International Journal on Applications in Information and Communication Engineering
Volume 1: Issue 9: September 2015, pp 51-55. www.aetsjournal.com ISSN (Online) : 2394-6237

Traffic Pattern Based Content Leakage Detection for Trusted Content Delivery Networks

K.B.Sakthipriya , K.Venkatesh

***Abstract*—**Due to the increasing popularity of multimedia streaming applications and services in recent years, the issue of trusted video delivery to prevent undesirable content-leakage has, indeed, become critical. While preserving user privacy, conventional systems have addressed this issue by proposing methods based on the observation of streamed traffic throughout the network. These conventional systems maintain high detection accuracy while copying with some of the traffic variation in the network (e.g., network delay and packet loss), however, their detection performance substantially degrades owing to the significant variation of video lengths. In this paper, we focus on overcoming this issue by proposing a novel content-leakage detection scheme that is robust to the variation of the video length. By comparing videos of different lengths, we determine a relation between the length of videos to be compared and the similarity between the compared videos. Therefore, we enhance the detection performance of the proposed scheme even in an environment subjected to variation in length of video. Through a test bed experiment, the effectiveness of our proposed scheme is evaluated in terms of variation of video length, delay variation, and packet loss.

***Index Terms*—** Multimedia streaming, Content leakage detection scheme, Conventional system, User privacy.

I. INTRODUCTION

In recent years, with the rapid development of broadband technologies and the advancement of high-speed wired/ wireless networks, the popularity of real-time video streaming applications and services over the Internet has increased by leaps and bounds. YouTube and Microsoft network video are notable examples of such applications. They serve a huge population of users from all around the world with diverse contents, ranging from daily news feeds to entertainment feeds including music, videos, sports, and so forth, by using streaming transmission technologies. In addition, real-time video streaming communications such as web conference in Intracompany networks or via Internet with virtual private networks (VPNs) are being widely deployed in a large number of corporations as a powerful means of efficiently promoting business activities without additional costs .

A crucial concern in video streaming services is the protection of the bit stream from unauthorized use, duplication and distribution. One of the most popular approaches to prevent undesirable contents distribution to unauthorized users and/or to protect authors' copyrights is the digital rights management (DRM) technology. Most DRM techniques employ cryptographic or digital watermark techniques. However, this kind of approaches have no significant effect on redistribution of contents, decrypted or restored at the user-side by authorized yet malicious users. Moreover, redistribution is technically no longer difficult by using peer-to-peer (P2P) streaming software. Hence, streaming traffic may be leaked to P2P networks. On the other hand, packet filtering by firewall-equipped egress nodes is an easy solution to avoid leakage of streaming contents to external networks. In this solution, the packet header information (e.g., destination and source Internet protocol addresses, protocol type, and port number of outgoing traffic) of every streamed packet is inspected.

In case the inspected packets do not verify the predefined filtering policy, they are blocked and dropped. However, it is difficult to entirely prevent streaming content leakage by means of packet filtering alone because the packet header information of malicious users is unspecified beforehand and can be easily spoofed. In this paper, we focus on the illegal redistribution of streaming content by an authorized user to external networks. The existing proposals in, and monitor information obtained at different nodes in the middle of the streaming path. The retrieved information is used to generate traffic patterns which appear as unique waveform per content, just like a fingerprint. The generation of traffic pattern does not require any information on the packet header, and therefore preserves the user's privacy.

Leakage detection is then performed by comparing the generated traffic patterns. However, the existence of videos of different length in the network environment causes a considerable degradation in the leakage detection performance. Thus, developing an innovative leakage detection method robust to the variation of video lengths is, indeed required. In this paper, by comparing different length videos, we determine a relationship between the length of videos to be compared and their similarity. Based on this relationship, we determine decision threshold enabling accurate leakage detection even in an environment with different length videos. The remainder of the paper is organized as follows: A typical video leakage scenario, detection system and procedures are described in Section 2. In Section 3, first we depict the drawback of the existing scheme due to the variation of video length in realistic environment,

K.B.Sakthipriya , PG Scholar, Department of Computer Science and Engineering, Mount Zion College of Engineering and Technology, Pudukottai

K.Venkatesh , Assistant Professor, Department of Computer Science and Engineering, Mount Zion College of Engineering and Technology, Pudukottai.

then we described the proposed leakage detection scheme, and we evaluate its calculation cost in comparison to that of the existing scheme. We evaluate the effectiveness and the accuracy of the proposed scheme with respect to different length videos, and its robustness to network environment changes.

The basic solution for streaming video over the Internet is the client-server service model. A client sets up a connection with a video source server and video content is streamed to the client directly from the server. One variation of client-server service model is the Content Delivery Network (CDN) based video streaming. In CDN based solution, the video source server first push video content to a set of content delivery servers placed strategically at the network edges. Instead of downloading from the video source server, a client is normally directed to a nearby content delivery server to download the video. CDN effectively shortens the users' startup delays, reduces the traffic imposed on the network, and serves more users as a whole. YouTube employs CDN to stream video to end users. The major challenge for server based video streaming solutions, though, is its scalability. A video session with good quality requires high bandwidth. With the current video compression technology, the streaming rate for a TV quality video is more than 400 kilobits-per-second. The bandwidth provision, at video source servers or in CDNs, must grow proportionally with the client population. This makes the server based video streaming solutions expensive.

Peer-to-Peer (P2P) networking has recently emerged as a new paradigm to build distributed network applications. The basic design philosophy of P2P is to encourage users to act as both clients and servers, namely as peers. In a P2P network, a peer not only downloads data from the network, but also uploads the downloaded data to other users in the network. The uploading bandwidth of end users is efficiently utilized to reduce the bandwidth burdens otherwise placed on the servers. P2P file sharing applications, such as have been widely employed to quickly disseminate data files on the Internet. More recently, P2P technology has been employed to provide media streaming services. Several P2P streaming systems have been deployed to provide on demand or live video streaming services over the Internet. Our recent measurement study of a P2P live video streaming system shows that, in early 2006, more than 200, 000 simultaneous users watched the live broadcast of an 4-hour event at bit rates from 400 to 800kpbs. The aggregate required bandwidth reaches 100 gigabits/sec, while Akamai reportedly has roughly 300 gigabits/sec bandwidth in its entire network at the end of year 2006.P2P streaming systems can be broadly classified into two categories based on the overlay network structure. They are tree-based and mesh-based. The tree-based systems, such as ESM, have well-organized overlay structures and typically distribute video by actively pushing data from a peer to its children peers. One major drawback of tree-based streaming systems is their vulnerability to peer churn. A peer departure will temporarily disrupt video delivery to all peers in the subtree rooted at the departed peer. In a mesh-based P2P streaming system, peers are not confined to a static topology. Instead, the peering relationships are established or terminated based on the content availability and bandwidth availability on peers. A peer dynamically connects to a subset of random peers in the system. Peers periodically exchange information about their data availability. Video content is pulled by a peer from its neighbors who have already obtained the content. Since multiple neighbors are maintained at any given moment, mesh-based video streaming systems are highly robust to peer churns. However, the dynamic peering relationships make the video distribution efficiency unpredictable. Different data packets may traverse different routes to users. Consequently, users may suffer from video playback quality degradation ranging from low video bit rates, long startup delays, to frequent playback freezes.

In the rest of the article we give a survey on the existing P2P media streaming systems. The P2P live streaming systems are described first in Section 2, followed by the P2P video-on-demand systems in Section 3. You will see how different design requirements influence the system architectures. Within each section, representative systems are used as examples to show both tree-based and mesh-based system architectures.

II. LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

Several scene-detection algorithms, which are only based on bit rate fluctuations, have been proposed. All of them are presented on the fixed thresholds, which are obtained by the empirical records of the video characteristics. Due to the sensitivity of these methods to the accuracy of the records, which are generally obtained by testing several values repeatedly, bad performance evaluation might be observed for the actual scene detection, especially for real-time video

traffic. In this paper, we review the previous works in this area, and study the correlation between the scene duration and the scene change at the frame level, and simultaneously investigate the local statistical characteristics of scenes such as variance and peak bit rate etc. Based on this analysis, an effective decision function is first constructed for the scene segmentation. Then, we propose [1] a scene-detection algorithm using the defined dynamic threshold model, which can capture the statistical properties of the scenechanges. Experimental results using 15 variable bit rate MPEG video traces indicate good performances of the proposed algorithm with significantly improved scene-detection accuracy.

In this paper [2], we have proposed to use the method of principal curves to describe and analyze the interaction among freeway traffic-stream variables and their joint behaviors without utilizing conventional assumptions made on the functional forms of interactions, as in previous studies. As a non-parameter modeling approach, the performance of the proposed method depends only on the data used and involves no assumed knowledge regarding the relationship among the traffic-stream variables. First, we discuss the basic algorithm for data analysis using principal curves and the corresponding data filter algorithm for determining principal curves for application in traffic-steam analysis. Second, a case study is used to compare the performance of the proposed method to that of the classical model proposed by Green shields; results indicate that the proposed model is better than the classical one in both data accuracy and curve shape. Finally, the traffic-stream models generated with principal curves at different locations and lanes are compared with each other's and the three-dimensional traffic-stream models developed from principal curves are discussed. Clearly, our results have demonstrated the feasibility and advantages of applying principal curves in freeway traffic-stream modeling and analysis.

Despite growing maturity in broadband mobile networks, wireless video streaming remains a challenging task, especially in highly dynamic environments. Rapidly changing wireless link qualities, highly variable roundtrip delays, and unpredictable traffic contention patterns often hamper the performance of conventional end-to-end rate adaptation techniques such as TCP-friendly rate control (TFRC). Furthermore, existing approaches tend to treat all flows leaving the network edge equally, without accounting for heterogeneity in the underlying wireless link qualities or the different rate utilities of the video streams. In this paper [3], we present a proxy-based solution for adapting the scalable video streams at the edge of a wireless network, which can respond quickly to highly dynamic wireless links. Our design adopts the recently standardized scalable video coding (SVC) technique for lightweight rate adaptation at the edge. Leveraging previously developed rate and quality models of scalable video with both temporal and amplitude scalability, we derive the rate-quality model that relates the maximum quality under a given rate by choosing the optimal frame rate and quantization step size. The proxy iteratively allocates rates of different video streams to maximize a weighted sum of video qualities associated with different streams, based on the periodically observed link throughputs and the sending buffer status. The temporal and amplitude layers included in each video are determined to optimize the quality while satisfying the rate assignment. Simulation studies show that our scheme consistently outperforms TFRC in terms of agility to track link qualities and overall subjective quality of all streams. In addition, the proposed scheme supports differential services for different streams, and competes fairly with TCP flows.

One of the significant workloads in current generation desktop processors and mobile devices is multimedia processing. Large on-chip caches are common in modern processors, but large caches will result in increased power consumption and increased access delays. Regular data access patterns in streaming multimedia applications and video processing applications can provide high hit-rates, but due to issues associated with access time, power and energy, caches cannot be made very large. Characterizing and optimizing the memory system is conducive for designing power and performance efficient multimedia application processors. Performance tradeoffs for multimedia applications have been studied in the past; however, power and energy tradeoffs for caches for multimedia processing have not been adequately studied in the past. In this paper, we characterize multimedia applications for I-cache and D-cache power and energy using a multilevel cache hierarchy. Both dynamic and static power increase with increasing cache sizes, however, the increase in dynamic power is small. The increase in static power is significant, and becomes increasingly relevant for smaller feature sizes. There is significant static power dissipation, in L1 & L2 caches at 70 nm technology sizes, emphasizing the fact that future multimedia systems must be designed by taking leakage power reduction techniques into account. The energy consumption of on-chip L2 caches is seen to be very sensitive to cache size variations. Sizes larger than 16 k for I-caches and32 k for D-caches will not be efficient choices to maintain power and performance balance. Since multimedia applications spend significant amounts of time in integer operations, to improve the performance, we propose implementing low power full adders and hybrid multipliers in the data path, which results in 9% to 21% savings in the overall power consumption. In order to use the proposed algorithm on data streams, we introduce a factor which applies exponential decay to the prominence of history data. Our result [4] exhibits the effectiveness of the proposed algorithm and discloses instinctive and exciting information for clustering text streams.

In this paper [5], a novel exemplar-based constructive approach using kernels is proposed for simultaneous pattern classification and multi domain pattern association tasks. The kernel networks are constructed on a modular basis by a simple one-shot self-structuring algorithm motivated from the traditional Hebbian principle and then, they act as the flexible memory capable of generalization for the respective classes. In the self-structuring kernel memory (SSKM), any arduous and iterative network parameter tuning is not involved for establishing the weight connections during the construction,

unlike conventional approaches, and thereby, it is considered that the networks do not inherently suffer from the associated numerical instability. Then, the approach is extended for multi domain pattern association, in which a particular domain input cannot only activate some kernel units (KUs) but also the kernels in other domain(s) via the cross-domain connection(s) in between. Thereby, the SSKM can be regarded as a simultaneous pattern classifier and associator. In the simulation study for pattern classification, it is justified that an SSKM consisting of distinct kernel networks can yield relatively compact-sized pattern classifiers, while preserving a reasonably high generalization capability, in comparison with the approach using support vector machines (SVMs).

III. Proposed Scheme

The proposed system provided an efficient interactive streaming service for diversified mobile devices and dynamic network environments. When a mobile device requests a multimedia streaming service, it transmits its hardware and network environment parameters to the profile agent in the cloud environment, which records the mobile device codes and determines the required parameters. The most suitable SVC code for the device according to the parameters, and then the SVC Transcoding Controller (STC) hands over the transcoding work via map-reduce to the cloud, in order to increase the transcoding rate. The multimedia video file is transmitted to the mobile device through the service.

Advantage

- The network bandwidth can be changed dynamically.
- This method could provide efficient self-adaptive multimedia streaming services.

A. Scalable Video Coding

Scalable Video Coding (SVC) is the name for the Annex G extension of the H.264/MPEG-4 AVC video compression standard. SVC standardizes the encoding of a high-quality video bit-stream that also contains one or more subset bit-streams. A subset video bit-stream is derived by dropping packets from the larger video to reduce the bandwidth required for the subset bit-stream. The subset bit-stream can represent a lower spatial resolution (smaller screen), lower temporal resolution (lower frame rate), or lower quality video signal. Temporal (frame rate) scalability: the motion compensation dependencies are structured so that completes pictures. Spatial (picture size) scalability: video is coded at multiple spatial resolutions. The data and decoded samples of lower resolutions can be used to predict data or samples of higher resolutions in order to reduce the bit rate to code the higher resolutions. SNR/Quality/Fidelity scalability: video is coded at a single spatial resolution but at different qualities. The data and decoded samples of lower qualities can be used to predict data or samples of higher qualities in order to reduce the bit rate to code the higher qualities.

B. Transcoder

Transcoding is the process of converting a file from one encoding format to another. This allows the conversion of incompatible data to a better-supported, more modern form of data. Transcoding is often performed if the target device does not support the format or has only limited storage capability. Transcoding is widely used in mobile phone content adaptation as well as in multimedia message servicing. Transcoding technology is also implemented in home theater PC software, enabling disk space reduction. The material is reformatted there and sent on to a proxy server. The proxy server accesses information about device preferences and adapts the material as needed before delivering it to the end user.

IV. System Architecture

The major part of the project development sector considers and fully survey all the required needs for developing the project. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations. Generally algorithms shows a result for exploring a single thing that is either be a performance, or speed, or accuracy, and so on. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them.

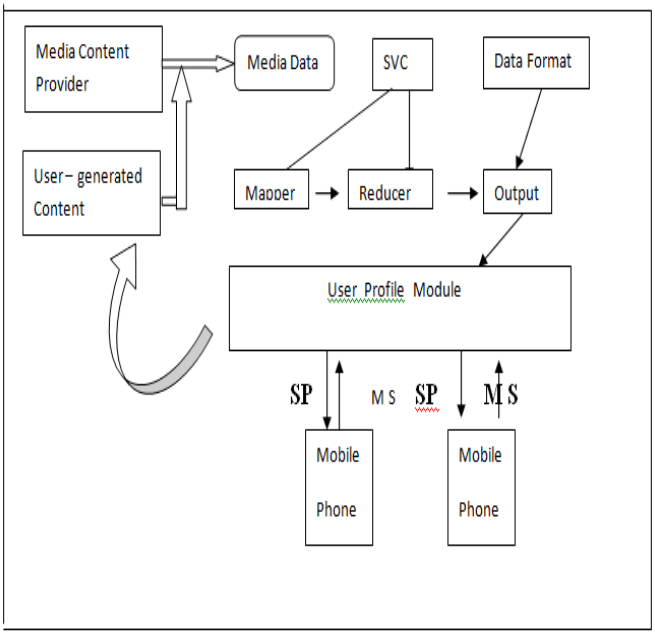

Fig 1: Architecture Diagram

I. Implementation

Following are the most frequently used project management Methodologies in the project management practice:

1. User Profile Module

2. Network device module
3. Cloud request or store video
4. Traffic pattern

A. User Profile Module

The profile agent is used to receive the mobile hardware environment parameters and create a user profile. The mobile device transmits its hardware specifications in XML-schema format to the profile agent in the cloud server. The XML-schema is metadata, which is mainly semantic and assists in describing the data format of the file. The metadata enables non owner users to see information about the files, and its structure is extensible. However, any mobile device that is using this cloud service for the first time will be unable to provide such a profile, so there shall be an additional profile examination to provide the test performance of the mobile device and sample relevant information. Through this function, the mobile device can generate an XML-schema profile and transmit it to the profile agent. The profile agent determines the required parameters for the XML-schema and creates a user profile, and then transmits the profile to the DAMM for identification.

B. Network Device Module

Network device module or Bayesian prediction module. In this module we calculate band width, device model and network provider, network type and sim state. However we find all details are stored and retrieved data. We calculate the bit rate and band width. How to calculate bandwidth we use three type of band width, Tested exiting, Avg available and standard deviation. When this parameter form is maintained, the parameters can be transmitted to the network estimation module and the device-aware Bayesian prediction module for relevant prediction.

C. Cloud Request or Store Video

In this modules we stored different types videos to directly or send mobile because the users request any videos and different kind of mobiles, some persons request any videos so well store all videos in cloud, it provides a simple services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the server. It gives any developer access to the same highly scalable, reliable, secure, fast, inexpensive infrastructure that cloud uses to run its own global network. The service aims to maximize benefits of scale and to pass those benefits on to users. Once cloud user send video to cloud , that video stored below types bit rate , band width , width, height, standard deviation, decoding and encoding. When the prediction error is greater than error boundary, the system shall reduce the weight modification of the predicted difference; relatively, when the prediction error is less than error boundary, the system shall strengthen the weight modification of the predicted difference. When the changed bandwidth of the system is greater than the standard difference, the predicted weight will increase as the corrected value of the standard deviation is reduced.

D. Traffic Pattern

Therefore parts of a scalable bit-stream can be removed in a way that the resulting sub-stream forms another valid bit-stream for a given decoder, which represents the source content with a reduced picture size, frame-rate, and/or reconstruction quality compared to the original bit-stream. From a video adaptation point of view, it shows an advantage of scalable video coding compared to non-scalable video coding. Although coding efficiency of non-scalable video coding is a little higher than SVC, SVC has more practical adaptation solutions in heterogeneous environment where diverse terminals are connected to varying bandwidth. In case that network conditions are better while a video steam with low quality is being transmitted, a stream coded by SVC can support more quality and/or spatial resolution, however one produced by non-scalable coding should be delivered in constant bit-rate and picture size.

V. CONCLUSION

The content leakage detection system based on the fact that each streaming content has a unique traffic pattern is an innovative solution to prevent illegal redistribution of contents by a regular, yet malicious user. Though three typical conventional methods, namely: T-TRAT, P-TRAT and DP-TRAT, show robustness to delay, jitter or packet loss, the detection performance decreases with considerable variation of video lengths. This paper attempts to solve these issues by introducing a dynamic leakage detection scheme. Moreover, in this paper, we investigate the performance of the proposed method under a real network environment with videos of different lengths. The proposed method allows flexible and accurate streaming content leakage detection independent of the length of the streaming content, which enhances secured and trusted content delivery.

In future work for we investigate the performance of the proposed method under a real network environment with videos of different lengths. The proposed method allows flexible and accurate streaming content leakage detection independent of the length of the streaming content, which enhances secured and trusted content delivery.

REFERENCES

[1] M. Dobashis, H. Nakayama, N. Kato, Y. Nemoto, and A. Jamalipour, “Traitor Tracing Technology of Streaming Contents Delivery Using Traffic Pattern in Wired/Wireless Environments,” Proc. IEEE Global Telecomm. Conf., pp. 1-5, Nov. /Dec. 2006.

[2] K. Matsuda, H. Nakayama, and N. Kato, “A Study on Streaming Video Detection Using Dynamic Traffic Pattern,” IEICE Trans. Comm.,vol. J19-B, no. 2, pp. 166-176, 2010.

[3] A. Asano, H. Nishiyama, and N. Kato, “The Effect of Packet Reordering and Encrypted Traffic on Streaming Content Leakage Detection (Invited Paper),”Proc. Int’l Conf. Computer Comm. Networks (ICCCN ’10), pp. 1-6, Aug. 2010.

[4] S. Amarasing and M. Lertwatechakul, “The Study of Streaming Traffic Behavior,”KKU Eng. J., vol. 33, no. 5, pp. 541-553, Sept. / Oct. 2006.

[5] Y.Gotoh, K.Suzuki, T.Yoshihisa, H.Taniguchi and M. Kanazawa, “Evaluation of P2P Streaming Systems for Webcast,” Proc. Sixth Int’l Conf. Digital Information Management, pp. 343-350, Sept. 2011.