ABSAW: Adaptive Buffer Spray And Wait, Delay Tolerant Routing Protocol designed for the Internet of Things

Anamika Chauhan[1], **Kapil Sharma**[1]
[1]Delhi Technological University, India

Abstract:
Delay Tolerant Network (DTN) is a class of network architectures designed for challenged networks. They primarily address the lack of network continuity, but can also simultaneously deal with other issues such as resource constraints and heterogeneity. This enables them to be studied in the recent decades as an alternative or extension of existing as well as promoting new routing protocols. Most recent DTN research attempts to redesign and propose solutions to support emerging network-based applications, for more delay, fault tolerance and flexibility. The most significant of these domains is the Internet of Things (IoT). The paper briefly surveys the similarities and areas of convergence in the use of DTN solutions in IoT applications. The design and implementation of a DTN-based routing protocol ABSAW, Adaptive Buffer Spray and Wait Protocol are proposed which can be a suitable alternative for buffer-constrained sensor-based IoT applications. The proposed protocol has been compared with three prevalent protocols, Epidemic, Spray & Wait, and ProPHET, and can be seen to outperform them in several evaluation metrics. The purpose of this work is to introduce a solution that enables delay-tolerant routing in IoT.
Keywords: Delay Tolerant Networks; Internet of Things; Routing; Spray and Wait; ProPHET
DOI: 10.24297/j.cims.2023.1-5

1. Introduction

As the internet of things continues to pervade its way into all areas of real-life applications a simultaneous requirement for supporting infrastructure also appears. The newer and more complex applications, that continue to be designed, require a network and communication system that is consistent and dependable. Another factor to be considered is, the pace of digitisation spread is much faster than what growth in physical infrastructure could keep up with. This manifests differently but simultaneously in both sparsely populated rural as well as densely populated urban areas. There is a requirement for alternative solutions that would help the consistent spread of IoT-based services even in areas that might lack full-fledged technical infrastructure support. Delay Tolerant Networking (DTN) can fill this gap by providing alternative as well as hybrid solutions for achieving this goal. Delay-tolerant solutions are designed for challenged or infrastructure-lacking environments. Thus they can aid in the expansion of IoT services in unstable environments.

In most IoT applications data acquisition is performed via sensors that are resource constrained, so the major challenge is processing a large amount of real-time, even multimedia, data from different types of sensors and maintaining reliable communication. The main requirement is that the network makes optimal usage of network resources as, well as has an assured Quality of Service. The major concerns in these sensor-based IoT networks are that permanent connections cannot be set up with the narrow spectrum available; also the limited processing ability and memory cannot maintain consistent state information per connection. The mobility of nodes adds a further layer of complexity. Research shows that Delay Tolerant solutions can improve overall network performance Furthermore; IoT applications that require multicast services would have even more suitable solutions as DTNs have excellent performance for multicast data dissemination to large groups of heterogeneous nodes.

Most solutions for DTN-enabled IoT centre around decentralising the routing process and replacing continuous connectivity with "Opportunistic" connectivity. Opportunistic behaviour means, that the neighbours and time of data transmission be decided in an opportune way i.e. when a chance meeting occurs. The success of such decision algorithms depends on the availability and accuracy of apriori knowledge (predictions), which is sometimes not available but research has shown that considering the resources and stability required for absolute optimum requires an unrealistic amount of computation, thus hybrid algorithms are proposed that are much more practical. The most prevalent algorithms fall under the class of the store-carry-forward mechanism. The nodes (stationary or mobile) are divided into different clusters and carry and forward messages across the network to deliver them as destined.

The most prevalent hybrid approaches embed a DTN-based layer also known as the Bundle layer in the existing TCP/IP protocol architecture. The approaches can be further classified based on the features and placement of the bundle layer. These adaptations do not completely cover the drawbacks of IoT applications but can offer a reasonable trade-off between performance and resource utilization. The given research explores all these topics and offers a novel alternative.

The main contributions of this paper are the following:

- The layered protocol architecture, routing and challenges in IoT are briefly discussed;
- The paper identifies the similarities and areas of convergence in the use of DTN solutions in IoT applications
- A survey of existing DTN routing solutions is performed with a focus on protocols adapted for IoT applications.
- The design and implementation of ABSAW protocol, a Buffer-adapted variation of the "Spray and Wait" protocol based on the combination of the benefits of the DTN routing strategies;
- The proposed ABSAW protocol is compared to the existing DTN routing solutions for IoT;

The remainder of this paper is organized according to the following plan: In Section 2, IoT architecture and challenges are discussed. In Section 3, the concept of Delay Tolerant Networks, the interrelationship between IoT and DTN, as well as existing hybrid solutions are discussed. Section 4 describes the proposed algorithm and the environment and parameters of the simulation. In Section 5, we present the obtained results to evaluate the performance of the ABSAW protocol in comparison to other existing DTN protocols. Lastly, Section 6 concludes the paper.

2. The Internet of Things

The Internet primarily started as a hierarchical web of connected networks of homogenous devices. As new devices and applications were developed, new models of connectivity like grid computing, mobile computing, and cloud computing were included. Heterogeneity became a norm and a new paradigm of networking the “Internet of Things” (IoT) came into being. Thus, enabling a range of devices and applications in a single global network, ranging from simple handheld devices to complex high-performance supercomputers, with the ability to sense, compute and communicate seamlessly.

This integration had the advantage of having additional functionalities and multiple applications, but at the same time, multiple concerns also came into play. The most significant issue is the interoperability between the wide range of heterogeneous objects. The heterogeneity and particularly wireless-ness introduced challenges associated with connectivity, such as less reliability and lack of trust between neighbour nodes, highly dynamic topologies, and significant disruption in connectivity and availability due to mobility.

Simultaneously IoT environment has grown, into an all-pervasive network, due to its advantages and flexibility. Research solutions are being designed for large-scale urban environments like Smart Cities, Vehicular Networks, Smart grids, etc. based on IoT solutions. Most of these solutions have sensors as communicating nodes and use some existing wireless communication methods like Wi-Fi, Bluetooth, and Zigbee, as they are meant to be low power consumption and cost-effective. But these networks suffer from several issues like low energy constraints due to battery life, reliability of link connection, and particularly security issues.

2.1. The Internet of Things architecture

IoT connects millions of objects across heterogeneous networks by the means of the internet so the architectural framework of IoT requires it to be flexible there are several architectures available based on a broad reference model. The most prevalent models are the three, four, and five-layered based on the functional perspective of the IoT network. [1,2]

2.1.1. Three-layer architecture

The Three-layer architecture is the most generic vertical separation of components of an IoT network. It consists of perception at the bottom and applications at the top, with the network layer sandwiched in the middle. The perception or sometimes referred to as the perceptron layer acts as firmware between physical devices like sensors and actuators and the higher layers the purpose of this here is the acquisition of data from 'Things' and processing it to send to upper layers.

The network layer consists of a variety of communication networks integrated into the internet. A host of switching devices form part of this most significant layer. The most important function of this layer is to transport faithfully the data and information acquired from the sources and then route them to appropriate destinations or applications. This paper attempts to modify and improve one of the routing protocols by making it adaptive to the available resources, in this case, the buffer capacity.

Application is the topmost layer and this is a complete software layer that consists of the various applications that require the data that has been acquired by the perception layer or implementations of decisions back to the perception layer.

Three-layered architecture is very basic and simple and this simplicity enables it to be used for supporting and building any IoT application. It can be further extended into complicated multiplayer architecture as per the need for a variety of applications.

2.1.2. Four-layer architecture

The Four-layer architecture is what is referred to as Service Oriented Architecture (SOA. It consists of sensing, networking, service, and interface layers. The design of this architecture is used as a framework for WSN, Cloud and Vehicular, and other smart networks. This architecture has been shown diagrammatically in fig 2.

It can be seen that both the sensing and the perception layer in the above two architectures acquire data from a variety of different sources of IT infrastructure like healthcare systems, vehicular networks or any other smart automation-based system so and these devices are interconnected with very high flexibility and most data travels in the real-time as these things are related located in a very heterogeneous environment that two main concerns at the networking layer, therefore, our quality of service control and the management of variety for different applications this leaves to a highly dynamic environment where different IT services are constantly in motion and neighbourhoods are changing fast devices will be capable of performing efficiently if they take care of issues like quality of service energy efficiency fast data processing and also a discovery of objects and retrieval of data and last and probably most important security

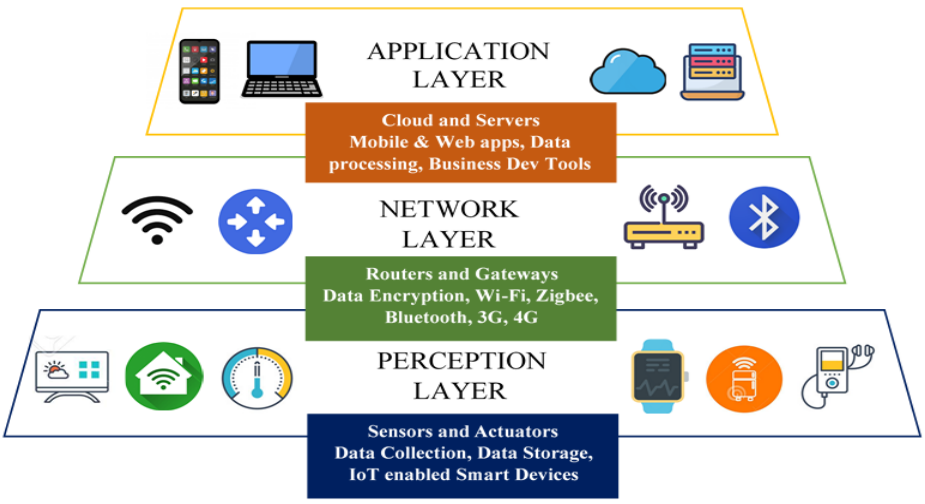


Fig 1. Three-layer IoT architecture

There is also a Five-layer architecture that makes a further break up into the elements other than infrastructure, here apart from sensing, communication and services, the focus is also on computation and semantics.

2.2. Routing in the Internet of Things

The purpose of a communication network is the faithful delivery of messages from the source to the destination. The success of end-to-end delivery is determined by finding the best possible path or route between the source and destination, and therefore an effective routing strategy is crucial to efficient communication. Most routing protocols designed for the Internet, and thus IoT, use a directed acyclic graph (DAG), such that there is always a path or route present from each node to every other node. Continuous connectivity is obtained by maintaining large routing tables at all times. This requires memory space, processing ability and bandwidth for the routing-related metadata exchange, at all times. Also, there can be one single path existing or multiple paths towards the destination, the necessary and sufficient condition being that at least one device element of the network must have access to the internet at all times.

TCP/IP is the most widely used reliable delivery protocol suite. Many DAG-based routing protocols are designed for IoT and interoperate with networks based on TCP/IP. This works fine as long as it has continuous connectivity. Continuous connectivity ensures that very few packets are being dropped; network topology is stable and the packet delivery rate is high.

2.3. Internet of Things challenges

When the theoretical framework given above, is applied to a wide variety of real-life applications one or several of the following challenges are faced:

Continuous availability: Anytime availability of services in a highly dynamic environment is one of the primary requirements for any IoT-based services. Thus, the hardware of devices and software protocols must be designed for data and services to be available at all times.

Reliability: Despite a highly dynamic environment, the behaviour of the system and the quality of data must be reliable. Also, it must not be prone to frequent failures this applies to both the hardware and software aspects of the IoT infrastructure.

Mobility: Many of the devices or things are smartphones and similar other portable devices, making mobility one of the prime challenges in an IoT-based environment. The users' and devices’ motions can request services and they need to be performed continuously with the same quality of service as stationary devices.

Consistent Performance: Any variation of load concerning a variety of factors must not affect the overall performance.

Heterogeneity Management: a variety of different heterogeneous systems connected requires maintaining standards and having a central management system. Also, the management protocols must be lightweight to handle this efficiently.

Scalability: Since the number of devices is continuously increasing scalability is another very important factor in IoT-based systems. Also, the network size varies due to mobility and demand fluctuations of users at different times with different applications.

Interoperability: Another feature necessary due to the variety of devices in the IoT environment is interoperability. Most standard available hardware, middleware, and protocols must be included.

Security and privacy: lastly, but probably most important security is essential to IoT-based networks. These things include devices like smartphones containing sensitive personal data

3. Delay Tolerant Networks

During the same period as IoT another class of networks the Delay/Disruption Tolerant Networks (DTN), sometimes also referred to as Opportunistic Networks (ON) was developed for providing routing in a challenged network where no stable end-to-end path is available [4]. A challenged network can be defined as a network with no stable and direct end-to-end path from source to destination. This is caused by such networks being infrastructureless [3,4,5]. It has frequent network disruptions and a lack of resources. The nodes are highly mobile and dynamic. DTN uses this very property of mobility of nodes to form paths opportunistically and deliver messages from one node to another node. The mobile nodes move in different clusters and carry and forward messages across the network to deliver them as destined. These networks initially had ad-hoc applications in areas such as tracking wildlife in difficult terrain using sensor networks, military, and underwater purposes, satellite networks, etc.

The reason for DTN successfully working in this application is that it overcomes the difficulty of accessing the network continuously, even in remote or dynamic environments where there is no guarantee of the availability of a complete and stable path from source to destination. The traditional infrastructure-based routing protocols quite naturally due to the nature of their design fail to deliver in such a challenging environment.

3.1. DTN and IoT Interdependency

Literature analysis shows several similarities between the design issues, node/ traffic behaviour and resource constraints, and performance metrics of DTN and IoT. This has led to an array of solutions being designed with hybrid mechanisms. The DTN-enabled IoT network solutions enable smart objects to effectively communicate with more efficiency even in the presence of frequent disruptions. This also addresses the much larger issue of lifetime constraints. Recent studies have shown that DTN within the IoT framework provides the most suitable and satisfactory results.

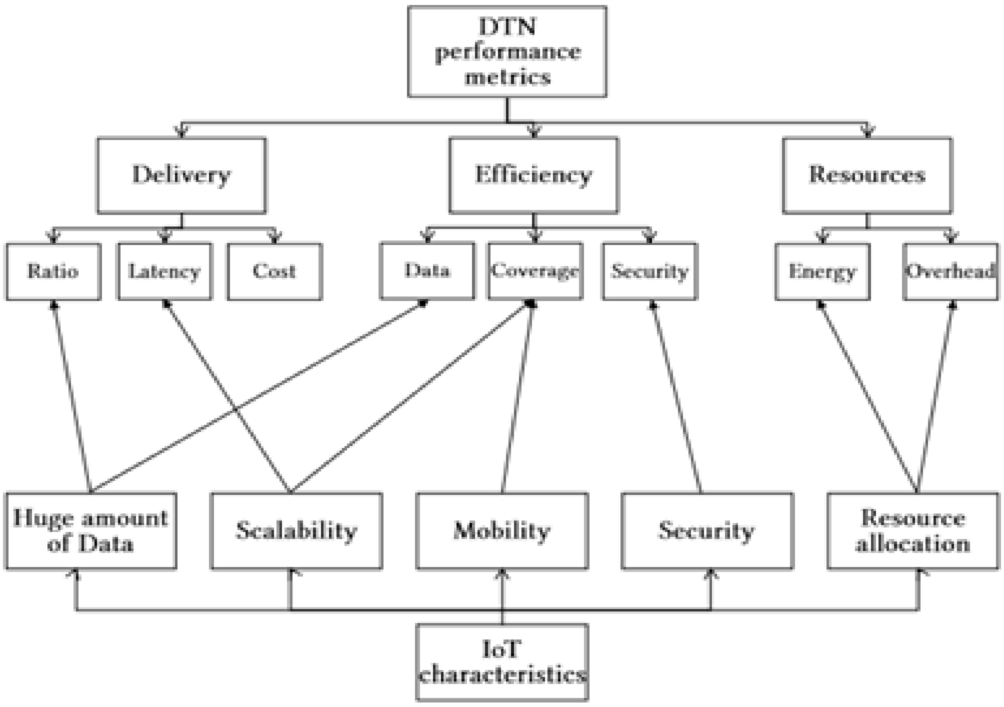


Fig3: Interdependency of DTN and IoT

3.2. DTN Routing and protocols

DTN can offer deterministic solutions as compared to a fully Ad-Hoc environment. Applications with large area coverage and heterogeneity are not very effectively handled using the existing TCP/IP model. A few of the primary reasons for the lack of suitability of only the TCP/IP protocol stack architecture are as follows:

Centralised nature
Maintenance of very large routing tables.
Connection orientation

Maintaining the above features would incur heavy expense, in the case of a partitioned and multi-hop environment. Therefore, DTN schemes in the IoT environment will be very suitable and have much better handling potential for lightweight applications. These solutions can be implemented separately very much like WSN and MANETs, or they can be implemented within the existing infrastructure. The placement of the DTN protocols in the existing TCP/IP protocol stack can be seen in fig 4. [26[27].

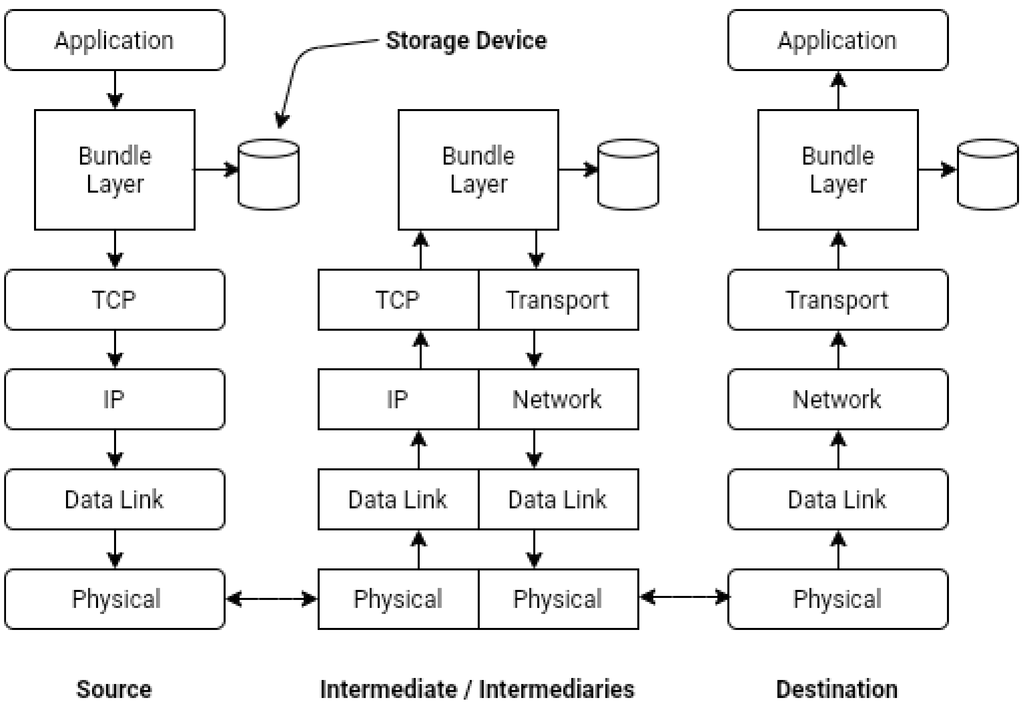


Fig4: TCP/IP Protocol stack with Bundle Layer

DTN is designed as network architecture with adaptation for variation and long latency. The basic principle of DTN is to use this very flaw and incorporate it into the system by having a store-carry-forward mechanism. All DTN routing solutions use this approach, it allows for some latency by keeping the data units to be stored in the transmitting nodes instead of directly forwarding. This also does not require maintaining large network forwarding tables like infrastructure networks.[1,10,13,18]

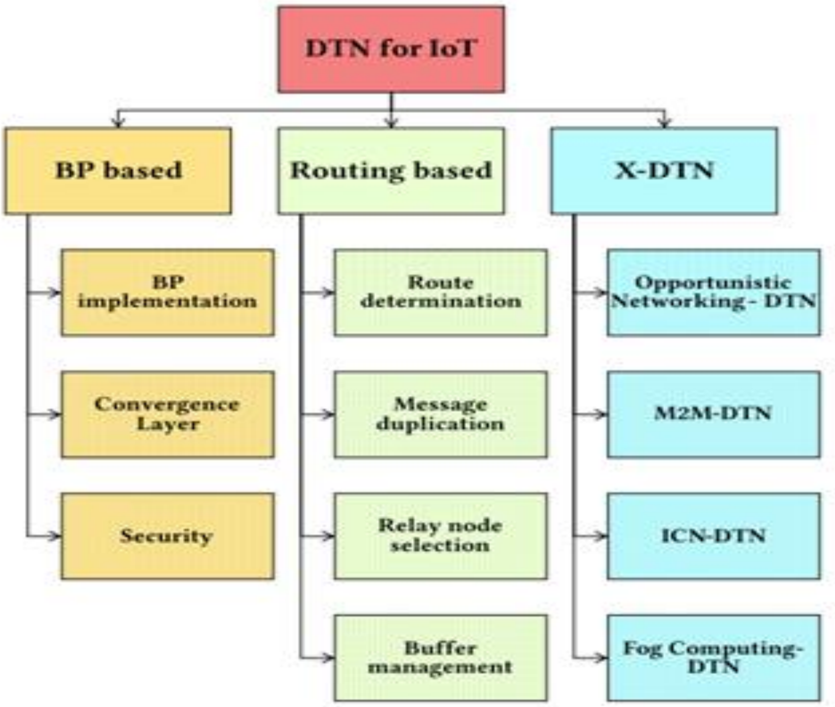
The three primary classes of DTN solutions are:
1. Bundle Based
2. Routing Based
3. Others referred to as X-DTN

Bundle Based: Bundle Protocol (BP) transports packets in form of messages or bundles together, not as individual packets. Bundling allows multiple packages to be sent in a message-oriented way on top of a disruptive network. This acts as an overlay over the existing network. This approach was derived from existing solutions for Wireless Sensor Networks (WSN). Such networks have disruptive mobility and data can be stored locally at each node till the next hop is available. The key change is to have a convergence layer on top of any transport layer protocol. This is very significant as uninterrupted interoperability is achieved, without change to the existing Physical and MAC layer and security concerns are also easily addressed.

Routing-based: This category of protocols addresses three major criteria for obtaining solutions, route selection, degree of replication, and also the selection of good next-hop or relay nodes so that a minimal amount of forwarding is required for routing. Another critical factor to be kept in mind is buffer management such that the overhead packet loss rate and the delay in delivery are minimized. **This paper suggests an alternative mechanism for enhancing the performance of, Spray and Wait routing and compares it with several existing protocols.**

Particularly this protocol has been selected as this type of routing allows for less bandwidth, is generally scalable, and can have devices with energy constraints. So this is probably the most suitable solution for making existing IoT networks more efficient.

The following table presents a taxonomy of solutions for IoT that is based on duty and protocols solutions are proposed on a DTN classification criteria



3.3. Literature Survey of DTN Routing protocols

Overall routing protocols will have to be divided into two parts, the Forwarding part, and the Replication part. The protocols have evolved to reduce the flooding mechanism which tends to be undesirable and inefficient, also redundant and duplicate copies of the same message, leading to congestion and waste of network resources. Also, nodes need a buffer to store the messages which have finite capacity and therefore lead to a high number of dropped or timed-out packets leading to wastage of bandwidth and poor efficiency.

Therefore, the two main criteria for goodness are efficient delivery to destination, while reducing the replicas and number of damaged or lost messages, and reducing overall latency. Therefore, most protocol designs concentrate on two separate aspects, the number of copies, single or multiple, and the number and fitness of forwarding nodes.[14,15]

Since the primary concern here is not just assured delivery but also the least cost of hops and replicas, therefore all of the protocols generally referred to as the replication-based algorithms, are classified into two classes. Based on the extent of replication they can be Limited or Unlimited classes. Unlimited replication-based algorithms [14] replicate the bundle and flood them amongst all neighbours that they are in contact with. Essentially, they have benefits and

drawbacks same to flooding, which are a high delivery probability but an increase in overhead. It needs and consequently wastes a lot of resources. But as resources are limited in DTN, there is a requirement for means to conserve them. Most protocols try to balance the two requirements.
One of the first protocols to be designed for DTN is the direct delivery routing (DD) protocol. In DD as the name suggests the delivery of the bundle is done by the source node directly to the end node. This is least effective as sometimes a direct path is unavailable and packets are dropped at a very high rate on being expired. Epidemic Routing Protocol [6,11] is the earliest example of unlimited replication-based routing algorithms. Epidemic routing is a simple approach here a node with a bundle to send just floods the bundle to all the neighbours that are in contact, and then these nodes in their turn flood the bundle to all their neighbours and so on. The packet is replicated and flooded on till the bundle reaches the destination or the TTL expires. The most commonly used and modified unlimited replication-based algorithm is the PRoPHET [19] protocol. Prophet attempts to minimize the flooding to reduce the overhead ratio. This control on the extent of flooding is achieved by ProPHET by taking the encounter history of each neighbour as an input parameter and using this to select a few of the neighbours to send the bundle instead of all neighbours. It has two benefits; it naturally reduces the number of replicas and also improves delivery probability. Since the previous encounter history is being used to search and select more suitable next-hops instead of all the bundles has a higher probability of reaching the destination. It establishes a probabilistic metric at every node to define the delivery probability of other nodes to each destination to make a good decision about whom should forward the bundle. It improves the delivery probability and decreases the overhead caused by flooding in the network.

The metrics of performance are also highly dependent on the routing strategy. Many new technologies have evolved recently, including theAn example of a limited replication-based algorithm [9,10] is Spray & Wait routing protocol. This protocol is explained later in detail in the proposed section as the paper proposes a variation of Spray & Wait with Adaptive Buffer. Other protocols based on Encounter history and heuristics are Prophet [25], Prophetv2 [25], and RAPID. only one is left, the node waits for a direct meeting of the destination node. “Spray and Focus” [9] after such distribution forwards the last replica to more promising carriers (and deletes from the sender). Most often, algorithms rely on statistical patterns since models of randomly wandering objects are usually assumed [10]: Random Walk (Brownian Motion), Random Waypoint (RWMM), Random Direction, Boundless Simulation Area, City Section, Cluster Mobility Model, Shortest Path Map-Based mobility model [11] and others.

The possibility of future meetings estimation is based on the history of the meetings (for example, “PRoPHET” [12] and “PRoPHETv2” [13], “MEED” [14], “EBR” [15]). “DTC” [16] also uses physical communication parameters (signal power level and so on). “MaxProp” [16], developed based on practical experiments on regional regular buses, additionally uses the history of each message. The degree of coincidence of the direction of movement can also be a parameter (“Vector routing” [18]). In [19], a message is sent only if the expected delivery time is better than the time of all those to whom copies were sent earlier. Some protocols rely on forwarding instead of copying (sending a message without saving a copy at the sender). For example, “Contact Graph Routing” [20], uses recursive analysis of all possible paths for that. An important way to reduce the used buffers is to distribute between the nodes information about the message status and delete already delivered packets from the buffers [21]. It is also proposed to use Reinforcement Learning for local balancing of the buffer load (for example, for autonomous communication in deep space).

More recent works have proposed DTN routing protocols in the context of IoT [6]. This inclusion of heterogeneity, smart grids, and drones (UAV), each of them uses different DTN routing protocols.

The following table shows the evolution of DTN routing protocols and adaptation for IoT:

Reference	DTN routing protocol	Replication Strategy	Routing Strategy	Year	Adapted for IoT
[6]	Epidemic	Unlimited	Flooding	2000	-
[8]	Prophet	Controlled	Forwarding	2003	-
[16]	Spray and Wait	Controlled	Flooding	2005	-
[14] [22]	Maxprop Spray and focus	Controlled Controlled	Forwarding Flooding	2006 2007	-
[1]	RAPID	Controlled	Flooding	2010	-
[1]	Prophetv2	Controlled	Forwarding	2011	-
[17]	IoB-DTN	Unlimited	Flooding	2018	Yes
[23]	Hybrid type dtn routing protocol considering storage	Hybrid	Hybrid	2019	Yes

	capacity				
[19]	Scheduling-PROPHET	Controlled	Forwarding	2019	Yes
[25]	Multi-objective based deployment of throwboxes in delay tolerant networks for the Internet of Things environment	Hybrid	Hybrid	2020	Yes

Proposed approach

The proposed approach ABSAW modifies the buffer capacity usage. The resource efficiency introduced would make this suitably useful for the IoT environment, especially low-power/memory sensor-based scenarios. Two new variables represented by *'Eligibility'* of transfer & *'Priority'* in the buffer queue are introduced. The queue is being managed as per *'Priority'*. Therefore, the proposed approach would be referred onwards as "**Adaptive Buffer Spray And Wait**" or abbreviated to" **ABSAW**".

The ABSAW routing protocol has a sequence of the following two phases:

1. Spray
2. Wait.

In the first phase, the message is 'Sprayed' thereby being replicated into a limited number of copies (as S&W is a limited-replications method) and then forwarded to several neighbour nodes. These nodes then in turn also engage in further spraying of the message, in a tree-like fashion. Different variations use different types of spraying and replication mechanisms.

After spraying the Wait phase begins. In this phase, the nodes wait till the message is delivered to its destination. If it fails to be delivered to the destination, the protocols switch into a direct delivery routing approach and the message is delivered directly. So, this protocol combines the features of having higher speed and simplicity as it combines epidemic routing and the direct delivery routing protocol.

The performance assessment and scope of improvement/modification of any routing protocol are dependent on the delay or latency, that it takes to deliver packets correctly and the number of copies required. As per this the work can be divided into two parts that can be modified separately or together:

Scheduling part: The strategy or way of sending messages to the next node(s)
Buffer Queue Management part: The strategy or way of deciding which message-deleting the message from the buffer.

Both of these strategies can impact the performance of a routing protocol in significant ways. A variety of scheduling methods are used, ProPHET uses the history of encounters [5] with other nodes, a statistical property of two parameters delivery predictability and transitivity. And the standard buffer management method is First-In-First-Out.

In this paper, the proposed algorithm suggests a modification to both of these two strategies to implement a buffer-adapted variation. The proposed modifications are as follows:

A. Scheduling Strategy: The proposed algorithm tries to reduce the flooding in the network by putting some conditions based on a new parameter. If the receiving node satisfies those conditions, then the sender will forward the message further, otherwise, it will not forward it to that node. The first condition is that if the receiving node is not the destination node then it should have at least two connections to forward the message further. The second condition is that it should be at a minimum distance from the sender node or has a minimum buffer load. This ensures less replication and a local minimum is achieved. As it is known that the shorter the distance, the higher is the delivery probability. Likewise, the lower the buffer load higher the chances that the message will not get dropped.

$$\mathrm{ED_{best}} \propto \frac{1}{\mathrm{distance}} \quad (1)$$

$$\mathrm{ED_{best}} \propto \frac{1}{\mathrm{buffer\ load}} \quad (2)$$

$$\mathrm{ED_{best}} \propto \frac{1}{\mathrm{distance}} * \frac{1}{\mathrm{buffer\ load+1}} \quad (3)$$

Effectively the sender will compute the '$\mathrm{ED_{best}}$ *or Best Delivery Eligibility'* of each node in the communication range and will forward the message to only the best two routers. And these the receiving node would have at least 2 connections and would have maximum eligibility value.

Algorithm 1 The Scheduling Process for calculation of ED_{best}

Input: $ED_{initial}$, initialisation of Eligibility for a node in the Neighbour circle
$S = \{s_i \mid 1<i< n\}$, set of all nodes in the network
$E_i = \{s_k \mid n_{ik} \neq 0, 1 \leq k \leq n\}$, set of nodes encountered by node s_i,
Initialize $ED_{initial} = Null$, for node S_i;

if s_i, Encountered s_j, and $s_j \notin E_i$, **then**
    $E_i = E_i \cup \{s_j\}$
    if $s_j \notin ED_{best}$ **then**
        $D_{(i,j)} = | E_i \cap E_j |$
        if $D_{(i,j)} > ED_{threshold}$ **then**
            $ED_{best} = ED_{initial} \cup \{s_j\}$
        End if
    End if
End if

B. Queue Management Strategy: The Epidemic and many other routing protocols in DTN follow the First-In First-Out mechanism. This is to say that any message in the buffer queue would be processed in the FIFO order for scheduling to the next eligible node. And once the buffer is full any newly arriving messages will be stored and the oldest message in the buffer queue would be deleted and discarded regardless.

The proposed work applies three new strategies to utilise the space in the buffer queue adaptively based on a new variable *'$P_{Transfer}$ or Transfer Priority'* message and compares them with 3 of the most applied protocols.

$$P_{transfer}(x,y) = P_{transfer}(x,y)_{old} + (1 - P_{transfer}(x,y)_{old}) \times P_{transfer_{init}} \quad (4)$$

And

$$P_{transfer}(x,y) = P_{transfer}(x,y)_{old} \times \gamma^k \quad (5)$$

Where γ is an ageing constant [2, 3] and $\gamma \in [0,1]$ and k is the ageing factor that depicts the time that has elapsed since the last delivery to that destination.

The rationale behind the proposed strategy is to identify the optimal path among the various possible paths. This traversal problem can be modelled as a resource, in this case, buffer, allocation model in which an incoming or existing packet will be discarded or assigned buffer space, to maximize the probability of the packet reaching the destination node. This is a combinatorial optimization, similar to a knapsack problem with a single limitation, which is an NP-hard problem. And hence an optimal solution can be obtained from a large set of possible solutions.

Due to the mobility, low energy and space requirement of sensors and devices in IoT, network routes are unpredictable and unstable and therefore delay tolerance is required. Therefore, instead of solving the end-to-end problem, the routing is converted into a set of simpler sub-problems. The optimal substructure ensures an overall optimal solution. The pseudo-code for the proposed strategy is given below followed by a detailed explanation of the same:

Algorithm 2 The Queue Management Strategy for calculation of $P_{transfer}$

Input: $S = \{s_i \mid 1<i< n\}$, set of all nodes in the network
$R_i(m_k)$, the replicas of message m_k carried by node s_i,
NC_d, the Neighbour circle of the destination
nNodes, the number of neighbour nodes in S_i's transmission range
SM_i, set of messages in queue carried by s_i,
SM_j, set of messages in queue carried by s_j,
Initialize $P_{transfer}$ = *Null*

if s_i encounters s_j and $R_i(m_k)) > 1$ and s_j NC_d **then**
    update $P_{transfer(i,j)}$ and NC_i
    $SM = SM_i \cap SM_j$,
    for each message m_k in SM **do**
        s_d = m_k's destination
        if $s_d == s_j$ **then**
            s_i directly forwards m_k to s_j
        else if $P_{transfer(i,d)} < P_{transfer(j,d)}$ or nodes < 2 **then**
            $R_j(m_k) = \lfloor R_i(m_k)/2 \rfloor$
            add $L_j(m_k)$ copies of m_k to $P_{transfer}$
        end if
    end for

if $P_{transfer}$!= *Null* **then**
    sort $P_{transfer}$ in ascending order of TTL
    s_i forward $P_{transfer}$ to s_j
end if
end if

B.i. The first strategy is to delete the message whose destination has been encountered by the node most recently. Because the nodes keep flooding the same messages until they are in their buffer. It means a node is receiving the same messages many times it may have forwarded the same message to other nodes many times. So, keeping this thing in mind, the proposed algorithm is deleting the message whose destination has been encountered by the node most recently.

B.ii. In our second queue management strategy, the nodes will delete the message whose destination has been encountered least recently. In this way, the proposed algorithm is applying the idea that the node will not receive the message with the destination address which the node has encountered least recently anytime soon. So, the node will delete that message. If there is a situation in which two or more message destinations have the same encountered age then in that case proposed protocol is applying the First In First Out strategy in which the node deletes the oldest message in the buffer. So if two message destinations have the same encountered age then the protocol will delete the older message.

B.iii. The third strategy is to delete the message whose destination address is farthest from the node. As earlier stated, the delivery probability is inversely proportional to the destination distance. So, the proposed algorithm has used the same idea here. If the destination node is very far away from the current node then delivery probability gets reduced and as DTN nodes have limited resources, the protocol will use those resources on those messages whose probability to get successfully delivered is higher. In this way, it will reduce the overhead and simultaneously increases the delivery probability.

Each node maintains a delivery probability table that shows the probability of a message getting delivered to the destination from the last wait cycle. Whenever nodes meet, they exchange their delivery predictability table and update their delivery probability table. Transitivity is if node X frequently encounters node Y and node Y frequently encounters node Z then node Z is a good relay to deliver the message to node X. As each node calculates the delivery predictability for all known destination nodes where $P(x, y) \in [0, 1]$. To calculate delivery predictability where a node encounters another node:

This protocol assumes that the bandwidth is unlimited so the time taken to deliver messages is ignored. The transitivity property decreases the message dropping rate and it also helps in decreasing the time a message wastes in the queue of a node. It lowers the load and the pressure of a node.

4. Simulation Environment

This Simulation of DTN routing protocol cannot be efficiently performed by tools used for traditional networks. The DTN routing protocol requires node and route characteristics that deal with mobility, intermittent connectivity and resource constraints.

One of the most prevalent tools is the Opportunistic Network Environment (ONE) simulator, as presented by Keränen et al. This software facilitates the modelling of different scenarios for existing and new DTN routing protocols. It is a powerful tool that allows recreating and testing of Epidemic, Spray-and-Wait, MaxProp, and Rapid ProPHet very easily. The ONE simulator is specifically designed for the investigation, comparison and evaluation of various DTN routing protocols. The software is simple, flexible and easy to use in comparison with OMNET++, NS2 and NS3 etc.

5.1. Advantages and evaluation parameters of ONE Simulator

Among these software tools suggested above, the ONE simulator has several advantages for modelling DTN routing protocols. It is most suitable as it has network scenarios that are designed specifically for evaluating the DTN routing protocols. Therefore, this paper uses the ONE simulator for implement and comparison of the new and existing DTN routing protocols.

The ONE simulator depends on mobile node movement, the density of the nodes, and the distance between the sender and receiver. These factors significantly affect the performance of the DTN routing protocols in terms of relying on latency, the delivering probability, and the overhead ratio, among other factors. The main criteria are to identify a suitable routing and forwarding approach and match these techniques with real-time mobility.

The ONE simulator is open-source software that allows front-end editing and execution of programs for development and result visualisation. However, the software requires Eclipse IDE for Java Developer v.2020-06. The ONE simulator software can be compiled in both ways, through Windows or using Eclipse IDE.

This flexibility of ONE simulator is another reason why most publications use and suggest its suitability for the evaluation of DTN routing protocols. Another feature of ONE simulator is the generation of mobility traces, running the DTN message, visualization of the simulation and presentation of a log of the results of execution.

5.2. Simulation and Evaluation parameters

The simulation parameters considered for analysis in the DTN routing protocols are summarized in Table.

Parameters	Values
Simulation Area	4500*3400
Simulation Time	43200
Mobility Model	Shortest Path Map-Based Movement
TTL	300
Buffer Size	5MB
Transmission Range	10
No. of Nodes	126
Bundle creation rate	25 to 35 seconds
Bundle size	500kB - 1MB

5.3. Metrics of Performance

Several factors are commonly utilized for the assessment of the performance of DTN routing protocols: overhead ratio, packet delivery ratio, average latency, and average hop count [21], [27]. These metrics are described as follows:

Overhead Ratio: One of the most important metrics used to assess the performance of DTN routing protocol Overhead Ratio. It can be defined as the number of duplicate packets that are required to be transmitted to ensure successful delivery. The overhead ratio provides a measure of the network congestion status, which is used to determine the bandwidth required and the number of successful replications required for the packet delivery [5], [21]. It is given by Eq. (7):

$$OR = (R - D)/D \quad (7)$$

Where R is the number of successful transmissions and D refers to the number of messages delivered to the destination [21], [27].

Delivery Probability: The delivery probability is yet another important metric to assess the performance of any DTN routing protocol. It is a measure of the ratio of the actual number of packets delivered to the destination and the actual number of packets sent from the source node. A high packet delivery ratio signifies less loss and thus better performance of the network It is calculated as presented by the authors in [28] and [29], and is given by Eq. (8):

$$PDR = D$$

Where DM is the number of successfully delivered messages, while CM is the number of created messages.

Average Latency: Average latency is defined as the time elapsed from the time the message is sent from the source node to the time it is delivered at the destination node. That is to say, it is the average time taken by the message to be created by the source node till the time it is received by the destination node [21], [27].

d) Number of Hops: According to Baek et al. [5], the hop account is defined as the number of nodes that a message has been sent to thus far. If the message is created at a node, the hop count is calculated as zero. That is to say, the hop account indicates the number of hops that the message makes between the source node and the destination node [21].

5. Results and Discussion

The results of the ONE simulation have been generated through reports that are created by report modules during the run time of the simulation. The Simulation engine provides the data to the report modules for the run-time events, reports are then created based on these received results [1].

As suggested above, the routing protocols reports contain measures and values for different factors such as lateness, packet delivery ratio and bandwidth consumption.

5.1. Reporting and Visualization

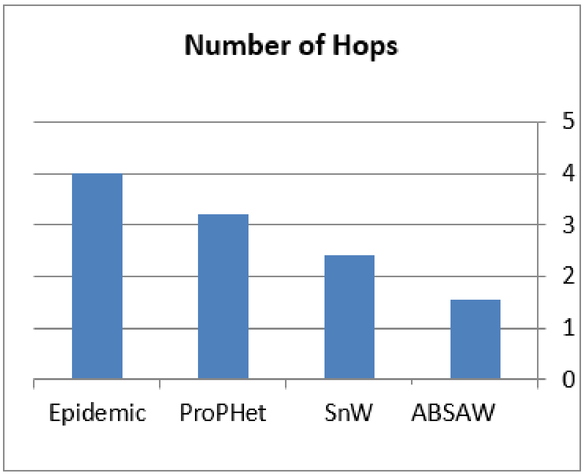
There are two ways to visualize the simulation results in the ONE simulator:

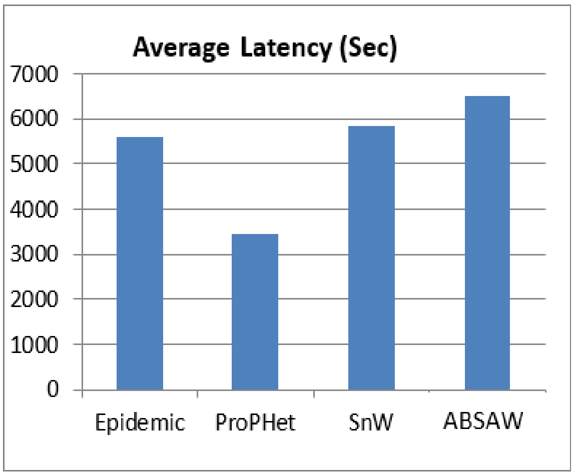
generating images from the information gathered during the simulation, and via an interactive GUI, The simulator has a graphical user interface (GUI) that is launched with Java. It is possible to zoom in and out and to change the speed by using the GUI update icon in this playfield graphics in the ONE simulator. The playfield graphics has various buttons and icons to play the simulation step forward internally, enable and disable fast forward, and play the simulation for a specific time [1].

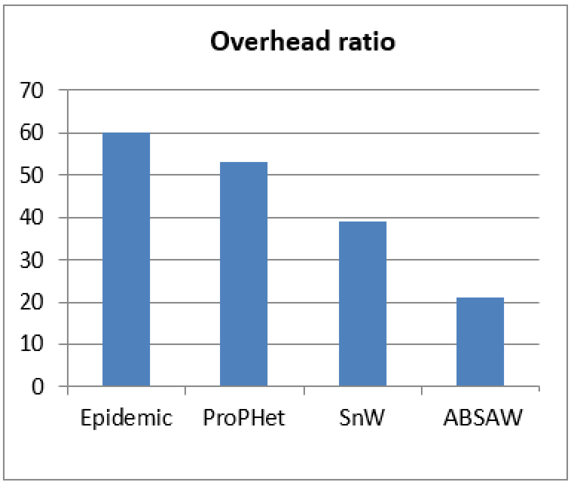
As result, this paper has compared our implemented routing protocols to epidemic and prophet. And the proposed protocol outperformed the epidemic, ProPHET and SnW routing protocols. This paper compared them based on the delivery probability, overhead, number of hops and latency.

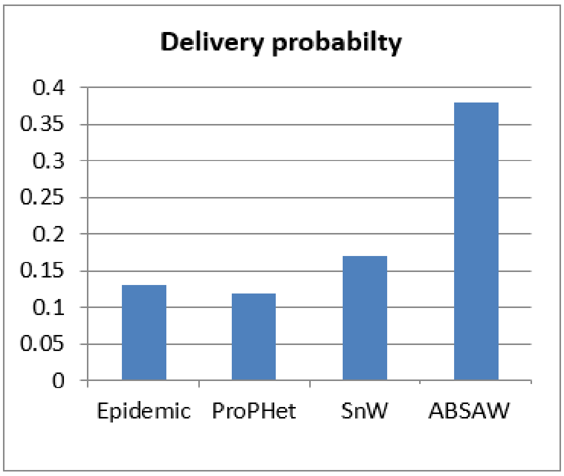
Simulation Results & discussion

The comparative graphs for the three existing protocols with the proposed approach 'ABSAW' are as follows:









As evident from the graphs for all 4-performance metrics, the proposed approach outperforms the existing protocols. It can be noted that for most parameters Epidemic and SnW protocols offer quite similar results, which are considered the optimum compared with the other protocols. In contrast, the PRoPHET behaves differently and represents the latency due to its history requirement.

The number of hops, also consistently improves among the four routing protocols for the same set of simulation parameters.

6. Conclusion

Many recent surveys and research have covered the use of DTN routing protocols in conventional networks. However, very few of them specifically address the problem of enabling delay-tolerant IoT. Proposing protocols and solutions dedicated to DTN solutions for IoT is quite a recent development with major scope for further research and applications. This paper attempts to bridge this gap by introducing a design of a buffer adaptive DTN-based routing protocol, that can effectively be used for enabling a delay-tolerant IoT architecture and/or application. The paper systematically elaborates on the convergence and overlap of infrastructure-based IOT and seemingly infrastructure-less DTN. Both areas have been discussed with their advantages and limitations.

The paper explores and confirms whether it is possible to adapt existing DTN architectures for IoT applications, with due consideration to resources and other constraints. The same is further evident from the superior performance of the proposed routing scheme. It can be confidently concluded that this is a promising and interesting area for all research communities. DTN and

IoT converged solutions will make possible the design of new and improved solutions for a huge diversity of existing and new IoT applications.

References

1. Bounsiar, Selma & Benhamida, Fatima & Henni, Abderrazak & Ipiña, Diego & Casado Mansilla, Diego. (2019). How to Enable Delay Tolerant Network Solutions for Internet of Things: From Taxonomy to Open Challenges. Proceedings. 31. 24. 10.3390/proceedings2019031024.
2. El Arbi Abdellaoui Alaoui, Stéphane Cédric KOumetio Tekouabou, Antoine Gallais, Said Agoujil, DTN Routing Hierarchical Topology for the Internet of Things, Procedia Computer Science, Volume 170, 2020, Pages 490-497, ISSN 1877-0509,
3. Narayan, Vipul, and A. K. Daniel. "Design consideration and issues in wireless sensor network deployment." (2020): 101-109.
4. K. Sakai, M.-T. Sun, W.-S. Ku, J. Wu, F.S. Alanazi, Performance and security analyses of onion-based anonymous routing for delay tolerant networks, IEEE Trans. Mob. Comput. 16 (12) (2017) 3473–3487
5. Y. Li, P. Hui, D. Jin, S. Chen, Delay-tolerant network protocol testing and evaluation, IEEE Commun. Mag. 53 (1) (2015) 258–266
6. A. Vahdat, D. Becker, Epidemic routing for partially-connected ad hoc networks, Tech. Keränen, Ari, Jörg Ott, and Teemu Kärkkäinen. "The ONE simulator for DTN protocol evaluation." In Proceedings of the 2nd international conference on simulation tools and techniques, p. 55. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009
7. Keränen, Ari, Teemu Kärkkäinen, and Jörg Ott. "Simulating Mobility and DTNs with the ONE." Journal of Communications 5, no. 2 (2010): 92-105.
8. Narayan, Vipul, and A. K. Daniel. "CHOP: Maximum coverage optimization and resolve hole healing problem using sleep and wake-up technique for WSN." ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal 11.2 (2022): 159-178
9. Musolesi, Mirco, and Cecilia Mascolo. "CAR: Context-aware adaptive routing for delay-tolerant mobile networks." IEEE Transactions on Mobile Computing 8, no. 2 (2009): 246-260
10. Narayan, Vipul, and A. K. Daniel. "CHHP: coverage optimization and hole healing protocol using sleep and wake-up concept for wireless sensor network." International Journal of System Assurance Engineering and Management 13.Suppl 1 (2022): 546-556.

11. Vahdat, Amin, and David Becker. "Epidemic routing for partially connected ad hoc networks." (2000)
12. Lindgren, Anders, Avri Doria, and Olov Schelén. "Probabilistic routing in intermittently connected networks." In ACM International Symposium on MobiIde Ad Hoc Networking and Computing, MobiHoc 2003: 01/06/2003-03/06/2003. 2003
13. Narayan, Vipul, and A. K. Daniel. "IOT based sensor monitoring system for smart complex and shopping malls." Mobile Networks and Management: 11th EAI International Conference, MONAMI 2021, Virtual Event, October 27-29, 2021, Proceedings. Cham: Springer International Publishing, 2022.
14. Psounis, Konstantinos, and Cauligi S. Raghavendra. Multiple-copy routing in intermittently connected mobile networks. Technical report CENG-2004-12, USC, 2004.
15. Spyropoulos, Thrasyvoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. "Single-copy routing in intermittently connected mobile networks." In 2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004., pp. 235-244. IEEE, 2004.
16. Burgess, John, Brian Gallagher, David D. Jensen, and Brian Neil Levine. "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks." In Infocom, vol. 6. 2006.
17. Hui, Pan, Jon Crowcroft, and Eiko Yoneki. "Bubble rap: Social-based forwarding in delay-tolerant networks." IEEE Transactions on Mobile Computing 10, no. 11 (2011): 1576-1589.
18. Narayan, Vipul, and A. K. Daniel. "Energy Efficient Protocol for Lifetime Prediction of Wireless Sensor Network using Multivariate Polynomial Regression Model." Journal of Scientific & Industrial Research 81.12 (2022): 1297-1309.
19. Wang, Yu, and Hongyi Wu. "Replication-Based efficient data delivery scheme (RED) for Delay/Fault-Tolerant mobile sensor network (DFT-MSN)." In Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), pp. 5-pp. IEEE, 2006.
20. Dubois-Ferriere, Henri, Matthias Grossglauser, and Martin Vetterli. "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages." In Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, pp. 257-266. ACM, 2003.
21. Narayan, Vipul, and A. K. Daniel. "FBCHS: Fuzzy Based Cluster Head Selection Protocol to Enhance Network Lifetime of WSN." ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal 11.3 (2022): 285-307.

22. T. Spyropoulos, K. Psounis, C.S. Raghavendra, Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility, in: Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW’ 07), IEEE, pp. 79–85.
23. Y. Zguira, H. Rivano, A. Meddeb, Iob-dtn: A lightweight dtn protocol for mobile iot applications to smart bike sharing systems, in: 2018 Wireless Days (WD), IEEE, pp. 131–136.
24. Borah S.J., Dhurandher S.K., Woungang I., Kumar V. A game theoretic context-based routing protocol for opportunistic networks in an iot scenario Comput. Netw., 129 (2017), pp. 572-584
25. Narayan, Vipul, et al. "To Implement a Web Page using Thread in Java." (2017)
26. Irfan, Daniyal, et al. "Prediction of Quality Food Sale in Mart Using the AI-Based TOR Method." Journal of Food Quality 2022 (2022).
27. Awasthi, Shashank, et al. "A Comparative Study of Various CAPTCHA Methods for Securing Web Pages." 2019 International Conference on Automation, Computational and Technology Management (ICACTM). IEEE, 2019.