University of Amsterdam

System and Network Engineering Master

Research Project

Effects of disruption and topology on the computer network packet delivery

Author:
Łukasz Makowski

Supervisor:
Marc Makkes

Thursday 18th August, 2016

Abstract

In the computer networks, the path a packet takes is often determined by the routing protocol. Its role is also the topology state changes detection (e.g. failed link) and propagating this information over the network. However, the reconvergence is not immediate, when it is ongoing the information used to route the packet may be invalid or missing. This research looks for improvements leading to the increased packet arrivals during reconvergence. We evaluate Barabási-Albert (BA), Watts–Strogatz (WS), Balanced Tree and Star models against established packet flow model. Our results show that for the most of the cases, BA enables the highest delivered packet ratio combined with the relatively short length of taken routes. Next, we partition the routing protocol domain, limiting the spread of updates originated after the link state change. For an unrealistically high probability of transient link failures, we observe the beneficial impact of routing domain partitioning on BA and WS topologies.

Contents

1 Introduction

Initially, the task of interconnecting two remote systems is an uncomplicated issue. After establishing the physical connection, two parties agree on a common network protocol (e.g. Internet Protocol). Since that moment, assuming the endpoints can reach each other protocol handle (e.g. IP address) the communication occurs. However, with the node amount increase, the number of possible paths between any two systems raise as well. The need for establishing a route brings new challenges into the computer network architecture. Ultimately, nodes are not directly interconnected with each other. Instead, they have to act as a relay for others. This requires intermediate nodes to forward the packets originated by the sender to the appropriate node so that it reaches its destination. Therefore, in order to send a packet across the network, node availability information has to be distributed among all nodes, allowing the node to select the shortest-path to the desired destination.

During the failure event (e.g. node/link failure) on the network, the role of the routing protocol is to detect the failure and spread the information to all nodes participating in the given routing protocol domain. Assuming the failure disturbed a node's currently used path, by receiving such notification it updates its local database with it and infers an alternative route. Nevertheless, the information does not propagate instantly. Depending on the characteristics and architecture of the network, as well as the routing protocol features this can take from 100 milliseconds [1] to 15 minutes [2].

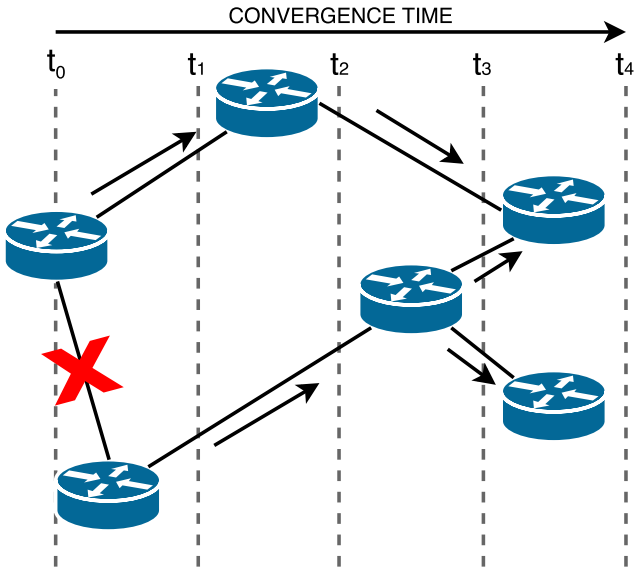


Figure 1: Routing protocol reconvergence after a link failure event.

Figure 1 illustrates the routing protocol reconvergence process. In the time t_0 a link connecting the two leftmost routers fails, which is perceived by the directly connected devices. After that, they sent the routing protocol updates (illustrated by the arrows parallel to the links). In the next time steps — t_1, t_2, t_3 the routing information travels across the network to notify all participating routers. However, only after routing information is fully propagated (t_4), every router has a complete and actual image of the network. The routing protocol required a time interval $t_4 - t_0$ to reconverge. In this moment in time, not all the nodes are equipped with the factual knowledge about the path availability within the network. Due to this fact, the packets can be routed to the intermediate node, which has an inoperative route to the packet destination. As specified by Labovitz et al. [3] the routing protocol reconvergence may result in connectivity interruptions and increase of packet loss and latency.

We analysed the impact of topology structure and the routing protocol update propagation area limiting, on the network packet delivery during its reconvergence period. This research aimed to answer the following questions:

- In the defined packet flow model, which of the tested topologies show the best packet delivery qualities? Is that consistent across the different topology sizes?
- How does the limiting of routing protocol update area influence the topology?

First, we introduced a model describing a packet transfer over a network governed by the shortest-path driven link-state routing protocol. We induced its reconvergence by introducing a probabilistically determined link state changes. The experiment allows assessing the behaviour of certain topology consisted of arbitrary selection of sender and recipient nodes. Next, a single packet was sent over an established shortest-path. During the transfer we tracked if the packet was delivered and if not what was a cause of dropping it. Moreover, the successful transfer was characterised by number of hops the packet traversed.

This paper is structured as follows. First, in Section 3 the used graph models are described, including the rationale for selecting these for the research. Next, in Section 4 we present the established network model. Whereas, in Section 5 the details on the simulator implementation are given. Moreover, extended with an overview on metrics we used and the structure of conducted experiments. The section 6 presents the results divided into two parts — an analysis of topologies packet delivery without routing domain partitioning and with 2, 4 and 8 partitions. Finally, the outcome discussion is performed in section 7, followed by the conclusions (Section 8).

2 Related work

The vast majority of routing protocols used in the computer networks determines the route using shortest-path algorithms (e.g. Dijkstra [4]). Wang and Crowcroft [5] identify consequences of the shortest-path based routing protocol has on the network. It is shown that in the dynamically changing environment such protocol will eventually cause performance degradation affecting data plane traffic. Authors argue this happens because routing protocols are often unable to converge into a stable topology. Moreover, they state the potential solutions to mitigate the issue. These are a better estimation of network distance to avoid loops and usage of multi-path route selection algorithms to utilise more links. Additionally, instead of relying on protocol convergence, a node should have sufficient amount of information to immediately infer the alternative path, without the need to receive external updates.

Nevertheless, modifying the routing algorithm is not the only approach aiming at enhancing network delivery efficiency under disruption. The scientific community has put significant interest into understanding the effects of topology type on the data transmission. The paper by Crucitti et al. [6] shows that BA network is well tolerant to the random node failures; however, the targeted attacks severely decrease its efficiency. Whereas for the Erdős–Rényi (ER) model both types of failures present much more uniform results. To evaluate the network performance, an inverse of shortest-path length between the two chosen nodes (efficiency metric [7]) is used. The failure is defined as a deletion of randomly determined node, whereas the attack is a targeted node deletion based on the metrics defining nodes significance (degree, betweenness, load). The authors explain the results by bringing models degree distribution metric. In this sense BA model is heterogeneous (i.e. there are few nodes with a significantly higher degree than the rest), therefore the attack deleting these eventually paralyses the network connectivity. On the contrary, ER model degree distribution is homogeneous, which results in the lack of such mission critical nodes as in BA case.

Another concept evaluated in the area of network research is an analysis of behaviour corresponding to Stochastic Resonance (SR) phenomena [8]. SR describes the effect when distortion in the form of noise improves the actual signal shipment. Czaplicka et al.[9] discover the beneficial role of noise on the data delivery within the hierarchical networks with communities (groups of nodes). The paper defines two types of noise:

- dynamic (q)

  It steers the routing behaviour. In specific, unless the current node neighbour is the package destination, the same node community membership determines the selected data path. In the case of lacking such node, with the probability q, the given package is transferred in the manner of random walk.

- topological (p)

  With a set probability p the edges of a given graph can be rewired (i.e. an edge can be disconnected from one of the nodes it interconnects and attached to another randomly selected vertex).

The conducted experiments show that the package transfer reveals features similar to the Stochastic Resonance. Ravasz-Barabási (RB) model has been found to be the most effective since it managed to deliver 90% of packets for any combination of p and q.

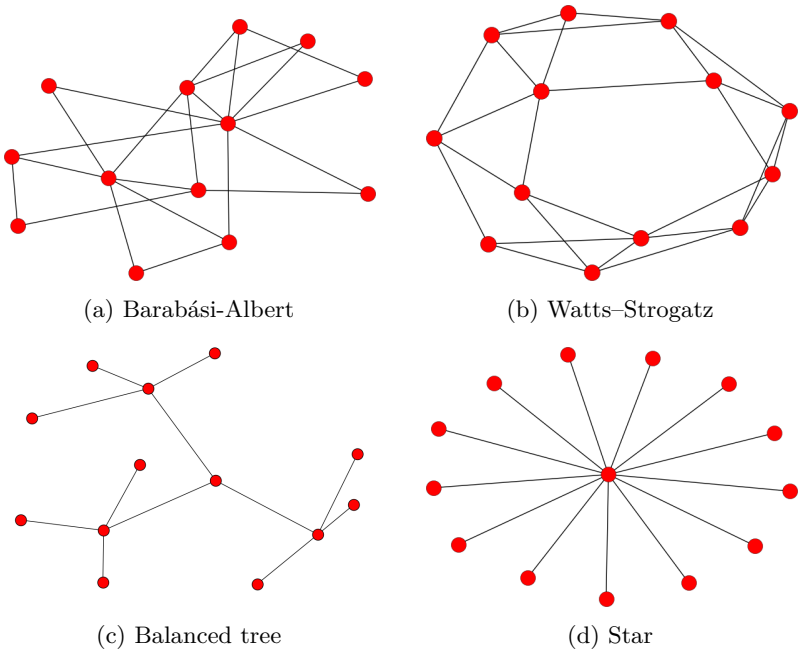
Inspired by previously described research, we define a model of computer network with noise. In this case, it is a transient link failure applied uniformly to every link in a given network. Next, we select the four graph models (BA, WS, Balanced Tree, Star) and evaluate their packet delivery capabilities with the defined metrics.

3 Graph models

This section gives an overview on the used graph models, motivation why those were selected for the experiments and the model specific parameters used for their generation. Table 1 summarizes the created sizes of each of used models (Figure 2).

Topology size	**Number of nodes (N)**
Small	13
Medium	121
Large	364

Table 1: Graph sizes created for each model.



(a) Barabási-Albert (b) Watts–Strogatz

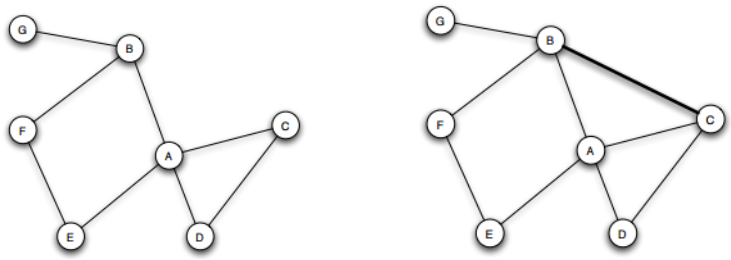
(c) Balanced tree (d) Star

Figure 2: Used graph models visualisations for N=13.

Barabási-Albert (BA) [10] (Figure 2a) is a random graph model generating the scale-free networks. A network is called scale-free when its node degree (number of edges a node is connected to) distribution follows the power-law [11]. More specifically, the probability $P(k)$ that a node is k-degree node adheres to $P(k) \sim k^{-\gamma}$, where the constant γ usually falls in the $< 2, 3 >$ range. As identified by Albert et al. [12], when errors in the form of failing nodes are introduced, BA displays prospective features in regard to the data delivery. Considering the above, as well as, the fact that BA reflects the features of real-world structures (e.g. Internet topologies [13]), we deemed interesting to examine its performance for the defined model (Section 4). BA utilises two mechanisms, which enable it to comply with power-law distribution. First, it is possible to indefinitely grow a network by adding a new node and connecting it to K different vertices, which are already part of the graph. During the conducted experiments, value K was set to 2. Next, the *preferential attachment* feature is responsible for the *"rich-gets-richer"* behaviour. Specifically, an edge spanned from a newly added node will be connected to an existing node i with a probability $\Pi = k_i / \sum_j k_j$, where k_i denotes the degree of a node i, divided by the sum of degrees of all graph vertices.

Figure 2b presents a connected variant of **Watts–Strogatz (WS)** model following the *small-world phenomenon* [14]. In detail, it demonstrates short paths between the nodes and triadic enclosures. The latter is the property stating

that if the nodes B and C have a common neighbour A (Figure 3a), adding an edge connecting B and C produces a *triangle* structure (Figure 3b). Watts and Strogatz [15] show that WS model is present in the reality. They show that social networks (i.e. collaboration graph of film actors) and Power grid structure of western United States adhere to the defined structure. Xia et al. [16] state that *scale-free* networks show robustness against random node failures; therefore we classified it as promising for this research. A WS graph creation starts with a creation of ring consisting of N nodes. Next, each vertex connects to K nearest neighbours. Finally, for a given probability β edges are rewired to other arbitrarily chosen node. In this research constructed graph was constructed using $K = 4$ and $\beta = 0.2$ values.



(a) Nodes B and C have the common neighbour A.

(b) Addition of edge B-C creates a triangle.

Figure 3: The creation of a triadic enclosure (from [14]).

Balanced tree (Figure 2c) is the example of hierarchical network model. Tree models are a common data structure where efficient searching plays important role [17], it is also a common design model for the data centre networks [18]. In the used model all leaves are placed at height h from its root. Next, the root node degree equals to r, whereas for other internal nodes $r+1$. In the conducted experiments, the r and h parameters were chosen as depicted in Table 2.

Graph size	**r**	**h**
small	3	3
medium	3	4
large	3	5

Table 2: Balanced tree parameters.

Star (Figure 2d) topology is essentially a special case of tree graph. A star having N nodes is practically a tree having a single root node and $N-1$ leaf nodes. As Elhedhli and Hu [19] state it is applicable for modelling airline passenger

travels, cargo delivery and packet delivery in computer networks. Additionally, as Ganesh et al. [20] suggest, the star model may support the analysis of power-law following graphs (e.g. WS).

4 Information flow model

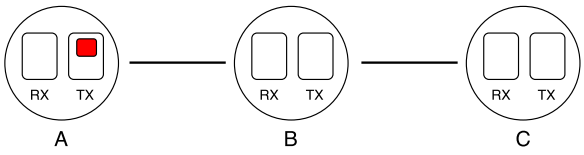
To be able to investigate the influence of topology on the packet delivery capabilities the environment allows introducing a defined network conditions was necessary. We decided to use simulation approach due to the following reasons. First, the simulation shows a low computational resources overhead in comparison with other approaches (e.g. emulation). This was particularly important assuming network topology with the hundreds of nodes. Secondly, the simulation allows simplifying the problem, narrowing down the problem domain. Finally, the ability to use relatively uncomplicated code base allowing easily extending and modifying model behaviour, posed an additional advantage.

In the model, the vertices represented network hosts, which were additionally capable of packet routing. The graph edges were used as the equivalent of network links interconnecting the nodes. Each vertex had the receive and transmit queues (later denoted as RX and TX), a local network structure image and a routing table holding the shortest paths to all other nodes. This essentially gave a regular host the ability to originate, forward and terminate certain packet flow.

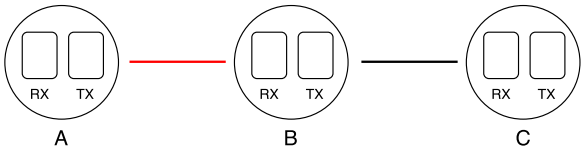
Figure 4 illustrates the model behaviour using the elementary case of node A sending a packet (red square) to node C. Node B which is placed in between the latter two, acts as an intermediate system (i.e. a router) which has to rely the packet to the appropriate link.

First, at node A, the packet is created with the destination set to C. Subsequently, it is placed it in the node A TX buffer (Figure 4a). A dequeues the packet from TX queue and looks up its destination in the routing table. If such entry exists (i.e. there is a functional path) the packet is sent over the pre-calculated shortest-path; otherwise, the packet is dropped. In this example the only possible path is the one via node B, therefore A transmits the packet over the A-B link. Moreover, before the link is used, with a certain probability p its state is changed to the opposite of the current one (Figure 4b). Depending on the result of the previous step A-B link may be changed to *down*, effectively invalidating the shortest and only possible path to C and causing the packet to be dropped. If A-B link remains as *up* the transmission continues and the packet arrives at B RX queue (Figure 4c). Next, B dequeues the packet and checks its Time-To-Live (TTL) value. If it is greater or equal to the predefined limit, the packet is dropped. If not, B places the packet in its TX buffer (Figure 4d). Later, it performs a routing table lookup for packet destination address resulting in transmitting the packet over the link being part of shortest-path to the node C (Figure 4e). However, similarly as for A-B link, B-C edge status is changed with the given value of p. Figure 4f presents the final state of the flow. Packet originated by A

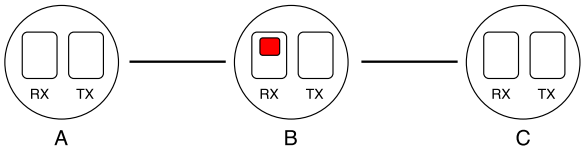
is stored in RX queue of node C which is also its recipient. Node C, removes it from the buffer and verifies TTL field value for being lower than the set limit. Finally, it notices that its own address is also the packet destination marking the packet as successfully delivered.



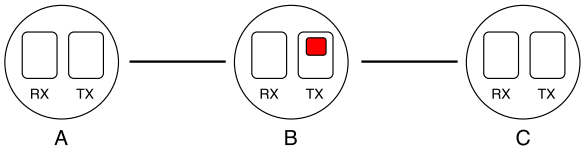
(a) Packet originated by A and destined for C is created and placed in the sender TX queue.



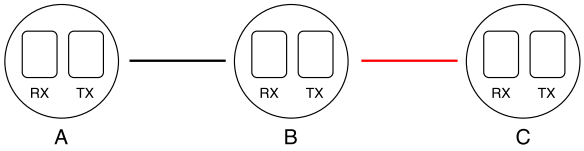
(b) The packet is transmitted over A-B link.



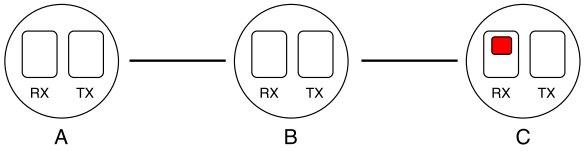
(c) Packet sent by A arrives into B RX queue.



(d) B analyses the destination address of packet and makes the decision to forward it (i.e. queues it in TX buffer)



(e) The packet is transmitted over B-C link.



(f) Node C receives the packet. It is placed in RX queue and later processed and marked as delivered.

Figure 4: Delivery behaviour of packet sent from A to C in the created model. The circle represents a particular network node and the dashed line symbolises ongoing transmission over the link. The inner part of the circle contains the boxes showing the state of receive and transmit buffers (RX and TX).

Besides holding its source and destination addresses, a packet is implemented with a TTL functionality known from Internet Protocol (IP) networks. A hop count field is incremented every time a node receives particular packet. If the value is greater than the maximum TTL dynamically calculated for a given topology, the packet is dropped. Effectively, this stands for dropping the packets, which traverse the path longer than tripled graph diameter.

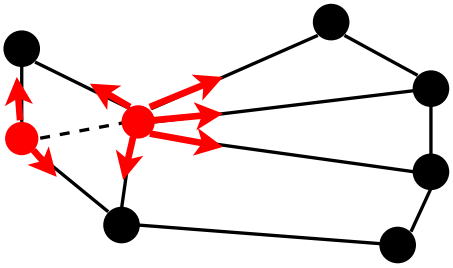
4.1 Link state change probability

Figures 4b and 4e show the situation when a node attempts to transmit the packet over a link being a part of the shortest-path to the packet destination. The defined model uniformly assigns the random link state probability p to all network links. Before node uses a link, its state is evaluated as follows:

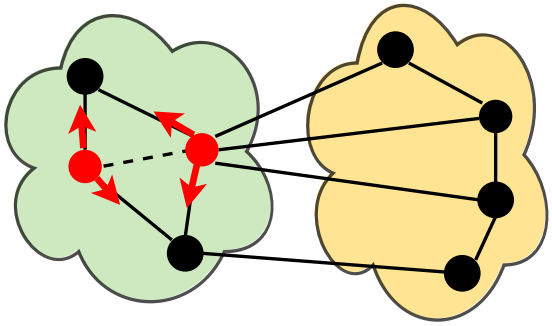
1. A link specific, pseudo random float number r from the range $< 0, 1 >$ is determined
2. If $r < p$, the link state is changed to the opposite of the current status $(up/down)$

4.2 Routing information propagation

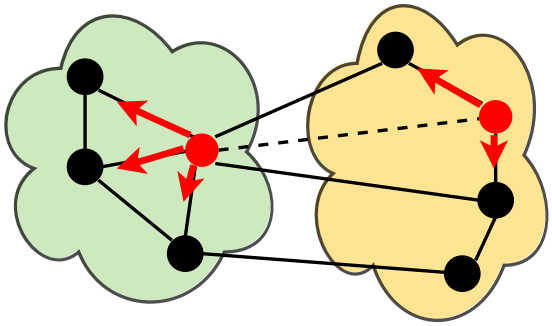
To establish a route over the topology we modelled a link-state routing protocol. Every node has a complete knowledge about the network structure (i.e. it is aware of all nodes and edges in a topology). Possessing such information allows executing Breadth-First Search (BFS) algorithm returning the shortest-path route between two nodes. Moreover, the modelled routing protocol detects topology changes (i.e. a link up or down status) and propagates this information about it to other nodes. The detection of edge status change (Figure 5) is instant for the nodes directly attached to it. They are recalculating the shortest-paths and flood the new information to the neighbours.



(a) A single routing protocol partition. The two directly attached nodes notice the interconnecting link state change. They flood the information to all its neighbours.



(b) Two routing protocol partitions with a state change of intra-area link. The link changing its state entirely belongs to the area marked in green. The directly connected nodes send the routing updates only to the other green area nodes.



(c) Two routing protocol partitions with a state change of inter-area link. The link interconnecting two nodes located in a separate routing domains changes its state. The routing updates are disseminated in both routing protocol domains.

Figure 5: Routing protocol state information spread. A dashed line represents the link, which changed its state. A red node colour symbolises a node directly attached to the affected link. Such node originates the updates showed as the red arrows. Green and orange clouds indicate a separate routing domain partitions limiting the routing information propagation.

On the occurrence of link failure, a node originates a *state* packet, and it is sent out to all other nodes participating in the topology. As described, this behaviour is adequate to the single routing information domain characteristics. To investigate the effect of having the multiple routing domains which do not exchange dynamic availability information, (i.e. link failure updates) the input graph was partitioned.

Figures 5b and 5c illustrate the concept behind topology partitioning. For such topology, a node located within the leftmost area (green) processes the routing information packet only if it contains the notification about the edge status connected to another green area node (Figure 5b). Inter-area edge state change (Figure 5c) is a special case of the previously specified behaviour. As it is connected to both areas, in the case of this link state change, the status update information is conducted across green and yellow areas.

5 Approach details

The Section 5.1 describes the details regarding the architecture and operations of used simulator. Next, in Section 5.2 we present the metrics used to answer posed research questions. Subsequently, the structure of conducted experiments is discussed in Section 5.3.

5.1 Simulator

We developed the simulator resembling used information flow model (Section 4). As the foundation the proof-of-concept simulator developed by the project supervisor Marc Makkes was adopted. Next, it was extended with the routing topology partitioning feature, as well as topology generation based on issued the command-line parameters.

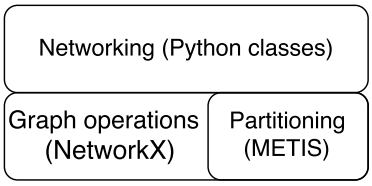


Figure 6: Simulator architecture.

Figure 6 presents the simulator architecture overview. A simulated network topology was implemented as a graph; therefore, actions such as the generation and manipulation of the network structure were essentially the NetworkX [21] library operations. As the NetworkX was lacking the sufficient quality graph partitioning mechanisms we used additional library providing the Python bindings

for METIS software [22]. Our tests showed that METIS partitioning algorithm produced the connected sub-graphs proportional in the number of nodes, which was the requirement we posed for the experiments. The features responsible for giving a graph vertex the notion of a network host were implemented as a set of Python classes.

The simulator creates the desired topology based on the given graph model. It performs a packet transfer between two arbitrarily chosen nodes, additionally introducing the random noise in the form of probabilistically determined link state changes. The described cycle is called an *iteration* and can be formalized as follows:

1. Randomly select N flows i.e. node pairs of A_i acting as a sender and B_i as a recipient, where $i = 0 \dots N - 1$
2. Transmit a packet over the shortest-path from A_i to B_i

5.2 Metrics

A set of four measures was gathered after each simulator run. The values of q_{rx}, q_{ttl}, q_{rt} described the ratio of packet delivery and drop events occurred during the simulated network transfer. By analysing these, we could quantify how many packets successfully reached the destination, how many of these were dropped and what the cause was. In total, these metrics gave the complete picture on the results of particular simulation, so that Equation (1) was always true. Lastly, $\bar{l^{-1}}$ presented the overview on the average path length taken by the packets, so that a conclusion about the quality of established shortest-paths could be inferred.

$$q_{rx} + q_{ttl} + q_{rt} = 1 \tag{1}$$

- packet delivery ratio (q_{rx})

  The ratio of the number of successfully delivered packets to the total amount of packets transmitted. It describes packet delivery capabilities of the certain network.

- "Max TTL" dropped packets ratio (q_{ttl})

  The ratio of the number of packets dropped because of reaching its defined TTL limit and the total number of sent packets.

- "No route" dropped packet ratio (q_{rt})

  The ratio of a number of packets dropped by the node because of missing destination route to the total number of transmitted packets.

- mean inverse path length ($\bar{l^{-1}}$)

  It represents the length of path a sent packet took (l) during the transmission. Where in the case of the dropped packet $l = 0$ is assumed. Therefore, it essentially gives the notion not only about the path length but also about the quantity of received packet. The mean path length is represented as its inverse to improve the readability of graph, effectively rescaling it to $< 0, 1 >$ range.

5.3 Experiment scenarios

The experiments consisted of running the simulator for the chosen topologies, each in the three size variants (*small*, *medium*, *large*). First, we selected ten logarithmically spaced link change probability values (p). In the next step for each of those, 1000 *iterations* (as described in Section 5.1), each establishing N=100 flows were performed for all sizes and topologies. Next, to investigate the effect of routing domain partitioning, BA and WS networks were partitioned and evaluated for the same number of *iterations* and flows as in the first part of the experiment. Tree and Star topologies were not included in this part because METIS algorithm was found unable to produce connected partitions for these models.

6 Results

This section presents the outcome of conducted simulations as described in Section 5.3. First, the results of simulations with models governed by the single area routing protocol are shown (Figures 7 to 9). Secondly, we exhibit the routing protocol domain partitioning effect on BA and WS topologies (Figures 10 to 12). For each of previously described subsections, we start with presenting all four metrics for a *small* network size. Subsequently, using the latter case as a baseline, we discuss the results for *medium* and *large* models displaying the same set of metrics.

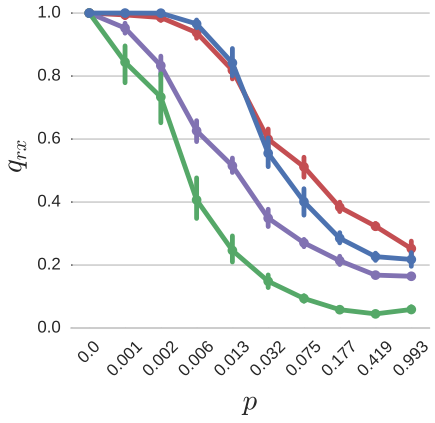
All metric graphs have the same layout. A particular graph x-axis represents the values of link state change probability p, whereas y-axis expresses the values measured for a given metric in the function of p. The plots of q_{rx} and $\bar{l^{-1}}$ should be interpreted in the higher the value, the better the performance, whereas q_{ttl} and q_{rt} in the opposite manner. We use `<type>-<partition amount>` notation to indicate the number of partitions within a particular network. As for the example, BA model partitioned to 4 areas results in the *ba-4* label. All plots were created with the assistance of seaborn [23] library at 95% confidence level.

6.1 Single partition experiments

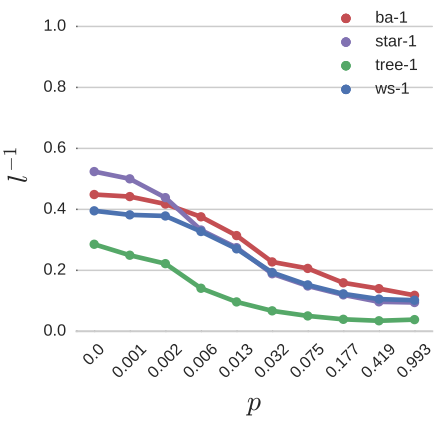
This part of results displays the behaviour of Star, Balanced tree, WS and BA models for the chosen network sizes. In this part of the experiment, the routing protocol update propagation domain was unlimited; such that, every node in the tested network was informed about any link state transition. We focus on discussing the model showing the highest data delivery related metrics (q_{rx}, $\bar{l^{-1}}$), followed by the evaluation of packet drop cause (q_{ttl}, q_{rt}).

Starting with an analysis of *small* network (Figure 7) we observed:

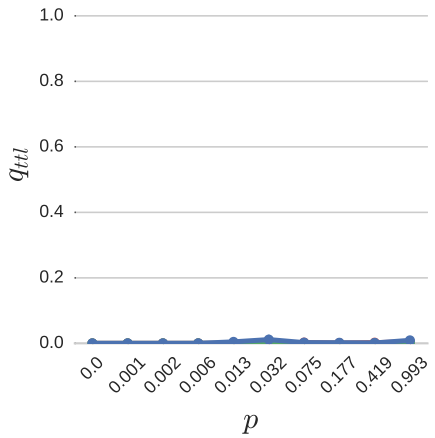
- WS and BA models achieved similar q_{rx} values for $p \leq 0.032$. However, for the subsequent p samples, a noticeably higher score was presented by BA (Figure 7a).
- In Figure 7b illustrating $\bar{l^{-1}}$, Star plot showed the highest values for $p \leq 0.002$, for the higher p samples giving back the precedence to BA.
- At $p = 0.032$ and $p = 0.993$ samples, for WS over 1% of sent packets experienced drops caused by exceeding TTL limit (Figure 7c).
- For $p =< 0.075 \ldots 0.419 >$ BA model showed q_{rt} rates roughly 10 percentage points lower than WS (Figure 7d).



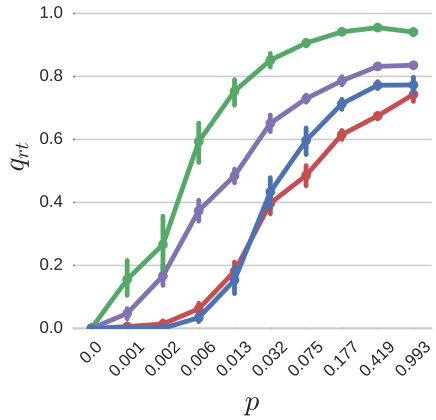
(a) $p \geq 0.032$, BA (red) delivered the most packets.



(b) $p \geq 0.006$, BA (red) showed the highest $\bar{l}^{-1}$



(c) $p = 0.032$ and 0.0993, WS (blue) experienced over 1% of TTL caused drops.



(d) $p \geq 0.032$, BA had the lowest ratio of packet drop caused by missing route.

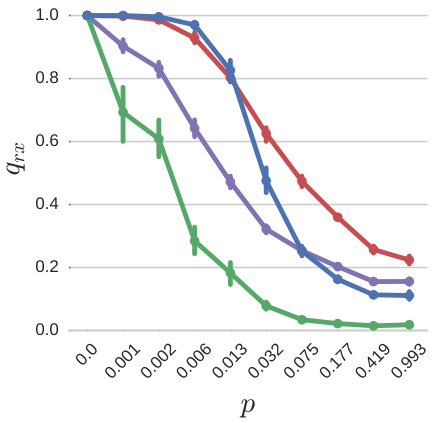
Figure 7: The metrics gathered for *small* network size in single routing partition experiment.

In *medium* and *large* network graphs we observe the behaviour similar to the smallest size. The following points describe the interesting facts:

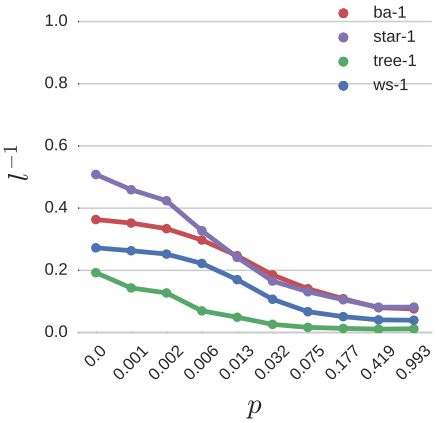
- BA and WS presented close q_{rx} values for shorter p range ($p \leq 0.013$)

(Figures 8a and 9a).

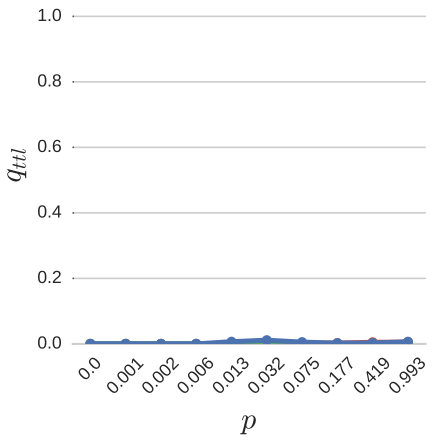
- Star model took the precedence over BA for a longer p range (Figures 8b and 9b).
- BA did not occur any significant At $p = 0.032$ WS experienced peak q_{ttl} values — over 1% (Figure 8c) and over 4% (Figure 9c) for *medium* and *large* size respectively.
- In Figures 8d and 9d we see that for both network sizes, BA $p \geq 0.032$ showed significantly lower q_{rt} values than WS. For *large* model differences reached over 20 percentage points.



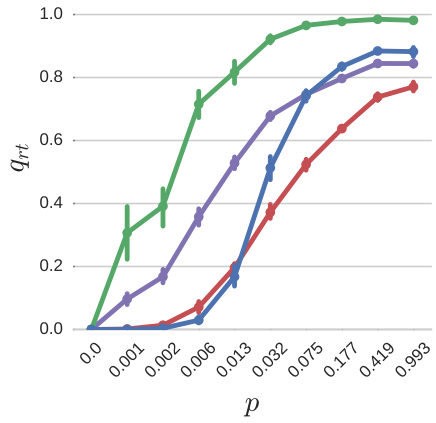
(a) $p \geq 0.032$, BA (red) delivered the most packets.



(b) $p \geq 0.013$, BA (red) and Star (violet) provided similar path length for the delivered packets.

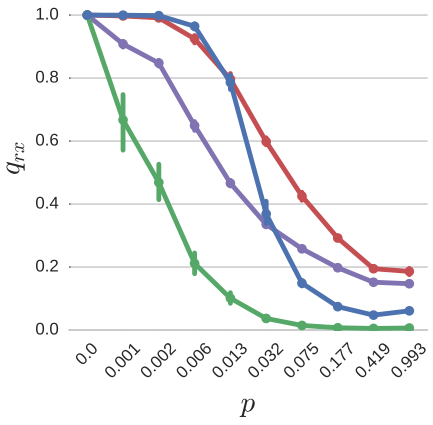


(c) $p = 0.032$, WS (blue) showed over 1% of TTL packet drops.

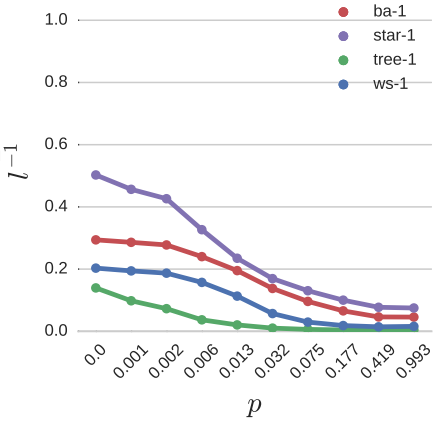


(d) $p \geq 0.032$ BA (red) showed significantly lower packet drops caused by missing routing table entry.

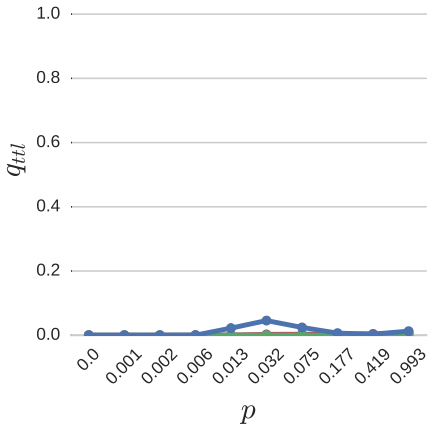
Figure 8: The metrics gathered for *medium* network size in single routing partition experiment.



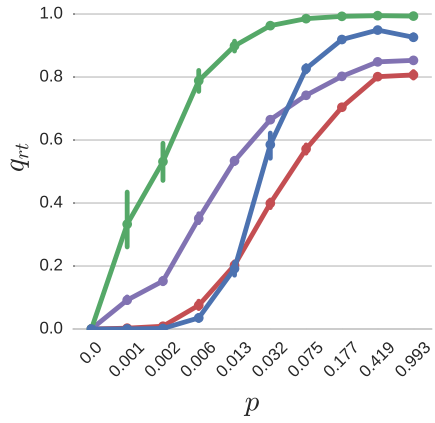
(a) $p \geq 0.032$, BA (red) delivered the most packets.



(b) Star (violet) showed the lowest mean path length.



(c) $p = 0.032$, WS (blue) experienced over 4% of "Max TTL" drops.



(d) Starting $p = 0.032$ BA (red) experienced the lowest "No route" events.

Figure 9: The metrics gathered for *large* network size in single routing partition experiment.

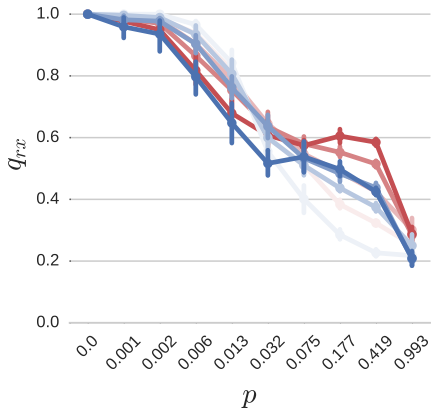
6.2 Partitioning experiments

This subsection shows the result of routing domain partitioning. In detail, the limiting of routing protocol updates spread by assigning the node to be a member

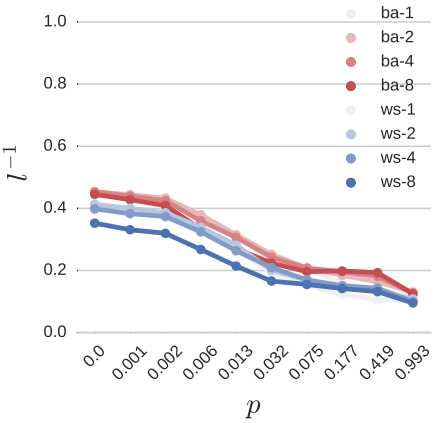
of certain area, and instructing it to ignore (i.e. do not propagate) an update related to other partition. Figures 10 to 12 show the four metric graphs for the *small*, *medium* and *large* networks, respectively. We display the performance of BA and WS models portioned to 2, 4 and 8 areas. Additionally, to ease the comparison with a single domain case, we plot it as well. As we present eight plots per single graph, readability could be difficult. However, such layout gives the clear comparison of all cases. The compromise is to use different hues of a red/blue colour to illustrate an increasing amount of partitions for a particular model. (e.g. we assign the lightest blue to *ba-1* and the darkest to *ba-8*). Furthermore, to keep the description concise we focus on describing the trend partitioning had i.e. how does the use of partitions influence a particular metric.

In *small* network packet delivery (Figure 10a) case, the partitioning had a similar influence on BA and WS:

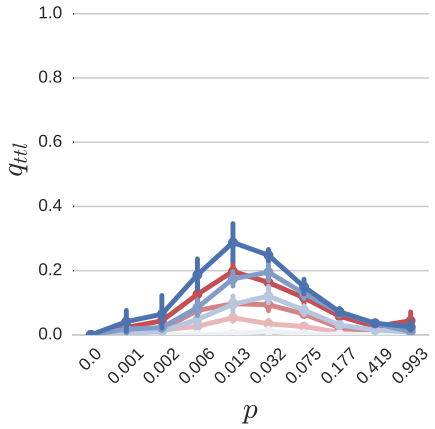
- For the link state change probability $p \leq 0.032$ (BA) and $p \leq 0.075$ (WS), more partitions degraded the amount of successfully transferred packets. Nevertheless, the samples occurring at the next p values showed the opposite behaviour, q_{rx} values of the eight partitions were noticeably higher than the single area variants.
- The partitioning did not have any significant influence on BA model (Figure 10b), however, *ws-8* case clearly extended $\bar{l}^{-1}$ in comparison with other WS plots.
- Figure 10c displays that the usage of multiple partitions amplified the ratio of "Max TTL" events for both network models.
- Starting with $p \geq 0.013$, the partitioned BA and WS networks experienced a noticeably lower ratio of packets dropped due to the missing route (Figure 10d).



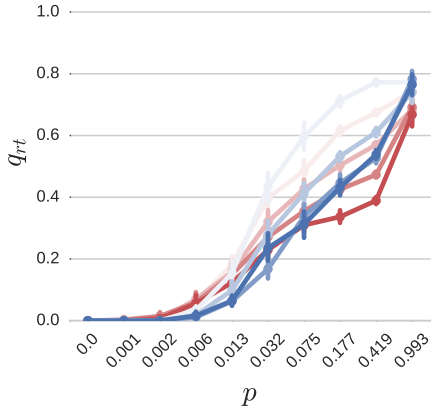
(a) $p = 0032$ (BA) and $p = 0.075$ (WS), breaking point changing partitioning effect.



(b) $p \leq 0.075$,



(c) Partitions cause "Max TTL" events increase.



(d) $p \geq 0.013$, less packet dropped because of lacking routing table entry.

Figure 10: The metrics for *small* network, with routing domain partitioning.

Medium and *large* model results (Figures 11 and 12) follow the *small* network observations:

- The packet delivery ratio (Figures 11a and 12a) has a similar characteristic.

For the same p ranges as in *small* topology case, WS and BA first performed the best unpartitioned. Whereas, higher p samples showed that more partitions improved the amount of packet which reached its destination.

- $\bar{l^{-1}}$ appeared to be noticeably more uniform, both WS and BA did not show any denoting differences between the plots representing different number of routing partitions (Figures 11b and 12b).
- Figures 11c and 12c depicting q_{ttl} metric ratio illustrate that the networks behaved uniformly. The highest number of packets dropped occurred at $p = 0.013, 0.032$ and did not pass 40%.
- In Figures 11c and 12d we see the general trend, as well. The samples which occurred for $p \geq 0.013$ depict the reduced amount of "No route" events for the partitioned topologies.

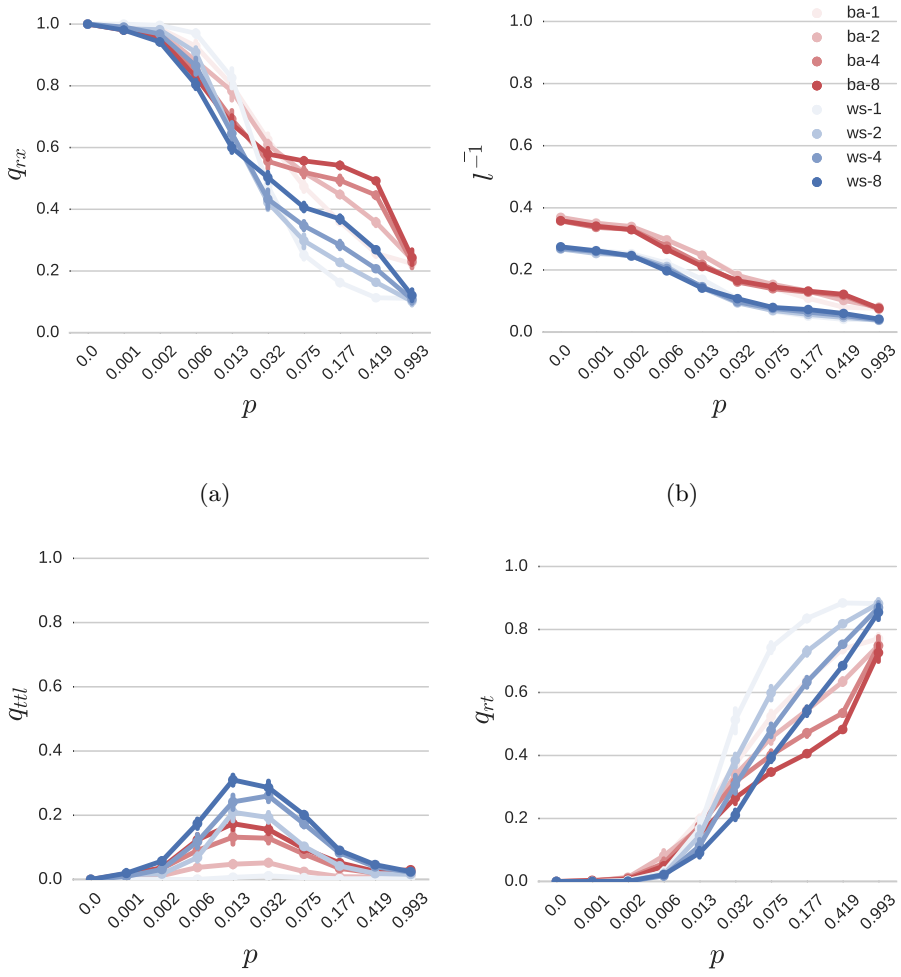

1.0
0.8
0.6
0.4
0.2
0.0
q_{rx}
$\bar{l^{-1}}$
q_{ttl}
q_{rt}
0.0
0.001
0.002
0.006
0.013
0.032
0.075
0.177
0.419
0.993
p
ba-1
ba-2
ba-4
ba-8
ws-1
ws-2
ws-4
ws-8
(a)
(b)
(c)
(d)

Figure 11

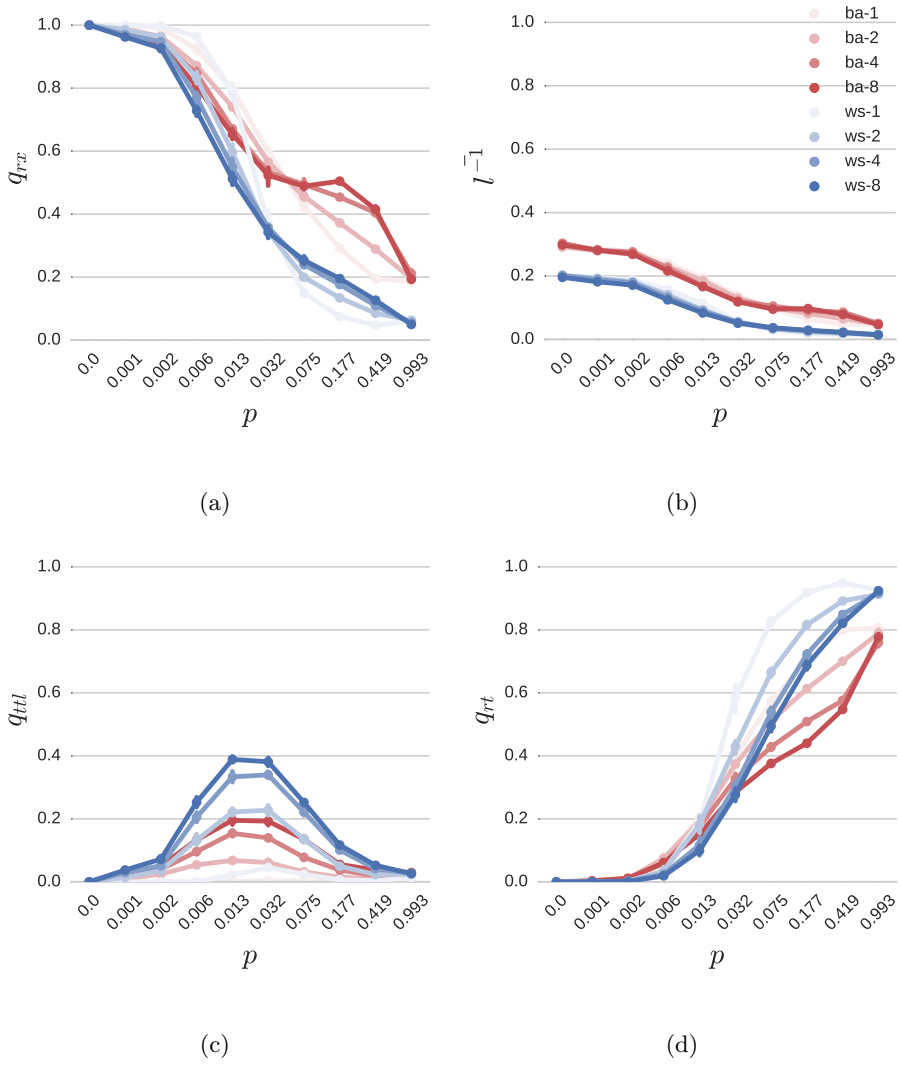


Figure 12: The metrics for *large* network, with routing domain partitioning.

7 Discussion

In this section, we analyse the results of the simulations in the context of research questions posed. First, in Section 7.1, the outcome of unpartitioned routing domain experiments is debated. Secondly, we move on to the analysis of partitioning effect on the BA and WS models packet delivery (Section 7.2).

7.1 Unpartitioned topology packet delivery

The simulation results for the networks managed by a single-domain routing protocol (Section 6.1) show that depending on the probability of a transitive link failure occurrence (*p)*; WS or BA topology may be used to get an optimal packet delivery rate.

In the relatively reliable environment, which we deem to be characterised by $p < 0.01$, both mentioned models perform similarly. To break the tie, we take $\bar{l^{-1}}$ metric into the consideration, we see that BA tends to provide shorter paths for the packets it delivers. With the network size growth, the latter observations are also true. Although we observe that BA's packet delivery slightly subsides and it is lower than WS's score ($p = 0.006$), for other data points and including $\bar{l^{-1}}$ score; BA shows good transfer conditions other evaluated models are unable to compete with.

In the unreliable network environment ($p \geq 0.01$), we observe that for the most cases, BA is capable of transporting significantly more data than WS and other models. Moreover, presenting superior score of $\bar{l^{-1}}$ metric. Considering the influence of network size, BA withholds its features in spite of obviously longer path an average packet has to traverse.

Given the BA and WS graph model features (Section 3), namely the adherence to *power-law* and *small-world* phenomenons, respectively. We suppose that evaluating this experiment for the networks consisting of a significantly higher number of nodes may reveal different findings. In specific, we expect that WS network has a potential to create shorter paths than BA. Therefore, in the assumed model, where every link may transition to different state, a chance for the successful packet transfer increases when a low number of nodes has to be traversed. However, the results for Star model show, a short distance between the nodes is not the remedy. An ability to establish an alternate path is crucial to deal with the link failures and possession of several links which interconnect a node with the rest of topology.

7.2 Partitioning influence

The experiments show the networks where the link failures are rare i.e. $p \leq 0.032$, do not benefit from limiting the routing protocol state messages. Moreover, such approach causes network loops ("Max TTL" events) and the deterioration of packet delivery ratio. This observation appears totally consistent with the intuition; when the network link changes its state, more the nodes can get the information about it, bigger the chance a certain node makes a better routing decision. Narrowing down the protocol's update flooding area for a low p appears harmful to the packet transmission.

Nonetheless, the situation becomes interesting when the link state change probability is set to the values higher than 0.075. The segmentation of the routing

notification domain result in an increase of delivered packets ratio. To explain this phenomenon we look at the q_{ttl} and q_{rt} metrics. Clearly, creating the routing information areas caused the routing loops, which resulted in the more packets being dropped due to exceeding TTL value. Whereas, the second drop metric shows that by effectively ignoring part of updates the amount of "No route" events decreased. The latter observation seems rather evident, since the information about the link state changes was ignored, however, as it turns out such approach provides the actual benefit in the form of q_{rx} increase. Therefore, we deduce that the part of the routing updates originated in such network conditions was invalid in the moment a remote node received and processed it. In other words, in the simulated conditions when the routing protocol is reconverging frequently, limiting routing state propagation lead to a more efficient network packet delivery capabilities.

8 Conclusions

We conclude that in the network model, characterised by the following features:

- uniformly distributed transient link failures
- shortest-path driven link-state routing protocol

The topology based on BA model shows the good capabilities for the data delivery resembled by the high ratio of transferred packets and short paths the latter took to arrive. This finding applies to all three topology sizes we tested the models for. Nonetheless, in the situation the high probability of link change is applied BA performance severely decreases. Our finding shows that part of the responsibility for that fact lies not only in the physical lack of network path, but in the routing protocol reconvergence. By limiting the update spread area we manage to achieve the increase in the number of sent packets, in this possible, yet a highly unrealistic conditions.

9 Future work

To further extend on the conducted research we state the following propitious topics:

- Extending on the size of tested networks
- Testing tree topologies with loops (e.g. Ravasz-Barabási)
- Using more complex graph models (i.e. parallel, weighted edges)
- Introducing new disruption types:
  - Unidirectional link failures (i.e. the link state is noticed only by one of nodes interconnected by the certain failing edge).

– Routing protocol *state* messages rate limiting

10 Acknowledgments

I wish to thank my supervisor Marc Makkes for his guidance and valuable support during this research project. I would also like to thank dr. Paola Grosso for her insightful recommendations on my paper and presentation.

References

[1] Hong Yan et al. 'Tesseract: A 4D Network Control Plane.' In: *NSDI*. Vol. 7. 2007, pp. 27–27.

[2] Craig Labovitz et al. 'The impact of Internet policy and topology on delayed routing convergence'. In: *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. Vol. 1. IEEE. 2001, pp. 537–546.

[3] Craig Labovitz et al. 'Delayed Internet routing convergence'. In: *ACM SIGCOMM Computer Communication Review* 30.4 (2000), pp. 175–187.

[4] E. W. Dijkstra. 'A Note on Two Problems in Connexion with Graphs'. In: *NUMERISCHE MATHEMATIK* 1.1 (1959), pp. 269–271.

[5] Zheng Wang and Jon Crowcroft. 'Analysis of Shortest-path Routing Algorithms in a Dynamic Network Environment'. In: *SIGCOMM Comput. Commun. Rev.* 22.2 (Apr. 1992), pp. 63–71. ISSN: 0146-4833. DOI: 10.1145/141800.141805. URL: http://doi.acm.org/10.1145/141800.141805.

[6] Paolo Crucitti et al. 'Error and attack tolerance of complex networks'. In: *Physica A: Statistical Mechanics and its Applications* 340.1–3 (2004). News and Expectations in Thermostatistics, pp. 388–394. ISSN: 0378-4371. DOI: http://dx.doi.org/10.1016/j.physa.2004.04.031. URL: http://www.sciencedirect.com/science/article/pii/S037843710400439X.

[7] Vito Latora and Massimo Marchiori. 'Efficient Behavior of Small-World Networks'. In: *Phys. Rev. Lett.* 87 (19 Oct. 2001), p. 198701. DOI: 10.1103/PhysRevLett.87.198701. URL: http://link.aps.org/doi/10.1103/PhysRevLett.87.198701.

[8] Kurt Wiesenfeld, Frank Moss et al. 'Stochastic resonance and the benefits of noise: from ice ages to crayfish and SQUIDs'. In: *Nature* 373.6509 (1995), pp. 33–36.

[9] Agnieszka Czaplicka, Janusz A. Holyst and Peter M. A. Sloot. 'Noise enhances information transfer in hierarchical networks'. In: *Scientific Reports* 3 (6th Feb. 2013). Article, pages. URL: http://dx.doi.org/10.1038/srep01223.

[10] Albert-Laszlo Barabasi and Reka Albert. 'Emergence of Scaling in Random Networks'. In: *Science* 286.5439 (1999), pp. 509–512. DOI: 10.1126/science.286.5439.509. eprint: http://www.sciencemag.org/cgi/reprint/286/5439/509.pdf. URL: http://www.sciencemag.org/cgi/content/abstract/286/5439/509.

[11] Mark EJ Newman. 'Power laws, Pareto distributions and Zipf's law'. In: *Contemporary physics* 46.5 (2005), pp. 323–351.

[12] R. Albert, H. Jeong and A.L. Barabasi. 'Error and attack tolerance of complex networks'. In: *Nature* 406.6794 (2000), pp. 378–382.

[13] Michalis Faloutsos, Petros Faloutsos and Christos Faloutsos. 'On power-law relationships of the internet topology'. In: *ACM SIGCOMM computer communication review*. Vol. 29. 4. ACM. 1999, pp. 251–262.

[14] Easley David and Kleinberg Jon. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. New York, NY, USA: Cambridge University Press, 2010, pp. 48–49, 611–644. ISBN: 0521195330, 9780521195331.

[15] D.J. Watts and S.H. Strogatz. 'Collective dynamics of 'small-world' networks'. In: *Nature* 393 (1998), pp. 440–442.

[16] Yongxiang Xia, Jin Fan and David Hill. 'Cascading failure in Watts-Strogatz small-world networks'. In: *Physica A: Statistical Mechanics and its Applications* 389.6 (2010), pp. 1281–1285. ISSN: 0378-4371. DOI: http://dx.doi.org/10.1016/j.physa.2009.11.037. URL: http://www.sciencedirect.com/science/article/pii/S0378437109009716.

[17] Antonin Guttman. 'R-trees: A Dynamic Index Structure for Spatial Searching'. In: *SIGMOD Rec.* 14.2 (June 1984), pp. 47–57. ISSN: 0163-5808. DOI: 10.1145/971697.602266. URL: http://doi.acm.org/10.1145/971697.602266.

[18] Mohammad Al-Fares, Alexander Loukissas and Amin Vahdat. 'A scalable, commodity data center network architecture'. In: *ACM SIGCOMM Computer Communication Review*. Vol. 38. 4. ACM. 2008, pp. 63–74.

[19] Samir Elhedhli and Frank Xiaolong Hu. 'Hub-and-spoke network design with congestion'. In: *Computers & Operations Research* 32.6 (2005), pp. 1615–1632. ISSN: 0305-0548. DOI: http://dx.doi.org/10.1016/j.cor.2003.11.016. URL: http://www.sciencedirect.com/science/article/pii/S0305054803003617.

[20] Ayalvadi Ganesh, Laurent Massoulié and Don Towsley. 'The effect of network topology on the spread of epidemics'. In: *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.* Vol. 2. IEEE. 2005, pp. 1455–1466.

[21] Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart. 'Exploring network structure, dynamics, and function using NetworkX'. In: *Proceedings of the 7th Python in Science Conference (SciPy2008)*. Pasadena, CA USA, Aug. 2008, pp. 11–15.

[22] George Karypis and Vipin Kumar. 'A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs'. In: *SIAM Journal on Scientific Computing* 20.1 (1999), pp. 359–392.

[23] Michael Waskom et al. *seaborn: v0.7.1 (June 2016)*. June 2016. DOI: 10.5281/zenodo.54844. URL: http://dx.doi.org/10.5281/zenodo.54844.

A Topology details

Size & Type	Nodes	Edges	Average node degree	Edge connectivity	Diameter	TTL
small ba	40	76	3.8000	2	4	12
medium ba	121	238	3.9339	2	6	18
large ba	364	724	3.9780	2	7	21
small ws	40	80	4.0000	2	6	18
medium ws	121	242	4.0000	2	9	27
large ws	364	728	4.0000	2	10	30
small star	40	40	1.9512	1	2	6
medium star	121	121	1.9836	1	2	6
large star	364	364	1.9945	1	2	6
small tree	40	39	1.9500	1	6	18
medium tree	121	120	1.9835	1	8	24
large tree	364	363	1.9945	1	10	30

Table 3: Detailed topology characteristics