ENSC 833: NETWORK PROTOCOLS AND PERFORMANCE

Performance and Analysis for Deflection Routing in Buffer-less Networks Using ns-3

Spring 2019

Final Report

Malsawmtluangi, Lucy (lmalsawm@sfu.ca)

Wang, Renjie (renjiew@sfu.ca)

https://renjiew.wixsite.com/website

Simon Fraser University

Abstract

Buffer-less networks such as Optical Burst Switching (OBS) networks which are high-bandwidth backbone networks (Ethernet interfaces of 40 Gigabit Ethernet and 100 Gigabit speed [1]), packets are assembled into bursts before they are transmitted over optical fibers [2]. Burst blocks or burst drops occur due to contention in these complex networks [3]. Contentions occur when the buffer space is unavailable for packet bursts that are scheduled for transmission [4]. Deflection routing is used to deflect the traffic flow to a different next-hop in case of such contention [5]. In this project, we study the performance of Deflection Routing in resolving packet contention problems in complex networks such as OBS using ns-3 network simulator [6].

Keywords: Optical Bursting Switching (OBS), Deflection routing, Contentions, ns-3

Acknowledgment

We would like to thank our advisor Dr. Ljiljana Trajkovic for all her guidance and continuous support throughout this project. We would like to thank you for encouraging our study and opening the doors of an amazing lab where we met wonderful, smart, and committed people. Your advice on both, study as well as on our career have been priceless. We are also deeply grateful to our Teaching Assistant Zhida Li for the brainstorming discussions and guidance. We have learned so much from you.

We also want to thank and acknowledge Dr. Soroush Haeri and Dr. Ljiljana Trajkovic for the iDef Framework and the source code that we studied and ran test cases for computing the performance of Deflection Routing algorithms.

Table of Contents

Table of Figures

Table of Tables

Table of Equation

List of Abbreviations

BFD	Bidirectional Forwarding Detection
MPLS	Multiprotocol Label Switching
OAM	Operation Administration and Maintenance
OBS	Optical Bursting Switching
OSPF	Open Shortest Path First
OXC	Optical Cross Connect
PQDR	Predictive Q-learning Deflection Routing
Q-NDD	Q-learning-based Node Degree Dependent
RLDRS	Reinforcement Learning-based Deflection Routing Scheme
TCP	Transmission Control Protocol
WDM	Wavelength Division Multiplexing

Introduction

In optical packet switching networks, smaller packets are being transmitted from the source to the destination over many routers and each router would have a buffer to hold the packets in a queue waiting to be transmitted. There maybe queuing delays and queuing losses associated with each router buffer. In the case of queuing losses, the packets are retransmitted thus increasing total transmission time such as for TCP traffic. These queuing delays and losses are eliminated in buffer-less networks such as Optical Bursting Switched (OBS) networks. OBS networks make use of data bursts in an all-optical form thus eliminating optical/electrical/optical conversions and delays [3]. However, contention may happen in these buffer-less networks. Contention is when two nodes attempt to transmit bursts over the same outgoing link and/or path, but only burst from one node is sent to the optimal outgoing link while burst from the other node is lost [5]. Deflection routing is a viable solution to the contention issues in these buffer-less networks.

Optical packet switching, and optical circuit switching are two methods very similar to the traditional packet switching and circuit switching methods in electrical data networks. However, packet switching routers have cheap buffers in order to enqueue packets in the egress links. On the other hand, the technology for optical signal storage is not mature enough as of this day to produce cost-efficient optical buffers [7] in case contention happens. An alternative to this is to use the deflection routing strategy.

In the ideal world scenario, every optical burst sent through OBS networks arrives at a router and gets forwarded to the desired optimal output port straight away without any interferences [3]. In the real-world scenarios, two or many nodes could send optical bursts at the same time, and they would arrive at the same router and based on the shortest path deduced by the routing protocols, they maybe bound to leave from the same output port. The concept of deflection routing is to send the extra bursts through other less optimal ports and paths that are idle, in order to avoid dropping the optical burst in the network that would cause traffic loss. This strategy is only used in the core components of the network. It also avoids the high costs of using optical buffers as mentioned in previous optical packet switching solution [5]. Therefore, deflection routing is comparable and even more desirable than other contention resolving methods. And we are going to study contention resolution in buffer-less networks such as OBS, enhance existing routing protocols by adding viable contention solution, and delve further into the deflection routing algorithm in resolving packet losses due to contentions with the help of ns-3 simulations.

Motivation

Packet loss and delays in Optical Packet Switching networks have always obstructed a high-quality end user experience in the communication network. Due to low bandwidth, if the packet is queued multiple times across the network in the egress queues of the routers before it reaches the destination node, it causes network latency and congestions. Moreover, if the packet is dropped/lost and the protocol in use is TCP, all the lost/dropped packets are retransmitted causing further network overhead and congestions. On the other hand, by leveraging the Optical Burst Switching (OBS) which aggregates packets destined to the same node into bursts provides a competitive solution for packet loss and delay. However, OBS networks suffer from contention problem leading to burst drops due to lack of available buffer. We can solve such contention issues using deflection routing algorithm from which OBS networks could be benefited and maybe widely used in the future in the high bandwidth deployments such as subsea optical fiber cable.

Figure 1 shows the Submarine Cable Map depicting how the data is transmission across different continents with precise time granularity of nanoseconds. New optical subsea cables are been laid out often in order to support growing need of high bandwidth demand. Therefore, the submarine cable deployment has a huge potential use for OBS technology, and it makes the communication faster, cheaper and more reliable.

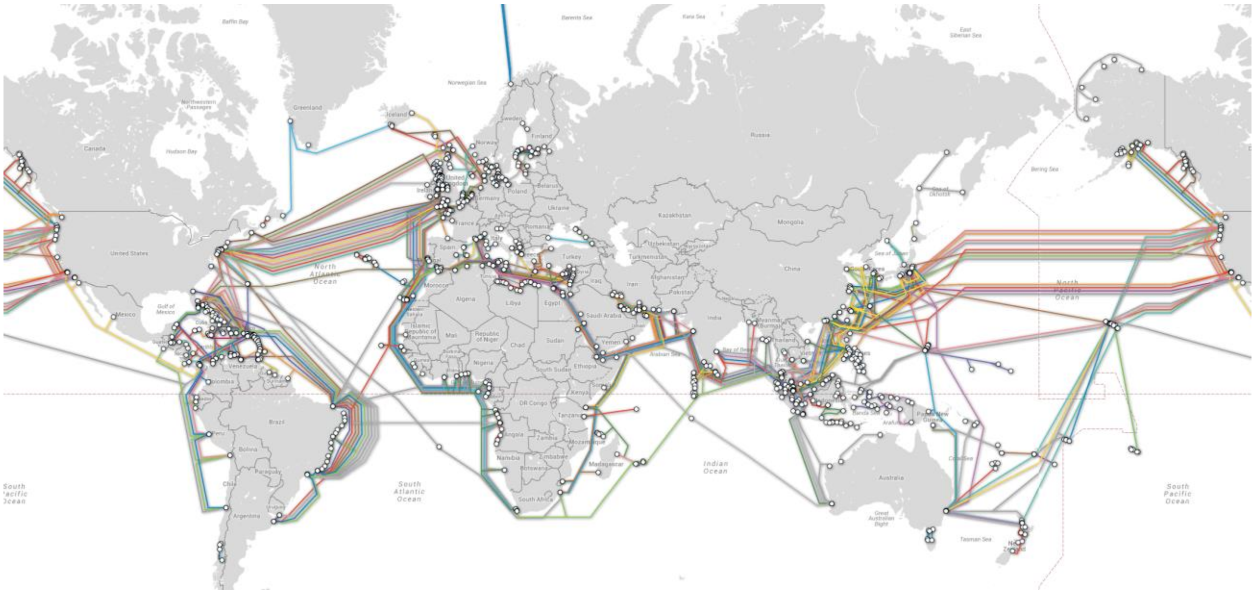


Figure 1: Submarine Cable Map [8]

Related works

As we know, “Routing in communication networks is a process of selecting a path that logically connects two end-points for packet transmission” [2]. The decision-making algorithm and how to transfer packets depend on the network topology and the weight of each edge such as hops to destination, congestion, latency, and etc. in order to minimize cost between source and destination. However, reinforcement learning is usually used for decision-making in the dynamic environment [9], and three components are required in reinforcement learning for deflection decision making including the state of the environment, reinforcement signals from the environment, and a learning algorithm. The state of the environment is represented into several integer variables and store into a table. Reinforcement signal from the environment is a signal mechanism which based on the state of the packet. When the state of the packet is updated, a reinforcement signal generates. A learning algorithm is to make the decision depending on the state [2].

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha * (r_{t+1} + \gamma_{a_{t+1}}^{max} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$$

Equation 1: Q-value

Q-learning [10] is a common approach in deflection routing. The above equation is how to calculate Q-value which will be collected into a Q-table in order to make the routing decision. The explanation for the equation expresses in the following. s_t is the state of the environment variable and updated based on time t. The reward r_{t+1}corresponds to the s_t moving from s_t to s_{t+1} and action a_t. The learning rate α and the discount factor γ are the coefficient between 0 and 1, and estimated optimal future value $\max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$ will be predicted based on previous experience. Obviously, the new Q-value is always according to the old Q-value and learned value [2]. The policy needs to be emphasized for Q-learning is that Q-learning apply for each node locally which decreases the computing intension [11]. However, Q-learning does not produce an optimal routing policy for packet transmission directly [2], and the method of distributed gradient ascent policy search generates a predictive Q-routing [12]. When a packet successfully transmits from source to destination, a reinforcement signal is sent for propose, and the best experience of the routing will be recorded in order to predict and reused to the traffic behavior [13].

In the large networks, the reinforcement learning-based routing algorithms haven't been widely used because it might affect the business relationships between Internet service providers. Also, there are too many randomnesses which is difficult to control [2].

Literature Review of Deflection Routing

Two routing protocols are working simultaneously in the network including Open Shortest Path First (OSPF) and deflection routing algorithm when contention occurs. When the multiple flows compete for the node, only one flow will pass and other flow will be discarded based on the routing table because of the lack of buffer to enqueue the flow. The deflection routing algorithm is used to deflect other flow by transmitting it to the destination node rather than losing them. The decision-making for the algorithm is the surround available link which named deflection sets, and its purposes to avoid routing loops in the performance [14]. Random deflection routing might be an efficient way to reduce traffic load and jitter in the networks [15]. Exchanging traffic node information plays an important role in deflection decision because of the better understanding of surrounding better path route [16]. Since the control signal in deflection routing is making decision, the reinforcement learning algorithms add a degree of randomness for better decision making [2].

There are three types of algorithm for the decision making which are the Q-learning-based Node Degree Dependent (Q-NDD) deflection routing algorithm, the Predictive Q-learning Deflection Routing (PQDR) algorithm, and the Reinforcement Learning-based Deflection Routing Scheme

(RLDRS) [2]. RLDRS generates reward procedures, and PQDR records the best experience in order to predict and reused the traffic behaviors [13]. However, the disadvantages of RLDRS and PQDR are the complexity of the algorithm depend on the size of the network, and whenever it deflects the feedback signal will be sent. On the other hand, Q-NDD depends on the node degree, so only gets feedback signal when discarding by another node [2].

Network Topologies

$$Pr(\{u,v\}) = \eta exp(\frac{-d(u,v)}{L\delta})$$

Equation 2: An edge that connects nodes u and v exists with a probability

For the engineering system, there are several algorithms in network topologies, and the Waxman algorithm is a common approach. The equation above is showing an edge that connects nodes u and v exists with a probability. η and δ are the coefficient between 0 and 1. $d(u,v)$ is the distance between u and v, and L represent the maximum distance between two nodes. However, Waxman algorithm only creates a graph for vision, but not guarantee that each node and edge are connected since each edge has a probability [17].

$$Pr(i,j) = \frac{d_j}{\sum_{k\in N} d_k}$$

Equation 3: A new node i that is added to the network connects to an existing node j with probability

Barab´asi-Albert creates a power-law degree distribution network [18], the scale-free network graph, which has been widely used in a computer network. The equation above is the probability that a new node i is added to the network connects to an existing node j. d_j is the degree of node j, and N represents all nodes in the network. The scale-free graph places the high-degree vertices in the middle of the network connecting them to the core, and make low-degree vertices surrounding to them. Since the internet is close to autonomous systems which is a various-level structure, Barab´asi-Albert algorithm and the scale-free network graph have been used to establish Internet-like graph [2].

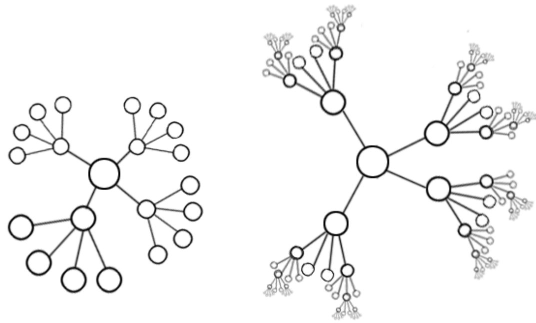


Figure 2: Scale-free networks [19]

Burst Traffic

As we discussed earlier, the better understanding of surrounding generates better path route, so the information on traffic measurements and analysis is very significant for the algorithm to predict the path. Since we transmit the same destination pack at the same time, traffic behavior is different from sending a single packet. Generally, the OBS algorithm generates time-based burst and length-based burst. Time-based burst has predefined the arrival time, and with the increasing burst packet, the burst length will follow the Gaussian distribution. A similar condition also applies to the length-based burst. The length-based burst has predefined packet size and length, and with the increasing burst time, the burst time will follow the Gaussian distribution. Also, long-range transmission only affects burst time but not burst length [20]. In a telephone network, a Poisson traffic model is used to predict traffic behavior. One of the important properties for the Poisson traffic model is Self-similarity which means that different data behave similarly at different time scale. Therefore, the traffic model in the OBS network is generally based on time followed by passion distribution [21].

Problem Statement

Though Optical Burst Switching (OBS) networks have the benefits of both Optical Circuit Switching and Optical Packet Switching technologies, due to lack of buffer to enqueue the bursts when contention happens, they face challenges to avoid traffic loss.

Background

Optical Networks (Metro, Long haul and Subsea) are facing unprecedented growth and challenges to deliver a quality end-user experience. One of the biggest challenges is to support a consistent increase in high demand for reliable bandwidth at lower cost.

Intense need for bandwidth led to multiple innovations and researches in Optical Networks and Engineering Physics. Wavelength Division Multiplexing (WDM) was a ground-breaking innovation as it allowed transmission of multiple wavelengths on the same fiber. There are three different WDM technologies – (1) Optical Circuit Switching, (2) Optical Packet Switching, and (3) Optical Burst Switching.

Optical Circuit Switching

In Optical Circuit Switching networks, the Ingress Node waits for a certain interval to aggregate bursts to the same destination and then it sends a control packet to create Optical Cross Connects (OXC) up to the Egress Node based on path computed by routing protocols such as OSPF.

Traffic from each ingress interface is mapped to explicit Timeslots and these Timeslots are reserved from end to end node. For example, in Figure 2 we see that Optical Router R1 has three different ingress interfaces each receiving traffic which is represented by a different color. Let's say each ingress interface can send one frame per second, then in such a case, we need to aggregate 3 egress interfaces so that it can send three frames per second in order to avoid any traffic loss. Furthermore, we see that the traffic from the first ingress interface is assigned Timeslot 1 (TS1), the second one is assigned TS2 and the third one is assigned TS3.

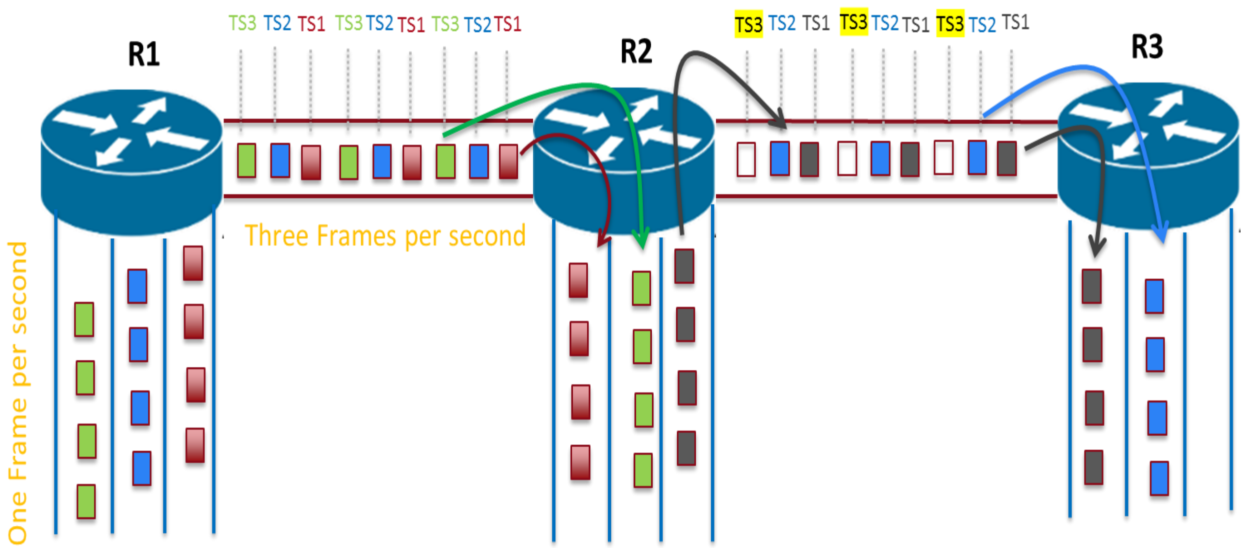


Figure 3: Timeslots reservations in Optical Circuit Switching$

$ www.infinera.com

We further see that at the Optical Router R2, two of the traffic is dropped (TS1 and TS3) and new traffic is added TS1. However, since TS2 continued till Optical Router 3, there has been no change in the bandwidth supported between the egress interfaces in R1, and R2. In other words, though only two Timeslots are used between R2 and R3, we continue to send empty TS3 because from node R1 to R3, the blue traffic is assigned TS2. Therefore, we see that there is a waste of bandwidth when traffic/bursts are added/dropped in the intermediate Optical Add/Drop Multiplexers (OADM). On the other hand, Optical Circuit Switching guarantees that the path is available from end to end for the bursts as the control packets are sent first to create OXCs before sending the data packet. It also helps to identify the network issues throughout the path (because in that case, the control packets would return errors) and hence provides high reliability.

Optical Packet Switching

In Optical Packet Switching, there is no need to assign respective timeslots to each traffic received on different ingress interfaces. Therefore, unlike in Optical Circuit Switching where a control packet has to be sent to reserve the timeslots and creating OXC across the best path before sending the bursts, traffic is sent on the OXC as and when they are received. Since bursts are not mapped to individual timeslots but they still need to be identified independently at each node, a label is attached to each burst based on destination using label assignment protocols such as Multiprotocol Label Switching (MPLS).

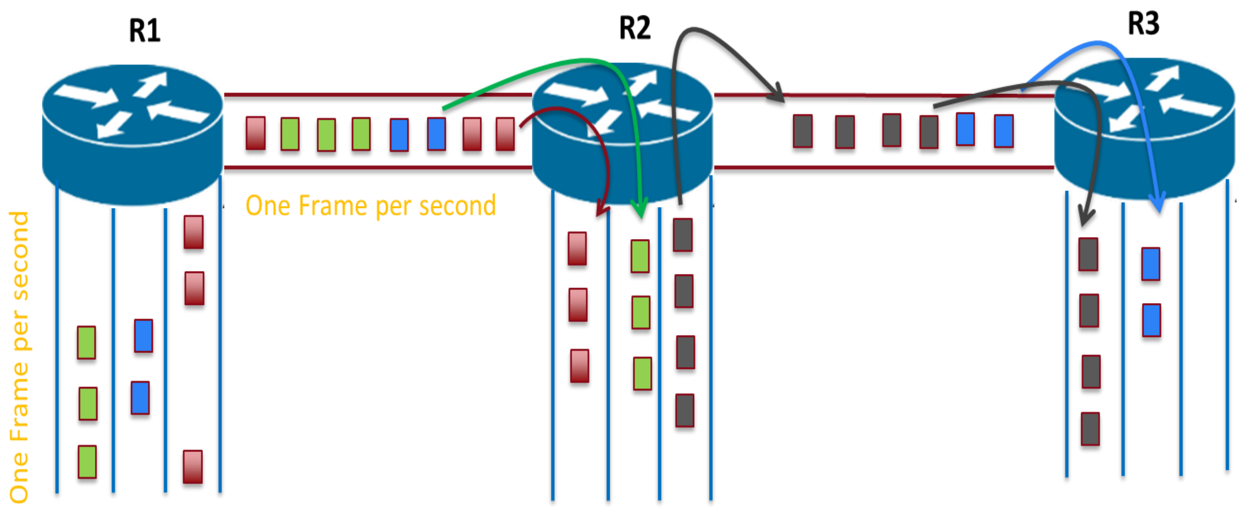


Figure 4: Traffic transmission in Optical Packet Switching$

$ www.infinera.com

Figure 3 shows an example of how traffic is transmitted in Optical Packet Switching. We see that when Optical Router R1 receives traffic on Ingress interfaces, without assigning a particular timeslot to each traffic types, it attaches a label and sends it to the next hop. Since no timeslot has to be assigned to the traffic, they can be transmitted as and when they come on the ingress interface without delaying, unlike Optical Circuit Switching where control packet has to be sent first. We see further that when two traffic types ■ and ■ are dropped in the intermediate Optical Router R2 and another traffic is added at R2 to R3, there are no fixed timeslot assignments. Therefore, as and when the traffic reaches the interfaces and the routing decision is made for them, they are forwarded without any delays. If there is no traffic to be transmitted, the link can be used by another type of traffic and there is no requirement to force send empty timeslots unlike in Optical Circuit Switching.

The above design makes Optical Packet Switching as a more affordable solution and hence reducing the overall cost of traffic transmission. Since there is no need to allocate respective timeslots to each traffic type, there is also no need to aggregate egress interfaces and hence we can use low bandwidth interface which can send one frame per second, similar to the ingress interfaces.

However, this leads to a need to ingress buffers to store ingress traffic until there is enough bandwidth in the egress link to send them. Moreover, since OXC are not created before the actual data is sent, there is no guarantee that the data will be transmitted successfully throughout the computed path to the destination node. Nonetheless, there are new technologies such as Bidirectional Forwarding Detection (BFD) and MPLS Operation Administration and Maintenance (MPLS OAM) are used to provide higher reliability as of Optical Circuit Switching networks.

Optical Burst Switching

Optical Burst Switching combines the capabilities of both Optical Circuit Switching and Optical Packet Switching technologies. When the traffic is received at the ingress nodes from two different ingress interfaces, the Optical node waits for a particular time interval to aggregate bursts to the same destination. Then a control packet is calculated based on the best path computed by routing protocols such as OSPF and the control packet is sent before the actual traffic so that the OXC can be created and respective bandwidth could be reserved in advance.

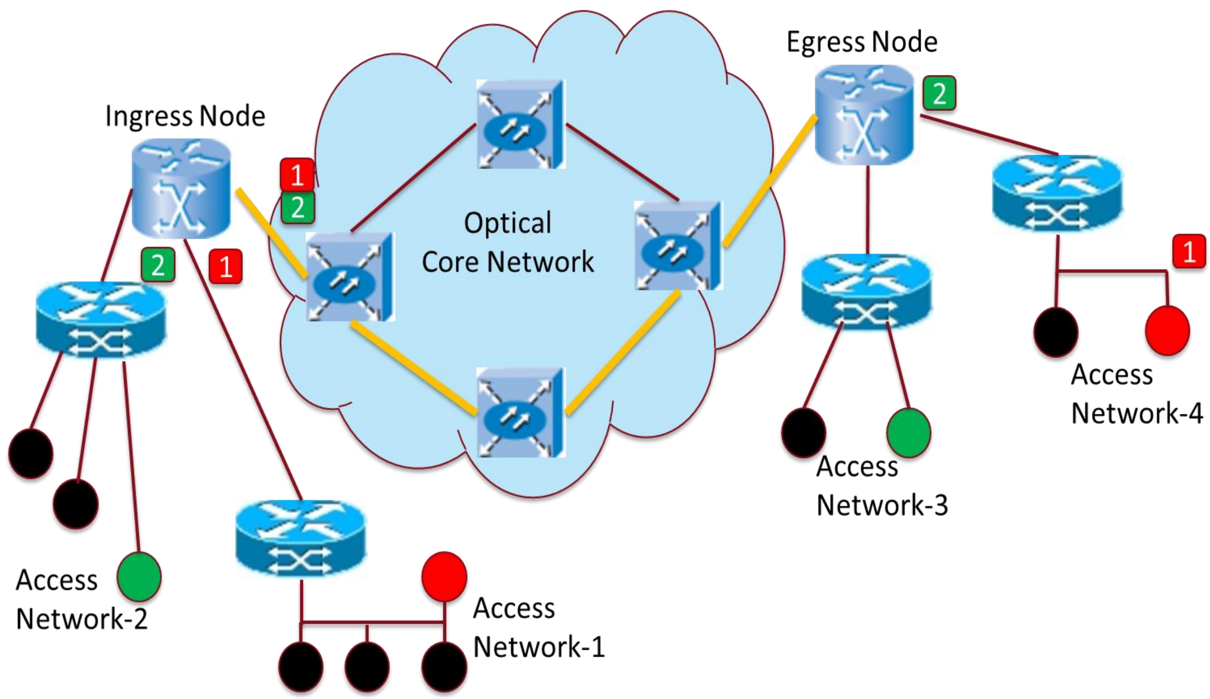


Figure 5: Traffic transmission in Optical Burst Switching

Unlike Optical Circuit Switching where after sending the control packet, the ingress node waits until the OXC create confirmation is received before sending the actual traffic, here the Control packet is sent first immediately followed by data traffic without any delays. This reduces the time delay significantly still providing the benefit of guaranteed bandwidth reservation.

In Figure 4 we see that the Ingress Node aggregates the traffic from two different access networks and sends both the burst together. However, the best path is computed using routing protocols and the control packet sent before the data traffic reserves the bandwidth creating OXC beforehand (shown as an orange link). Since bandwidth is reserved before the traffic is sent, it guarantees that the traffic will reach the destination node, unlike in Optical Packet Switching where there is no guarantee of bandwidth availability. However, when two or more traffic arrives for the same destination at the same time, one of the bursts has to be dropped since the bandwidth is already in use for the first traffic and there is no buffer to store the other burst unlike in Optical Packet Switching where the traffic can be queued in ingress queues.

Table 1: Summary of differences between WDM technologies

Attributes	Optical Circuit Switching	Optical Packet Switching	Optical Burst Switching
Bandwidth Utilization	Low	High	High
Setup Latency	High	Low	Low
Optical Buffer	Not Applicable	Required	Not Applicable
Signaling Scheme	Bidirectional (Out of Band)	Unidirectional (In-Band)	Unidirectional (Out of Band)
Overall cost	High	Low	Low
Reliability	High	Low	High

Contention in Optical Burst Switching (OBS) Networks

Contention happens when two or more bursts for the same destination reaches the ingress nodes at the same time. In Figure 5 we see that the Optical Core router is connected to two different Ingress Nodes – Node 1 and Node 2. Furthermore, the burst from Access Node 2 reaches first to Ingress Node 1 compared to the burst from Access Network 1 reaching Ingress Node 2. Due to which, Ingress Node 1 creates an OXC from Node 1 all the way till Egress Node 1 reserving the bandwidth in Optical Core Routers based on the optimal path computed by the routing algorithm. The optimal path is highlighted with an orange link.

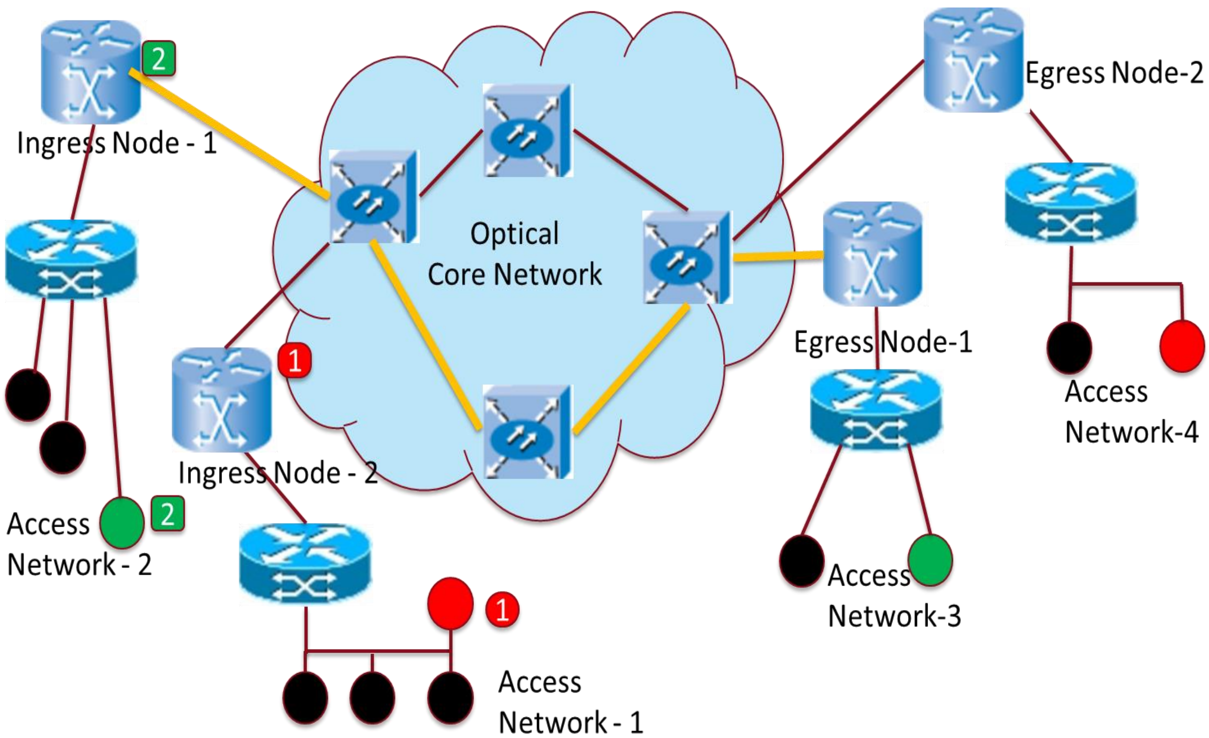


Figure 6: Contention in Optical Burst Switching

In a conventional OBS network, the traffic from Access Network 1 will be dropped since the optimal path is already in use and there is no buffer to enqueue the burst temporarily in the Ingress Node 2. This would lead to inconsistent quality of service to the end users. Multiple types of research are conducted by Universities and Industry leaders such as Ciena and Coriant to provide **scalable and secure network** capacity that is both faster and smarter giving them a huge competitive advantage.

Deflection routing is one such solution which can effectively reduce the traffic loss due to contention hence providing high-quality end-user experience.

Deflection Routing

Deflection Routing is a viable contention resolution scheme that is employed in buffer-less networks. It comprises of algorithms that can deflect the packets/traffic through another sub-optimal path in case two or more bursts have to be routed through the same outgoing link (based on routing decisions that computed the best path). The algorithms to identify the sub-optimal path to the same destination could be predictive based or non-predictive based. Predictive-based algorithms use machine learning to compute the next based sub-optimal path that would guarantee to reduce further deflection leading to more reliable delivery of the burst to the destination node.

Let's take an example of applying a basic deflection routing without any prediction logic. In Figure 5 when two bursts for the same destination are reaching two different ingress nodes in an Optical Core network, and the Ingress Node 1 is able to create an OXC until Egress Node 1 using the optimal path computed by the routing protocol, Ingress Node 2 will not drop the bursts received from Access Network 1 due to unavailability of bandwidth on the optimal links. Using deflection routing algorithms, Ingress Node 2 will recompute the path and send the burst through the non-optimal path to the destination Egress Node 2. This way, since the burst is deflected through another path, it is guaranteed to reach its destination, though it is not the best path to send the traffic.

Implementation and Results

Tools Used

In our study, we mainly used three different tools to measure the performance of different Q-Learning based Deflection Routing algorithms.

Network Simulator (ns-3)

ns-3 is an open source software that can be used to simulate network discrete-event, mainly used for research and educational purposes. Moreover, ns-3 can also be used to interconnect virtual machines by adding links as well as real-time scheduling of ns-3-generated packets on physical network devices.

Google Test Framework (G-Test)

GTest is a Unit Testing Library from Google primarily used for C++ programming language. In our study, we used Parameterized Tests to validate our network topologies in order to compute the performance of different Q-Learning based Deflection Routing algorithms.

Boston University Representative Internet Topology Generator (BRITE)

BRITE is a synthetic topology generator that accurately reflects many aspects of the actual Internet topology. Using BRITE, we generated multiple **Waxman** topologies with node counts of **10, 20, 50, 100, 200, 500 and 1000** to validate **Q-NDD, PQDR** Deflection Routing Algorithms.

iDef Framework [6]

iDef Framework designed to facilitate the development of reinforcement learning based deflection routing protocols. It comprises of four different components or modules. Each module can be individually replaced without changing the entire design which makes it very flexible and easily portable. Deflection Manager is the module that glues the other iDef modules including ***mapping***, ***decision making,*** and ***signaling*** components. Moreover, it interacts with the IP Routing component that computes the best route for the bursts and the OBS Interface Card.

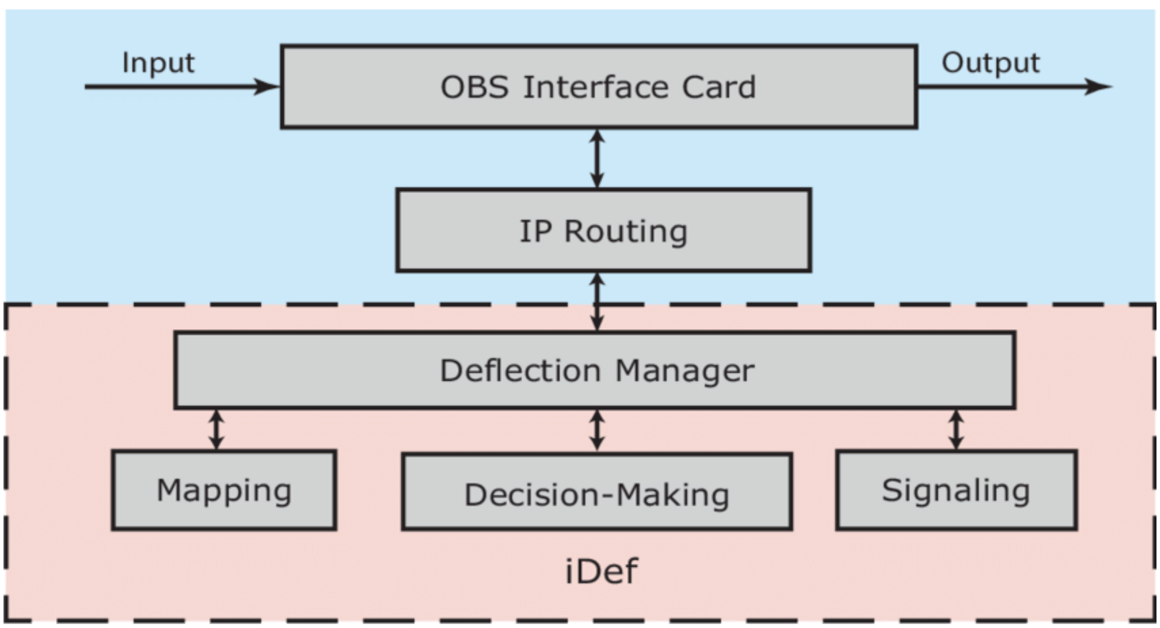


Figure 7: iDef Framework for Deflection Routing

When the control packet for a burst is received by the OBS interface card to create OXC, the interface card delegates the burst header to the Deflection Manager. Then the deflection manager refers to the IP Routing Table is deriving the outgoing link in order to create the OXC. If the outgoing link is busy (i.e., there is already an OXC on that link) then the burst header is further processed by the deflection manager. Otherwise, the OXC is created and the burst header is sent to the next hop in the best path computed by the routing protocol at the ingress node. If deflection manager needs to process the burst header further due to a busy outgoing link, it creates a list of environment variables including the burst's destination, the blocking state of the egress link and the next hop based on the best path computed by the ingress node. This list is provided to the mapping module which maps all or few of those environment variables to generate an integer called ***state***. This state is passed to the decision-making module which runs a learning algorithm based on the heuristic data such as statistics, the history, and other related information about the topology and the previous deflection decisions. The output of the decision-making module is the best possible alternate path/outgoing link for the deflected burst. The signaling module adds the header to the control packet to the deflected burst. Moreover, it also inspects the burst header received from the deflection manager and sends feedback messages.

Design

Figure 7 shows the steps in our study to compute and compare the performance of various Q-Learning based Deflection routing algorithms.

Step 1: Define the Simulation Attributes, for instance, the Line Speed (we used 1 Gbps link speed), Drop wait time (we set it to 50ms) and Packet Size of 12500 bytes.

Step 2: Create Aggregators such as Hop's aggregators, Bytes Tx aggregators, and Bytes Rx aggregators.

Step 3: Create Waxman topology using BRITE by adding the topology details in the configuration file such as a total number of nodes.

Step 4: Add Links and Install IP Protocol Stack. Moreover, assign IP addresses to the links.

Step 5: Enable iDef Signaling to run deflection routing algorithm.

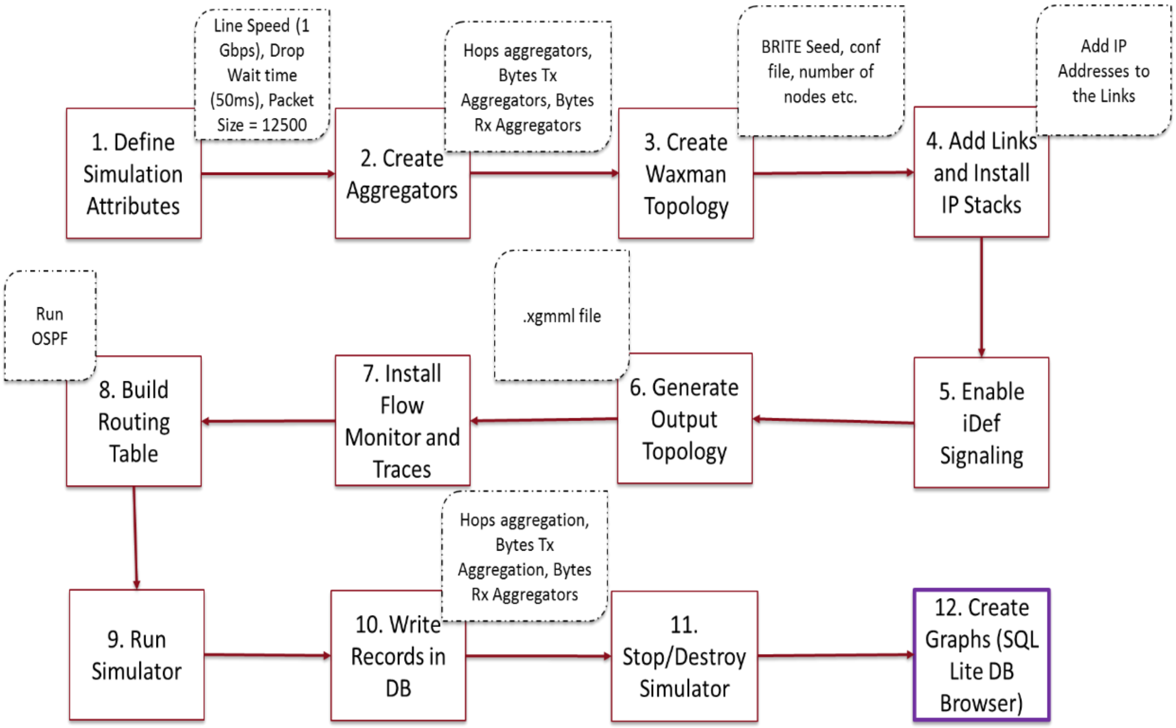


Figure 8: Control Flow for Performance Computation

Step 6: Generate XGMML output file with the defined number of nodes.

Step 7: Install Flow Manager and traces for the purpose of debugging flows.

Step 8: Run OSPF to build a routing table across the topology.

Step 9: Run ns-3 simulator for each of the parameterized GTests.

Step 10: Write the record in the database.

Step 11: Stop and Destroy the simulator.

Step 12: Read from the database to create Graphs using SQL Lite DB browser.

Deep Dive

iDef Framework is used extensively for this project and as explained before can be found in this link below.

[Intelligent Deflection Routing in Buffer-less network]

Download Software

To download the software package, either click on the link below to directly fetch the tar files.
[idef_framework.tar]

Or, if you have Mercurial installed, the following command can be used:

http://www.bitbucket.org/shaeri/hmm-deflection

Below are the software prerequisites to run the framework:

- Ubuntu 16.04 LTS (64 bit)
- Python 2.5.6
- ns-3.11 (and other prerequisite software as per ns-3 tutorial)
- DB Browser (SQLite)
- Cytoscape (for topology in XGMML output files)

Steps to build the framework

Run the command below to configure and build the framework

```
>  CXXFLAGS='-Wno-error -std=c++98' ./waf --enable-examples configure
>  ./waf
```

Running the test cases

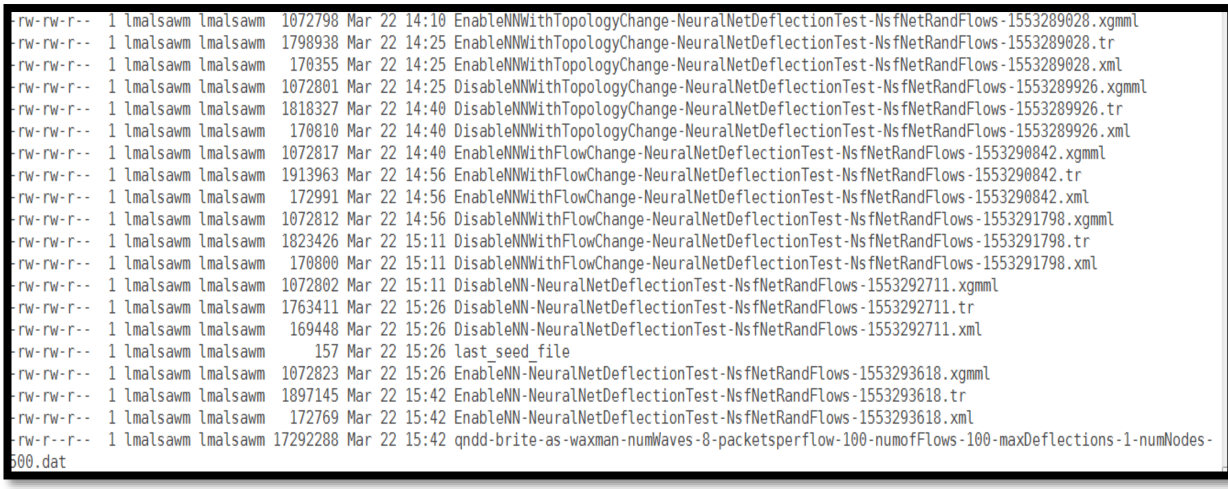
The test cases can be found in the examples directory. All test cases which are built in the libraries are available in wscript where the definition of each test case is given.

Below are examples of how the test case can be built

```
>  ./waf --run 'nn-nsf-net --NumFlows=2200 --Alpha=0.1 --
RewardZoffset=-100 --PacketsPerFlow=50 --MaxDeflections=2 --
numWavelenghts=64 --useEpisodicReinforcement=0 --
gtest_filter=*EnableNNWithFlowChange/NeuralNetDeflectionTest* --
EnableNsAnimation=0 --EnableProgressBar=1 --gtest_catch_exceptions=0'

> ./waf --run 'rldrs-brite-waxman --NumFlows=2200 --nNodes=1000'
```

```
lmalsawm@lmalsawm-VirtualBox:~/repos/hmm-deflection$ ./waf --run qndd-
brite-waxman
Waf: Entering directory `/home/lmalsawm/repos/hmm-deflection/build'
Waf: Leaving directory `/home/lmalsawm/repos/hmm-deflection/build'
'build' finished successfully (0.347s)
Modules built:
aodv                      applications              bridge
.
.
wimax                     xgmml-writer

Don't know how to configure dynamic library path for the platform
'linux4'; assuming it's LD_LIBRARY_PATH.
[==========] Running 6 tests from 6 test cases.
[----------] Global test environment set-up.
[----------] 1 test from
EnableNNWithTopologyChange/NeuralNetDeflectionTest
[ RUN      ]
EnableNNWithTopologyChange/NeuralNetDeflectionTest.NsfNetRandFlows/0
Place seed used: 929 18840 38318
Connect seed used: 26883 38699 1089
Edge conn seed used: 53489 32975 5141
Grouping seed used: 34898 48253 21264
Assignment seed used: 44174 29196 31893
Bandwidth seed used: 2445 3871 43415
Parsing AS Waxman model...
Placing nodes...
Interconnect nodes...
Assigning bandwidth...
Topology is connected!!!
Place seed stored: 18793 48888 44078
Connect seed stored: 55689 41948 6802
Edge Connect seed stored: 53489 32975 5141
Grouping seed used: 34898 48253 21264
Assignment seed stored: 44174 29196 31893
Bandwidth seed stored: 42073 54539 17144
Done!
Number of links: 1500
[       OK ]
EnableNNWithTopologyChange/NeuralNetDeflectionTest.NsfNetRandFlows/0
(897787 ms)
[----------] 1 test from
EnableNNWithTopologyChange/NeuralNetDeflectionTest (897787 ms total)

[----------] 1 test from
DisableNNWithTopologyChange/NeuralNetDeflectionTest
.
.
.
[----------] Global test environment tear-down
[==========] 6 tests from 6 test cases ran. (5535132 ms total)
[  PASSED  ] 6 tests.
```

Figure 9: Sample Output



```
-rw-rw-r--  1 lmalsawm lmalsawm  1072798 Mar 22 14:10 EnableNNWithTopologyChange-NeuralNetDeflectionTest-NsfNetRandFlows-1553289028.xgmml
-rw-rw-r--  1 lmalsawm lmalsawm  1798938 Mar 22 14:25 EnableNNWithTopologyChange-NeuralNetDeflectionTest-NsfNetRandFlows-1553289028.tr
-rw-rw-r--  1 lmalsawm lmalsawm   170355 Mar 22 14:25 EnableNNWithTopologyChange-NeuralNetDeflectionTest-NsfNetRandFlows-1553289028.xml
-rw-rw-r--  1 lmalsawm lmalsawm  1072801 Mar 22 14:25 DisableNNWithTopologyChange-NeuralNetDeflectionTest-NsfNetRandFlows-1553289926.xgmml
-rw-rw-r--  1 lmalsawm lmalsawm  1818327 Mar 22 14:40 DisableNNWithTopologyChange-NeuralNetDeflectionTest-NsfNetRandFlows-1553289926.tr
-rw-rw-r--  1 lmalsawm lmalsawm   170810 Mar 22 14:40 DisableNNWithTopologyChange-NeuralNetDeflectionTest-NsfNetRandFlows-1553289926.xml
-rw-rw-r--  1 lmalsawm lmalsawm  1072817 Mar 22 14:40 EnableNNWithFlowChange-NeuralNetDeflectionTest-NsfNetRandFlows-1553290842.xgmml
-rw-rw-r--  1 lmalsawm lmalsawm  1913963 Mar 22 14:56 EnableNNWithFlowChange-NeuralNetDeflectionTest-NsfNetRandFlows-1553290842.tr
-rw-rw-r--  1 lmalsawm lmalsawm   172991 Mar 22 14:56 EnableNNWithFlowChange-NeuralNetDeflectionTest-NsfNetRandFlows-1553290842.xml
-rw-rw-r--  1 lmalsawm lmalsawm  1072812 Mar 22 14:56 DisableNNWithFlowChange-NeuralNetDeflectionTest-NsfNetRandFlows-1553291798.xgmml
-rw-rw-r--  1 lmalsawm lmalsawm  1823426 Mar 22 15:11 DisableNNWithFlowChange-NeuralNetDeflectionTest-NsfNetRandFlows-1553291798.tr
-rw-rw-r--  1 lmalsawm lmalsawm   170800 Mar 22 15:11 DisableNNWithFlowChange-NeuralNetDeflectionTest-NsfNetRandFlows-1553291798.xml
-rw-rw-r--  1 lmalsawm lmalsawm  1072802 Mar 22 15:11 DisableNN-NeuralNetDeflectionTest-NsfNetRandFlows-1553292711.xgmml
-rw-rw-r--  1 lmalsawm lmalsawm  1763411 Mar 22 15:26 DisableNN-NeuralNetDeflectionTest-NsfNetRandFlows-1553292711.tr
-rw-rw-r--  1 lmalsawm lmalsawm   169448 Mar 22 15:26 DisableNN-NeuralNetDeflectionTest-NsfNetRandFlows-1553292711.xml
-rw-rw-r--  1 lmalsawm lmalsawm      157 Mar 22 15:26 last_seed_file
-rw-rw-r--  1 lmalsawm lmalsawm  1072823 Mar 22 15:26 EnableNN-NeuralNetDeflectionTest-NsfNetRandFlows-1553293618.xgmml
-rw-rw-r--  1 lmalsawm lmalsawm  1897145 Mar 22 15:42 EnableNN-NeuralNetDeflectionTest-NsfNetRandFlows-1553293618.tr
-rw-rw-r--  1 lmalsawm lmalsawm   172769 Mar 22 15:42 EnableNN-NeuralNetDeflectionTest-NsfNetRandFlows-1553293618.xml
-rw-r--r--  1 lmalsawm lmalsawm 17292288 Mar 22 15:42 qndd-brite-as-waxman-numWaves-8-packetsperflow-100-numofFlows-100-maxDeflections-1-numNodes-
500.dat
```

Figure 10: List of output files

As shown previously from the output traces, the test case is run six times by randomly changing the topology and flow for each test. The random generator uses the seed file "*last_seed_file*" to generate different cases. Individual traffic traces for each test run are written in the XML file and TR file. The individual test topology has been saved in XGMML and can be viewed using Cytoscape software. The aggregated traces of the six tests run and congregated into the DAT file which is an SQLite table and can be check in an OS platform using DB Browser (SQLite).

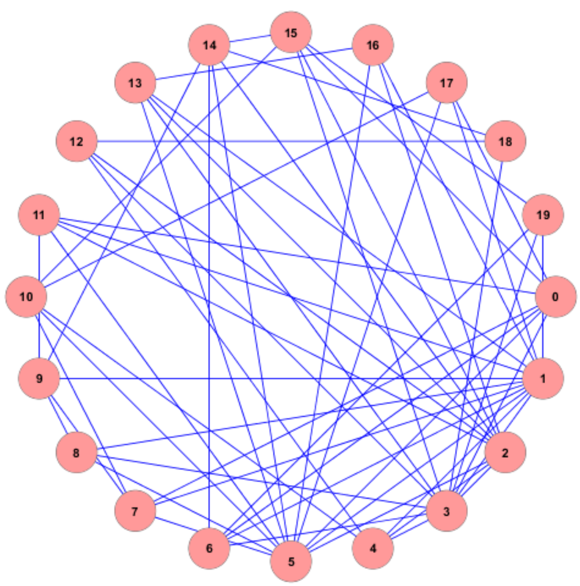


Figure 11: Example of the Waxman graph for 20 nodes

Waxman topology is generated using BRITE topology generator. Waxman topology is a class of randomly generated graphs and is widely used for generating routing algorithm in network simulation. It is widely used for generating internet-like topology as we can see why clearer using the figure below.

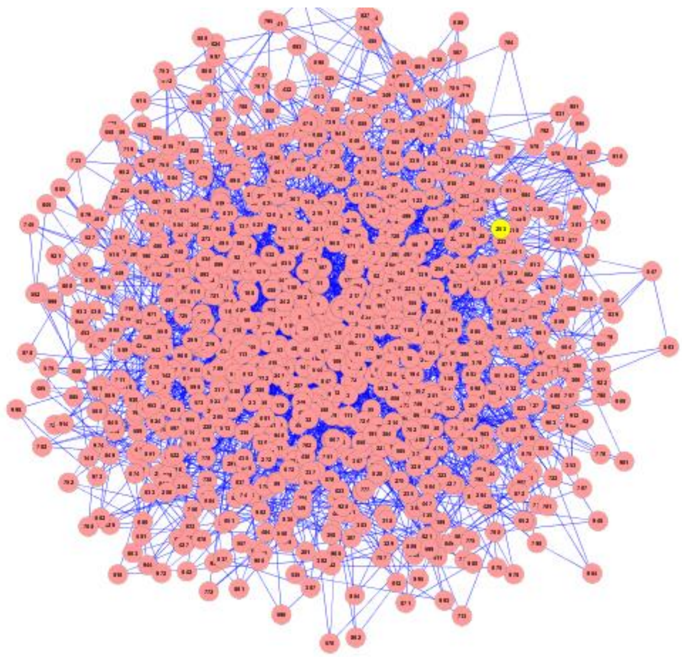


Figure 12: Example of the Waxman graph for 1000 nodes

Simulation Scenario

Table 2: Simulation Scenario

<table>
<tr><th>Algorithm</th><th>Number of nodes</th><th>Number of links</th><th>Number of flows</th></tr>
<tr><td rowspan="7">Q-NDD
PQDR</td><td>10</td><td>30</td><td>24</td></tr>
<tr><td>20</td><td>60</td><td>48</td></tr>
<tr><td>50</td><td>150</td><td>120</td></tr>
<tr><td>100</td><td>300</td><td>240</td></tr>
<tr><td>200</td><td>600</td><td>480</td></tr>
<tr><td>500</td><td>1500</td><td>1200</td></tr>
<tr><td>1000</td><td>3000</td><td>2400</td></tr>
</table>

Result

Burst loss probability: This variable is significant for comparing the contention scenario between different learning algorithms.

Burst loss probability is used to determine the probability of loss for the total transmitted packets.

$$Burst\ loss\ probability\ = \frac{Number\ of\ burst\ loss}{Total\ transmitted\ burst}$$

Equation 4: Burst loss probability

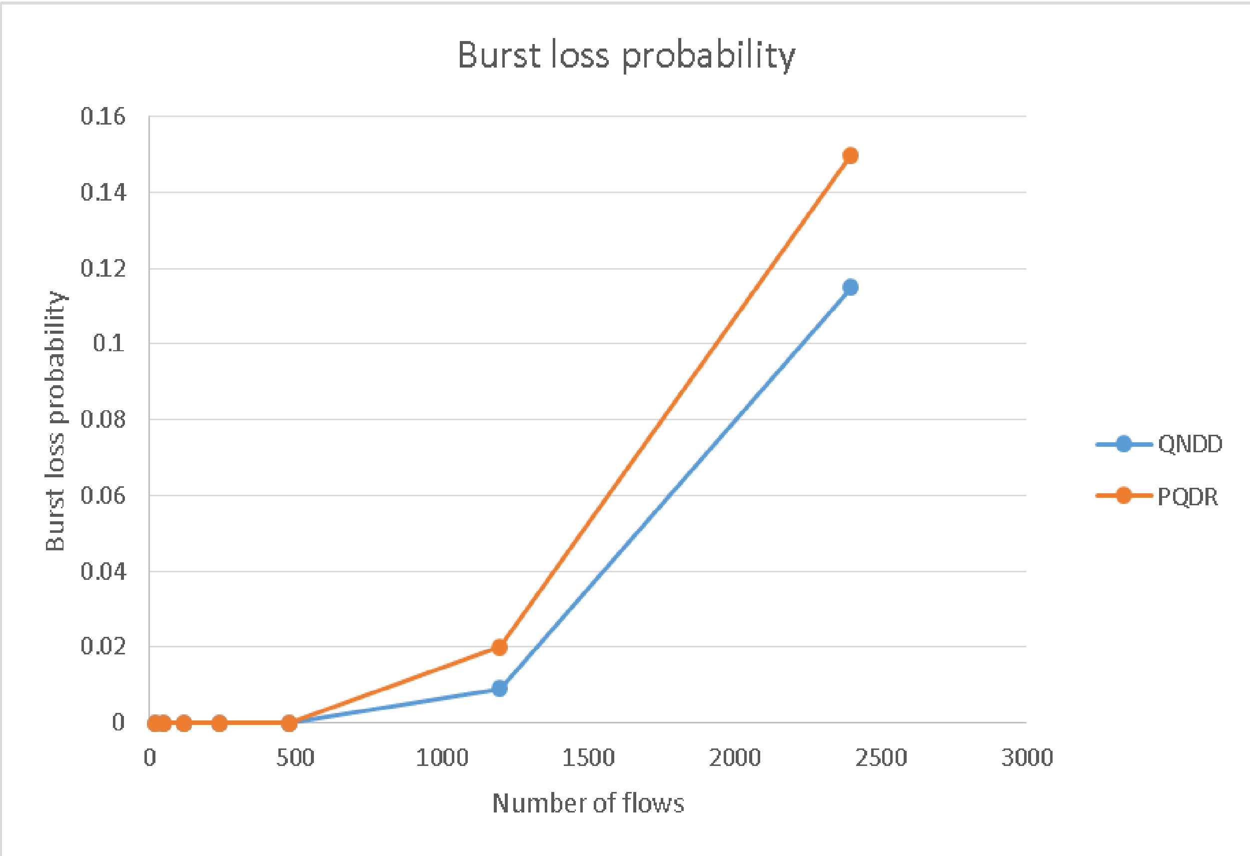


Figure 13: Burst loss comparison

The comparison shows that Q-NDD algorithm displays lower burst loss probability compared to the PQDR algorithm with an increase in the number of flows (or the size of the graph increases). This shows that **Q-NDD handles contention better than PQDR**.

Deflection ratio: Deflection ratio is another significant parameter that can be used for comparing the burst loss scenario. A higher number of deflected bursts implies lesser burst loss. A higher number of deflections is not desirable for traffic engineering because deflection causes an increase in traffic load to the network.

$$Deflection\ ratio = \frac{Number\ of\ deflected\ bursts}{Number\ of\ transmitted\ bursts}$$

Equation 5: Deflection ratio

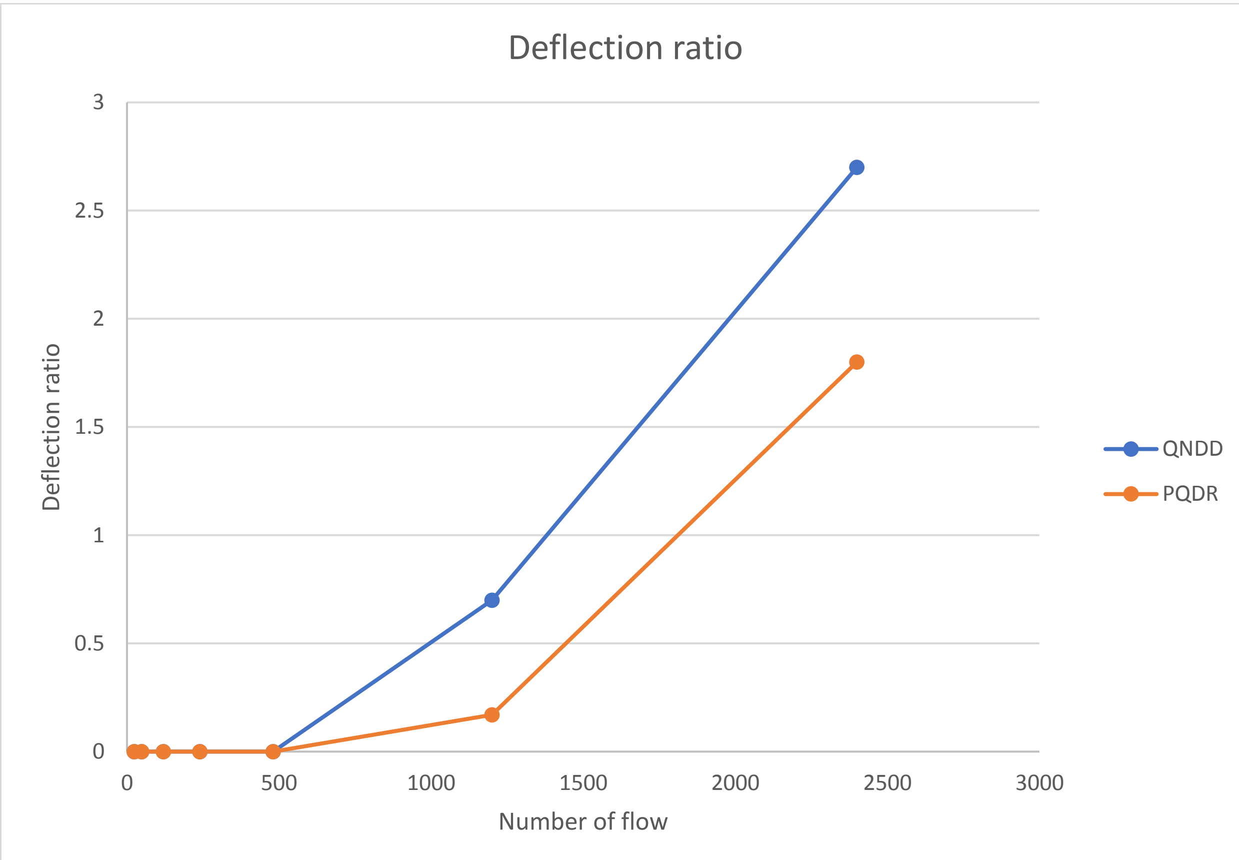


Figure 14: Deflection ratio comparison

The comparison shows that **Q-NDD algorithm deflects more number of bursts compared to the PQDR** routing algorithm.

An average number of hops: The average number of hops is the number of hops traveled by the burst.

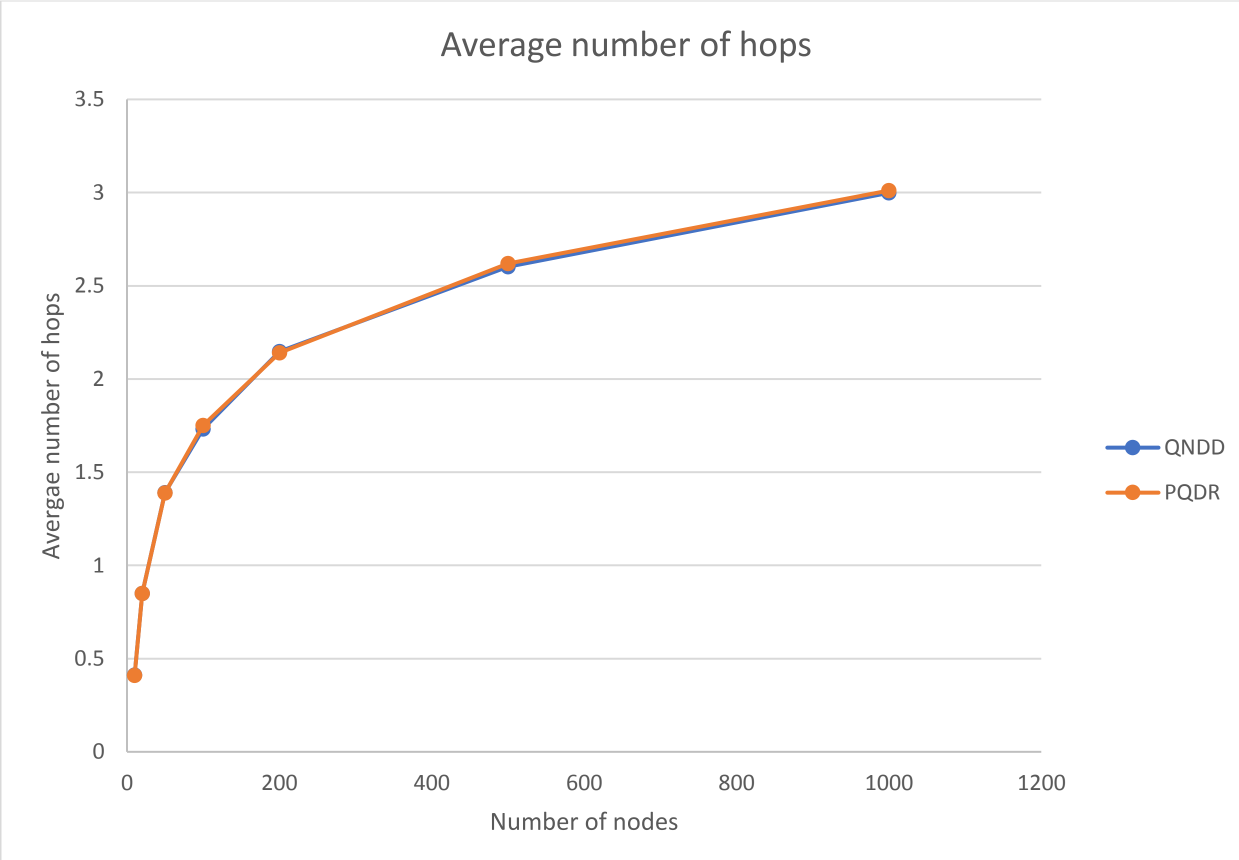


Figure 15: Average number of hops

As shown in the graph above, Q-NDD shows a similar trend as PQDR. Q-NDD is slightly higher than PQDR which shows using Q-NDD burst traveled more number of hops.

Discussion and Conclusion

Table 3: Result summary

Parameters	Q-NDD Algorithm	PQDR Algorithm
Burst loss probability	Low	High
Deflection ratio	High	Low
Average hop count	Equal	Equal

Both the algorithms solve the problem of contention, but Q-NDD algorithm performs much better as compared to the PQDR algorithm. PQDR Algorithm has higher burst loss probability. This means that packets have a higher chance to be lost than Q-NDD algorithm. PQDR sends the feedback signal when deflecting burst. The deflecting burst tends to go shorter paths which makes more congestion in the network, so the PQDR algorithm has higher burst loss probability. On the other hand, Q-NDD has a higher number of deflections, and both algorithms have equal average hop count. Therefore, deflection routing with Q-NDD algorithm in the OBS network has the best performance.

With this project, we were able to study Deflection routing and its implementation using different learning algorithms and its application in the buffer-less networks such as Optical Burst Switching (OBS) using ns-3. By understanding on the underlying technology of optical networking, we were able to get a wider perspective of how Deflection routing can be implemented to help improving the performance of high-speed networks and solve issues such as traffic loss. In addition, we have a better understanding about ns-3 simulator and experience in the different software tools such as Google Test Framework (G-Test) for unit testing of source code in C++ programming language and BRITE libraries for generating different types of random Waxman graph that are leveraged to exploit the benefits of the ns-3 simulator. In each step of the project, we study multiple related tools such as SQLite and xgmml to simulate the networks. As we have learned the function of the route learning algorithm and iDef framework, which can deploy on an existing network.

References

[1] "High Speed Networks," Lightwave, [Online]. Available: https://www.lightwaveonline.com/network-design/high-speed-networks.html. [Accessed 9 Feb 2019].

[2] S. Haeri and L. Trajkovic´, "Deflection Routing in Complex Networks," in *Complex Systems and Networks*, Berlin, Springer, 2015, pp. 395-422.

[3] C. Qioa and M. Yoo, "Optical burst switching (OBS) - a new paradigm for an optical internet," *Journal of High Speed Networks,* vol. 8, no. 1, pp. 69-84, Mar. 1999.

[4] A. Zalesky, H. Vu, Z. Rosberg, E. Wong and M. Zukerman, "OBS contention resolution performance," *Performance Evaluation,* vol. 64, no. 4, p. 357–373, May 2007.

[5] D. H. Patel and D. K. Kothari, "Overview and framework for dynamic deflection contention resolution in OBS network," in *2013 Nirma University International Conference on Engineering (NUiCONE)*, Ahmendabad, India, Nov 2013.

[6] S. Haeri and L. Trajkovic, "Intelligent deflection routing in buffer-less networks," *IEEE Transactions on Cybernetics,* vol. 45, no. 2, pp. 316-327, Feb. 2015.

[7] S. Waheed, "COMPARING OPTICAL PACKET SWITCHING AND OPTICAL BURST SWITCHING," Mawlana Bhashani Science and Technology University, Tangail, 2011.

[8] TeleGeography, "Submarine Cable Map," TeleGeography, [Online]. Available: https://www.submarinecablemap.com/. [Accessed 12 April 2019].

[9] L. P. Kaelbling, M. L. Littman and A. W. Moore, "Reinforcement learning: a survey," *J. of Artificial Intelligence Research,* vol. 4, p. 237–285, 1996.

[10] C. J. C. H. Watkins and P. Dayan, "Technical note, Q-learning," *Machine Learning,* vol. 8, no. 3, pp. 279-292, 1992.

[11] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: a reinforcement learning approach," *Advances in Neural Inform. Process. Syst.,* vol. 6, pp. 671-678, 1994.

[12] L. Peshkin and V. Savova, "Reinforcement learning for adaptive routing," *Proc. Int. Joint Conf. Neural Netw,* vol. 2, pp. 1825-1830, 2002.

[13] S. P. M. Choi and D. Y. Yeung, "Predictive Q-routing: a memory-based reinforcement learning approach to adaptive traffic control," *Advances in Neural Inform. Process. Syst,* vol. 5, pp. 945-951, 1996.

[14] K. L. Calvert, M. B. Dora and E. W. Zegura, "Modeling Internet topology," *IEEE Commun.Mag.,* vol. 35, no. 6, pp. 160-163, 1997.

[15] A.-L. Barab´asi and R. Albert, "Emergence of scaling in random networks," *Science,* vol. 286, no. 5439, pp. 509-512, 1999.

[16] University of Groningen, "Scale-free networks," University of Groningen, [Online]. Available: https://www.futurelearn.com/courses/complexity-and-uncertainty/0/steps/1855. [Accessed 12 April 2019].

[17] X. Yu, J. Li, X. Cao, Y. Chen and C. Qiao, "Traffic statistics and performance evaluation in optical burst switched networks," *J. Lightw. Technol,* vol. 22, no. 12, pp. 2722-2738, 2004.

[18] A. Zalesky, Z. R. H. Vu, E. Wong and M. Zukerman, "OBS contention resolution performance," *Perform. Eval,* vol. 64, no. 4, pp. 357-373, 2007.

[19] S. Iyer, S. Bhattacharyya, N. Taft and C. Diot, "An approach to alleviate link overload as observed on an IP backbone," *Proc. IEEE INFOCOM,* vol. 1, pp. 406-416, 2003.

[20] W. W.-K. Thong and G. Chen, "Jittering performance of random deflection routing in packet," *Communications in Nonlinear Science and Numerical Simulation,* vol. 18, no. 3, pp. 616-624, 2013.

[21] X. Gao and M. Bassiouni, "Improving fairness with novel adaptive routing in optical burstswitched networks," *J. Lightw. Technol,* vol. 27, no. 20, pp. 4480-4492, 2009.

[22] S. Haeri, M. Arianezhad and L. Trajkovi´, "A predictive Q-learning-based algorithm for deflection routing in buffer-less networks," *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics,* pp. 764-769, 2013.