Efficient Congestion Minimisation by Successive Load Shifting in Multilayer Wireless Networks

Md. Emdadul Haque[a,*], Faisal Tariq[b], Laurence Dooley[c], Ben Allen[d], Yan Sun[b]

[a] *Department of Information & Communication Engineering, Rajshahi University, Bangladesh*
[b] *School of Electronic Engineering & Computer Science, Queen Mary University of London, United Kingdom*
[c] *Department of Computing & Communications, The Open University, United Kingdom*
[d] *Department of Engineering Science, University of Oxford, United Kingdom*

Abstract

Congestion in wireless networks is one of the major causes of system inefficiency, and with router load being the main contributor to overall network traffic flow, congestion is very dependent on the level of router load and how it can be effectively managed. This paper presents a novel low-complexity *Successive Load Shifting* (SLS) technique for intelligently shifting router load between network routers by predicting the probability of congestion occurrences in the network and exploiting the topology to reassign load to minimise congestion. Crucially, SLS does not compromise the data rate in avoiding congestion and is able to be seamlessly embedded into existing protocols with only a small increase incurred in system overheads. The performance of SLS has been extensively tested and critically evaluated using the widely adopted TCP and UDP protocols, with results confirming both significant throughput gains and superior packet loss performance.

Keywords: wireless network, congestion prediction and detection, load shifting, congestion control, successive load shifting.

1. Introduction

While wireless networks are an increasingly attractive solution for communication purposes compared to their wired counterparts, their application and performance is often affected by a range of issues from propagation loss, poor indoor coverage and medium access interference, through to congestion. For example, congestion can be a consequence of the prevailing medium access policy that allows only one device to access the medium at any one time [1]. Other factors which can influence congestion occurrence include: collision, noise-related losses and dynamic changes in routing paths for message forwarding, especially in multi-hop networks. In practice however, congestion mainly occurs due to the complexity

*Corresponding author
Email address: `haque-ice@ru.ac.bd` (Md. Emdadul Haque)

of network organization and its related traffic patterns, which are highly dynamic in nature and difficult to predict. This means organising the network topology so congestion is either avoided or mitigated is a key design objective from a congestion management perspective.

Many wireless networking technologies exist [2] including *Wireless Local Area Network* (WLAN), being a widely-deployed, multi-layer architecture. Its nodes are the basic elements which are logically placed at the base of the network topology, with routers located in various layers of hierarchy as shown in Figure 1, with each node connected to a router. The routers are in turn connected to nearby nodes as well as neighbouring routers. Any two nodes in the network can communicate with each other in a multi-hop fashion through these routers, so if the network is considered as being formed of clusters, then routers act as *clusterheads* with their associated nodes forming separate clusters.

An alternative example of a multilayer network is the *Wireless Mesh Network* (WMN). Figure 1 shows a simplified multilayer wireless architecture, with the mesh routers typically undertaking functions like data forwarding and acting as an internet access gateway for connected nodes. Mesh clients usually include a range of devices like laptops, smart phones, pocket PC, IP phones and RFID readers. Cellular networks can also be considered as mesh networks with the base station acting as a router [3]. Any WMN router generally handles two classes of message load: i) *Router load* which refers to messages from those nodes connected to it, and ii) *Routing load* which are messages received from other routers for forwarding onwards to their destination. All control and data messages are categorised as either router or routing load, with the level of the latter varying considerably depending on a routers position in the routing path. In contrast, router load is a major constituent part of the total traffic flow so is significantly influential in terms of the occurrences of congestion in the network.

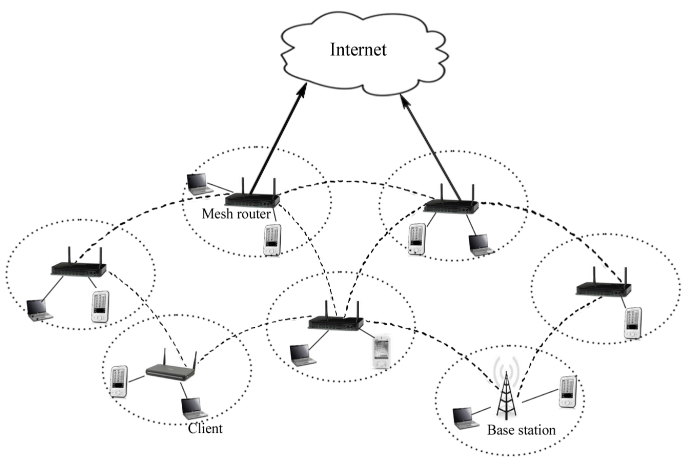


Figure 1: Simplified multilayer wireless network architecture

Traffic flow in a wireless network is dependent on the radio transmission scenario. For example, transmissions in the IEEE 802.11 *Medium Access Control* (MAC) cause interference to all nodes located within its coverage which inevitably impacts on traffic flow. This means congestion can occur not only due to its own traffic but also neighbouring routers' traffic load. To address this problem, various congestion control protocols have been proposed to exploit different network characteristics, with examples including: estimation of link state [4], [5], packet losses [6], *Explicit Congestion Notification* (ECN) [7], and *Active Queue Management* (AQM) marking [8].

Most existing congestion control protocols [4-9] are ineffectual in managing congestion re-occurrence as they rely on end-to-end control mechanisms. When network congestion is alleviated, nodes begin forwarding messages again at a minimum preset rate. This rate is then incrementally increased depending on whether the previous transmission was successful, however this approach eventually leads to a transmission rate being reached which again leads to congestion recurring. This provided the motivation to investigate the development of a new congestion reduction mechanism which to able to either prevent or minimise traffic congestion and its re-occurrence.

This paper introduces a novel low-complexity *Successive Load Shifting* (SLS) technique for congestion minimisation. Its key feature is that it can be seamlessly embedded into any existing congestion control and data transfer protocol incurring only a very small cost in the overall traffic flow. SLS dynamically moves router load between routers by means of an efficient *Cubic Spline-based Congestion Prediction* (CSCP) algorithm, which minimises congestion without crucially compromising the overall data rate. A mechanism for avoiding congestion re-occurrence is also integrated into the SLS model.

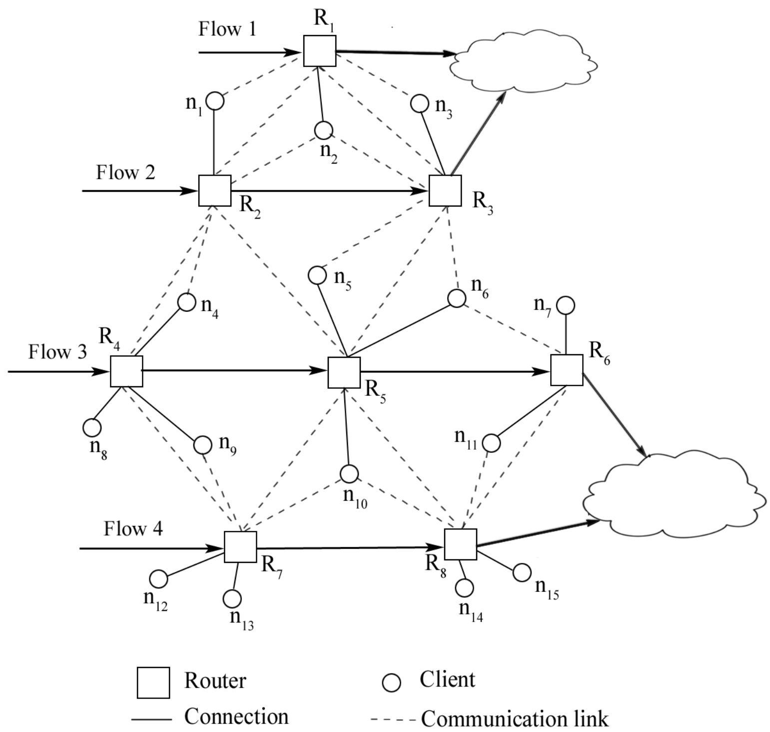


Figure 2: A simple network illustrating different flows

To critically evaluate the SLS performance in shifting router load from one router to either a single or multi-hop distant router, the algorithm was firstly implemented in the simple network shown in Figure 2, before being extended to more complex networks. In both cases, SLS was embedded into widely-accepted protocols including *Transport Control Protocol* (TCP) and *User Datagram Protocol* (UDP). Performance results for both scenarios corroborate that significant system performance improvements are achieved in terms of both throughput and packet loss rate. Furthermore, two variants of the SLS technique have been developed to ensure more effective congestion reduction which are applicable to a wide range of network situations.

The rest of the paper is organized as follows: Section 2 provides a detailed taxonomy of related congestion control methods. The new SLS technique is then formally introduced in Section 3, before a critical analysis of the performance results is presented in Section 4. Finally, Section 5 provides some concluding remarks and outlines possible future research directions.

2. Related Work

Congestion control protocols are mainly categorised as either rate control or re-routing based. Numerous methods [11-27] having been proposed to address the problem, with a comprehensive survey of the different protocols available for improving classical TCP in wireless multi-hop networks being presented in [9].

In rate control techniques, the underlying policy is that when congestion is detected the traffic rate is varied to control the congestion. Examples of such rate control algorithms include: TCP Adaptive Delayed-ACK Window (TCP-ADW) [10], Mobile Ad Hoc-TCP (MAD-TCP) [11], AIMD-based multipath congestion control [12], Multi-Armed Bandit Congestion Control (MABCC) [13], and early packet loss notification/best effort ACK delivery [14], with each employing a different strategy to eliminate congestion. In [10] for example, an adaptive delayed ACK scheme is proposed to reduce contention due to per packet ACK generation strategy of TCP. To achieve this a dynamically adjustment of the TCP receiver delay window is adopted based on channel condition. In contrast [11] proposes a cross-layer TCP enhancement schemes, referred as Mobile Ad Hoc TCP (MAD-TCP) and Adaptive-Dynamic Source Routing (A-DSR) schemes. The mechanism utilizes network events such as disconnections, channel errors, buffer overflow, and linklayer contention based on which the network adjusts the behavior of the network and regulates itself to all the network events. In these protocol as highlighted earlier, when a particular node reaches to the previously achieved data rate, congestion may reoccur.

In routing based congestion control techniques [15], [16], [17], the routers discover congestion free paths for diverting messages when congestion occurs. Authors in [15] propose a second-order joint congestion control and routing optimization framework that provides rate-optimality, queuing stability, fast convergence, and low delays. The paper implements second-order joint congestion control and routing framework based on a primal-dual interior-point approach that is well-suited for implementation in practical network systems. However, due to high complexity the algorithm does not scale well with the network size. Conversely, the load-balancing congestion adaptive routing protocol in [17] uses both traffic load density and lifetime associated with a routing path, as its decision metric. However, it requires regular parameter updates for the entire network which compromises the efficiency of this technique.

A recurring feature of these routing-based protocols is the need to regularly identify and update substitute routes whenever congestion occurs, with the corollary being that they generate considerable system overheads. Also if the alternate paths are significantly longer then delays occur and the number of messages increases commensurately with the number of hops. So, while these techniques manage network congestion they do so at the cost of increased end-to-end latency and lower throughput.

While various congestion control strategies have been proposed, from both a transport and network layer perspective, the issue of the physical load upon a router has largely been ignored. The main drawback of existing congestion control techniques is that they all compromise packet loss and throughput performance. To solve this problem, this paper proposes an innovative congestion reduction model which shifts routers physical load, without recourse to compromising the packet loss and data rate. The new SLS technique will now be discussed in detail.

3. Successive Load Shifting (SLS) Paradigm

The SLS paradigm comprises three distinct steps:

1. When a router first detects congestion, it attempts to shift some of its load by identifying neighbouring congestion-free routers and requesting them to accept the new load.
2. The congestion-free router predicts the likelihood of congestion occurrence if it accepts the new load before deciding whether to accept or decline the load shifting request.
3. The congested router shifts certain nodes to an alternative router provided it satisfies two key conditions: i) the destination router is congestion free, and ii) the router lies within the relevant nodes radio range. Multi-hop node shifting can be achieved in an analogous manner.

The complete SLS process is illustrated in Figure 3(a) and 3(b), where router R_1 shifts node n_3 to R_2 and R_2 shifts n_5 to the congestion-free router R_3. The dotted line shows the availability of the router within the radio coverage of a node, while the solid line represents an existing connection.

In the next section, both the congestion prediction and detection mechanisms of the SLS model are described, before the actual load shifting process is detailed.

3.1. Congestion Detection Technique

Network congestion mainly occurs when a set of nodes receive more messages than they are able to either deliver or forward, in a particular area of the network. The first task therefore for any congestion control algorithm is to identify such circumstances.

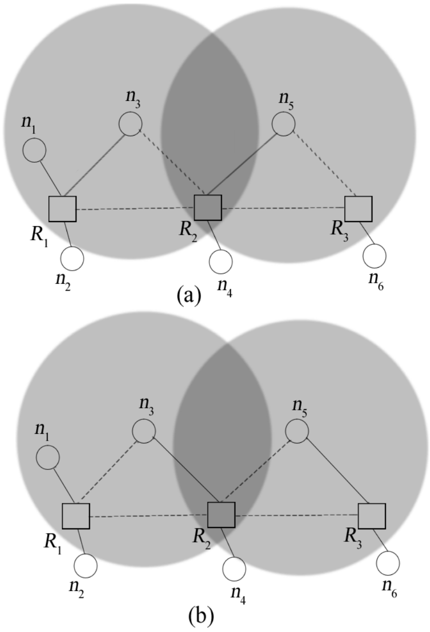


Figure 3: Multi-hop load shifting: (a) before and (b) after the load shifting.

Various congestion detection techniques have been proposed which generally involve monitoring different aspects of network characteristics such as, the average number of retransmissions [18], the mean packet loss recovery time [19], the channel utilisation [20], and the weighted moving average of the queue size [21], [22]; , average queue size and dynamic threshold based [23], queue size, link layer contention and signal strength [24]. For example, in [18] the number of retransmissions reflects that the queue limit for certain nodes has been exceeded and packets probably dropped, while in [22], a node monitors the queue size and detects a congested state if the available space in the queue falls below some predefined threshold. Moreover in this latter technique, no cognisance is made of either the message arrival or departure rates.

A recurring feature of these existing techniques is that they are reactive, i.e. they detect congestion after its actual occurrence. In contrast, the new SLS technique is preemptive, with load shifting taking place before congestion detection by predicting its occurrence so avoiding the initiation of a congestion control protocol. A primary indication of congestion is an increasing message arrival rate allied with a decreasing message departure rate leading to a longer queue size. In SLS all three of these parameters are combined to provide an accurate detection mechanism for load shifting based upon a prediction of the likelihood of congestion.

Assume the average message arrival rate of the queue of router R_i is λ_{R_i} and the average message service time is T_{R_i}. The message service time is defined as the average time elapsed from the arrival of a message until it successfully leaves the queue, so the message interarrival time is $\frac{1}{\lambda_{R_i}}$ and message departure rate is $\frac{1}{T_{R_i}}$. If the queue length q_{R_i}, of the i^{th} router R_i exceeds a predefined threshold q_{H_i}, then the router is flagged as congested if the following relationship holds:

$$\frac{q_{R_i}}{Q_{R_i}} > q_{H_i} \quad and \quad \lambda_{R_i} > \frac{1}{T_i} \tag{1}$$

where Q_{R_i} is the total queue length of R_i. If q_{R_i} exceeds the threshold while the message departure rate is less than the message arrival rate, then the likelihood of messages being dropped from the queue within a very short period will become much higher. Conversely, if the departure rate is greater than the arrival rate, then q_{R_i} will reduce and the router is deemed congestion free. If however, R_i experiences message drop from the queue, irrespective of the reason, it will then be detected as congested.

3.2. Congestion Prediction Technique

Before receiving any load from other routers, a router must firstly predict the possibility of congestion if it accepts this load since after receiving a node from the source router, it may become congested. Thus every router must determine its own congestion probability before accepting any new load. The congestion prediction technique for SLS consists of two distinct operational stages. The first is the departure rate-based congestion prediction scheme which is used by the routers between the initiation of the operation and the occurrence of the first congestion. The second stage uses an efficient congestion prediction technique based on cubic splines and exploits the dataset from the first stage and operates for the remainder of the networks lifetime. Both stages will now be discussed in detail.

3.2.1. Departure Rate Based Congestion Prediction

This is the most straightforward approach to congestion prediction and is suitable for the initial stage of operation when no *a priori* information is available on congestion occurrences. In this technique, the router predicts congestion by comparing message departure rates, aggregated message arrival rates and the predicted rate due to new load. If it is assumed R_i wishes to shift a connected node n_{R_i} to another router R_j and this node generates M_{R_i} messages per unit time, then a congestion free router R_j will accept the node only when the following condition is satisfied:

$$\frac{1}{\lambda_{R_j} + M_{R_i}} \geq T_{R_j}. \tag{2}$$

If R_j fails to uphold (2) then it predicts the probability of congestion before receiving the node. If more than one request for node shifting is received at a time, then a congestion free router calculates the total router load it is able to receive according to the following process:

Suppose R_j can receive a maximum of $N_{R_{ij}}$ nodes from a neighbouring router R_i then:

$$\frac{1}{\lambda_{R_j} + (N_{R_{ij}} \times M_{R_i})} \geq T_{R_j}. \tag{3}$$

So,

$$N_{R_{ij}} = \left\lfloor \frac{1 - (T_{R_j} \times \lambda_{R_j})}{T_{R_j} \times M_{R_i}} \right\rfloor \tag{4}$$

where $\lfloor a \rfloor$ indicates the smallest integer.

Similarly, a congested router may need to shift either one or more nodes to another router(s). The router calculates the number of nodes necessary to be shifted to avoid congestion as follows:

Assume R_i needs to shift N_{R_i} nodes to other routers then:

$$T_{R_i} - \frac{1}{\lambda_{R_i} - \left(N_{R_i} \times M_{R_i}\right)} \leq 0. \tag{5}$$

So, N_{R_i}can be expressed as:

$$N_{R_i} = \left\lceil \frac{\left(\lambda_{R_i} \times T_{R_i}\right) - 1}{M_{R_i} \times T_{R_i}} \right\rceil \tag{6}$$

where $\lceil a \rceil$ indicates the largest integer.

Note, (6) is only valid when the relationship between the arrival and departure rates is linear. As the arrival rate increases however, depending on the system capacity, at some point this relationship will become nonlinear. In these circumstances, a nonlinear congestion predictor is more appropriate and this provided the rationale for the CSCP mechanism [25], which will now be introduced.

3.2.2. Cubic-Spline Based Congestion Prediction

This technique requires a CSCP graph to be constructed of the message arrival time vs. message service time of a router as depicted in Figure 4. Each router collects data during network operation and periodically updates the graph, which means the router is aware of the current network status in real time and ensures node changes are reflected in the corresponding CSCP graph. When the message arrival time is small (i.e., high arrival rate), then the message departure time is high (as the departure rate is low) and vice versa.

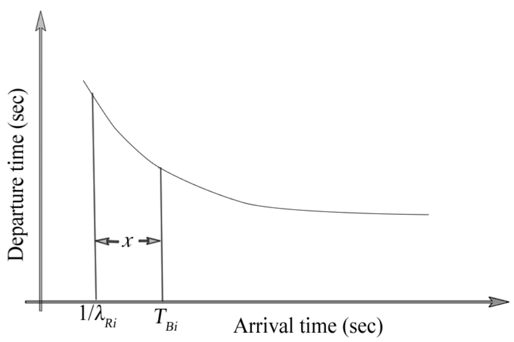


Figure 4: Example CSCP graph showing the relationship between message arrival and departure times

In the second stage, router R_i firstly finds the balancing point T_{B_i} shown in Figure 4, where the arrival and departure times are the same. This is achieved by means of a binary search algorithm [26]. In this algorithm, the total interval which is actually the bandwidth of routers, is equally divided repeatedly until T_{B_i} is found. Each router then calculates the number of nodes required to be shifted (for congested router) and the maximum number of nodes it can receive (for congestion free router) by exploiting the curve, which reflects the routers historical behaviour at different message arrival rates. Using the CSCP curve, each router can determine the number of nodes required to be either shifted or able to be received, as follows:

Suppose at a particular time instant, a congested router R_i has a message arrival rate λ_{R_i}. To reduce congestion, R_i needs to increase the message arrival time by x as shown in Figure 4, so that:

$$\frac{1}{\lambda_{R_i}} + x \geq T_{B_i}. \tag{7}$$

If R_i needs to shift N_{R_i} nodes (router load) to decrease λ_{R_i} and each node of N_{R_i} generates M_{R_i} messages per second, then:

$$\frac{1}{\lambda_{R_i} - (N_{R_i} \times M_{R_i})} \geq T_{B_i} \tag{8}$$

(8) can be rewritten as:

$$N_{R_i} \geq \frac{(T_{B_i} \times \lambda_{R_i}) - 1}{T_{B_i} \times M_{R_i}} \tag{9}$$

from which the maximum number of nodes to be shifted can be determined as:

$$N_{R_i} = \left\lceil \frac{(T_{B_i} \times \lambda_{R_i}) - 1}{T_{B_i} \times M_{R_i}} \right\rceil. \tag{10}$$

A congestion-free load receiver router estimates the number of nodes it is able to receive before accepting any new load in a similar fashion. For a congestion free router R_j, the point $\frac{1}{\lambda_{R_j}}$ will be greater than T_{B_j} as in Figure 4, so the corresponding relationship can be expressed as:

$$\frac{1}{\lambda_{R_j} + (N_{R_{ij}} \times M_{R_j})} \geq T_{B_j}. \tag{11}$$

This can be rewritten as:

$$N_{R_{ij}} \leq \frac{1 - (T_{B_j} \times \lambda_{R_j})}{(T_{B_j} \times M_{R_j})}. \tag{12}$$

So $N_{R_{ij}}$ is now given by:

$$N_{R_{ij}} = \left\lfloor \frac{1 - (T_{B_j} \times \lambda_{R_j})}{(T_{B_j} \times M_{R_j})} \right\rfloor. \tag{13}$$

In the proposed departure rate based congestion prediction technique, the current message departure rate is assumed to be the maximum data forwarding rate, with $N_{R_{ij}}$ and N_{R_i} given by (4) and (6) respectively. For SLS, the departure rate is determined using the CSCP approach, with N_{R_i} and $N_{R_{ij}}$ given by (10) and (13) respectively.

3.3. The SLS Technique

The prime aim of SLS is to enable any congested router in a network to identify potentially congestion-free routers in order to handover excess load. The process is triggered by a router R_i detecting congestion whereupon, it tries to find either one or more congestion-free routers to shift its connected load without any end-to-end communications. The notation adopted in this section is described in Table 1.

Table 1: The notation used in the SLS technique.

Z_i	Set of nodes connected to router R_i.
$\xrightarrow[R_iR_j]{Z_i}$	R_i sending a request to its neighbour router R_j to receive some of its nodes.
$C_{ij} \subseteq Z_i$	Set of nodes belonging to R_i that fall within the coverage of R_j.
$z_{ij} \subseteq Z_i$	Number of nodes R_j can receive from Z_i
$\xleftarrow[R_iR_j]{z_{ij}}$	Reply message from R_j to R_i
$\|S\|$	Cardinality of set S
$\sum_i C_{ij} = C_j$	Set of all nodes within the radio range of R_j but not connected to it.

The router firstly determines N_{R_i} using either (6) or (10) before commencing information sharing with neighbouring routers to locate congestion-free routers which are able to receive router load. The router initiates this by sending a request message as to all its neighbour routers. This message includes a series of lists relating to the connected nodes (Z_i), the nodes under communication range (C_i), neighbouring routers, the number of nodes necessary to be shifted (N_{R_i}) and the maximum searching depth limit (h_i). A neighbouring congestion free router R_j, sends a reply message as $\xleftarrow[R_iR_j]{z_{ij}}$ indicating the availability of spare load capacity provided. If however, R_j cannot receive all the load, i.e., $N_{R_i} \geq N_{R_{ij}}$ due to either traffic load in R_j or the absence of nodes within the requisite communication range, then it responds to R_i as well as forwarding the request onto its neighbouring routers, provided $|C_{ij}| \geq R_{ij}$ and R_j lies within the search depth limit. If a neighbouring router of R_j can receive some additional load then it sends a reply message to R_j. Finally R_j sends another reply message to R_i.

The reply messages each router sends include various information such as the message arrival rate (λ_{R_j}), the message service time (T_{R_j}) and queue length (q_{R_j}). Using these information the request sender can identify those free routers which are able to receive a maximum number of nodes. The router also sends the list of nodes it can receive (Z_{ij}). Note that, R_i only receives reply messages if both the congestion free routers and load shifting path are available within the depth limit.

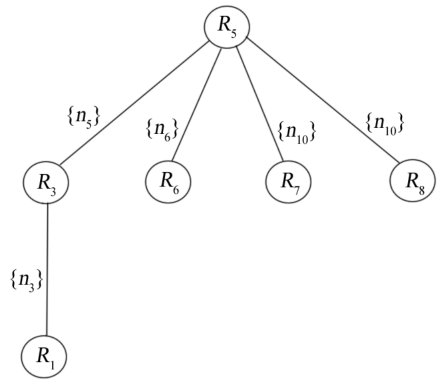


Figure 5: The breadth first search tree generated by R_5

From the reply messages, R_i determines the number of free-space available at receiver router for the nodes connected to R_i. It then selects the routers which respectively have the highest hop distance and minimum load at that depth, by means of the following relationship:

$$\alpha_1\left(\frac{q_{R_i}}{Q_{R_i}}\right) + \alpha_2\left(\frac{1}{\lambda_{R_i} \times T_{R_i}}\right) \tag{14}$$

where α_1 and α_2 are constants.

To illustrate this idea, assume in Figure 2 that at a particular time instant, R_5 is congested with $N_{R_5} = 2$. It thus sends a request message $\xrightarrow[R_5R_j]{\{n_5, n_6, n_{10}\}}$ in order to search for congestion-free routers. If R_3 is congestion-free at that time and can also receive router load with $|z_{53}| = 1$, then it sends reply message $\xleftarrow[R_5R_3]{\{n_5\}}$ with $z_{53} = \{n_5\}$ to R_5. Upon receipt of this message, R_3 forwards the request message to its neighbours. If neighbour R_1 can now for example, receive router load (n_3) it sends a reply message to R_3. R_3 finally sends another reply message to R_5 with $z_{53} = \{n_6\}$. z_{52} of $\xleftarrow[R_5R_2]{\{\phi\}}$ is empty, hence R_2 sends a reply message with with $z_{52} = \{\phi\}$ which means R_2 cannot receive any nodes from R_5 because all the nodes connected to R_5 are out-of-radio range of R_2. In a similar way R_6, R_7 and R_8 send the respective reply messages: $z_{56} = \{n_6\}$, $z_{57} = \{n_{10}\}$ and with $z_{58} = \{n_{10}\}$.

After receiving all reply messages, R_5 generates a breadth first search tree (as shown in Figure 5), to identify the possible load receiver routers and shifting paths. An ordered list of the free nodes for R_5 is created, with the destination routers ranked according to their depth. All routers with the same depth then apply (14) to rank them in ascending order.

The router then determines how many routers are required from the ordered list to shift the required amount of load and starts allocating the load accordingly. This process terminates when $\sum_{all j} N_{R_{ij}} \geq N_{R_i}$.

A router is not allowed to shift load to another router on both its up and downstream routing paths as this will contribute towards congestion. As a consequence, load is only shifted to a router which is not part of its routing paths in either direction. So in Figure 5 for example, since R_6 is located on the routing path of R_5, load will not be shifted to R_6. The tree nodes indicate either congestion-free routers or routers on the path to the congestion-free router, while an edge indicates that a parent router (i.e., R_3) has a connected node (n_3) that is within the communications range of a child router (R_1). During tree exploration, the router stores information on the load shifting path, nodes to be shifted along the path, and the corresponding destination router.

To shift one or more nodes, router R_i executes the following sequence: Firstly, it sends a load shifting request message to the destination router with the identification (ID) of the node to be shifted. If the destination router is able to receive this node, it replies with a positive acknowledgement. After receiving the acknowledgement, R_i then instructs the node to disconnect itself and reconnect to the neighbouring router which is made available to the node.

If it is required to shift load to a router located more than one hop away, R_i sends a load shifting request message to the destination router. The router also sends the same message to all other routers along the shifting path, which includes the candidate node ID. This single message is sufficient to pass all the requisite information to allow all routers along the path to determine both the shifting candidate node and destinations router. When R_i receives a positive reply from a receiver, it sends another message to start the shifting process. After receiving the message, all routers on the path shift the node to neighbouring routers according to the one-hop shifting process illustrated in Figure 3. Some major characteristics of the SLS are now analysed below:

3.3.1. How SLS ensures that participating nodes will not be congested?

In the load shifting process of SLS, three kinds of router nodes are involved: the load sender, load receiver and load exchanger (for more than one hop load shifting). Since the load sender is the congested router, it initiates SLS to reduce its incoming message rate thereby avoiding congestion, provided that the sender can shift the required number of nodes. If R_i is the load sender, then it transmits a request $\xrightarrow[R_iR_j]{Z_i}$ as in Section 3.3. A congestion free router R_j (load receiver) replies with $\xleftarrow[R_iR_j]{z_{ij}}$, where $|z_{ij}| \geq 1$.

To reply, R_j finds $|z_{ij}| \geq 1$ when the current message inter-arrival time ($\frac{1}{\lambda_{R_j}}$) is much higher so the summation of $\frac{1}{\lambda_{R_j}}$ and the new client nodes message inter-arrival time remains above T_{B_j} as in Figure 4. As R_j is always congestion-free provided this sum is greater than T_{B_j}, then the load receiver router will not be congested after receiving the new load.

To exchange load, a router disconnects some client nodes from one side and connect the equal number of nodes from another neighbouring router (please refer to Figure 3). Since the newly connected client nodes will generate the same number of messages as the disconnected nodes, the total message inter-arrival time remains unchanged which ensure the router will not be congested.

3.3.2. How SLS prevents congestion re-occurrence?

In SLS, a router shifts its physical load (client node) to either one or more hop distant neighbouring router(s). Assume in a SLS process that R_i shifts load to $h-$hop distant router R_k through R_j. In this process R_i shifts N_{R_i} client nodes. If load shifting is either not or only partially possible i.e., $N_{R_i} > \sum_{all j} |z_{ij}|$, then the immediate upstream router of N_{R_i} can become congested very quickly, so the upstream router will then initiate SLS. This continues until the start of the flow as shown in the example in Figure 6 where R_8 initiates the SLS process.

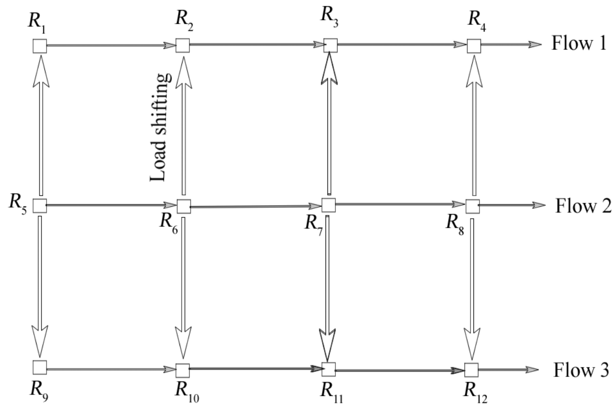


Figure 6: Sequence of the SLS process

If all upstream routers can shift load then the total message reduction in the flow is likely to be sufficient to avoid congestion at the initiating router. Note nodes participating in SLS will not encounter congestion so as shifted nodes messages are not forwarded through the previous flow path, the possibility of congestion reoccurring in the flow is negligible, especially for the same traffic.

3.3.3. How many nodes a router can shift in a specific area?

A router can shift client nodes of a specific area as stated earlier in this section. In this case we compute the maximum area of a network in the grid topology where a part of the network is shown in Figure 7.

In the proposed network configuration, many client nodes are located under the communication range of more than one router. For example, in Figure 7, the client nodes in area

A_1 lies within the radio range of both R_1 and R_3, so if nodes in this area are connected to R_3, then it can shift the nodes to R_1 provided it has spare capacity.

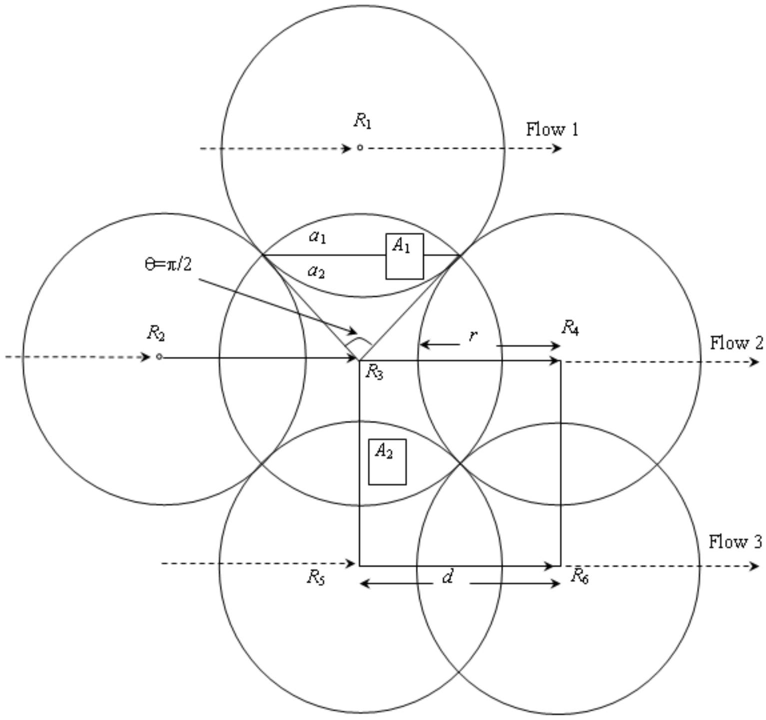


Figure 7: Client nodes coverage diagram

The area of the segment $A_1 = A_2$ and $a_1 = \frac{r^2(\theta - \sin\theta)}{2}$, where θ is in radians. When each client router lies within the radio range of at least one router, then $\theta = \frac{\pi}{2}$.

So, the area of $a_1 = a_2 = \frac{r^2}{2}(\frac{\pi}{2} - 1) = \frac{(\pi-2)r^2}{4}$ and therefore the area of $A_1 + A_2 = (\pi - 2)r^2$. If the traffic flows in the directions shown in Figure 7, then R_3 can shift the nodes of area A_1 and A_2. Hence a router can shift nodes from a total $(\pi - 2)r^2$ area of the network.

3.3.4. How much throughput gain can be achieved?

Since a router's load shifting area in a specific network has a theoretical upper bound when SLS is used, the achieved throughput increase similarly will have an upper bound, which will now be determined for a network having a grid topology.

If d is the length of each grid square of the topology in Figure 7, when a client node lies within the radio range of at least one router, the relationship between d and r can be determined as $d^2 + d^2 \leq 4r^2$ and so:

$$r = \frac{d}{\sqrt{2}}. \tag{15}$$

A router can shift a maximum load of $(\pi - 2)r^2$ area of the network, so the router can shift a maximum of $\frac{(\pi-2)r^2}{\pi r^2} \times 100\% = \left(\frac{\pi-2}{\pi}\right) \times 100\%$ of its load. This means even if all the network traffic flows become congested and each router can shift nodes from the $(\pi - 2)r^2$

area, then the overall throughput will increase by the maximum of 36.34%, provided (15) is uphold.

3.3.5. Does SLS affect the net traffic rate?

If all the requisite load shifting is assumed then the formerly congested router will not have to reduce its data rate, as the total network flow will be unchanged. Similarly no node involved in the SLS process will compromise its own data rate as analysed below:

For simplicity it is assumed that the packet generation rate of each client node is same. As discussed above, router nodes involved in the SLS process do not incur congestion, so client nodes shifting from one router to another need not to reduce the traffic rate. Note a node only lowers the traffic generation rate if congestion happens and a request comes from the router to which it is connected but not taking part SLS process. So, there is only one option in SLS to reduce the network traffic and that is when some nodes are disconnected.

Suppose using the SLS, R_i shifts load to h hop distance router R_k through R_j. In SLS, R_i shifts a subset of nodes $C_{ij} \forall j$ of Z_i. i.e. $(C_{ij} \subseteq Z_i)$ to R_k through R_j as described in Section 3.3. It is thus sufficient to prove that for all $i = i, i+1, i+2, \cdots, i+h-1$ and the subset C_{ij} remain connected. If $h = 1$ i.e., R_i shifts nodes (C_{ik}) to a one hop neighbor R_k, then to shift nodes, R_i sends a request message $\xrightarrow[R_iR_k]{Z_i}$ to R_k and R_k sends reply message $\xleftarrow[R_iR_j]{z_{ik}}$ where $(z_{ik} \subseteq Z_i)$.

After receiving the reply message R_i shifts the nodes in z_{ik} to R_k, and $(C_{ik} - z_{ik})$ nodes remain connected with R_i. So, for $h = 1$, the subset C_{ij} remains connected with routers. For $h > 1$, i.e., for $h = n$, it needs to be proven that for all n the subset $C_{i+n-1,j}$ remains connected. In the SLS process, for $n < h$, each R_{i+n} connects the set of nodes $z_{i+n-1,j}$ from R_{i+n-1} before that it disconnects the set of nodes $z_{i+n,j}$. The disconnected set $z_{i+n,j}$ will then be connected to router R_{i+n+1}, where $z_{i+n-1,j} = z_{i+n,j}$, since the router nodes other than the sender and receiver exchange the same number of client nodes. Hence no nodes become disconnected with the SLS process. Thus we finally conclude that SLS does not impact upon the overall traffic generation rate.

3.4. The Neighbour Load Shifting

The above scheme is predicated on both congestion-free router and load shifting paths. If neither is available then the load shifting process cannot progress. In these circumstances R_i requests neighbours to shift some of their load to other routers in the network so that they can allow R_i to forward more data in it's path. It may happen that the amount of available free space for accepting nodes is less than necessary i.e., $\sum_{all j} N_{R_{ij}} < N_{R_i}$. In these situations, R_i firstly shifts part of its load to the available spaces before sending a request to all of its neighbours, with the exception of the downstream router, to shift their load to other routers according to the following process:

Assume R_i has one router on the downstream routing path and has NB_{R_i} neighbours. In this scenario, R_i sends a request message to $(NB_{R_i} - 1)$ routers, to shift each of their

$\left\lceil \frac{N_{R_i} - \sum_{allj} N_{R_{ij}}}{NB_{R_i} - 1} \right\rceil$ loads. After receiving the request, the neighbour routers shift their load using the same method. To illustrate this process, assume R_7 in Flow 4 in Figure 2 is congested at a particular time instant and needs to shift $N_{R_7} = 1$ router load. Since R_7 is unable to shift any load to its neighbours, it sends a request message to both R_4 and R_5 to shift $\left\lceil \frac{1}{(3-1)} \right\rceil = 1$ unit of load. In this particular scenario, if any of the routers shift load then this will be sufficient for R_7, though R_7 must still request each neighbouring router because it has no *a priori* knowledge of their load shifting capacity.

As each router shifts nodes to a neighbouring router upon considering the traffic load, applying SLS ensures no router will be congested. Furthermore, none of the nodes will compromise the data rate, so if all the requisite load shifting is preformed then the formerly congested router will not have to reduce its data rate because the total amount of network flow remains unchanged.

3.5. Time complexity of SLS

If a message traverses a maximum of $h-$hops, the time complexity of SLS is $O(h)$. Now suppose each router has an average of R_n neighbouring routers, so the maximum number of nodes in the search tree rooted by the congested router is $\frac{R_n^{h+1}-1}{R_n-1}$, and the overall number of message exchange for SLS proces is in the order of $O(R_n^h)$. In the simulations, $h = 2$ is used with only one message being required for a congested router for each SLS process initialization. If SLS is able to shift the required load to eliminate the congestion, then no additional messages are required until congestion reoccurs.

In the next section, the performance of the SLS technique will be rigorously evaluated and a results analysis presented to corroborate its effectiveness.

4. Performance Evaluation and Analysis

4.1. Experimental Setup

To critically analyse the performance of the new SLS technique, a simulation network environment was constructed on an *ns-2* platform, which is widely used for network performance evaluation [27], with key parameter settings being summarised in Table 2. While an area of *400 m X 500 m* was chosen for all simulations, the results are valid for any area size provided the node and router deployment densities per unit area remain the same. Two distinct network scenarios were considered in appraising the SLS model.

The simple network topology configuration in Figure 2 is used to demonstrate the proof-of-concept for SLS. The scenario is similar to [28] and comprises 8 routers and 15 nodes. R_1, R_3, R_6 and R_8 are the internet gateway routers with the flows of the network shown in Figure 2.

A larger complex deployment comprising 30 routers and 180 nodes, with client nodes being distributed according to a uniform random distribution in the simulated area in Figure 8. In contrast, router nodes are deployed by means of a specialized pseudo random distribution. In a square grid of size of *100 m X 100 m*, the routers are deployed according to a normal distribution with $\sigma = 7.5$ m along both axes to match realistic scenarios. In Figure 8, the black, red and blue circles respectively represent the routers, nodes having only one router within its radio range, and nodes with more than one router within their communications range. This means in the case of congestion, blue nodes have multiple connection options so they can easily be shifted to alternate neighbouring routers if they satisfy the above conditions. In this scenario, the four corner routers are gateway routers, with all routers choosing the closest gateway router for traffic flow.

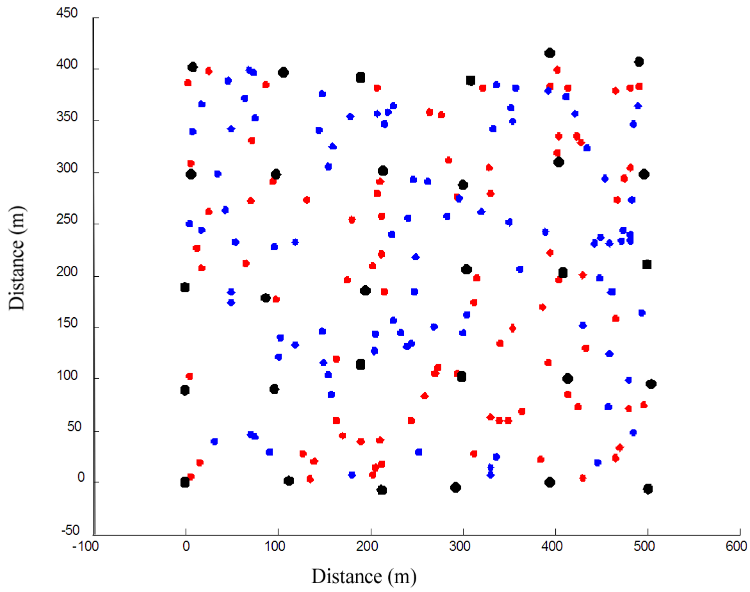


Figure 8: The complex network deployment scenario (distance in metres)

4.2. Dataset Collection

As stated in Section 3, the CSCP algorithm requires the dataset of both message arrival and corresponding message departure times to be determined. To calculate the former, each router counts the number of messages arriving in the queue per second. Similarly, each router calculates the message departure time by counting the number of messages leaving the queue and averaging the time difference between arriving and successfully departing messages. The message arrival and corresponding departure times are then recorded every second. This approach to dataset collection is inefficient and requires large memory space, so to reduce the required storage, dataset records were limited to 10 equidistant data bins ranging between 0 Kbps and 2 Mbps (peak data rate), with each bin covering a 200 Kbps range. For example, if the data arrival rate at a certain time is between 400 Kbps and 600 Kbps, then it is recorded in the 3rd bin. If another arrival rate within the same range is subsequently found then it is averaged with the previous bin value and the new averaged value recorded. Each router continues to collect data until the first occurrence of congestion

Table 2: Summary of network environment parameter values and Settings.

Parameter		Value
Area size		$400mX500m$
Total queue length, $Q_{R_1} = Q_{R_2} = \cdots$		20
Queue load threshold, $q_{H_1} = q_{H_2} = \cdots$		80%
Number of hops (h_i)		2
$\alpha_1 = \alpha_2$		0.5
MAC		IEEE 802.11
MAC layer transmission rate		2 Mbps
Data set update interval		5 s
Simple Network		
	No. of routers	8
	No. of nodes	15
	Router transmission range	250 m
	Node transmission range	200 m
Complex Network		
	No. of routers	30
	No. of nodes	180
	Router transmission range	180 m
	Node transmission range	150 m

when a router enters the CSCP mode, while still continuing to update the data set at periodic intervals. At every 5 s interval, each router calculates the message arrival and departure rates for 1 s.

4.3. Performance Analysis of Simple Network

The first set of experiments was undertaken on the simple network scenario in Figure 2. To critically analyse the performance of the SLS paradigm, TCP and UDP agents are separately investigated and their respective performances evaluated with and without SLS. To clarify the terminology adopted in all figures in Section 4, whenever either TCP or UDP is stated this represents the original network protocols, whereas when SLS is used, this refers to the new technique being implemented along with either TCP or UDP.

Packet size has a direct influence on the throughput as well as the probability of congestion in a wireless network [29] since it determines how long the medium will be occupied by each user. Figures 9 (a) and (b) respectively display the impact of packet size on throughput and packet loss rate for TCP and SLS. It is evident that while TCP throughput increases with packet size, when SLS is applied, consistently superior performance is achieved for all packet sizes, with the improvement being especially notable for larger packet sizes where the probability of congestion is greater. On average, a 10% throughput improvement and 10% reduction in message loss is achieved when SLS is used in combination with TCP.

UDP is a connectionless protocol which does not have a congestion control mechanism, so when the bit rate increases, the likelihood of congestion is commensurately higher as is the

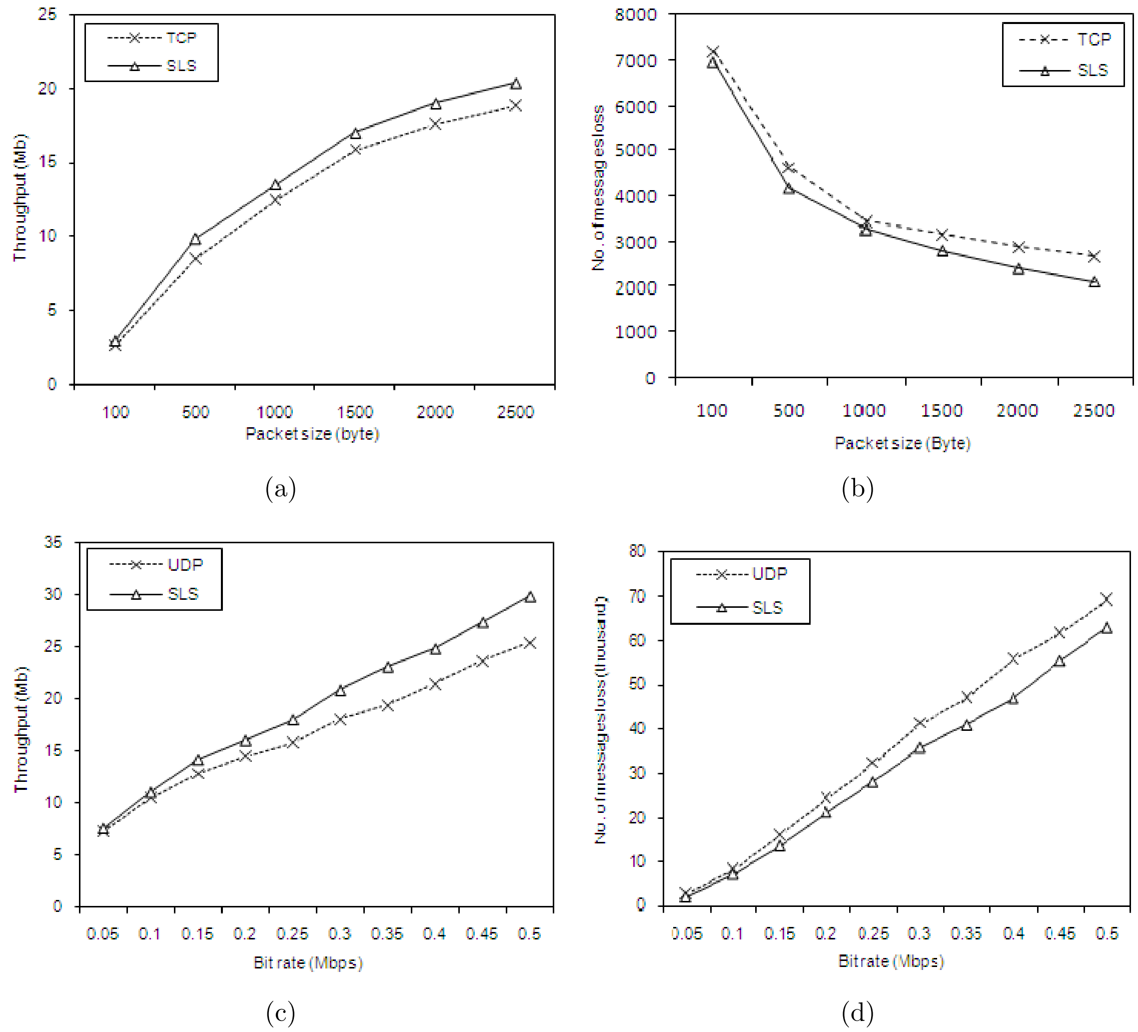


Figure 9: (a) TCP throughput over packet size (b) Total number of messages loss for TCP (c) UDP throughput over bit rate (d) Total number of messages loss for UDP

packet loss rate. This is because, regardless of the congestion situation, UDP tries to send packets at a constant rate. Figures 9 (c) and (d) show the respective performances for the UDP agent, with and without the SLS technique, for a bit rate varying between 0.05 Mbps and 0.5 Mbps. The packet size was fixed at 1KB. The corresponding UDP results in Figures 9 (c) and (d) reveal a significant improvement when SLS is used, with the most conspicuous performance being as the bit rate increases towards 0.5 Mbps, where SLS provides more than 5 Mbps throughput gain as a result of continually load shifting to lower congestion and enhance network performance. Similar SLS improvements are secured in the packet loss performance rate, with for example, at 0.4 Mbps, a 16% packet loss reduction attained from 56 K to 47 K.

Figure 9 also reveals that SLS performance gain is significantly higher for UDP then TCP, with on average a 13% throughput improvement and 13% reduction in message loss achieved. This can partly be accounted for by TCP possessing a congestion management mechanism, so the improvement in using SLS is as a consequence of preventing further reoccurrences of congestion. UDP in contrast has no such mechanism so gains achieved are directly due to either preventing the initial occurrence or reoccurrence of congestion, which is especially challenging at higher bit rates.

We know from previous analysis that the SLS upper bound throughput improvement is 36.34% which occurs in the ideal scenario where there are always routers available for load shifting. In real-world situations, this may not always be feasible and the performance improvement gradually reduces depending on the availability of free routers for load shifting. The achieved rate is thus normally much lower than the maximum, with in this simulated simple network case, the overall gain achieved by deploying SLS being ≈12%

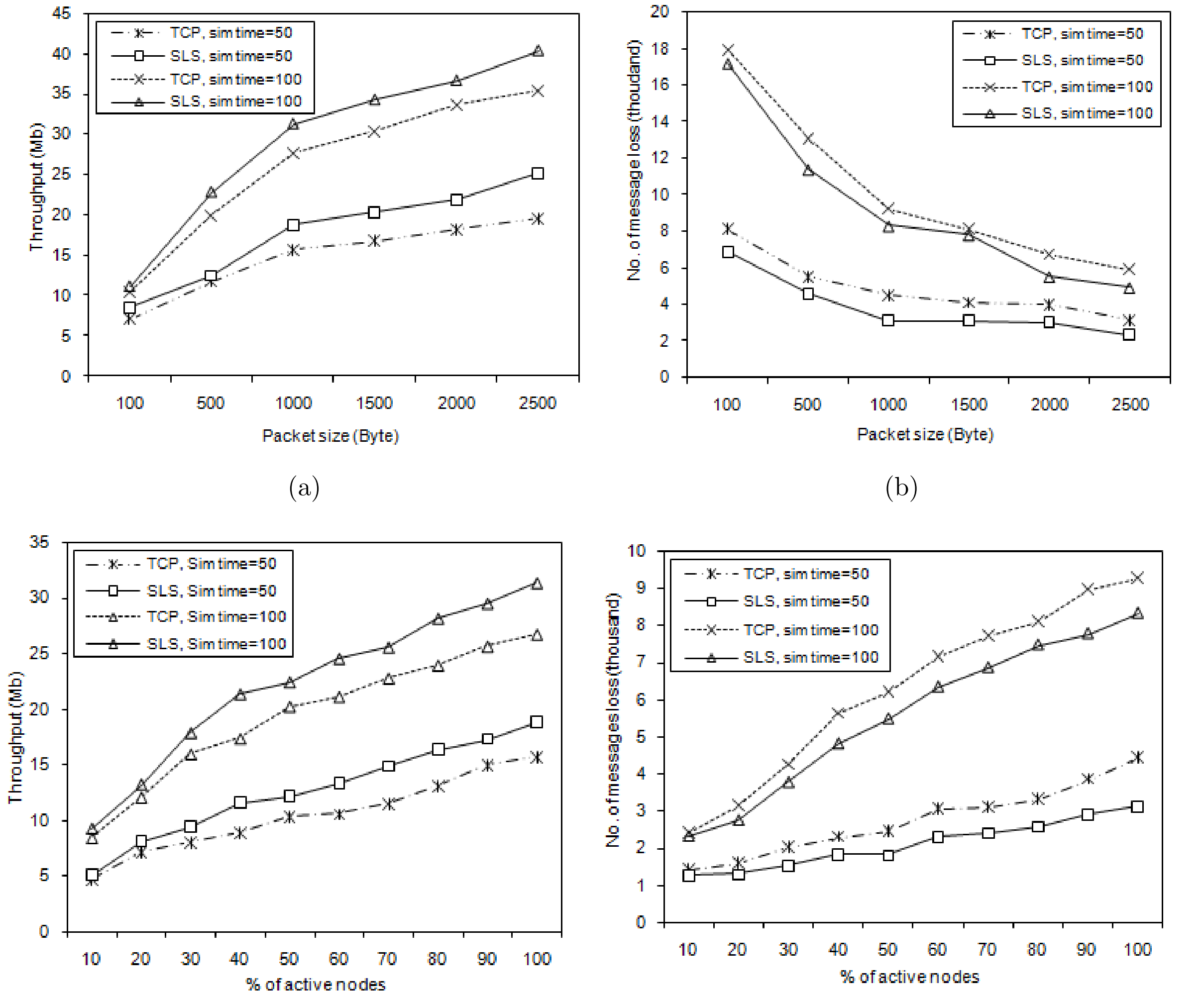
4.4. Performance Analysis of Complex Networks

The second series of experiments were designed to examine the performance of SLS in a complex network scenario, with much higher node and router densities compared to the first scenario. For TCP, a variable packet size between 100 B and 2.5 KB was used, with the corresponding throughput and message loss results, both with and without SLS being displayed in Figures 10 (a) and (b) respectively. The same scenario was simulated at two different time frames (50 s and 100 s) and similar trends has been observed. Both the throughput and message loss rates are noticeably improved for larger packet sizes as with the simple network, because the medium is occupied by a node for a longer time period so throughput is significantly increased.

In this scenario however, on average 15% of the throughput improvement is a consequence of the SLS technique. Figures 10 (c) and (d) show the results when the number of active nodes varies between 10% and a fully active set (100%), with all active nodes randomly selected. While the performance of SLS and TCP is very similar during low activity periods, since the likelihood of congestion is low, with an increasing percentage of active nodes, SLS progressively outperforms TCP by a significant margin. For example, in Figure 10 (d) when only 10% of nodes are active, the message loss difference between SLS and TCP is negligibly small, compared with around 20% improvement when all nodes are active. It needs to be stressed that unlike other congestion management mechanisms, SLS does not compromise the transmission rate when congestion is either detected or predicted, so both the throughput and message loss rates will be better in most cases.

In general, when SLS is embedded with TCP, an average 16% throughput improvement is secured in all the simulations for this complex network scenario, with analogous gains observed in the message loss rate. While the performance significantly improved for this scenario, there is scope for further enhancement in terms of achieving the theoretical upper bound. In reality, to achieve the upper bound, all the flows of the network must be congested and the corresponding routers able to shift the load.

Similar finding are observed for the corresponding UDP agent results in Figure 11, with SLS being evaluated for various bit rates and active node percentages in the network. A packet size of 1 KB was chosen for these experiments. Figures 11 (a) and (b) present the system throughput and message loss performance respectively for simulation times of 50 s t!]



(a) (b)

(c) (d)

Figure 10: (a) TCP throughput over packet size (b) Number of messages loss over packet size (c) TCP throughput over percentage of active nodes in the network (d) Number of messages loss over percentage of active nodes in the network

and 100 s, while varying the bit rate between 0.01 Mbps and 0.1 Mbps. At low bit rates, there is negligible performance difference between SLS and UDP, but at higher bits rates, SLS provides markedly higher throughput than UDP. At a bit rate of 0.1 Mbps per node for example, ≈ 15MB improvement in throughput is achieved by SLS, with on average, an 18% higher throughput observed across the range. The matching messages loss performance is displayed in Figure 11 (b), where 16% message loss reduction is achieved by SLS compared to UDP.

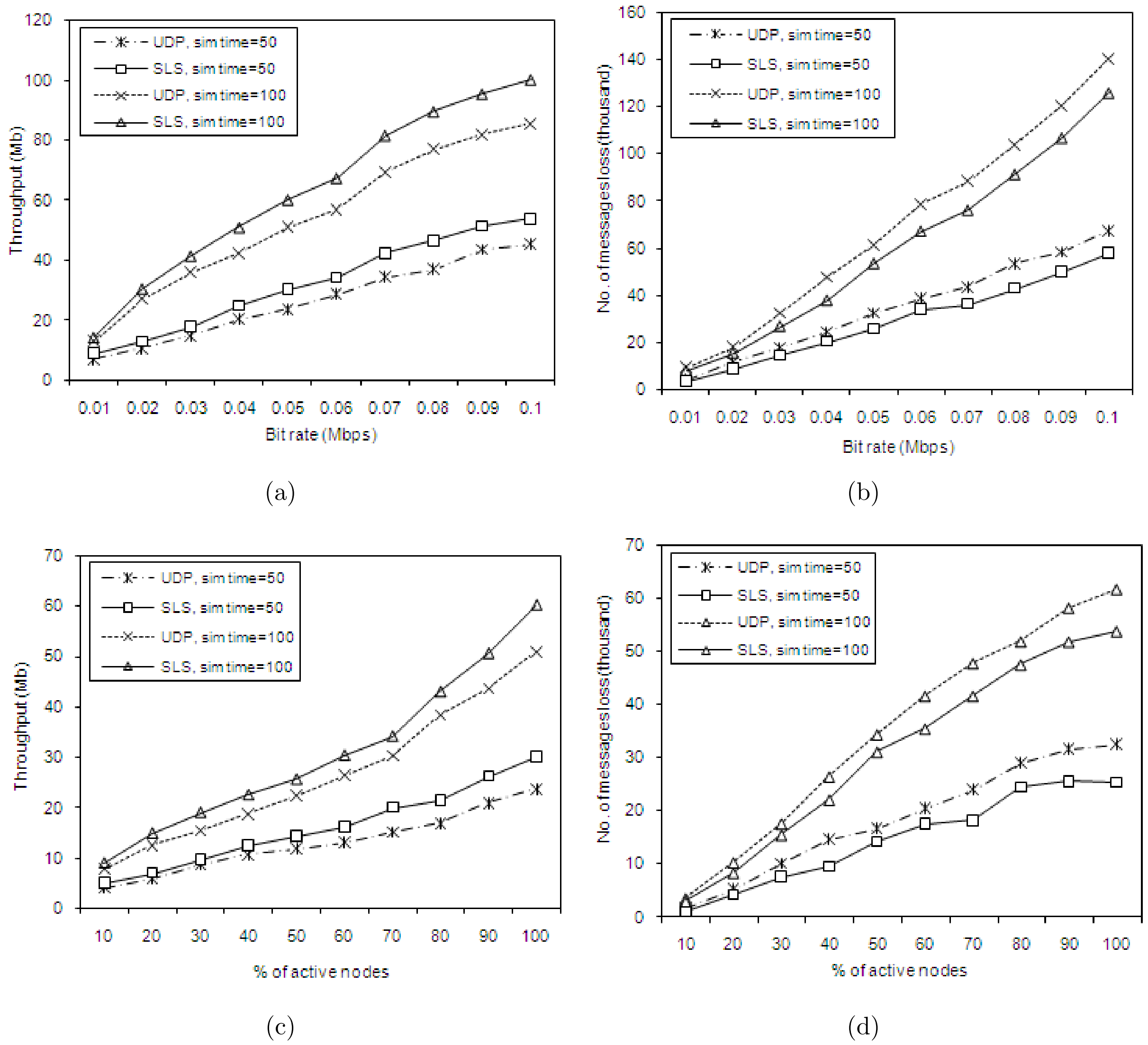


Figure 11: (a) UDP throughput versus bit rate (b) Number of messages loss over bit rate (c) UDP throughput over percentage of active nodes (d) Number of messages loss over percentage of active nodes in the network

Finally, the network was simulated with varying percentages of active client nodes, namely between 10% and 100% in steps of 10%. The respective throughput and message loss results are plotted in Figures 11 (c) and (d). The packet size and bit rate in both the simulations were fixed at 1KB and 0.05 Mbps respectively. For both simulation times, the throughput improvement between SLS and UDP gradually increases, with approximately 7 Mbps and 10 Mbps higher throughput achieved for the simulation times 50 s and 100 s respectively, when all network nodes were active. Significant performance improvement was also observed when message loss was evaluated, with SLS having nearly 8000 fewer message losses than UDP when the system was simulated for 100 s with 100% active nodes.

In summary, the results corroborate that irrespective of the network complexity and size, embedding SLS into TCP and UDP significantly improves both throughput and message loss performance. Additionally, as the SLS paradigm only shifts load by disconnecting and then reconnecting nodes, the technique can be integrated into any congestion control protocol, with crucially minimal additional cost being incurred in terms of the congestion management overheads. SLS also guarantees to uphold the data rate provided both a congestion free router and load shifting path exist.

5. Conclusion

This paper has presented a novel successive load shifting (SLS) paradigm to significantly reduce congestion in multilayer heterogeneous wireless networks. In contrast to conventional load shifting techniques, SLS does not compromise the bit-rate to manage congestion, but instead exploits network dynamics to shift some load from the congested router to neighbouring uncongested routers to minimise congestion. SLS also prevents the re-occurrence of congestion which is a major cause of throughput degradation and message loss in networks. A further benefit of SLS is it can be seamlessly embedded into any protocol with minimal system overheads.

Acknowledgement

Dr. Tariq would like to thank British Council's Researchers Link program for providing travel grant used to carry out part of this collaborative work.

References

[1] S. Kumar, V. S. Raghavan, J. Deng, Medium access control protocols for ad hoc wireless networks: A survey, Ad Hoc Networks 4 (3) (2006) 326 – 358.

[2] K. Pahlavan, Principles of Wireless Networks: A Unified Approach, 2nd Edition, John Wiley & Sons, Inc., New York, NY, USA, 2011.

[3] K. Son, E. Oh, B. Krishnamachari, Energy-efficient design of heterogeneous cellular networks from deployment to operation, Computer Networks 78 (2015) 95 – 106, special Issue: Green Communications.

[4] J. Perry, H. Balakrishnan, D. Shah, Flowtune: Flowlet control for datacenter networks, CSAIL Technical Report, MIT (2016) 1–15.

[5] X. Li, H. Yousefi'zadeh, Mpcp: Multi packet congestion-control protocol, SIGCOMM Comput. Commun. Rev. 39 (5) (2009) 5–11.

[6] H. Wu, Z. Feng, C. Guo, Y. Zhang, ICTCP: Incast Congestion Control for TCP in Data-center Networks, IEEE/ACM Trans. Netw. 21 (2) (2013) 345–358.

[7] B. Trammell, M. Kühlewind, D. Boppart, I. Learmonth, G. Fairhurst, R. Scheffenegger, Enabling internet-wide deployment of explicit congestion notification, in: J. Mirkovic, Y. Liu (Eds.), Passive and Active Measurement: 16th International Conference, PAM 2015, New York, NY, USA, March 19-20, 2015, Proceedings, Springer International Publishing, Cham, 2015, pp. 193–205.
[8] Z. Na, Q. Guo, Z. Gao, J. Zhen, C. Wang, A novel adaptive traffic prediction aqm algorithm, Telecommun. Syst. 49 (1) (2012) 149–160.
[9] C. Lochert, B. Scheuermann, M. Mauve, A survey on congestion control for mobile ad hoc networks: Research articles, Wirel. Commun. Mob. Comput. 7 (5) (2007) 655–676.
[10] A. M. Al-Jubari, M. Othman, B. Mohd Ali, N. A. W. Abdul Hamid, An Adaptive Delayed Acknowledgment Strategy to Improve TCP Performance in Multi-hop Wireless Networks, Wireless Personal Communications 69 (1) (2013) 307–333.
[11] H.-P. Chang, H.-W. Kan, M.-H. Ho, Adaptive TCP congestion control and routing schemes using cross-layer information for mobile ad hoc networks, Computer Communications 35 (4) (2012) 454 – 474.
[12] D. Zhou, W. Song, Y. Cheng, A study of fair bandwidth sharing with aimd-based multipath congestion control, IEEE Wireless Communications Letters 2 (3) (2013) 299–302.
[13] A. B. M. A. A. Islam, S. M. I. Alam, V. Raghunathan, S. Bagchi, Multi-armed bandit congestion control in multi-hop infrastructure wireless mesh networks, in: 2012 IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2012, pp. 31–40.
[14] X. Yu, Improving TCP performance over mobile ad hoc networks by exploiting cross-layer information awareness, in: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, MobiCom '04, ACM, New York, NY, USA, 2004, pp. 231–244.
[15] J. Liu, N. B. Shroff, C. H. Xia, H. D. Sherali, Joint congestion control and routing optimization: An efficient second-order distributed approach, IEEE/ACM Transactions on Networking 24 (3) (2016) 1404–1420.
[16] D. Tran, H. Raghavendra, Congestion adaptive routing in mobile ad hoc networks, Parallel and Distributed Systems, IEEE Transactions on 17 (11) (2006) 1294–1305.
[17] G. S. Tomar, L. Shrivastava, S. Bhadauria, Load balanced congestion adaptive routing for randomly distributed mobile adhoc networks, Wireless Personal Communications 77 (4) (2014) 2723–2733.
[18] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, M. Gerla, The impact of multihop wireless channel on TCP throughput and loss., in: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, Vol. 3, 2003, pp. 1744–1753.
[19] J. Paek, R. Govindan, RCRT: Rate-controlled reliable transport protocol for wireless sensor networks, ACM Trans. Sen. Netw. 7 (3) (2010) 20:1–20:45.
[20] K. Xu, M. Gerla, L. Qi, Y. Shu, Enhancing tcp fairness in ad hoc wireless networks using neighborhood red, in: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, MobiCom '03, ACM, New York, NY, USA, 2003, pp. 16–28.
[21] S. Rangwala, A. Jindal, K.-Y. Jang, K. Psounis, R. Govindan, Understanding congestion control in multi-hop wireless mesh networks, in: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, MobiCom '08, ACM, New York, NY, USA, 2008, pp. 291–302.
[22] B. Hull, K. Jamieson, H. Balakrishnan, Mitigating congestion in wireless sensor networks, in: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys '04, ACM, New York, NY, USA, 2004, pp. 134–147.
[23] T. Senthilkumaran, V. Sankaranarayanan, Dynamic congestion detection and control routing in ad hoc networks, Journal of King Saud University - Computer and Information Sciences 25 (1) (2013) 25 – 34.
[24] M. Anuradha, G. S. Anandha Mala, Cross-layer based congestion detection and routing protocol using fuzzy logic for manet, Wireless Networks (2016) 1–13.
[25] S. McKinley, M. Levine, Cubic spline interpolation, College of the Redwoods 45 (1) (1998) 1049–1060.
[26] T. Cormen, Introduction to Algorithms, MIT Press, 2009.
[27] T. Issariyakul, E. Hossain, Introduction to Network Simulator NS2, 1st Edition, Springer Publishing Company, Incorporated, 2008.

[28] A. Warrier, S. Janakiraman, S. Ha, I. Rhee, DiffQ: Practical Differential Backlog Congestion Control for Wireless Networks, in: INFOCOM 2009, IEEE, 2009, pp. 262–270.

[29] G. Xylomenos, G. Polyzos, TCP and UDP performance over a wireless LAN, in: INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 2, 1999, pp. 439–446.