一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

Q1. Total Received Flit :

1. Flit是Flow control unit。
2. 而Total Received Flit 是simulation的期間，成功收到的所有flit的數量。
3. 而計算方法為所有flit數量的加總。

Q2. Total Received Packet :

1. Data通常被打包成packet進行傳輸，每個packet通常由多個flit組成。
2. 而Total Received Packet是simulation期間，成功收到的所有packet數量
3. 而計算方法為所有packet數量的加總。

Q3. Global Average Delay :

1. Global Delay是指一個packet從src到des的時間
2. Global Average Delay是指所有Global Delay的平均時間
3. 通常以clock cycle作為單位
4. 公式為 Global Average Delay = (所有packet delay) / (所有packet數)

Q4. Network Throughput :

1. Throughput是指單位時間內，能夠處理的平均資料數。
2. 用來衡量網路資料傳輸能力的重要指標
3. 通常以bits/second或是bps作為單位，而此處以flits/second當單位
4. 公式為Network Throughput = (接收資料總單元數) / (模擬的總週期數)

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

開始的warm\_up\_time 1000cycles

所以總共剩餘10000 – 1000 = 9000

* Network Throughput = total flits / total cycles = 11374 / 9000 = 1.26378

Q5. Average IP Throughput :

1. Average IP through是指每個IP在單位時間內所處理的平均資料數。
2. 用來衡量每個IP是否公平運用的重要指標
3. 公式為 Average IP through = (Network Through / IP總數量)

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述根據config\_examples.yaml的內涵，

我們知道這裡有4 x 4 共16個IP node

拿剛剛的throughput，再根據公式

* Average IP through = (Network Through / IP總數量) = 1.26378 / 16 = 0.078986

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

Default\_config.yaml的原始數據 :

Max\_packet\_size = min\_packet\_size = 8

packet\_injection\_rate = 0.01

simulation\_time = 10000



一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

當max & min\_packet\_size從8 -> 16(**上升**)的時候，可以發現 ：

**Total received packet**s 從 1422 -> 1418，**幾乎不變**

**Total received flits** 從11374 -> 22682，可能因為packet的數量雖然沒變，但增加數量不多，而min&max \_packet\_size卻增加，**導致Total flits大幅上升。**

**Global average delay** 從 11.4578 -> 41.0275，可能因為packet size變大，致使每個packet在router處理時間更長，**因此增加了delay。**

**Network Throughput** 從 1.26378 -> 2.52022，因為total flits增加，而cycle數依舊，**所以Network Throughput會增加。**

**Average IP Throughput** 從 0.0789861 -> 0.157514，因為Throughput增加，而IP\_node數沒有改，**所以Average IP Throughput也跟著增加。**



一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述當max&min\_packet\_size 從 8 -> 2(**下降**)的時候，可以發現：

**Total received packets** 從 1422 -> 1453，**幾乎沒變。**

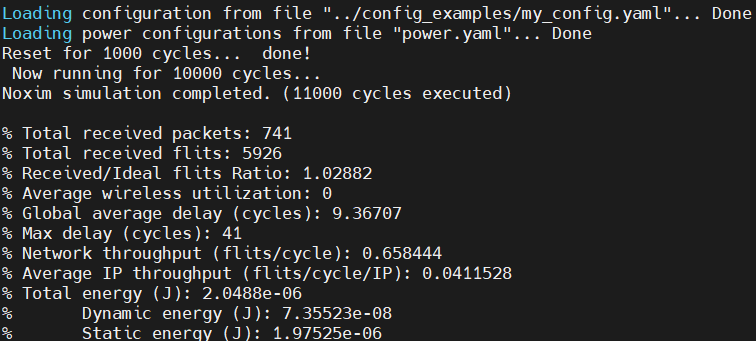
**Total received flits** 從11374 -> 2906，可能因為packet的數量雖然沒變，但最大packet\_size卻大幅下降，**至使Total flits大幅下降。**

G**lobal average delay** 從 11.4578 -> 7.47488，可能因為packet size變小，致使每個packet在router處理時間需要更快，**因此降低了delay。**

**Network Throughput** 從 1.26378 -> 0.322889，因為total flits降低，而cycle數依舊，**所以Network Throughput會降低。**

**Average IP Throughput** 從 0.0789861 -> 0.0201806，因為Throughput降低，而IP\_node數沒有改，**所以Average IP Throughput也跟著降低。**





當packet\_injection\_rate從 0.01 -> 0.005(**下降**)的時候，可以發現：

**Total received packets** 從 1422 -> 719，可能因為injection rate下降，造成生成packet的頻率下降，**因此packets大幅下降(將近一半)**

**Total received flits** 從11374 -> 5926，可能因為injection rate下降，造成生成packet的頻率下降，**致使flits大幅下降(將近一半)**

**Global average delay** 從 11.4578 -> 9.36707，可能因為injection rate下降， 讓packet在router的buffer排隊時間減少，因此使得**Global 的delay下降。**

**Network Throughput** 從 1.26378 -> 0.658444，因為total flits下降，而cycle數依舊，**所以Network Throughput會下降**。

**Average IP Throughput** 從 0.0789861 -> 0.411528，因為Throughput下降，而IP\_node數沒有改，**所以Average IP Throughput也跟著下降。**



一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

當packet\_injection\_rate從 0.01 -> 0.02(**上升**)的時候，可以發現：

**Total received packets** 從 1422 -> 2894，可能因為injection rate上升，造成生成packet的頻率上升，**因此packets大幅上升(將近2倍)**

**Total received flits** 從11374 -> 23150，可能因為injection rate上升，造成生成packet的頻率上升，**致使flits大幅上升(將近2倍)**

**Global average delay** 從 11.4578 -> 21.8386，可能因為injection rate上升， 讓packet在router的buffer排隊時間增加，因此使得**Global 的delay上升**。

**Network Throughput** 從 1.26378 ->2.57222，因為total flits上升，而cycle數依舊，**所以Network Throughput會上升**。

**Average IP Throughput** 從 0.0789861 -> 0.160764，因為Throughput上升，而IP\_node數沒有改，**所以Average IP Throughput也跟著上升**。



一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

當simulation\_time從 10000 -> 20000(上升)的時候，可以發現：

**Total received packets** 從 1422 -> 2966，可能因為simulation time變成雙倍，使得cycle數變為原來的2倍，使得運送的**packet大概為原來2倍**

**Total received flits** 從11374 -> 23150，可能因為simulation time變成雙倍，cycle便2倍，使得packet變為2倍，**致使flits大幅上升(將近2倍)**

**Global average delay** 從 11.4578 -> 11.8479，可能因為只有simulation time上升， cycle上升，不會影響任何原來的排隊時間，因此將近**沒變**。

**Network Throughput** 從 1.26378 ->1.24889，因為total flits上升，而cycle數也改變，**所以Network Throughput幾乎沒變**。

**Average IP Throughput** 從 0.0789861 -> 0.0780559，因為Throughput沒變，而IP\_node數也沒有改，所以**Average IP Throughput也幾乎沒變。**



一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

當simulation\_time從 10000 -> 5000(**下降**)的時候，可以發現：

**Total received packets** 從 1422 -> 628，可能因為simulation time變少1/2倍，使得cycle數變為原來的1/2倍，使得運送的**packet大概為原來1/2倍**

**Total received flits** 從11374 -> 5005，可能因為simulation time變少1/2倍，cycle變1/2倍，使得packet變為1/2倍，致使**flits大幅下降(將近1/2倍)**

**Global average delay** 從 11.4578 -> 11.7484，可能因為只有simulation time下降， cycle下降，不會影響任何原來的排隊時間，因此**將近沒變。**

**Network Throughput** 從 1.26378 -> 1.25125，因為total flits下降2倍，而cycle數也下降2倍，所以**Network Throughput會跟原來很像。**

**Average IP Throughput** 從 0.0789861 -> 0.0782031，因為Throughput沒變，而IP\_node數也沒有改，所以**Average IP Throughput幾乎沒變。**

一張含有 文字, 字型, 螢幕擷取畫面, 白色 的圖片

自動產生的描述

一張含有 文字, 行, 圖表, 繪圖 的圖片

自動產生的描述

Saturation point = 0.03左右，從那邊開始大幅上升，

且flits不太動，大概維持在4500左右。

一張含有 文字, 字型, 螢幕擷取畫面, 行 的圖片

自動產生的描述

一張含有 文字, 行, 繪圖, 圖表 的圖片

自動產生的描述

從圖表中可以觀察到，隨著 simulation time的增加，總能量 Total Energy 也呈現線性增加的趨勢。

Simulation time短，代表系統需要運行的cycle更少，因此整體的energy消耗較低。而simulation time長時，系統需要完成更多的計算，消耗的能量自然增加。