Real-time Traffic Management in an RTP/RTCP Environment using CoDel and DRR

By

Atique Ahmed

B.Eng. Mehran University of Engineering and Technology, Jamshoro, Pakistan 2011

A Report Submitted in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF ENGINEERING

in the Department of Electrical and Computer Engineering

Real-time Traffic Management in an RTP/RTCP Environment using CoDel and DRR

By

Atique Ahmed

B.Eng. Mehran University of Engineering and Technology, Jamshoro, Pakistan 2011

Supervisory Committee

Dr. T. Aaron Gulliver, Supervisor

(Department of Electrical and Computer Engineering)

Dr. Mihai Sima, Departmental Member

(Department of Electrical and Computer Engineering)

ABSTRACT

Communication systems involve streaming media including video conferencing, television services, and web-based video applications. This real-time traffic is carried using the Real-time Transfer Protocol (RTP). The Real-time Transmission Control Protocol (RTCP) is used to provide feedback on the Quality of Service (QoS) and information about the network. The increasing number of packets in the network causes bufferbloat. Bufferbloat occurs when routers buffer too much data. There has been significant research in Active Queue Management (AQM) to overcome bufferbloat issues. In this project, two different scheduling schemes, namely Controlled Delay (CoDel) and Deficit Round Robin (DRR) are investigated for an RTP/RTCP environment. CoDel was introduced recently by Nicholas and Jacobson [3]. In the past, CoDel was evaluated using Transmission Control Protocol (TCP) and File Transfer Protocol (FTP) traffic. In this project, the performance of CoDel and DRR is evaluated using the RTP/RTCP protocols.

Table of Contents

List of Figures

List of Tables

Glossary

AQM	Active Queue Management
CoDel	Controlled Delay
DRR	Deficit Round Robin
ECN	Explicit Congestion Network
FTP	File Transfer Protocol
IETF	Internet Engineering Task Force
NS-2	Network Simulator 2
PQM	Passive Queue Management
RED	Random Early Detection
RTCP	Real-time Transfer Control Protocol
RTP	Real-time Transfer Protocol
RTT	Round-Trip Time
TCP	Transmission Control Protocol

ACKNOWLEDGEMENTS

I would like to thank Salahuddin Jokhio, Manzoor Abro, and Noman Lashari who helped me during my entire M.Eng project. I would also like to thank Dr. T Aaron Gulliver for his support throughout my Master's degree.

Dedication

I dedicate this work to my lovely parents (Ama and Baba Saen) for their moral and financial support and everything they have given to me my entire life.

Chapter 1 Introduction

A network is formed by a series of nodes interconnected by communication links. As the number of users and data increases in the network, traffic congestion occurs. Network congestion refers to the state where a node or link carries too much data. This overload affects network service quality, resulting in queue delay, packet loss, and low throughput. When user demands exceed the available network resources it causes network congestion. It is necessary to prevent network congestion since it affects the performance of the network.

Queue delay is the time a packet waits in a queue before it is transmitted to the destination. Packet loss results in wasted resources and possible network failure. The number of packets transferred from the sender to receiver in a specified amount of time is known as throughput. When throughput drops to zero, all data transfer through the network stops and the response time will be infinite. The aim of, then, is to avoid congestion.

An increase in the buffer size due to an increase in network traffic is called bufferbloat and creates packet delay which ultimately degrades network performance [14]. Packets in queues are served by routers which forward packets to destination nodes. In routers, Active Queue Management (AQM) operates on the queues to reduce packet delay, increase throughput, and avoid packet loss. AQM proactively drops packets before the router buffer space is full. AQM has been proposed as a solution for the bufferbloat problem [1].

Several AQM schemes such as CoDel and DRR have been proposed, but they have not been tested in an RTP/RTCP environment. In this project, a bottleneck dumbbell topology is used to analyze real-time traffic flow for CoDel and DRR queues. A bottleneck occurs in a network when there are too many users attempting to access specific resources.

Globally, video traffic will be 82 percent of all consumer internet traffic by 2020, up from 70 percent in 2015 [9]. Such an increase indicates the importance of multimedia traffic. Therefore, transmitting multimedia traffic with real-time quality of service (QoS) limitations is an important research challenges for next coming years.

1.1 RTP/RTCP Protocols

Real-time traffic dominates current communication systems. The Real-time Transport Protocol (RTP) is a network protocol used for sending real-time data such as audio and video over a network. RTP is used to transmit multimedia content on a real-time basis. RTP guarantees neither data delivery nor packet delivery in order. The main functions of RTP include [17]:

- Identification of the source sending the RTP packets.
- Sequence numbers for the RTP packets.

The function of the Real-time Transmission Control Protocol (RTCP) is to monitor the network Quality of Service (QoS). This protocol provides session feedback information concerning the network conditions. Therefore, RTP and RTCP are frequently used in a joint manner.

The main functions of RTCP are [16]:

- Network QoS measurements (packet loss ratio and delay jitter), and send receiver reports.
- Identification of the source sending the RTCP packets.

1.2 Problem Statement

The internet is now mostly comprised of multimedia traffic. Technical improvements are continuing to decrease the performance requirements for multimedia applications to avoid data loss, packet delay, and low throughput. In this project, we consider the problem of network congestion caused by bufferbloat. Two AQM schemes, namely Controlled Delay (CoDel), and Deficit Round Robin (DRR), are compared to determine the behavior of real-time traffic in various bottleneck conditions.

CoDel is a recently developed AQM mechanism [3]. Based on an analytical comparison of Transmission Control Protocol (TCP), File Transfer Protocol (FTP) and real-time traffic management using different AQM mechanisms, CoDel is considered to be the best to mitigate bufferbloat [3]. Relevant previous work includes real-time traffic performance using DRR, TCP evaluation using CoDel and Random Early Detection (RED), and the effectiveness of CoDel for AQM [2], [4], [6]. The goal of this project is to evaluate CoDel and DRR for real-time traffic management using RTP and RTCP

1.3 Report Organization

The report is organized as follows. Chapter 1 introduced the concept of network and network congestion, and the RTP/RTCP protocols. It also describes the method and goal of this project. Chapter 2 introduces AQM, CoDel and DRR. This chapter also explains in detail the working of the AQM algorithms with relevant flowcharts. Chapter 3 provides a performance comparison of between CoDel and DRR. The simulation steps, tools, and network topology involved in CoDel and DRR are also described. Chapter 4 finalizes the report with some conclusions and ideas for future work.

Chapter 2 Network Traffic Scheduling

2.1 Active Queue Management

During the past few years, there has been significant growth in both the number of users and the demand for services over networks leading to traffic congestion. TCP is an end-to-end control protocol which is used to detect congestion in a network [7]. However, routers present in the network can also be used to detect congestion. To mitigate congestion, routers make use of queue management schemes to control the growth of queues at the routers. The traditional queue management scheme is Passive Queue Management (PQM), which uses a drop-tail mechanism. In PQM, packets are dropped by the routers only when the queues at the routers become full [6]. PQM has the advantage of being easily deployable on routers, however, it has multiple shortcomings including persistently full buffers, lock-out [6] and global synchronization [4]. Hence, to overcome the drawbacks of PQM, the Internet Engineering Task Force (IETF) proposed Active Queue Management (AQM) [5]. AQM is now extensively deployed. AQM proactively monitors the queue and can drop packets before the queue is full. AQM can also make use of Explicit Congestion Notification (ECN) to mark packets in order to inform end-users about congestion [5], [8].

The goal of all AQM designs is to keep the average queue size in the routers small. AQM designs function by detecting impending queue build-up and notifying sources before the queues overflow. The proposed AQM algorithms differ in the mechanisms used to detect congestion and in the control mechanisms used to achieve a stable queue size [5].

2.2 AQM Algorithms

Since many different AQM algorithms have been proposed and these only differ in minor details, this overview is limited to the most important algorithms and algorithm classes. It is undesirable to have large queues that are full most of the time since this will significantly increase delays. Keeping in mind the ever-increasing speed of networks, it is important to have a mechanism that keeps the throughput high while keeping the average queue size as low as possible. There have been many AQM algorithms developed in the past, two of which are considered in this study [4]. The goals of AQM are to maintain a stable queue, achieve high resource utilization, and lower the queue delay. In order to fulfill these needs, a wide range of AQM algorithms have been proposed, including CoDel, Drop-tail, DRR, and Random Early Detection (RED). RED is also known as Random Early Drop or Random Early Discard. In RED, packets are dropped on the basis of the average queue size. If the buffer is empty it accepts all incoming packets. As the number of packets in the buffer increases, the probability of dropping packets increases. When the buffer is full this probability is 1 [4].

2.3 CoDel

CoDel is a recently proposed AQM mechanism that aims to provide a better solution than existing techniques [3]. Unlike other AQM mechanisms, CoDel is independent of network parameters such as queue size, queue size averages, queue size thresholds, rate measurements, link utilization, drop rate, queue occupancy time, and round-trip delay [3].

When a new packet arrives at the router, it is enqueued and marked with its enqueue time. When a packet leaves the queue it is dequeued. CoDel calculates the sojourn time as the current time minus the arrival time of the packet. If the sojourn time of a dequeued packet is higher than the target value, the queue will drop the packet. If the sojourn time is below the target value, then the packet is simply forwarded [3].

The target value is the acceptable queue delay for the packets. A target value of 5 ms has shown to provide consistently high link utilization across a range of bandwidths [3]. If it is more than 5 ms there is minor or no improvement in utilization. The Round Trip Time (RTT) is the length of time it takes for a packet to be sent plus the length of time it takes for an acknowledgment of that packet to be received. The interval time is the acknowledgment time of RTT. An interval time of 100 ms has shown to avoid longer response times [3].

2.3.1 CoDel Mechanism

The CoDel mechanism works in two phases. In the first phase, a packet is enqueued and in the second phase, a packet is dequeued. When a packet arrives at a router, it is enqueued if there is available space in the queue. If the queue is full, the incoming packet is dropped [12]. While enqueuing the incoming packet, a timestamp is added to the packet header, which becomes the enqueue time. When the packet reaches the end of the queue, it is dequeued and the packet sojourn time is calculated. CoDel is either in a dropping state or a non-dropping state. If the packet sojourn time remains above the target value for a specified interval of time, CoDel enters the dropping state and starts dropping packets. While the algorithm is in dropping state, if the packet sojourn time becomes less than the target or if the queue does not have sufficient packets to fill the outgoing link, the algorithm leaves the dropping state. Unlike other AQM algorithms which drop packets when entering the queue, CoDel drops packets when leaving the queue [3].

There are two important CoDel parameters to be set, the target value and the interval time. These parameters are fixed and their values were chosen based on observations from several experiments [12]:

- Target value = 5 ms [3].
- Interval time = 100 ms [3].

Figure 2.1 illustrates the CoDel algorithm for a packet. Initially, when a packet arrives, the queue size is checked. If the queue is not full, the packet is enqueued. When the sojourn time is more than the target value CoDel enters the dropping state and drops packets.

The count indicates the total number of packets dropped since the dropping state was entered. Initially, the count is 0 and each time a packet is dropped the count is increased by 1 [11].

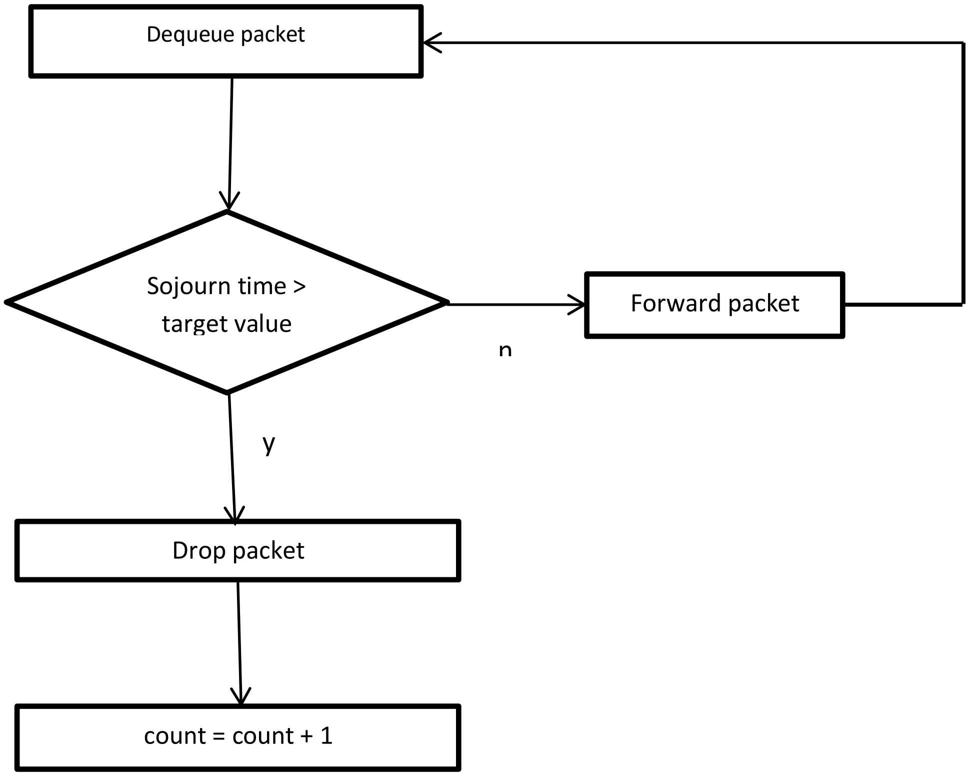

Fig. 2.1. The CoDel mechanism flowchart [11].

2.4 Deficit Round Robin

Deficit Round Robin (DRR) is a network scheduling technique based on a modified form of the round robin algorithm [6]. DRR allows fair scheduling of multiple flows over a single shared link. It is designed to deal with flows having unequal packet sizes. A flow is defined as packets traveling the same path from a source to a destination. Packets coming from different flows are stored in different queues [6].

DRR uses round-robin servicing with a quantum of service assigned to each queue. Quantum is the number of bytes that each queue can transmit in a turn. The only significant difference between DRR and traditional round-robin is that if a queue was not able to send a packet in the previous round because the packet size was too large, the remainder from the previous quantum is added for the next round [6].

2.4.1 Deficit Round Robin (DRR) Mechanism

The DRR scheduling mechanism manages traffic flow in a round robin manner. For each queue there is a deficit counter that keeps track of the amount of service granted to a flow. Before the DRR algorithm starts, all deficit counters are initialized to zero. In each round a flow is allocated a quantum value that is stored in a deficit counter when serving the flow [6]. Any remaining bytes of quantum are accumulated in a deficit counter for the next round. The active flow is the flow being served [6]. The deficit counter of a flow is equal to the sum of the quantum value and the remaining bytes from the previous round. The DRR serves the next flow if the current queue is empty or if there is insufficient value remaining in the counter to serve the next packet of the active flow. The remaining bytes are gfsaved in the deficit counter for the next round. If the queue of the active flow is empty, the deficit counter is set to zero and the system immediately switches to the next flow [13].

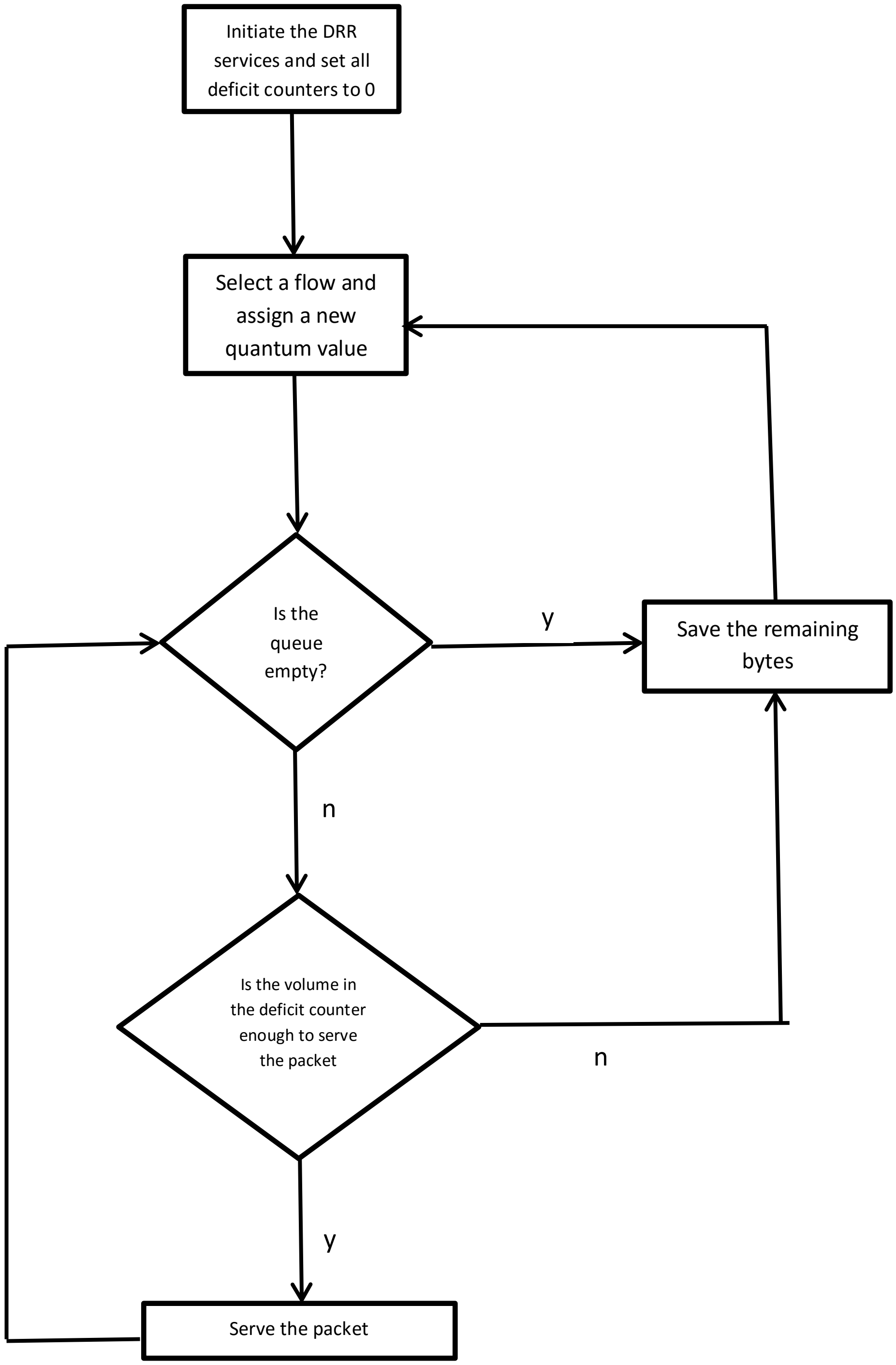

Fig. 2.2. The DRR mechanism flowchart [13].

Chapter 3 Simulation and Results

3.1 Performance Evaluation

To evaluate the performance of real-time traffic using the DRR and CoDel algorithms, simulations were conducted using Network Simulator-2 (NS-2).

3.1.1 Simulation Setup

A real-time network was implemented in NS-2 using a single bottleneck dumbbell topology. The bottleneck bandwidth link varies from 1 Mbps to 10 Mbps. The CoDel and DRR algorithms were implemented in the network. There are six nodes in the network comprised of two senders (S1 and S2), two receivers (R1 and R2), and two routers (A and B). The routers are connected by the bottleneck link. In the first scenario the bandwidth of the link between sender S1 and router A is 1 Mbps, and between sender S2 and router A is 10 Mbps. In the second scenario the bandwidth of the link between S1 and router A is 5 Mbps and between S2 and router A is 10 Mbps. The links between router B and the receiving nodes R1 and R2 has sufficient bandwidth for all incoming packets. When a simulation is run in NS-2, a trace file is generated to get information for the different bottleneck bandwidths. The trace file provides overall information about the network. These files are used to gather network information using a tool called a trace graph. A trace graph is a free network trace file analyzer developed for NS-2 trace processing [8]. A trace string for each packet is located in the trace file. There are 12 fields in the trace string of which 7 are used by the trace graph as shown below.

Time	Source node	Destination node	Packet name	Packet size	Source address	Destination address

Time indicates when the packet trace string was created. Source node and destination node have the source and destination IDs. The packet name is the type of packet and the packet size is expressed in bytes.

The performance of CoDel and DRR is evaluated by running the simulation with different bottleneck link capacities. Once all the required trace files are generated, a trace graph is used to analyze the results. The following metrics are evaluated for each of the algorithms

- Bottleneck link utilization
- Packet queue delay
- Packet drop rate
- Throughput

3.1.2 Dumbbell Topology

The single bottleneck dumbbell topology shown in Fig. 3.1 is used in virtually all AQM papers and so is used here. RTP packets flow from router A to router B through the bottleneck link. In this figure, S1 and S2 send RTP traffic, A and B are the edge routers, and R1 and R2 are the receiving nodes. S1 sends packets to R1 and S2 sends packets to R2.

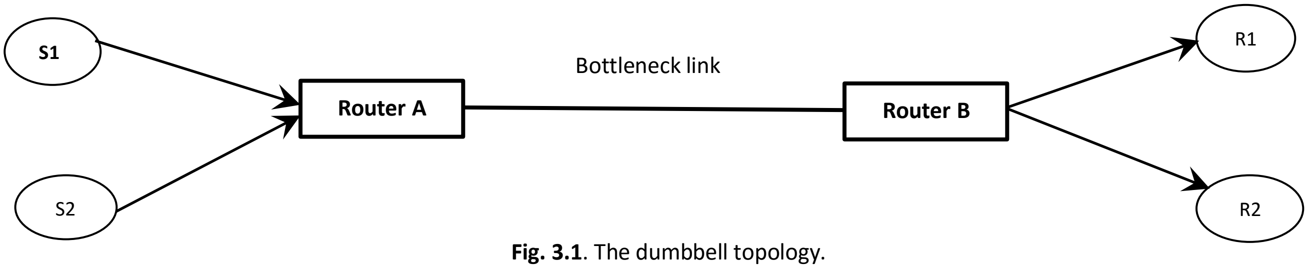

Fig. 3.1. The dumbbell topology.

3.2 Results

Fig. 3.2 shows the bottleneck link utilization for a bandwidth of 5 Mbps between S1 and router A, while Fig. 3.3 shows the bottleneck link utilization for a bandwidth of 1 Mbps between S1 and router A. It can be seen in the figures that the link utilization decreases as the bottleneck bandwidth increases. Both algorithms perform similarly across the range of bandwidths.

In Fig. 3.4, the bandwidth between S1 and router A is 5 mbps. The CoDel packet delay is compared with that of DRR for bottleneck bandwidths from 1 to 10 Mbps. For all bandwidths, CoDel has less packet delay. The increase in the bottleneck bandwidth enables the router to empty its queue more quickly by forwarding packets through the link. Hence, the delay is expected to diminish at higher bottleneck bandwidths. Table 3.1 gives the total number of packets sent from nodes S1 and S2 for each of the schemes.

In Fig. 3.5, the bandwidth between S1 and router A is changed from 5 Mbps to 1 Mbps. There is no significant change in average delay by changing the bandwidth. CoDel still has comparatively less packet queue delay. The increase in bottleneck bandwidth enables the router to empty its queue more quickly by forwarding packets through the bottleneck link. Table 3.2 gives the total number of packets sent from nodes S1 and S2 for each of the schemes. From Tables 3.1 and 3.2, it can be seen that if the bandwidth link is low then CoDel sends fewer packets as compared to DRR.

In Fig. 3.6, the link between S1 and router A has a bandwidth of 5 Mbps. The CoDel packet drop rate is compared with DRR for varying bottleneck bandwidths from 1 Mbps to 10 Mbps. When the bottleneck bandwidth is low, the packet drop rate is higher. As the bandwidth increases, the packet drop rate is reduced. The packet drop rate behavior, however, is similar for both mechanisms. The packet drop rate approaches zero when the bottleneck bandwidth is greater than 3 Mbps.

Fig. 3.7 illustrates the packet drop rate for CoDel and DRR with a link bandwidth between S1 and router A of 1 Mbps. When the bottleneck bandwidth is less than 4 Mbps there are fewer packets dropped as compared to the results in Fig. 3.6 when the bandwidth between S1 and router A has a bandwidth of 5 Mbps. The packet drop rate behavior, however, is similar for both mechanisms.

Fig. 3.8 shows the throughput of CoDel and DRR as the bottleneck bandwidth varies. As this bandwidth increases, the network throughput improves. The results in Tables 3.3 and 3.4 were used to produce Fig. 3.8. These tables show statistics on the received packets with CoDel and DRR when the link between S1 and router A has a bandwidth of 5 Mbps. The throughput in Mbps is given by

$$\boldsymbol{Throughput = (bits\ per\ sec)/1000000}$$

Fig. 3.9 shows the throughput of CoDel and DRR varies as the bottleneck bandwidth varies. Fig. 3.9 is based on the data in Table 3.5 and 3.6. As the bottleneck bandwidth increases, the number of received packets also increases along with an increase in the average packet size. With an increase in the bottleneck bandwidth, more packets are received with CoDel compared to DRR.

Tables 3.5 and 3.6 show the statistics on the received packets when the link between S1 and router A has bandwidth of 1 Mbps. There is not much difference in the total number of packets received as compared to Tables 3.3 and 3.4.

Tables 3.7 and 3.8 show statistics on the received packets for each receiving node, the data in the table is used to calculate throughput. Tables 3.7 and 3.8 were generated when the link between S1 and router A is 5 Mbps.

Fig. 3.10 and 3.11 show throughput for each node separately. Higher bandwidth gives better throughput for both schemes. Throughput values in Figs. 3.10 and 3.11 are calculated using Tables 3.7 and 3.8. Node R1 received packets from S1 and node R2 received packets from S2. In this scenario, the link between S1 and router A is 5 Mbps.

Fig. 3.12 and 3.13 show throughput for each node separately. Node R1 receive packets from S1 and node R2 receives packets from S2. In this scenario, the link between S1 and router A is 1 Mbps. These results show how the throughput decreases with lower network link. Throughput values in these figures are calculated using Tables 3.9 and 3.10.

Tables 3.9 and 3.10 show statistics on the received packets for each receiving node and, the data in the tables is used to calculate throughput. Tables 3.9 and 3.10 were generated when the link between S1 and router A has bandwidth of 1 Mbps.

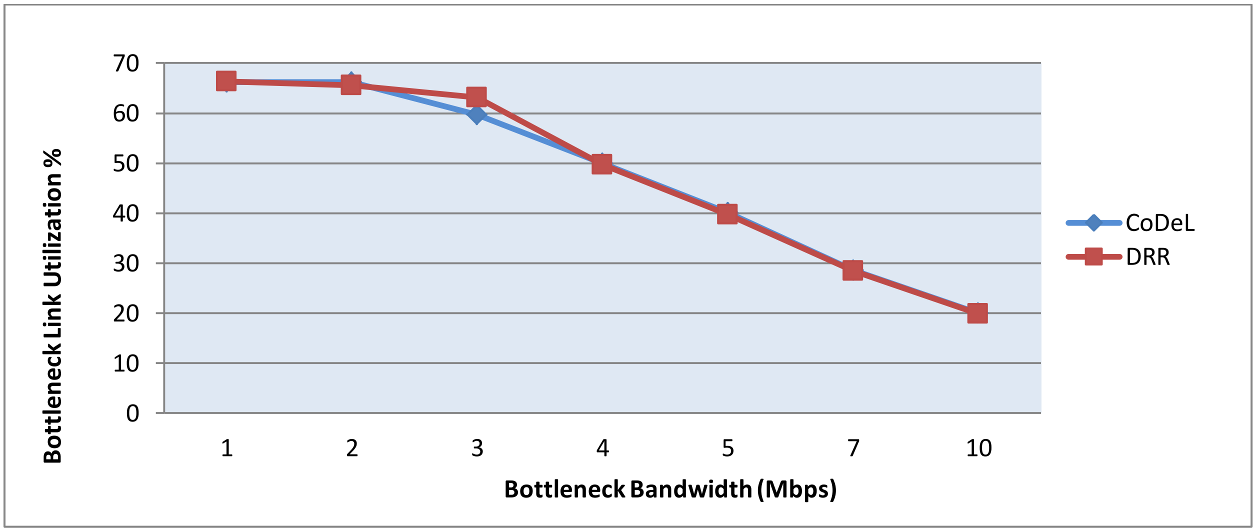

Fig. 3.2. The bottleneck link utilization with CoDel and DRR for varying bottleneck bandwidths.

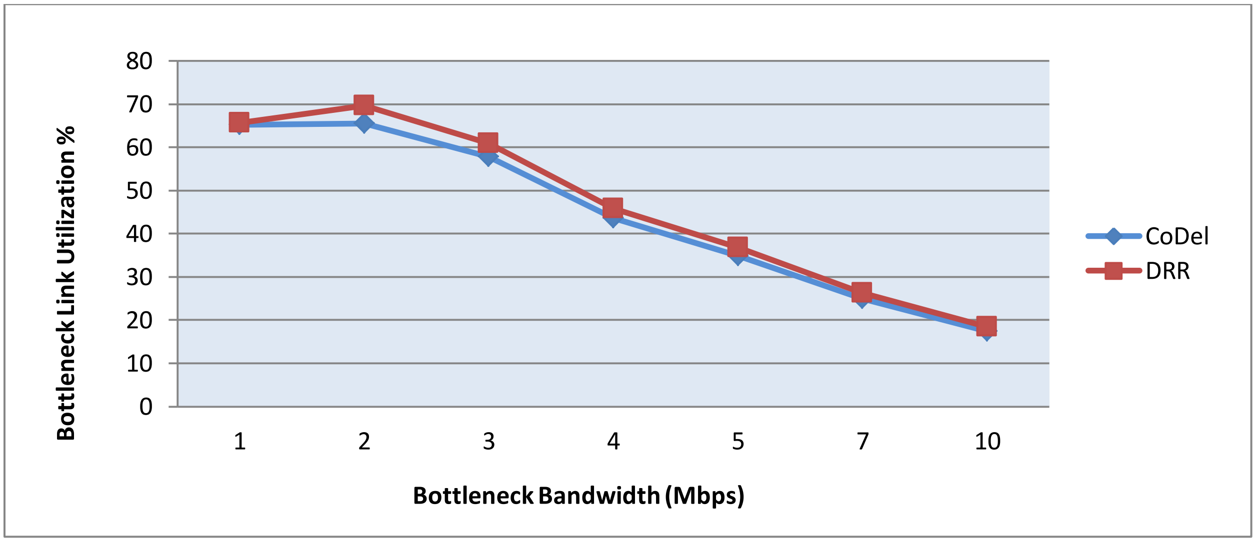

Fig. 3.3. The bottleneck link utilization with CoDel and DRR for varying bottleneck bandwidths.

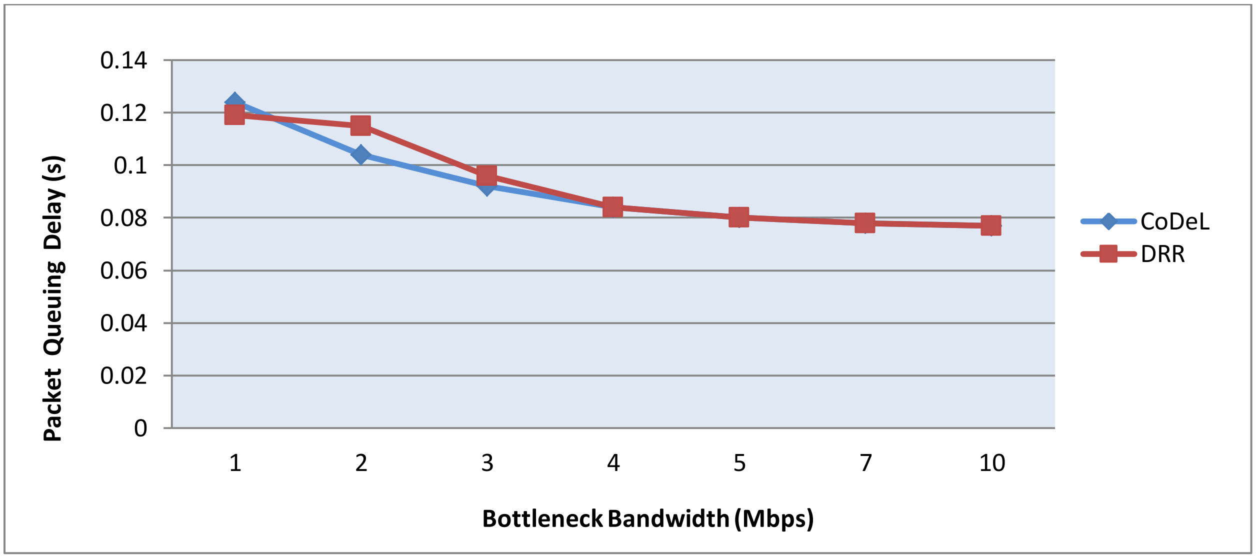

Fig. 3.4. The packet queuing delay with CoDel and DRR for varying bottleneck bandwidths.

	Node S1 sent packets	**Node S2 sent packets**	**Total**
CoDel	2015	2890	4905
DRR	1986	2890	4876

Table 3.1 The total number of packets sent by each sender.

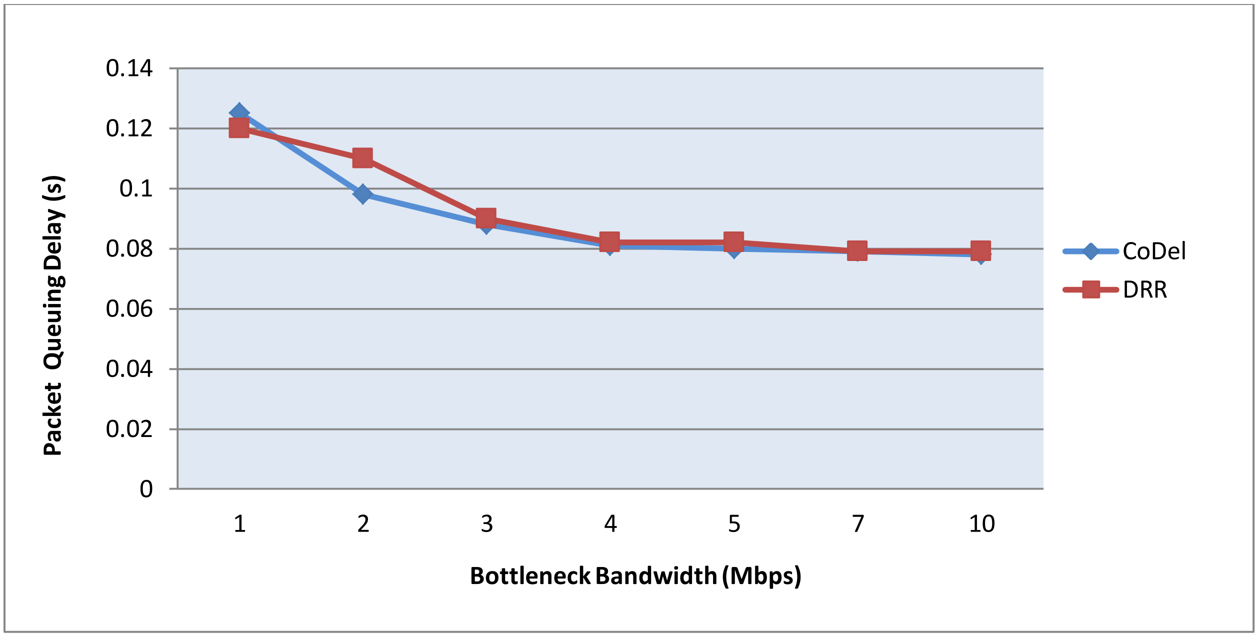

Fig. 3.5. The packet queuing delay with CoDel and DRR for varying bottleneck bandwidths.

	Node S1 sent packets	**Node S2 sent packets**	**Total**
CoDel	1493	2890	4383
DRR	1676	2890	4566

Table 3.2 The total number of packets sent by each sender.

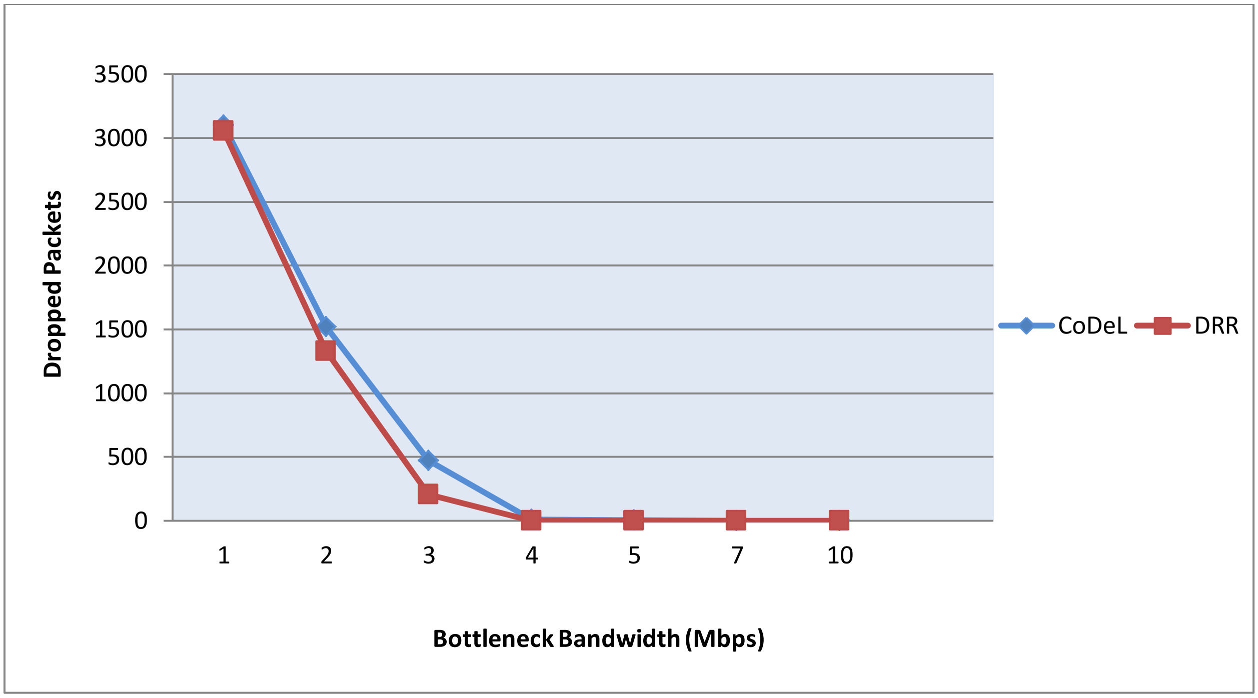

Fig. 3.6. The number of packets dropped with CoDel and DRR for varying bottleneck bandwidths.

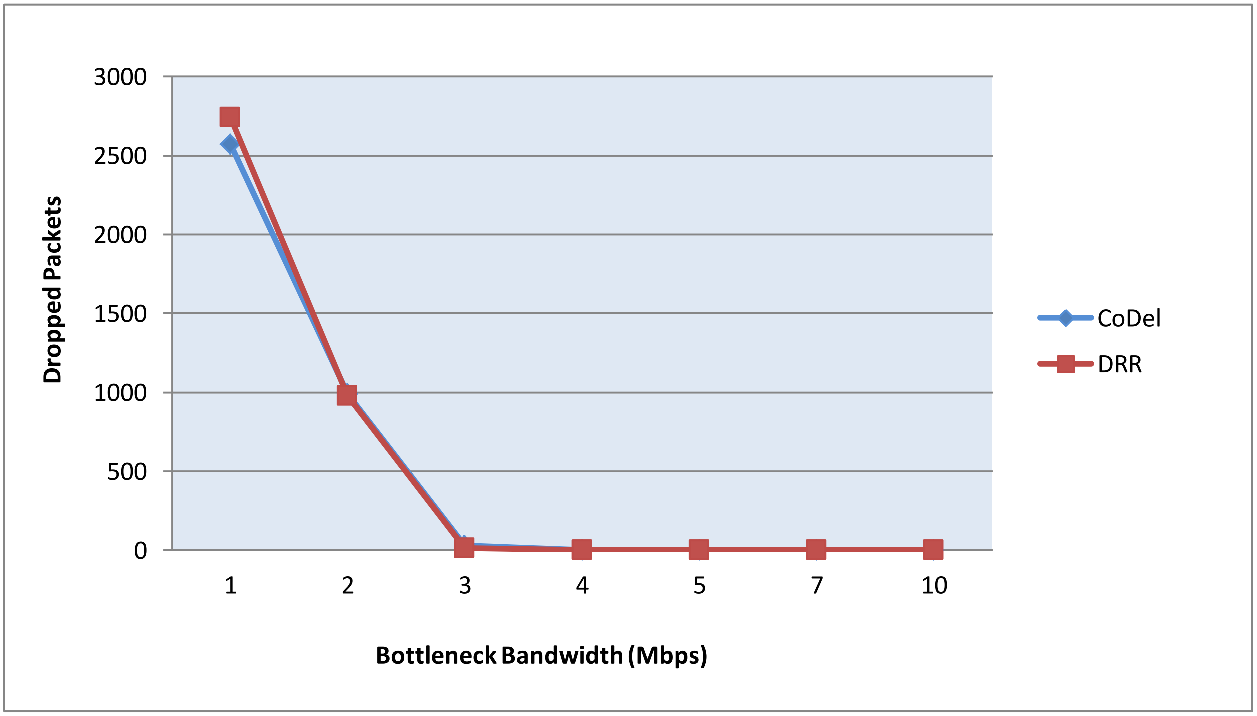

Fig. 3.7. The number of packets dropped with CoDel and DRR for varying bottleneck bandwidths.

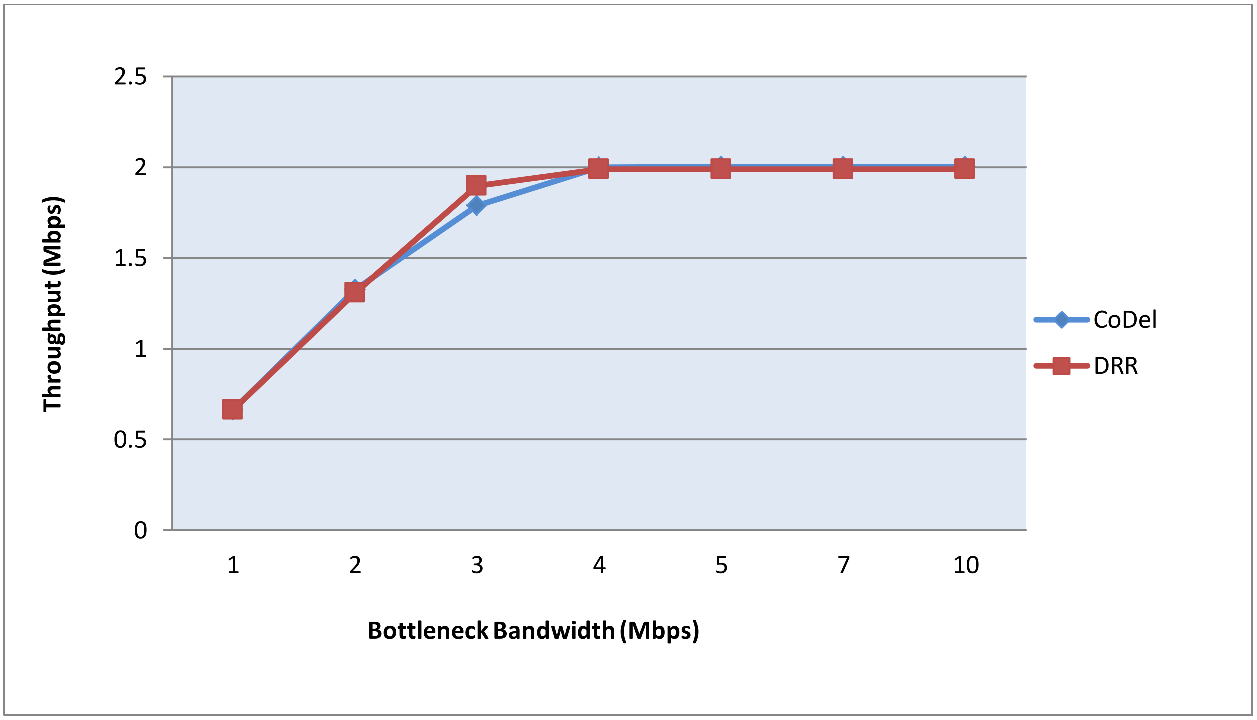

Fig. 3.8. Throughput for the dumbbell topology

Bottleneck bandwidth	Total packets received	Average packet size	Packets/sec
1 Mbps	1766	469	176.6
2 Mbps	3347	495	334.7
3 Mbps	4395	509	439.5
4 Mbps	4854	515	485.4
5 Mbps	4863	515	486.3
7 Mbps	4865	515	486.5
10 Mbps	4866	515	486.6

Table 3.3 Statistics on the received packets with CoDel.

Bottleneck bandwidth	Total packets received	Average packet size	Packets/sec
1 Mbps	1771	468	177.1
2 Mbps	3492	469	349.2
3 Mbps	4628	512	462.8
4 Mbps	4835	514	483.5
5 Mbps	4836	514	483.6
7 Mbps	4836	514	483.6
10 Mbps	4837	514	483.7

Table 3.4 Statistics on the received packets with DRR.

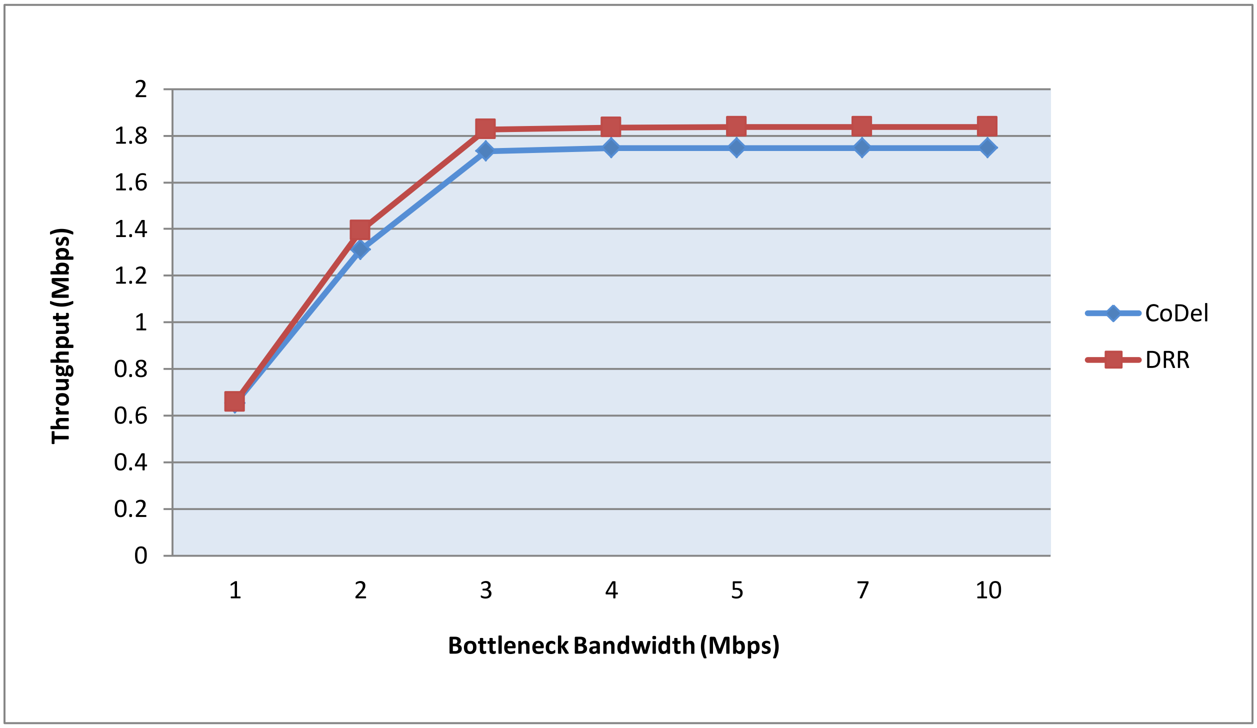

Fig. 3.9. Throughput for the dumbbell topology

Bottleneck bandwidth	Total packets received	Average packet size	Packets/sec
1 Mbps	1770	461	177.0
2 Mbps	3352	489	335.2
3 Mbps	4307	503	430.7
4 Mbps	4340	503	434.0
5 Mbps	4340	503	434.0
7 Mbps	4340	503	434.0
10 Mbps	4340	503	434.0

Table 3.5 Statistics on the received packets with CoDel.

Bottleneck bandwidth	Total packets received	Average packet size	Packets/sec
1 Mbps	1771	464	177.1
2 Mbps	3526	494	349.2
3 Mbps	4505	507	462.8
4 Mbps	4521	507	483.5
5 Mbps	4521	508	483.6
7 Mbps	4521	508	483.6
10 Mbps	4521	508	483.7

Table 3.6 Statistics on the received packets with DRR.

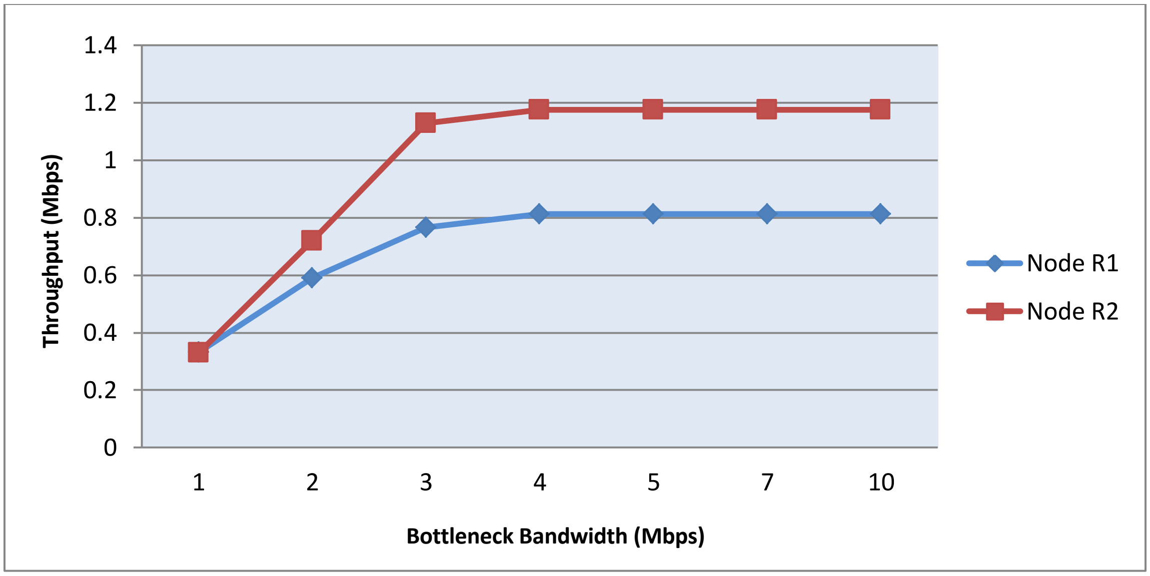

Fig. 3.10. Throughput for each receiving node with DRR.

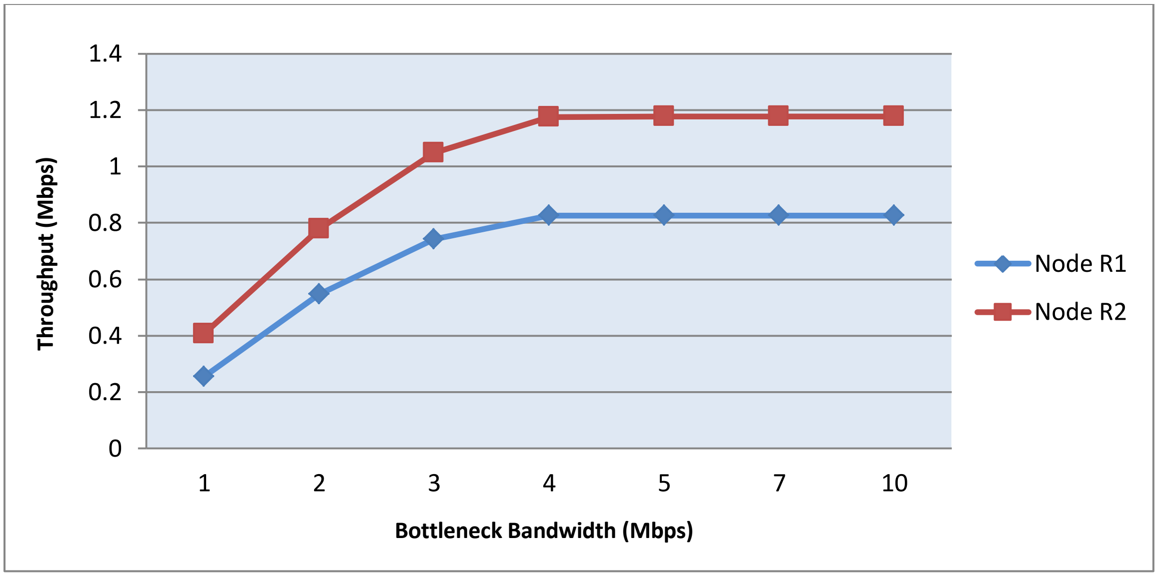

Fig. 3.11. Throughput for each receiving node with CoDel.

Bottleneck bandwidth	Packets received node R1	Packets received node R2
1 Mbps	682	1084
2 Mbps	1381	1966
3 Mbps	1820	2575
4 Mbps	2003	2851
5 Mbps	2005	2858
7 Mbps	2005	2860
10 Mbps	2006	2860

Table 3.7 Statistics on the received packets for each receiving node with CoDel.

Bottleneck bandwidth	Packets received at node R1	Packets received at node R2
1 Mbps	888	883
2 Mbps	1573	1919
3 Mbps	1873	2755
4 Mbps	1976	2859
5 Mbps	1976	2860
7 Mbps	1976	2860
10 Mbps	1977	2860

Table 3.8 Statistics on the received packets for each receiving node with DRR.

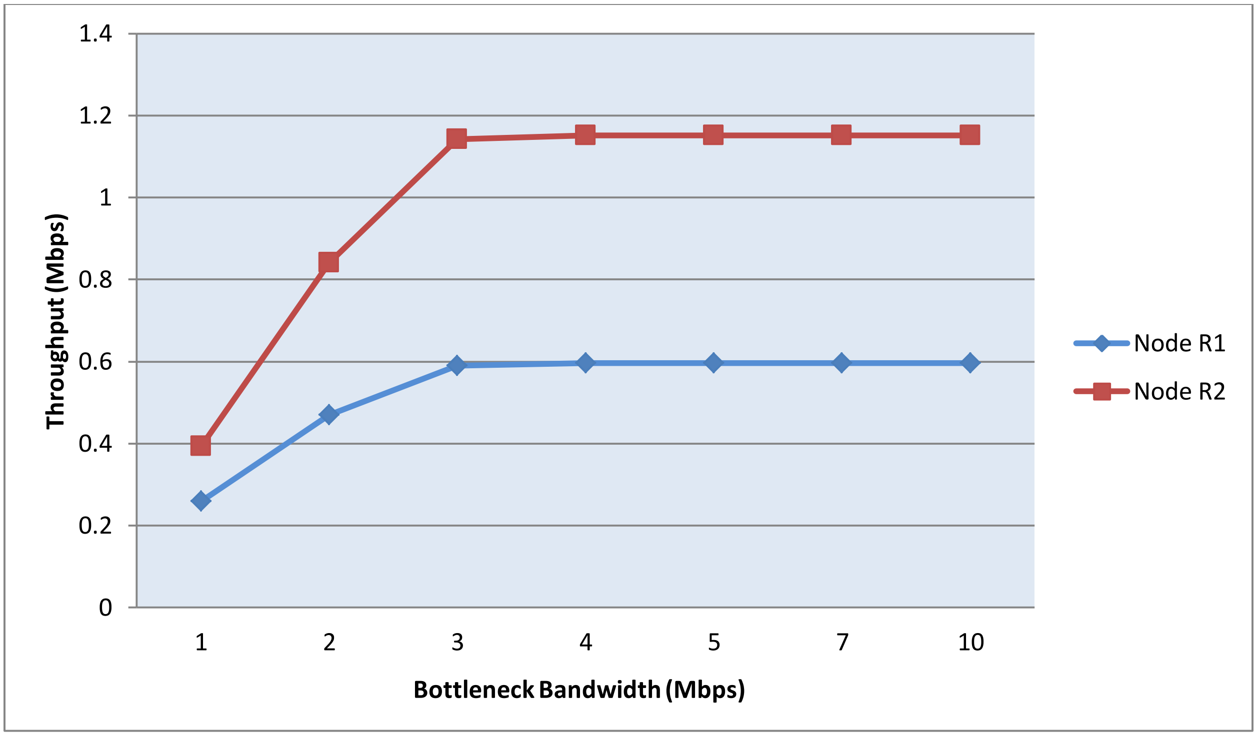

Fig. 3.12. Throughput for each receiving node with CoDel.

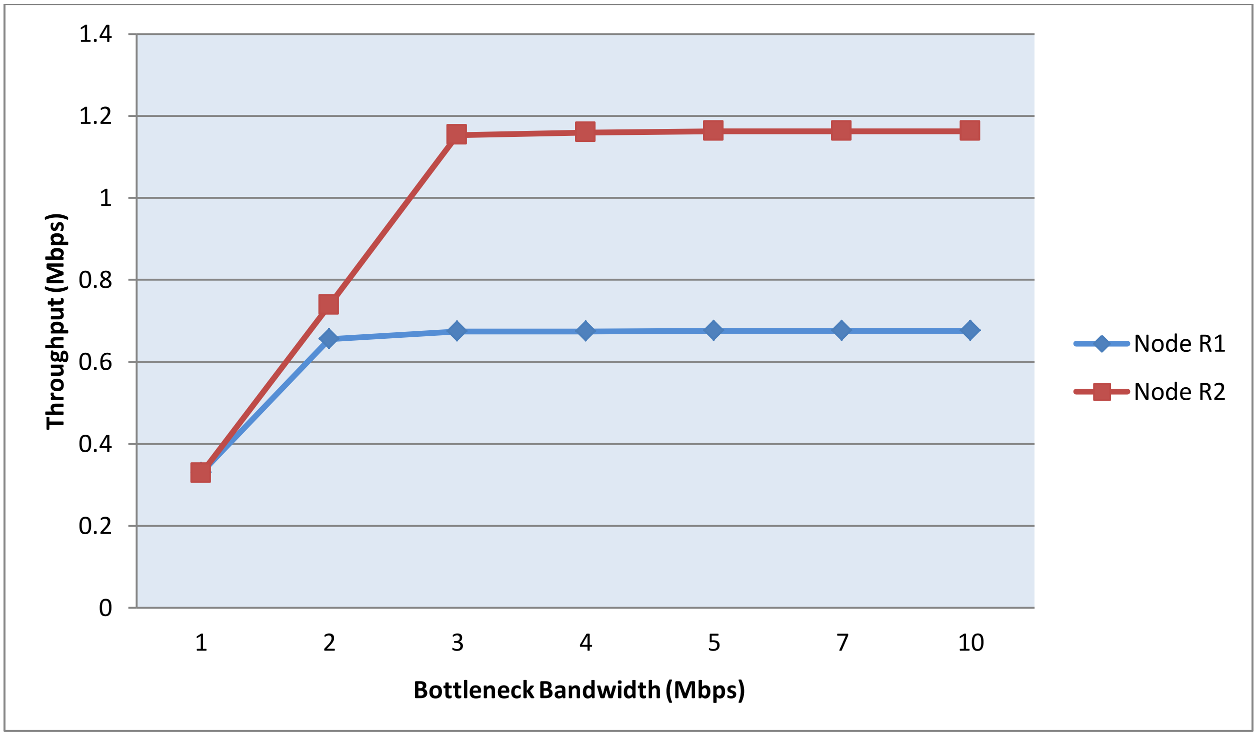

Fig. 3.13. Throughput for each receiving node with DRR.

Bottleneck bandwidth	Packets received node R1	Packets received node R2
1 Mbps	702	1068
2 Mbps	1203	2149
3 Mbps	1468	2839
4 Mbps	1480	2860
5 Mbps	1480	2860
7 Mbps	1480	2860
10 Mbps	1480	2860

Table 3.9 Statistics on the received packets for each receiving with CoDel.

Bottleneck bandwidth	Packets received at node R1	Packets received at node R2
1 Mbps	887	884
2 Mbps	1660	1919
3 Mbps	1661	2844
4 Mbps	1661	2860
5 Mbps	1661	2860
7 Mbps	1661	2860
10 Mbps	1661	2860

Table 3.10 Statistics on the received packets for each receiving with DRR.

3.3 Discussion

Based on the simulation results, in terms of link utilization, CoDel and DRR performed comparably for a 5 Mbps link between S1 and router A, but DRR performs better with a 1 Mbps link bandwidth. CoDel and DRR achieved similar results in terms of packet queue delay in both scenarios. The number of packets dropped for CoDel is more than DRR when the link between S1 and router A has a bandwidth of 5 Mbps. When the link between S1 and A has bandwidth of 1 Mbps CoDel and DRR provide have similar number of packets dropped. For the throughput, DRR achieved better results than CoDel when the link between S1 and router A has bandwidth of 1 Mbps. Therefore it is clear that when the bandwidth link increases, it results in better throughput in the network. Overall CoDel and DRR performed similar in different network scenarios.

Chapter 4 Conclusion and Future Work

Internet traffic is increasing daily, resulting in more traffic congestion. With a rapid increase in the diversity of applications, the inherent problems of Passive Queue Management (PQM) have become increasingly apparent. As a result, there has been an active interest in deploying efficient Active Queue Management (AQM) mechanisms in networks.

In this project, the effectiveness of two different AQM schemes, namely Controlled Delay (CoDel) and Deficit Round Robin (DRR), was evaluated. The performance of CoDel was compared with DRR in different network scenarios with various bottleneck bandwidths and link bandwidths between sender S1 and router A.

The performance parameters examined were bottleneck link utilization, packet queue delay, network throughput and packet drop rate. In the future, CoDel can be optimized to increase its robustness. Moreover, an in-depth investigation is also required to determine the effectiveness of CoDel in wireless scenarios.

Bibliography

[1] J. Gettys, “Bufferbloat: Dark Buffers in the Internet,” IEEE Internet Computing, vol. 15, no. 3, pp. 95-96, Jun. 2011.

[2] D. M. Raghuvanshi, B. Annappa and M. P. Tahiliani, “On the Effectiveness of CoDel for Active Queue Management,” in International Conference Advanced Computing and Communication Technologies, pp. 107-114, Apr. 2013.

[3] K. Nichols and V. Jacobson, “Controlling Queue Delay,” Communications of the ACM, vol. 55, no. 7, pp. 42–50, Jul. 2012.

[4] S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance”, IEEE/ACM Transactions on Networking, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[5] B. Braden, “Recommendations on Queue Management and Congestion Avoidance in the Internet,” RFC 2309, Apr. 1998.

[6] M. Shreedhar and G. Varghese, “Efficient Fair Queuing Using Deficit Round Robin,” IEEE/ACM Transactions on Networking, vol. 4, no. 3, pp. 375-385, Jun. 1996.

[7] W. Stevens, M. Allman, and V. Paxson, “TCP Congestion Control,” RFC 2581, Apr. 1999.

[8] http://ns-2.blogspot.ca/2008. Mar. 2016 [Online]

[9] Cisco Visual Networking Index: Forecast and Methodology, Jun. 2016. [online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html.

[10] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin, “REM: Active Queue Management,” IEEE Network, vol. 15, no. 3, pp. 48-53, May 2001.

[11] T. Sharma, “Controlling Queue Delay (CoDel) to Counter the Bufferbloat Problem in Internet,” International Journal of Current Engineering and Technology, vol. 4, no. 3, pp. 2210-2215, Jun. 2014.

[12] S. Matilda, B. Palaniappan, and P. Thambidurai, “Bufferfloat Mitigation for Real-time Video Streaming using Adaptive Controlled Delay Mechanism,” International Journal of Computer Applications, vol. 63, no. 20, pp. 1-6, Feb. 2013.

[13] L. Liu, X. Jin, G. Min, and K. Li, “An Analytical Model of Deficit Round Robin Scheduling Mechanism under Self-Similar Traffic,” Concurrency and Computation: Practice and Experience, vol. 22, no. 13, pp. 1911-1926, Oct. 2010.

[14] Bufferbloat Project, Jan. 2016. [Online].

Available: http://www.bufferbloat.net

[15] The Network Simulator - ns-2 Project, Dec. 2015. [Online].

Available: http://www.isi.edu/nsnam/ns/

[16] H. Schulzrine, S. Casner, R. Fredrick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” RFC 3550, Jul. 2003.

[17] C. Bouras, A. Gkamas, and G. Kioumourtzis, “Extending the Functionality of RTP/RTCP Implementation in Network Simulator (NS-2) to Support TCP-Friendly Congestion Control,” in International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, Mar. 2008

[18] L. Zhang, “Virtual Clock: A New Traffic Control Algorithm for Packet Switched Networks,” Proceedings of the ACM Symposium on Communications Architecture & Protocols, vol. 9, pp. 101-125, May 1991.