

Modified Token Based Congestion Control Scheme for Opportunistic Networks

Emmanuel Adewale Adedokun¹, Hamisu Abubakar Adamu² and Idris Salawu Shaibu³

¹Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria

²Department of Communication Engineering, Ahmadu Bello University, Zaria, Nigeria

³Department of Computer Science, Federal Polytechnic, Kaura Namoda, Zamfara State, Nigeria

To address congestion issues in Opportunistic Networks (OppNets), a modified token-based congestion control with adaptive forwarding mechanism is proposed. The mechanism allows the network nodes holding a valid token to inject message into the network or other neighboring node. At the point of congestion, the algorithm has the potential to redirect the traffic from a more congested node to congestion free node for the purpose of effective resource utilization and fairness in the network. Tokens are evenly distributed throughout the network. Using the opportunistic network environment (ONE) simulator, we illustrate the performance of a modified token-based congestion control algorithm, which results in reduction of dropped messages and network transit time due to congestion across all the scenarios considered. At different queue sizes of (QS-10, QS-20, QS-30 and QS-40), modified token based congestion control algorithm has 13.91%, 10.71%, 5.46%, and 4.22%, respectively, reduction in dropped messages. In addition, at the greatest connected component values of 50%, 60%, 70% and 80%, the modified token-based congestion control has 8.34%, 2.19%, 4.61%, and 7.63%, respectively, decrease in network transit time. These results are substantial, because they indicate a reduction in both network storage and time.

ACM CCS (2012) Classification: Networks → Network types → Ad hoc networks → Mobile ad hoc networks

Networks → Network properties → Network mobility

Networks → Network protocols → Network layer protocols → Routing protocols

Networks → Network performance evaluation → Network simulations

Networks → Network performance evaluation → Network performance analysis

Keywords: opportunistic networks, token-based congestion control, modified token-based congestion control scheme, prophet routing protocol

1. Introduction

In recent years, mobile communication devices like smart phones, smart watches, smart vehicles and activity trackers have been integrated into our daily life. These devices have unceasingly evolved with the advancement in technologies which include short-range mobile communication, as well as improvement in power consumption [1]. These novel technologies have encouraged the growth of opportunistic networks (OppNets). An OppNet is similar to mobile wireless ad hoc network (MANET); nevertheless, it has distinguishing features different from the traditional MANET [2]. An OppNet is comprised of spatially distributed and human-carried mobile devices with a short-range wireless communication module. There is no stable link between a source and a destination in OppNet [3]. The traditional routing protocols for MANETs are unsuitable because of the sparse density as well as high node mobility in OppNet. Therefore, store-carry-and-forward approach is implemented in OppNet, such that messages are relayed through the intermediate nodes by random contacts [4].

A node always buffers messages, moves about until contact opportunity exists to forward or duplicate the messages to the destinations or other relay nodes. Thus, the buffer capacity of nodes is constrained; hence uneven load distribution and large volumes of traffic drive the buffer capacity to saturation. Due to the congestion, the throughput of the network is ex-

tremely reduced [5]. Thus, congestion control is considered fundamental to ensure that OppNet nodes are free from congestion and can serve as relays to aid in message delivery process. OppNets have attracted much research effort in the areas of security and routing [6]. Meanwhile, one can see that the convergence layer for OppNets still depends heavily on TCP. Therefore, the congestion control techniques contained in the TCP cannot adapt to the dynamic evolving topology presented by the OppNets. So, finding an appropriate congestion control approach for OppNet is considered a critical issue.

The remainder of this paper is structured as follows. Section 2 describes the history of congestion control mechanism in different networking environments and clearly highlights previous congestion control techniques for OppNet. Section 3 presents modified token-based congestion control and the general algorithm. Simulation environment and results are presented in Section 4. Discussion of the results is in Section 5. Finally, we conclude the paper in Section 6.

2. Background and Related Works

Congestion control has been given a significant concern for networking technologies from the time when congestion collapse [7] was noticed in the early Internet. The solution adopted then, which still enables protection against congestion collapses today, is that sources should mitigate their sending speed into the network discovered to be congested [8]. Recent research efforts show that a scheme can be implemented into the routing protocol to offload the traffic from congested node to congestion free nodes instead of instantly dropping the packets, which eventually constitutes data to be retransmitted there by mitigating the performance of the networks[9].

Even though, there exist substantial research efforts on congestion control schemes for wireline networks [10], [11], [12] and for wireless networks [13], [14], [15], only a few address OppNets. These congestion control schemes can be categorized as either operating as congestion avoidance, reacting to congestion events or utilizing hybrid technique, which attempt avoidance, and thus reserve the potential to redirect the traffic from congested to conges-

tion free nodes of the OppNets. The traditional networks depend on connected paths to transmit congestion information back to the source from either the relay node or the destination node via acknowledgement method. The acknowledgement mechanisms transit the network through undisrupted paths back to the source and have bounded time to travel the reverse path. OppNets lacks stable continuous paths, and therefore cannot notify the source of any network congestion. Thus, it cannot depend on timely feedback approach [6].

In [16], the authors approach congestion avoidance based on local information and economic concept. The algorithm used computes the price and dropping probability of a bundle being dropped from a node in order to facilitate the acceptance and rejection decision making for the bundle. The concept of price denotes the ranks for each bundle and node, based on the bundle and node's properties. The cash-in-hand represents the present buffer capacity of the node to store more bundles and confidence level determines the degree of confidence of node to forward the buffered bundles to their respective next hops. [5] implemented congestion control based message deleting and transferring mechanism. The congested node computes the storage value of each message as a function of forwarding probability and message time-to-live. The message with the least storage value is deleted. The relay node determines the receiving value of the message in accordance with the forwarding probability and its unused buffer space prompting the congested node to transfer the message with maximum receiving value. Our work differs from the aforementioned mechanism, such that our scheme intends to match the traffic input to the capacity of the network and reroute the traffic from congested node to congestion-free node within the network. In this regard, our algorithm can be termed hybrid congestion technique.

3. Modified Token Based Congestion Control Scheme (mTBCC)

The fundamental concept of the Modified Token Based Congestion Control (mTBCC) scheme emanates from the need to match the amount of traffic entering a network to the total

network capacity. When congestion occurs, it should redirect the traffic from congested node to congestion free node or otherwise drop the message if buffers of all nodes are full. If the network or individual node is supplied with an amount of traffic it can transmit, hence congestion collapse [17] can be prevented. The token concept is like token ring, and token bus, such that nodes ought to hold a valid token to forward data. However, it differs from the proposed approach, as mTBCC simply needs a token to forward data into the network or to other nodes and not to execute per hop transferring of messages. In this context, the entire network is regarded as a constrained resource whose access is enabled by tokens, contrary to the ring approach mentioned earlier. Modified token-based congestion control studies the network as a black box. The fee for a node to infuse a single data message into the black box is a single token, and tokens are recovered once a message leaves the network. Messages can leave the network in one of the following ways:

- Messages arrive at their destination
- Messages are erased due to insufficient buffer space at all intermediate nodes
- Messages are dropped based on time-to-live timer expiration.

Nodes that perform any one of the above-mentioned techniques for message termination increase their token count, thereby recovering such message's token. It is assumed that nodes do not disappear from the network and that the total number of tokens remains unchanged over time. The token mechanism operates as follows: initially tokens are uniformly circulated among nodes within the network. Once the applications produce data to be disseminated, it is buffered in an application queue. Meanwhile, the algorithm is equivalent to OppNet, an assumption was made that all nodes within the network are cooperating in message forwarding and sharing of extra tokens. The token mechanism is independent of the routing algorithm and is only activated when best next hop is found. The routing protocol used to assess modified token-based congestion control is presented in Section 4 in a simulation environment. The mTBCC algorithm is utilized to determine how a message gets into the network and the response of nodes once message existence is presented below and

its flow chart is shown in Figure 1, where: BS_O = buffer space over; BS_{th} = buffer space threshold; MC_{th} = migration cost threshold; FBS = free buffer space; SC = storage cost; TC = transmission cost; $TTL-t$ = message time-to-live.

1. Source queries routing protocol to determine if one of the new neighbours is the best next hop for the message held in the application queue.

Algorithm 1. Source node encounters new neighbour(s).

```

if best next hop  $\neq$  NULL AND my token count  $\geq$  1 then
    decrement token count and transmit message to
    the best next hop
else if token count  $<$  1 then
    query neighbours for extra token
    if extra token is available then
        decrement neighbours token count and
        transmit message to the best next hop
    end if
end if

```

2. Source queries routing protocol to determine if one of the new neighbours is the best next hop for the message held in the application queue.

Algorithm 2. Source node generates new message.

```

if the best next hop  $\neq$  NULL AND my token count  $\geq$  1
    then decrement token count and transmit
    message to the best next hop
else if token count  $<$  1 then
    query neighbours for extra token
    if extra token is available then
        decrement neighbours token count and
        transmit message to the best next hop
    end if
end if

```

3. Redirect the traffic from congested node to congestion free neighbouring node taking cognizance of least migration cost and largest available free buffer space. If buffer of all nodes is full, drop the message AND increment token count end if.

Algorithm 3. Message dropped in the network.

```

Upon receiving message node checks
if queue state = FULL then

```

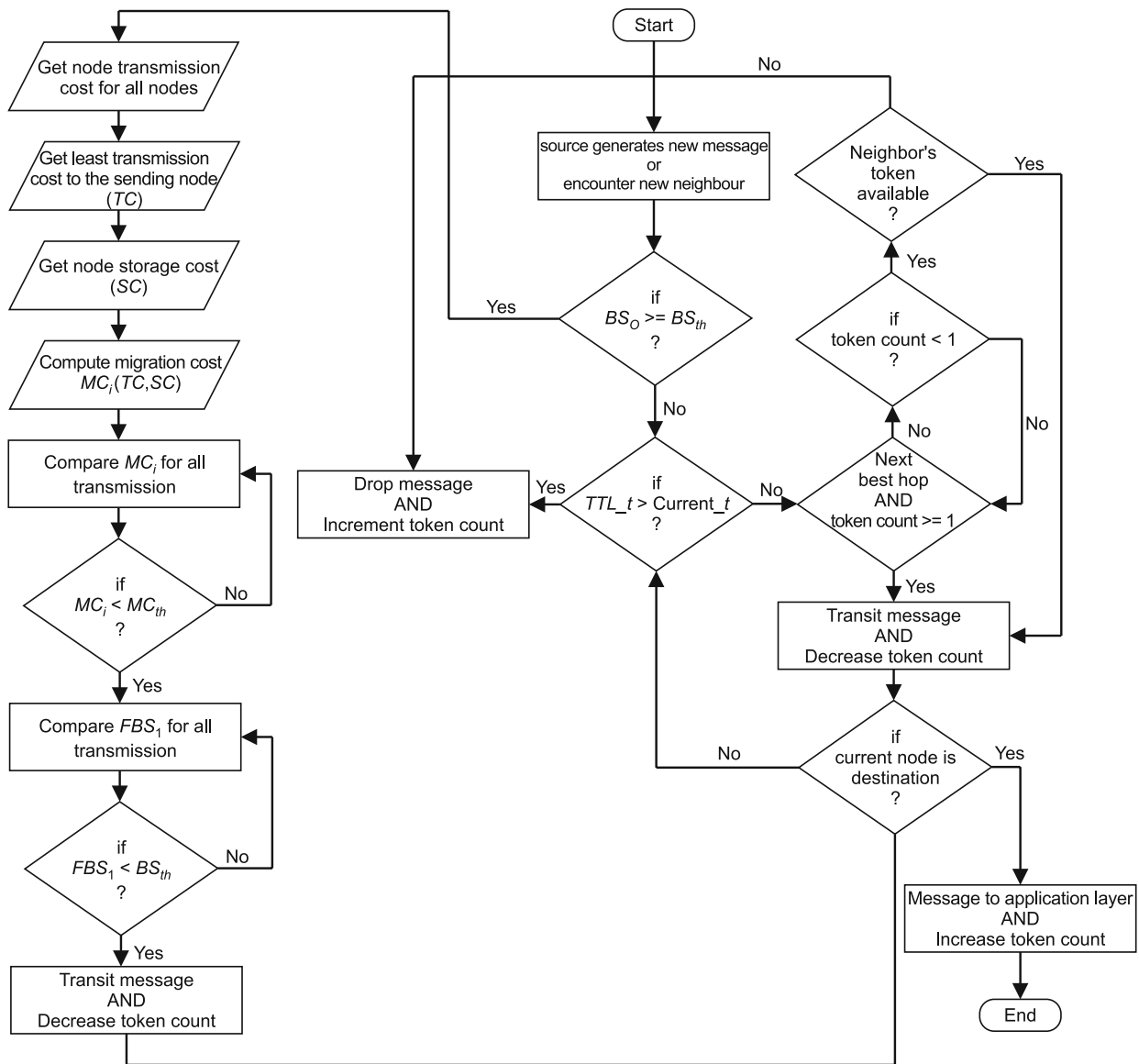


Figure 1. Flow chart for mTBCC.

4. Upon routing beacon update receipt for each message in the queue check.

Algorithm 4. Message TTL timer expires.

```

if TTL_timer > Current_timer then
    Drop message AND increment token count
end if
  
```

5. Upon receiving message node checks.

Algorithm 5. Message reaches destination.

```

if message destination = this then
    Pass the message to application layer
    AND increment token count
end if
  
```

4. Simulation Environment and Analysis

Modified token-based congestion control algorithm was simulated using the opportunistic network environment (ONE) simulator written in java. The ONE simulator has inbuilt graphical user interface for visualizing node mobility, inter-node contact and message passing during simulation in real time. The collection of results and analysis are carried out via visualization reports as well as post-processing tools [18]. The algorithm was validated using the Helsinki default map of the ONE simulator. This simulator has been widely used to evaluate various

OppNets routing protocols performance and congestion control mechanism [17]. The following subsection describes how the network of interest was created.

4.1. Network of Interest

The essence of this research is to focus on networks which exhibit intermittent connectivity due to fast node mobility. The modified token based congestion control focuses on network nodes that see each other regularly, although due to fast mobility contemporaneous end-to-end connectivity cannot be sustained. Networks were created with random mobility model and varying node densities as well as node speed controlled connectivity. After examining these networks, we required another control parameter to vary which would enhance better control over network connectivity characteristics. The greatest connected component (GCC) metric provides better control for network generation. Utilizing greatest connected component we produced a mobility model that enhances additional control over the network connectedness. This is significant in carrying out research for OppNets that are similar to MANETs, but are not MANETs. MANET means the nodes meet each other regularly, however, their fast mobility makes MANET protocols unsuitable, compared to OppNets where latencies between contacts are unpredictable. The model starts by initially placing nodes on a grid within their radio coverage so that only certain percentage of network would consist of nodes belonging to the greatest connected component. Assume a network of 100 nodes and a desired greatest connected component of 65% nodes: these nodes are positioned so that for a given range approximately 65 nodes are in the greatest connected component. It follows that an individual node executes the algorithm 6.

4.2. Environment Setup and Discussion

The networks utilized were generated from the greatest connected component mobility model as outlined above, which comprised 60 nodes operating in a Helsinki region (4500 × 3400 m) with a range of 5 m. Nodes move within this region at the speed of 5 m/s. The routing proto-

Algorithm 6. Node behavior within the greatest connected component.

```

if Current_GCC > Desired_GCC then
    Nodes inside the GCC move outside with
    probability
     $P = (\text{Current\_GCC} - \text{Desired\_GCC}) / \text{Current\_GCC}$ 
    Nodes outside the GCC stay outside
    (we are already over desired GCC)
end if
if Current_GCC < Desired_GCC then
    Nodes inside the GCC stay inside
    (we need them to stay)
    Nodes outside the GCC move toward the GCC
    centre with probability  $P = 1$ 
    (this improves the GCC stability and prevents
    oscillations)
end if

```

col is based on predictability probability. Nodes update their routing table with predictability metric and compare neighbouring predictability metric values for the destination with this node's value. The node with maximum value for predictability metric is considered next suitable hop. The data stated in this section are obtained from all the simulations carried out for ten (10) average different runs over the same network.

As mentioned earlier, our mobility model is centred on random movement and we generate sufficient traffic to cause congestion within the network. The modified token-based congestion control mechanism cannot promise zero message drop. Thus, we evaluate the modified token-based congestion control algorithm over other interesting metrics, which include dropped messages obtained from the Message Status Report and total network transit time computed from equation 1. The main simulation parameters are presented in Table 1 and the simulation process of the mTBCC is shown in Figure 2.

$$NTT = \text{Latency} - \text{contact time of first node} \quad (1)$$

4.3. The Parameter Values

Firstly, we determine the value of token to distribute across the network nodes and observe the changes of dropped messages as greatest connected component increases from 50% to 80% respectively. We utilize PROPHET routing algorithm in the simulation. For various

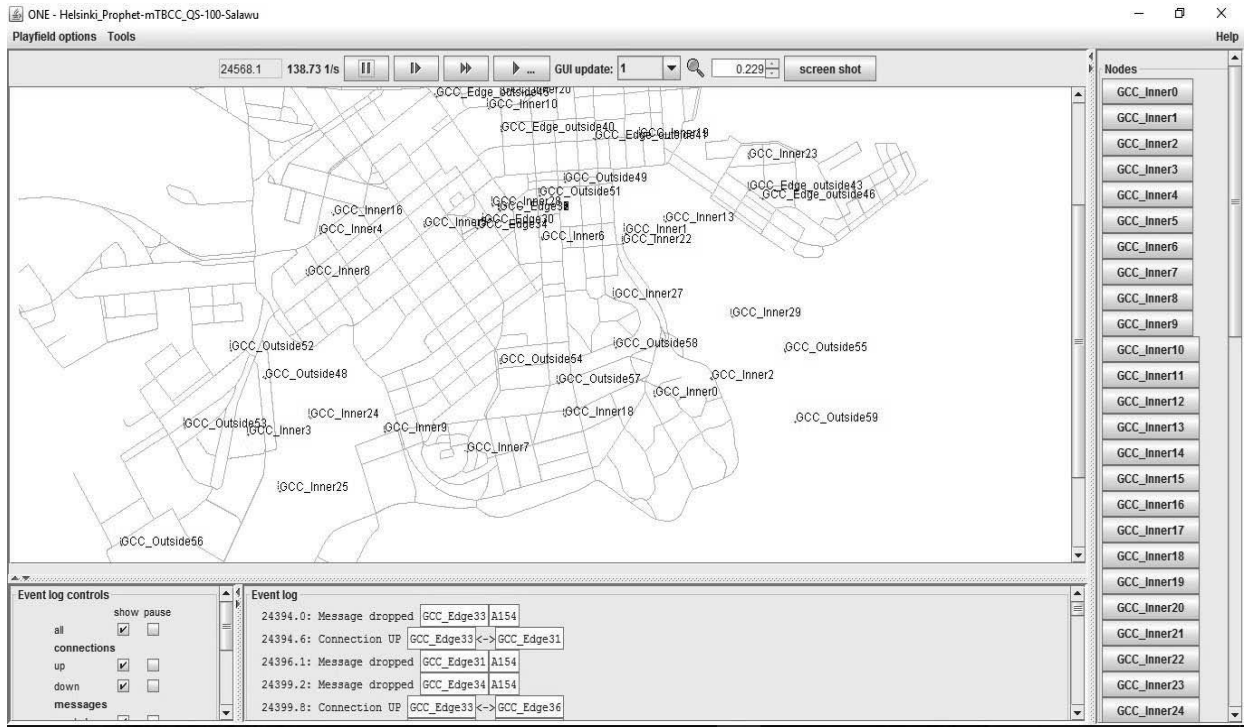


Figure 2. Simulation process of modified token based congestion control.

Table 1. Simulation parameters.

Parameter	Value
Simulation area	4500 × 3400 m
Transmission range	5 m
Transmission speed	2 Mbps
Total number of nodes	60
Message size	500 KB – 3 MB
Message creation interval	25 ~ 35 s
Simulation time	43 200 s
Node movement speed	4.5 ~ 5 m/s
Node buffer size	5 MB, 20 MB, 60 MB 200 MB, 300 MB

Table 2. Dropped messages for TBCC.

Connected	QS-10	QS-20	QS-30	QS-40
50%	22893	17012	16292	14769
60%	11136	9402	8500	6190
70%	3389	2962	2762	1484
80%	1174	954	802	707

Table 3. Dropped messages for mTBCC.

Connected	QS-10	QS-20	QS-30	QS-40
50%	21504	15467	15031	14599
60%	10737	8828	8293	5460
70%	3037	2830	2733	1436
80%	756	720	710	702

values of token in the network, the corresponding decrease in dropped message is shown in Table 2 for the conventional token-based congestion control and in Table 3 for the modified token-based congestion control. It is observed in tables 2 and 3, that as the values of queue

size increase from 10 to 40, a corresponding decrease in dropped messages is obtained for different values of the GCC, expressed in percentage from 50% to 80% for both algorithms.

Table 4. Network transit time for TBCC.

Initial number of token/node	50% GCC Connected	60% GCC Connected	70% GCC Connected	80% GCC Connected
	NTT(secs)	NTT(secs)	NTT(secs)	NTT(secs)
10	2859.69	2501.51	1739.38	1549.76
20	3466.32	2590.92	1860.30	1735.95
40	4930.32	2750.34	2200.32	2060.15
60	6048.83	2825.75	2459.41	2329.08
80	6048.24	2825.62	2459.38	2329.43
100	6048.30	2825.15	2459.42	2329.18

Table 5. Network transit time for mTBCC.

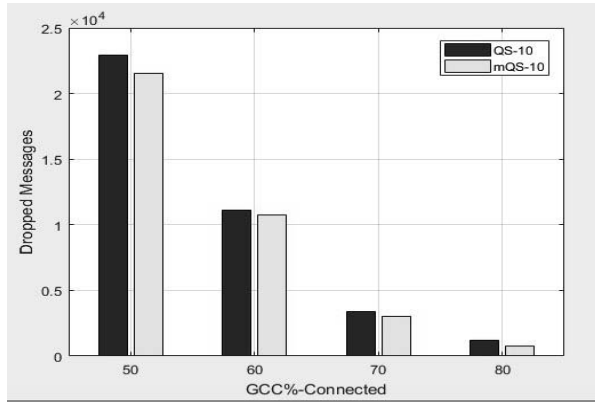
Initial number of token/node	50% GCC Connected	60% GCC Connected	70% GCC Connected	80% GCC Connected
	NTT(secs)	NTT(secs)	NTT(secs)	NTT(secs)
10	2566.49	2431.29	1669.66	1339.91
20	3031.41	2510.18	1760.25	1543.74
40	4406.23	2680.19	2010.28	1950.54
60	5715.84	2781.61	2380.20	2206.32
80	5715.57	2781.58	2380.34	2206.61
100	5715.87	2781.57	2380.28	2206.11

Secondly, we determine the amount of token dissemination across the network nodes and also study their effects on network transit time as the number of tokens per node varies for the given routing algorithms. For different values of token in the network, the corresponding network transit time for the conventional token-based congestion control is shown in Table 4 and that of the modified token-based congestion control in Table 5 respectively, as obtained from all the simulations carried out using Helsinki simulation environment. Tables 4 and 5 show that network transit time increases as token increases from 10 to 60, beyond which there is no substantial variation in the network transit time at tokens of 80 and 100, because the network is saturated with token. Finally, the small differences observed in the total network transit time are due to the effect of selfish nodes in the network.

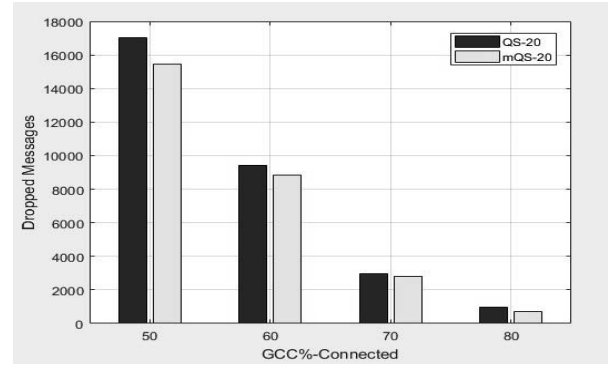
5. Results

In this section, we compare performances of the proposed modified token-based congestion control with the conventional token-based congestion control algorithm under the same simulation setup. The performance metrics used are dropped message and network transit time through ONE simulator. Figure 3 shows the amount of dropped messages recorded for both conventional token-based congestion control and the modified token based congestion control mechanisms, using Helsinki area test bed.

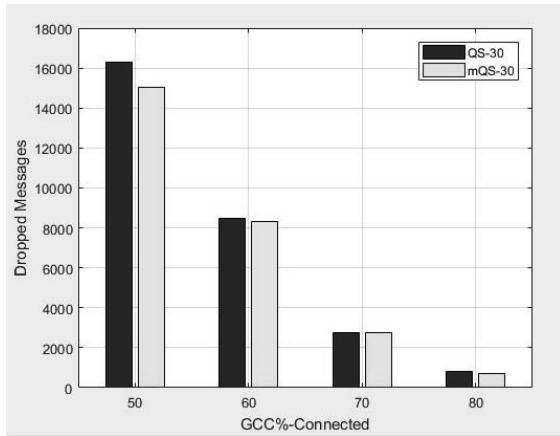
Figure 3(a) – 3(d) compares the performance of dropped messages for both algorithms. It is observed from the bar charts that the modified token-based congestion control algorithm significantly minimizes the number of dropped messages regardless of the queue size in com-



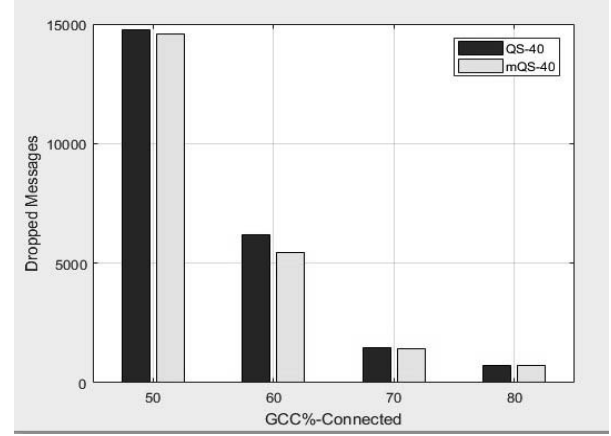
(a) QS-10



(b) QS-20



(c) QS-30



(d) QS-40

Figure 3. Dropped messages versus GCC % for mTBCC and TBCC.

parison with conventional token-based congestion control algorithm as the network connectivity is increased in terms of GCC expressed in percentage from 50% to 80%. This occurs due to the fact that mTBCC algorithm has the ability to enhance effective resource utilization and fairness in the network by rerouting the message from more congested to congestion free nodes of the OppNet.

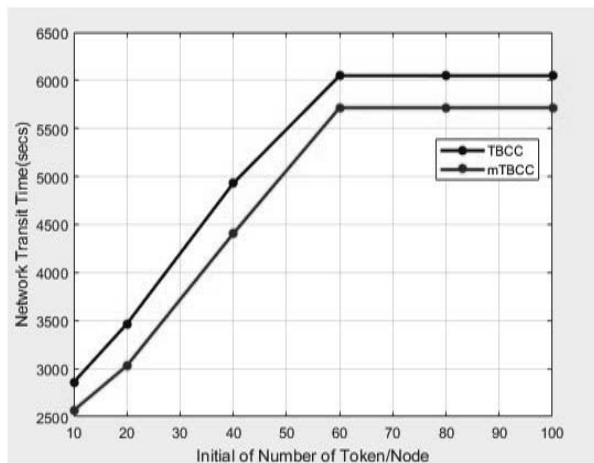
Figure 4 depicts the network transit time for both the conventional token-based congestion control mechanism and the modified token based congestion control mechanism using Helsinki area test bed.

Figure 4(a) – 4(d) compares the performance of network transit time for both algorithms. The network transit time for both algorithms increases linearly as token increases from 10 to

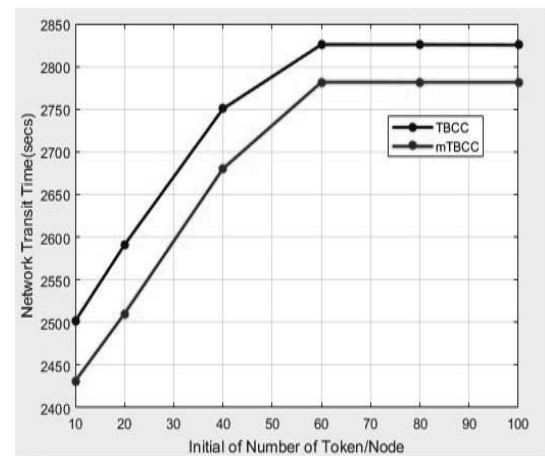
60 and becomes constant for both algorithms at token of 80 and 100, due to the saturation of the network with token. Further, the mTBCC algorithm significantly reduced the network transit time as compared with the conventional TBCC algorithm which is attributed to the fact that mTBCC algorithm has the potential to redirect the message from the congested node to congestion free node of the OppNet, by providing effective resource utilization and fairness in the network. This brings the message closer to the desired destination.

6. Conclusion and Future Work

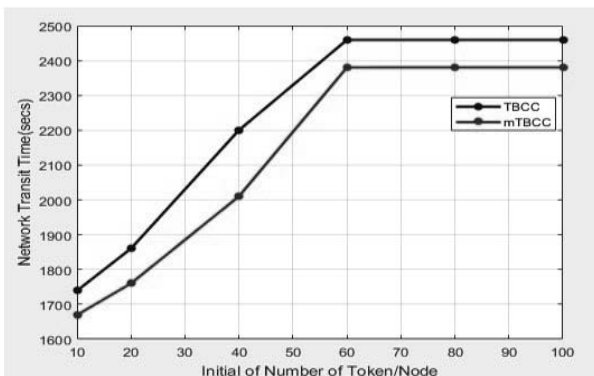
In this paper, we have studied the performance analysis of both conventional token-based congestion control algorithm and the modified to-



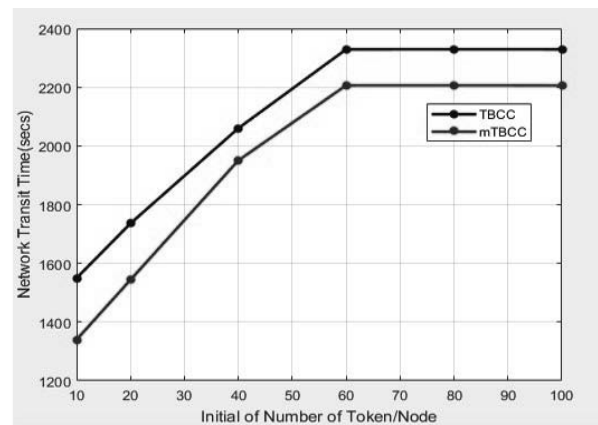
(a) 50% GCC



(b) 60% GCC



(c) 70% GCC



(d) 80% GCC

Figure 4. Network transit time versus initial number of token/node for mTBCC and TBCC.

ken-based congestion control algorithm. Both algorithms implement simple token scheme that lets nodes holding a valid token to infuse message into the network. Simulation results illustrate that the modified token-based congestion control mechanism produce substantial reduction in dropped messages and decrease network transit time as compared to the conventional token based congestion control mechanism. In our future work, we will study dynamic creation of token and deletion utilizing localized congestion detection mechanism. Also, the concept of token scheme is associated with network capacity. There has been little work in determining the opportunistic network capacity, so we will examine the effects of selfish node in the network using incentive strategy.

References

- [1] F. Zhang *et al.*, "Energy-aware Congestion Control Scheme in Opportunistic Networks", *IEEE Transactions on Electrical and Electronic Engineering*, vol. 12, no. 3, pp. 412–419, 2017. <https://doi.org/10.1002/tee.22392>
- [2] B. Zhao *et al.*, "Wireless Sensor Network Reliability Modelling based on Masked Data", *International Journal of Sensor Networks*, vol. 17, no. 4, pp. 217–223, 2015. <https://doi.org/10.150/IJSNET.2015.069584>
- [3] Y. Cao and Z. Sun, "Routing in Delay/Disruption Tolerant Networks: A Taxonomy, Survey and Challenges", *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 654–677, 2013. <https://doi.org/10.1109/SURV.2012.042512.00053>

- [4] V. F. Mota *et al.*, "Protocols, Mobility Models and Tools in Opportunistic Networks: A Survey", *Computer Communications*, vol. 48, pp. 5–19, 2014.
<https://doi.org/10.1016/j.comcom.2014.03.019>
- [5] D. Y. Zhang *et al.*, "Congestion Control Strategy for Opportunistic Network based on Message Values", *Journal of Networks*, vol. 9, no. 2, pp. 1132–1138, 2014.
- [6] E. Coe, and C. Raghavendra, "Token based Congestion Control for DTNs", in *IEEE Aerospace Conference*, 2010.
<https://doi.org/10.1109/AERO.2010.5446944>
- [7] J. Nagle, "Congestion Control in IP/TCP Internet Works", 1984.
- [8] V. Jacobson, "Congestion Avoidance and Control", in *ACM SIGCOMM Computer Communication Review*, 1988.
- [9] M Radenkovic and A. Grundy, "Efficient and Adaptive Congestion Control for Heterogeneous Delay-tolerant Networks", *Ad Hoc Networks*, vol. 10, no. 7, pp. 1322–1345, 2012.
<https://doi.org/10.1016/j.adhoc.2012.03.013>
- [10] L. Xu *et al.*, "Binary Increase Congestion Control (BIC) for Fast Long-distance Networks", in *INFOCOM 2004, Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004.
- [11] D. X. Wei *et al.*, "FAST TCP: Motivation, Architecture, Algorithms, Performance", *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1246–1259, 2006.
<https://doi.org/NET.2006.886335>
- [12] D. Katabi *et al.*, "Congestion Control for High Bandwidth-delay Product Networks", *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4, pp. 89–102, 2002.
- [13] K. Chandran *et al.*, "A Feedback-based Scheme for Improving TCP Performance in Ad Hoc Wireless Networks", *IEEE Personal Communications*, vol. 8, no. 1, pp. 34–39, 2001.
<https://doi.org/10.1109/98.904897>
- [14] H. Lim *et al.*, "TCP Performance over Multipath Routing in Mobile Ad Hoc Networks", in *Communications 2003, ICC'03, IEEE International Conference on*, 2003.
- [15] X. Yu, "Improving TCP Performance over Mobile Ad Hoc Networks by Exploiting Cross-layer Information Awareness", in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, 2004.
- [16] R. Das *et al.*, "A Novel Congestion Control Scheme for Delay Tolerant Networks", in *Proceedings of the International Conference on Selected Topics in Mobile and Wireless Networking (iCOST)*, 2011.
- [17] A. Goundan *et al.*, "Efficient Broadcasting in Delay Tolerant Networks", *IEEE GLOBECOM 2008, Global Telecommunications Conference*, 2008.
- [18] A. Keränen *et al.*, "The ONE Simulator for DTN Protocol Evaluation", in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, 2009, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Received: October 2017

Revised: January 2018

Accepted: April 2018

Contact addresses:

Emmanuel Adewale Adedokun
Department of Computer Engineering
Ahmadu Bello University
Zaria, Nigeria
e-mail: wale@forum.org.ng

Hamisu Abubakar Adamu
Department of Communication Engineering
Ahmadu Bello University
Zaria, Nigeria
e-mail: hamisu99@yahoo.com

Idris Salawu Shaibu
Department of Computer Science
Federal Polytechnic
Kaura Namoda,
Zamfara State, Nigeria
e-mail: salawuidriss@gmail.com

EMMANUEL ADEWALE ADEDOKUN holds a PhD in electrical engineering (network security), as well as MSc and BE in electrical engineering, in 2014, 2009, and 1999, respectively, all from the Ahmad Bello University Zaria, Nigeria. He is currently a senior lecturer at the Department of Computer Engineering, Ahmadu Bello University. His research interests include: network security, networking and network management, network congestion control, IoT, WSN, artificial intelligence, bandwidth management and optimization, and system audit. Dr. Adedokun is a member of IEEE, COREN, CPN, and a life member of NCS.

HAMISU ABUBAKAR ADAMU is a graduate of Ahmadu Bello University, Zaria Nigeria. He obtained his BE, MSc, and PhD degrees in electrical engineering in 1991, 2005, and 2014, respectively. Dr. Adamu is currently working at the Department of Communication Engineering, Ahmadu Bello University, Zaria. His areas of specialization are instrumentation and computer control.

IDRIS SALAWU SHAIIBU received his BE degree in electrical and computer engineering from the Federal University of Technology, Minna, Nigeria, in January 2006, and an MSc in computer engineering from Ahmadu Bello University Zaria, Nigeria in December 2017. Mr. Shaibu is member of the academic staff, currently working at the Department of Computer Science, Federal Polytechnic, Kaura Namoda, Zamfara State, Nigeria. His research interests include network security, congestion control in active networks, routing in opportunistic networking, IoT, and WSN.
