User Behavior Detection Based on Statistical Traffic Analysis for Thin Client Services

Mirko Suznjevic[1], Lea Skorin-Kapov[1], and Iztok Humar[2]

[1] University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3,
10000 Zagreb, Croatia,
{mirko.suznjevic, lea.skorin-kapov}@fer.hr,
[2] University of Ljubljana, Faculty of Electrical Engineering, Tržaška 25, 1000
Ljubljana, Slovenia
iztok.humar@fe.uni-lj.si,

Abstract. Remote desktop connection (RDC) services offer clients access to remote content and services, commonly used to access their working environment. With the advent of cloud-based services, an example use case is that of delivering virtual PCs to users in WAN environments. In this paper, we aim to analyze common user behavior when accessing RDC services. We first identify different behavioral categories, and conduct traffic analysis to determine a feature set to be used for classification purposes. We then propose a machine learning approach to be used for classifying behavior, and use this approach to classify a large number of real-world RDCs. Obtained results may be applied in the context of network resource planning, as well as in making Quality of Experience-driven resource allocation decisions.

Keywords: user behaviour, remote desktop connection, traffic classification, machine learning

1 Introduction

Today numerous solutions exist on the market supporting remote desktop connections (RDC), including those provided by Microsoft, Oracle, and Citrix. With the advent of cloud computing and data centres offering virtual desktop solutions to end users, thin-client solutions are becoming an increasingly popular mechanism for end users to access and interact with remote content and services in an easy-to-maintain and cost effective manner.

A key challenge in making such solutions viable from an end user point of view is meeting the stringent network performance requirements dictating low delays and high response times [1]. As opposed to a standard, "local" desktop, whereby user inputs are locally processed and rendered nearly immediately, RDCs require inputs to be transmitted to a remote computer, processed, and returned to the thin-client [2]. Consequently, the screen updates and response times become a critical issue, in particular in WAN environments.

RDC traffic generally corresponds to encrypted video bitmaps exchanged between a remote virtual PC and a thin-client using a single server port, imposing challenges in terms of detecting end user tasks and applications that are being remotely run (e.g., a user editing a document, browsing the Web, or viewing audio/video content)[3]. RD protocols (e.g., Microsoft Remote Desktop Protocol, RDP) run over TCP and offer a reliable connection. Different user behaviours while using RDC (in terms of conducted task or application being used) result in different traffic characteristics and different impacts of network performance on user perceived quality [4]. Therefore, a prerequisite in effectively managing the end user Quality of Experience (QoE) is related to determining the user behavior in the context of RDC for the purpose of accurately mapping between QoE and performance indicators (i.e., delay, bandwidth).

In this paper, we focus on detecting user behavior while using RDC based on employing statistical traffic analysis and machine learning. We compare achieved results with those reported in related work [3]. Further, going beyond related work, we apply our proposed approach to real-world RDC traffic traces collected at the University of Ljubljana to study actual user behavior. Knowledge regarding such traffic characteristics may prove beneficial in making QoE estimations and resource planning decisions.

The paper is structured as follows. In section 3 we present a traffic classification algorithm for RDC employing a machine learning approach. In section 4, we use our algorithm to analyze actual user behavior captured in 18.5 GB of real RDC traces. Section 5 presents concluding remarks and future research.

2 Related Work

Traffic classification.

While approaches such as those based on Deep Packet Inspection have been commonly used to identify and classify network traffic [5], a key drawback lies in the fact that traffic may be encrypted, preventing payload analysis. Additional drawbacks include necessary knowledge regarding payload formats, and potential government imposed privacy regulations. Approaches based on statistical traffic analysis present a different approach, whereby relevant traffic features (e.g., packet length and inter-arrival times) are extracted and analysed to identify a particular application. Emmert et al. [6] previously reported on traffic patterns for different types of thin client users when working with popular office applications such as Microsoft Word, Excel, or PowerPoint. A thorough workload characterization and modeling (session arrival process, interarrival times, duration) of RDC traffic was done by Humar *et al.* [7]. The authors further confirmed self-similar characteristics of RDC traffic at packet level in [8].

In the past, Machine Learning (ML) algorithms have also been used for traffic classification purposes [9, 10]. Dusi *et al.* [3] propose a novel approach to RD application identification based on IP-level statistical traffic analysis and machine learning techniques, their goal being to make QoE estimations targeted towards verifying that Service Level Agreements are met. In their tests, the authors ran several applications over an RDP connection, categorized as audio (e.g., VLC, WMP, skype), video (e.g., Flash video, WMP), and data (e.g., Web browsing, Adobe reader). The authors extracted traffic features (bit rate and packet rate at IP and TCP levels, TCP payload length, and number of observed packets) and evaluated different statistical classification techniques in terms of accuracy. Their results show that on-the-fly application identification for thin-client flows can achieve over 90 percent accuracy (when applying Support Vector Machines and Decision Tree algorithms, and considering a window size of 10 s) even when the testing set includes applications for which their classifier was not trained.

Relation to QoE studies.

Previous research has studied RDC traffic and its relation to user perceived QoE. The user experience on RDC was quantified by Tolia *et al.* [11] through operation response times on different bandwidths (10 and 100 Mbps) and for different types of office applications (typing, tracking changes, creating slides for presentation and manipulating photos). Staehle *et al.* [4] studied how different network parameters (packet loss, jitter and delay) influence subjective and objective QoE in a controlled testbed environment. Further, while Dusi *et al.* proposed a mapping between QoE scores and round trip time (RTT) thresholds for different types of RD applications, a more extensive study addressing the QoE of remote desktop users was conducted by Casas *et al.* [2]. In a laboratory study involving 52 participants, the authors looked to identify relationships between network performance parameters (i.e., delay, bandwidth) and end user subjective quality scores for four typical remote desktop tasks found in enterprise scenarios: text typing (e.g., email and document editing), screen scrolling (e.g., web browsing), drag and drop of images, and menu browsing (e.g., menu selection). The authors further provide a traffic characterization whereby they constitute that the four aforementioned tasks present nearly identical traffic patterns in the uplink, while in the downlink typing sends much less data than the other three tasks, which share a common packet size behavior. Differences in throughput are linked with different requirements with regards to sending screen updates. Given that RDC traffic uses TCP as a transport protocol, network delay has been found to have the most significant impact on end user QoE.

In this paper, we focus on user behavior prediction for RDC and build on the work reported by Dusi *et al.* [3]. Given that their results showed that the Decision Tree ML algorithm (tested using the freely available WEKA software [12]) provided very good results in terms of traffic classification accuracy, we have opted to use this algorithm in our study.

3 User behavior detection based on machine learning

3.1 Methodology

The first step in our work was to train a ML algorithm to classify different types of RDC behavioral activities, to be subsequently used for the analysis of real RDC traffic traces. The behavior labelled traces which were used for training the Decision Tree algorithm were collected at the Department of Telecommunications, Faculty of Electrical Engineering and Computing, University of Zagreb. The capture was performed on Dell Optiplex 390 computer (configuration: i3@3,3 GHz, 4GB RAM, ATI Radeon HD 6450) connected to another Dell Optiplex 390 via a 100 Mbit LAN. The behavior categories we have addressed are as follows:

- idle - no actions for 10 seconds, static screen (desktop image);
- document editing - editing of a word document (including text writing, picture pasting, text copying etc.);
- browsing - searching for accommodation on the www.booking.com web page;
- audio - listening to a 128bit/s online radio station from www.radio365.com;
- video - watching a 10 minute full-screen movie on www.youtube.com.

The measurements for each of the behavior categories lasted 10 minutes, i.e., during those 10 minutes, only a given action was being performed. We note that previous cited efforts [3, 2] have not considered the category of *idle*, even though this is commonly observed in RDC traffic. The reason is that previous efforts focused either on identifying certain applications being run over an RDC or studying end user QoE, while our goal is to study overall user behavior exhibited when using RDCs. Following identification of a number of traffic features to be used for classification, we conducted a traffic analysis and specified a decision tree algorithm. We then used our collected traces to train the ML algorithm. Finally, we validated the algorithm using a validation dataset.

3.2 Traffic feature extraction

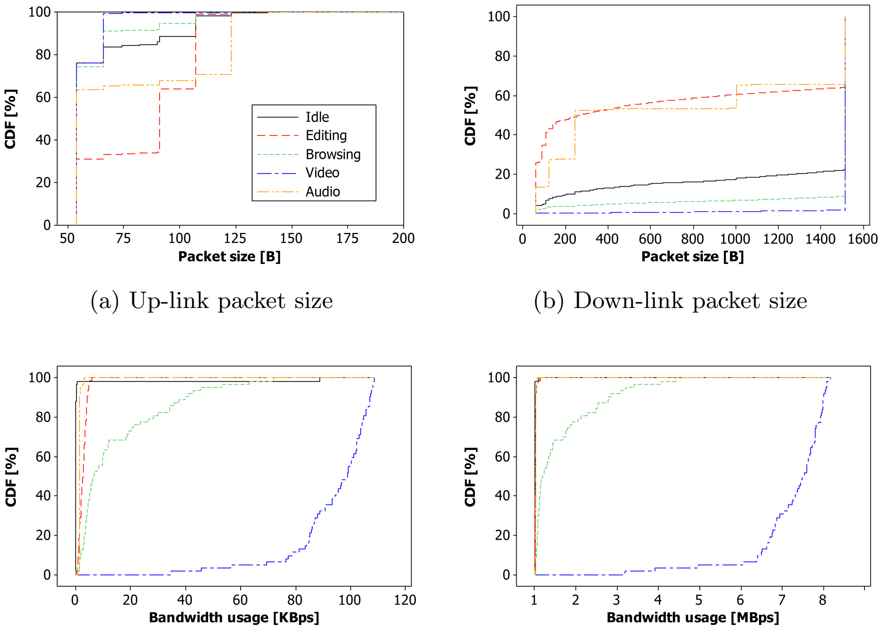
Features are attributes of flows calculated over multiple packets, used to train a ML classifier in associating sets of features with given application types [9]. Behavior labelled traces were processed such that for an epoch (time window) of 10 seconds, six features were extracted:

- up-link packet number;
- up-link average packet size;
- up-link average bandwidth usage;
- down-link packet number;
- down-link average packet size;
- down-link average bandwidth usage;

By *up-link* we refer to packets originating from the RDC client, while *down-link* refers to packets originating from the RDC server. The reason for choosing an epoch of 10 seconds is due to the fact that previous research has shown this approach as providing good results in terms of accuracy [3]. Traffic features were extracted from the network traffic traces using a custom built Java-based parser.

3.3 Machine learning approach

Following traffic capture, we analyzed the traces to determine different traffic patterns for the identified behavioral categories. With regards to the traffic characteristics, we note that the most distinguishing feature was found to be the number of down-link packets. Figure 1 portrays CDFs for packet size in both uplink and downlink directions, as well as CDFs for bandwidth usage across the different behavioral categories. In the case of traffic corresponding to the client being idle, only keep-alive packets were sent from server to client. As a result, we subsequently classified all traffic with <5 packet/s as idle. Video clearly represented the most resource demanding traffic, with average bit rates over 7 MBit/s. We once again note that we used full-screen video. Such high resource demands resulted in poor video quality in terms of image freezes and jerkiness, linked also to the fact that TCP usage is not optimized for real-time media. Browsing and editing exhibited highly variable traffic characteristics, with browsing resulting in higher down-link bandwidth utilization. Nevertheless, we presume that this is highly dependent on the end user interaction behavior resulting in screen updates. With regards to audio, we used a 128-bit radio channel and a static screen, i.e., there was no need for screen updates. We further note that all behavioral categories exhibit similar up-link characteristics (mostly composed of ACK packets), as also reported by Casas et al [2]. In the case of video, the higher bandwidth usage is due to the large number of ACKs being transmitted.



(a) Up-link packet size

(b) Down-link packet size

(c) Up-link bandwidth usage

(d) Down-link bandwidth usage

Fig. 1. Traffic analysis of RDC traces

Having statistically analyzed captured traffic, we used the freely available WEKA Java library (developed at the University of Waikato, New Zealand) [12]. While the WEKA software contains a large collection of tools for data processing, machine learning, and data analysis, we focused on support for implementation of the Decision Tree algorithm (noted in WEKA as J.48). The algorithm we specified in our study is portrayed in Figure 2, relying on the identified feature set for classification purposes. During the training phase, the algorithm uses extracted features from the traffic samples to build the mapping functions from the samples to the behavior category [1]. We used the approach proposed in [3] to split the samples into fixed time epochs (i.e., time windows) of 10 s and extract the features for each epoch. However, while in [3] the authors use only down-link traffic for algorithm training, we incorporate also up-link samples.

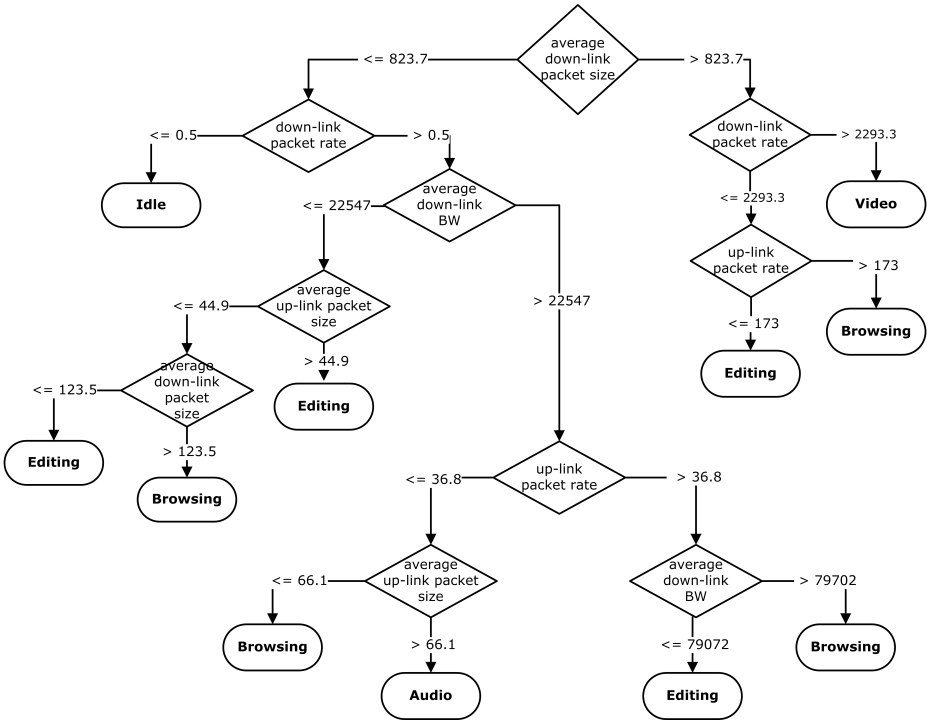


Fig. 2. A view of the Decision Tree algorithm used for traffic classification

Having completed the training phase, we generated an annotated dataset with the addressed behavioral categories for validation purposes. The accuracy in terms of correct classification (in terms of epochs) was found to be 78%,

[1] Interested readers may contact the authors for access to the data samples used for training purposes.

while the accuracy in terms of correctly classified bytes was found to be 89.7%. While this is somewhat lower than the accuracy reported by [3], we note that this may be attributable to the fact that we consider document editing and web browsing as different categories, whereas the cited authors considered a general "data" category (including web browsing, Adobe Reader, and Microsoft Powerpoint). Furthermore, the authors do not consider *idle* periods in their traffic classification. Finally, while the authors considered only down-link traffic, they used an extended feature set for classification purposes as compared to our set.

4 Analysis of real RDC traces

Following training and validation of the machine learning algorithm, we apply the algorithm to real RDC traces to study actual user behavior in a real world scenario (outside of a laboratory environment). To obtain empirical traffic traces, we collected packet-level traces on a 100 Mbit/s Ethernet link that connects the Faculty of Electrical Engineering, University of Ljubljana (FE) to the external Internet, as shown in Figure 3. The traces were collected from Sept. 24^{th}-Nov. 16^{th} 2012. We used a traffic capturing system on a personal computer running Linux OS and with a DAG network interface card. The traces were collected, filtered and stored in .cap format. We focused on the most frequently used RDC traffic, Microsoft Remote Desktop Protocol, which was recognized by transport layer (TCP) port numbers, configured to 3389 by default. To manage long traces, the individual files were limited to 0.5 GB. The traces contain RDC flows generated by students and staff of the Laboratory of Telecommunications, FE. The measurement procedure resulted in a total of 37 traffic traces which comprised 18.5 GB of RDC traffic. Each trace consisted of multiple RDCs and corresponding flows. A total of 1364 sessions were established when summarized across all traces (excluding sessions with a session length less than 10 s).

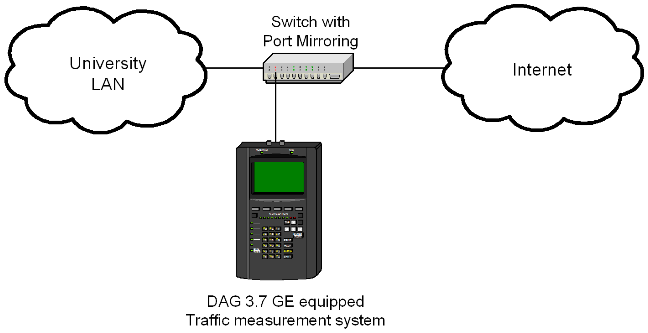


Fig. 3. Testbed environment for collecting RDC traces at FE, Ljubljana

Figure 4 shows the CDF for the different session lengths that were collected in the traces. Results show that while the majority of sessions were of short length, there is a significant portion of longer sessions, some lasting up to multiple days. This is indicative of the fact that users established RDCs and left them open for extended periods of time, even though the connections were for the most part idle.

Figure 5 summarizes the duration of total time that users spent engaged in the given behaviors across all considered traffic traces. Results clearly show that for the majority of time (92.91 %), the RDC was idle, indicating that no end user activity was detected. Document editing (6.14 %) proved to be the most common activity, with small percentages of time spent browsing (0.89 %) and listening to audio (0.06 %). Throughout the entire collection of traces, there was no detection of behavior indicating that end users were running video applications. This supports the assumption that the users involved in this study generally did not watch video content via a remote desktop connection, most likely due to both the nature of the tasks that users perform when using RDC, and also due to poor video quality resulting from network delays. While delay is not a determining issue in LAN scenarios, the use of RDC services in WAN scenarios clearly leads to the increased impact of RTT.

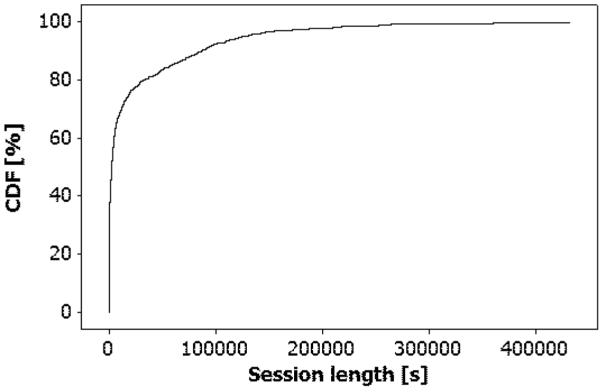


Fig. 4. CDF portraying session lengths for collected traces

5 Conclusions and future work

The conducted study has served to provide insight into the actual behavior users engage in when using thin client services. The applicability of such results can be considered in the context of efficient planning and allocation of resources in a cloud environment, whereby passive monitoring of network traffic can be used

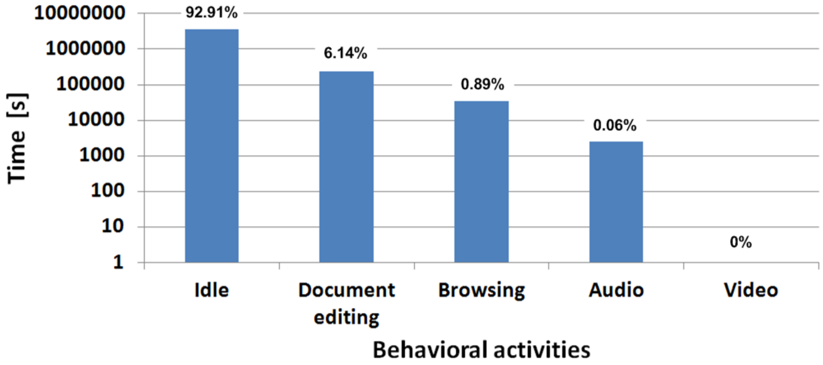


Fig. 5. Total time users spent engaged in a given behavioral category (logarithmic scale)

to determine user behavior and make estimations with regards to network resource requirements. Further, by relying on studies that have explored QoE for RDC services [4, 2, 3], behavior detection can be used together with utility functions modeling QoE as a function of network delay and bandwidth for different types of remote desktop services to make QoE estimations. An example of recent work focused on exploiting RDC application identification using statistical mechanisms for the purpose of QoE-driven scheduling is discussed in [13].

In our work, we have employed a machine learning approach to study user behavior, with actual traces collected in an academic setting. Future work will focus on improving our classification accuracy, and further analyzing additional real world scenarios, such as those focused on use of thin client services in an enterprise/business setting.

Acknowledgements. This work was partially supported by the Ministry of Science, Education and Sports of the Republic of Croatia and the Ministry of Higher Education and Technology of the Republic of Slovenia under the joint Croatia-Slovenia Cooperation in Science and Technology Project "Services and Applications in the Future Internet: Ensuring Quality and Energy Efficiency".

References

1. Lai, A.M., Nieh, J.: On the Performance of Wide-Area Thin-Client Computing. ACM Transactions on Computer Systems (TOCS) **24**(2) (2006) 175–209
2. Casas, P., Seufert, M., Egger, S., Schatz, R.: Quality of Experience in Remote Virtual Desktop Services. In: 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), IEEE (2013) 1352–1357
3. Dusi, M., Napolitano, S., Niccolini, S., Longo, S.: A Closer Look at Thin-Client Connections: Statistical Application Identification for QoE Detection. IEEE Communications Magazine **50**(11) (2012) 195–202

4. Staehle, B., Binzenhöfer, A., Schlosser, D., Boder, B.: Quantifying the Influence of Network Conditions on the Service Quality Experienced by a Thin Client User. In: Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 2008 14th GI/ITG Conference-, VDE (2008) 1–15
5. Sen, S., Spatscheck, O., Wang, D.: Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures. In: Proceedings of the 13th international conference on World Wide Web, ACM (2004) 512–521
6. Emmert, B., Binzenhöfer, A., Schlosser, D., Weiß, M.: Source Traffic Characterization for Thin Client Based Office Applications. In: Dependable and Adaptable Networks and Services. Springer (2007) 86–94
7. Humar, I., Bester, J., Tomazic, S.: Characterizing Graphical Desktop Sharing System's Workload in Collaborative Virtual Environments. In: Consumer Communications and Networking Conference, 2009, IEEE (2009) 1–5
8. Humar, I., Pustisek, M., Bester, J.: Evaluating Self-Similar Processes for Modeling Graphical Remote Desktop Systems' Network Traffic. In: 10th International Conference on Telecommunications, 2009, IEEE (2009) 243–248
9. Nguyen, T.T., Armitage, G.: A Survey of Techniques for Internet Traffic Classification Using Machine Learning. Communications Surveys & Tutorials, IEEE **10**(4) (2008) 56–76
10. Park, B., Won, Y.J., Choi, M.J., Kim, M.S., Hong, J.W.: Empirical Analysis of Application-Level Traffic Classification Using Supervised Machine Learning. In: Challenges for Next Generation Network Operations and Service Management. Springer (2008) 474–477
11. Tolia, N., Andersen, D.G., Satyanarayanan, M.: Quantifying Interactive User Experience on Thin Clients. Computer **39**(3) (2006) 46–52
12. University of Waikato: WEKA - Waikato Environment for Knowledge Analysis. available at http://www.cs.waikato.ac.nz/ml/weka/
13. Arumaithurai, M., Seedorf, J., Dusi, M., Monticelli, E., Lo Cigno, R.: Quality-of-Experience driven Acceleration of Thin Client Connections. In: 12th IEEE International Symposium on Network Computing and Applications (NCA), 2013, IEEE (2013) 203–210