DETECTION TECHNIQUE TO HANDLE VIDEO CONTENTS OF DIFFERENT LENGTHS IN TRUSTED NETWORKS

A.VIJAY VASANTH[1]

G.SHIVASHANKARI[2]

R.PRIYADHARSHINI[3]

1.Assisant Professor(Senior grade),Christ college of engineering and technology,Pondicherry,

2.Student,Christ college of engineering and technology,Pondicherry,

3.Student,Christ college of engineering and technology,Pondicherry,

ABSTRACT

Traffic balancing in the wireless network environment has an important impact on the performance. Multimedia streaming applications and services, trusted video delivery to prevent undesirable content-leakage has, indeed, become critical. Their detection performance substantially degrades owing to the significant variation of video lengths. We focus on overcoming this issue by proposing a round robin algorithm and best-partition searching algorithm that is robust to the variation of the video length. 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.

Good Traffic balancing makes wireless network more efficient and improves user satisfaction. This article introduces a better traffic balance model for the public Network based on the Networking concept with a switch mechanism to choose different strategies for different situations. The algorithm applies the game theory to the Traffic balancing strategy to improve the efficiency in the public Network environment.

KEY-WORDS -video lengths, bed-test experiment, delay variation, packet loss

1. INTRODUCTION

Over the last decade, researchers have studied how group communication applications like audio and video conferencing, multi-party games, content distribution, and broadcasting can be supported using IP Multicast [4]. However, over ten years after its initial proposal, IP Multicast is yet to be widely deployed due to fundamental concerns related to scalability, and support for higher layer functionality like reliability and congestion control. Recently, there has been a reevaluation by the research community of whether IP is indeed the right layer to support multicast-routing related functionality. A growing number of researchers [2, 3, 6, 9] have advocated an alternate architecture, where all multicast related functionality, including group management and packet replication, is implemented at end systems. We refer to such architecture as End System Multicast. In this architecture, end systems participating in a multicast group self-organize into an overlay structure using a completely distributed protocol. Further, end systems attempt to optimize the efficiency of the overlay by adapting to network dynamics and considering application level performance.

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 have increased by leaps and bounds. real-time video streaming communications such as web conference in intercompany 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 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.

We evaluate our techniques by testing the redesigned Naradaprotocol on a wide-area test-bed. Our test-bed comprisestwenty machines that are distributed around NorthAmerica, Asia and Europe. Our results demonstrate thatour techniques can provide good performance, both from theapplication perspective and from the network perspective.With our scheme, the end-to-end bandwidth and latencyattained by each receiver along the overlay is comparableto the bandwidth and latency of the unicast path from thesource to that receiver. Further, when our techniques are incorporatedinto Narada, applications can see improvementsof over 30–40% in both throughput, and latency. Finally,the costs of our approach can be

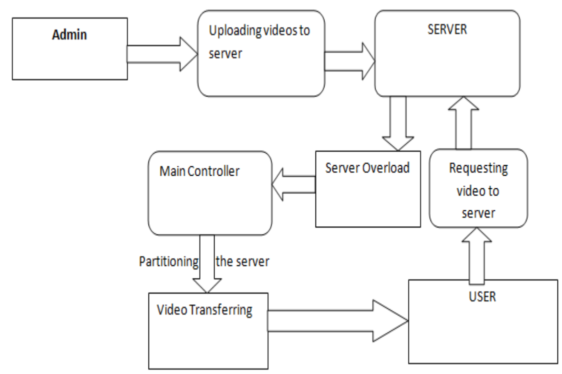


Fig 1. Overview of a leakage scenario and leakage detection scenario.

2. CONTENT LEAKAGE DETECTION

In this section, we first take a look at a typical video leakagescenario, and we present an overview of existing traffic patternbased leakage detection technologies.

A. Typical video leakage scenario

A typical contentleakagescenario can be described by the following steps asdepicted in Fig. 1. First, a regular user in a secure networkreceives streaming content from a content server. Then, withthe use of a P2P streaming software, the regular yet malicioususer redistributes the streaming content to a non-regular useroutside its network. Such content-leakage is hardly detectedor blocked by watermarking and DRM based techniques.

B. Leakage detection procedures

This topologyconsists of two main components, namely the traffic patterngeneration engine embedded in each router, and the trafficpattern matching engine implemented in the managementserver. Therefore each router can observe its traffic volumeand generate traffic pattern. Meanwhile, the traffic patternmatching engine computes the similarity between traffic patternsthrough a matching process, and based on specificcriterion, detects contents leakage. The result is then notifiedto the target edge router in order to block leaked traffic.

C. Pattern generation algorithm

Time slot-based algorithm is a straightforward solution togenerate traffic patterns by summing the amount of trafficarrival during a certain period of time, _t. In case somepackets are delayed, they may be stored over the followingslot, xi+1, instead of the primary slot, xi. Therefore, delayand jitter of packets distorts the traffic pattern, and as aconsequence, decreases the accuracy in pattern matching.Moreover, time slot-based algorithm is affected by packet loss.

Packet size-based algorithm defines a slot as the summationof amount of arrival traffic until the observation of a certainpacket size. Thisalgorithm only makes use of the packetarrival order and packet size, therefore is robust to change inenvironment such as delay and jitter. However, packet sizebasedalgorithm shows no robustness to packet loss.

D. Pattern matching algorithm

In pattern recognition, the degree of similarity is defined tobe the similarity measure between patterns [16]. The server sidetraffic patterns represents the original traffic pattern and isexpressed as XS = (x1; x2; :::; xS)t according to Eq. 1. Theuser-side traffic pattern is expressed as YU = (y1; y2; :::; yU)t.Here, S and U are number of slots, and the length of theuser-side observation is shorter than that of the server- side,i.e., S > U.

First, we set a window of size, U, which snips off apartial pattern, XU, from the server-side traffic pattern, XS.Next, we compute the similarity between the partial pattern,XU, and the user-side pattern, YU, (partial similarity). Thewindow is then moved from left to right by one slot. Thesethree steps are repeated until the window reaches the rightmostpart of the server-side pattern. Thus, we obtain (S - U + 1)values of similarity. The maximum value is thenretrieved andrepresents the degree of similarity of the compared videos.

$$X'_U = \begin{pmatrix} (x_1 - \overline{x})/s_x \\ (x_2 - \overline{x})/s_x \\ \vdots \\ (x_U - \overline{x})/s_x \end{pmatrix}, \qquad Y'_U = \begin{pmatrix} (y_1 - \overline{y})/s_y \\ (y_2 - \overline{y})/s_y \\ \vdots \\ (y_U - \overline{y})/s_y \end{pmatrix}$$

E. Leakage detection criterion

The similarity data obtained from thematching of time slot-based generated traffic patterns areconsiderably small and their distribution is considered to benormally distributed around zero, since the distribution ofcross-correlation coefficient values of two random waveformsis approximated to a normal distribution [17]. Therefore, [12]uses a dynamic decision threshold based on the Chebyshev'sinequality, and given by the following equation:

ϴ= min(μR + 4σR; 1:0);

Here,whether or not compared patterns are similar is decided bycomparing the maximum value of RXUYU with _ from Eq.4. Meanwhile, during the matching process of packet sizebasedgenerated traffic patterns, the similarity resulting fromthe comparison of different videos is considerably small,while the similarity resulting from the comparison of similarvideos is considerably large. A suitable fixed value is thereforeused as the decision threshold [12]. To determine whether ornot the compared traffic patterns are similar, the maximumvalue of RXUYU is retrieved and compared to the decisionthreshold, i.e.,max(RXUYU)> threshold, which indicates thatthe compared traffic patterns are similar.

F. Summary of the conventional methods

The conventional approaches, namely, Time slot-basedTraitor Tracing (T-TRAT), Packet size-based Traitor Tracing(P-TRAT) and Dynamic Programming based Traitor Tracing(DP-TRAT), based on the aforementioned algorithms aresummarized in Table I. The time slot-based pattern generationalgorithm used in T-TRAT is influenced by packet delay andjitter, which deteriorate the user-side traffic pattern. On theother hand, P-TRAT and DP-TRAT utilize a traffic patterngeneration method based on packet size instead of time-slot.As a result, P-TRAT and DP-TRAT show robustness againstpacket delay and jitter.

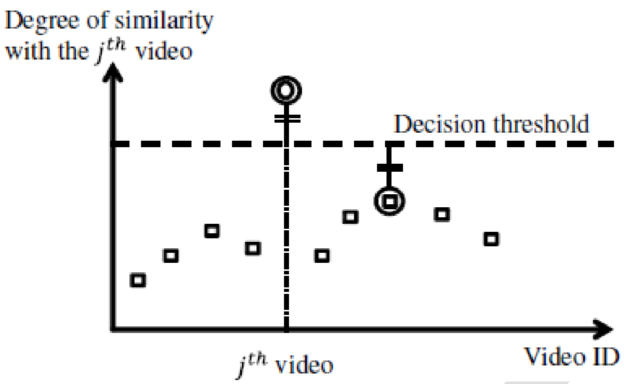


Fig 2. Description of decision threshold in existing leakage detection schemes.

3. PROPOSED SYSTEM

Traffic balancing schemes depending on whether the system dynamics are important can be either static or dynamic. Static schemes do not use the system information and are less complex while dynamic schemes will bring additional costs for the system but can change as the system status changes.A dynamic scheme is used here for its flexibility. The model has a main controller and balancers to gather and analyze the information. Thus, the dynamic control has little influence on the other working nodes. The system status then provides a basis for choosing the right Traffic balancing strategy.Thus, this model divides the public Network into several Networks. When the environment is very large and complex, these divisions simplify the Traffic balancing. The Network has a main controller that chooses the suitable s for arriving jobs while the balancer for each Network chooses the best Traffic balancing strategy.

3.1 Proposed Technique:

A)Best Partition Searching algorithm

The network partition balancer gathers load information from every node to evaluate the network partition status. This evaluation of each node’s load status is very important. The first task is to define the load degree of nodes. The node load degree is related to various static parameters and dynamic parameters. The static parameters include the number of CPU’s, the CPU processing speeds, the memory size, etc.

B.Round robin scheduling algorithm

The Round Robin algorithm is one of the simplest load balancing algorithms, which passes each new request to the next server in the queue. Round Robin based on the load degree evaluation”. The algorithm is still fairly simple. Before the Round Robin step, the nodes in the load balancing table are ordered based on the load degree from the lowest to the highest. The system builds a circular queue and walks through the queue again and again. Jobs will then be assigned to nodes with low load degrees. The node order will be changed when the balancer refreshes the Load Status Table. However, there may be read and write inconsistency at the refresh period T . When the balance table is refreshed, at this moment, if a job arrives at the network partition, it will bring the inconsistent problem.

When the flag = “Write”, the table is being refreshed, new information is written into this table. Thus, at each moment, one table gives the correct node locations in the queue for the improved Round Robin algorithm, while the other is being prepared with the updated information. Once the data is refreshed, the table flag is changed to “Read” and the other table’s flag is changed to “Write”.

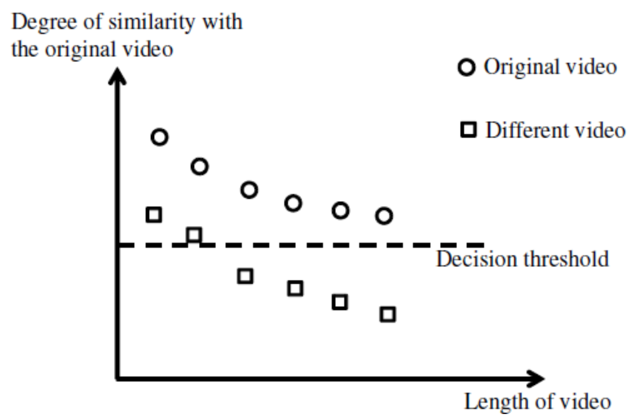


Fig 3 Example of erroneous decision in comparison of different length videos

3.2 ENHANCEMENT OF DETECTION TECHNIQUE TO HANDLE VIDEO CONTENTS OF DIFFERENT LENGTHS

However,the existence of videos of different lengths subjected totime variation in real content delivery environment causes DPTRAT'saccuracy to decrease. In this section, we take a lookat the issue caused by the existence of different length videosin network environments. While focussing on DP-TRAT, weintroduce a new threshold determination method based on anexponential approximation, and evaluate the computation costof both the proposed scheme and an eventual enhancement ofthe previous scheme.

A. Issue due to different lengths of videos

Traffic patterns of streaming videos represent the skeletoncarrying their characteristics [21], and are unique per content.Therefore, the longer the traffic pattern is, the moreinformationon the video it displays. In conventional methods, itis assumed that a certain length of content can always beobtained through the network for all contents. Therefore itis possible to utilize a fixed decision threshold in both PTRATand DP-TRAT methods. However, there is no suchguarantee in actual network environments. Fig3 shows anillustration of the occurrence of an erroneous decision in anetwork environment with different length videos.

B. Exponential approximation-based threshold determination and leakage detection

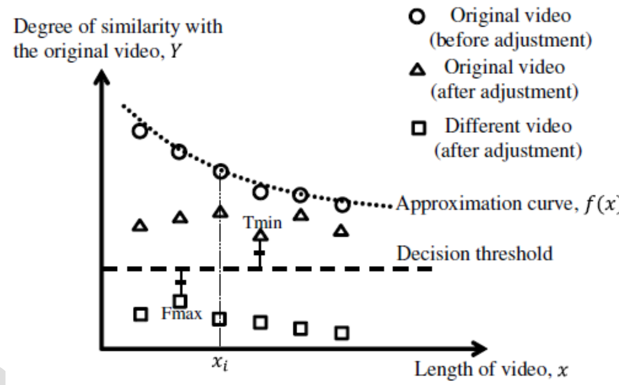


Fig 5. Determination of the decision threshold for detecting leakage.

Fig5 depicts the determination of the decision threshold.From the original video, we create portions of videos ofvarying lengths, and we generate their corresponding trafficpatterns. These patterns are then compared to the originaltraffic pattern to perform a sampling of thelength of videosand their corresponding degree of similarity. With the distributionof thesampling result, we perform an exponentialapproximation [22] of the form

f(x) = exp(α . x + β):

$$\alpha = \frac{n \cdot C - B \cdot D}{n \cdot A - D^2},$$

$$\beta = \frac{A \cdot B - C \cdot D}{n \cdot A - D^2},$$

$$A = \sum_{i=0}^{n} x_i^2, \qquad B = \sum_{i=0}^{n} \ln(f(x_i)),$$

$$C = \sum_{i=0}^{n} x_i \cdot \ln(f(x_i)), \qquad D = \sum_{i=0}^{n} x_i,$$

C. Schemes for Constructing Overlays

Our schemes for constructing overlays are derived fromthe Narada protocol [3], and differ from each other basedon which network metrics they consider. We compare thefollowing schemes for overlay construction:

• **Sequential Unicast:**To analyze the efficiency of a schemefor constructing overlays, we would ideally like to comparethe overlay tree it produces with the “best possible overlaytree” for the entire set of group members. We approximatethis by the Sequential Unicast test, which measures thebandwidth and latency of the unicast path from the sourceto eachrecipient independently (in the absence of other recipients).Thus, Sequential Unicast is not a feasible overlayat all but a hypothetical construct used for comparison purposes.

• **Random**: This represents a scheme that produces random,but connected overlay trees rooted at the source. Thisscheme also helps to validate our evaluation, and addressesthe issue as to whether our machine set is varied enoughthat just about any overlay tree yields good performance.

• **Prop-Delay-Only:**This represents a scheme that buildsoverlays based on propagation delay, a static network metric.Measuring propagation delay incurs low overhead, andoverlays optimized for this metric have been shown to yieldreasonably good simulation results [3]. In our evaluation,we computed the propagation delay of an overlay link bypicking the minimum of several one-way delay estimates.

• **Latency-Only and Bandwidth-Only:** These two schemesconstruct overlays based on a single dynamic metric withno regard to the other metric. They are primarily usedto highlight the importance of using both bandwidth andlatency in overlay construction.

• **Bandwidth-Latency:** This represents our proposed schemethat uses both bandwidth and latency as metrics to constructoverlays.

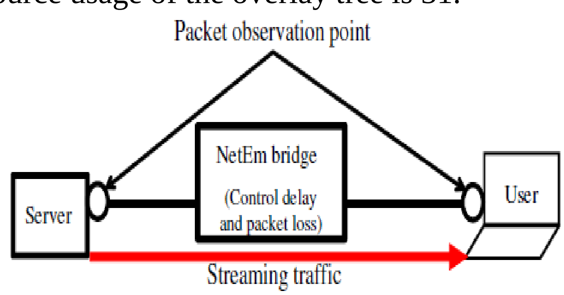
D. Performance Metrics

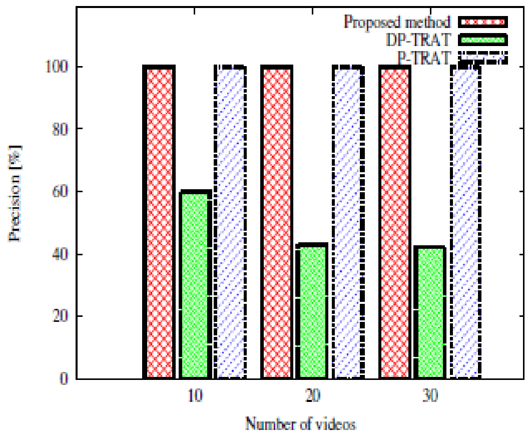
We use the following metrics to capture the quality of anoverlay tree:

• **Bandwidth:**This metric measures the application levelthroughput at the receiver, and is an indicator of the qualityof received video.

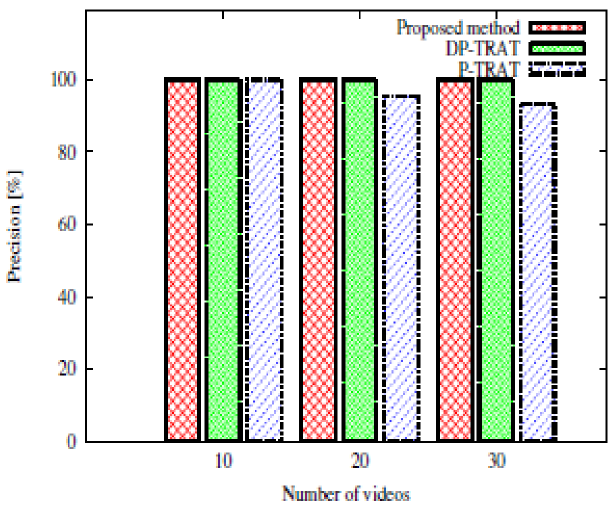
• **Latency:**This metric measures the end-to-end delay fromthe source to the receivers, as seen by the application. Itincludes the propagation and queuing delays of individualoverlay links, as well as queueing delay and processingoverheadat end systems along the path. We ideally wish tomeasure the latency of each individual data packet. However,issues associated with time synchronization of hostsand clock skew adds noise to our measurements of one-waydelay that is difficult to quantify. Therefore, we choose toestimate the round trip time (RTT). By RTT, we refer to thetime it takes for a packet to move from the source to a recipientalong a set of overlay links, and back to the source, usingthe same set of overlay links but in reverse order. Thus, theRTT of anoverlay path S-A-R is the time taken to traverseS-A-R-A-S. The RTT measurements include all delays associatedwith one way latencies, and are ideally twice theend-to-end delay.

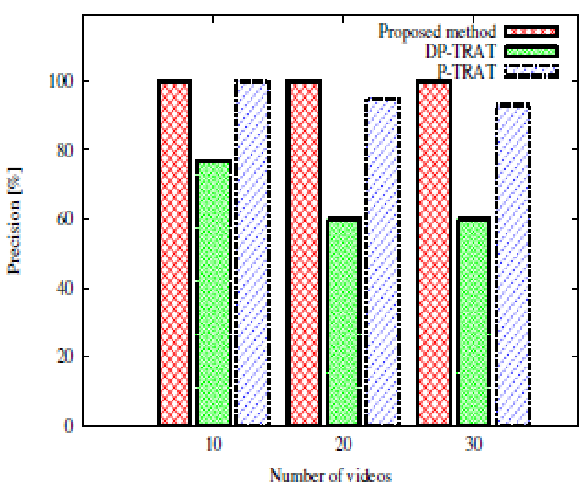
• **Resource Usage:** This metric defined in [3] captures thenetwork resources consumed in the process of delivering datato all receivers. The resource usage of an overlay tree is thesum of the costs of its constituent overlay links. The cost ofan overlay link is the sum of the costs of the physical linksthat constitute the overlay link. In our evaluation, we assumethe cost of a physical link to be the propagation delayof that link, guided by the intuition that it is more efficientuse of network resources to use shorter links than longerones. For example, in Figure 1, the cost (delay) of physicallink R1 − R2 is 25, the cost of the overlay link A− C is 27,and the resource usage of the overlay tree is 31.





(a) Accuracy





Robustness to network environment changes

To evaluate the robustness of the proposed scheme to the variation in network environment, we perform two experiments.Here, we consider a network environment similar to the previous, with 30 videos of lengths varying from 30 to 300 seconds. For the first experiment, we generate delay at the NetEm varying from 0 to 200ms every 25ms. Fig. 9. shows that none of the methods is affected by delay. This is due to the fact that all of these methods generate traffic patterns using the packet size-based generation algorithm, which shows robustness against packet delay jitter.

CONCLUSION

Focus of this paper is to develop an effective traffic balancing algorithm using round robin optimization technique to maximize or minimize different performance parameters like CPU load, Memory capacity, Delay or network load for the networks of different sizes. 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.

REFERENCE

[1] Y. Chu, S.G. Rao, S. Seshan, and H. Zhang, "Enabling Conferencing Applications on the Internet Using an Overlay Multicast Architecture," Proc. ACM SIGCOMM, pp. 55-67,
Aug. 2001.

[2] E.I. Lin, A.M. Eskicioglu, R.L. Lagendijk, and E.J. Delp, "Advances in Digital Video Content Protection," Proc. IEEE, vol. 93, no. 1, pp. 171-183, Jan. 2005.

[3] Y. Liu, Y. Guo, and C. Liang, "A Survey on Peer-to-Peer Video Streaming Systems," Peer-to-Peer Networking and Applications, vol. 1, no. 1, pp. 18-28, Mar. 2008.

[4] O. Adeyinka, "Analysis of IPSec VPNs Performance in A Multimedia Environment," School of Computing and Technology, University of East London.

[5] E.I. Lin, A.M. Eskicioglu, R.L. Lagendijk, and E.J. Delp, "Advances in digital video content protection," Proc. IEEE, vol.93, no.1, pp.171-183, Jan. 2005

[6] S. Craver, N. Memon, B. L. Yeo, and M. M. Yeung, "Resolving rightful ownerships with

invisible watermarking techniques: Limitations, attacks, and implications," IEEE J. Sel. Areas Commun., vol.16, no.4, pp.573- 586, May 1998.
[7] M. Barni and F. Bartolini, "Data hiding for fighting piracy," IEEE Signal Process. Mag., vol.21, no.2, pp.28-39, Mar. 2004.
[8] K. Su, D. Kundur, and D. Hatzinakos, "Statistical invisibility for collusion-resistant digital video watermarking," IEEE Trans. Multimedia, vol.7, no.1, pp.43-51, Feb. 2005.