On Constructing Contention Aware Connected Dominating Sets (CDS) for inter-connectivity among Internet of Things (IoT) devices

Md. Sadiqur Rahman Sohel

Department of Computer Science and Engineering,
Bangladesh University of Engg. and Technology, Dhaka, Bangladesh.
E-mail:sadiqurr8@gmail.com

Chowdhury Nawrin Ferdous

Department of Mathematics and Computer Science,
University of lethbridge, Canada.
E-mail:c.nawrinferdous@uleth.ca

Ashikur rahman

Department of Computer Science and Engineering,
Bangladesh University of Engg. and Technology, Dhaka, Bangladesh.
E-mail:ashikur@cse.buet.ac.bd

Akib Jawad Nafis

Department of Computer Science and Engineering,
Bangladesh University of Engg. and Technology, Dhaka, Bangladesh.
E-mail:a.j.nafis@gmail.com

Salsabil Arabi Shushmi

Department of Computer Science and Engineering,
Bangladesh University of Engg. and Technology, Dhaka, Bangladesh.
E-mail:salsabilarabishusmi007@gmail.com

Raqeebir Rab

Department of Computer Science and Engineering,
Ahsanullah University of Science and Technology, Dhaka, Bangladesh.
E-mail:jishan005@gmail.com

Abstract: The heterogeneous IoT devices (such as smartphones, or sensors) are often equipped with wireless interfaces which can be used to create self-organising multi-hop networks. Any device/node in such networks often requires to broadcast packets for data dissemination and route discovery. The straightforward way to accomplish broadcasting is through *flooding*, where each node transmits the broadcast message only once. Drawback of such a naive mechanism is its serious bottleneck on network throughput caused by redundant traffic, serious contention, and collision. In the literature, several methods for creating a connected dominant set (CDS) were proposed to overcome performance bottleneck. CDS can be used as a virtual backbone for broadcasting where only the member nodes of the CDS would forward the message. CDS construction could be either centralised or distributed. The algorithms proposed in the literature aim at minimizing the number of forwarding. Neither centralised nor distributed approaches minimise *contention* that we address in this paper. The contention occurs when certain nodes located in close vicinity of each other want to access a shared channel. Notably, at the time of contention one exclusive node obtains opportunity to transmit on the medium and others need to postpone their sending on the shared medium. In this paper, we also provide a novel mathematical analysis of contention and show that contention is heavily dependent on node density in the network and transmission radius of each node. Then a new centralised algorithm called *Contention Aware Connected Dominating Set (CACDS)* is devised which intellectually selects member nodes while creating a CDS. This helps to reduce *contention*. Since collecting global network topology is very difficult to achieve entirely, a distributed algorithm and a hybrid distributed algorithm has also been devised. Finally, the proposed algorithms have been implemented in the state-of-the-art simulator *NS-2*. The proposed heuristics' performance has been captured using simulation experiments under practical settings which shows a significant reduction in contention. Moreover, their performance outperforms some other state-of-the-art algorithms' performance with regards to contention and delay minimisation, although the number of forwarding has been increased marginally.

Keywords: Wireless Ad-hoc Network; Broadcasting in IoT-devices; Connected Dominating Set (CDS); Network Contention; Mathematical analysis of Contention; Contention Aware Connected Dominating Set (CACDS);

Reference to this paper should be made as follows: Sadiqur Rahman Sohel, Chowdhury Nawrin Ferdous, Ashikur rahman, Akib Jawad Nafis, Salsabil Arabi Shushmi and Raqeebir Rab. (2019) 'On Constructing Contention Aware Connected Dominating Sets (CDS) for inter-connectivity among Internet of Things (IoT) devices', *International Journal of Metadata, Semantics and Ontologies*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Ashikur Rahman completed B.Sc. and M.Sc. degrees in Computer Science and Engineering from the Department of CSE, Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in 1998 and 2001 respectively. He earned his Ph.D. in 2006 from the Department of Computing Science, University of Alberta, Canada. After finishing his Ph.D., he worked as a postdoctoral researcher at Simon Fraser University (2007), University of Calgary (2010 - 2011), McGill University (2011), Canada and Binghamton University (2012), USA. His research interests include cyber-physical systems, wireless ad-hoc and sensor networks, machine learning, swarm intelligence, data center networks, and neural networks. At present he is working as a Professor in the Department of CSE, BUET.

Raqeebir Rab earned his B.Sc. (Hons.) degree in Science (major in Computer Science) from the Augustana Faculty, University of Alberta, Canada in 2004. She received M.Sc. in Computer Science from the Concordia University, Montreal, Canada in 2012. At present, she is working as an Assistant Professor at the Department of CSE, Ahsanuallah University of Engineering & Technology (AUST), Dhaka, Bangladesh. Her research interests include Wireless multi-hop networks emphasizing on mathematical modeling, protocol design, and performance analysis.

Akib Jawad Nafis is currently pursuing his B.Sc in Computer Science and Engineering from Department of CSE, Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh. He is in his final year of undergraduate study. His research area and interest includes Network and Security, Operating System, Machine Learning, Deep Learning, and Bioinformatics.

Md. Sadiqur Rahman Sohel is currently pursuing his B.Sc in Computer Science and Engineering from Department of CSE, Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh. Currently, he is in his final year of undergraduate study. His research area and interest includes wireless ad-hoc networks, machine learning, cloud computing, and blockchain.

Salsabil Arabi is currently pursuing her B.Sc in Computer Science and Engineering from Department of CSE, Bangladesh University of Engineering and Technology (BUET), Bangladesh. She is in her final term of undergraduate study. Her research area and interest includes Network and Security, Algorithm Analysis, Deep Learning, and Cloud Computing.

Chowdhury Nawrin Ferdous is currently pursuing her second M.Sc in Computer Science and Engineering from Department of Mathematics and Computer Science, University of Lethbridge, Lethbridge, Alberta, Canada. She has completed her M.Sc in Computer Science and Engineering from Department of CSE, Bangladesh University of Engineering and Technology (BUET), Bangladesh. She has done her B.Sc in CSE from Military Institute of Science and Technology (MIST), Bangladesh. Her research area and interest includes Wireless network, Algorithms, Optimization.

1 Introduction

The emergent of Internet of Things (IoT) needs inter-connectivity among different heterogeneous devices ranging from complex smartphones to simple sensors. It requires out of the ordinary technologies and communication solutions to achieve this kind of inter-connectivity. Notably, most of the IoT devices such as smartphones, tablets, cameras, sensors, and other instrumentation are often well-equipped with wireless transceivers. As more and more IoT devices are being added, more opportunities are being created for wireless multi-hop communication technologies either on a regular or on an ad-hoc basis. As a result, the role of wireless multi-hop networks in the successful deployment of emerging IoT is essential in order to provide better connectivity level between objects, processes, services, and people.

A wireless multi-hop network can be thought of as a set of a several portable IoT devices which dynamically creates a network to establish connection among individual devices with no previous framework such as router or access point or any centralised system. The wireless ad-hoc network is suitable for a variety of application for its minimal configuration and quick deployment. Communicating with each other is quite difficult for mobile nodes. This is due to lack of power supply, inefficient use of channel and more. To overcome these problems, the nodes often depend on intermediate nodes to send and receive data over the network and forward packets on behalf of others.

The most general communication method in a network is **Broadcasting**. Broadcasting is the process of sending data to all the devices on the network simultaneously. Even when we are not broadcasting data, this method is specially effective for searching new paths for a destination and finding the distance of a specific node. But it is also the most intensive as a large number of messages are required.

Blind flooding is the simplest approach for broadcasting where every node in the network rebroadcast a message exactly once after being originated from any source. Such a naive memory less approach (surprisingly) guarantees reachability to all nodes even under high mobility. Unfortunately naive flooding causes too much redundant traffic, creates contention, and collision ultimately resulting into *broadcast storm problem* [1]. This problem can be reduced to some extent by creating a *virtual backbone* on top of actual topology. Once the backbone is created it is easy to run any traditional routing protocol on the overlay network created by this virtual backbone [2].

A good number of virtual backbone has been established to deal with the Broadcast Storm Problem. The most commonly used one is the Connected Dominating Set (CDS). A CDS is usually developed by representing the whole network as a graph. If graph $G = (V, E)$ represents the network, V is the set of all nodes in the network and E is the set of communication links among them. CDS is the subset D of the graph G if it asserts the following two conditions.

- All nodes in D are connected.
- Nodes that don't belong to D are 1-hop neighbour of at least one of the member of D.

Just forwarding by the members of CDS a packet can be transmitted efficiently throughout the network.

Many centralised and distributed algorithms [3, 2, 4] have already been formulated to construct CDS. Nevertheless all of these are destined to scale down packet re-transmission. There is a fact that, most of the CDS algorithms aim at decreasing the number of redundant packet generation, not at reducing contention problem. Contention implies race for shared resources. This term is used particularly in networks to portray the situation where at least two nodes are trying to transmit a message over a shared channel. In a wireless ad-hoc network, once the broadcasting of a message is initiated, contention can occur if two or more adjacent nodes are within the transmission range of each other and try to rebroadcast the message.

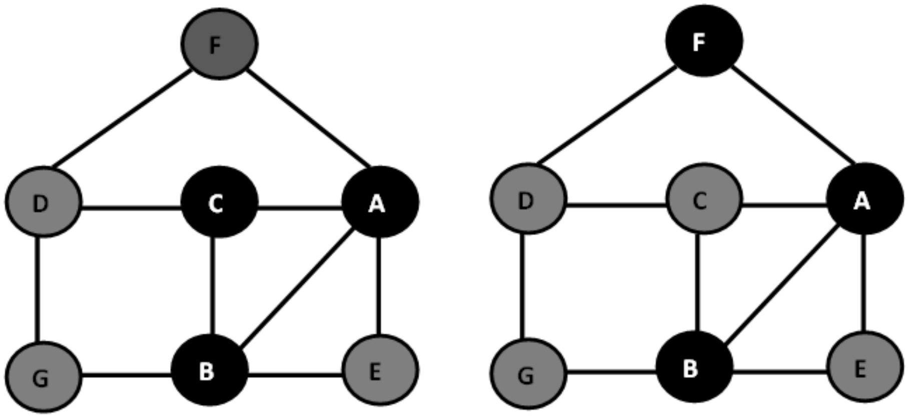
The major contributions of the paper are summarized below:

(a) In this paper, we mathematically analyze contention and demonstrate that, it is an element of a few system parameters, for example, **density of the devices** in the network and the **transmission ranges of the devices**.

(b) Additionally, we have proposed novel centralized and distributed algorithms for constructing CDS that minimize contention and consequently, the delay. The algorithms proposed in the previous works along the same direction aim at minimizing number of forwarding but neither contention nor delay.

(c) To develop CDS by utilising the complete topology data of an entire network, a centralised algorithm has been developed. As it is inconvenient to collect and gather the complete topology data, we have also presented a couple of distributed algorithms.

(d) The performance analysis of the centralised and distributed algorithms with regards to delay and number of forwarding nodes is further shown using ns-2 simulator under realistic settings. Experimental results show that an increase in size of the forwarding list by



(a) Constructing CDS with Contention (b) Contention free CDS Construction

Figure 1: Construction of CDS

0-5%, reduces the delay of the network up to 1-8% as the contention has been significantly reduced.

In a nutshell, this paper is arranged as follows. In Section 2, basis of this work has been explained. In Section 3, the state-of-the-art research works are presented. In Section 4 we mathematically analyse contention and lay out the foundation of this work. In Section 5, the crucial terminologies, considered state-of-art algorithms, and the methodology of the newly developed algorithms are discussed with examples. Section 6 represents the simulation and performance evaluation.

2 Working principle

A network is depicted by a graph $G-(V,E)$. Here, V is the collection of devices and E is the collection of connections among the devices. The set D which is a subset of V can form CDS if it has the characteristics stated below:

(i) if a node $v \in D$ then v is reachable from any other node in D.

(ii) For a node u, if u is a member of V then either u is a member of D or u is a neighbour of a node in D.

After constructing CDS, broadcasting is executed as pursues:- in the event that the broadcast initialiser belongs to CDS, every node will receive packet in due process of forwarding by the nodes in CDS. Since nodes that do not belong to CDS are adjacent to at least one of the node in CDS. The initialiser can broadcast packet to it's adjacent nodes in the event that it does not belong to CDS. Again, there is at least there is one node in CDS adjacent to that initialiser. Thus just forwarding by the nodes in CDS packet will reach to every node in the network.

The primary objective of this work is to build a CDS in such a way so that when a message is forwarded by the nodes in CDS, the contentions among the nodes are as minimum as possible. In other words, a CDS will be constructed that will keenly choose the nodes to diminish contention. Let us consider the instance delineated in Figure 1.

Here all possible CDSs are:

$$CDS = \{A, B, F\}, \{A, B, C\}, \{C, D, B\}, \{A, B, G\}, \{A, B, C, D\}$$

For this graph the minimum size CDSs are:

$$MCDS = \{A, B, G\}, \{A, B, F\}, \{A, B, C\}, \{C, D, B\}$$

And contention aware CDSs are:

$$CACDS = \{A, B, F\}, \{A, B, G\}, \{C, D, B\}$$

Contention occurs if two neighbouring nodes attempts to broadcast a packet subsequently because only one node can use the shared channel at a time. Consider the Figure 1(a), node A, B and C are in the forward list. After receiving a packet from node A, node B and C create contention while rebroadcasting the packet. But, if we select forwarding nodes intelligently we can reduce contention. In the scenario of Figure 1(b) A, B and F are selected as a forwarding nodes. After receiving a packet from node A, node B and F can rebroadcast packet in parallel at the same time. This is because node B and F are not within the transmission range of one another.

3 Related Works

A few methodologies have been proposed to diminish the *Broadcast Storm Problem*. By constructing virtual backbone for broadcasting, this problem can be reduced. Ephermides [5] first introduced such an idea of constructing connected dominating set (CDS) which acts as a virtual backbone. After that, numerous research have been performed where several methodologies have been proposed for constructing CDS which can be ordered into two categories: *centralised* and *distributed*.

CDS can be constructed in numerous ways. However, constructing minimum connected dominating set (MCDS) is NP-complete problem. Constructing a broadcast tree in which packet forwarding is minimum is similar but harder than MCDS. So, to construct a broadcast tree which is nearly optimal, several algorithms have been proposed using approximations and heuristics.

Such heuristic algorithm was first proposed by Guha and Khullar [6] which uses greedy approach to construct minimum connected dominating set (MCDS) [7] and Weakly connected dominating set (WCDS). Ruan et al. [8] proposed a new one-step greedy approximation with a non supmodular potential function and with a performance ratio 2 was proposed to construct a Minimum Connected Dominating Set(MCDS). In [9], authors propose a greedy algorithm for MCDS in unit-disk graphs based on Maximal Independent Set (MIS). Min et al. [10] propose to use a Steiner tree with the minimum number of Steiner nodes (ST-MSN). Das et al. [11] tried to improve it with a new degree-based multiple leaders initiated greedy approximation algorithm (PSCASTS) and improved Steiner-tree construction. Another work by Al-Nabhan et al. [12] proposed three centralised algorithms to construct CDS by dividing the set of nodes into two basic sets (core nodes and supporting nodes). Another improvement of CDS construction algorithm has been devised by Lou et al. [13]. In [14], the authors for the first time studied the MCDS construction under the discrete beeping model.

In real environment it is complex to collect the information of global topology. Hence, to develop CDS in multi-hop networks, distributed algorithms are more preferable. The distributed version of the Guha and Khuller's [6] algorithms was implemented by Das and Bharghavan [15] which has high message overheads. Though Wu and Li in [16] proposed an algorithm using two-hop neighbourhood information, any performance analysis was not provided. Self Pruning (SP), a reactive algorithm was proposed by Lim and Kim [17] where neighbourhood information is used directly to choose forwarding nodes. An improved probabilistic broadcast based on self pruning was devised here[18]. In another algorithm titled Dominant Pruning (DP) a node selects forwarding nodes from it's one-hop neighbour in such a way that a packet can reach to all it's two-hop neighbours. DP incurs more overhead compared to self-pruning as DP requires two hop neighbour information whereas self-pruning requires only one hop information. To reduce this overhead Nowak et al. provided a global queue pruning method[19]. This one also provides assurance of the delivery of the messages to every node in the network. Later on, several optimisations have been proposed to reduce the number of packet forwarding.

In particular, Anannya et al. [20] propose further reduction in number of forwarding using extended three-hop neighbourhood information. Rahman et. al. [21] propose an extension of dominant pruning to provide fault tolerance that will continue to work under unreliable environment. They reformulate the broadcasting problem from the set cover problem (as in DP) to set multi-cover problem. Shi et al.[22] provides a greedy algorithm to build a fault tolerant CDS modeled as a k-connected m-fold dominating set ((k,m) - CDS for short). However, their work also overlooked contention.

Though all the above mentioned algorithms aim to diminish overheads and number of forwarding, no algorithms have been designed to minimise *contention*. In this paper, we introduce new algorithms to construct "contention-aware" connected dominating set (CDS) to fill this notable gap. When contention is significantly reduced, broadcast message can propagate faster throughout the network experiencing less delay on a per hop basis.

4 Analysis of Contention

In this section, we develop an analytic model for determining *contention* in wireless multi-hop networks.

4.1 Contention Probability of a node

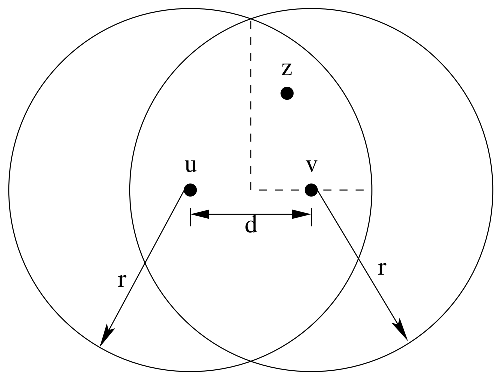


Figure 2: Intersection region of two nodes *u* and *v* located *x* distance apart.

Let us consider *n* number of nodes have been uniformly deployed over a rectangular area *A*. Therefore, the average node density becomes: $\mu = n/A$.

Consider the scenario illustrated in Figure 2 where node u has transmitted a broadcast packet and some hosts including node v has received the packet. If all these hosts try to rebroadcast contention may occur because two or more hosts around node u are likely to be very close to each other and the channel is shared wireless channel. Let us analyse the case of node v. Node *u* and node *v* are located *d* distance apart. Let, S_u and S_v denote the circular region covered by node *u*'s and node *v*'s transmission range, respectively. $S_{u\cap v}$ is the intersection area of the two circles S_u and S_v centered at two nodes *u* and *v*. When v would forward/rebroadcast the packet, another node z may contend with it if z is located in the area $S_{u\cap v}$. Thus contention probability clearly depends on the size of the intersection area $S_{u\cap v}$. Let us find this intersecting area as shown below.

A homogeneous system is assumed where each node in the network has the similar transmission range. Suppose *r* is the radious of S_u and S_v. We can define $S_{u\cap v}$ by the following simple equation:

$$|S_{u\cap v}| = INTC(d) = 4\int_{d/2}^{r} \sqrt{r^2 - x^2}dx \quad (1)$$

In another work [23] the derivation of the integration in Equation 1 has been shown. For better readability we present the derivation again in this paper. To get the result of integration we need to translate the Cartesian co-ordinates into polar co-ordinates by replacing $x = rsin\theta$. Therefore, $\theta = sin^{-1}(\frac{x}{r})$. The upper limit r becomes $sin^{-1}(\frac{r}{r}) = sin^{-1}(1) = \frac{\pi}{2}$. Similarly the lower limit $d/2$ becomes $sin^{-1}(\frac{d/2}{r}) = sin^{-1}(\frac{d}{2r})$. By substituting lower and upper limits derived for polar coordinates, Equation 1 becomes:

$$INTC(d) = 4\int_{\sin^{-1}(d/2r)}^{\pi/2} \left(\sqrt{r^2 - r^2\sin^2\theta}\right) r\cos\theta d\theta$$

$$= r^2 \times \left[\pi - 2\sin^{-1}\left(\frac{d}{2r}\right) - \sin 2\sin^{-1}\left(\frac{d}{2r}\right)\right] \tag{2}$$

When nodes u and v are x distance apart the intersection area can be obtained from Equation 2 by putting $d = x$ which becomes:

$$INTC(x) = r^2 \times \left[\pi - 2\sin^{-1}\left(\frac{x}{2r}\right) - \sin 2\sin^{-1}\left(\frac{x}{2r}\right)\right] \tag{3}$$

Using Equation 3, the intersection area $S_{u\cap v}$ becomes:

$$\begin{aligned}
|S_{u\cap v}| &= INTC(x) \\
&= r^2 \times \left[\pi - 2\sin^{-1}\left(\frac{x}{2r}\right) - \sin 2\sin^{-1}\left(\frac{x}{2r}\right)\right] \\
&= \pi r^2 - r^2\left[2\sin^{-1}\left(\frac{x}{2r}\right) + \frac{x}{r}\sqrt{1 - \frac{x^2}{4r^2}}\right] \\
&= \pi r^2 - r^2\left[2\sin^{-1}\left(\frac{x}{2r}\right) + \frac{x}{2r^2}\sqrt{4r^2 - x^2}\right]
\end{aligned} \tag{4}$$

Upon receiving the broadcast packet from node *u*, the node *v* will contend with its neighbour(s) only if it has at least one node in the *intersection* area of u and v. If the intersection area is empty, then no neighbours of *v* will contend when node *v* will forward the packet. Therefore, contention probability of node *v*, $P_C(x)$ becomes the probability that there exist at least one node in the intersection area $S_{u\cap v}$ of *u* and *v* node pair where *x* is the distance between two nodes.

The probability that a node exists in an area B inside the deployment area A (Under uniform distribution) is [24]:

$$P_B = \frac{|B|}{|A|}$$

Hence, the probability that a node exists in the intersecting area *INTC(x)* within the deployment area *A* is:

$$P_\delta = \frac{INTC(x)}{A} = \frac{INTC(x) \times \mu}{n} \tag{5}$$

The probability $P_k(INTC(x))$ that exactly *k* nodes exist in the intersection area is:

$$P_k\left(INTC(x)\right) = \binom{n-2}{k} P_\delta^k \times (1 - P_\delta)^{n-2-k} \tag{6}$$

In Equation 6, we have used *n-2* instead of *n* as we want to exclude *u* and *v*. For small P_δ and large *n*, the Poisson distribution can be used to approximate the binomial distribution [24], [25]. So,

$$P_k\left(INTC(x)\right) = \frac{(nP_\delta)^k \times e^{-nP_\delta}}{k!} \tag{7}$$

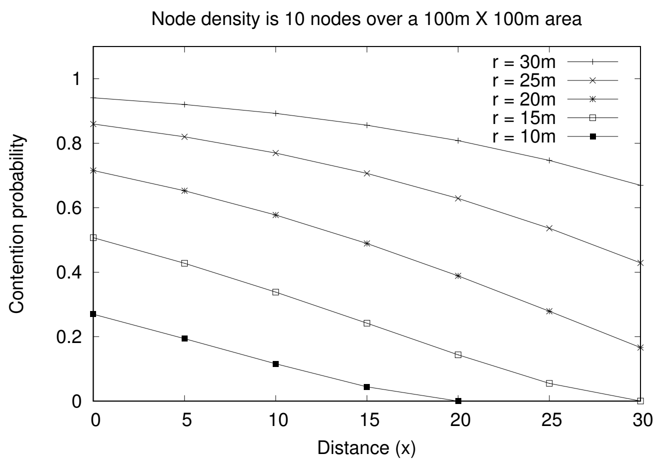


Figure 3: Effect of distance on contention probability

The contention probability becomes the probability that there exists at least one node in the intersection region, *INTC(x)*.

$$\begin{aligned} P_C(x) &= \sum_{k=1}^{n} P_k\left(INTC(x)\right) \\ &= \sum_{k=1}^{\infty} \frac{(nP_\delta)^k \times e^{-nP_\delta}}{k!} \\ &= 1 - e^{-\pi\mu r^2 + \mu r^2\left\{2\sin^{-1}\left(\frac{x}{2r}\right) + \frac{x}{2r^2}\sqrt{4r^2-x^2}\right\}} \end{aligned} \tag{8}$$

The detail derivation of Equation 8 is given in the Appendix (see Equation 12).

4.2 Analysis

In Equation 8 the contention probability is function of three network parameters, node distance (x), maximum communication range (r) and node density (μ). When $x = r$, the distance from the source is largest and from Equation 8 we get:

$$\begin{aligned} P_C(x) &= 1 - e^{-\pi\mu r^2 + \mu r^2\left\{2\sin^{-1}\left(\frac{x}{2r}\right) + \frac{x}{2r^2}\sqrt{4r^2-x^2}\right\}} \\ &= 1 - e^{-\pi\mu r^2 + \mu r^2\left\{2\sin^{-1}\left(\frac{r}{2r}\right) + \frac{r}{2r^2}\sqrt{4r^2-r^2}\right\}} \\ &= 1 - e^{-\pi\mu r^2 + \mu r^2\left[\frac{\pi}{3} + \frac{\sqrt{3}}{2}\right]} \\ &= 1 - e^{-\mu r^2\left[\frac{2\pi}{3} + \frac{\sqrt{3}}{2}\right]} \end{aligned} \tag{9}$$

When the distance from the source is smallest with $x = 0$, we get:

$$\begin{aligned} P_C(x) &= 1 - e^{-\pi\mu r^2 + \mu r^2\left\{2\sin^{-1}\left(\frac{x}{2r}\right) + \frac{x}{2r^2}\sqrt{4r^2-x^2}\right\}} \\ &= 1 - e^{-\pi\mu r^2 - \mu r^2\left\{2\sin^{-1}\left(\frac{0}{2r}\right) + \frac{0}{2r^2}\sqrt{4r^2-0^2}\right\}} \\ &= 1 - e^{-\pi\mu r^2 - \mu r^2\left\{2\sin^{-1}(0) + 0\right\}} \\ &= 1 - e^{-\pi\mu r^2} \end{aligned} \tag{10}$$

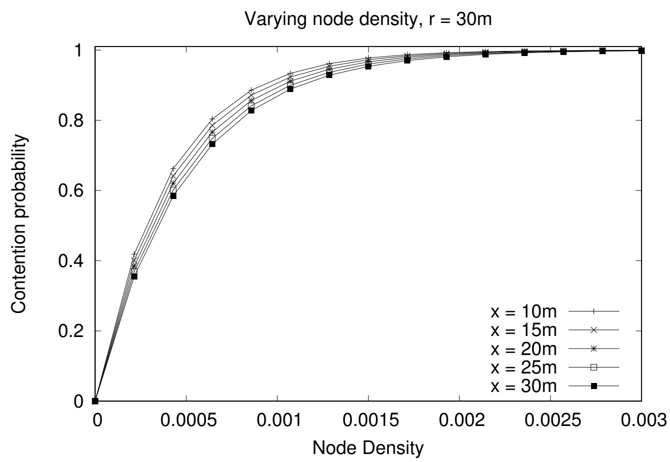


Figure 4: Effect of node density on contention probability

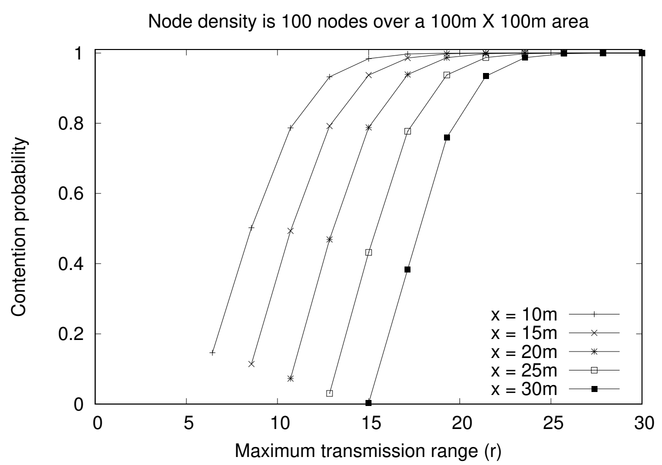


Figure 5: Effect of maximum transmission range on contention probability

Thus the probability of contention varies between $1 - e^{-\mu r^2\left[\frac{\pi}{3}+\frac{\sqrt{3}}{2}\right]}$ to $1 - e^{-\pi\mu r^2}$. Consequently, the maximum value $1 - e^{-\pi\mu r^2}$ increases with the increase of μ and r.

Figure 3 shows the effect of node distance (x) on contention probability. The contention probability decreases when the distance is increased. The main reasoning is as follows. If we consider Figure 2, the size of the intersection area decreases with the distance from the source. Thus, more distant nodes from the source have smaller intersection area and less number of nodes is probable to fall in this intersection area under uniform distribution. Thus, less number of nodes would contend with the node causing lower contention probability $P_C(x)$.

Figure 4 shows node density (μ) affects contention probability. If we increase node density, more nodes will exist in the intersection area. In this way the probability of contention increases exponentially if node density increases.

Finally, Figure 5 shows the effect of maximum communication range (r) on contention probability. The contention probability exponentially increases with maximum communication range r. Large transmission range covers more nodes in the intersection area and the contention probability increases accordingly.

In summary, contention probability drastically increases with the increase in node density and maximum communication range. Also it is inversely proportional to the distance. Therefore, every node of a neighbour can not be equally considered for forwarding tasks and someone needs to select them intelligently. This phenomena motivates us to develop the contention aware CDS construction algorithms, both centralised and distributed versions which we present next.

5 Contention Aware CDS construction Algorithms

In this section, we introduce three algorithms:- first, a centralised algorithm, secondly, a distributed algorithm and finally, a hybrid algorithm which is inherently distributed in nature. Global topology information is used to construct CDS in centralised algorithm whereas only two-hop neighbourhood information is used to generate forwarding list in both distributed algorithms. As distributed algorithms are more likely to be used in practical scenarios, we are providing two distributed algorithms.

Before going into details, at first, we provide some background information and definitions next.

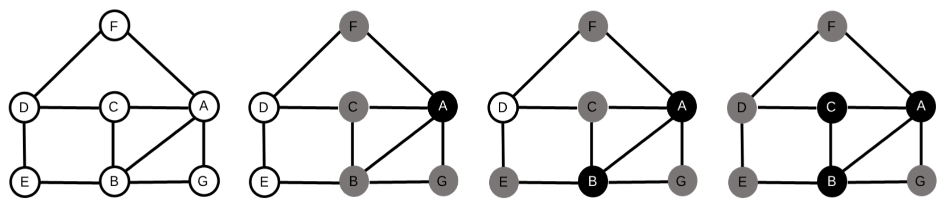
5.1 Preliminaries

We will represent the ad-hoc network with a graph G(V, E). Here V is the set of nodes in the network and E is the edges that indicate connectivity between two nodes. The set of all the one-hop neighbours (adjacent nodes) of node u is defined by $N(u)$ and the set of all the nodes that are at most two-hop away from node node u is represented by $N(N(u))$. The nodes are that selected from the adjacent nodes of u to be the forwarding nodes are defined by F_{u}. B_{u} represents all the one-hop neighbours of node u which are possible candidates to be the selected as forwarding nodes by node u and finally, U_{u} represents the set of all the uncovered two-hop neighbours of node u. While constructing the forwarding list F_{u}, node u selects some nodes from B_{u} set that will cover the nodes listed in U_{u} set.

We have used MCDS [6] and Dominant Pruning [17] as background approach to assess the new proposed methodologies.

(i) MCDS Construction Algorithm: MCDS construction algorithm uses colouring method to find out the MCDS from a graph. Each node of the graph is given colour white at the beginning of this algorithm. Among all the white node, a node is selected which is connected to most of the nodes, in other word, the node which has the maximum cardinality is selected first and it is coloured black. The adjacent nodes of that selected node is coloured gray. The next node of this process is selected from the gray ones.

The selection is done on the basis of the highest quantity of white adjacent nodes. The selected one is kept in the list of black nodes. This selection procedure continues to run as long as any white node exists. The MCDS is consist of all the black nodes.



(a) Each node is coloured WHITE

(b) Node A has been chosen and coloured BLACK

(c) Node B has been chosen and the process continues

(d) Finally node A,B,C constructs the MCDS

Figure 6: Construction of MCDS

The above described algorithm is illustrated in the following example. Figure 6(a) represents a network which has 7 nodes (A,B,C,D,E,F,G). Node A is selected and coloured black as it has maximum cardinality. All its one hop neighbours (node B,C,F,G) are coloured gray (shown in Figure 6(b)). Among the gray nodes, the next step is to find the node which one has the maximum number of white neighbours. Nodes B, C and F each has one white neighbour and G has no white neighbour. Suppose, node B is chosen randomly and coloured black. Its one hop neighbour (node E) is coloured gray (shown in Figure 6(c). Now from all the gray nodes we have node C and F having maximum number of white neighbour which is 1. Suppose, C is chosen and colored black and its neighbouring node (node D) is colored gray (shown in Figure 6(d)). So,

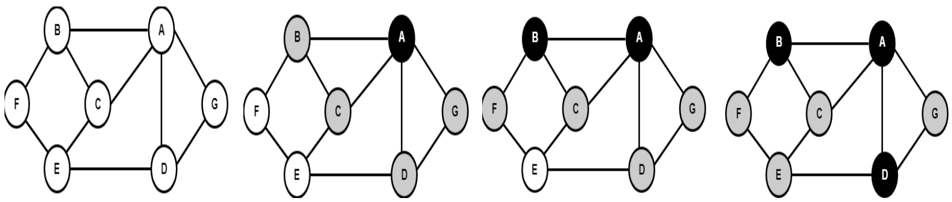
$$MCDS = \{A, B, C\}$$

.

(ii) Dominant Pruning: Let us discuss the procedure of creating the forwarding list that is followed by dominant pruning. Let us presume that, u sends a message to v. A *forwarding list (F_u)* is attached with the packet header. For a node v in *forwarding list*, it will create next forwarding list before rebroadcasting the packet. For constructing the new forwarding list, node u will need all the two-hop neighbours U_{v}) that are not listed. Node v selects a neighbour $p \in B_{\mathrm{v}}$ and $p \notin F_{\mathrm{v}}$ to cover the highest number of nodes in U_{v}, in other word, a node p is selected if $|N(p) \cap U_{\mathrm{v}}|$ is maximum among all the nodes in B_{v}. Next v added p in F_{v}. The set U_{v} is updates as $U_{\mathrm{v}} = U_{\mathrm{v}} - N(p)$. The process terminates when there remains no nodes to cover.

5.2 Centralised Contention Aware Connected Dominating Set Algorithm

We showed the centralised CACDS in Algorithm 1 . We have coloured the nodes white, gray and black at different situations. Initially, every node of the network are coloured white. The white nodes are stored into a set named $WhiteSet$. Among the nodes in $WhiteSet$, the node which has maximum number of adjacent nodes is chosen and stored in $BlackSet$ (shown in line 7-8 of Algorithm 1). Black nodes are the member of CDS. If several nodes have equal maximum cardinality, then a node is randomly selected from all them. All the one-hop neighbours of the selected node are coloured gray and placed in a set $GraySet$. The selected node as well as its one hop neighbouring nodes are discarded from $WhiteSet$ set. Among the nodes in $GraySet$, the algorithm looks for a node which has minimum



(a) Each node is coloured WHITE

(b) Node A has been chosen and coloured BLACK

(c) Node B has been chosen and the process continues

(d) Finally node A,B,D constructs the CACDS

Figure 7: Construction of centralised CACDS

number of black neighbouring nodes. These nodes are kept in $Candidates$ set (shown in line 19-24 of Algorithm 1). As we know, the black nodes are the member of CDS, so any node selecting from $Candidates$ set will surely minimise the probability of occurring contention. If there arise an occurrence of a tie, a node is chosen from the $Candidates$ set which one has the highest number of white neighbours. The chosen node is placed in $BlackSet$ as well as all of its white neighbours are coloured gray (shown in line 25-31 of Algorithm 1). This recursive selection procedure continues until no white node exists in the network. Finally the $BlackSet$ is our resultant $CACDS$.

The above described algorithm is illustrated in the following example. Figure 7(a) represents a network which has 7 nodes (A,B,C,D,E,F,G). The maximum cardinality of a node is 4 for this graph and A is such a node. So, as we described in the algorithm, node A is selected. Node A is coloured black and all its one hop neighbours (node B,C,D,G) are coloured gray (shown in Figure 7(b)). Among the gray nodes, the next step is to find the node which one has the minimum number of black neighbours. At this stage, all the gray nodes has only one black neighbour which is node A. As we know, in case of a tie, the algorithm search for a node who has maximum number of white neighbours. Here, node B, C and D each has one white neighbours where node G has no white neighbours. Suppose, node B is selected randomly and coloured black. Neighbouring node of node B, node F is coloured gray shown in 7(c). Now, we have left with only one white node which is node E. Node E can be covered by node C, D and F. Among these three nodes, node C now has maximum number of black neighbours (node A and B) where both node D and F has only one black neighbour. Hence, node C is no longer a candidate to be chosen to cover node E. In the Figure 7(d), node D is selected to cover node E and finally, node A, B and node D are the members of centralised CACDS. So,

$$CACDS = \{A, B, D\}$$

.

Complexity Analysis of Centralised CACDS: In this algorithm we have three sets:- $WhiteSet$, $BlackSet$ and $GraySet$. Initially, all the nodes are in $WhiteSet$. So, the outer $WhileLoop$ starting from line 11 will run in $O(V)$ times. This $While\ Loop$ ends on line 39 and it encloses three inner-loops. The maximum number of nodes in $GraySet$ is V-1. So, these three loops can run at most V-1 times in worst case. Hence, $O(V^2)$ is the run time complexity of Centralised CACDS.

Algorithm 1 Centralised CACDS

INPUT: G(V,E)
RESULT {CACDS}
$BlackSet = \emptyset$;
$GraySet = \emptyset$;
$WhiteSet = V$;
$CACDS = \emptyset$;
Find the node v $\in$ V with maximum cardinality max(degree(v));
$BlackSet = BlackSet \cup \{v\}$;
$GraySet = N(v) - BlackSet$;
$WhiteSet = WhiteSet - N(v)$;
while $WhiteSet$ is not empty **do**
  $WhiteMax = 0, BlackMin = \|V\|, Candidates = \emptyset$;
  for all $s \in GraySet$ **do**
    $BlackCount = \|BlackSet \cap N(s)\|$;
    if $BlackCount < BlackMin$ **then**
      $BlackMin = BlackCount$;
    end if
  end for
  for all node $s \in GraySet$ **do**
    $BlackCount = \|BlackSet \cap N(s)\|$;
    if $BlackCount = BlackMin$ **then**
      $Candidates = Candidates \cup \{s\}$;
    end if
  end for
  for all $s \in Candidates$ **do**
    $WhiteCount = \|WhiteSet \cap N(s)\|$;
    if $WhiteCount > WhiteMax$ **then**
      $WhiteMax = WhiteCount$;
      $selected = s$;
    end if
  end for
  if $WhiteMax > 0$ **then**
    $BlackSet = BlackSet \cup \{selected\}$;
    $WhiteSet = WhiteSet - N(selected)$;
    $GraySet = GraySet \cup (N(selected) - BlackSet)$;
  else
    $GraySet = GraySet - Candidates$;
  end if
end while
$CACDS = BlackSet$;

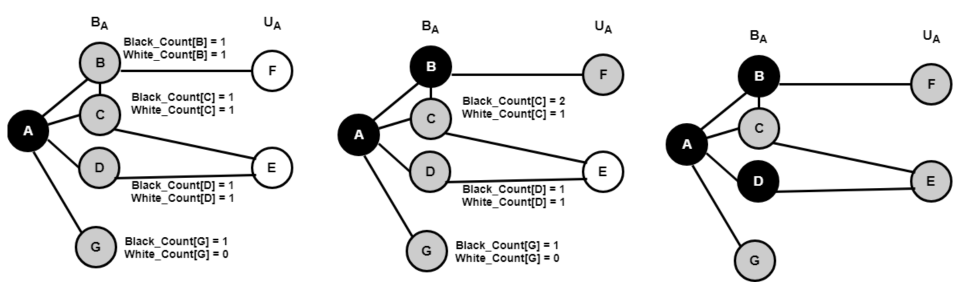
5.3 Distributed Contention aware Connected Dominating Set (Distributed CACDS) Algorithm

The algorithm is designed assuming that, every node is provided with the information of its 2-hop neighbours. If node v receives a packet from node u and if it is already in the *forwarding list* of u, it creates its own *forwarding list*. To create the *forward list* (F_{v}), it discovers all of its undiscovered two-hop neighbours (U_{v}) by using its one-hop neighbours (B_{v}). We can calculate U_v and B_v from the following formulas:

$$U_{\mathrm{v}} = N(N(v)) - N(v) - N(u)$$

$$B_{\mathrm{v}} = N(v) - N(u)$$

In Algorithm 2, we have described our Distributed CACDS algorithm . For this algorithm, each node $u \in B_{\mathrm{v}}$ needs to store its neighbouring nodes in B_{v} which are already in the *forwarding list*. An array of nodes named $BlackCount$ is used for this purpose. Initially, for all the nodes, all elements of $BlackCount$ array is kept one. We sort the nodes with minimum $BlackCount$ among all nodes in B_{v} and they are stored into a $Candidates$ set (shown in line 14-18 of Algorithm 2). In each iteration, we select a node s from the $Candidates$ set that gives maximum number of nodes coverage in U_{v}. Here, $\|N(s) \cap U_{\mathrm{v}}\|$ is maximised (shown in line 19-31 in Algorithm 2). Node s is added in the forwarding list F_{v} of v. The value of $BlackCount$ of the nodes in B_{v} neighbour of s are also increased by one (shown in line 33-37 of Algorithm 2). U_{v} and B_{v} are updated respectively by $U_{\mathrm{v}} - N(s)$ and $B_{\mathrm{v}} - s$. The loop breaks when all nodes in U_v is covered or set B_v reaches saturation and no change is possible.



(a) Connection of node A with B_A and U_A (b) Node B is chosen for F_A (c) Finally Node D is chosen for F_A

Figure 8: Generating Forwarding List of node A.

Let us consider an example in Figure 7. Here, node A is initiating the broadcast. According to the algorithm, we need to make the *forward_list* of node A. Let us consider Figure 8 for better which is the redrawing of Figure 7. Node B, C, D and G are stored in B_A, and node E and F are in set U_A. If we want select forwarding list of node A according to Dominant Pruning algorithm, nodes B and C will be selected. As node B and C are within their transmission range contention will occur at the time of rebroadcasting the packet by node B and C.

Our algorithm is designed to minimise this contention. if node B is selected as a forwarding node by node A so that node F is covered, our algorithm will choose node D to cover node E. Thus the algorithm avoids contention. Finally, the forwarding list of node A will include node B and D. Hence, the forwarding list of A, $F_{\mathrm{A}} = \{B, D\}$. The complete process is shown in Figure 8.

Algorithm 2 Forward list creation of a node v

$F \leftarrow v$
$F_{\mathrm{v}} = \emptyset$;
$size = 0$;
for all $u \in B_{\mathrm{v}}$ **do**
  $BlackCount[p] \leftarrow 1$;
end for
while U_{v} is not empty or B_v does not change **do**
  $max = 0$, $min = \|V\|$, $Candidates = \emptyset$;
  for all $w \in B_{\mathrm{v}}$ **do**
    if $BlackCount[w] < min$ **then**
      $min = BlackCount[w]$;
    end if
  end for
  for all $w \in B_{\mathrm{v}}$ **do**
    if $BlackCount[w] = min$ **then**
      $Candidates = Candidates \cup \{w\}$;
    end if
  end for
  for all $s \in Candidates$ **do**
    for all $t \in U_{\mathrm{v}}$ **do**
      if $t \in N(s)$ **then**
        $WhiteCount[s] = WhiteCount[s] + 1$;
      end if
    end for
  end for
  for all $w \in Candidates$ **do**
    if $WhiteCount[i] > max$ **then**
      $max = WhiteCount[i]$;
      $s = w$;
    end if
  end for
  if $max > 0$ **then**
    $F_{\mathrm{v}}[size] = \{s\}$;
    $size = size + 1$;
    for all $y \in (B_{\mathrm{v}} \cap N(s))$ **do**
      $BlackCount[y] = BlackCount[y] + 1$;
    end for
    $U_{\mathrm{v}} = U_{\mathrm{v}} - N(s)$;
    $B_{\mathrm{v}} = B_{\mathrm{v}} - \{s\}$;
  end if
end while

Previous node (u)	**Current node** (v)	$B_{\mathbf{v}}$	$U_{\mathbf{v}}$	$F_{\mathbf{v}}$
$\emptyset$	A	{B,C,D,G}	{E,F}	{B,D}
A	B	{F}	{E}	{F}
A	D	{E}	{F}	{E}
B	F	{E}	{D}	{E}
D	E	{C,F}	{B}	{C}
E	C	{A,C}	{G}	{A}

Table 1 Illustrating Distributed CACDS algorithm by showing Forwarding lists creation process for the scenario of Figure 7(a) (Assume node A has initiated the broadcast)

.

Complexity Analysis of Distributed CACDS: The set B_v contains all the adjacent nodes of v. Assume, Δ is the maximum degree of the graph. So, the size of B_v is at most Δ. Initially, set U_v contains the nodes that are adjacent from each node of B_v. Thus, the maximum size of U_v is Δ^2. Hence, the run time complexity of creating $forwarding\_list$ of a node is $O(\Delta^3)$

5.4 Hybrid Distributed Contention Aware Connected Dominating Set

The ultimate objective is to find a CDS which can provide a lower number of forwarding and smaller delay. The cardinality (Number of elements in a set) of CDS constructed from CACDS/DCACDS is high. As the number of nodes within the deployment area goes up, the total number of forwarding also drastically increases. Consequently, it takes more time to finish the forwarding of all nodes which adversely affects broadcast delay in the network. Our simulation result shows that when the number of nodes becomes higher than 70 the delay performance of DCACDS degrades (please see Figure 14). For this reason, a third hybrid heuristic has been constructed which combine the benefit of both DP and distributed DCACDS. The Hybrid Distributed Contention Aware Connected Dominating Set (HDCACDS) is shown in Algorithm 3.

Algorithm 3 Creation of forwarding list of node v with HDCACDS

$Forwarding\_node \leftarrow v$

$F_{\text{v}} = \emptyset$

$F_{\text{dp}} \leftarrow$ Created Forwarding list of node v using Dominant Pruning (DP) algorithm.

$F_{\text{dcacds}} \leftarrow$ Created Forwarding list of node v using Distributed CACDS

if $(|F_{dp}| >= |F_{dcacds}|)$ **then**

  $F_v = F_{dcacds}$

else

  $F_v = F_{dp}$

end if

In the hybrid algorithm both DP and distributed CACDS have been used to build two forwarding lists of a node. Then the node selects the final forwarding list using the following two rules.

- If the cardinality of the forwarding list created using DP is greater than the cardinality of the one created by distributed CACDS the distributed CACDS over DP is chosen.
- Else the forwarding list created by DP is chosen

Although the hybrid algorithm might face some contention when DP is chosen over distributed CACDS, there is a significant improvement in minimising delay and the average number of forwarding.

6 Experimental Results

In this section, we provide elaborated insight into exhibitions of our proposed algorithms with simulation results. We have a tendency to conjointly offer a performance comparison of the planned algorithms with other best in class algorithms.

6.1 Simulation Environment

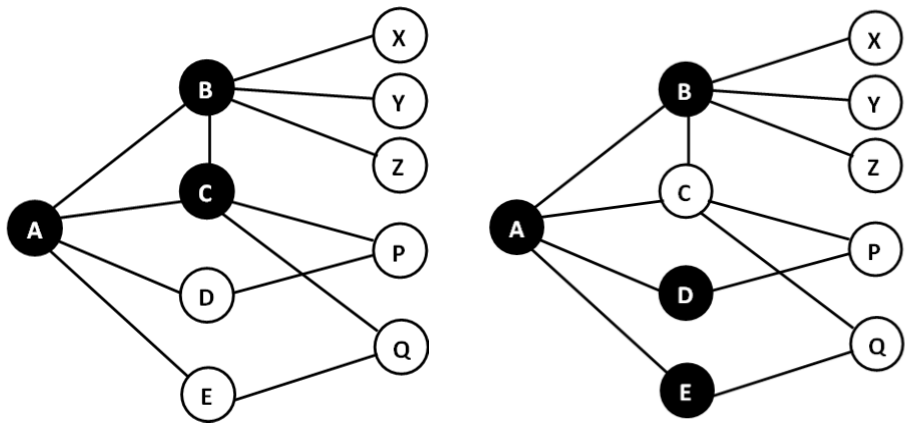
To assess the execution, haphazardly set out networks of 20-400 nodes over an inflexible 625m X 625m square unit locale have been imitated in Network Simulator version 2 (NS-2.) The transmission span is constrained between 175m to 275m. We need to keep our locale little, because dense networks suffer more in terms of contention and delay. For each scenario, 10 distinct networks of different connection between nodes are generated. Thereupon, we have selected mean estimation of the number of forwarding nodes and the measure of delay to assess the viability of the designed algorithms.

6.2 Performance Metrics

At this stage, we are adopting two terminologies to measure performance of these CDS construction algorithms. Elaborate definitions of these terminologies and reasons behind adopting them is defined below.

(i) Number of forwarding nodes: During the broadcast of a packet a node might forward or purge the packet. A forwarding node is the one which forwards a packet instead of purging it. Cardinality of a set of forwarding nodes is defined as number of forwarding nodes. Thereupon, number of nodes required to complete the transmission of a packet throughout the network in the simulator(NS-2) is interpreted as the number of forwarding nodes. For centralised algorithms, It is identical to the extent of the Connected Dominating Set. In order to minimise contention, nodes with highest number of 1-hop neighbour in the network may get selected to forward the packet. Thereupon, With the purpose of subduing contention, there might be a slight increase in number of forwarding nodes. In Figure 9 an example of such scenario is illustrated. While running traditional MCDS algorithm, number of forwarding nodes is 3. But to reduce contention, number of forwarding nodes increases to 4 in Figure 9(b).

Number of forwarding nodes is chosen as a performance criteria because it is a good indicator of how many transmissions are needed to broadcast a packet throughout the network. Number of transmissions are directly related to the efficient usage of bandwidth.



(a) CDS to minimise redundancy (b) CDS to minimise contention

Figure 9: Different CDS for Different Goals

(ii) Delay: Delay is the time required to complete the network- wide broadcast of a packet. Which means delay is the elapsed time between the generation of the packet and the time the last host finishing it's rebroadcasting. In our example scenario of 9, if node A initiates a broadcast packet at time t_1, node B forwards it at time t_2 and node C forwards it at time t_3 then delay is measured as:

$$delay = \begin{cases} t3 - t1, & \text{if } t3 > t2 \\ t2 - t1, & \text{if } t2 > t3 \end{cases} \tag{11}$$

When several nodes contend to access a common channel for transmission, one exclusive node get holds of and all other nodes postpone transmission and wait for the channel to be free. Which results in increment of delay. Therefore, average delay is a good indicator of contention in the network.

Delay is our another performance criteria because delay indicates how quickly can we complete broadcast of a packet. Minimum delay is always preferred for efficient broadcast.

6.3 Experimental Results on the Example Scenario

Initially we simulated the scenario shown in Figure 1. After-effects of simulation is exhibited in Table 2. From the table it is clearly evident that the two centralised algorithms MCDS and CACDS generate the same number of forwarding although the delay of CACDS is much less compared to MCDS. This happened because CACDS has less contention among it's chosen forwarding nodes compared to the chosen nodes in MCDS. When contention is low the nodes get quicker access to the shared channel and need less time disseminate a broadcast packet. Although the distributed algorithm DP has less number of forwarding nodes in contrast to Distributed CACDS and HDCACDS, when it comes to degree of latency DP is not on par with its other two counterparts.

In Table 2, the performances of distributed CACDS and HDCACDS are exactly the same. This is because distributed CACDS and HDCACDS generate exactly the same forwarding list for this special example scenario of 7 nodes. As the forwarding nodes are the same the simulation exhibits similar delay and requires same number of forwarding.

Performance Metrics	MCDS	CACDS	DP	Distributed CACDS	HDCACDS
Delay(ms)	1.7	1.26	2.83	2.63	2.63
Avg. forwarding nodes	3	3	4.91	5	5

Table 2 Performance on the scenario of Figure 1

6.4 Experimental Results on Random scenarios

At this stage, we are comparing performance of our algorithm with previous works in random scenarios. We are comparing performance both in sparse and dense scenarios.

6.4.1 Performance based on the Number of forwarding

Firstly, we scrutinise average number of forwarding in these haphazardly deployed networks. To clearly scrutinise the outcomes, graphs of centralised and distributed algorithms are presented solely. Performance of centralised and distributed algorithms is disclosed in Figure 10 and 11.

In figure 10(a) and 10(b) performance is studied over various transmission range. The number of nodes has been set to 40. The number of forwarding nodes exponentially decreases with the increment of the transmission range. Because with the broad transmission range the network diameter drastically reduces and a fewer number of re-transmissions are good enough to transmit packets to all nodes throughout the network. Accordingly, size of the CDS becomes very small which causes a lesser number of forwarding.

In centralised CACDS and MCDS the number of forwarding nodes is almost alike. In CACDS the number of forwarding nodes is slightly higher than in MCDS.

In distributed environment (Figure10(b)), the distributed CACDS needs 1-5% additional transmissions which can be considered insignificant. Nevertheless, it is noticeable that HDCACDS generates the same number of transmissions as DP.

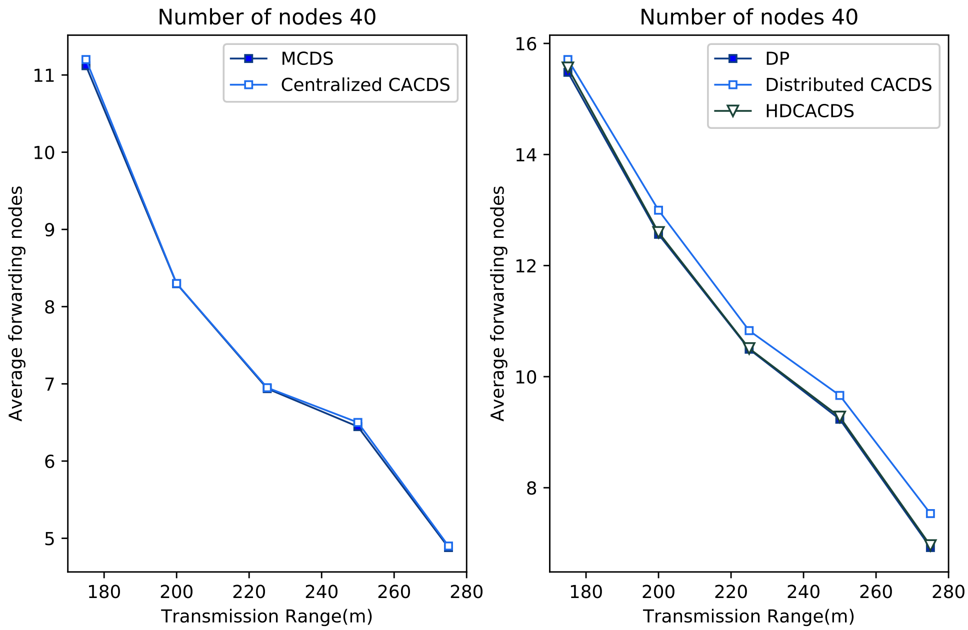
Figure 11(a) and 11(b) shows the required number of transmissions for various number of nodes. Transmission range is kept bounded at 250m. With the increment of number of nodes in the network, number of forwarding nodes boosts. For Centralised algorithms when all out number of nodes is greater than 200, the number of forwarding nodes falls off slightly. In this case network becomes dense. As the number of nodes goes up, CDS is also being larger. It is clear from the graph that in case of the number of forwarding centralised algorithms differs by only a slight amount.

Be that as it may, among distributed algorithms, DP performs best incurring the smallest number of forwarding. The distributed CACDS performs worst requiring the largest number of forwarding because of considering contention free CDS. The number of forwarding in HDCACDS is slightly higher than that of DP but lower than that of distributed CACDS.

6.4.2 Performance based on average delay

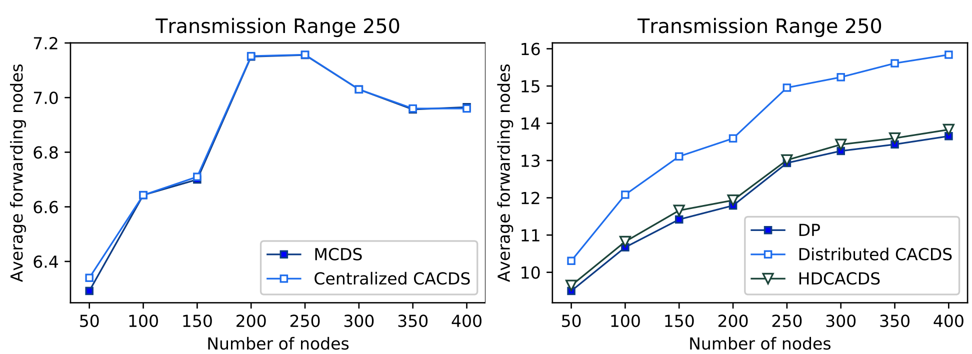
Figure 12, Figure 13, Figure 14 present the broadcast latency of sparse and dense networks for the centralised and distributed algorithms.

Figure 12 demonstrates the difference of delay among centralised and distributed algorithms. 40 nodes are taken to asses the performance. Transmission range is varied over the range of 180m-280m. In Figure12, for the centralised algorithms it is noticeable that the curve of the centralised CACDS goes below the MCDS curve. Therefore, the centralised CACDS reduces delay from MCDS even though the performance gap is very small. On the



(a) Centralised Algorithms (b) Distributed Algorithms

Figure 10: Performance of centralised and distributed algorithms for the number of transmissions required for various ranges of transmission



(a) Centralised Algorithms (b) Distributed Algorithms

Figure 11: Performance of centralised and distributed algorithms for the number of transmission required for various number of nodes

other hand, in distributed algorithms, HDCACDS performs best by incurring least amount of delay. Distributed CACDS performs slightly worse than HDCACDS while DP generates largest delay among all three distributed algorithms.

In the scenario appeared in Figure 13 and Figure 14 we are comparing the performance of centralised and distributed algorithms respectively for various number of nodes from 20 to 400. Transmission range is bounded at 250m. For better visualisation, we have plotted

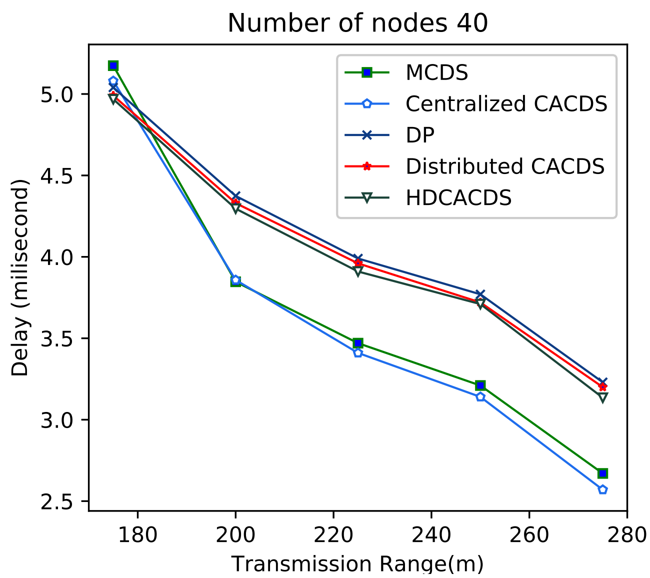
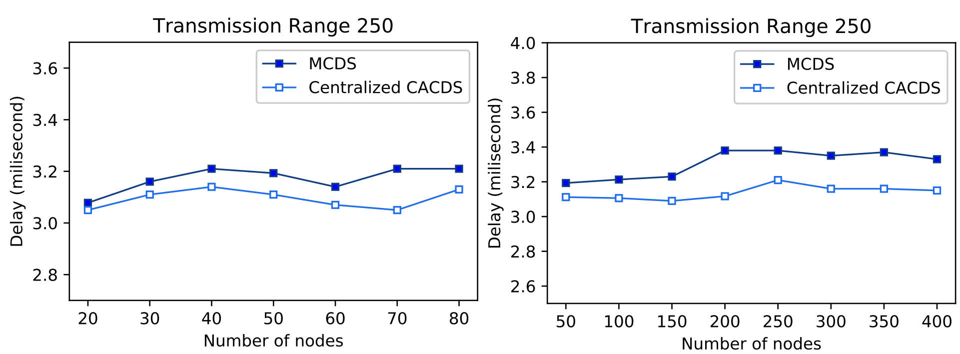


Figure 12: Delay of centralised and distributed algorithms for various transmission range

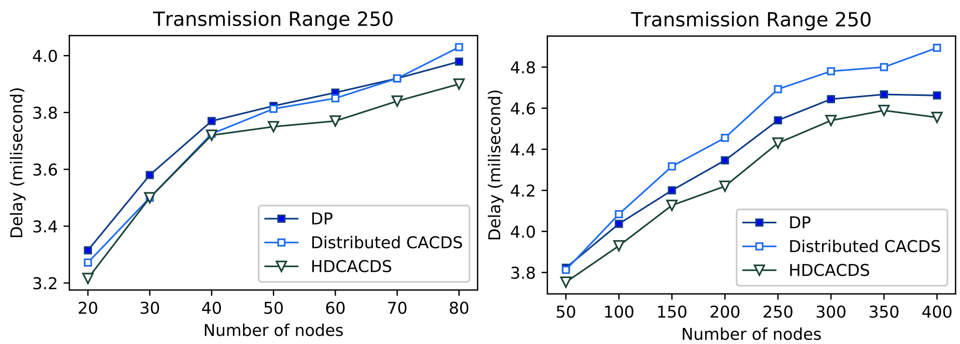


(a) Sparse networks (b) Dense networks

Figure 13: Performance of centralised algorithms with regards to delay for various number of nodes

two graphs for each case. In one plot we have shown sparse networks containing 20 to 80 nodes with an increase of 10 nodes at each step within the same deployment area 625m by 625m. In the other plot, we made networks denser by setting different number of nodes from 50 to 400 within the same region.

In Figure 13, performance of centralised algorithms are compared for sparse scenarios in (a) and dense scenarios in (b). The average delay of the proposed centralised CACDS is $1-8\%$ less than the delay incurred by the MCDS algorithm. The difference in delay becomes larger in dense networks (i.e., scenarios with number of nodes $50-400$) as shown in Figure 13(b).



(a) Sparse networks (b) Dense networks

Figure 14: Performance of distributed algorithms with regards to delay for various number of nodes

In Figure 14, performance of decentralised algorithms are compared. Figure14(a) and Figure14(b) shows comparison in sparse networks and dense networks respectively. Here, delay is measured against various number of nodes. In both scenarios our designed hybrid algorithm HDCACDS outperforms both DP and Distributed CACDS with regards to small delay. The average delay of the HDCACDS is $2 - 5\%$ less than the average delay in DP and $0 - 7\%$ than the average delay in Distributed CACDS. In dense scenarios, the difference in delays among three distributed algorithms is larger than sparse scenarios.

In summary, the proposed centralised CACDS outperforms MCDS with regards to the number of forwarding and delay for both sparse and dense networks. Among three distributed algorithms, HDCACDS exhibits the best results with regards to delay with the slight increase in number of forwarding compared to DP which is negligible. In sparse networks distributed CACDS performs almost similar to HDCACDS in terms of delay, although the difference is visible in dense networks.

7 Conclusion

In this paper, we show the mathematical analysis of contention probability of a node participating in broadcast communication. A new approach for developing CDS has also been developed to minimise contention. The centralised and distributed version of the CACDS algorithm has been designed as well as a new HDCACDS algorithm. A comprehensive simulation results have been presented to analyse the behaviour of the developed algorithms. The performance of the newly developed algorithms is better than the others in terms of reducing the contention though the number of forwarding nodes increases slightly. With Centralised CACDS and HDCACDS we have reduced the amount of delay.

References

[1] Yu-Chee Tseng et al. “The broadcast storm problem in a mobile ad hoc network”. In: *Wireless networks* 8.2 (2002), pp. 153–167.

[2] Sergiy Butenko et al. "A new heuristic for the minimum connected dominating set problem on ad hoc wireless networks". In: *Recent developments in cooperative control and optimization* 3 (2004), pp. 61–73.

[3] Jasaswi Prasad Mohanty, Chittaranjan Mandal, and Chris Reade. "Distributed construction of minimum connected dominating set in wireless sensor network using two-hop information". In: *Computer Networks* 123 (2017), pp. 137–152.

[4] Jasaswi Prasad Mohanty et al. "Construction of minimum connected dominating set in wireless sensor networks using pseudo dominating set". In: *Ad Hoc Networks* 42 (2016), pp. 61–73.

[5] Anthony Ephremides, Jeffrey E Wieselthier, and Dennis J Baker. "A design concept for reliable mobile radio networks with frequency hopping signaling". In: *Proceedings of the IEEE* 75.1 (1987), pp. 56–73.

[6] Sudipto Guha and Samir Khuller. "Approximation algorithms for connected dominating sets". In: *Algorithmica* 20.4 (1998), pp. 374–387.

[7] David Lichtenstein. "Planar formulae and their uses". In: *SIAM journal on computing* 11.2 (1982), pp. 329–343.

[8] Lu Ruan et al. "A greedy approximation for minimum connected dominating sets". In: *Theoretical Computer Science* 329.1-3 (2004), pp. 325–330.

[9] Xiuzhen Cheng, Min Ding, and Dechang Chen. "An approximation algorithm for connected dominating set in ad hoc networks". In: *Proc. of International Workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor, and Peer-to-Peer Networks (TAWN)*. Vol. 2. 4. 2004.

[10] Manki Min et al. "Improving construction for connected dominating set with Steiner tree in wireless sensor networks". In: *Journal of Global Optimization* 35.1 (2006), pp. 111–119.

[11] A. Das et al. "An improved greedy construction of minimum connected dominating sets in wireless networks". In: *2011 IEEE Wireless Communications and Networking Conference*. Mar. 2011, pp. 790–795. DOI: 10.1109/WCNC.2011.5779233.

[12] Najla Al-Nabhan et al. "Three connected dominating set algorithms for wireless sensor networks". In: *International Journal of Sensor Networks* 21.1 (2016), pp. 53–66.

[13] Chuanwen Luo et al. "A New Greedy Algorithm for Constructing the Minimum Size Connected Dominating Sets in Wireless Networks". In: *Wireless Algorithms, Systems, and Applications*. Ed. by Liran Ma et al. Cham: Springer International Publishing, 2017, pp. 109–114. ISBN: 978-3-319-60033-8.

[14] J. Yu et al. "Minimum connected dominating set construction in wireless networks under the beeping model". In: *2015 IEEE Conference on Computer Communications (INFOCOM)*. Apr. 2015, pp. 972–980. DOI: 10.1109/INFOCOM.2015.7218469.

[15] Bevan Das and Vaduvur Bharghavan. "Routing in ad-hoc networks using minimum connected dominating sets". In: *Communications, 1997. ICC'97 Montreal, Towards the Knowledge Millennium. 1997 IEEE International Conference on*. Vol. 1. IEEE. 1997, pp. 376–380.

[16] Jie Wu and Hailan Li. "On calculating connected dominating set for efficient routing in ad hoc wireless networks". In: *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*. ACM. 1999, pp. 7–14.

[17] Hyojun Lim and Chongkwon Kim. "Flooding in wireless ad hoc networks". In: *Computer Communications* 24.3 (2001), pp. 353–363.

[18] Raqeebir Rab, Ashikur Rahman, and Fatema Tuz Zohra. "Analytical modeling of self-pruning and an improved probabilistic broadcast for wireless multihop networks". In: *Ad Hoc Networks* 52 (2016). Modeling and Performance Evaluation of Wireless Ad Hoc Networks, pp. 106–116. ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2016.08.007. URL: http://www.sciencedirect.com/science/article/pii/S1570870516302001.

[19] Sławomir Nowak et al. "Global Queue Pruning Method for Efficient Broadcast in Multihop Wireless Networks". In: *Computer and Information Sciences*. Ed. by Tadeusz Czachórski et al. Cham: Springer International Publishing, 2016, pp. 214–224. ISBN: 978-3-319-47217-1.

[20] Tasmiah Tamzid Anannya and Ashikur Rahman. "Extended Neighborhood Knowledge based Dominant Pruning (ExDP)". In: *5th International Conference on Networking, Systems and Security, NSysS 2018, Dhaka, Bangladesh, December 18-20, 2018*. 2018, pp. 19–27.

[21] Ashikur Rahman, Pawel Gburzynski, and Bozena Kaminska. "Enhanced Dominant Pruning-based Broadcasting in Untrusted Ad-hoc Wireless Networks". In: *Proceedings of IEEE International Conference on Communications, ICC 2007, Glasgow, Scotland, UK, 24-28 June 2007*. 2007, pp. 3389–3394.

[22] Yishuo Shi et al. "A greedy algorithm for the minimum $$2$$2-connected $$m$$m-fold dominating set problem". In: *Journal of Combinatorial Optimization* 31.1 (Jan. 2016), pp. 136–151. ISSN: 1573-2886. DOI: 10.1007/s10878-014-9720-6. URL: https://doi.org/10.1007/s10878-014-9720-6.

[23] Raqeebir Rab, Ashikur Rahman, and Fatema Tuz Zohra. "Analytical modeling of self-pruning and an improved probabilistic broadcast for wireless multihop networks". In: *Ad Hoc Networks* 52 (2016), pp. 106–116. DOI: 10.1016/j.adhoc.2016.08.007. URL: https://doi.org/10.1016/j.adhoc.2016.08.007.

[24] B. Milic and M. Malek. "Dropped Edges and Faces' Size in Gabriel and Relative Neighborhood Graphs". In: *2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems*. Oct. 2006, pp. 407–416. DOI: 10.1109/MOBHOC.2006.278580.

[25] *Poisson distribution*. http://en.wikipedia.org/wiki/Poisson_distribution.

Appendix

$$\begin{aligned}
P_C(x) &= \sum_{k=1}^{n} P_k\left(INTC(x)\right)\\
&= \sum_{k=1}^{\infty} \frac{(nP_\delta)^k \times e^{-nP_\delta}}{k!}\\
&= e^{-nP_\delta}\left(\sum_{k=0}^{\infty} \frac{(nP_\delta)^k}{k!} - \frac{(nP_\delta)^0}{0!}\right)\\
&= e^{-nP_\delta}\left(e^{nP_\delta} - 1\right)\\
&= 1 - e^{-nP_\delta}\\
&= 1 - e^{-n\frac{INTC(x)\times\mu}{n}}\\
&= 1 - e^{-INTC(x)\times\mu}\\
&= 1 - e^{-\left[\pi r^2 - r^2\left\{2\sin^{-1}\left(\frac{x}{2r}\right) + \frac{x}{2r^2}\sqrt{4r^2-x^2}\right\}\right]\times\mu}\\
&= 1 - e^{-\pi\mu r^2 + \mu r^2\left\{2\sin^{-1}\left(\frac{x}{2r}\right) + \frac{x}{2r^2}\sqrt{4r^2-x^2}\right\}} \qquad (12)
\end{aligned}$$