The University of Texas At El Paso

Laboratory #1 Analysis.

The Brogrammers

Erick Garcia

Fernando Martinez

**Introduction:**

This project will have its performance tested under the protocols given to us in the project specifications. Below is the graphical representation of what each of them is.

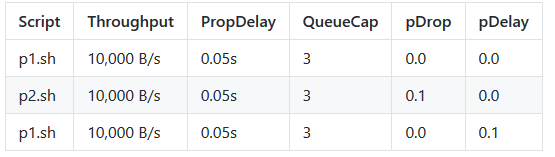


Figure 1: Scripts to be applied to the protocol.

The scripts differ in the amount of delay that there is between each packet and the amount of dropped packages. Script #2 expects a .1 drop rate while script #3 expects a .1 second delay. Script #1 should be regular behavior

This script applied to the proxy will be tested on both the sliding window and the stop and wait protocol for both the put and get. For the sliding window, we will be using a window size of 4. The file to be used for testing purposes is cited below:

This is a testThis is a testThis is a testThis is a test

This is a testThis is a testThis is a testThis is a test

This is a testThis is a testThis is a testThis is a test

This is a test

This is a testThis is a testThis is a testThis is a test

This is a testThis is a testThis is a testThis is a test

This is a testThis is a testThis is a testThis is a test

This is a test

This is a testThis is a testThis is a testThis is a test

This is a testThis is a testThis is a testThis is a test

This is a testThis is a testThis is a testThis is a test

This is a test

Figure 2: Test File which repeats “This is a test”

This file results in exactly 10 packets sent and received for both the put and get protocols.

Testing with the provided scripts:

**Stop and wait:**

GET method performance (seconds):

|  |  |  |  |
| --- | --- | --- | --- |
| Packet # | ./p1.sh | ./p2.sh | ./p3.sh |
| #1 | 0.00773406028748 | 0.00766921043396 | 0.00718283653259 |
| #2 | 0.113914012909 | 0.114939928055 | 0.113559007645 |
| #3 | 0.114323854446 | 0.113868951797 | 1.11622190475 |
| #4 | 0.113611936569 | 0.113831043243 | 0.113765954971 |
| #5 | 0.114147186279 | 0.114104032516 | 0.113753795624 |
| #6 | 0.11420416832 | 0.113989114761 | 1.11352992058 |

PUT method performance(seconds):

|  |  |  |  |
| --- | --- | --- | --- |
| Packet # | ./p1.sh | ./p2.sh | ./p3.sh |
| #1 | 0.116815090179 | 0.116059064865 | 0.116206169128 |
| #2 | 0.231116056442 | 0.231692075729 | 0.231088161469 |
| #3 | 0.345794916153 | 0.346776008606 | 1.34745502472 |
| #4 | 0.460702896118 | 0.462593078613 | 1.46169114113 |
| #5 | 0.575320005417 | 0.577679157257 | 1.57603406906 |
| #6 | 0.689610004425 | 0.69344496727 | 1.69133901596 |

**SLIDING WINDOW:**

GET method performance (seconds):

|  |  |  |  |
| --- | --- | --- | --- |
| Packet # | ./p1.sh | ./p2.sh | ./p3.sh |
| #1 | 0.0062780380249 | 0.00694513320923 | 0.00669693946838 |
| #2 | 0.120355129242 | 0.12077999115 | 1.12114596367 |
| #3 | 0.234594106674 | 0.234589099884 | 1.23514604568 |
| #4 | 0.348350048065 | 0.348433971405 | 2.34922909737 |
| #5 | 0.114703893661 | 0.114161014557 | 0.113929986954 |
| #6 | 0.226347923279 | 0.22511100769 | 0.22482585907 |

PUT method performance (seconds):

|  |  |  |  |
| --- | --- | --- | --- |
| Packet # | ./p1.sh | ./p2.sh | ./p3.sh |
| #1 | 0.015625 | 0.115841150284 | 0.115693092346 |
| #2 | 0.115494012833 | 0.230030059814 | 0.230268955231 |
| #3 | 0.229990005493 | 0.344094991684 | 0.344095945358 |
| #4 | 0.344339847565 | 0.458379983902 | 0.457788944244 |
| #5 | 0.457815885544 | 0.113801002502 | 0.114180803299 |
| #6 | 0.225337982178 | 0.225175857544 | 1.22492694855 |

**Analysis:**

The first script was quite clearly faster as it had no delay and would be what we expect in the ideal world. Similarly, sliding window was faster than using the stop and wait protocol. The second script managed to complete but took a considerable amount of time compared to the first one. Finally, the delays for the third one added up and resulted in a slower transfer.

The scripts prove to be challenging ways to test the program. It exposes it to real world difficulties and how to react to them. However, the protocol seems to have a reasonable time span to complete a package transfer of that size.

**Conclusion:**

This lab allowed for the learning experience of how sliding window is considerably faster than the stop and wait. Similarly, that the real internet does not behave nicely all of the time and could prove to be difficult to react to such adversities.