Chapter Four

A Dynamic RED Algorithm Based on an Adaptive Maximum Threshold

4.1 Introduction

In Subsection 2.4.3.2, the RED algorithm drawbacks were discussed, which included the RED dependency on setting its parameters (queue weight (qw), maximum dropping probability value ($D_{\max}$), $\min threshold$ and $\max threshold$) to optimal values in order to achieve good network performance. To obtain performance measure results better than that of RED or at least similar to it, a new AQM method called Dynamic RED (REDD) is proposed. The new algorithm utilises a dynamic $\max threshold$ position, and aims to: 1) achieve more satisfactory mean queue length (mql) and average queueing delay (D) results than the RED and ARED algorithms, especially when congestion is present, 2) lose fewer packets due to router buffer overflows than RED and ARED when congestion is present, 3) sustain the aql between the $\min threshold$ and the $\max threshold$ at a

certain level represented by T_{aql}, and 4) Decrease a large dependency of RED on its parameters through using an adaptive $\max threshold$ parameter rather than a constant one.

This chapter is outlined as follows: Section 4.2 presents the proposed Dynamic RED (REDD) algorithm. The tuning of the REDD's parameters is introduced in Section 4.3. A comparison of the proposed REDD algorithm with both RED and ARED algorithms with regard to the performance measures is revealed in Section 4.4. Finally, this chapter is summarised in Section 4.5.

4.2 The Proposed Dynamic RED

In this section, the new method, REDD, which is an extension of RED [46] is introduced. REDD employs a dynamic $\max threshold$ position to manage the congested router buffer as early as possible and before the router buffer overflows. It is shown experimentally in Section 4.4 that REDD offers performance measure results at least equal to the RED and ARED algorithms. Moreover, the proposed algorithm maintains mql and D slightly better than RED and ARED when a congestion situation is generated. Moreover, REDD loses packets marginally fewer than either RED or ARED when congestion occurs. The REDD algorithm also aims to stabilise the aql at T_{aql} between the $\min threshold$ and the $\max threshold$ by updating the $\max threshold$ parameter. This certain level is half way from the $\min threshold$ position to the $\max threshold$ position at each router buffer. The proposed method uses the RED algorithm policy in marking/dropping packets, where it

probabilistically drops packets when the *aql* is between the $\min threshold$ and the $\max threshold$. In conformance with the recommendation for RED, ARED and GRED (see Subsections 2.4.3, 2.4.4 and 2.4.5) the initial $\max threshold$ is set to $3 \times \min threshold$ and T_{aql} is the midpoint of the interval between $\min threshold$ and $\max threshold$ (see Subsection 4.3.1). This is likely a consequence of the form of the initial values of $\min threshold$ and $\max threshold$. The change in $\max threshold$ should be related to range $\min threshold$ to $\max threshold$, something like $(\max threshold - \min threshold) \times \frac{1}{\min threshold}$. But this always gives 2 approximatly. So 2 is used to increment and decrement $\max threshold$ in Figure 4.1 (the REDD algorithm).

In Figure 4.1, the equation of $(\max threshold - \min threshold) \times \frac{1}{\min threshold}$ is performed only before the $\max threshold$ value gets updated, and its result equals 2. The 2 result from the above equation was obtained by setting the $\max threshold$ to $3 \times \min threshold$ (see Subsection 4.3.2 for more details). Further, the *BufferCapacity* parameter represents the capacity of each router buffer. The rest of the parameters in Figure 4.1 are similar to those of RED's, which were discussed in Subsection 2.4.3.1. In REDD, the $\max threshold$ value stabilises between $2 \times \min threshold$ and $(BufferCapacity - \min threshold)$ to present the network performance from degrading and to drop fewer packets. In Figure 4.1, "2" denotes the number of packets that are added to or removed from the $\max threshold$ in order to sustain the *aql* at T_{aql}. In REDD, if the *aql*

is less than the $\min threshold$ position, no packets are dropped since no congestion is present at the router buffer. Whereas, if the aql is between the $\min threshold$ and the $\max threshold$ positions, the REDD router buffer probabilistically drops packets in a similar way to that of RED. However, REDD examines the position of the aql in which if the aql is between the $\min threshold$ and T_{aql}, the REDD router buffer decreases its $\max threshold$ position by two to move the aql towards T_{aql} aiming to stabilise it.

For every packet arriving at the router buffer

$if\left(aql < T_{aql} \;\&\&\; \max threshold \geq 2 \times \min threshold\right)$

{

//Decreasing the $\max threshold$ by 2 as follows:

$\max threshold = \max threshold - 2;$

}

$if\left(aql > T_{aql} \;\&\&\; \max threshold \leq \left(BufferCapacity - \min threshold\right)\right)$

{

//Increasing the $\max threshold$ by 2 as follows:

$\max threshold = \max threshold + 2;$

}

Figure 4.1: The pseudocode for the REDD algorithm.

On the other hand, if the aql is between T_{aql} and the $\max threshold$ positions, the REDD router buffer increases it's $\max threshold$ position by two aiming to move the aql

towards T_{aql} in order to serve more packets and to alleviate the aql value. Finally, if the aql reaches the $\max threshold$ position, the REDD router buffer processes the arriving packets similar to RED by dropping every arriving packet to manage the congestion.

4.3 Tuning the Proposed REDD Parameters

In this section, the tuning of the REDD parameters is explained.

4.3.1 Setting $D_{\max}$, qw and T_{aql}

The $D_{\max}$ and qw parameters have been set to values similar to those of the RED algorithm [46], in which $D_{\max}$ is set to 0.1, and qw is set to 0.002. T_{aql} is calculated according to equation (4.1)

$$\frac{\left(\min threshold + \max threshold\right)}{2} \quad \text{..... (4.1)}$$

4.3.2 Setting $\min threshold$ and $\max threshold$

The $\min threshold$ and the $\max threshold$ have been set to the same values as those in the RED and ARED algorithms. The $\max threshold$ relies on the computed aql value, where if the aql value is less than the $\min threshold$ position, the $\max threshold$ is set according to equation (4.2).

$(3 \times \min threshold)$ [49]…… (4.2)

However, if the aql value is between the $\min threshold$ and T_{aql}, the $\max threshold$ is set according to equation (4.3) by increasing the aql in order to stabilise it at T_{aql}.

$\max threshold - 2$ …….. (4.3)

However, if the aql becomes larger than T_{aql} and less than the $\max threshold$, the $\max threshold$ value gets increased in order to move the aql towards T_{aql} and to alleviate the congestion at the router buffer by serving more packets. In this case, the $\max threshold$ is set using equation (4.4).

$\max threshold + 2$ ……. (4.4)

Finally, if the aql value is equal to or larger than $(BufferCapacity - \min threshold)$ position in the router buffer, then the $\max threshold$ is set to value of equation (4.5).

$(BufferCapacity - \min threshold)$ ……. (4.5)

From equation (4.5), the $(BufferCapacity - \min threshold)$ value represents the highest boundary value that it can take. On the other hand, the lowest boundary value for the $\max threshold$ is given in equation (4.6).

$2 \times \min threshold$ ……..... (4.6)

The highest and lowest boundary values for the $\max threshold$ are shown in equations (4.5-4.6). It should be noted that the reason for choosing $(BufferCapacity - \min threshold)$ as a higher boundary in equation (4.5) is because the highest queueing delay for packets can be specified and to determine when every arriving packet will be marked/dropped. Moreover, for the lower boundary in equation (4.6), the $\max threshold$ is set to $2 \times \min threshold$ conforming to the recommendations in [46].

4.3.3 The Adjustment Mechanism for the $\max threshold$

This subsection presents a method for adjusting the $\max threshold$. It has been mentioned in Subsection 4.3.2 that the $\max threshold$ is updated by the results obtained from equations (4.3, 4.4). Equations (4.3, 4.4) show the amount of decrease or increase to the $\max threshold$, respectively, in order to detect the congestion at early stages. This amount can be represented by two since this amount moves the aql from below the T_{aql} position to above it, or vice versa in a slow manner. Therefore, this move can be estimated according to equations (4.3) and (4.4). Furthermore, this is likely a consequence of the form of the equations and values of $\min threshold$ and $\max threshold$ after the REDD

parameters have been set and before the $\max threshold$ starts to be modified. The $\max threshold$ value decreases or increases by two each time the aql is between $\min threshold$ and $\max threshold$.

4.4 Performance Evaluation of the REDD Algorithm

This section presents the simulation results of the REDD algorithm with respect to the performance measures ($mql, T, D, P_L, D_p, P_{Loss}$). Additionally, the proposed algorithm is compared with the RED and ARED algorithms with reference to the above performance measures in order to obtain 1) The algorithm which gives the most satisfactory performance measure results, and 2) The algorithm, which loses and drops fewer packets at its router buffers, where the loses is due to overflow the router buffer. All the performance measure results are obtained based on the values of the packet arrival probability. Therefore, the decision which algorithm is better in terms of the performance measure results is given only based on the values of the packet arrival probability parameter. The statistical analysis of the ten run results for each performance measure in the RED, ARED and proposed algorithm are introduced to determine a degree of dispersion for the ten performance measure results from their mean, and this shows how much the performance measure mean is accurate. RED, ARED and the present algorithm are implemented by simulation in a Java environment on a 1.66 Centrino with 1024 MB RAM. This section is organised as follows: Subsections 4.4.1 and 4.4.2 present the ten run mean results for (mql, T, D, P_{Loss}) and

(P_L, D_p) performance measures of the RED, ARED and REDD algorithms, respectively. The statistical analysis for the RED, ARED and REDD algorithms is highlighted in Subsection 4.4.3. Finally, a chapter summary is given in Section 4.5.

4.4.1 Performance Evaluation of RED, ARED and REDD

This subsection introduces a comparison between the RED and ARED algorithms and the proposed REDD algorithm with reference to mql, T, D and P_{Loss} to identify which algorithm offers the most satisfactory performance measure results. The parameters of the RED and ARED algorithms were set to values the same as in 3.1.2. Furthermore, the REDD's parameters (Probability of packet arrival and departure, *BufferCapacity*, which includes packets in service (System capacity), $\min threshold$, $\max threshold$, T_{aql}, Time interval, qw, $D_{\max}$, Number of slots) have been set to values similar to those of the ARED algorithm.

The performance measure results of the RED, ARED and REDD algorithms are obtained after the system has reached to a steady state. The performance measure results are given with reference to the values of probability of packet arrival, where the value of probability of packet arrival varies from 0.18 to 0.93. Every algorithm compared was run ten times with the seed changing between runs, with the objective of producing a statistical analysis for each packet arrival probability value and to examine the behaviour of the performance measure results with regard to the probability of packet arrival.

Figures [4.2-4.5] show the performance measure mean results of the ten runs for the RED, ARED and REDD algorithms versus the probability of packet arrival. Specifically Figures 4.2 and 4.3 display the mql and T mean results, respectively, and the mean results of the D and P_{Loss} are provided in Figures 4.4 and 4.5, respectively.

It is observed in Figures 4.2 and 4.4 that the proposed REDD algorithm gives marginally better mql and D results than either RED or ARED when the packet arrival probability is set to a value larger than the packet departure probability (occurrence of congestion). However, with the probability of packet arrival set to a value lower than the probability of packet departure (there is either no congestion or a light congestion), the RED, ARED and REDD algorithms offer similar mql and D results.

Clearly, Figures 4.3 and 4.5 show that the RED, ARED and REDD algorithms obtain similar T and P_{Loss} results whether the probability of packet arrival was set to a value lower or higher than the probability of packet departure value. Figure 4.3 indicates that if the packet arrival probability is equal to a value lower than the packet departure probability, the results of T for the compared algorithms increase as long as the packet arrival probability increases. On the contrary, the T results for all compared algorithms are stabilised at the value of packet departure probability when there is congestion at the router buffer of the algorithms. Figure 4.5 depicts that when there is no congestion or a light congestion at the router buffers of the RED, ARED and REDD algorithms (The probability of packet arrival = 0.18, 0.33 or 0.48), the RED, ARED and REDD give very small values of P_{Loss} (zero or almost equal to zero). On the other hand, when there is a congestion situation since the packet arrival probability > packet departure probability, all algorithms

achieve similar P_{Loss} results and these results are increased as long as the packet arrival probability increases due to the increasing in overflow losing and dropping probabilities of packets.

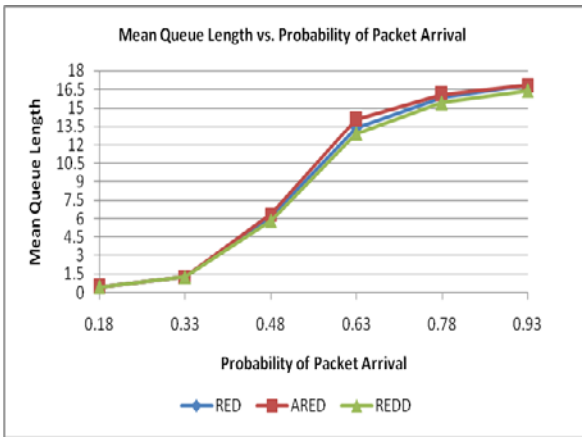

Figure 4.2: mql vs. probability of packet arrival.

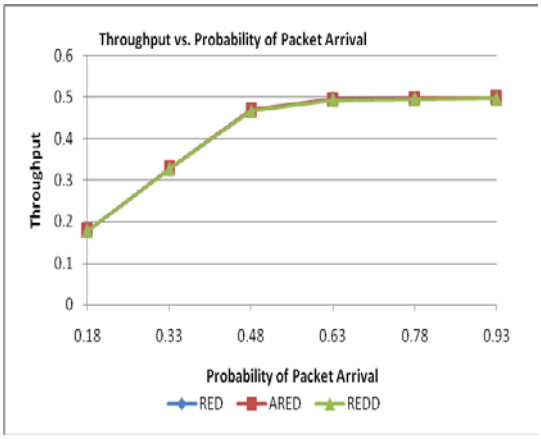

Figure 4.3: T vs. probability of packet arrival.

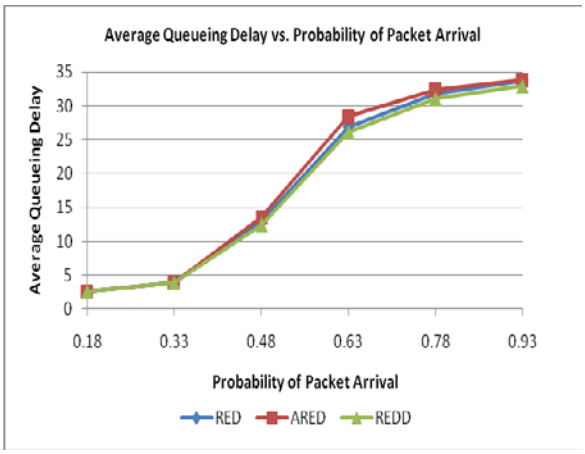

Figure 4.4: D vs. probability of packet arrival.

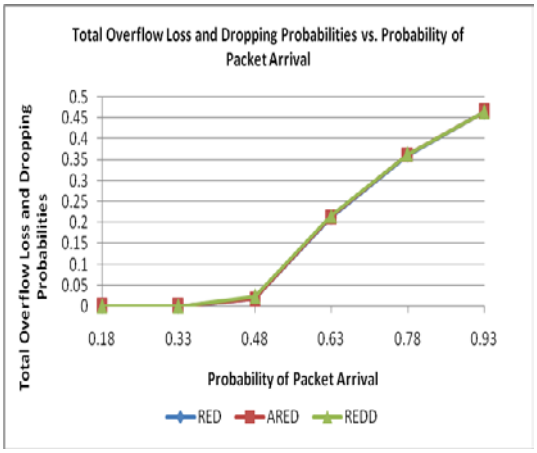

Figure 4.5: P_{Loss} vs. probability of packet arrival.

4.4.2 The Overflow Loss and Dropping Probability Results For RED, ARED and REDD

In this subsection, the proposed REDD algorithm is compared with the RED and ARED algorithms in terms of P_L and D_P to identify which algorithm loses and drops fewer numbers of packets at its router buffer. These packet loses (P_L) happen due to routers buffers of RED, ARED and the proposed REDD are overflow. The parameters of the RED and ARED algorithms have been set as in Subsection 3.1.2 and the REDD parameters were set as in Subsection 4.4.1. The performance measure mean results of the REDD, RED and ARED algorithms concerning P_L and D_P are given in Figures 4.6 and 4.7, respectively.

The performance measure results of the P_L and D_P are computed after the system has reached a steady state. The results of P_L and D_P have been obtained as before, by running the algorithm simulations ten times and by changing the seed each time, then by taking the mean for the ten results. In addition, the statistical analysis of the RED, ARED and REDD

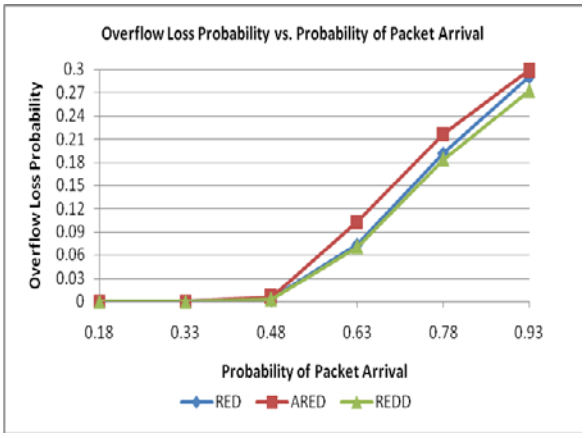

Figure 4.6: P_L vs. probability of packet arrival.

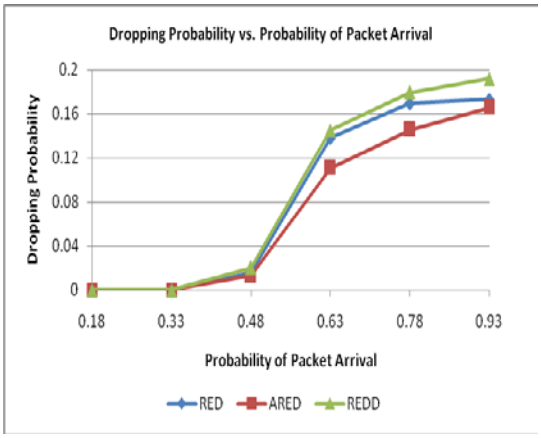

Figure 4.7: D_p vs. probability of packet arrival.

with reference to P_L and D_P are achieved through analysing the results of the ten runs.

Figure 4.6 illustrates that the REDD algorithm marginally produces the smallest P_L result when the probability of packet arrival is larger than the probability of packet departure (existence of congestion), and this is because the router buffer of the REDD algorithm overflows at times smaller those of the RED and ARED algorithms. Overall, the REDD algorithm obtains P_L results better than those of the RED and the ARED algorithms when the packet arrival probability is higher than the packet departure probability. Moreover, when the packet arrival probability is smaller than the packet departure probability, all algorithms provide similar P_L and D_p results under either light congestion or no congestion situations.

It is obvious from Figure 4.7 that the REDD algorithm drops more packets at its router buffer than the RED or ARED algorithms when the probability of packet arrival is higher than the probability of packet departure. This is because the router buffer of the REDD algorithm overflows fewer times than either RED or ARED.

4.4.3 Statistical Analysis of RED, ARED and REDD

A statistical analysis for the performance measures of the RED, ARED and REDD algorithms is introduced in this subsection. This statistical analysis aims to determine how much the results of individual runs are likely to be dispersed from their mean. For instance, after deriving the ten run results for T, the statistical measurements (mean, σ^2, σ, 95% CI)

are applied to describe these results and to identify how much they are dispersed from their mean.

A statistical analysis of the RED algorithm was given in Subsection 3.1.3 whereas the ARED statistical analysis was discussed in Subsection 3.1.3 since it was found that it is enough to present a single AQM algorithm implemented based on aql, i.e. (RED). In this subsection, the statistical analysis of the REDD concerning the probability of packet arrival is introduced in Tables [4.1-4.6]. Specifically, Tables [4.1-4.3] display the statistical analysis with respect to mql, T and D, respectively. In addition, the statistical analysis for the P_L, D_p and P_{Loss} are exhibited in Tables [4.4-4.6], respectively.

After analysing Tables [4.1-4.6], it is clear that the results of statistical measurements (σ^2, σ and 95% CI, upper limit, lower limit) are extremely small than suggesting accurate and repeatable mean results. The largest deviations from the mean were for the REDD method when the packet arrival probability value was set to 0.48, although these deviations were still relatively small.

Table 4.1: Mean mql vs. probability of packet arrival with statistical analysis for the REDD method.

REDD Method						
Probability of Packet Arrival	0.18	0.33	0.48	0.63	0.78	0.93
Mean mql	0.457	1.279	5.812	12.890	15.420	16.392
σ^2	2.883E-06	0.0003	0.041	0.001	0.0003	0.0001
σ	0.001	0.017	0.203	0.040	0.018	0.011
95% CI	0.001	0.011	0.126	0.025	0.011	0.007
Upper Limit	0.458	1.290	5.938	12.916	15.431	16.399
Lower Limit	0.456	1.268	5.685	12.865	15.409	16.385

Table 4.2: Mean T vs. probability of packet arrival with statistical analysis for the REDD method.

REDD Method						
Probability of Packet Arrival	0.18	0.33	0.48	0.63	0.78	0.93
Mean T	0.1786	0.327	0.4671	0.493930	0.49654	0.49759
σ^2	3.640E-07	1.856E-06	1.679E-06	6.826E-09	1.947E-09	4.430E-10
σ	0.0006	0.001	0.001	8.262E-05	4.413E-05	2.104E-05
95% CI	0.0003	0.0008	0.0008	5.121E-05	2.735E-05	1.304E-05
Upper Limit	0.1790	0.328	0.4679	0.493981	0.49657	0.49760
Lower Limit	0.1782	0.326	0.4663	0.493879	0.49651	0.49758

Table 4.3: Mean D vs. probability of packet arrival with statistical analysis for the REDD method.

REDD Method						
Probability of Packet Arrival	0.18	0.33	0.48	0.63	0.78	0.93
Mean D	2.5601	3.903	12.440	26.098	31.055	32.943
σ^2	0.0001	0.002	0.169	0.006	0.001	0.0004
σ	0.010	0.047	0.412	0.081	0.034	0.022
95% CI	0.006	0.029	0.255	0.050	0.021	0.013
Upper Limit	2.5666	3.933	12.696	26.149	31.077	32.956
Lower Limit	2.5536	3.874	12.185	26.048	31.033	32.929

Table 4.4: Mean P_L vs. probability of packet arrival with statistical analysis for the REDD method.

REDD Method						
Probability of Packet Arrival	0.18	0.33	0.48	0.63	0.78	0.93
Mean P_L	0	0	0.0031	0.0702	0.1840	0.27227
σ^2	0	0	1.340E-06	9.741E-07	8.459E-07	8.115E-08
σ	0	0	0.001	0.0009	0.0009	0.000284873
95% CI	0	0	0.0007	0.0006	0.0005	0.000176563
Upper Limit	0	0	0.0038	0.0708	0.1846	0.27245
Lower Limit	0	0	0.0024	0.0696	0.1834	0.27210

Table 4.5: Mean D_p vs. probability of packet arrival with statistical analysis for the REDD method.

REDD Method						
Probability of Packet Arrival	0.18	0.33	0.48	0.63	0.78	0.93
Mean D_p	0	0	0.02009	0.14530	0.17938	0.1924
σ^2	0	0	7.302E-06	3.714E-07	2.689E-07	2.106E-07
σ	0	0	0.002	0.0006	0.0005	0.0004
95% CI	0	0	0.001	0.0003	0.0003	0.0002
Upper Limit	0	0	0.02177	0.14568	0.17970	0.1926
Lower Limit	0	0	0.01842	0.14492	0.17906	0.1921

Table 4.6: Mean P_{Loss} vs. probability of packet arrival with statistical analysis for the REDD method.

REDD Method						
Probability of Packet Arrival	0.18	0.33	0.48	0.63	0.78	0.93
Mean P_{Loss}	0	0	0.023	0.215	0.363	0.4646
σ^2	0	0	1.140E-05	1.247E-06	1.151E-06	2.047E-07
σ	0	0	0.003	0.001	0.001	0.0004
95% CI	0	0	0.002	0.0006	0.0006	0.0002
Upper Limit	0	0	0.025	0.216	0.364	0.4649
Lower Limit	0	0	0.021	0.214	0.362	0.4643

4.5 Chapter Summary

In this chapter, a new AQM algorithm is proposed based on the RED algorithm to control the congested router buffers as early as possible. The proposed algorithm (REDD) also

aims to stabilising the aql between the $\min threshold$ and the $\max threshold$ by updating the $\max threshold$ position at the router buffer. The $\max threshold$ is decreased or increased by two each time the aql is between $\min threshold$ and T_{aql} or between T_{aql} and $\max threshold$, respectively. This decreasing or increasing of the aql helps in stabilising the aql at the T_{aql}. Also, the proposed REDD algorithm decrease a large RED reliance upon $\max threshold$ parameter as a constant rather it is used an adaption $\max threshold$ parameter. A comparison between REDD and both the RED and ARED algorithms was given in Section 4.4 with taking in account that the performance measure results of the compared algorithms achieved based on setting the packet arrival probability parameter to different values. Hence, the decision about which algorithm presents more satisfactory performance measure results is generated only based on the parameter values of packet arrival probability. Moreover, the statistical analysis results for the compared algorithms are obtained according to the packer arrival probability values. From a comparison, the following can be summarised:

- The RED, ARED and REDD algorithms provide similar performance measure results ($mql, T, D, P_L, D_p, P_{Loss}$) when the probability of packet arrival was set to a value lower than the probability of packet departure or when there is either no congestion or light congestion.
- The REDD algorithm marginally offers better mql and D results than that of RED and ARED algorithms when the values of probability of packet arrival are greater than the

value of packet departure probability. Also, at these values of packet arrival probability the REDD, RED and ARED algorithms obtain similar T and P_{Loss} results.

- The REDD algorithm marginally outperforms both the RED and ARED algorithms for P_L when the value of probability of packet arrival is larger than the value of probability of packet departure. Also, at these values of the packet arrival probability, both RED and ARED drop fewer packets (D_p) at their router buffers than REDD.
- The statistical measurement results (σ^2, σ and 95% CI, upper limit, lower limit) of the REDD algorithm are extremely small and not widely dispersed from their mean. This suggests accurate and repeatable results for the performance measure means.
- The largest deviations were observed when the packet arrival probability was equal to 0.48 although these were still very small. This suggests the system is less stable when the probability of packet arrival and probability of packet departure are approximately the same. This is likely because the system could drift from a congestion situation to a non-congestion situation depending on statistical fluctuations in the arrival and departure processes as the simulation progresses.