International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 22 (2017) pp. 12701-12706

Integration Scheme of Network Coding and Address Bit Vector in Wireless Network to Acquire More Reliability

[1]Mayank Kumar Goyal, [2]Satya Prakash Ghrera and [3]Jai Prakash Gupta

[1]Research Scholar, Department of Computer Science and Engineering Information Technology, Jaypee University of Information Technology, Waknaghat, Distt. Solan, Himachal Pradesh, India.

[2]Professor, Department of Computer Science and Engineering Information Technology, Jaypee University of Information Technology, Waknaghat, Distt. Solan, Himachal Pradesh, India.

[3]Professor, Department of Computer Science and Engineering, Bennett University, Greater Noida, Uttar Pradesh, India.
Orcid: *Orcid Id: 0000-0001-8966-496X*

Abstract

Network Coding (NC) is affirmed to be power and transmission capacity proficient procedure, in light of the less number of transmitted packets over the system. The proposed situation in this paper applies the upsides of NC over wireless network to acquire more reliability even in case of increase in loss of un-coded packets. Network coding uses the XOR operator to code overheard packets. Our algorithm uses bit addresses to where every node is distinguished by bit value '1' in an address bit vector. Distinguishing packets and processing the network coding of packets can be effectively done utilizing address bit vectors. This algorithm also acknowledges redundancy with total no. of coded packets sent with respect to actual no. of nodes present in the system.

Keywords: Network Coding; Redundancy; Relay Nodes; Wireless Networks

INTRODUCTION

However, one method for guaranteeing unwavering quality is to infuse repetition. Repetition can be expert by retransmissions. However, fundamental retransmissions are excessively costly as far as latency and power utilization. In this way, a trade off must be taken with a specific end goal to guarantee high unwavering quality mulling over restricted power utilization.

Network coding (NC) [1] is a strategy which is used to join a set of packets together into a solitary packet. Packet Combination is typically carried out in 2 routes: (i) By playing out a straight mix of the packets (ii) By playing out XOR operation. Advantages for NC can be in unwavering quality, storage & security. Enhanced unwavering quality can be unmistakably shown in Figure 1 which represents NC (network coding) utilizing XOR operator. S_2 & S_1 are 2 sources & 4 collectors D_4, D_3, D_2 and D_1. By noticing packet P_2 as packet generated by S_2 & P_1 created by S_1 & recipients need to get all packets, a middle hub, R can XOR the two packets to make network- coded packet & send it to collectors which are able to remove packet P_1 having P2 & skilled to extricate packet P_2 having packet P_1.

Network coding implementation is difficult due to its complex nature and the trouble of mixing the packets with existing framework. Including a consistent combination of packets, for example, XOR requires each hub in the system to have the ability to create such mixtures. Late work recommends that Network coding turns out to be excessively complex when managing Gaussian elimination[2].

Network coding can be ordered into 3 primary classes relying upon the sort of combinations; which are: Inter-session NC Intra- session NC, & Joint Inter & Intra Session NC [3].

Inter Session NC can decrease the no. of transfers when the packets from distinctive source are network- coded. We code the packets from a similar source; we are doing intra-session NC.

Reliability can likewise be accomplished utilizing linear combinations rather than the XOR operation; nonetheless it is all the more expensive as far as intricacy. For the most part sensor hubs have restricted computational capabilities which ruin them from performing complicated arithmetic operations.

A Network coding scheme called Network coding codes with origin information to remove redundancy of combination is introduced in this paper. Ordinarily, in these applications, the sink hub starts round of gathering where sensor hubs communicate sensed/processed information towards the sink node. In systems where the sink is found far from the sensor hubs and with dynamic paths, a gathering based convention, for example, CTP [2] can be productively used to makes the courses. The principle reason with utilizing network coding for accomplishing unwavering quality is that coding countless will prompt an expansive enough overhead that defeats its points of interest.

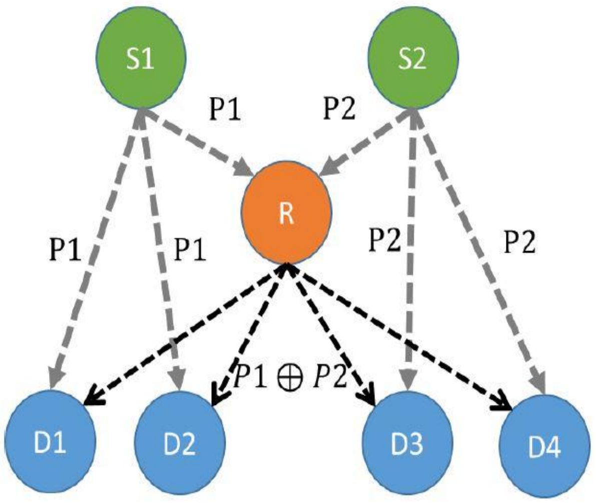

Figure 1: Network Coding using XOR

LITERATURE REVIEW

The significance of packets combination using network coding was a fundamental subject for a large portion of the specialists in the previous couple of years. The principle center of system configuration has been founded on effortlessness of operation & versatility. However, late works demonstrates additionally that viable power sparing and physical asset administration should be possible by communicating with the different schemes of NC. Hui and Culler[4] presented Deluge a dependable information spread convention to beat the propagation of huge information objects that can hear it.

However an extensive new convention was presented in [5] where 40% less packets were utilized as a part of vast systems than Deluge and load adjusting was additionally mulled over despite the fact that it was not obviously checked in the algorithm.

AdapCode used to disperse information in the system utilizing Network coding [6] yet in a versatile route as indicated by link quality. To concentrate on information accumulation algorithms, Wang et al. [7] presented another scheme in NC called PNC (Partial Network Coding) attempting to conquer the test of consistent information accumulation from sensors to server where information expulsion was accessible.

SenseCode [2] was presented as an accumulation convention in wireless networks utilizing NC as many two one correspondence convention in which the time was separated into rounds.

SYSTEM MODEL

Our network coded algorithm is a NC convention which accomplishes reliability for information gathering application for remote systems. We accept a two-level system with static courses where every hub conveys its packet to a relay node. Thus, this transfer will convey got packets to the sink. We accept systems where transfers, as a rule, are sufficiently far that they don't catch each other. This will include complexity while accomplishing reliability since caught packets by transfers are major in recouping packets.. In our algorithm NC is performed utilizing the XOR operation. Just, when a data packet is received, the hubs will XOR the substance with at least one packet which is accessible in its cradle. Then, resulting packet is sent to the next hop.

With a specific end goal to recognize the packets which were XORed, address of every packet is incorporated into the header of the coded message. Be that as it may, this represents a vast overhead on the span of the coded packets which at times can surpass the measure of the payload. To conquer this issue, our algorithm uses bit addresses where it alludes to the identity of the actual originating packets.

For an illustration as appeared in Fig.2, consider that relay node got packet P_5 and packet P_2 and needs to network-code them together. Accept for effortlessness, there are just eight hubs in system with a bit vector of size as appeared in Fig.2.

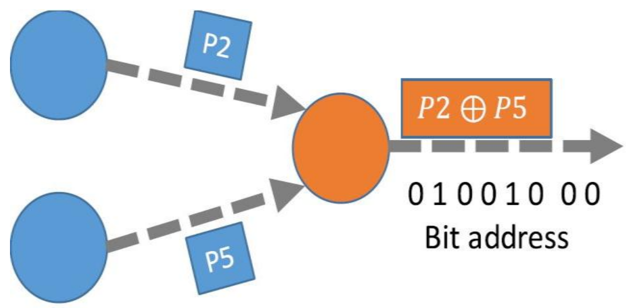

Figure 2: Binary coding of addresses

When Network coding is performed using XOR operation, Set the relating bits that allude to the address of the originating packets to bit value as '1'. For example shown in figure 2, bit 5 & 2 will be set to show that the coded packets contains data from hub 5 & hub 2. a bigger piece vector will be utilized in bigger systems.

ALGORITHM DESIGN

Principle thought behind our algorithm is to infuse enough excess to accomplish unwavering quality without exhausting the power hold of the remote hubs. Our algorithm design indicates 3 actions for hubs, each with an different assignment. In Figure 3, sensor hubs at the primary level will send recorded information and can code packets. Relay nodes are in charge of decoding, coding & sending packets. Sink hub is in charge of uncoding packets.

Gathering begins when the sink starts a round for information accumulation. At the point when sensor hubs get the demand from the sink, hubs will start information exchange by sending the uncoded parcels, for instance, Node 2 will send bundle P_2 to the relay hub. Then, it sits tight for a pre-decided time to catch neighboring parcels. For every new uncoded bundle got, the hubs will XOR its substance with its content with its list of uncoded parcels each one in turn. Just un-coded packets are coded, as demonstrated in figure 3 where sensor hubs don't action decoding. For instance, if hub 2 catches parcels P1 & P3, it will along these lines send, to the relay hub, P2 ⊕ P1 and afterward P3 ⊕ P1 ⊕ P2. Additionally, for every parcel, this will set the corresponding bit addresses in the headers. Normally, in the primary packet the hub will set bit location 2 and 1, and in the second bundle it will set piece area 3, 2, & 1.

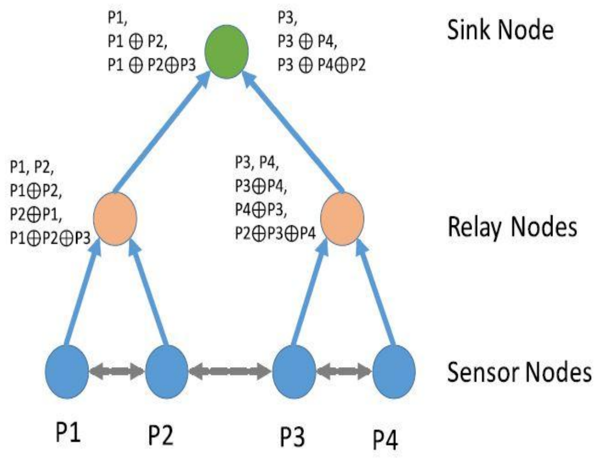

Figure 3: Wireless network

Algorithm1: Network Coding at Sensor Nodes

a) if !Network_Coding() then

b) Hold_up for sink packet()

c) else

d) Begin Network_Coding()

e) Timer.begin()

f) Bit_Address = 1<<a

g) P_a.Network_coded=0

h) send_Packet(P_a)

i) end if

j) if Hear_Packet(P_b) then

k) if Heard_Packet_Is_Uncoded(P_b) then

l) P_a = P_a XOR P_b

m) P_a.Coded=1

n) Address = 1<<a | 1<<b

o) Forward(P_a)

p) end if

q) if Timer.fired() then

r) end Timer()

s) end if

After gathering of packets (single or combined) at the relay hubs, parcels are ordered whether they are uncoded or coded. The relay hub will interpret these bundles in planning for sending as appeared in Algorithm 2. The point of the interpreting calculation is to achieve all parcels in their un-coded unique format. Decoding calculation functions as follows: The initial step of translating begins with un-coded bundles. It will emphasize over all un-coded bundles each one in turn. The calculation checks if the un-coded parcels exist in coded bundles. It is finished by a checking if the bit address at the area of the address of the un-coded bundle is set to 1.

Algorithm 2: Decoding at Relay Nodes

a) Hold_up for Start();

b) if Heard_Packet(P_a) then

c) if P_a.uncoded == 1 then

d) Uncoded_Buffer.add(P_a)

e) else

f) Coded_Buffer.add(P_a)

g) end if

h) end if

i) while(All_Packets_Uncoded() | m<n)

j) for a=1:size.Uncoded_Buffer

k) for b=1:size.Coded_Buffer

l) P = Codded_Bufferb)>

m) if Cardinality(P_a) >1 then

n) if Uncoded_Buffer(a).exists then

o) P = P XOR Uncoded_Buffer(a)

p) P.Address = P.Address XOR Uncoded_Buffer(a). Address

q) if Sum_up(P) ==1 then

r) Uncoded_Buffer.add(P)

s) end if

t) end if

u) end if

v) end for

w) end for

x) end while

Sum_up() test is finished by summing up the bits in the bit address vector. If outcome is 1, at that point we have achieved the first uncoded bundle. This algorithm finds these packets by summing up the bits in the bit address vector. If a message is still un-decodable now, it will be put in a different cradle anticipating further packets if any is gotten a short time later. When task of decoding completes, the relay will get ready packets to be sent. To show, consider a relay with 4 un-coded parcels P_1 to P_4. Relay node will send one un-coded parcel and three coded bundles as below.

First: P_1

Second: P_1 XOR P_2

Third: P_1 XOR P_2 XOR P_3

Fourth: P_4 XOR P_3 XOR P_2 XOR P_1

At last the sink will get the parcels from all transfers and will play out the interpreting calculation depicted before.

RESULT

To confirm the accuracy of the proposed work, we have tested the above said algorithm on three distinct systems. The systems comprise of a solitary sink, 5 relay nodes and we shifted the quantity of sensor hubs from 10 to 20 as indicated Figs. 4-6.

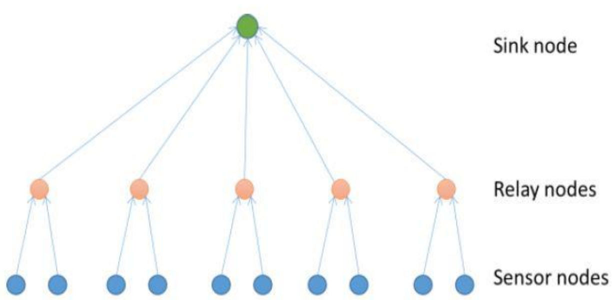

Figure 4: Network with Ten sensor nodes

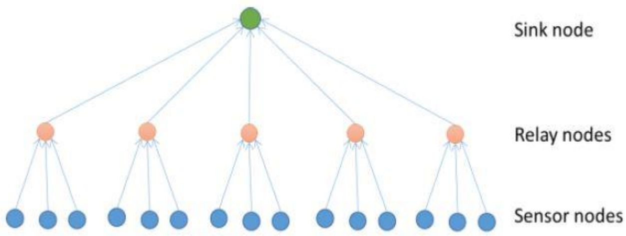

Figure 5: Network with Fifteen sensor nodes

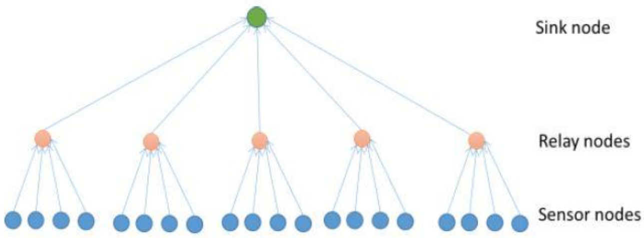

Figure 6: Network with Twenty sensor nodes

Simulation for each of the 3 systems was rehashed ten times. For every emphasis, we varied the quantity of packet loss from zero percent to almost hundred percent of the first parcels sent by sensor hubs. The packet loss was randomized. As appeared in Figs 7-9, our algorithm could recoup most parcels even in unforgiving conditions where the error presented was hundred percent. The achievement proportion was up to 85% percent when the system comprises of ten hubs as appeared in Figure 7.

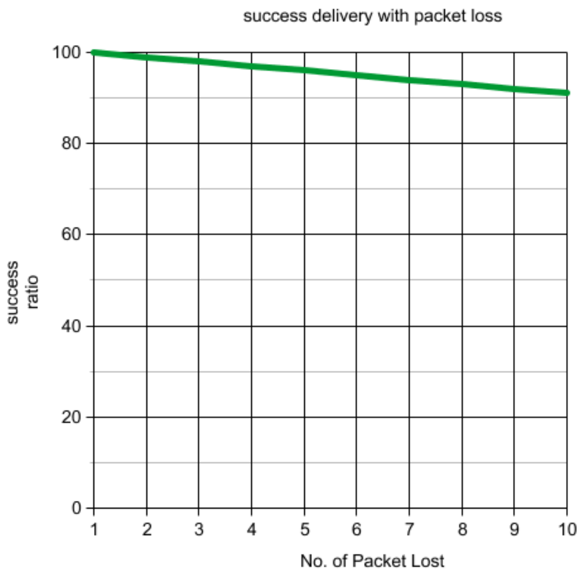

Figure 7: Simulation Results- Network with Ten sensor nodes

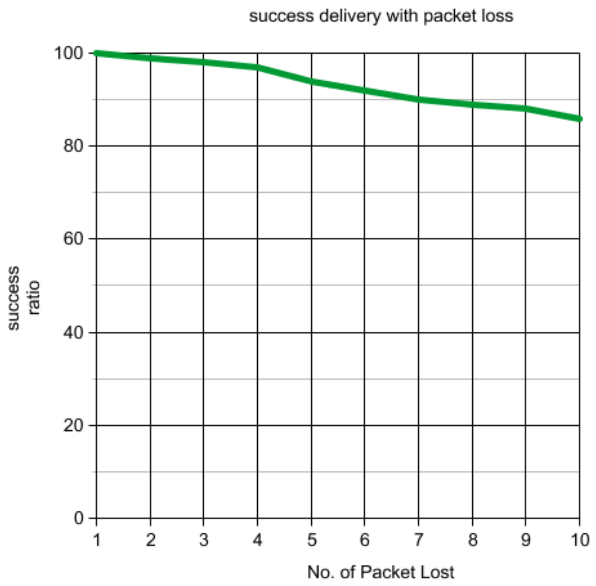

Figure 8: Simulation Results- Network with Fifteen sensor nodes

In Figure 8 and 9, the execution weakens in light of the fact that the quantity of parcels sent is less since the quantity of hubs per relay node increased. In this manner, when the quantity of hubs increment in a similar hop, the quantity of overheard parcels is lessened and thusly the quantity of sent packets. Our algorithm just sends the original un-coded packet and coded packets from nearby neighbors.

To exhibit the impact of including more hubs, we reenacted a system of 8 hubs and 4 transfers. Then, we quantified the quantity of packet sent and the level of redundancy. The redundancy is figured as for original parcels sent. For a case, an excess of two hundred percent implies that the quantity of bundles sent is double the quantity of unique parcels.

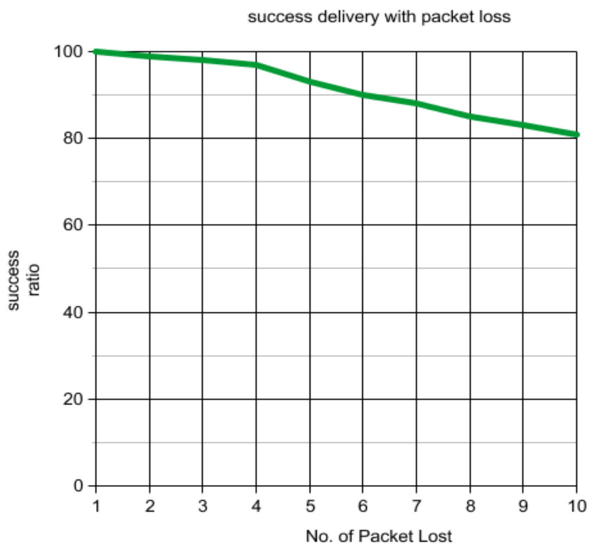

Figure 9: Simulation Results- Network with Twenty sensor nodes

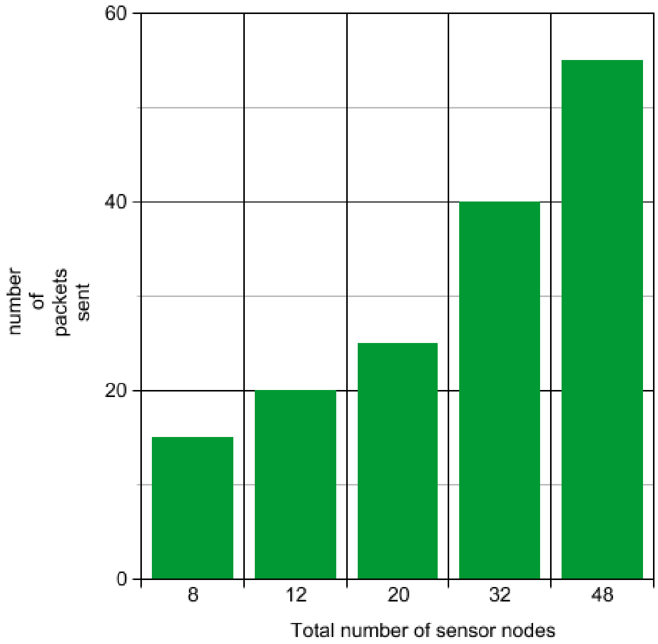

Figure 10: Total no. of packets sent with respect to no. of nodes

Considering Figures 10 and 11, quantity of hubs is set to twenty while keeping network with neighboring hops to one, the repetition is 1.4. If 100% of the bundles are lost; the framework won't have the capacity to recoup all parcels. In this way, in unforgiving conditions, it is accepted that the quantity of hubs ought to be increased.

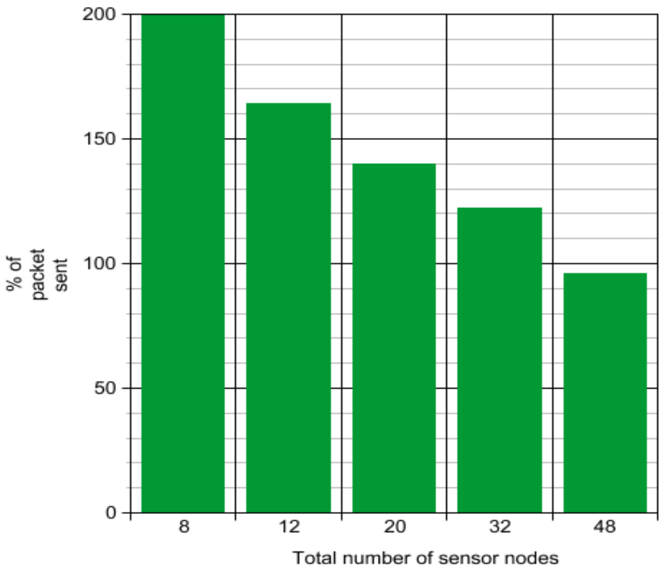

Figure 11: Total no. of packets sent with respect to no. of nodes

CONCLUSION

The algorithm uses the XOR operator to network-code packets. Besides, this utilizes bit addresses to which brings down the measure of bytes sent per packet. Also, utilizing bit addresses to help in recognizing the quantity of packets incorporated into each coded packet and permits to effortlessly distinguishing which packet are incorporated into the coded message. Testbeds were led on three distinct systems where the quantity of sensor nodes is expanded. Proposed algorithm displayed strength even in brutal conditions where packets losses were nearly hundred percent of the aggregate number of unique parcels sent.

REFERENCES

[1] C. Fragouli, J. Y. Boudec, and J. Widmer, "Network Coding: An Instant Primer," ACM SIGCOMM Computer Communication Review, pp. 63-68, Vol. 36, No. 1, Jan. 2006

[2] Keller, Lorenzo, Emre Atsan, Katerina Argyraki, and Christina Fragouli. "SenseCode: Network coding for reliable sensor networks." ACM Transactions on Sensor Networks (TOSN) 9, no. 2 (2013): 25.

[3] Ostovari, Pouya, Jie Wu, and Abdallah Khreishah. "Network coding techniques for wireless and sensor networks." The Art of Wireless Sensor Networks. Springer Berlin Heidelberg, 2014. 129-162

[4] Gnawali, Omprakash, et al. "CTP: An efficient, robust, and reliable collection tree protocol for wireless sensor networks." ACM Transactions on Sensor Networks (TOSN) 10.1 (2013): 16.

[5] Hui, Jonathan W., and David Culler. "The dynamic behavior of a data dissemination protocol for network programming at scale." In Proceedings of the 2nd international conference on Embedded networked sensor systems, pp. 81-94. ACM, 2004..

[6] Hou, I-H., Y-E. Tsai, Tarek F. Abdelzaher, and Indranil Gupta. "Adapcode: Adaptive network coding for code updates in wireless sensor networks." In INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, pp. 1517-1525. IEEE, 2008.

[7] Wang, Dan, Qian Zhang, and Jiangchuan Liu. "Partial network coding: Theory and application for continuous sensor data collection." In Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on, pp. 93-101. IEEE, 2006.