**Available Online at www.ijcsmc.com**

**International Journal of Computer Science and Mobile Computing**

**A Monthly Journal of Computer Science and Information Technology**

**ISSN 2320–088X**
**IMPACT FACTOR: 6.199**

***IJCSMC, Vol. 9, Issue. 3, March 2020, pg.38 – 42***

# TO IMPROVE NETWORK PERFORMANCE MEASUREMENT USING INCREMENTAL COUNTER ESTIMATION BROADCAST ALGORITHM

**Dr. P.Chitti Babu[1]; D.J.Samatha Naidu[2]; A.Syam Prasad[3]**
[1]Professor & Principal, APGCCS, Rajampet, YSR Kadapa, India
[2]Assistant Professor, MCA Department & APGCCS, Rajampet, YSR Kadapa, India
[3]MCA Department & APGCCS, Rajampet, YSR Kadapa, India
[1] drpcbit@gmail.com; [2] samramana44@gmail.com; [3] avvarusyam136@gmail.com

***Abstract— Networks are collection of autonomous system, generally networks degradation problem are upcoming research challenge for researchers to improve performance measurement proposing a new novel based algorithm.***
***Keywords— Introduction, ICEB Problems, ICEB.***

## I. Introduction

Measurement applications track several million flows and their counters are updated with the arrival of every packet. These capabilities are an important enabling factor for networking algorithms in many fields such as load balancing, routing, fairness, network caching and intrusion detection  Counter arrays are also used in popular approximate counting sketches such as multi stage filters and count min sketch [9], as well as in network monitoring architectures. Such architectures are used to collect and analyze statistics from many networking devices.

## II. Related Work

**In Existing Work,** the problem of transmitting energy consumption optimization is referred to a Minimum-Energy Multicast (MEM) problem, and has been extensively studied in static networks. Since both MEM problem and Minimum- Energy Broadcast (MEB) problem, a special case of MEM, are proven to be NP-hard, the community focuses on designing efficient heuristics energy efficient multicast/broadcast algorithms for specific applications is also an active topic as noted earlier, prior literature only focuses on the minimization of transmitting energy, relies on the assumption that the receiving energy is directly proportional to the number of receivers or considers the receiving energy in the context of network-wide broadcast.

In contrast, the proposed framework in our work jointly minimizes the transmitting and receiving energy, with applicability to more general scenarios. The major limitations are there is no Accurate Less Energy Node detection due to absence of filtering techniques. More packets drops frequently.

## III. PROPOSED WORK

**The Proposed Work,** formulated our problem called solution for energy consumption and end-to-end delay constrained problems in sensor networks using DMET scheme for optimizing the problem and the approximation ratio of polynomial distribution-based algorithm for NP-hardness issues, which yields to improve energy and delay constrained problems.

### CONTRUBUTION OF WORK

The contribution scheme of algorithms works as follows, It combines shared counter and multistage filters. A tiny table with estimates. ICEB offers law error, replacing the counters in the estimates could potentially improve their space to accuracy ratio. Finally this study extensively achieves, the estimators can return the estimated value in 0(1)time, and maintains the access efficiency of such combined schemes.

## IV. PROPOSED RELATED WORK

While all counter arrays are required to monitor traffic at line speed, their implementations differ in the availability of the monitored data. Specifically, an online counter array can be updated and read at line speed while an offline counter array can only be updated at line speed. Naturally, offline counter arrays are used for high level tasks such as data analysis and identifying performance bottlenecks. On the other hand, online counter arrays are used to answer low level queries such as what priority to give a certain flow, how much bandwidth it requires and where to route its packets. Hybrid DRAM/SRAM counter arrays [1], [2] store only the least significant bits of each counter in SRAM and the rest of the counter in (slower) DRAM. In CounterBraids [15], counters are compressed in order to fit inside SRAM, but the decoding process is slow. Alternatively, Randomized Counter Sharing (RCS) [16] reduces the overhead required to maintain a flow to counter association. In that solution, each flow is randomly associated

**Proposed Algorithms**

In ICE-Buckets we partition the counter array into small independent buckets and each bucket utilizes a different estimation function according to its maximal counter. Intuitively, in ICE-Buckets a large counter only increases the estimation error for its own bucket rather than to every other counter. This architecture is illustrated in Figure 3.To make the discussion more concrete, we first intro- duce additional notations specific to this section in Table II. ICE-Buckets employs a base error parameter that is called $s_{step}$. As mentioned, symbols are separated into buckets. Each bucket maintains a scale parameter $w_i$ of size $\log_2 E$ bits, where $E$ is a parameter of the data structure. To estimate counters in a bucket with scale $w$, we use the estimation function $A_{s_{step}} \cdot w$ .

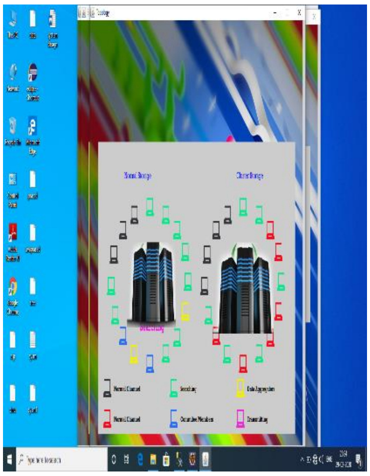
**Dynamic Configuration**

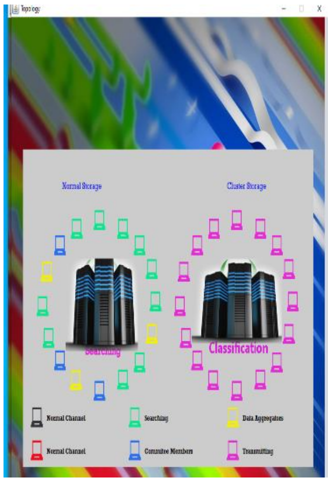
In Theorem 7 below we show that there is an ICE-Buckets configuration for which the error is low. However, that low error configuration is unknown. In this section, we present an algorithm that dynamically configures ICE-Buckets to achieve a low error for any workload. This process is composed of local and global upscale operations.

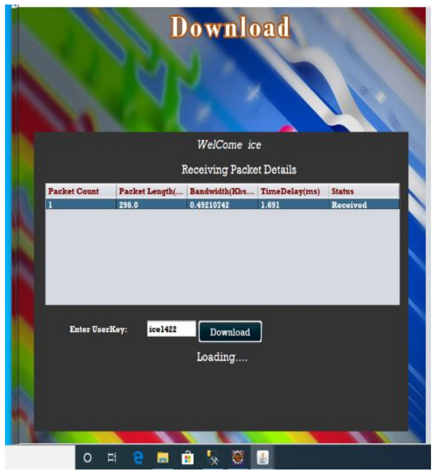
**Algorithm 1 Global Upscale**

Upscale is a way to dynamically adjust the counter scale to the actual workload [30]. It is useful in case the counting capacity M is unknown. We begin with a small that gives a counting capacity of A(L−1) and dynamically increase it when necessary. That is, when a symbol approaches L−1, we increase to >. Then, we update all symbols to maintain unbiased estimation under the new scale.

## V. COMPARATIVE RESULTS








## VI. CONCLUSIONS

A novel counter estimation data structure that minimizes the relative error. ICE-Buckets uses the optimal estimation function with a scale that is optimized independently for each bucket.

## VII. ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their comments that helped improve the presentation of this work.

# REFERENCES


[1] D. Shah, S. Iyer, B. Prahhakar, and N. McKeown, “Maintaining statistics counters in router line cards,” *IEEE Micro*, vol. 22, no. 1, pp. 76–81, Jan. 2002.

[2] S. Ramabhadran and G. Varghese, “Efficient implementation of a statistics counter architecture,” *ACM SIGMETRICS*, vol. 31, no. 1, pp. 261–271, 2003.

[3] A. Kabbani, M. Alizadeh, M. Yasuda, R. Pan, and B. Prabhakar, “AF-QCN: Approximate fairness with quantized congestion notification for multi-tenanted data centers,” in *Proc. 18th IEEE Symp. High Perform. Interconnects (HOTI)*, Aug. 2010, pp. 58–65.

[4] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, “Network intrusion detection,” *IEEE Netw.*, vol. 8, no. 3, pp. 26–41, May 1994.

[5] D.J.Samatha Naidu, Dr.P.Chitti Babu,Dr.C.Shobha Bindu, Dr.A.Subramanyam **published a research paper on**” A Dual Covered Broadcast Technique to avoid piggybacking Neighbor problems in Mobile Adhoc Networks” **“** International Journal of Communications And Engineering (IJCAE) issue 2012.

[6] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges,” *Comput. Secur.*, vol. 28, nos. 1–2, pp. 18–28, 2009.

[7] G. Dittmann and A. Herkersdorf, “Network processor load balancing for high-speed links,” in *Proc. Int. Symp. Perform. Eval. Comput. Telecommun. Syst.*, 2002, pp. 1–9.

[8] G. Einziger, B. Manes, and R. Friedman, “TinyLFU: A highly efficient cache admission policy,” in *Proc. Euromicro PDP*, 2014, pp. 146–153.

[9] C. Estan and G. Varghese, “New directions in traffic measurement and accounting,” in *Proc. ACM Conf. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*, 2002, pp. 323–336.

[10] G. Cormode and S. Muthukrishnan, “An improved data stream summary: The count-min sketch and its applications,” *J. Algorithms*, vol. 55, no. 1, pp. 58–75, 2004.

[11] A. Ciuffoletti and M. Polychronakis, “Architecture of a network monitoring element,” in *Proc. Eur. Conf. Parallel Process.*, 2006, pp. 5–14.

[12] V. Paxson, J. Mahdavi, A. Adams, and M. Mathis, "An architecture for large scale Internet measurement," *IEEE Commun. Mag.*, vol. 36, no. 8, pp. 48–54, Aug. 1998.

[13] C. Estan, K. Keys, D. Moore, and G. Varghese, "Building a better NetFlow," in *Proc. ACM SIGCOMM*, 2004, pp. 245–256.

[14] Y. Lee and Y. Lee, "Toward scalable Internet traffic measurement and analysis with Hadoop," *ACM SIGCOMM*, vol. 43, no. 1, pp. 5–13, Jan. 2012.

[15] H. Wang, H. Zhao, B. Lin, and J. Xu, "DRAM-based statistics counter array architecture with performance guarantee," *IEEE/ACM Trans. Netw.*, vol. 20, no. 4, pp. 1040–1053, Aug. 2012.

[16] D.J.Samatha Naidu, Dr.P.Chitti Babu,Dr.C.Shobha Bindu, Dr.A.Subramanyam **published a research paper on" A Dual Covered Broadcast Technique to avoid piggybacking Neighbor problems in Mobile Adhoc Networks" ,** In International Journal Publications of problems and Applications in engineering research paper.Vol 4, Special Issue 01,2013, Issn:2230-8547, e-ISSN:2230-8555.

[17] Y. Lu, A. Montanari, B. Prabhakar, S. Dharmapurikar, and A. Kabbani, "Counter braids: A novel counter architecture for per-flow measurement," in *Proc. ACM SIGMETRICS*, 2008, pp. 121–132.

[18] T. Li, S. Chen, and Y. Ling, "Per-flow traffic measurement through randomized counter sharing," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1622–1634, Oct. 2012.

[19] M. Chen and S. Chen, "Counter tree: A scalable counter architecture for per-flow traffic measurement," in *Proc. IEEE Int. Conf. Netw. Protocols (ICNP)*, 2015, pp. 111–122.

[20] N. Hua, B. Lin, J. J. Xu, and H. C. Zhao, "BRICK: A novel exact active statistics counter architecture," in *Proc. ACM/IEEE ANCS*, 2008, pp. 89–98.

[21] B.-Y. Choi, J. Park, and Z.-L. Zhang, "Adaptive random sampling for load change detection," in *Proc. ACM SIGMETRICS*, 2002, pp. 272–273.

[22] D.J.Samatha Naidu, Dr.P.Chitti Babu **published a research paper on" A Comparative Study on Different Object-oriented Metrics"** International Journal of engineering sciences research -IJSER VOL 4, 2013 ISSN:2230-8504,E-ISSN:2230-8512) (pgno:1017-1021) **impact factor 2.56** citated and indexed with Google scholar, Monterey bay library DoAJ,

[23] Y. Zhang, S. Singh, S. Sen, N. Duffield, and C. Lund, "Online identification of hierarchical heavy hitters: Algorithms, evaluation, and applications," in *Proc. ACM IMC*, 2004, pp. 101–114.

[24] A. Metwally, D. Agrawal, and A. EI Abbadi, "Efficient computation of frequent and top-k elements in data streams," in *Proc. 10th Int. Conf. Database Theory (ICDT)*, 2005, pp. 398–412.

[25] R. M. Karp, S. Shenker, and C. H. Papadimitriou, "A simple algorithm for finding frequent elements in streams and bags," *ACM Trans. Database Syst.*, vol. 28, no. 1, pp. 51–55, Mar. 2003.

[26] R. Ben-Basat, G. Einziger, R. Friedman, and Y. Kassner, "Heavy hitters in streams and sliding windows," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.

[27] D.J.Samatha Naidu, Dr.P.Chitti Babu **published a research paper on " Prevention Of Data Leakage System with Mining Techniques"** International Journal of Communications and Engineering February, 2014 ISSN No: 0988 – 0341E Pgno 1 to 4 . Indexing: EI (INSPEC, IET), Google Scholar, Crossref, Engineering & Technology Digital Library, ProQuest, and Electronic Journals Library

[28] D. Ficara, A. Di Pietro, S. Giordano, G. Procissi, and F. Vitucci, "Enhancing counting bloom filters through Huffman-coded multilayer structures," *IEEE/ACM Trans. Netw.*, vol. 18, no. 6, pp. 1977–1987, Dec. 2010.

[29] S. Cohen and Y. Matias, "Spectral bloom filters," in *Proc. ACM SIGMOD*, 2003, pp. 241–252.

[30] G. Einziger and R. Friedman, "Counting with TinyTable: Every bit counts!" in *Proc. Int. Conf. Distrib. Comput. Netw. (ICDCN)*, 2016, p. 27.

[31] R. Morris, "Counting large numbers of events in small registers," *Commun. ACM*, vol. 21, no. 10, pp. 840–842, 1978.

[32] D.J.Samatha Naidu, Dr.P.Chitti Babu , **Dr.C.Shobha Bindu, Dr.A.Subramanyam published a research paper on "a solution for fast ip network failures using mrc recovery scheme"** International Journal of Computer Applications and engineering, IJCAE, Vol 3, Issue 3, November 2012, pgno 101-104 ISSN no:0988-0382E

[33] R. Stanojevic, "Small active counters," in *Proc. IEEE INFOCOM*, May 2007, pp. 2153–2161.

[34] C. Hu *et al.*, "Discount counting for fast flow statistics on flow size and flow volume," *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 970–981, Jun. 2014.

[35] C. Hu and B. Liu, "Self-tuning the parameter of adaptive non-linear sampling method for flow statistics," in *Proc. Int. Conf. Comput. Sci. Eng. (CSE)*, Aug. 2009, pp. 16–21.

[36] E. Tsidon, I. Hanniel, and I. Keslassy, "Estimators also need shared values to grow together," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1889–1897.

[37] L. Yang *et al.*, “Case: Cache-assisted stretchable estimator for high speed per-flow measurement,” in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.

[38] C. Hu *et al.*, “Accurate and efficient traffic monitoring using adaptive non-linear sampling method,” in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 26–30.

[39] Wand Network Research Group. *Waikato Internet Traffic Storage 2009-01-06 0:00-23:30 UTC, ISPDSL-I*. Accessed: Apr. 17, 2018. [Online]. Available: http://wand.net.nz/wits/ispdsl/1/

[40] P. Hick. *The CAIDA UCSD Anonymized Internet Traces-2015-02-19. 13:00 UTC, Direction A.* Accessed: Apr. 17, 2018. [Online]. Available: https://www.caida.org/data/

[41] P. Hick. *CAIDA Anonymized 2013 Internet Trace, Equinix-Sanjose 2013-1-17. 13:55 UTC, Direction B.* Accessed: Apr. 17, 2018 [Online].Available: https://www.caida.org/data/passive/passive_2013_dataset.xml