Unique TCP Approach wrt Fountain as Asynchronous and Modulated Data in Secured Transmission with CMM

V.V Bhavani
Department of Computer Science and Technology
V R Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India

S.Kranthi
Department of Computer Science and Technology
V R Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India

K.Pranathi
Department of Computer Science and Technology
V R Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India

Abstract- Always transmitting of data in TCP asynchronous mode from single source to multiple destinations is big challenge in regular topologies. So to achieve the feasible transmission there are some models. So one of the models is the fountain one. Basic fountain model is weak in asynchronous mode but our work is to achieve and overcome this as well as security. The data will be partitioned in multiple and valid clusters to inject into fountains with the help of central repository. So for this we use modulation based fountain transmission model to send the data to destinations. Here our work is based on MST(Modulation based secured transmission). The entire work is under one unique frame work called CMM(close monitoring and logging) which works with mirror based technology. This approach is entirely switches accordingly to legacy and adoptive networks in asynchronous fountain models. And our work is extended parallelly by using SET(swift encoding technique) for security purpose.

Keywords – Fountain ,topology , MST ,CMM, SET, asynchronous, mirror

I. INTRODUCTION

The number of parallel data transmission channels from a single data source with TCP will fall under one unique framework which is our current work called **CMM**(close monitoring and logging). The central repository server monitor which works as root framework and acts as central controlled repository and maintains micro log information for each and every process of transmission.This log helps our work to overcome of loss of duplicate transmission for future process of transmissions. This technique called as mirror mechanism.And also uses to assign modulation for each and every channel, all channels for one single transmission process belongs to one fountain. All fountains logged by **CMM** with respect to modulation with respect to routing technique. All the routers will be framed by CMM instantly and will be assigned unique modulation depends on the transmission capacity for that particular tunnel. The tunnel(s) will be totally under one fountain for one transmission process.

Once the data selected by source to be broadcasted CMM will track the available routers and assigns modulation numbers based on transmission capacity of each and every router. Here modulation means , assigning of network packet transmission capacity per router based on number of fountain tunnels. The tunnel will have clustered modulation threshold. All tunnels will fall under one fountain , and flushes data to destination by taking available routers for feasible transmission. The Logger will be in CMM central repository which will be updated per process with mirror technique. The mirror log works with mainly for duplicate data transmission per process. Here log will be tracked by CMM if any data to be broadcasted repeatedly in other process of transmission. If this is the case of repetition or duplication transmission, mirror methodology will give proper information to CMM for avoiding retransmission of the data which leads to less loss of data.

The main theme is to use TCP to transmit the data in fountain approach. Normally data will be transmitted in single flush or multiple flushed depends on the topology model. If the topology is based on multiple routers to transmit the

data, we use fountain technique for broad casting. But basically fountain technique is too sensitive in transmission; reason is loss of data and less security. And also here biggest challenge is to control the router to choose correct destination to broad cast. Our work is totally depends on modulated and secured transmission approach called **MST**(Modulation based secured transmission). This is totally depends on security, modulation, monitoring and log maintenance. So here we use data rotational based algorithm called **SET**(swift encoding technique) to encode and decode the data. This is totally depends on the EASCII regeneration depends on the data to be transmitted. This contains 2 models of encoding, one for normal ASCII and second for EASCII.

The central repository or master server is the key for transmission in our entire frame work. This server will act as a service based on service oriented architecture which serves ever running services to all networks. Once the data is selected from source central repository will take an initiative to frame feasible routers to frame secured fountains for broad cast. So once the fountains are ready the active routers among available routers will be active and collects the clustered data which is scheduled by again central repository. But before the data acquired by routers the data will be encoded and packets address will be maintained in LOG. Once the data is ready to be transmitted the asynchronous fountain transmission is activated to transmit the data. Here the acknowledgement for each transmission will be updated in LOG though alert is received by actual source. This technique will be useful to recollect the best transmission if the same paths of routers with fountain for central repository to establish the same previous approach if that was fair transmission. This controls the time to reallocate the selection of routers and fountains. This is totally depends on selective modulation based on LOG.

The SET mechanism is totally depends on the model of the data to be transmitted. Some data which is sensitive based on the EASCII and strange result in the form of presence at the destination side. So the central repository is having a parallel data analization frame work to track the data. Once the data is chooses this framework scans if any EASCII this will frame high shifting of the bits. And with normal data low shifting frame work will be active. This switch mechanism which is depends on data but selected by central repository. All time central server overcomes the loss of transmission percentage by comparing with the previous log of duplicated or new data with mirror mechanism. The main mirror work comes with duplicated data which was transmitted earlier and if any loss of data with that and if the same data to be fountained again our log helps with mirror technique to transmit with better and less loss of data. The security provided with unique encoding and decoding technique with shifter algorithm which also works with extended ASCII data also. Here SET approach is used for shifting technique. This can used to encode and decode with clustered data and with perfect clubbing of decode at the destination side.

The rest of the paper is organized as follows. Proposed embedding and extraction algorithms are explained in section II. Experimental results are presented in section III. Concluding remarks are given in section IV.

II. PROPOSED ALGORITHM

A. Modulation Allocation algorithm –

This is enhancement to raptor mechanism and modulation allocation done by central repository to each and every available routers. This allocation is based on fountain transmission capacities and router's energy levels.

$$\sum \mathbf{r} \approx \mathbf{R(r1 , r2 , r3 \ldots. Rn)} \text{ //available routers per fountain.}$$

$$\int \mathbf{m} \; \forall \; \mathbf{[m1 \; m2 \; m3 \ldots\ldots \; mn]} \text{ // modulation vector}$$

$$\mathbf{r(m1 \leq x \leq mn) = \int f(x)dx} \text{ // x is the number of router.}$$

Once the modulations allocated for routers in the fountains the selected packets buffer will be tracked by central repository. And packets in the form of data will be allocated to preceding best possible routers per fountain based on the modulation levels.

The total data(D) selected by source peer and the allocation of data to routers in the fountain is as follows.

$$\mathbf{n = SIZE(D)}$$

$$\mathbf{r(\psi) = D(n) \sim padding(n)}$$

Once the routers are assigned with modulations the data will be broadcasted to destinations.

Mirror: This is unique work based on log mechanism from central server. If any of the data is already been broadcast then normally that process will be saved in the log about the data and P2P . If the same process has to be repeated then log will traced by CMM and the data will be duplicated from central servers to end peer, but not from the available fountains. This reduces consumption time, cost of network and loss of packets.

B{t1 , t2 , t3 , t4……tn}if the broad cast vector which is logged at central repository and following are the modulated routers which are used as best transmission among the peer to peer.

Available routers with modulations assigned by central repository.

M{m1 , m2 , m3 , m4 … mn} with available routers ***R{r1 , r2 , r3, r4……rn}***in the network**.**

t1 [r3(m3 | m5) -> r4(m4 | mn)],***t2[r2(m1 ~ m3) -> r4(m1+m2)] [r3(m3 | m5) -> r4(m4 | mn)]*** …..**tn[r4(m4-m3) -> r5(m2 & m4)].** In these communications recent broadcasting to be done is similar to t1. So the central repository tracks the log and stops the process and checks the packets loss and transmits the lost packet to destination. Again this will be logged and comes tracker umbrella.

A1.txt	R1:R2(56)	16	5
A2.txt	R1:R2(56)	16	5

In the above picture the 2 text files with energies 16 and path number 5 with r1 and r2 are the part of the fountain and this is duplication and log will be updated with mirror.
So retransmission will be stopped by central repository.

B. Star Shifter encryption and decryption algorithm –

In star shifter method data files will be encrypted in the form of cipher encoded text. When broad casting the packets from server to server or P2P the source file data will be encrypted before framing the packets. All charecters will be formed in the form of ASCII values. The values are internally turned into binary bits. Left shift and right shift operations will be made on binary data. If the ASCII byte is even our approach will shift operation and right shift if the ASCII byte is odd. Totally now it generates In star shifter method data files will be encrypted in the form of cipher encoded text. When broad casting the packets from server to server or P2P the source file data will be encrypted before framing the packets. All charecters will be formed in the form of ASCII values. The values are internally turned into binary bits. Left shift and right shift operations will be made on binary data. If the ASCII byte is even our approach will shift operation and right shift if the ASCII byte is odd. Totally now it generates ciper text and if the operation is vice versa the normal text will be made.

ENCRYPTION:
Input : Normal Text
Output: Ciper encoded text
Intialization:

n←0 //*total number of characters in input string.*

∑D←0 //*total data*

Cipher←1

∑E←0 //*total Encrypted data*

t1←null //*temp variable*

t2←null //*temp variable*

Loop for each c in D

n=0

t1←LSHIFT(c,n,CIPHER)

t2←RSHIFT(c,n+1,CIPHER)

n+1

E← APPEND(t1)

E←APPEND(t2-1)

end loop

Decryption:
Input: Cipher text

Output: Plain text

Initialization:

N←0 // initialization for total number of characters.

∑E←0 //Cipher texts

∑N←0 // plain text

T1←null

T2←null

Loop for each c in E

n=0

t1←RSHIFT(c,n,CIPHER)

t2←LSHIFT(c,n+1,CIPHER)

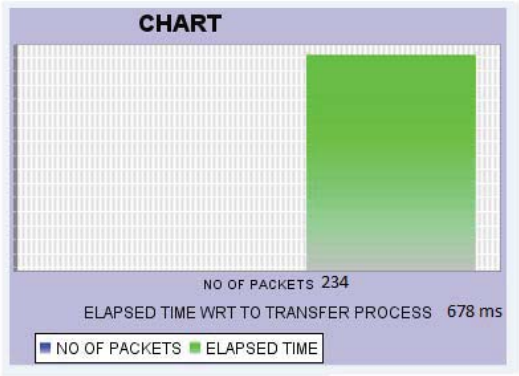
n+1

D← APPEND(t1)

D←APPEND(t2+1)

end loop

III. Experiment and Result



(a)

IV.CONCLUSION

This work is having the scope to extend to multi servers adoptions ie existing work can be combined to legacy and also furious types of network topologies. And fountains can be asynchronous with other topologies and can be extended as TTL(time to live) mechanism. This methodology can work on travelling gravity mechanism. Time of fountain packet travelling can modified in the middle of transmission where buffer of fountain with less energy is low. So then gravity will be made low till the buffer occupies its space of packets to be transmitted without loosing packets.

REFERENCES

[1] M. Becke, T. Dreibholz, H. Adhari, and E. Rathgeb, “On the fairness of transport protocols in a multi-path environment,” in Proc. IEEE ICC, Ottawa, ON, Canada, 2012, pp. 2666–2672.

[2] Y. Cui, X. Ma, H. Wang, I. Stojmenovic, and J. Liu, “A survey of energy efficient wireless transmission and modeling in mobile cloud computing,”Mobile Netw. Appl., vol. 18, no. 1, pp. 148–155, 2013.

[3] C. Lai, K. Leung, and V. Li, “Enhancing wireless TCP: a serialized timer approach,” inProc. IEEE INFOCOM, 2010, pp. 1–5.

[4] S. Mascolo, C. Casetti, M. Gerla, M. Sanadidi, and R. Wang, “TCP westwood: Bandwidth estimation forenhanced transport over wireless links,” inProc. 7th Annu. ACM MobiCom, 2001, pp. 287–297.

[5] Y. Huang, M. Ghaderi, D. Towsley, and W. Gong, “TCP performance in coded wireless mesh networks,” in Proc. 5th Annu. IEEE SECON, 2008, pp. 179–187.

[6] Y. Li, Y. Zhang, L. Qiu, and S. Lam, "Smarttunnel: Achieving reliability in the Internet," in Proc. 26th IEEE INFOCOM, 2007, pp. 830–838.
[7] J. Sundararajan, D. Shah, M. Médard, M. Mitzenmacher, and J. Barros, "Network coding meets TCP," inProc. IEEE INFOCOM, 2009, pp. 280–288.
[8] A. Ford, C. Raiciu, M. Handley, S.Barré, and J. Iyengar, "Architectural guidelines for multipath TCP development,," RFC 6182, 2011.
[9] C. Pluntke, L. Eggert, and N. Kiukkonen, "Saving mobile device energy with multipath TCP," inProc. 6th ACM MobiArch, 2011, pp. 1–6. [28] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath TCP,"Comput. Commun. Rev., vol. 41, no. 4, pp. 266–277, 2011.
[10] C.Raiciu,D.Niculescu,M.Bagnulo,andM.J.Handley,"Opportunistic mobility with multipath TCP," in Proc. 6th ACM MobiArch, 2011, pp. 7–12.
[11] J. Iyengar, P. Amer, and R. Stewart, "Performance implications of a bounded receive buffer in concurrent multipath transfer,"Comput.Commun., vol. 30, no. 4, pp. 818–829, 2007[12] Y. Cui, X. Wang, H. Wang, G. Pan, and Y. Wang, "FMTCP: A fountain code-based multipath transmission control protocol," inProc. 32nd ICDCS, 2012, pp. 366–375.
[12] M. Luby, "LT codes," in Proc. 43rd Annu. IEEE FOCS, 2002, pp. 271–280.
[13] A. Shokrollahi, "Raptor codes,"IEEE Trans. Inf. Theory, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.[15] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacityapproaching irregular low-density parity-check codes,"IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 619–637, Feb. 2001.
[14] T. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes,"IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 638–656, Feb. 2001.
[15] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer, "Raptor forward error correction scheme for object delivery," RFC5053, 2007 [Online]. Available: http://tools.ietf.org/html/rfc505