Performance evaluation of Adaptive Congestion Protocol
(Draft 1)

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Performance evaluation of Adaptive Congestion Protocol

Νεόφυτος Ηλιάδης

Επιβλέπων Καθηγητής
Ανδρέας Πιτσιλλίδης

Η Ατομική αυτή Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Ιούνιος 2007

Ευχαριστίες

Ιδιαίτερες ευχαριστίες θα ήθελα να εκφράσω στον επιβλέποντα καθηγητή της παρούσας διπλωματικής Δρ. Πιτσιλλίδη Αντρέα. Επίσης, θα ήθελα να ευχαριστήσω τους Δρ. Μάριο Λέστα και Μαρίνο Στυλιανού για την βοήθεια που απλόχερα μου έδωσαν σε κάθε δυσκολία που αντιμετώπισα. Ακόμη θα ήθελα να ευχαριστήσω τα μέλη του network group του Πανεπιστημίου Κύπρου για τις πολύτιμες συμβουλές και εισηγήσεις που μου έδιναν. Τέλος θα ήθελα να πω ότι είμαι ευγνώμων προς την οικογένεια και τους φίλους μου για την ηθική υποστήριξη και συμπαράσταση την οποία μου προσέφεραν.

Περίληψη

Η συνεχής ανάπτυξη των απαιτήσεων εφαρμογών που εξαρτώνται και χρησιμοποιούν κάποιας μορφής δικτυακή πρόσβαση μας οδηγεί στην ανάγκη αναβάθμισης των χαμηλότερων δικτυακών στρωμάτων. Από την θεωρία αλλά και από δοκιμές που έχουν γίνει μπορούμε να δούμε ότι καθώς το per-flow product του bandwidth και του latency αυξάνεται, το πρωτόκολλο TCP δεν είναι αποδοτικό και ταυτόχρονα η απόδοση του είναι ιδιαιτέρα ασταθής, ανεξάρτητα από την μεθοδολογία που ακολουθείται για το queuing στο δίκτυο. Αυτή η ανεπάρκεια του TCP γίνεται σταδιακά πολύ σημαντική καθώς το Internet εξελίσσεται χρησιμοποιώντας συνδέσεις οπτικών γραμμών, ασυρμάτων γραμμών και μεγάλης καθυστέρησης δορυφορικών γραμμών.

Το e**X**plicit **C**ontrol **P**rotocol (XCP) δημοσιεύτηκε ως RFC draft τον Οκτώβριο του 2004. Το πρωτόκολλο δημιουργήθηκε με σκοπό να υπερκεράσει το υφιστάμενο TCP congestion control protocol, το οποίο με την πάροδο του χρόνου λειτουργεί ως μειονέκτημα στην ανέλιξη των επικοινωνιών. Το XCP δεν προσπαθεί να προσφέρει συμβατότητα προς τα πίσω σε σχέση με το υφιστάμενο σχήμα, αλλά εισηγείται ένα καινούριο πρωτοποριακό τρόπο επίλυσης των προβλημάτων congestion. Γενική ιδέα αυτού του πρωτοκόλλου είναι πως ο αποστολέας ενημερώνει τους ενδιάμεσους κόμβους προς τον παραλήπτη για τον επιθυμητό ρυθμό αποστολής δεδομένων, οι ενδιάμεση κόμβοι τροποποιούν αυτή την τιμή και την προωθούν στον επόμενο κόμβο, οι κόμβοι που ακολουθούν πράττουν ανάλογα, μέχρι το πακέτο να φτάσει στον παραλήπτη ο οποίος με την σειρά του θα στείλει τον τελικό ρυθμό αποστολής στον αρχικό αποστολέα, ενημερώνοντας τον έτσι για τον ρυθμό αποστολής με τον οποίο το δίκτυο μπορεί να δεκτή δεδομένα χωρίς απώλειες και υπερφόρτωση.

Παρόλα τα ενθαρρυντικά αποτελέσματα που δοκιμές στον XCP έδωσαν το πρωτόκολλο, αυτό τουλάχιστο στα αρχικά στάδια της υλοποίησης του έδειξε ότι έχει αρκετά προβλήματα. Αυτά τα προβλήματα καθώς και περεταίρω βελτίωση του υφιστάμενου τρόπου λειτουργίας έρχεται το **A**daptive **C**ongestion **P**rotocol (ACP) να βελτιώσει. Το ACP λειτουργεί με την λογική που το XCP εργάζεται, αλλάζοντας όμως τους τρόπους υπολογισμού των διάφορων παραμέτρων αλλά και το header του εισηγημένου πρωτοκόλλου. Κάνοντας αυτό όπως διαφαίνεται το ACP λειτουργεί καλύτερα και από το TCP αλλά και από το XCP.

Στην διπλωματική αυτή εργασία η προσπάθεια μας είναι να υλοποιήσουμε τα δύο αυτά πρωτόκολλα και ακολούθως να τρέξουμε σενάρια για τα δύο και να διαπιστώσουμε ποιο πρωτόκολλο υπερέχει

και να βρούμε όποια προβλήματα υπάρχουν. Η υλοποίηση των δύο πρωτοκόλλων έχει γίνει ως ξεχωριστό πρωτόκολλο ανάμεσα στα TCP και IP σε λειτουργικό σύστημα Linux. Παρόλα αυτά μια τέτοια υλοποίηση δεν ήταν εύκολη αφού το γεγονός ότι το ACP χρειαζόταν παραμέτρους από το TCP όπως congestion window.

Κεφάλαιο 1. 8
1. Εισαγωγή 8
Κεφάλαιο 2. 12
2. Explicit Control Protocol (XCP) 12
2.1 Εισαγωγή 12
2.2 Το Header 15
2.3 Μαθηματικοί υπολογισμοί 18
2.3.1 Delta throughput 18
2.3.2 Congestion window 19
2.3.3. Efficiency Controller 20
2.3.4. Queue estimation 21
2.3.5. Άλλες απλές μαθηματικές εξίσωσης στους router 21
2.4 Γενική αξιολόγηση – περίληψη 23
Κεφάλαιο 3. 24
3. Adaptive Congestion Protocol (ACP) 24
3.1 Εισαγωγή 24
3.2. Το Header 27
3.3. Μαθηματικοί υπολογισμοί 29
3.3.1. Sender 29
3.3.1.1. Αρχικοποίηση τιμών 29
3.3.1.2. Desired window 30
3.3.1.3. Congestion Window 30
3.3.2. Routers 32
3.3.2.1. Υπολογισμός ρυθμού αποστολής δεδομένων 32
3.3.2.2. Υπολογισμός αριθμού ροών 33
3.3.2.3. Άλλες απλές μαθηματικές εξίσωσης στους router 33
3.3.3. Receiver 34
3.3.4. Περίληψη 34
Κεφάλαιο 4. 35
4. Εφαρμογή του ACP σε λειτουργικό σύστημα Linux 35
4.1. Υλοποίηση ACP ως ξεχωριστό layer 35
4.2. Υλοποίηση host 35
4.2.1. Sender 35
4.2.2. Receiver 35
4.3. Υλοποίηση router ACP 35
4.3.1. Παραλαβή πακέτου 35
4.3.2. Υλοποίηση υπολογισμών 35
4.3.3. Αποστολή πακέτου 35
4.4. Περίληψη 35
Κεφάλαιο 5. 35
5. Αποτελέσματα ελέγχου απόδοσης ACP 35
5.1 35
5.2. 35
5.3. 35
Κεφάλαιο 6. 35
6. Συμπεράσματα και μελλοντική εργασία 35
Κεφάλαιο 7. 35

7. Βιβλιογραφία..35

Κεφάλαιο 1.

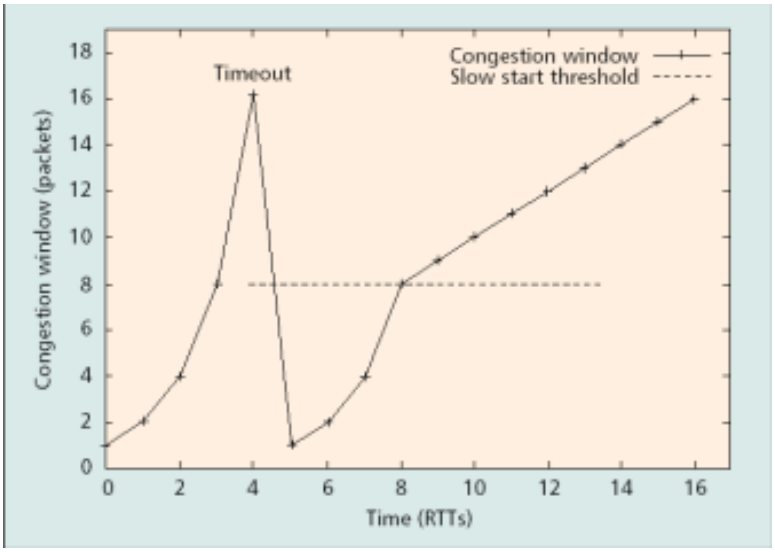
1. Εισαγωγή

Τα τελευταία χρόνια τα δίκτυα έχουν εξαπλωθεί σε όλο το φάσμα των ηλεκτρονικών υπολογιστών και ίσως και σε όλες της καινούριες ηλεκτρικές συσκευές, επίσης η χρίση των ασύρματων δικτύων έχει αυξηθεί εκθετικά. Ταυτόχρονα η ταχύτητα και τα διαθέσιμα δεδομένα στο διαδίκτυο αυξάνονται, ο αριθμός των χρηστών που χρησιμοποιούν το διαδίκτυο επίσης αυξάνεται και οι απαιτήσεις των προηγουμένων μεγαλώνουν ούτως ώστε να έχουν την ευχέρεια να χρησιμοποιούν τις αναπτυσσόμενες αυτές τεχνολογίες. Για παράδειγμα εφαρμογές που ασχολούνται με video streaming είναι ένα καλό παράδειγμα τύπου εφαρμογής το οποίο έχει απαιτήσεις από το δίκτυο, τέτοιες εφαρμογές συχνά λειτουργούν καλύτερα όταν έχουν σταθερό ρυθμό αποστολής δεδομένων αντί για μεγάλους αριθμούς bandwidth. Με την μεγάλη και ραγδαία ανάπτυξη του bandwidth και των ασύρματων δικτύων έχουμε μεγάλη ανάπτυξη του bandwidth-delay product ενός δικτύου. Με τον όρο bandwidth delay product εννοούμε τον αριθμό των δεδομένων που βρίσκονται «εν πτήση» στο δίκτυο.

Είναι κοινός αποδεκτό ότι το TCP πρωτόκολλο είναι το πλέον διαδεδομένο πρωτόκολλο στο Internet για τα τελευταία 20 και πλέον χρόνια . Σε αυτά τα χρόνια ο κόσμος των υπολογιστών έχει υποστεί μια εκπληκτική ανάπτυξη που πολύ λίγοι θα περίμεναν. Οι ταχύτητα επεξεργασίας των υπολογιστών, οι αποθηκευτικοί χώροι, η δικτυακές ταχύτητες είναι μερική παράγοντες που έχουν υποστεί αυτή την μεγάλη αλλαγή, παρόλα αυτά το TCP/IP protocol έχει μεν υποστεί κάποιες αλλαγές , αλλά δεν παύει να είναι βασισμένο στις ίδιες αρχές με τις οποίες δημοσιεύτηκε το 1981. Παρόλα αυτά το TCP έχει αποδεικτέ πολύ ευπροσάρμοστο σε όλες της αλλαγές και σε αυτό μάλιστα οφείλεται η μακροζωήα του.

Το αρχικό πρότυπο του TCP δεν ανάφερε πουθενά τον παράγοντα congestion control, αυτό είναι πολύ λογικό δεδομένου ότι οι εφαρμογές και δοκιμές τις τότε εποχής μακράν απείχαν από τα σημερινά δεδομένα, (πολύ μικρός ρυθμός αποστολής, πολύ χαμηλές δυνατότητες των ίδιον τον υπολογιστών). Πολλές βελτιστοποιήσεις του TCP δημοσιεύτηκαν από την πρώτη μέρα παρουσίασης του, κάποιες από αυτές υιοθετήθηκαν από το πρωτόκολλο, μετατρέποντας το στο πρωτόκολλο που σήμερα κυριαρχεί στο διαδίκτυο. Παρόλα αυτά είναι μαθηματικός αποδεδειγμένο ότι καθώς οι αποτίσεις αυξάνονται ο αλγόριθμος που χρησιμοποιείται σήμερα για το congestion control είναι ανεπαρκής.

Ένα ακόμη πρόβλημα του TCP είναι ο τρόπος με τον οποίο διαχειρίζεται το bandwidth στο δίκτυο όταν έχουμε μεγάλο bandwidth-delay product. Το AIMD (**A**dditive **I**ncrease, **M**ultiplicative **D**ecrease) του TCP αποτρέπει το TCP από το να χρησιμοποιήσει το διαθέσιμο bandwidth σε μικρό χρόνο. Όπως ξέρουμε το TCP δεν πρόκειται να αυξήσει το bandwidth του παρά μόνο κατά ένα πακέτο ανά ACK. Σε περίπτωση απώλειας πακέτου το TCP θα χρησιμοποιήσει το «Multiplicative Decrease» για να μείωση το throughput (TCP Reno). Από εκείνη την στιγμή και μετά ο αλγόριθμος θα μπει στο κομμάτι του «Additive Increase» αυξάνοντας έτσι το παράθυρο του κατά ένα πακέτο ανά RTT (Round Trip Time). Από τα πιο πάνω μπορούμε να καταλάβουμε ότι ο χρόνος που θα χρειαστεί ένα TCP δίκτυο για να αύξηση των ρυθμό αποστολής δεδομένων είναι ανάλογος του RTT κάτι που μπορούμε να φανταστούμε πόσο μεγάλο πρόβλημα θα ήταν σε δίκτυα με μεγάλο RTT , όπως για παράδειγμα δίκτυα με δορυφορικές συνδέσεις. Παρόλο που υπάρχουν αρκετές παραλλαγές των αλγορίθμων congestion control του TCP όλες χρησιμοποιούν το ίδιο μοντέλο AIDM διαφοροποιώντας πότε ο αλγόριθμος θα μπει σε slow start είτε congestion avoidance phase (TCP Reno, TCP Tahoe, TCP Vegas).



Το TCP χρησιμοποιεί την απώλεια πακέτων ως τρόπο αναγνώρισης congestion στο δίκτυο μειώνοντας έτσι τον ρυθμό αποστολής του. Ο τρόπος αυτός αναγνώρισης congestion δίνει στο δίκτυο πολύ λίγες πληροφορίες για το τι προκάλεσε την απώλεια αυτή, με αποτέλεσμα ο αποστολέας να μην μπορεί να αντιδράσει με τον ιδανικότερο τρόπο για να κρατήσει ψηλά την απόδοση του δικτύου. Ένα ακόμη

ελάττωμα του παρόντος συστήματος είναι το γεγονός ότι το μήνυμα για απώλεια πακέτου συνήθως εξαρτάται από τον χρόνο στον οποίο θα έχουμε timeout του αποστολέα, αυτή η χρονική περίοδος είναι ουσιαστικά ίση με το RTT, αυτό όπως έχουμε αναφέρει προηγουμένως είναι πρόβλημα αφού σε πολλές περιπτώσεις έχουμε μεγάλα RTT. Όπως αντιλαμβανόμαστε το να περιμένει το δίκτυο ένα timeout σε πολλές περιπτώσεις και ειδικά όταν το RTT είναι μεγάλο στοιχίζει στην απόδοση του δικτύου αφού πολύ πιθανόν να μην εκμεταλλευόμαστε στο μέγιστο τις δυνατότητες του δικτύου μας περιμένοντας την απόκριση του αποδέκτη ή το timeout.

Αν σκεφτούμε πως το πρωτόκολλο TCP βλέπει το δίκτυο, θα διαπιστώναμε ότι δεν γνωρίζει απολύτως τίποτα για αυτό, το TCP δεν προσπαθεί να μάθει τίποτα για τις δυνατότητες του δικτύου στο οποίο βρίσκεται. Ο λόγος για αυτή την συμπεριφορά του TCP είναι πολύ απλός, όταν αυτό δημιουργήθηκε δεν υπήρχε λόγος να γνωρίζει οτιδήποτε για το δίκτυο στο οποίο εφαρμοζόταν για να λειτουργεί, αυτή η άγνοια του TCP συνεχίζεται μέχρι και σήμερα. Η Dina Katabi προσπάθησε να πάρει μια άλλη προσέγγιση στο θέμα του congestion control, αποφάσισε να φτιάξει κάτι καινούριο και να μην προσπαθήσει να διορθώσει την παρούσα κατάσταση. Αυτό ήλθε με την εισήγηση του explicit congestion protocol (XCP), το οποίο ήταν ένα πρωτόκολλο που αναδημιουργούσε όλο τον τρόπο λειτουργίας του TCP αλγόριθμου congestion control.

Η προσπάθεια της Katabi ήταν να δημιουργήσει ένα πρωτόκολλο όπου το δίκτυο θα έδινε εκτενής πληροφορίες προς τον αποστολέα για την κατάσταση του δικτύου, και ιδιαίτερα το επίπεδο συμφόρησης αυτού. Η σκέψη της ήταν πως ο αποστολέας θα έπαιρνε ανατροφοδότηση από τον παραλήπτη για τα επίπεδα συμφόρησης , φυσικά αυτό δεν θα μπορούσε να γίνει χωρίς την ενεργή συμμετοχή των router, με την παραλαβή της ανατροφοδότησης ο αποστολέας θα είχε την δυνατότητα να αλλάξει δυναμικά τον ρυθμό αποστολής του, χρησιμοποιώντας έτσι το δίκτυο στο μέγιστο και αποφεύγοντας σενάρια συμφόρησης.

Αργότερα όταν τα πρώτα προβλήματα του XCP εμφανίστηκαν ερευνητές του Πανεπιστημίου Κύπρου πρότειναν ένα νέο πρωτόκολλο το οποίο ήταν βασισμένο στην ιδέα του XCP αλλά αποσκοπούσε στην βελτίωση των προβλημάτων που αυτό είχα καθώς και στην βελτιστοποίηση των υπολογισμών για καλύτερη ανατροφοδότηση προς τον αποστολέα. Το πρωτόκολλο αυτό που προτάθηκε ήταν το Adaptive Congestion Protocol (ACP) συγγραφής αυτού οι Μάριος Λέστας, Αντρέας Πιτσιλλίδης, Πέτρος Ιωάννου και Γιώργος Χατζιπολλάς.

Σε αυτή την διπλωματική εργασία θα παρουσιάζουμε την σχεδίαση καθώς και τον τρόπο λειτουργίας του XCP(κεφάλαιο 2), θα δούμε πώς αυτό εργάζεται και πως προσπαθεί να επιτύχει τα ποθητά αποτελέσματα, στην συνέχεια θα δούμε πως ακριβός είναι δομημένο και αφού δούμε πως αυτό χειρίζεται τις διάφορες μαθηματικές πράξεις στο εσωτερικό του θα το αξιολογήσουμε.

Το επόμενο κεφάλαιο της διπλωματικής (Κεφάλαιο 3) θα κάνουμε μια εκτενή αναφορά στο πρωτόκολλο ACP, θα δούμε την προσέγγιση που ακολουθείτε για την αποφυγή συμφόρησης στο δίκτυο. Θα αναφερθούμε στην ιδέα του αλγορίθμου στης μαθηματικές πράξεις που τον απαρτίζουν, καθώς και στις διαδικασίες που λαμβάνουν χώρα σε κάθε εμπλεκόμενο κόμβο ενός δικτύου. Όπως έχουμε προαναφέρει και θα αναφέρουμε εκτενέστερα στην συνέχεια οι routers λαμβάνουν μέρος στην διαδικασία υπολογισμού του ρυθμού αποστολής πληροφοριών.

Με σκοπό την καλύτερη αξιολόγηση του πρωτοκόλλου ACP υλοποιήσαμε μια εκδοχή του αλγορίθμου για λειτουργικό σύστημα Linux. Η διαδικασία της υλοποίησης αυτού του πρωτοκόλλου επεξηγείτε στο κεφάλαιο 3 μαζί με δυσκολίες που αντιμετωπίσαμε κατά την υλοποίηση. Εξαιτίας της ιδιομορφιών του λειτουργικού συστήματος έπρεπε να υλοποιήσουμε το πρωτόκολλο ως ξεχωριστό κομμάτι ενδιάμεσα του TCP και IP , αυτό όμως δεν ήταν εύκολο δεδομένου ότι το πρωτόκολλο θα έπρεπε να παίρνει πληροφορίες από το TCP.

Το κεφάλαιο 5 της διπλωματικής εργασίας περιέχει διάφορα αποτελέσματα που πήραμε χρησιμοποιώντας την υλοποίηση μας, γίνεται σύγκριση των αλγορίθμων ACP, XCP, TCP και τα αποτελέσματα που περνούμε δείχνουν ότι ο ACP πετυχαίνει τα αναμενόμενα αποτελέσματα, φυσικά όπως διαφάνηκε και από την υλοποίηση ο αλγόριθμος για να λειτουργήσει χρειάζεται τουλάχιστο ένα router ο οποίος να είναι συμβατός με αυτόν.

Τα συμπεράσματα και τα όσα διαπιστώσαμε από την εργασία μας στο αντικείμενο βρίσκονται στο κεφάλαιο 6.

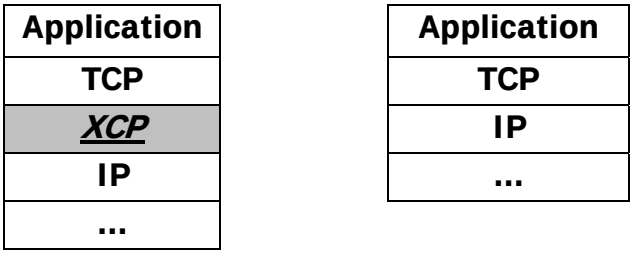
Τέλος όλα οι πηγές οι οποίες χρησιμοποιήθηκαν για την αποπεράτωση της εργασίας αυτής βρίσκονται στο κεφάλαιο 7

Κεφάλαιο 2.
2. Explicit Control Protocol (XCP)

2.1 Εισαγωγή
Οι εκδότες του XCP έχουν εισηγηθεί ένα αλγόριθμο ο οποίος προσπαθεί να υπερκεράσει το ήδη υπάρχον TCP congestion control. Όπως έχουμε προαναφέρει το πρωτόκολλο που χρησιμοποιείται σήμερα έχει πολλά προβλήματα ιδιαιτέρα όταν πρέπει να αντιμετωπίσει εφαρμογές της νέας γενεάς οι οποίες χρησιμοποιούν gigabyte συνδέσεις για να μεταφέρουν αρχεία, ασύρματα δίκτυα που όπως γνωρίζουμε έχουν πολλές απώλειες και συνδέσεις που χαρακτηρίζονται από μεγάλη καθυστέρηση (latency). Ο σκοπός δημιουργίας όπως η σχεδιαστές του το περιγράφουν είναι να προσφέρει την μέγιστη δυνατή απόδοση σε όλων των ειδών τα δίκτυα. Το XCP μπορεί να χειριστεί συνδέσεις με πολύ μεγάλο bandwidth, καθυστέρηση, και ταχύτητα, κάτι που όπως φαίνεται κάνει με αρκετά καλά.

Όπως έχει αναφερθεί και προηγουμένως το XCP υιοθετεί κάποιες νέες αρχές σε σχέση με τον τόπο που λειτουργεί. Αυτό που κάνει το XCP είναι να μεταφέρει ένα per-flow congestion state στα πακέτα που αποστέλλει, με αυτό εννοούμε πως στο header του πακέτου υπάρχει ένα πεδίο το οποίο δηλώνει τον επιθυμητό ρυθμό αποστολής δεδομένων από τον χρήστη-αποστολέα, αυτό το πεδίο σε συνδυασμό με το γεγονός ότι το πρωτόκολλο υποθέτει πως το δίκτυο απαρτίζεται από router οι οποίοι είναι σε θέσει να διαπιστώνουν την κατάσταση του δικτύου ενημερώνοντας την τιμή αυτή, έχουν την δυνατότητα να ενημερώνουν τον αποστολέα για την παρούσα κατάσταση του δικτύου επιτρέποντας του να ρυθμίσει δυναμικά τον ρυθμό αποστολής δεδομένων του. Με το να αφήνει το δίκτυο να του δίνει πληροφορίες για την κατάσταση στην οποία βρίσκεται το XCP προσπαθεί να αποτρέπει σενάρια υπερφόρτωσης.

Ο XCP δεν έχει σκοπό να προσθέσει στα ήδη υπάρχοντα στρώματα δικτύου, αλλά αποσκοπεί στο να προσθέσει ένα επιπλέον επίπεδο που θα τοποθετηθεί ενδιάμεσα στα IP και TCP επίπεδα. Ο λόγος για τον οποίο το XCP δεν υλοποιήθηκε ως μια IP επιλογή είναι λόγω του ότι οι router βλέποντας ένα πακέτο το οποίο έχει κάποιο IP option το κρατούν για περεταίρω επεξεργασία, κάτι που προφανώς κάνει την διαδικασία του forwarding πολύ πιο χρονοβόρα.



Για να βελτιώσουν περεταίρω τον τρόπο με τον οποίο το TCP αντιδρά ως προς την αύξηση του RTT θα πρέπει το πρωτόκολλο να αναγνωρίζει τέτοιες περίπτωσης και να αποστέλλει τα πακέτα του με πιο αργό ρυθμό. Το XCP αυτόματα μειώνει τον ρυθμό αποστολής δεδομένων όταν παρατηρεί αύξηση στον χρόνο RTT. Έτσι το πρωτόκολλο επιτυγχάνει μια πιο σταθερή συμπεριφορά σε σχέση με το TCP, το οποίο είναι εξαιρετικά ασταθές και επιρρεπές σε τέτοια προβλήματα.

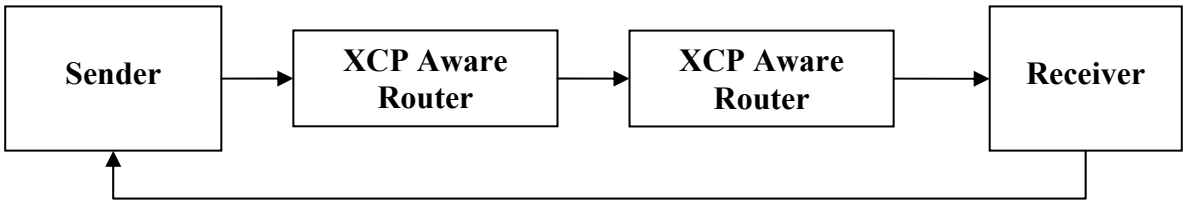
Προβλήματα fairness εμφανίζονται στο TCP, ιδιαίτερα σε περιπτώσεις οπού το RTT είναι μεγαλύτερο, αν σκεφτούμε ότι κάθε RTT έχουμε την αύξηση του ρυθμού αποστολής σε ένα host. Το XCP δεν πάσχει από τέτοιου είδους προβλήματα, επίσης στο XCP δεν παρατηρείται καθυστέρηση στο να πάρουμε το μέγιστο δυνατό bandwidth κάτι που αν σκεφτούμε τον τρόπο λειτουργίας του TCP θα διαπιστώσουμε ότι χρειάζεται κάποιο χρονικό διάστημα. Αυτό συμβαίνει εξαιτίας του AIMD όπως το περιγράψαμε ποιο πάνω, έτσι βλέπουμε ότι το TCP πάσχει και από αυτό το πρόβλημα. Το XCP χρησιμοποιεί ένα πολύ πιο επιθετικό τρόπο κατανομής του bandwidth ο οποίος τρόπος φαίνεται να δουλεύει καλύτερα από το TCP.

Το XCP βασίζεται στην δημιουργία ενός νέου protocol layer και header το οποίο θα τοποθετηθεί μεταξύ των IP και TCP. Το header έχει μέγεθος 20bytesκαι τοποθετείτε πριν από το TCP και μετά από το IP. Οι router στο XCP δεν διατηρούν κρατούν καμία μεταβλητή σε σχέση με την κατάσταση του δικτύου, η τιμή της ανατροφοδότησης υπολογίζεται με την παραλαβή κάθε πακέτου. Με αυτό τον τρόπο επιτυγχάνεται μια απλή υλοποίηση του XCP σε επίπεδο router.

Το πρωτόκολλο XCP λειτουργεί με το να τοποθετεί επιπλέον πληροφορίες στο πακέτο που αποστέλλεται από τον αποστολέα. Η πληροφορίες που αποστέλλονται υπολογίζονται από τον αποστολέα και τροποποιούνται κατά την διαδρομή του πακέτου προς τον παραλήπτη από τους router, τέλος ο παραλήπτης επιστρέφει της πληροφορίες στον αρχικό αποστολέα στο πακέτο ACK. Η πληροφορίες που αποστέλλονται έχουν σχέση με τον υπολογισμό του RTT, την διαφορά στο throughput (η επιθυμητή αλλαγή στο παρών throughput) και το throughput κατά την στιγμή της αποστολής του πακέτου.

Όπως έχουμε προαναφέρει για να λειτουργήσει ο αλγόριθμος πρέπει να υπάρχει τουλάχιστο ένας router ο οποίος να είναι συμβατός με το πρωτόκολλο. Σε περίπτωση που κανένας router δεν υποστηρίζει το XCP τότε ο αποστολέας θα αποστείλει τα δεδομένα με την μέγιστη δυνατή ταχύτητα, κάτι που προφανώς θα είναι ανάλογο με το να έχουμε TCP χωρίς κανένα congestion control μηχανισμό (το XCP ακάθιστα το υφιστάμενο TCP congestion control αλγόριθμο). Για να

λειτουργήσει με μέγιστη απόδοση ο XCP πρέπει όλοι οι router να είναι συμβατή με το πρωτόκολλο, σε περίπτωση που δεν είναι όλη τότε προφανώς το πρωτόκολλο θα δουλέψει αλλά όχι στον μέγιστο βαθμό.



Ο κάθε router στο πρωτόκολλο αυτό έχει υποχρέωση να ελέγχει τον φόρτο δεδομένων που τον διαπερνά την συγκεκριμένη στιγμή, αν μπορεί να δεκτή τον αριθμό δεδομένων που ο αποστολέας θέλει να αποστείλει δεν προβαίνει σε οιεσδήποτε αλλαγές. Αν όμως ο router είναι υπερφορτωμένος, τότε θα προβεί σε αλλαγή του επιθυμητού αριθμού αποστολής δεδομένων. Αλλάζοντας τον στην τιμή που αυτός θα μπορεί να αποδεκτή. Έτσι όταν το πακέτο φτάσει στον προορισμό του ο παραλήπτης θα αποστείλει το πακέτο στον αποστολέα ενημερώνοντας τον για την κατάσταση του δικτύου. Ο χρόνος ο οποίος απαιτείτε για ένα αποστολέα να αλλάξει τον ρυθμό αποστολής δεδομένων δεν είναι άλλος από ένα RTT.

2.2 Το Header

Μια ποιο αναλυτική παρουσίαση του header.

Version	Format	Protocol	Length	Unused
RTT				
Throughput				
Delta Throughput				
Reverse Feedback				

Το header έχει μέγεθος 20bytes και η μορφή του έχει την ποιο πάνω διαμόρφωση.

Τα στοιχειά που απαρτίζουν το header περιγράφονται αναλυτικότερα ποιο κάτω.

- **Version:**
  - 4bits
  - Σε αυτό το πεδίο τοποθετείτε η έκδοση του XCP, η τιμή αυτή είναι προκαθορισμένη, για την υλοποίηση με την οποία εμείς ασχοληθήκαμε η τιμή του version είναι 0x01

- **Format:**
  - 4bits
  - Το πεδίο αυτό περιέχει ένα κωδικό ο οποίος χαρακτηρίζει την μορφή την οποία έχει το υπόλοιπο πακέτο, υπάρχουν δύο διαφορετικές μορφές του XCP για την υλοποίηση που μελετούμε.
    - **Standard Format** (code : "0x1")
      Η μορφή αυτή είναι η κοινή μορφή για το XCP, η μορφή αυτή χρησιμοποιείτε κατά την αποστολή δεδομένων, δηλαδή σε αυτή την κωδικοποίηση τα πεδία RTT, throughput, delta_throughput είναι σε χρίση από το πρωτόκολλο. Προφανώς ένας router βλέποντας την τιμή αυτή έχει το δικαίωμα της αλλοίωσης των τιμών.
    - **Minimal Format** (code : "0x2")
      Η μορφή αυτή χρησιμοποιείται όταν πρόκειται για πακέτο ACK, δηλαδή πακέτο ανατροφοδότησης, σε αυτή την περίπτωση τα πεδία RTT, delta_throughput έχουν την τιμή μηδέν και η router δεν έχουν κανένα δικαίωμα μεταβολής τους. Αντίθετα το πεδίο reverse_feedback περιέχει τον επιτρεπόμενο ρυθμό αποστολής του αρχικού αποστολέα.

- **Protocol:**
  - 8bits
  - Αυτό το πεδίο περιέχει τον τύπο του πρωτοκόλλου που ακολουθεί, συνήθως αυτό το πρωτόκολλο είναι το TCP.

- **Length:**
  - 8bits
  - Αυτό το πεδίο περιέχει το μέγεθος του congestion header στην προκειμένη περίπτωση όπως ήδη έχουμε αναφέρει το μέγεθος αυτό είναι 20bytes.

- **Unused:**
  - 8bits
  - Αυτό το πεδίο παραμένει κενό στην προκειμένη έκδοση του XCP, η ύπαρξη του οφείλεται σε πιθανή μελλοντική χρίση.

- **RTT:**
  - 32bits
  - Το πεδίο αυτό είναι ένας unsigned integer ο οποίος περιγράφει το RTT (round trip time). Το RTT μετριέται σε milliseconds. Η μικρότερη τιμή που μπορεί να πάρει είναι το 1ms και η μεγαλύτερη είναι περίπου 49 ημέρες. Αν η τιμή του RTT είναι 0 τότε αυτό σημαίνει ότι ο αποστολέας δεν γνωρίζει την τιμή του RTT (συνηθίζεται κατά το πρώτο πακέτο κάποιας μετάδοσης) . Η τιμή αυτή είναι πάντοτε στρογγυλοποιημένη προς τα πάνω.

- **Throughput:**
  - 32bits
  - Η τιμή αυτή αναπαριστά το throughput όπως αυτό υπολογίστηκε από τον αποστολέα την στιγμή που το πακέτο αναχωρεί από αυτόν. Η τιμή αυτή είναι στρογγυλοποιημένη προς τα πάνω και μετριέται σε millisecond. Η μέγιστη τιμή είναι περίπου τα 34Gbps.

- **Delta throughput:**
  - 32bits
  - Η τιμή αυτή αναφέρεται στην επιθυμητή αλλαγή που θέλει να κάνει ο αποστολέας στο throughput (για παράδειγμα αύξηση του ρυθμού αποστολής κατά 10byte). Η τιμή αυτή υπόκειται σε μείωση από τους router που βρίσκονται στο μονόπατη μεταξύ αποστολέα και παραλήπτη, ούτως ώστε να συνάδει με την κατάσταση του δικτύου. Το πεδίο αυτό μπορεί να πάρει και αρνητικές τιμές, και το πεδίο τιμών του είναι από πλην 17Gbps μέχρι τα συν 17Gbps

- **Reverse Feedback**:
  - 32bits
  - Το πεδίο αυτό περιέχει την τιμή του delta_throughput κατά την παραλαβή του πακέτου από τον τελικό αποδέκτη. Πολύ απλά ο αποδέκτης του πακέτου τοποθετεί την τιμή του delta_throughput στο πακέτο ACK που θα αποστείλει πίσω στον αρχικό αποστολέα.

2.3 Μαθηματικοί υπολογισμοί

2.3.1 Delta throughput

Το delta throughput υπολογίζεται από τον αποστολέα σε κάθε πακέτο που θα αποσταλεί. Η εξίσωση που χρησιμοποιείται για τον υπολογισμό αυτού είναι η ποιο κάτω…

$$\Delta = \frac{C-(T*1000)}{T*\frac{RTT}{MSS}}$$

Επεξήγηση γραμμάτων όπως αυτά παρουσιάζονται στην ποιο πάνω εξίσωση.

Δ delta throughput, η τιμή αυτή μετριέται σε bytes/second

C Είναι το μέγιστη ταχύτητα με την οποία ο αποστολέας μπορεί να στείλει δεδομένα ή καλύτερα είναι η μέγιστη επιθυμητή ταχύτητα αποστολής δεδομένων του αποστολέα. Η τιμή αυτή μετριέται σε bytes/second

T Είναι ο ρυθμός αποστολής δεδομένων του sender, την στιγμή του υπολογισμού. Μετριέται σε bytes/millisecond.

RTT Είναι προφανώς ο χρόνος που χρειάζεται ένα πακέτο για να ταξιδέψει από τον αποστολέα στον παραλήπτη, μαζί με τον χρόνο που χρειάζεται το ACK για να επιστρέψει. Ο χρόνος αυτός μετριέται σε milliseconds.

MSS Χαρακτηρίζει το maximum segment size, και μετριέται σε bytes

2.3.2 Congestion window

- When a sender receives feedback on the throughput of the network he then has to adjust the congestion window. This is calculates by the following formula.

Όταν ο αποστολέας λάβει την ανατροφοδότηση από τον παραλήπτη του πακέτου που έστειλε πρέπει να ρυθμίσει ανάλογα το congestion window. Ο υπολογισμός που γίνεται για να αποφασιστεί το congestion window είναι ο εξής.

$$cwnd = max(cwnd + feedback * RTT * 1000, MSS)$$

cwnd Είναι το congestion window την παρούσα χρονική στιγμή και μετριέται σε bytes

feedback Είναι η τιμή του reverse_feedback όπως ο αποστολέας το παραλαμβάνει δια μέσου του ACK μηνύματος, η τιμή αυτή μετριέται σε bytes/seconds. Υπενθυμίζουμε ότι η τιμή αυτή μπορεί να είναι θετικός ή αρνητικός αριθμός.

RTT Είναι προφανώς ο χρόνος που χρειάζεται ένα πακέτο για να ταξιδέψει από τον αποστολέα στον παραλήπτη, μαζί με τον χρόνο που χρειάζεται το ACK για να επιστρέψει. Ο χρόνος αυτός μετριέται σε milliseconds.

MSS Χαρακτηρίζει το maximum segment size, και μετριέται σε bytes

Με την χρίση του max() πετύχουμε να έχουμε ως μικρότερη τιμή για το παράθυρο το MSS, με αυτό τον τρόπο αποφεύγουμε σενάρια όπου το παράθυρο θα μπορούσε να πάρει τιμές μικρότερες του ένα και συγκεκριμένα 0, σε τέτοια περίπτωση δεν θα είχαμε αποστολή δεδομένων αφού το congestion window θα ήταν πολύ μικρό για να επιτρέψει αποστολή πληροφοριών.

2.3.3. Efficiency Controller

Σκοπός του efficiency controller είναι να μεγιστοποιήσει το utilization των Links χωρίς όμως να προκαλεί απώλεια πακέτων. Η μαθηματική αυτή εξίσωση δεν σχετίζεται με την δικαιοσύνη αφού όπως εξηγείται στο 2.3.5 αυτή είναι δουλειά του fairness controller.

$$F = \alpha * avg(RTT) * \Delta C - \beta * Q$$

Όπου:

F Η συνολικό ανατροφοδότηση κατά το control interval, υπολογίζεται σε bytes/seconds.

α Μια σταθερά, η τιμή που έχει εισηγηθεί είναι 0.4

avg(RTT) Ο μέσος όρος των RTT κατά το τελευταίο control interval, η τιμή αυτή μετριέται σε seconds

ΔC Η τιμή αυτή μετριέται σε bytes/seconds. Και αντιπροσωπεύει την διαφορά μεταξύ της χωρητικότητας του link και των εισερχόμενων πληροφοριών του link. Εδώ αξίζει να σημειώσουμε ότι αν το link είναι υπερφορτωμένο η τιμή αυτή θα είναι αρνητική.

Β Μια σταθερά, η τιμή που έχει εισηγηθεί είναι 0.226

Q Είναι το persistent queue ή όπως περιγράφεται από την Katabi η μικρότερη δυνατή τιμή του queue se κάποιο time control interval (μικρότερο του RTT)

2.3.4. Queue estimation

Το μέγεθος queue έχει ως σκοπό να υποδηλώνει στο αλγόριθμο κατά το control interval το επίπεδο του congestion στο οποίο βρίσκεται ο router.

$$T = \max (Q, (avg(RTT) - iq / C) / 2)$$

Όπου:

T Το επόμενο timeout το οποίο μετριέται σε milliseconds.

Q Το queue σε milliseconds το οποίο μπορούμε να αποδεχτούμε, συνήθως μια τιμή της τάξης των 2ms είναι αποδεκτή, αλλά αυτό είναι κάτι που αποφασίζετε από τον δημιουργό.

avg(RTT) Το μέσο RTT όπως αυτό υπολογίστηκε στο προηγούμενο control interval, μετριέται σε milliseconds.

iq Το στιγμιαίο μέγεθος του queue μετρημένο σε bytes (instantaneous queue length)

C Η χωρητικότητα του εξερχόμενου link μετρημένη σε bytes/seconds

2.3.5. Άλλες απλές μαθηματικές εξίσωσης στους router

Πριν αναφερθούμε στις μαθηματικές αυτές εξισώσεις καλό είναι να δούμε σε ποια χρονική στιγμή γίνονται αυτές.
Για να λειτουργεί καλύτερα το XCP κατά συχνές περιόδους (control interval timeout) υπολογίζει κάποια στατιστικά στοιχεία.
Ο σκοπός τον υπολογισμών αυτόν είναι η διατήρηση της απόδοσης του δικτύου καθώς και η εξασφάλιση του δικαίου για όσους διαπερνούν τον εν λόγω router.
Ο λόγος για τον οποίο υπάρχει το interval timeout και δεν χρησιμοποιούμε την παραλαβή ενός πακέτου ως σήμανση για να γίνουν οι υπολογισμοί είναι γιατί υπάρχουν μερική πολύπλοκη υπολογισμοί που γίνονται σε αυτή την χρονική στιγμή, έτσι για να μην επηρεάζεται η απόδοση του συστήματος καθορίζεται το interval timeout ως ο χρόνος που γίνονται οι υπολογισμοί.

Όπως είδαμε ποιο πάνω το XCP διαχωρίζει το utilization control από τον έλεγχο fairness control, οι δύο αυτή υπολογισμοί εκτελούνται ξεχωριστά ο ένας από τον άλλο. Το γεγονός αυτό επιτρέπει όπως αυτή η δύο αλγόριθμοι αλλάξουν μελλοντικά δίνοντας καλύτερη απόδοση και δικαιοσύνη στο δίκτυο

Τα στατιστικά που κρατούνται με την άφιξη κάθε πακέτου είναι.

Sum (bytes received) = άθροισμα όλων τον IP packet size που παραλείφτηκαν.

swT = ps / xcpT

swRTT = RTT * ps / xcpT

Η επεξήγηση των παραπάνω συντομογραφιών

swT Sum of weighted throughput. Είναι το άθροισα του σταθμικού μέσου του throughput

swRTT Sum of weighted Round trip time. Είναι το άθροισμα του σταθμικού μέσου του round trip time

ps IP packet size. Το μέγεθος του IP πακέτου όπως αυτό προκύπτει από το πεδίο που βρίσκεται στο ίδιο το TCP header.

xcpT XCP throughput. Ισούται με το περιεχόμενο του πεδίου throughput στο header του XCP

RTT Είναι προφανώς ο χρόνος που χρειάζεται ένα πακέτο για να ταξιδέψει από τον αποστολέα στον παραλήπτη, μαζί με τον χρόνο που χρειάζεται το ACK για να επιστρέψει. Ο χρόνος αυτός μετριέται σε milliseconds.

2.4 Γενική αξιολόγηση – περίληψη

Το πρωτόκολλο XCP μας παρουσιάζει ένα νέο τρόπο διαχείρισης προβλημάτων που σχετίζονται με το congestion control και την δικαιοσύνη στο δίκτυο. Αυτό επιτυγχάνεται με την προσθήκη στα ήδη υφιστάμενα μοντέλο των στρωμάτων ενός ακόμη στρώματος το οποίο τοποθετείτε ενδιάμεσα στα TCP και IP. Το XCP υπόσχεται δικαιοσύνη στο δίκτυο, με μηδαμινές απώλειες πακέτων και μέγιστο utilization των γραμμών. Επίσης το XCP επιτρέπει στους router να κάνουν τροποποιήσεις στα header τον πακέτων που περνούν από αυτά ενημερώνοντας τον παραλήπτη για την κατάσταση του δικτύου στο οποίο ανήκει. Λόγω αυτού το XCP είναι πολύ δύσκολο να χρησιμοποιηθεί ως standard RFC και γι’ αυτό ο σκοπός της δημιουργού είναι να φτιάξει ένα πειραματικό RFC.

Το πρωτόκολλο XCP είναι εξορισμού διαφορετικό από το TCP/IP πρωτόκολλο έτσι η υλοποίηση του για επιβεβαίωση των όσων η συγγραφέας του αναφέρει είναι μια μεγάλη πρόκληση.

Κεφάλαιο 3.
3. Adaptive Congestion Protocol (ACP)

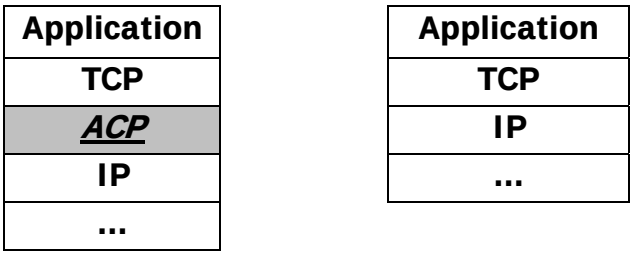
3.1 Εισαγωγή

Οι συγγραφής του ACP έχουν εισηγηθεί ένα αλγόριθμο ο οποίος προσπαθεί να υπερκεράσει το ήδη υπάρχον TCP congestion control αλλά και το νέο και πολλά υποσχόμενο XCP. Όπως έχουμε προαναφέρει το πρωτόκολλο που χρησιμοποιείται σήμερα (TCP)έχει πολλά προβλήματα ιδιαιτέρα όταν πρέπει να αντιμετωπίσει εφαρμογές της νέας γενεάς οι οποίες χρησιμοποιούν gigabyte συνδέσεις για να μεταφέρουν αρχεία, ασύρματα δίκτυα που όπως γνωρίζουμε έχουν πολλές απώλειες και συνδέσεις που χαρακτηρίζονται από μεγάλη καθυστέρηση (latency). Επίσης μέσα από αρκετές δημοσιεύσεις έχει αποδείξει ότι και το XCP πάσχει από αρκετά προβλήματα όπως είναι η αποτυχία του να πετύχει max-min δικαιοσύνη , ακόμη οδηγεί σε υπό χρησιμοποίηση των διαθέσιμων πόρων του δικτύου σε περιπτώσεις όπου έχουμε πολλαπλές congested γραμμές. Ο σκοπός δημιουργίας όπως η σχεδιαστές του το περιγράφουν είναι να προσφέρει την μέγιστη δυνατή απόδοση σε όλων των ειδών τα δίκτυα. Και να υπερκαλύψει όχι μόνο τα προβλήματα του TCP αλλά και αυτά του τόσα υποσχόμενου XCP.

Το ACP στηρίζει την γενική ίδιες του τρόπου λειτουργίας του στις ίδιες αρχές όπως και ο XCP δηλαδή μεταφέρει ένα per-flow congestion state στα πακέτα που αποστέλλει, με αυτό εννοούμε πως στο header του πακέτου υπάρχει ένα πεδίο το οποίο δηλώνει τον επιθυμητό ρυθμό αποστολής δεδομένων από τον χρήστη-αποστολέα. Αυτό το πεδίο σε συνδυασμό με το γεγονός ότι το πρωτόκολλο υποθέτει πως το δίκτυο απαρτίζεται από router οι οποίοι είναι σε θέσει να διαπιστώνουν την κατάσταση του δικτύου ενημερώνοντας την τιμή αυτή, έχουν την δυνατότητα να ενημερώνουν τον αποστολέα για την παρούσα κατάσταση του δικτύου επιτρέποντας του να ρυθμίσει δυναμικά τον ρυθμό αποστολής δεδομένων του. Με το να αφήνει το δίκτυο να του δίνει πληροφορίες για την κατάσταση στην οποία βρίσκεται το ACP έχει την δυνατότητα δυναμικής τροποποίησης του ούτως ώστε να ανταποκρίνεται στις όποιες αποτίσεις προκύπτουν.

Ο ACP όπως και ο XCP δεν έχει σκοπό να προσθέσει στα ήδη υπάρχοντα στρώματα δικτύου, αλλά αποσκοπεί στο να προσθέσει ένα επιπλέον επίπεδο που θα τοποθετηθεί ενδιάμεσα στα IP και TCP επίπεδα. Οι λόγοι για τους οποίους ακολουθείτε αυτή η μέθοδος

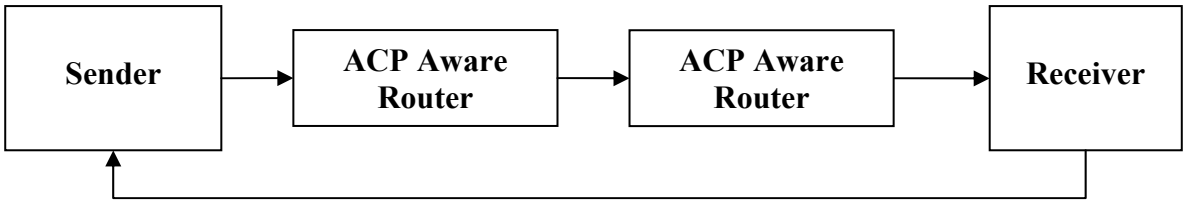
έχουν αναλυθεί εκτενώς στο κεφάλαιο 2, όπου γίνεται αναφορά για τον τρόπο λειτουργίας του XCP.



Προβλήματα fairness εμφανίζονται στο TCP, ιδιαίτερα σε περιπτώσεις οπού το RTT είναι μεγαλύτερο, αν σκεφτούμε ότι κάθε RTT έχουμε την αύξηση του ρυθμού αποστολής σε ένα host. Το XCP έχει διαπιστωθεί ότι αν και στης περισσότερες περιπτώσεις λειτουργεί ορθά και χωρίς να έχει προβλήματα δικαιοσύνης, μελέτες έχουν δείξει ότι ύστερη σε σενάρια όπου έχουμε πολλαπλά congested links, κάτι που το ACP αντιμετωπίζει με επιτυχία.

Όπως και στο XCP το ACP λειτουργεί με το να τοποθετεί επιπλέον πληροφορίες στο πακέτο που αποστέλλεται από τον αποστολέα. Η πληροφορίες που στέλνονται υπολογίζονται από τον αποστολέα και τροποποιούνται κατά την διαδρομή του πακέτου προς τον παραλήπτη από τους router, τέλος ο παραλήπτης επιστρέφει της πληροφορίες στον αρχικό αποστολέα στο πακέτο ACK. Η πληροφορίες που αποστέλλονται έχουν σχέση με τον υπολογισμό του RTT, την διαφορά στο throughput (η επιθυμητή αλλαγή στο παρών throughput) και το throughput κατά την στιγμή της αποστολής του πακέτου.

Όπως έχουμε προαναφέρει για να λειτουργήσει ο αλγόριθμος πρέπει να υπάρχει τουλάχιστο ένας router ο οποίος να είναι συμβατός με το πρωτόκολλο. Σε περίπτωση που κανένας router δεν υποστηρίζει το ACP τότε ο αποστολέας θα στείλει τα δεδομένα με την μέγιστη δυνατή ταχύτητα, κάτι που προφανώς θα είναι ανάλογο με το να έχουμε TCP χωρίς κανένα congestion control μηχανισμό (το ACP ακάθιστα το υφιστάμενο TCP congestion control αλγόριθμο). Για να λειτουργήσει με μέγιστη απόδοση ο ACP πρέπει όλοι οι router να είναι συμβατή με το πρωτόκολλο, σε περίπτωση που δεν είναι όλη τότε προφανώς το πρωτόκολλο θα δουλέψει αλλά όχι στον μέγιστο βαθμό.



Υπάρχουν πολλές σημαντικές αλλαγές στον ACP από τον XCP τις οποίες θα προσπαθήσουμε να απαριθμήσουμε και να κατονομάσουμε στην συνέχεια του εγγράφου. Κάποιες από τις αλλαγές βρίσκοντας της μαθηματικές πράξεις που εκτελούνται κατά την αποστολή παραλαβή και μετακίνηση τον πακέτων από τους αποστολής παραλήπτες και ενδιάμεσους κόμβους, άλλες διαφορές βρίσκονται στο header, όπως για παράδειγμα η τιμές που το απαρτίζουν , όπου στο ACP έχουμε ένα congestion bit του οποίου οι router θέτουν την τιμή ως ένα όταν φτάνουν σε επίπεδα utilization της τάξης του 95%.

3.2. Το Header

Όπως έχουμε προαναφέρει αυτό το πρωτόκολλο δεν διαφέρει ιδιαίτερα ως προς τις γενικές του αρχές σε σχέση με το XCP. Η γενική ιδέα είναι ότι η πληροφορίες που αφορούν το δίκτυο μεταφέρονται σε μορφή επιπλέον header μέσα στο δίκτυο με κάθε αποστολή πακέτου, και οι τιμές του header αλλάζουν κατά το πέρασμα τους από τους διάφορους router.
Έτσι το ACP χρησιμοποιεί ένα header στο οποίο έχει τις απαραίτητες τιμές- μεταβλητές οι οποίες χρειάζονται για να διασφαλίσει τον έλεγχο του congestion , το μέγιστο δυνατό utilization αλλά και την δικαιοσύνη σε αυτό.

Με ένα παρόμοιο ως προς το XCP πρωτόκολλο header το ACP αποτελείτε από τρία πεδία. Το πρώτο είναι το H_rtt το δεύτερο το H_feedback και τέλος το H_congestion.

The ACP Header

H_rtt (sender's RTT estimate)
H_feedback(desired sending rate)
H_congestion (congestion bit)

Τα πεδία του πακέτου επεξηγούνται αναλυτικά ποιο κάτω.

- **H_rtt:**
  - Το πεδίο αυτό μεταφέρει την πληροφορία για το μέγεθος του RTT(round trip time) όπως αυτή υπολογίστηκε από τον αρχικό αποστολέα. Η τιμή αυτή δεν επιδέχεται καμία αλλαγή, οι router που βρίσκονται στην διαδρομή του πακέτου προς τον παραλήπτη δεν αλλοιώνουν την τιμή αυτή, παρόλα αυτά διαβάζουν την τιμή για να υπολογίσουν το control period τους. Αν δεν μπορέσουμε να υπολογίσουμε το RTT τότε η τιμή αυτή θα είναι 0, αυτή είναι μια τυπική περίπτωση ειδικά όταν ασχολούμαστε με το πρώτο πακέτο που αποστέλλεται.

- **H feedback:**
  - Αυτό το πεδίο περιέχει τον ρυθμό αποστολής δεδομένων όπως αυτό υπολογίζεται από τον αρχικό αποστολέα. Η τιμή αρχικοποιείτε με τον επιθυμητό ρυθμό αποστολής δεδομένων και αλλοιώνεται κατά μήκος του ταξιδιού του πακέτου προς τον παραλήπτη από τους ενδιάμεσους router. Η τιμή αυτή μπορεί να συγκριθεί με αυτή του delta_throughput στο XCP. Σε κάθε κόμβο, η τιμή αυτή συγκρίνεται με τον επιθυμητό ρυθμό αποστολής και η μικρότερη τιμή αποθηκεύεται στο πεδίο. Έτσι επιτυγχάνουμε στο τέλος της ημέρας να έχουμε την μικρότερη τιμή που βρήκαμε στο μονοπάτι σε αυτό το πεδίο.

- **H congestion:**
  - Το πεδίο αυτό έχει μέγεθος ένα bit, και αρχικοποιείται από τον αποστολέα με την τιμή μηδέν. Η τιμή αυτή σκοπό έχει να μας ενημερώσει για πιθανό congestion που ίσως παρατηρηθεί στο δίκτυο. Δηλαδή αν το πακέτο περάσει από κάποιο router ο οποίος έχει 0.95 utilization. Τότε αυτός θα θέσει την τιμή αυτή 1. Έτσι όταν το ACK επιστρέψει ο αποστολέας θα γνωρίζει ότι κάποιο link στο δίκτυο βρίσκεται κοντά σε υπερφόρτωση, και έτσι θα εφαρμόσει μια άλλη τακτική αποφυγής congestion στο δίκτυο.

3.3. Μαθηματικοί υπολογισμοί

3.3.1. Sender

3.3.1.1. Αρχικοποίηση τιμών

Οι τιμές που αρχικοποιούνται κατά την αποστολή δεν είναι άλλες από αυτές του header. Κατά την πρώτη αποστολή δεν μπορέσουμε να εφαρμόσουμε τις μαθηματικές εξισώσεις που χαρακτηρίζουν το πρωτόκολλο. Αυτό συμβαίνει για το λόγω ότι το πρωτόκολλο βασίζεται στην αρχή ότι μαθαίνει και προσαρμόζεται στο δίκτυο αναλόγως με τις τιμές που παίρνει ως ανατροφοδότηση. Έτσι κάποιες τιμές πρέπει να αρχικοποιηθούν για να μπορέσει ο αλγόριθμος λα λειτουργήσει.

Κατά την αποστολή του πρώτου πακέτου η τιμή του H_rtt θέτεται από τον αποστολέα ως 0

$H\_rtt_1 = 0$

Το πεδίο h_feedback αρχικοποιείτε με τον επιθυμητό ρυθμό αποστολής δεδομένων

H_feedback = desired sending rate

Το H_congestion την πρώτη φορά καθώς και για κάθε αποστολή πακέτου θέτεται ως 0, ο λόγος για αυτό περιγράφεται στην ενότητα 3.2.

H_congestion = 0

Κατά την πρώτη αποστολή πακέτου το congestion window παίρνει την τιμή 1, μιας και αυτή είναι η μικρότερη επιτρεπτή τιμή που μπορεί να πάρει η μεταβολίτη αυτή. Ο λόγος είναι προφανώς το γεγονός ότι αν η τιμή είναι μικρότερη αυτής τότε ο αποστολέας δεν θα στέλνει πακέτα προς το δίκτυο.

3.3.1.2. Desired window

Ο υπολογισμός του επιθυμητού παραθύρου αποστολής γίνεται με την πιο κάτω μαθηματική εξίσωση.

$$\text{Desired window} = \frac{H\_feedback * mrtt}{size}$$

όπου:

H_feedback Η τελευταία τιμή H_feedback πού έχει παραληφθεί από τον αποστολέα μέσω ACK

mrtt Το μικρότερο RTT το οποίο έχει καταγραφεί μέχρι την συγκεκριμένη χρονική στιγμή.(milliseconds)

size Το μέγεθος του πακέτου σε bytes

3.3.1.3. Congestion Window

Αν το desired window όπως υπολογίζεται στο 3.3.1.2 , είναι μεγαλύτερο από το cwnd (**c**ongestion **wi**n**d**ow) της παρούσας χρονικής στιγμής και το πακέτο που έχουμε παραλάβει από το δίκτυο έχει την τιμή H_congestion = 1 (αυτό σημαίνει ότι κάποιο link έχει utilization της τάξης του 95%) .Ο υπολογισμός του congestion window έχει ως εξής.

cwnd = cwnd + (0.1/cwnd)*(desired window - cwnd)

Όπου:

cwnd Το congestion window

desired window Το desired window όπως αυτό υπολογίζεται στο 3.3.1.2

Σε περίπτωση τώρα που δεν ισχύει η ποιο πάνω παράγραφος δηλαδή ότι το desired window είναι μεγαλύτερο του congestion window και ότι το H_congestion είναι ένα, τότε εφαρμόζεται ο ακόλουθος τύπος.

cwnd = max((cwnd + (1/cwnd)* (desired window – cwnd) , 1)

Όπου:

cwnd	Το congestion window
desired window	Το desired window όπως αυτό υπολογίζεται στο 3.3.1.2

3.3.2. Routers

3.3.2.1. Υπολογισμός ρυθμού αποστολής δεδομένων

Το ACP υπολογίζει τον ρυθμό αποστολής δεδομένων. Ο υπολογισμός αυτός γίνεται με την πιο κάτω μαθηματική πράξη. Η αρχική τιμή του p(0) είναι ίση με 0.

$$p(k+1) = Pr[p(k) + \frac{1}{\hat{N}(k)}[k_i(0.99 * C - y(k)) - \frac{1}{d(k)}k_q q(k)]], \quad p(0) = 0$$

$$Pr[x] = \begin{cases} 0 & \text{if } x < 0 \\ C & \text{if } x > C \\ x & \text{otherwise} \end{cases}$$

Όπου:

p Είναι ο επιθυμητός ρυθμός αποστολής δεδομένων για όλους τους χρήστες που διαπερνούν τον router

K_i Το K_i είναι μία σταθερά οι οποία για την παρούσα έκδοση του πρωτοκόλλου είναι 0.1587

K_q Το K_q είναι μία σταθερά οι οποία για την παρούσα έκδοση του πρωτοκόλλου είναι 0.3175

C Είναι το συνολικό link capacity

y Είναι ο ρυθμός με τον οποίο έρχονται τα δεδομένα

q Είναι το μικρότερο queue size που έχει φτάσει στον router.

d Είναι ο μέσος όρος των round trip time

N(k) Είναι ο υπολογισμός του αριθμού ροών όπως αυτό επεξηγείτε στο 3.3.2.2

3.3.2.2. Υπολογισμός αριθμού ροών

Σε αυτό το κομμάτι της διπλωματικής θα δούμε πως γίνεται ο υπολογισμός του αριθμού ροών στο router. Ας σημειώσουμε ότι η αρχική τιμή του N(0) είναι ίση με 10.

$$\hat{N}(k+1) = Pr[\hat{N}(k) + \frac{\gamma[y(k) - \hat{N}(k)p(k)]p(k)}{1+p^2(k)}], \quad \hat{N}(0) = 10$$

$$Pr[x] = \begin{cases} x & \text{if } x > 1 \\ 1 & \text{otherwise} \end{cases}$$

p Είναι ο επιθυμητός ρυθμός αποστολής δεδομένων για όλους τους χρήστες που διαπερνούν τον router. Ο υπολογισμός περιγράφεται στο 3.3.2.1

γ Το γ είναι μία σταθερά οι οποία για την παρούσα έκδοση του πρωτοκόλλου είναι ίση με 0.1

3.3.2.3. Άλλες απλές μαθηματικές εξίσωσης στους router

Εδώ θα πρέπει να αναφέρουμε ότι το H_rtt έχει αρχική τιμή 0.05 στους router.
Ακόμη θα πρέπει να αναφέρουμε ότι για σκοπούς υλοποίησης χρησιμοποιούμε αρκετές μαθηματικές πράξεις η οποίες αναφέρονται στο πρωτόκολλο XCP

3.3.3. Receiver

Όταν ο αποδέκτης λάβει ένα μήνυμα ACP αντιδρά όπως ακριβώς και στο XCP, δηλαδή απλά αντιγραφή την τιμή του H_feedback και H_congestion στα αντίστοιχα πεδία του ACK μηνύματος και τα στέλνει πίσω στον αρχικό αποστολέα. Εδώ αξίζει να σημειώσουμε ότι όταν έχουμε ACK μήνυμα οι router που βρίσκονται στην διαδρομή δεν πρέπει να αλλοιώσουν στα πεδία που περιέχουν της πληροφορίες ανατροφοδότηση. Αυτό στο XCP επιτυγχάνεται με το να θέτουμε την τιμή του format σε 0x02 (για ACK messages, 0x01 για μηνύματα αποστολής δεδομένων) .

3.3.4. Περίληψη

Το πρωτόκολλο ACP βασισμένο στης προφανώς πολλά υποσχόμενες αρχές του XCP έρχεται να βελτιώσει την απόδοση του XCP. Το ACP προσπαθεί να επιλύσει τα προβλήματα που παρατηρήθηκαν κατά τις δοκιμές που έγιναν στον XCP. Ποιο αναλυτικά το πρωτόκολλο υπόσχεται max-min δικαιοσύνη, high utilization , μικρά queue sizes και πολύ μικρή απώλεια πακέτων.

Όλα αυτά που υπόσχεται το ACP μας έχουν κάνει να ενδιαφερθούμε περεταίρω για αυτό. Έτσι αποφασίσαμε να δημιουργήσουμε μια Linux έκδοση του αλγορίθμου για να δοκιμάσουμε στην πράξει τα όσα αναφέρονται στην δημοσίευση του πρωτοκόλλου αυτού.

Η υλοποίηση μας έχει σκοπό να αποφανθεί κατά πόσο τα όσα αναφέρονται είναι ακριβή. Επίσης στα πλαίσια της υλοποίησης θα μας δοθεί η ευκαιρία να καταγράψουμε όλες τις λεπτομέρειες που αφορούν το πρωτόκολλο αυτό. Ακόμη δεδομένης της υλοποίησης θα μπορούν να γίνουν ποιο εντατικές δοκιμές αναφορικά με τον αλγόριθμο, δοκιμές που ίσως οδηγήσουν σε βελτίωση κάποιων κομματιών του.

Κεφάλαιο 4.
4. Εφαρμογή του ACP σε λειτουργικό σύστημα Linux
4.1. Υλοποίηση ACP ως ξεχωριστό layer
4.2. Υλοποίηση host
4.2.1. Sender
4.2.2. Receiver
4.3. Υλοποίηση router ACP
4.3.1. Παραλαβή πακέτου
4.3.2. Υλοποίηση υπολογισμών
4.3.3. Αποστολή πακέτου
4.4. Περίληψη

Κεφάλαιο 5.

5. Αποτελέσματα ελέγχου απόδοσης ACP
5.1…
5.2.
5.3.

Κεφάλαιο 6.
6. Συμπεράσματα και μελλοντική εργασία

Κεφάλαιο 7.
7. Βιβλιογραφία