1 or 2, …, or n;
end if
```

```
procedure n:
if Successor(key(i)) == LCAID(n) then
    send record(i) to LCAID(n);
end if
```

```
procedure 1:
if Successor(key(i)) == LCAID(b) then
    send record(i) to LCAID(b);
end if
```

```
procedure 2:
if Successor(key(i)) == LCAID(c) then
    send record(i) to LCAID(c);
end if
```

Figure 4. web user access records dissemination algorithm

Step 4: After *record*(*i*) arrived, the LCA(*k*) will cache a new record to its *Webpage Access Records Caching* Table shown in Table III.

Table III. Webpage Access Records Caching Table

CNID	DIP	DPort	URL

After the *PCPP*(*p*) uploading records to its connected *LCA*, then it will send *request*(*SIP*) to the *UID*; then the *UID* returns user information to the *PCPP*(*p*).

2.2 Webpage Access Popularity Analyzing

To reduce LCAs' stress, a period τ is set for webpage access popularity analysis. Webpage access popularity will be stored according with Table IV.

Table IV. Webpage Access Popularity Table

CNID	DIP	DPort	URL	$\Re$	*LastAccessTime*

When τ is coming, the LCAs will analyze webpage popularity as below algorithm detailed as Algorithm 3.

Algorithm 3: Webpage Access Popularity Analysis Algorithm

Definition:

R_i^3	the i_{th} record in Table 3, it consists of $< CNID_i^3, DIP_i^3, DPort_i^3, URL_i^3 >$
$CNID_i^3$	the *CoreNetwork ID* value in R_i^3
DIP_i^3	the *Destination IP* value in R_i^3
$DPort_i^3$	the *Destination Port* value in R_i^3
URL_i^3	the viewed URL record in R_i^3
R_j^4	the j_{th} record in Table 4, it consists of $< CNID_j^4, DIP_j^4, DPort_j^4, URL_j^4, LAT_j^4 >$
$CNID_j^4$	the *CoreNetwork ID* value in R_j^4
DIP_j^4	the *Destination IP* value in R_j^4
$DPort_j^4$	the *Destination Port* value in R_j^4
URL_j^4	the *URL* record in R_j^4
LAT_j^4	the *Last Access Time* in R_j^4

```
for i = 1 to m    //m is the total records count in Table 3
    for j = 1 to n    //n is the total records count in Table 4
        if ( $CNID_i^3 == CNID_j^4$ ) && ( $DIP_i^3 == DIP_j^4$ ) &&
           ( $DPort_i^3 == DPort_j^4$ ) && ( $URL_i^3$ equal $URL_j^4$ ) then
            $\Re$ ++;    //URL popularity $\Re$ increases 1
            set $LAT_j^4$ as LCA(k)'s current system time ;
            delete $R_i^3$ from Table 3; //cleanup cached record
        else
            create new record in Table 4;
            set $\Re$ = 1;    //set default $\Re$ for new viewed URL
            set LastAccessTime as LCA(k)'s current system time;
            delete $R_i^3$ from Table 3;  //cleanup cached record
        end if
    end for
end for
```

To reduce the load of *LCA*s and improve prediction accuracy, the redundant URLs which had never been viewed for a long time should be removed periodically. Thus, We set a time threshold (denoted as *thresh*) to remove related redundant records once *timerange* is larger than *thresh* . Where *timerange* is defined by

$$timerange = CurrentTime - LastAccessTime \qquad (2)$$

When a LCA leaves the DHT network, it should transfer its *Webpage Access Popularity* Table to one of its neighbor peers along the DHT network. When a LCA joins the DHT network, it should get some webpage access popularity information from its neighbor peers as original information. We notice that the LCA may leave the DHT network unexpectedly during popularity analyzing. If so, once the LCA leaves, a task flag will be set to false to ensure the LCA finish its analysis once it comes back to DHT network again.

3. Adaptive Policy Pre-fetching and Caching

Based on user behavior sensing and prediction provided by DUBP, a further Adaptive Policy Pre-fetching and Caching scheme (APPC) is addressed in this section for fine-grained and efficient network management. The major stages of APPC scheme are addressed as below.

Once a record (i.e. *record*(*i*)) arriving at LCA(*k*), it will trigger the LCA(*k*) executing some steps as follow immediately:

1) uses <CNID, DIP, DPort> to obtain the total matched URL counts (denoted as $Count_{Existing}$) from Table 4.
2) inquires the specified URL counts (denoted as $Count_{Specified}$) which is set by administrator in web console.
3) gets κ value via E.q (3). Then return a *replying* consisted of URLs with $\Re$ in top κ and source IP to the PCPP(p) where the record(i) comes from.
4) if there are more than κ URLs has same popularity $\Re$, the URL with less *timerange* will be selected to reach a more fine-grained and accurate predict.

$$\kappa = \begin{cases} Count_{Existing}; & Count_{Existing} < Count_{Specified} \\ Count_{Specified}; & Count_{Existing} \geq Count_{Specified} \end{cases} \quad (3)$$

After *replying* returned, the PCPP(*p*) prefetches corresponding policies with user information and the κ URLs from the PD.

To achieve policy pre-fetching more efficiently, we also employ reinforcement learning to construct a prediction model to predict web user navigating behavior.

a) ***Reinforcement Learning*** (RL). RL is learning what to do-how to map situations to actions, so as to maximize a numerical reward signal. As in Fig. 5, A RL Agent learns knowledge via interacting with the Environment. That is, once the Agent changes its state from one to another, a reward will be returned the environment. From the rewards, the Agent will learn a policy how to gain a more benefit rewards. Previous work [15] detailed the RL problem using Markov decision process (MDPs).

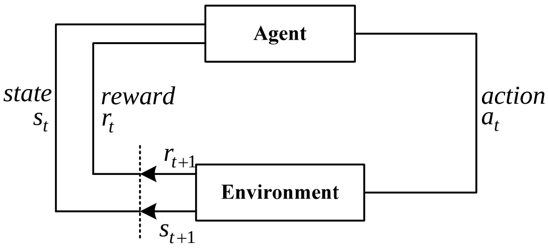


Figure 5. The reinforcement learning illustration

b) *Q*-learning is an off-policy temporal difference control algorithm [15] to learn *state-action* values. E.q (4) shows the on-step Q-learning:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \times [r_{t+1} + \gamma \times \max_{a'} Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (4)$$

Where s_t denotes current state of the *Agent*, a_t is the action adopted by the *Agent*. $Q(s_t, a_t)$ represents how good the a_t in the s_t. α denotes the learning rate. r_{t+1} is the reward returned from the *Environment*. γ is a discount parameter. As mentioned above, an optimal policy can be generated via the $Q(s, \alpha)$ function.

An improved and efficient *Q*-learning algorithm mentioned in [6] is employed in our DUBP model to decide which URLs should be pre-fetched and corresponding policies should be cached.

4. Universal Network based Testbed Design

We have already implemented the basic functions of BPPP mentioned above and confirmed their operations in our *Universal Network* (UN) [16-17]. Our UN is a novel next generation-oriented network architecture which is based on the well-known identifier/locator separation protocol [18]. To help the reader in understand the idea of Universal Network based testbed, we first introduce the context of identifier/locator separation.

4.1 Identifier/locator Separation

The networks with identifier/locator separation commonly consists of two parts, *transit core* and *edge networks*. Fig. 6 shows a basic wireless network topology with identifier/locator separation. We briefly detail how to forward packets in such network topology with assumptions that 1) the two terminal users are in different edge networks; and 2) routing in edge networks is separated from routing in the transit core.

Assuming that the User B with ID_B wants to open a connection to the User A with ID_A, following steps will be complied [19-20].

i. The User A first issue a data packet to its *Ingress Tunnel Router* (ITR). In this case, ID_A and ID_B act as the source and the destination of the packet, respectively.

ii. Every ITR maintains a table to cache some recently used identifier-to-locator (ID2LT) mappings. When the ITR receives the packet from the User A, it looks up a locator for ID_B in its local identifier-to-locator (ID2LT) mapping table which is used to cache some recently used ID2LT mappings, as shown by ② in Fig. 6. If the cache hits, go to step iv; otherwise, go to step iii.

iii. The ITR resolves the locator(s) for ID_B by querying a mapping server (shown by ③ in Fig. 6). When the ITR receives the resolved locators for ID_B, it caches them into its local ID2LT mapping table.

iv. Denote the locator of ITR and the resolved locator for ID_B by $Locator_1$ and $Locator_2$, respectively. The ITR encapsulates the received packet with an outer header whose destination and source are $Locator_1$ and $Locator_2$, respectively. Then the ITR then sends the encapsulated packet out with destination of *Egress Tunnel Router* (ETR), as shown by ④ in Fig. 6.

v. When the ETR receives the encapsulated packet, it 1) strips the outer header of the encapsulated packet; 2) stores the mapping from ID_A onto $Locator_1$ into its local ID2LT mapping table for possible future usage; and 3) sends the decapsulated packet to corresponding destination, namely User B in the Fig. 6.

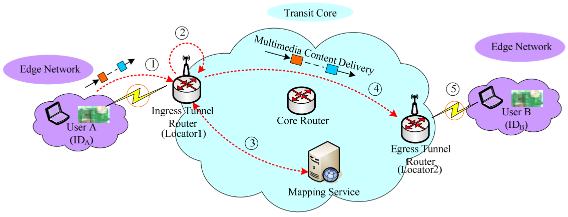


Figure 6. A basic wireless network with identifier/locator separation

4.2 The Framework of UN-based Testbed

The framework of testbed is shown as Fig.7. It consists of two layers, *management layer* and *switch routing layer*, which are described in details next.

a) ***Management Layer***. In order to meet the information security requirements for multimedia communication, computation and service in the identifier/locator separation context, it is necessary to consider some useful management components that refer to multimedia security. For this purpose, the framework of testbed provides management layer to launch security management. The management layer includes *Identifier Mapping Server* (IDMS), *Access-control Policy Database* (ACPD), *User Registrant/Authentication Server* (URAS), *Service Registrant/Authentication Server* (SRAS) and DUBP. Table V describes the components in Management Layer.

b) ***Switch Routing Layer***. As shown in figure 3, Switch Routing Layer consists of Ingress Tunnel Router (ITR), Egress Tunnel Router (ETR), and Core Router (CR) and so on. Next the major functions of these components are detailed. ITR in the testbed still keeps capabilities same as that in original identifier/locator separation context, such as 1) acts as an access point for terminals in Edge Network access to Transit Core; 2) caches and provides ID2LT mappings; and 3) forwards packet. Moreover, for sake of security management, it 4) enables APPC to pre-fetch and cache the policy instances sent from ACPD; 5) requests UTag and STag from URAS and SRAS, respectively; 6) decides whether to grant access in conjunction with the policy instances, UTag and STag. ETR in the testbed still keeps capabilities same as that in original identifier/locator separation context. Previous work [19-20] detailed the functions of ETR. The CR still keeps capabilities same as that in original identifier/locator separation context. That is, it just routes and forwards packet in *transit core*.

5. Conclusions and Future Work

Users' behavior log attracts more and more attentions in current researches. However, Log server employed in current user behavior analysis, network censoring systems usually base on the traditional Client-Server (C/S) architecture. With more and more user behavior records arising, the single C/S tracker server will inevitably become a bottleneck during communicating, caching and analyzing required to process large-scale network service request. In this paper, a novel Distributed web User Behavior Prediction model (DUBP) for network censoring systems is proposed, which extends a distributed DHT network structure. The DUBP makes all nodes' available resources and predicts user behaviors fine-grained by means of users' core network type and historical access records, it can support high scalability for large number of users with efficiently and robustly. Based on user behavior sensing and prediction provided by DUBP, a further Adaptive Policy Pre-fetching and Caching scheme (APPC) is addressed for fine-grained and efficient network management.

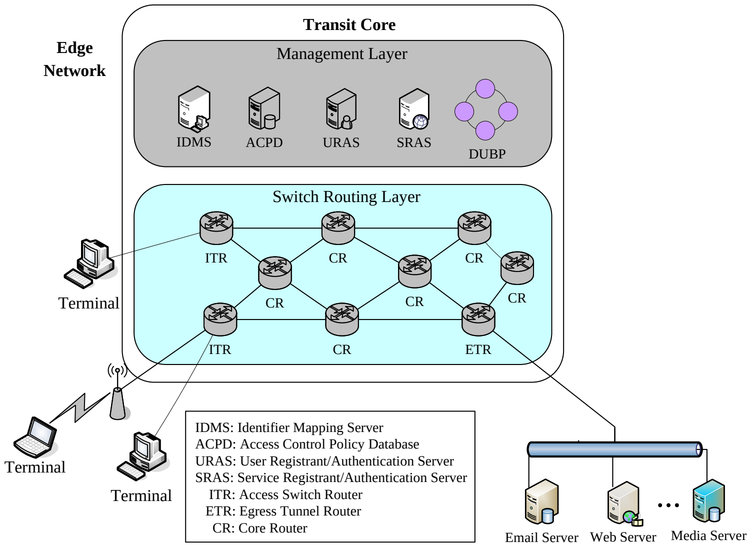

Transit Core
Edge Network
Management Layer
IDMS
ACPD
URAS
SRAS
DUBP
Switch Routing Layer
ITR
CR
CR
CR
CR
CR
CR
ITR
CR
ETR
Terminal
Terminal
Terminal
IDMS: Identifier Mapping Server
ACPD: Access Control Policy Database
URAS: User Registrant/Authentication Server
SRAS: Service Registrant/Authentication Server
ITR: Access Switch Router
ETR: Egress Tunnel Router
CR: Core Router
Email Server
Web Server
Media Server

Figure 7. The framework of Universal Network based testbed

Table V. Components in Management Layer

Components	Descriptions
IDMS	IDMS stores identifier-to-locator (ID2LT) mappings for each terminal and subnet which are under its control.
ACPD	ACPD stores the policy instances pre-fetched by APPC and sends them to ITR and ETR periodically. Moreover, a new or updated policy instance will be sent to ITR and ETR instantly by ACPD.
URAS	URAS includes three function modules, which User Registration Module (URM) provides a user interface (i.e. web) for user registration, while User Tag Generator (UTG) creates a User Tag (UTag) for each registered user accordance with 1) user basic information (i.e. age, interest, education and so on); and 2) dynamic information (i.e. malicious behavior) provided by DUBP. And User Authentication Module (UAM) creates a User ID (UID) for each registered user to access Internet. A user without UID will be failed to access Internet by UAM.
SRAS	like URAS, SRAS also includes three function modules, which Service Registration Module (SRM) provides a registration interface (i.e. web) for Internet Service Provider (ISP) or Internet Content Provider (ICP) to register services, while Service Tag Generator (STG) creates a static Service Tag (STag) for each registered service accordance with 1) service basic information (i.e. fee, language, constraint-level and so on) provide by ISP/ICP; and 2) dynamic information (i.e. constraint content). And Service Authentication Module (SAM) creates a Service ID (SID) for each registered media. A service without SID cannot be deployed into Internet.
DUBP	DUBP aims to store and analyze user access behavior (i.e. malicious behavior), and send out analysis results to URAS.

For the experimental validation, we are now extending the UN for optimizing the communication capability with our previous work [21-27]. We will then evaluate and compare the performance of UN with or without the proposed scheme mentioned in this paper during it implements network management.

Acknowledgments

This work was partially supported by the National High-Tech Research and Development Program of China (863) under Grant No. 2011AA010701, in part by the National Basic Research Program of China (973 Program) under Grant 2013CB329102, in part by the National Natural Science Foundation of China (NSFC) under Grant No. 61001122, 61003283, 61232017, in part by the Fundamental Research Funds for the Central Universities under Grant No. 2012RC0603, and in part by the Jiangsu Natural Science Foundation of China under Grant No. BK2011171.

References

[1] M. Grossglauser and J. Rexford, “Passive traffic measurement for IP Operations,” in The Internet as a Large-Scale Complex System, 2005, pp. 91–120.

[2] Shenk, Jerry, “SANS Annual 2009 Log Management Survey,” Technical Report, SANS, 2009.

[3] Baker, Wade, A. Hutton, C. David Hylender, C. Novak, C. Porter, B. Sartin, P. Tippett, and J. Andrew Valentine, “Data Breach Investigations Report,” Technical Report, Verizon Business RISK Team, 2009.

[4] J. Myers, M. Grimaila, R. Mills, “Log-Based Distributed Security Event Detection Using Simple Event Correlator,” In Proceedings of the 44th Hawaii International Conference on System Science, Jan. 2011.

[5] E. Hilgenstieler, E. Duarte, G. Mansfield-Keeni, N. Shiratori, “Improving the Precision and Efficiency of Log-based IP Packet Traceback,” In Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'07), Nov. 2007.

[6] T. Xu, W. Wang, B. Ye, W. Li, S. Lu, Y. Gao, “Prediction-based prefetching to support VCR-like operations in gossip-based P2P VoD systems,” In Proceedings of the 15th International Conference on Parallel and Distributed Systems (ICPADS), Shenzhen, China, Dec. 2009.

[7] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, “The BitTorrent P2P File-Sharing System: Measurements and Analysis,” in Proceedings of Fourth Int’l Workshop Peer-to-Peer Systems (IPTPS), 2005.

[8] I. Foster and A. Iamnichi, “On Death, Taxes, and Convergence of P2P and Grid Computing,” In Proceedings of the Second Int’1 Workshop Peer-to-Peer Systems (IPTP3'03), Feb. 2003.

[9] I. Kelenyi, J.K. Nurminen, M. Matuszewski, “DHT Performance for Peer-to-Peer SIP-A Mobile Phone Perspective,” In Proceedings of 7th IEEE Consumer Communications and Networking Conference (CCNC'10), 2010.

[10] C. Xu, E. Fallon, Q. Yuansong, Z. Lujie and M. Gabriel-Miro, “Performance Evaluation of Multimedia Content Distribution Over Multi-Homed Wireless Networks,” IEEE Transactions on Broadcasting, vol. 57, no. 2, June 2011.

[11] G. Urdaneta, G. Pierre, M. Steen, “A Survey of DHT Security Techniques,” ACM Computing Surveys, 2009.

[12] R. Zhou and K. Hwang, “GossipTrust for Fast Reputation Aggregation in Peer-to-Peer Networks,” IEEE Transactions on Knowledge and Data Engineering, vol.20, no. 9, pp. 1282-1295, Sept. 2008.

[13] Y. Liu, W. Xue, K. Li, *et al*. “DHTrust: A Robust and Distributed Reputation System for Trusted Peer-to-Peer Networks,” In Proceedings of 2010 IEEE Global Telecommunications Conference (GLOBECOM’10), Dec. 2010.

[14] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for Internet applications,” In Proceedings of ACM SIGCOMM’01, Aug. 2001.

[15] R. S. sutton and A. G. Barto, “Reinforcement Learning: An Introduction,” MIT Press, Cambridge, MA, USA, 1998.

[16] Hongke Zhang, “An Architecture of Universal Network Services,” Patent Application, no. 200510134579.1, 2005.

[17] Hongke Zhang, “A method of implementing pervasive service in Universal Network,” Patent Application, no.200610169727.8, 2006.

[18] D. Farinacci, V. Fuller, D. Meyer, D. Lewis, “Locator/ID Separation Protocol (LISP),” IETF Internet Draft, draft-ietf-lisp-23.txt (work in progress), May 2012.

[19] H. Luo, H. Zhang, and C. Qiao, “Efficient Mobility Support by Indirect Mapping in Networks with Locator/Identifier Separation,” IEEE Transactions on Vehicular Technology, vol.60, no.5, pp.2265-2279, June 2011.

[20] H. Luo, H. Zhang, and M. Zukerman, “Decoupling the design of identifier-to-locator mapping services from identifiers,” *Computer Networks*, vol. 55, no. 4, pp. 959–974, March 2011.

[21] C. Xu, T. Liu, J. Guan and H. Zhang, G.-M. Muntean, “CMT-QA: Quality-aware Adaptive Concurrent Multipath Data Transfer in Heterogeneous Wireless Networks,” IEEE Transactions on Mobile Computing, vol.PP, no.99, Aug. 2012.

[22] Y. Cao, C. Xu, J. Guan, F. Song, H. Zhang, “Environment-aware CMT for Efficient Video Delivery in Wireless Multimedia Sensor Networks,” International Journal of Distributed Sensor Networks, vol.2012, Article ID 381726, 12 pages, 2012.

[23] Y. Cao, C. Xu, J. Guan, H. Zhang, “Background Traffic-based Retransmission Algorithm for Multimedia Streaming Transfer over Concurrent Multipaths,” International Journal of Digital Multimedia Broadcasting, vol.2012, Article ID 789579, 10 pages, 2012.

[24] Y. Cao, C. Xu, J. Guan, J. Zhao, H. Zhang, “Cross-layer Cognitive CMT for Efficient Multimedia Distribution over Multi-homed Wireless Networks,” In Proceedings of IEEE Wireless Communications and Networking Conference (WCNC’13), accepted.

[25] Y. Cao, C. Xu, J. Guan, H. Zhang, “Cross-layer Retransmission Approach for Efficient VoD Transfer over Multi-homed Wireless Networks,” International Journal of Digital Content Technology and its Applications, vol.6, no.23, pp.98-109, Dec. 2012.

[26] Y. Cao, C. Xu, J. Guan, et al., “Relational Analysis Based Concurrent Multipath Transfer Over Heterogeneous Vehicular Networks,” International Journal of Computer Science Issues, vol.9, issue 5, no.2, pp.1-10, Sep. 2012.

[27] Y. Cao, C. Xu, J. Guan, “A record-based retransmission policy on SCTP’s Concurrent Multipath Transfer,” In Proceedings of 2011 International Conference on Advanced Intelligence and Awareness Internet, pp.67-71, Oct. 2011.

Yuanlong Cao received his B.S. degree from Nanchang University of China in 2006, received his M.S degree from Beijing University of Posts and Telecommunications (BUPT) in 2008. During 2007-2009, he worked as an intern in BEA China Telecommunications Technology Center (BEA TTC) and IBM China Development Lab (IBM CDL). During 2009-2010, he worked as a software engineer in DT Research (Beijing). He is currently working toward the Ph.D. degree in the Institute of Network Technology, BUPT. He is broadly interested in computer networks, multimedia communications, wireless networking, network security, and next generation Internet technology.

Jianfeng Guan received his B.S. degree from Northeastern University of China in July 2004, and received the Ph.D. degrees in communications and information system from the Beijing Jiaotong University, Beijing, China, in Jan. 2010. He is a Lecturer in the Institute of Network Technology at Beijing University of Posts and Telecommunications (BUPT), Beijing, China. His main research interests focus around mobile IP, mobile multicast and next generation Internet.

Wei Quan received his B.S. degree in information and computer science from China University of Petroleum (Beijing) in 2009. He is currently working toward the Ph.D. degree in the Institute of Network Technology, Beijing University of Posts and Telecommunications (BUPT). He is broadly interested in computer network technology. In particular, his research interests include wireless sensor network, cognitive wireless network, mobile IP, and next generation Internet technology.

Jia Zhao received the M.S. degree in electrical engineering from Beijing Jiaotong University, China, in 2011. He is pursuing the Ph.D. degree at national engineering laboratory for next generation Internet interconnection devices, Beijing Jiaotong University. His research interests include traffic engineering, overlay routing, game theory, social mobility and mobile ad hoc networks.

Changqiao Xu is an Associate Professor in the Institute of Network Technology and Associate Director of the Next Generation Internet Technology Research Center at Beijing University of Posts and Telecommunications (BUPT), China. He received his PhD degree in Computer Applied Technology from Institute of Software, Chinese Academy of Sciences (ISCAS) in Jan 2009. He was an Assistant Research Fellow in ISCAS from 2002 to 2007, where he held role as a project manager in the research & development area of communication networks. During 2007-2009, he worked as a researcher in Software Research Institute at Athlone Institute of Technology, Ireland. He joined BUPT in Dec 2009 and was a Lecturer from 2009 to 2011. His research interests include computer networks, multimedia communications, wireless networking, network security, and next generation Internet technology.

Hongke Zhang received his M.S. and Ph.D. degrees in Electrical and Communication Systems from the University of Electronic Science and Technology of China in 1988 and 1992, respectively. From Sep. 1992 to June 1994, he was a post-doc research associate at Beijing Jiaotong University. In July 1994, he jointed Beijing Jiaotong University, where he is a professor. He has published more than 100 research papers in the areas of communications, computer networks and information theory. He is the director of the National Engineering Laboratory for Next Generation Internet Interconnection Devices.