www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 11 November, 2013 Page No. 3174-3179

Tcp performance in coded wireless mesh networks with cope implementation: An overview

Nikumani Choudhury

Department of IT, Gauhati University
Guwahati, India
nikumani2010@rediff.com

Abstract: ***This paper investigates the benefit of network coding for TCP traffic in a wireless mesh network. COPE, a new architecture for wireless mesh networks is also introduced. Packets are mixed (i.e., coded) by routers from different source nodes to increase the information content during every transmission. Intelligently mixing the packets increase network throughput. The results show that COPE largely increases network throughput. Depending on the traffic pattern, congestion level, and transport protocol used, gains in throughput vary. Network coding not only reduces the number of transmissions by sending multiple packets via a single transmission but also results in a smaller loss probability due to reduced contention on the wireless medium. Coding opportunity can be increased by inducing small delays at intermediate nodes. However, this extra delay at intermediate nodes results in longer round-trip-times that adversely affect TCP throughput.***

Keywords: Wireless Mesh Networks, COPE, Network Coding, TCP.

1. COPE Introduction:

COPE, a new forwarding architecture that substantially improves the throughput of wireless networks, has been described in this paper. COPE inserts a coding shim between the IP and MAC layers, which identifies coding opportunities and benefits from them by forwarding multiple packets in a single transmission.

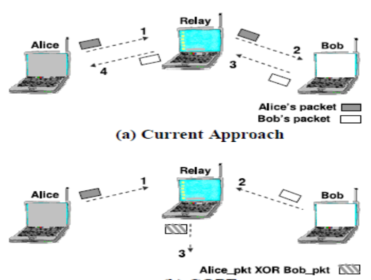

Figure 1: A simple example of how COPE increases the throughput. It allows Alice and Bob to exchange a pair of packets using 3 transmissions instead of 4 (numbers on arrows show the order of transmission).

Let us consider the scenario in Fig. 1, where Alice and Bob want to exchange a pair of packets via a router. In current approaches, Alice sends her packet to the router, which forwards it to Bob, and Bob sends his packet to the router, which forwards it to Alice. This process requires 4 transmissions. Now consider a network coding approach. Alice and Bob send their respective packets to the router, which XORs the two packets and broadcast the XOR-ed version. Alice and Bob can obtain each other's packet by XOR-ing again with their own packet. This process takes 3 transmissions instead of 4. Saved transmissions can be used to send new data, increasing the wireless throughput.

Table 1: Technical terms associated with COPE

Term	Definition
Native packet	A non-encoded packet
Encoded or XOR-ed Packet	A packet that is XOR of multiple native packets
Nexthops of an Encoded packet	The set of next hops for the native packets XOR-ed to generate the native packet
Packet Id	A 32-bit hash of the packet's IP source address and IP sequence number
Output Queue	A FIFO queue at each node, where it keeps the packets it needs to forward
Packet Pool	A buffer where a node stores all packets heard in the past T seconds
Coding Gain	The ratio of the number of transmissions required by the current non-coding approach to the number of transmissions used by COPE to deliver the same set of packets
Coding + MAC Gain	The expected throughput gain with COPE when an 802.11 MAC is used, and all nodes are backlogged

In fact, COPE leads to larger bandwidth savings than are apparent from this example. COPE exploits the shared nature of the wireless medium which, for free, broadcasts each packet in a small neighborhood around its path. Each node stores the overheard packets for a short time. It also tells its neighbors which packets it has heard by annotating the packets it sends. When a node transmits a packet, it uses its knowledge of what its neighbors have heard to perform opportunistic coding; the node XORs multiple packets and transmits them as a single packet if each intended next hop has enough information to decode the encoded packet.

COPE incorporates three main techniques:

(a) Opportunistic Listening: In this mode, the nodes snoop on all communications over the wireless medium and the overheard packets are stored for a time period T (the default is $T = 0.5s$).

Reception reports are broadcasted by each node to tell its neighbors which packets it has stored. Reception reports are sent by annotating the data packets the node transmits. Special control packets are used to send reception reports of those nodes that have no packets to send.

(b) Opportunistic Coding: The dilemma always was what packets to code together to maximize throughput. A node network coding may have multiple options, but it should aim to maximize the number of native packets delivered in a single transmission, while ensuring that each intended nexthop has enough information to decode its native packet. Packets from multiple unicast flows may get encoded together at some intermediate hop. But their paths may diverge at the nexthop, at which point they need to be decoded. If not, unneeded data will be forwarded to areas where there is no interested receiver, wasting much capacity. The coding algorithm should ensure that all nexthops of an encoded packet can decode their corresponding native packets. This can be achieved using the following simple rule: To transmit n packets, $p_1, ..., p_n$, to n nexthops, $r_1, ..., r_n$, a node can XOR the n packets together only if each next-hop r_i has all $n - 1$ packets p_j for $j \neq i$.
This rule ensures that each nexthop can decode the XOR-ed version to extract its native packet. Whenever a node has a chance to transmit a packet, it chooses the largest n that satisfies the above rule to maximize the benefit of coding.

(c) Learning Neighbor State: At times of severe congestion, reception reports may get lost in collisions, while at times of light traffic, they may arrive too late, after the node has already made a suboptimal coding decision. Therefore, a node cannot rely solely on reception reports, and may need to guess whether a neighbor has a particular packet.

To guess intelligently, we leverage the routing computation. Wireless routing protocols compute the delivery probability between every pair of nodes and use it to identify good paths. For e.g., the ETX metric [15] periodically computes the delivery probabilities and assigns each link a weight equal to *1/(delivery probability)*. These weights are broadcast to all nodes in the network and used by a link-state routing protocol to compute shortest paths. We leverage these probabilities for guessing. In the absence of deterministic information, COPE estimates the probability that a particular neighbour has a packet as the delivery probability of the link between the packet's previous hop and the neighbor.

If a node makes an incorrect guess that causes the coded packet to be undecodable at some nexthop, the relevant native packet is retransmitted, potentially encoded with a new set of native packets.

2. Introduction To Network Coding

In traditional networks, data packets are carried by store and- forward mechanisms in which the intermediate nodes (relays or routers) only repeat data packets that they have received. The concept of was introduced for satellite communications in [1] and then fully developed in [2] for general networks. With network coding, a network node is allowed to combine several packets that it has generated or received into one or several outgoing packets. The original paper of Ahlswede *et al.* [2] showed the utility of network coding for multicast in wireline networks. Recently, network coding has been applied to wireless networks and received significant popularity as a means of improving network capacity and coping with unreliable wireless links [3]– [7]. In fact, the unreliability and broadcast nature of wireless links make wireless networks a natural setting for network coding. Moreover, network protocols in wireless networks, *e.g.*, wireless mesh networks and mobile ad hoc networks, are not fully developed yet and hence there is more freedom to apply network coding in such environments compared to wireline networks such as the Internet [5].

The paper aims at characterizing and quantifying throughput improvements in the context of wireless mesh networks (WMNs) employing network coding. It also describes the procedure for implementing network coding for Wireless Mesh Networks.

TCP traffic is bidirectional, *i.e.*, data packets in one direction and ACK packets in the opposite direction, and hence network coding can be applied at intermediate nodes along the path even for a single TCP flow. Unfortunately, due to random delays in networks, coding opportunities at intermediate nodes may be too small to benefit TCP. Inducing a small delay at each intermediate node can increase coding opportunity for TCP traffic, especially when there are only a few TCP flows in the network. However, there is a tradeoff between increased coding opportunity and increased TCP round-trip-time by increasing delay at intermediate nodes.

Network coding improves TCP throughput in two ways:

1) By increasing the wireless channel capacity due to coding packets together, and

2) By reducing packet loss probability due to reduced contention on the wireless channel.

3. Network coding

The concept of network coding is easiest explained using the famous butterfly example depicted in Fig. 2. All links have unit capacity, *e.g.*, one packet per time unit, and senders S1

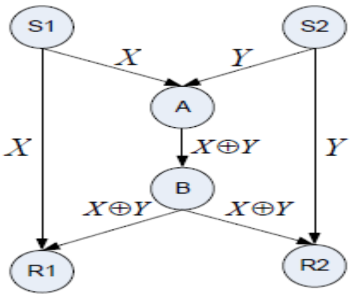

Figure 2: Butterfly example.

and S2 want to send two packets X and Y to both receivers R1 and R2. Clearly, link A-B is the bottleneck link, and hence, *four* units of time are required to transmit X and Y to R1 and R2 using a store-and-forward mechanism. However, using the transmissions outlined in the figure, the multicast problem can be solved using only *three* units of time. In this example, node A combines X and Y using XOR operation (denoted by $\oplus$) and transmits X $\oplus$ Y in a single transmission.

In general, nodes can use different coding techniques to combine packets, however, Li *et al.* [9] showed that linear coding suffices to achieve the max-flow, *i.e.*, the optimum, in single source multicast networks. Then, the problem of network coding is how to select the linear combinations that each node of the network performs. In practice, most network coding approaches are based on the concept of *random linear coding* proposed by Ho *et al.* [10]. With random linear coding, each node in the network selects the linear coding coefficients uniformly at random over a finite field in a completely independent and decentralized manner.

In wireless environments, network coding has been applied to various problems including broadcasting in ad hoc networks [7], data collection in sensor networks [7], file sharing in mesh networks [11] and reliability in lossy networks [4]. In the context of mesh networks, in particular, Wu *et al.* [3] investigated the use of network coding for the mutual exchange of independent information between two nodes in a wireless network. They showed that network coding can be used to increase the capacity of a wireless network with bidirectional traffic. Consider the network depicted in Fig. 3. Node A wants to send packet X to node C and node C wants to send packet Y to node A. With traditional store-and-forward routing, X and Y belong to two different unicast flows, one from A to C and the other from C to A. Hence, two routes are created to exchange packets between A and C. In this case, to exchange X and Y, four time slots are required.

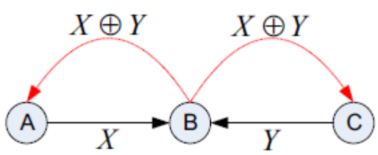

Figure 3: Network coding for bidirectional traffic.

However, by using network coding and broadcasting, as shown in the figure, the exchange can be performed in only three time slots. In the first two time slots, X and Y are transmitted to node B, and then in the third time slot, node B broadcasts X $\oplus$ Y to nodes A and C. Upon receiving X $\oplus$ Y from B, node A (C) extracts Y(X) using its existing copy of X(Y). Therefore, one transmission is saved, which effectively increases the capacity by 25% compared to traditional store and-forward scheme.

3.1 IMPACT OF NETWORK CODING ON TCP

This section studies the impact of network coding as described in the previous section on TCP throughput. Despite the simplicity of the model, it provides interesting insight about the interactions between TCP and network coding. TCP dynamics, specifically the AIMD congestion control mechanism, have a significant impact on the benefits of network coding. TCP congestion control mechanism, continuously adapts TCP sending rate to network conditions and available capacity. In particular, the AIMD mechanism is extremely sensitive to packet losses and interprets them as signs of congestion. Upon detecting a loss, TCP halves its sending rate by reducing its congestion window size to half.

In a WMN based on a contention-based MAC protocol such as IEEE 802.11, there are significant number of packet losses due to wireless channel errors and contention on the wireless medium. TCP reacts to all such packet losses by reducing its transmission rate which results in poor throughput and low wireless channel utilization. Consequently, TCP throughput is primarily limited by the end-to-end loss probability rather than the available end-to-end capacity. Hence, TCP may not significantly benefit from the increased capacity due to coding compared to a non-congestion-controlled traffic such as UDP traffic (as considered in [3]).

In the topology considered, due to close proximity of wireless nodes, contention on the wireless medium is relatively high. High contention results in high end-to-end packet loss probability which *prevents* TCP from fully utilizing the channel. In fact, TCP is not even able to utilize the channel capacity available to it without coding, and hence increasing channel capacity with coding does not significantly benefit TCP. Instead, TCP benefits from coding indirectly. Coding reduces the number of transmissions which results in lower contention on the wireless medium. This helps TCP to increase its sending rate because it faces packet loss less frequently compared to no-coding case. However, the increased throughput, in turn, leads to increased contention. In steady-state, there will be a balance between increased TCP throughput and increased contention.

3.1.1 Packet Loss Probability

To understand network coding impact on TCP throughput, simple line topology with n hops is considered. There is a TCP flow from the first node (source) to the last node (destination) in the line topology. A homogenous scenario is considered and assumed that packets (data packets or ACK packets) are lost over each hop with probability p_l. It is further assumed that the receiver sends an ACK for every successfully received data packet. If either the data packet or the ACK packet is lost, a TCP loss occurs. Therefore, to successfully send one packet from the source to the destination, $2n$ transmissions are required (n transmissions for the data packet and another n transmission for the ACK). Let

p denote the end-to-end packet loss probability seen by TCP. It is obtained that

$$p = 1 - (1 - p_l)^{2n}, \quad (1)$$

$$\approx 2np_l, \text{ for small } p_l \text{ and large } n\,. \quad (2)$$

A packet is lost either due to channel errors or collision with other transmissions. Channel errors depend on the inherent characteristics of the wireless medium and are independent of traffic load. However, collision induced losses depend on traffic load, and hence can be different with and without network coding. Let p_e and p_c denote packet loss probability due to channel errors and contention respectively. We have

$$p_l = 1 - (1 - p_e)(1 - p_c), \quad (3)$$

$$\approx p_c\,, \text{ for small } p_e / p_c. \quad (4)$$

The above approximation is valid when collision is the dominant cause of packet loss. Therefore, the end-to-end loss probability is given by

$$p = 2np_c$$

3.1.2. Collision Probability

To estimate the collision probability, we consider a time slotted system where every transmission takes one time slot. In our line topology, at most three nodes interfere with each other. Consider three such interfering nodes and assume that packets arrive at a node according to a Bernoulli process with mean X packets/slot (for a normalized channel capacity of $C = 1$ packets/slot). Let λ_1 and λ_2 denote the transmission probability in a time slot with and without coding respectively. With network coding, for every other packet arrival there is one transmission. Hence the transmission probability is given by $\lambda_1 = X/2$. Without coding, for each arrival there is one transmission. Hence the transmission probability is equal to the arrival rate $\lambda_2 = X$. Let p_{c1} and p_{c2} denote the collision probability with and without network coding. A collision occurs if more than one node transmits at the same time. Thus p_{c1} (i=1,2) is :

$$p_{ci} = {}^3C_2\,(1-\lambda_i)\,\lambda_i^2 + {}^3C_3\,\lambda_i^3 \quad (5)$$

$$\approx 3\lambda_{2i}, \quad \text{for small } \lambda_i \quad (6)$$

3.1.3. TCP Throughput

Let p_1 and p_2 denote the end-to-end loss probability with and without coding. We assume that losses due to channel errors are negligible so that all losses are due to contention. Let X_1 and X_2 denote the mean TCP throughput with and without coding. Using (6), it is obtained that

$$p_{c1} = 3(X_1/2)^2, \quad (7)$$

$$p_{c2} = 3X_2^{\,2}\,. \quad (8)$$

By substituting in (2), we obtain that $p_1 = 6/4nX_1^{\,2}$ and $p_2 = 6nX_2^{\,2}$.

Let L and R denote the TCP packet size and round-trip time respectively. Then, using the well-known square root formula [12], TCP throughput can be approximated by

$$X_i \approx \frac{L}{R\sqrt{(pi/2)}} \quad (9)$$

In steady-state, TCP sending rate X_i and packet loss probability P_i balance each other. Therefore, it is obtained that:

$$X_1 = \frac{2L}{RX1\sqrt{3n}} \quad (10)$$

$$X_2 = \frac{L}{RX1\sqrt{3n}} \quad (11)$$

which yield,

$$X_1 = \sqrt{\frac{2L}{R\sqrt{3n}}}\,, \quad (12)$$

$$X_2 = \sqrt{\frac{L}{R\sqrt{3n}}} \quad (13)$$

Several observations can be made regarding the above expressions:

(a) Impact of path length*:* As n increases, TCP throughput decreases to zero because end-to-end loss probability approaches 1.

(b) Impact of coding: It is easy to see that $X_1 = \sqrt{2}X_2$, indicating a factor of improvement $\sqrt{2}$ in TCP throughput when coding is implemented.

4. Implementation

Network coding module generates either pseudo-unicast packets or coded packets based on LQSR packets. Fig. 5 depicts the operations of our network coding implementation. The module consists of sending and receiving procedures. Two First-In-First-Out (FIFO) queues are used to store the packets. The first, called NC Cache is used to keep copies of all outgoing and incoming packets for decoding purpose. Another FIFO queue is the NC Buffer. It holds the outgoing packets that have not found a coding opportunity yet.

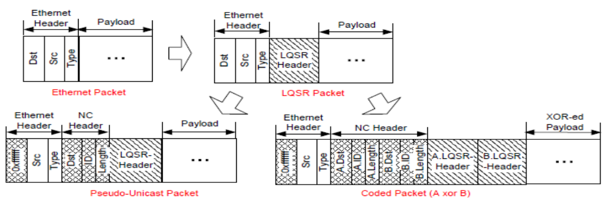

Figure 4: Packet generated by different modules

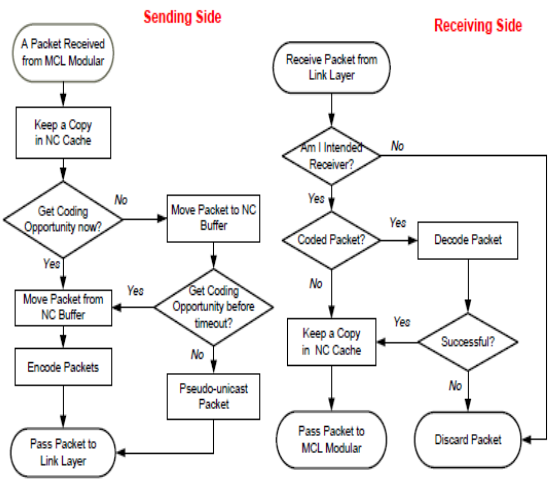

Figure 5: Network coding implementation

The sending procedure seeks coding opportunities for all outgoing packets. First, a local copy of an outgoing packet is stored in the NC Cache for decoding purpose. Then, if there is another packet in the NC Buffer going in the opposite direction, the two packets are encoded immediately. Finally, the coded packet is passed to the link layer to be transmitted.

If there is no coding opportunity when the packet arrives, it waits in the NC Buffer for a future coding opportunity. Every packet in the NC Buffer has an associated timer with an initial value of NCBufTimeout. If this timer expires, the packet will be pseudo-unicasted. NCBufTimeout is an important parameter. Note that NCBufTimeout may increase TCP throughput by creating more coding opportunity, but may decrease TCP throughput by increasing the round-trip time. Hence, there is a trade-off between the increased coding opportunity and increased round-trip-time.

Ethernet packets are from IP layer, LQSR packets are generated by MCL, and network coding module generates either pseudo-unicast packets or coded packets that are passed to MAC layer for transmission.

When receiving a packet, as shown in Fig. 5, a receiver first checks whether its address is included in the NC Header of the received packet. If not, the packet is discarded immediately.

Otherwise, if the packet is a coded packet, the receiver tries to decode it. To decode a packet, *e.g.*, packet A from a coded packet A$\oplus$ B, the receiver will look for packet B in its NC Cache. If packet B is found then packet A can be successfully decoded, and a local copy of packet A will be stored in the NC Cache. Otherwise, packet A$\oplus$B is simply discarded.

On the other hand, if the received packet is pseudo-unicasted (*i.e.*, it is a none-coded packet), the receiver only needs to store a local copy of the received packet into its NC Cache. Finally, the processed packet is passed to the upper layer via the MCL module.

5. Conclusion

Implementing network coding on selected nodes will reduce transmissions and on the overall network throughput will increase, thus reducing load on the network. Efficient and optimal algorithms could be developed to further improve the network coding.

References

[1] R. W. Yeung and Z. Zhang, "Distributed source coding for satellite communications," *IEEE Trans. Inform. Theory*, vol. 45, no. 4, pp. 1111– 1120, May 1999.
[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.
[3] Y. Wu, P. A. Chou, and S.-Y. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," Microsoft Research, Tech. Rep. MSR-TR-2004-78, Aug. 2004.
[4] D. S. Lun, M. Medard, and M. Effros, "On coding for reliable communication over packet networks," in *Proc. Allerton*, Urbana Champaign, USA, Sept. 2004.
[5] S. Deb, M. Effros, T. Ho, D. R. Karger, R. Koetter, D. S. Lun, M. Medard, and N. Ratnakar, "Network coding for wireless applications: A brief tutorial," in *Proc. International Workshop on Wireless Ad-hoc Networks*, London, UK, May 2005.
[6] S. Katti, H. Rahul, W. Hu, D. Katabi, M. M. M´edard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *Proc. ACM SIGCOMM*, Pisa, Italy, Sept. 2006.
[7] C. Fragouli, J. Widmer, and J.-Y. L. Boudec, "On the benefits of network coding for wireless applications," in *Proc. WiOpt*, Boston, USA, Apr. 2006, pp. 1–6.
[8] J. Liu, D. Goeckel, and D. Towsley, "Bounds on the gain of network coding and broadcasting in wireless networks," in *Proc. IEEE INFOCOM*, Anchorage, USA, May 2007.
[9] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.
[10] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.
[11] A. A. Hamra, C. Barakat, and T. Turletti, "Network coding for wireless mesh networks: A case study," in *Proc. IEEE WoWMoM*, Buffalo, USA, June 2006, pp. 103–114.
[12] T. J. Ott, J. Kemperman, and M. Mathis, "The stationary distribution of ideal TCP congestion avoidance," in *Proc. DIMACS Workshop on Performance of Realtime Applications on the Internet*, South Plainfield, USA, Nov. 1996.
[13] Microsoft Research, Mesh Networking. [Online]. Available: http: //research.microsoft.com/mesh/
[14] Y. Huang, M. Ghaderi, D. Towsley, and W. Gong, "TCP performance in coded wireless mesh networks," Electrical and Computer Engineering Department, University of Massachusetts Amherst, Tech. Rep. TR- 07-CSE-02, Apr. 2007. [Online]. Available: http://tennis.ecs.umass.edu/ yhuang/papers/tcp-netcoded-techreport.pdf
[15] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *ACM MobiCom '03*, San Diego, California, September 2003.
[16] Internet packet size distributions: Some observations. http://netweb.usc.edu/ rsinha/pkt-sizes/.
[17] T. Ho, R. Koetter, M. M´edard, D. Karger, and M. Effros. The Benefits of Coding over Routing in a Randomized Setting. In *ISIT, 2003*.
[18] T. Ho, B. Leong, M´edard, R. Koetter, Y. Chang, and M. Effros. The Utility of Network Coding in Dynamic Environments. In *IWWAN*, 2004.
[19] p. a. IEEE 802.11 WG. Wireless lan medium access control (mac) and physical layer (phy) specifications. *Standard Specification,IEEE, 1999.*
[20] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, 2003.
[21] A. Kamra, J. Feldman, V. Misra, and D. Rubenstein. Growth codes: Maximizing sensor network data persistence. In *ACM SIGCOMM*, 2006.
[22] B. Karp. *Geographic Routing for Wireless Networks*. PhD thesis, Harvard University, 2000.
[23] S. Katti, D. Katabi, W. Hu, H. S. Rahul, and M. M´edard. The importance of being Opportunistic: Practical network coding for wireless environments, 2005.
[24] R. Koetter and M. M´edard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking, 2003*.
[25] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The click modular router. *ACM Transactions on Computer Systems, August 2000*.
[26] S. R. Li, R. W. Yeung, and N. Cai. Linear network coding. In *IEEE Transactions on Information Theory*, 2003.
[27] D. S. Lun, M. M´edard, R. Koetter, and M. Effros. Further results on coding for reliable communication over packet networks. In *IEEE International Symposium on Information Theory (ISIT 05)*, 2005.

[28] D. S. Lun, N. Ratnakar, R. Koetter, M. M´edard, E. Ahmed, and H. Lee. Achieving Minimum-Cost Multicast: A Decentralized Approach Based on Network Coding. In *IEEE INFOCOM, 2005*.
[29] Internet packet size distributions: Some observations. http://netweb.usc.edu/ rsinha/pkt-sizes/.

Author Profile

Nikumani Choudhury completed B.Tech(IT) from Assam University, Silchar, Assam 2012, and pursuing M.Tech(IT) from Gauhati University, Guwahati, Assam. Active field of interest in Computer Networks.