Predicting Social Dynamics based on Network Traffic Analysis for CCN/ICN Management

Satadal Sengupta (Indian Institute of Technology Kharagpur, India; Email: satadal.sengupta@iitkgp.ac.in)

***Abstract*—Proliferation of online social networks (OSNs) has resulted in an unprecedented surge in the volume of multimedia content consumed by users on a daily basis. Popular OSNs such as Facebook enable users to view and share embedded videos and images on their feeds, which increases visibility, prompting repeated requests for the same piece of content. Maintaining desirable quality of service for all users becomes challenging in such a scenario, especially when low-bandwidth cellular network is being used for data download. Such problems have prompted the research community to focus heavily on the emerging paradigm of Information- or Content-Centric Networking (ICN/CCN), where in-network content management (e.g., content distribution, caching, etc.) forms the crux of an enhanced user experience. In this abstract, we argue that social dynamics among OSN users can provide concrete hints regarding future popularity of content. We propose a strategy to identify viewing and sharing patterns of Facebook users served by a cellular base station, by analyzing network traffic. We utilize these patterns to infer social dynamics among cellular users (mapped to cellphone numbers). We validate our strategy with proof-of-concept experiments on real data, and extensive simulations on a simulation framework proposed by us.**

I. Introduction

The cellular network has been subjected to unprecedented volumes of data traffic in recent times. The rise of online social networks (OSN), such as Facebook, have resulted in a never-seen-before demand for multimedia on mobile devices. Facebook, among other OSNs, allow embedded videos on the user's feed ("timeline"). These videos can be streamed in-place; there is a user-adjustable 'autoplay' feature which starts playing the video as soon as it comes in-focus on the application. These videos can be liked or shared by a user to make them appear on the feeds of other users socially connected to her. All these factors contribute to growing data demand at the content servers.

A new paradigm of networking, known as *Information-* or *Content-Centric Networking* (ICN/CCN) [1] has emerged, which supports named content access as opposed to the traditional *host-resolution* approach. CCNs employ the concept of *in-network management* for content storage and distribution, where intermediate routers or base stations apply content storage and distribution policies based on the underlying content usage patterns [1].

While CCNs assume content popularity as the default criterion for deciding content storage (caching) and distribution policies [3], these decisions are typically made based on the history of accesses for a specific content [3]. However, we argue that history may not always be indicative of future access patterns. For example, assume that an instructor shares a video with his class; typically, students would watch this video a number of times until the end of the course, and then stop watching it completely. A naive CCN caching implementation would cache the content for a considerable period of time, owing to heavy accesses in the recent past. An intelligent strategy would take into account social behaviour (or, dynamics) surrounding the instructor, instead of focusing solely on the content, to make caching decisions.

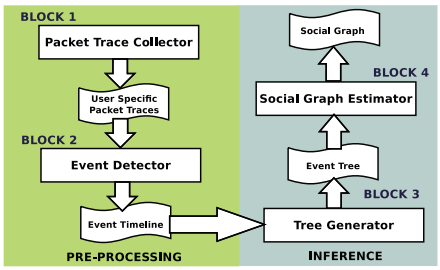

Fig. 1: System Architecture

While cellular networks do not have the ability to leverage on the social dynamics among users being served, it would be immensely useful to have a mechanism to do so. In this abstract, we propose a strategy to infer the social dynamics among users served by a base-station, by analyzing the packet traces generated at the base-station, to identify events from their OSN usage. Knowledge of OSN dynamics can be exploited to derive a caching policy at the base-station, which is based on the popularity of the content generators or content distributors (the persons who "share" a content on Facebook). While the prediction mechanism is inferred based on the analysis over a real data set from users (§ II), we also develop a comprehensive simulation framework to analyze the accuracy of our OSN behavior prediction mechanism (§ IV).

II. System Architecture and Design

Figure 1 summarizes the proposed system, and presents the building blocks. Blocks 1 and 2 together form the *Pre-Processing* subsystem while Blocks 3 and 4 combine to create the *Inference* subsystem.

Packet Trace Collector: This module runs on the cellular base-station as a background service, and generates packet traces for each user, using standard trace collection applications like `tcpdump`. We assume that it is possible for the operator at the cellular base-station to attribute each packet to a unique user, since it has information about the traffic generated by each cellphone number/user (this is how it charges cellular users for their data usage).

978-1-5090-4250-0/17/$31.00 ©2017 IEEE

Event Detector: This module first filters packets generated from *facebook.com*, using information in the packet headers. It then identifies 2 types of events: (1) *View:* We use traffic pattern for identifying the launch of a video – the bursty pattern for video is clearly distinguishable from the pattern generated during other user actions; the video ID is obtained from the DNS query at the start of video launch – the ID, even if encrypted, is unique. (2) *Share:* We observe that Facebook makes a call to `graph.facebook.com` when the share button is hit; our hypothesis is that a user will share a content within seconds of viewing it (completely or partially) if she likes it – we set the threshold at 20 secs in our experiments, i.e., if a user views a video, and follows that with a share within 20 secs, we assume that the same video has been shared. A time-stamp is attached to each identified event. These $(user\_id, content\_id, event, timestamp)$ quadruplets form the *events timeline*, which is the output of *Block 2*.

Tree Generator: This module takes into account the sequence of events as obtained in the *events timeline* and generates an *event tree* (or, cascade) for each unique piece of content. Every user is represented as a node of the tree and the parent-child relationship is inferred from the nature of the detected events and their order of occurrences. An *event forest* is thus created.

Social Graph Estimator: This module implements an event-tree aggregation strategy which produces a directed weighted graph as the intermediate output. The confidence of an edge originating from node v is computed as its weight normalized over the sum of all edge-weights for edges originating from v. Based on a threshold on this confidence value (a tunable parameter), some edges are dropped as coincidental edges, while the rest are retained. This filtering step produces the final directed, weighted inferred social graph. This inferred graph, we argue, embeds features which aids the operator in designing a superior caching mechanism.

III. Simulation Framework

Due to lack of large-scale real data, we propose a simulation framework to evaluate our proposed strategy. This framework replaces the "pre-processing" sub-system (Fig. 1). We model the following in our simulation framework:

(1) *Real-world social network* – We use an existing social (Facebook) graph consisting of 4039 nodes and 88,234 edges.

(2) *Influence propagation* – Each node in the graph is assigned 2 probability values for every adjacent node - one for view, another for share. The probabilities are assigned based on a binomial distribution.

(3) *Content popularity* – We assume that a given piece of content can belong to one of the three levels of popularity – (i) highly popular (level 3), (ii) popular (level 2), (iii) not-so-popular (level 1). Depending on the popularity level of the current content being circulated, nodes will exhibit increased or decreased likelihood of watching or sharing the content – this behavior is captured by providing a proportional boost to the probability values in P^v_{view} and P^v_{share}.

(4) *Content introduction* – Many users may independently watch a video and then share it with their respective friends. We randomly choose n points of introduction, where n is based on values reported in "Gesundheit' [2].

(5) *Content propagation* – Event trees start from each of the n nodes. An edge is created if the recipient of a share views a content; new edges are created from the recipient if she decides to share it further.

(6) *Base-station cluster* – We use a community detection algorithm to obtain a well-knit community from the friendship graph, which forms the base-station cluster. While simulations run on the entire graph, we assume that we have information regarding users in this cluster only (as happens in the case of a cellular base-station).

IV. Preliminary Results and Future Directions

We have studied the effect of changing the following variables in our simulation framework – (1) number of contents introduced in the graph at a time, (2) the (mean, standard deviation) pair of the probability distributions, (3) content popularity-level, (4) confidence threshold value. We present the effect of number of introduced contents only (Fig 2), in this abstract. We observe that the recall rate remains fairly high

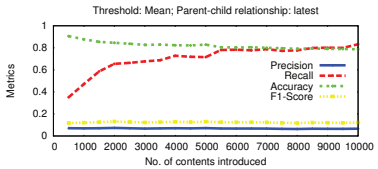

Fig. 2: Classification metrics for variation in content count

and shows an increasing trend; accuracy too remains high, and while it shows a slight decrease, the changes are marginal. Precision remains low, however, and as a consequence, F1-score remains low as well. Increasing recall and sustained high accuracy can be attributed to the fact that more number of contents allow for more difference in confidence values to build between genuine edge and coincidental edge cases, since more and more edges are added over time.

Although we are able to identify most 'good' edges (high recall), we pick many co-incidental edges as well (low precision). This can be traced back to our inability to decide with certainty *who viewed whose share*. The key future work of this strategy would be to investigate how inference precision can be improved. Furthermore, it would be interesting to see how the inferred information can be used to design an intelligent caching mechanism for CCNs.

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 1–12.

[2] E. Sun, I. Rosenn, C. Marlow, and T. M. Lento, "Gesundheit! modeling contagion through facebook news feed." in *ICWSM*, 2009.

[3] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, 2014.