KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2009
Cluj-Napoca (Romania), July 2–4, 2009, pp. 288–291

NEW DATA MINING TECHNIQUES FOR MACROFLOWS DELIMITATION IN CONGESTION CONTROL MANAGEMENT

DARIUS BUFNEA [(1)]

Abstract. State of the art approaches in Internet congestion control suggest the collaboration between streams in a so called macroflow, instead of the current approach, where streams compete with each other for scarce bandwidth. However, the macroflows granularity follows a simple approach, a macroflow being constructed on host pair bases. This paper presents new data mining techniques for grouping flows into macroflows based on their similar behavior over time.

1. Introduction

We are proposing in this paper a new method for grouping flows into macroflows based on their similar behavior. This paper generalizes and puts in a common template the methods suggested by the author in [2] and in [3], revealing that most state variables maintained inside the TCP/IP stack of a sender can be used in a similar fashion for macroflows identification. Also, we complement from a sender's perspective, the method designed to be implemented inside a receiver stack suggested in [4]. The advantage is a finer macroflow granularity which can be extended to all flows that share the same source LAN and the same destination LAN or even to the flows that share the same network bottleneck.

2. Formal Models

Our model is built around a highly accessed upload server (TCP sender) that maintains continuous data flows towards its clients. The goal is to infer in the incoming connection subsets containing connections having a similar behavior over time. A Congestion Manager running inside the TCP/IP stack of our upload server will treat such an inferred subset as a macroflow. We denote by S the upload server itself or its Internet IP address. Each incoming connection from a client is identified by a pair (C_{IP}:C_{port}) where C_{IP} is the client IP's address and C_{port} is the client used port for the outgoing connection. During a connection life time, server S will periodically measure and store values of some state variables such as the congestion window's size or the round trip time.

Round Trip Time Vectors. From the point of view of the upload server S, the

2000 *Mathematics Subject Classification.* 68M14, 90B18, 90B20.
Key words and phrases. congestion control, macroflow.

incoming connection $f = (C_{IP} : C_{port})$ during the time interval (t_b, t_e) is described by the *Round Trip Time (RTT) vector* $V = (r_1, r_2, \ldots, r_k)$ where: $(t_b, t_e) \subseteq (C_{IP} : C_{port})$ connection's life time; Δt is the interval between two consecutive measurements; $k = (t_e - t_b)/\Delta t$; r_i is the RTT value measured at the $t_b + \Delta t * (i-1)$ time moment. We say that the *RTT vector* associated to a connection describes the connection's behavior. For two connections f_1 and f_2 coming from the same client or LAN the RTT values measured at the same moment in time are quasi-identical. Therefore, their associated *RTT vectors* during the same time interval are also quasi-identical. This means that f_1 and f_2 manifest a similar behavior, which justifies their placement in the same macroflow.

Congestion Window Size Vectors. From the point of view of the upload server S, the incoming connection f=$(C_{IP} : C_{port})$ during the time interval (t_b, t_e) is described by the *Congestion Window Size (CWnd) vector* $V = (r_1, r_2, \ldots, r_k)$ where: $(t_b, t_e) \in (C_{IP} : C_{port})$ connection's life time; Δt is a fixed time interval; $k = (t_e - t_b)/\Delta t$; $r_i = 0$ if the congestion window size decreased at least once during the time interval $T_i = [t_b + \Delta t * (i-1), t_b + \Delta t * i)$, and $r_i = 1$ otherwise (e.g. the congestion window size increased or remained constant during that time interval), $1 \leq i \leq k$. For a connection f, the congestion window size represents its own estimation about the network's available transport capacity. A decrease of the congestion window size occurs when a congestion situation appears along the network path from S towards the destination host. If, during a larger time interval, the congestion window size decreases for two connections f_1 and f_2 in approximately the same time this means that congestion happens for both of them together, in the same moments. So it is very likely that these two connections share a bottleneck. For this reasons, it is justified to place f_1 and f_2 in the same macroflow.

Similarity and Distance Measures in the RTT Vector Space. We associated to a connection an *RTT vector* describing its behavior. The *RTT vector* reflects the RTT temporal evolution of that flow. Two connections will be considered more similar as they are more linearly correlated. A statistical measure for the linear correlation of two vectors is the *Pearson coefficient*. Given two connections, $f_1 = (C^1_{IP} : C^1_{port})$ and $f_2 = (C^2_{IP} : C^2_{port})$ measured during the time interval (t_b, t_e) and their associated *RTT vectors* $V_1 = (r_{11}, r_{12}, \ldots, r_{1k})$ and $V_2 = (r_{21}, r_{22}, \ldots, r_{2k})$, the *Pearson correlation coefficient* of f_1 and f_2 is defined as: $P(V_1, V_2) = \frac{\sum_{i=1}^{k} (r_{1i} - \overline{r_1}) \cdot (r_{2i} - \overline{r_2})}{\sqrt{\left(\sum_{i=1}^{k} (r_{1i} - \overline{r_1})^2\right)\left(\sum_{i=1}^{k} (r_{2i} - \overline{r_2})^2\right)}}$, where $\overline{r_1}$ and $\overline{r_2}$ are the mean values of V_1 and V_2. The similarity measure we use for comparing connections will be: $\overline{P}(V_1, V_2) = \frac{P(V_1, V_2) + 1}{2}$. For differentiating connections the distance function is defined by: $d_P(V_1, V_2) = 1 - \overline{P}(V_1, V_2)$.

Similarity and Distance Measures in the CWnd Vector Space. This section will reveal the distance and the similarity measures used in clustering process in the *CWnd vector* space. A *Cwnd vector* reflects the Cwnd timely evolution of that flow. Two connections will be considered more similar as they meet congestion together more often. We express next the similarity of two given connections, $f_1 = (C^1_{IP} : C^1_{port})$ and $f_2 = (C^2_{IP} : C^2_{port})$ measured during the time interval (t_b, t_e), in terms of their associated *CWnd vectors* $V_1 = (r_{11}, r_{12}, \ldots, r_{1k})$ and $V_2 = (r_{21}, r_{22}, \ldots, r_{2k})$.

Definition 1. Given a radius *step*, which is an integer number, $0 \leq step \leq k$, and a time interval $T_i = [t_b + \Delta t * (i-1), t_b + \Delta t * i)$, $1 \leq i \leq k$ we call f_1 and f_2:
a) *Congestion Neighbors* on interval T_i iif either: $r_{1i} = r_{2i} = 0$, which means that during T_i both streams faced congestion or $r_{1i} \neq r_{2i}$ and $\exists d \in \{1, 2\}$ so that $r_{di} = 0$ and $\exists j$, $\max\{1, i - step\} \leq j \leq \min\{k, i + step\}$ so that $r_{3-d,j} = 0$.
b) *Congestion Disassociated* on interval T_i iif $r_{1i} \neq r_{2i}$ and $r_{di} = 0$, $d \in \{0, 1\}$ and not $\exists$ j, $\max\{1, i - step\} \leq j \leq \min\{k, i + step\}$ so that $r_{1-d,j} = 0$.

Definition 2. Given a radius *step*, which is an integer number, $0 \leq step << k$ we define for f_1 and f_2 the following sets:
a) $CN(V_1, V_2) = \{i \mid f_1$ and f_2 are Congestion Neighbors on $T_i,\ i = 1..k\}$;
b) $CD(V_1, V_2) = \{i \mid f_1$ and f_2 are Congestion Disassociated on $T_i,\ i = 1..k\}$.

Given a radius *step*, $0 \leq step << k$, the congestion similarity coefficient of f_1 and f_2 is $CS(V_1, V_2) = \begin{cases} \frac{|CN(V_1,V_2)| - |CD(V_1,V_2)|}{|CN(V_1,V_2)| + |CD(V_1,V_2)|}, & if\,|CN(V_1, V_2)| + |CD(V_1, V_2)| \geq 0, \\ 0, otherwise \end{cases}$. For differentiating connections the congestion distance function is defined by: $d_C(V_1, V_2) = \frac{1 - CS(V_1, V_2)}{2}$.

3. Macroflows identification using clustering techniques

Let $F = \{f_1, f_2, \ldots, f_n\}$ be the set of all incoming concurrent connections served by S. For the (t_b, t_e) time interval, the server will take samples of the state variables values that we choose to describe a flow's behavior. Function of the chosen state variables, we will use the corresponding distance and similarity measures. For the (t_b, t_e) time interval, we consider the measured *RTT* or *CWnd* vectors $V = \{V_1, V_2, \ldots, V_n\}$, where V_i is the vector associated to the f_i connection, $f_i = (C^i_{IP} : C^i_{port}), V_i = (r_{i1}, r_{i2}, \ldots, r_{ik})$, $i = 1..n$. We use an agglomerative hierarchical clustering algorithm [1] for grouping in macroflows the concurrent connections described by similar cwnd vectors. This bottom-up strategy starts by placing each connection in its own cluster (macroflow) and then merges these atomic clusters into larger and larger clusters (macroflows) until a termination condition is satisfied. At each iteration, the closest two clusters (macroflows) are identified. The distance between two clusters M_i and M_j is considered to be the maximum distance of any pair of objects in the cartezian product $M_i \times M_j$. If the distance between these two closest clusters does not exceed a given threshold *thr_max_dist*, we merge them and the algorithm continues by a new iteration. Otherwise, the algorithm stops.

Algorithm $MacroflowIdentification$ is:
Input: n, the number of concurrent connection at server S;
$F = \{f_1, f_2, \ldots, f_n\}$ the set of concurrent connection at S;
$V = \{V_1, V_2, \ldots, V_n\}, V_i = (r_{i1}, r_{i2}, \ldots, r_{ik})$, $i = 1..n$, the vectors associated to the connections;
thr_max_dist, the maximal distance threshold for two connections to be admitted in the same

macroflow.
Output: m, the number of macroflows inferred in the concurrent connections set;
$M = \{M_1, \ldots, M_m\}$, the inferred macroflows, where
$M_i \neq \phi, i = 1..m, \bigcup_{i=1}^{m} M_i = F, M_i \cap M_j = \phi, i, j = 1..m, i \neq j.$

```
m := n; M := φ;
For i:= 1 to m do M_i := {f_i}; M := M ∪ {M_i}; End For;
 While (m > 1) and ( Continue (M, thr_max_dist, M_merge1, M_merge2) == true) do
  M_new := M_merge1 ∪ M_merge2;
  M := M − {M_merge1, M_merge2} ∪ {M_new};
  m := m-1;
 End While;
End Algorithm.
Function Continue (M the set of current macroflows, thr_max_dist, out M_merge1, out M_merge2):boolean
is
 min_dist := ∞;
 For each M_i ∈ M
   For each M_j ∈ M, M_j ≠ M_i
     dist(M_i, M_j) = max{d(v_r, v_t)|f_r ∈ M_i, f_t ∈ M_j};
     If dist(M_i, M_j) < min_dist
       min_dist := dist(M_i, M_j); M_merge1 := M_i; M_merge2 := M_j;
     End If;
   End For;
 End For;
 If min_dist < thr_max_dist Return True; Else Return False; End If;
End Function.
```

Function *Continue* determines the closest two clusters from the clusters set M. It will return true if these clusters are closer than *thr_max_dist* and false otherwise. For $d(v_r, v_t)$ we will use either $d_C(v_r, v_t)$ or $d_P(v_r, v_t)$, function of the chosen state variable.

4. Conclusions and Future Work

We suggested in this paper a data model for extending the macroflow granularity outside of the host-pair approach. Our method will prove its advantages in a Congestion Manager framework. As future work we plan to explore the use of different similarity measures and other state variables to compare the timely evolution of the connections being analyzed.

References

[1] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2001.

[2] D. V. Bufnea, A. Câmpan and A. S. Dărăbant. Fine-Grained Macroflow Granularity in Congestion Control Management. *Studia Universitatis*, Vol. L(1), pp. 79-88, 2005.

[3] A. Câmpan and D. V. Bufnea. Delimitation of Macroflows in Congestion Control Management Using Data Mining Techniques. *4th ROEDUNET International Conference, Education/Training and Information/Communication Technologies* - ROEDUNET '05, Romania, pp. 225-234, 2005;

[4] D. V. Bufnea. A New Method for Macroflows Delimitation from a Receiver's Perspective. *Proceedings of the IEEE 2nd International Conference on Computers, Communications & Control (ICCCC 2008)*, Felix Spa, Romania, Vol. III (2008), pp. 201-205.

(1) Faculty of Mathematics and Computer Science, Babes-Bolyai University, 1 M. Kogalniceanu, 400084 Cluj-Napoca, Romania

E-mail address: bufny@cs.ubbcluj.ro