

14 Γιατί το Διαδίκτυο δεν καταρρέει λόγω συμφόρησης;

14.1 Μία σύντομη απάντηση

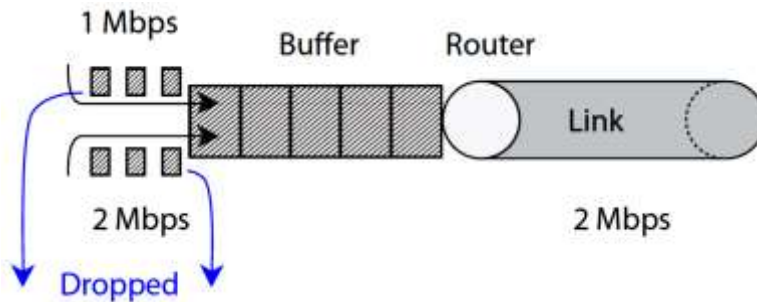
14.1.1 Αρχές του καταναμημένου ελέγχου συμφόρησης

Όταν η ζήτηση υπερβαίνει την προσφορά, έχουμε συμφόρηση. Αν η προσφορά είναι σταθερή, πρέπει να μειώσουμε τη ζήτηση για να μετριάσουμε τη συμφόρηση. Αν υποθέσουμε ότι η ζήτηση έρχεται από διάφορους κόμβους σε ένα δίκτυο, πρέπει να τη συντονίσουμε με έναν καταναμημένο τρόπο.

Καθώς η ζήτηση για χωρητικότητα στο Διαδίκτυο υπερβαίνει την προσφορά κάθε τόσο, ο έλεγχος της συμφόρησης γίνεται ουσιαστικός. Το χρονοδιάγραμμα του ελέγχου συμφόρησης είναι της τάξης των ms, σε αντίθεση με τη διαμόρφωση της συμπεριφοράς των πελατών με χρήση μεθόδων τιμολόγησης των Κεφαλαίων 11 και 12. Η ανάγκη για έλεγχο συμφόρησης έγινε αντιληπτή τον Οκτώβριο του 1986, όταν το Διαδίκτυο είχε την πρώτη του κατάρρευση από συμφόρηση. Έλαβε χώρα σε μία μικρή σύνδεση τριών βημάτων ανάμεσα στο Lawrence Berkeley Lab και στο UC Berkeley. Ο κανονικός ρυθμός μεταβίβασης δεδομένων ήταν 32 kbps (σωστά, kbps, όχι τα Mbps που ακούμε σήμερα). Αυτού του είδους η ταχύτητα του διεπιλογικού διαπομορφωτή (dial-up μόντεμ) ήταν ήδη αρκετά μικρή, αλλά κατά τη διάρκεια του γεγονότος της συμφόρησης έπεσε στα 40 bps, κατά ένα συντελεστή της τάξεως του 1000.

Ο κύριος λόγος ήταν ξεκάθαρος, όπως είδαμε στο τελευταίο κεφάλαιο στη δρομολόγηση: όταν οι χρήστες στέλνουν πάρα πολλά bit ανά δευτερόλεπτο, που το συνολικό τους φορτίο σε μία σύνδεση υπερβαίνει τη χωρητικότητα αυτής της σύνδεσης, αυτά τα πακέτα αποθηκεύονται σε μία ενδιάμεση μνήμη/ενταμιευτή (buffer) και περιμένουν στην ουρά για να μεταδοθούν. Αλλά όταν αυτή η αναμονή διαρκεί πολύ, συσσωρεύονται περισσότερα εισερχόμενα πακέτα στην ενδιάμεση μνήμη, μέχρι αυτή να υπερχειλίσει και να απορριφθούν τα πακέτα. Αυτό απεικονίζεται στο Σχήμα 14.1.

Αυτά τα απορριφθέντα πακέτα δεν φτάνουν ποτέ στον προορισμό τους, κι έτσι ο προοριζόμενος παραλήπτης δεν στέλνει ποτέ μία επιβεβαίωση (ένα πακέτο ACK) πίσω στον αποστολέα, όπως έπρεπε να κάνει στον προσανατολισμένο σε σύνδεση, από-άκρο-προς-άκρο έλεγχο στο TCP. Όπως αναφέρθηκε στο τελευταίο κεφάλαιο, η εξέλιξη του σχεδιασμού του Διαδικτύου ανάθεσε διαφορετικές εργασίες στα επίπεδα 3 και 4, καταλήγοντας τελικά σε ένα προσανατολισμένο προς τη σύνδεση επίπεδο 4 και σε ένα χωρίς σύνδεση επίπεδο 3 ως τυπική δομή. Σύμφωνα με το TCP, ο αποστολέας πρέπει να ξαναστείλει τα μη επιβεβαιωμένα πακέτα. Αυτό οδηγεί σε ένα φαύλο κύκλο, ένα βρόχο θετικής-ανάδρασης που τροφοδοτεί τον εαυτό του: η συμφόρηση παραμένει, καθώς το ίδιο σύνολο αποστολέων που προκάλεσαν αρχικά τη συμφόρηση συνεχίζουν να στέλνουν τα απορριπτέα πακέτα. Τα πακέτα συνεχίζουν να απορρίπτονται στη συνωστισμένη σύνδεση, ξαστελνούνται από την πηγή, απορρίπτονται από τη συνωστισμένη σύνδεση... Οι αποστολείς πρέπει να ξασκεφτούν πώς μπορούν να αποφύγουν τη συμφόρηση αρχικά και πρέπει να κάνουν πίσω όταν συμβαίνει συμφόρηση. Πρέπει να αλλάξουμε το βρόχο θετικής-ανάδρασης σε βρόχο αρνητικής ανάδρασης.



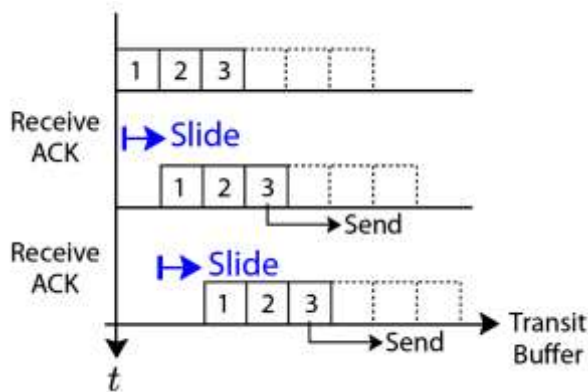
Σχήμα 14.1 Μία απεικόνιση της συμφόρησης στο άκρο μίας σύνδεσης. Δύο σύνοδοι φτάνουν στην ενδιάμεση μνήμη με συνολική ζήτηση 3 Mbps, αλλά υπάρχει προσφορά 2 Mbps στην εξερχόμενη σύνδεση. Η ενδιάμεση μνήμη γεμίζει και τα πακέτα αρχίζουν να απορρίπτονται. Το ποια πακέτα απορρίπτονται εξαρτάται από τις λεπτομέρειες των πρωτόκολλων διαχείρισης ουράς.

Αυτό ήταν που πρότεινε ο Van Jacobson στον πρώτο μηχανισμό ελέγχου συμφόρησης που προστέθηκε στο TCP το 1998 και ονομαζόταν TCP Tahoe. Από τότε μελετήθηκε εκτενώς και βελτιώθηκε σημαντικά αρκετές φορές. Αλλά οι περισσότερες από τις βασικές ιδέες του ελέγχου συμφόρησης για το Διαδίκτυο ήταν ήδη στο TCP Tahoe.

- Από-άκρο-προς-άκρο έλεγχος μέσω αρνητικής ανάδρασης. Μπορούμε να φανταστούμε τον έλεγχο συμφόρησης μέσα σε ένα δίκτυο, όπου, βήμα προς βήμα, οι δρομολογητές αποφασίζουν με τι ρυθμό θα πρέπει να αποστέλλουν τα πακέτα τους στους τελικούς κόμβους (end hosts). Αυτό στην πραγματικότητα είναι κάτι που κάποιο άλλο πρωτόκολλο, που ονομάζεται ATM

(Asynchronous Transmission Mode), κάνει σε έναν τύπο της κυκλοφορίας του, στην κυκλοφορία τυχαίου ρυθμού δεδομένων (arbitrary bit rate). Αλλά ο έλεγχος συμφόρησης του TCP υιοθετεί μία εναλλακτική προσέγγιση, να έχει ένα ευφυές δίκτυο άκρων και ένα χαζό δίκτυο κορμού. Ο ρυθμός με τον οποίο ο αποστολέας στέλνει πακέτα αποφασίζεται από τον ίδιο τον αποστολέα. Αλλά το δίκτυο παρέχει υποδείξεις μέσω κάποιων πληροφοριών ανάδρασης στους αποστολείς. Τέτοιες πληροφορίες ανάδρασης μπορεί να συμπεραίνονται από την παρουσία και το συγχρονισμό των πακέτων επιβεβαίωσης, που μεταδίδονται από τον παραλήπτη πίσω στον αποστολέα, επιβεβαιώνοντας την παραλαβή κάθε πακέτου.

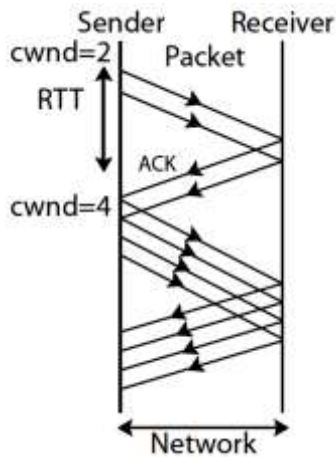
- Έλεγχος βάσει συρόμενου/ολισθαίνοντος παραθύρου. Το να πρέπει ο αποστολέας να περιμένει για την επιβεβαίωση ενός απεσταλμένου πακέτου, πριν του επιτραπεί να στείλει ένα άλλο πακέτο, δημιουργεί καθυστερήσεις. Οπότε προβαίνουμε σε σύνδεση με την προϋπόθεση περισσότερων ανοχών. Κάθε αποστολέας διατηρεί ένα συρόμενο παράθυρο, που ονομάζεται παράθυρο συμφόρησης, με το μέγεθός του να συμβολίζεται με $cwnd$ (congestion window). Αν το μέγεθος του παραθύρου είναι 5, αυτό σημαίνει ότι μπορεί να σταλούν μέχρι πέντε πακέτα, πριν να πρέπει ο αποστολέας να σταματήσει και να περιμένει να επιστρέψουν πακέτα επιβεβαίωσης από τον παραλήπτη. Για κάθε νέο πακέτο επιβεβαίωσης που παραλαμβάνεται από τον αποστολέα, το παράθυρο σύρεται κατά ένα πακέτο εμπρός και αυτό επιτρέπει την αποστολή ενός νέου πακέτου, ως εκ τούτου και το όνομα συρόμενο παράθυρο. Αυτός ο τρόπος της εφαρμογής ενός περιορισμού στο ρυθμό μετάδοσης εισάγει το καλούμενο ως ιδιότητα αυτοχρονισμού, που οδηγείται από τα πακέτα επιβεβαίωσης. Ένα σχήμα που απεικονίζει τη λειτουργία του συρόμενου παραθύρου φαίνεται στο Σχήμα 14.2.



Σχήμα 14.2 Μία απεικόνιση ενός συρόμενου παραθύρου με σταθερό μέγεθος τρία. Όταν εκκρεμούν, δηλαδή δεν έχουν επιβεβαιωθεί, τρία πακέτα, η μετάδοση πρέπει να

σταματήσει. Καθώς παραλαμβάνεται κάθε μία επιβεβαίωση, το παράθυρο σύρεται κατά ένα πακέτο, επιτρέποντας σε ένα νέο πακέτο να μεταδοθεί.

- Προσθετική αύξηση και πολλαπλασιαστική μείωση. Δεν θα έχουμε το χρόνο να συζητήσουμε τις λεπτομέρειες του πώς αρχικοποιείται η τιμή του *cwnd*, όταν δημιουργείται μία νέα σύνδεση TCP, κατά τη διάρκεια της καλούμενης φάσης καθυστερημένης εκκίνησης. Αντί αυτού, εστιάζουμε στη φάση αποφυγής συμφόρησης. Αν δεν υπάρχει συμφόρηση, το *cwnd* επιτρέπεται να μεγαλώνει, για να αξιοποιεί αποδοτικά τις χωρητικότητες των συνδέσεων. Η αύξηση της τιμής του *cwnd* είναι διαφορετικό γεγονός από το σύρσιμο του παραθύρου με μία δεδομένη τιμή του *cwnd*: το *cwnd* μεγαλώνει σε αντίθεση με το να σύρεται εμπρός. Και στο TCP, όταν το *cwnd* μεγαλώνει, μεγαλώνει γραμμικά: το *cwnd* μεγαλώνει κατά $1/cwnd$ μετά την παραλαβή κάθε επιβεβαίωσης. Αυτό σημαίνει ότι μετά από ένα χρονικό διάστημα πλήρους διαδρομής (round-trip time - RTT), το *cwnd* μεγαλώνει κατά 1, αν παρελήφθησαν κανονικά όλες οι επιβεβαιώσεις (ACK). Αυτή η λειτουργία φαίνεται στο διάγραμμα χώρου-χρόνου στο Σχήμα 14.3. Αλλά εάν υπάρχει συμφόρηση, το *cwnd* πρέπει να μειωθεί, για να μετριαστεί η συμφόρηση. Και το TCP λέει ότι όταν το *cwnd* μειώνεται, μειώνεται πολλαπλασιαστικά: το *cwnd* την επόμενη φορά θα είναι, ας πούμε, το μισό της τρέχουσας τιμής του. Η αύξηση του *cwnd* αθροιστικά και η μείωσή του πολλαπλασιαστικά σημαίνει ότι ο έλεγχος της ροής των πακέτων στο δίκτυο είναι συντηρητικός (ελεγχόμενος). Θα ήταν πολύ πιο επιθετικός αν συνέβαινε το αντίθετο: πολλαπλασιαστική αύξηση και προσθετική μείωση.
- Εξαγωγή συμπεράσματος συμφόρησης λόγω απώλειας πακέτων ή καθυστέρησης. Αλλά πώς γνωρίζουμε ότι υπάρχει συμφόρηση; Αν είστε ένα iPhone που τρέχετε μία σύνδεση TCP, δεν έχετε πραγματικά ιδέα πώς είναι η τοπολογία του δικτύου, ποιες διαδρομές παίρνουν τα πακέτα σας, ποιοι είναι οι άλλοι τελικοί κόμβοι που μοιράζονται συνδέσεις μαζί σας και ποιες συνδέσεις στη διαδρομή είναι συνωστισμένες. Έχετε μόνο μία τοπική και νεφελώδη άποψη και πρέπει κιόλας να κάνετε μία εικασία: αντιμετωπίζει η σύνδεσή σας κάπου συμφόρηση ή όχι; Οι αρχικές εκδόσεις του ελέγχου συμφόρησης του TCP έκαναν μία σημαντική υπόθεση: αν υπάρχει απώλεια πακέτων, υπάρχει συμφόρηση. Αυτό ακούγεται αρκετά λογικό, αλλά μερικές φορές η απώλεια πακέτων οφείλεται σε ένα κακό κανάλι, όπως στις ασύρματες συνδέσεις και όχι στη συμφόρηση. Επιπρόσθετα, συχνά είναι πολύ αργά να αντιδράσουμε στη συμφόρηση, όταν τα πακέτα ήδη απορρίπτονται. Το πρώτο πρόβλημα αντιμετωπίστηκε με πολλές προτάσεις για το TCP για ασύρματα δίκτυα. Το δεύτερο πρόβλημα λύθηκε κατά ένα μεγάλο μέρος με τη χρήση της καθυστέρησης των πακέτων ως ένα σήμα ανάδρασης συμφόρησης. Αντί για ένα δυαδικό ορισμό της ύπαρξης ή όχι συμφόρησης, μία ορισμένη τιμή της καθυστέρησης υπονοεί το βαθμό της συμφόρησης.



Σχήμα 14.3 Ένα διάγραμμα χώρου-χρόνου των πακέτων TCP που στέλνονται και επιβεβαιώνονται. Η οριζόντια απόσταση μεταξύ δύο κάθετων γραμμών αντιπροσωπεύει τη χωρική απόσταση μεταξύ του αποστολέα και του παραλήπτη. Ο κάθετος άξονας αντιπροσωπεύει το χρόνο. Καθώς δύο επιβεβαιώσεις παραλαμβάνονται από τον αποστολέα, το παράθυρο συμφόρησης δεν σύρεται μόνο, αλλά, επίσης, και αυξάνει κατά 1.

- Εκτίμηση απώλειας πακέτων και καθυστέρησης από χρονοδιακόπτες. Ας πούμε ότι συμφωνείτε ότι η απώλεια πακέτων ή η καθυστέρηση υπονοεί συμφόρηση, πώς μπορείτε να πείτε εάν ένα πακέτο έχει χαθεί και πώς υπολογίζετε την καθυστέρηση; Το TCP χρησιμοποιεί δύο προσεγγίσεις κοινής λογικής: (1) Αν ο αποστολέας περιμένει αρκετό χρόνο και δεν έρχεται επιβεβαίωση, πιθανόν το πακέτο να έχει χαθεί. Πόσος είναι ο «αρκετός χρόνος»; Ας πούμε ότι αυτός ο χρονοδιακόπτης είναι τρεις φορές ο κανονικός χρόνος πλήρους διαδρομής ανάμεσα στον αποστολέα και στον παραλήπτη. Και ποιος είναι ο κανονικός RTT; Ο αποστολέας τοποθετεί μία χρονοσήμανση σε κάθε πακέτο και μπορεί να πει ποιος είναι ο RTT αυτού του πακέτου, μόλις παραληφθεί αργότερα η επιβεβαίωση. Αυτός είναι ο τρόπος που ο αποστολέας υπολογίζει την καθυστέρηση κάθε πακέτου. Τότε μπορεί να υπολογίσει έναν κυλιόμενο μέσο όρο του RTT. Ο μικρότερος RTT σε μία περίοδο του χρόνου είναι κατά προσέγγιση ο «κανονικός», χωρίς συμφόρηση RTT. (2) Κάθε πακέτο που αποστέλλεται έχει έναν αριθμό αλληλουχίας και αν ο αποστολέας ειδοποιηθεί από τον παραλήπτη ότι έχουν παραληφθεί ορισμένα, ας πούμε τρία, μεταγενέστερα πακέτα (με νούμερα 10, 11 και 12), αλλά αυτό το συγκεκριμένο πακέτο 9 δεν έχει παραληφθεί ακόμη, αυτό σημαίνει ότι το πακέτο 9 ενδεχομένως έχει χαθεί. Το πακέτο 9 μπορεί να ακολούθησε διαφορετική διαδρομή με μεγαλύτερο RTT (όπως

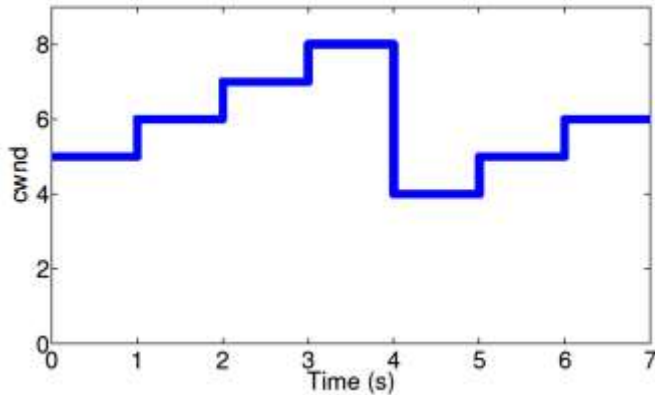
είδαμε στη δρομολόγηση IP στο τελευταίο κεφάλαιο), αλλά αν τρία πακέτα έχουν ήδη παραληφθεί, δεν είναι απλά καθυστερημένο, αλλά απολεσθέν.

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, το TCP/IP αποτελεί τη «λεπτή μέση» της στοιβάδας των επιπέδων των πρωτόκολλων του Διαδικτύου. Κολλάει τις λειτουργικές μονάδες από κάτω του, όπως το φυσικό επίπεδο και το επίπεδο σύνδεσης, με αυτά από πάνω του, όπως το επίπεδο εφαρμογών. (Υπάρχουν εναλλακτικές του TCP σε αυτή τη λεπτή μέση, όπως το χωρίς σύνδεση UDP, το οποίο δεν διατηρεί ένα από-άκρο-προς-άκρο έλεγχο ανάδρασης, όπως θα δούμε στο Κεφάλαιο 17). Σαν κομμάτι αυτής της λεπτής μέσης, τα παραπάνω πέντε στοιχεία του ελέγχου συμμόρφωσης, που είναι σχεδιασμένα στο TCP, οδήγησαν σε μεγάλη επιτυχία. Το υπέροχο γεγονός ότι το Διαδίκτυο δεν έχει καταρρεύσει, παρ' όλη την φοβερή και ασταμάτητη εκτόξευση της ζήτησης, μπορεί εν μέρει να αποδοθεί στις δυνατότητες του ελέγχου συμμόρφωσης.

Ξεκινώντας με το TCP Tahoe το 1988 και ακολουθώντας με μία ελαφρώς διαφορετική παραλλαγή του, το TCP Reno το 1990, ο έλεγχος συμμόρφωσης του TCP έχει περάσει πάνω από είκοσι χρόνια συνεχών βελτιώσεων. Για παράδειγμα, το TCP Vegas το 1995 άλλαξε από τα σήματα συμμόρφωσης βασισμένα στην απώλεια σε σήματα συμμόρφωσης βασισμένα στην καθυστέρηση. Το FAST TCP το 2002 σταθεροποίησε τον έλεγχο συμμόρφωσης για να επιτύχει υψηλή αξιοποίηση της χωρητικότητας της σύνδεσης. Το CUBIC το 2005 συνδύασε σήματα συμμόρφωσης με βάση την απώλεια με άλλα σήματα βάσει της καθυστέρησης και τώρα είναι το προεπιλεγμένο TCP στον πυρήνα του Linux. Υπάρχουν επίσης πολλές άλλες παραλλαγές του ελέγχου συμμόρφωσης του TCP, που προτάθηκαν τις τελευταίες δύο δεκαετίες.

14.1.2 Εξαγωγή συμπεράσματος συμμόρφωσης βάσει απώλειας

Αν το σκεφτείτε, για ένα από άκρο-σε-άκρο έλεγχο συμμόρφωσης χωρίς τη διέλευση κάποιου μηνύματος από το δίκτυο, ένας τελικός κόμβος (όπως το iPad σας) πραγματικά έχει πολύ λίγα δεδομένα με τα οποία μπορεί να δουλέψει. Οι εκτιμήσεις της απώλειας πακέτων και οι υπολογισμοί της καθυστέρησης πακέτων είναι δύο πληροφορίες που μπορεί να αποκτήσει μέσω της χρονοσήμανσης και της αρίθμησης των πακέτων.



Σχήμα 14.4 Μεγεθυμένη προβολή της εξέλιξης του $cwnd$ για το TCP Reno με $RTT=1$ μονάδα χρόνου.

Για τον έλεγχο συμφόρησης βάσει απώλειας, όπως στο TCP Reno, μίας κύριας παραλλαγής του TCP ειδικά για το λειτουργικό σύστημα Windows, οι κύριες λειτουργίες είναι οι παρακάτω:

- Αν όλα τα εκκρεμή πακέτα $cwnd$ παραλαμβάνονται κανονικά από τον παραλήπτη (δηλαδή στην ώρα τους και όχι πάνω από δύο φορές με λάθος σειρά), αύξησε το $cwnd$ κατά 1 για κάθε RTT.
- Αλλιώς, μείωσε το κόβοντάς το στη μέση, π.χ. από $cwnd$ σε $0,5 \times cwnd$. Υπάρχουν επίσης και άλλα λεπτά χαρακτηριστικά, όπως η Γρήγορη Αναμετάδοση και η Γρήγορη Επαναφορά, που δεν θα έχουμε το χρόνο να δούμε.

Ας δούμε ένα παράδειγμα. Για απλότητα, ας πούμε ότι το $RTT=1$ μονάδα και υποθέτουμε ότι είναι μία σταθερά. Στην πραγματικότητα, το RTT είναι περίπου 50 ms κατά μήκος των ΗΠΑ και αλλάζει όταν οι συνθήκες συμφόρησης αλλάζουν. Αρχικοποιούμε το $cwnd$ να είναι 5. Υποθέστε ότι όλα τα πακέτα παραλαμβάνονται επιτυχώς και επιβεβαιώνονται (ACK) κατά τη διάρκεια κάθε RTT, εκτός στο $t = 4$, οπότε συμβαίνει μία απώλεια πακέτου.

Στο $t = 0$, $cwnd = 5$, έτσι ο αποστολέας στέλνει πέντε πακέτα και σταματάει.

Στο $t = 1$, ο αποστολέας έχει λάβει πέντε ACK, και έτσι σύρει το παράθυρο συμφόρησης κατά πέντε πακέτα και αυξάνει το $cwnd$ κατά 1. Στέλνει έξι πακέτα.

Στο $t = 2$, ο αποστολέας έχει λάβει έξι ACK, και έτσι στέλνει επτά πακέτα.

Στο $t = 3$, ο αποστολέας έχει λάβει επτά ACK, και έτσι στέλνει οκτώ πακέτα.

Στο $t = 4$, ο αποστολέας εντοπίζει μία απώλεια πακέτου. Διαιρεί στα δύο το $cwnd$, στο τέσσερα, και στέλνει τέσσερα πακέτα.

Στο $t = 5$, ο αποστολέας έχει λάβει τέσσερα ACK, και έτσι στέλνει πέντε πακέτα.

Στο $t = 6$, ο αποστολέας έχει λάβει πέντε ACK, και έτσι στέλνει έξι πακέτα.

Το Σχήμα 14.4 δείχνει αυτές τις τιμές του $cwnd$ στο χρόνο. Όταν δεν υπάρχει απώλεια πακέτου (στις χρονικές στιγμές $t = 0, 1, 2, 3$), το $cwnd$ αυξάνεται γραμμικά. Όταν υπάρχει απώλεια πακέτου (στη χρονική στιγμή $t = 4$), το $cwnd$ μειώνεται απότομα, μετά ξεκινά πάλι να αυξάνεται γραμμικά (στη χρονική στιγμή $t = 5, 6$).

Το Σχήμα 14.5(α) δείχνει μία τυπική εξέλιξη του $cwnd$ του TCP Reno κατά τη διάρκεια του χρόνου. Στον άξονα y παρατίθεται το μέγεθος του παράθυρου συμφόρησης. Αν το διαιρέσετε με το RTT και το πολλαπλασιάσετε με το μέσο μέγεθος πακέτου, θα λάβετε τον πραγματικό ρυθμό μετάδοσης σε bps (bits per second).

14.1.3 Εξαγωγή συμπεράσματος συμφόρησης βάσει καθυστέρησης

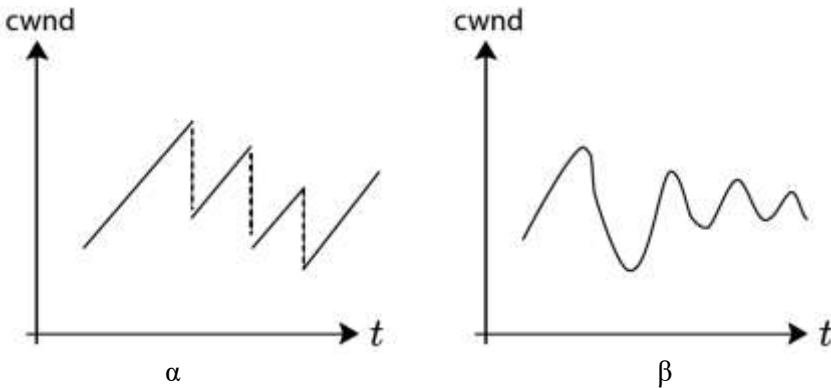
Τώρα θα στραφούμε στον έλεγχο συμφόρησης βάσει καθυστέρησης, όπως γίνεται στο TCP Vegas. Πρώτα πρέπει να αναγνωρίσουμε ότι το συνολικό RTT αποτελείται κυρίως από την καθυστέρηση διάδοσης, ο χρόνος που χρειάζεται για να περάσουν τα πακέτα μέσα από τις συνδέσεις, και την καθυστέρηση ουράς αναμονής, ο χρόνος που ένα πακέτο περνάει στην ουρά λόγω συμφόρησης. Όσο πιο μεγαλύτερη είναι η συμφόρηση, τόσο μεγαλύτερη είναι η αναμονή. Έτσι ο αποστολέας πρέπει να εκτιμήσει το RTT_{min} , το ελάχιστο RTT που λέει στον αποστολέα ποια θα είναι η τιμή της καθυστέρησης εάν (σχεδόν) δεν υπάρχει συμφόρηση.

Μετά, αφού λάβει κάθε μία επιβεβαίωση, ο αποστολέας κοιτάει τη διαφορά μεταξύ των $cwnd/RTT_{min}$ και $cwnd/RTT_{now}$. Είναι η διαφορά ανάμεσα στο ρυθμό μετάδοσης (σε πακέτα ανά δευτερόλεπτο) χωρίς μεγάλη καθυστέρηση συμφόρησης και σε αυτόν με την τρέχουσα καθυστέρηση συμφόρησης.

- Αν αυτή η διαφορά είναι μικρότερη από ένα προκαθορισμένο κατώφλι, ας πούμε 3, αυτό σημαίνει ότι υπάρχει μικρή συμφόρηση και το $cwnd$ αυξάνεται κατά 1.
- Αν η διαφορά είναι μεγαλύτερη από το όριο, αυτό σημαίνει ότι υπάρχει κάποια συμφόρηση και το $cwnd$ μειώνεται κατά 1.
- Αν η διαφορά είναι ακριβώς ίση με το όριο, το $cwnd$ παραμένει το ίδιο.
- Αν όλες οι πηγές σταματήσουν να προσαρμόζουν το $cwnd$ τους, έχει επιτευχθεί ισορροπία.

Μπορούμε να συγκρίνουμε αυτό τον έλεγχο συμφόρησης με τον έλεγχο ισχύος στο Κεφάλαιο 1, όπου σε κατάσταση ισορροπίας όλοι σταματούν να αλλάζουν τη μεταβλητή τους ταυτόχρονα. Θα θέλαμε να γνωρίζουμε ποιος ακριβώς είναι ο καταμερισμός πόρων σε μία τέτοια ισορροπία και αν μπορεί να επιτευχθεί μέσω ενός απλού, κατανεμημένου επαναληπτικού αλγόριθμου.

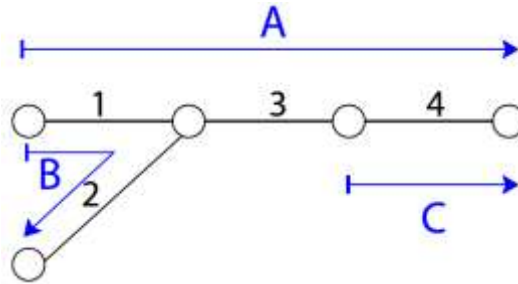
Το Σχήμα 14.5(β) δείχνει μία τυπική εξέλιξη του $cwnd$ του TCP Vegas κατά τη διάρκεια του χρόνου. Μπορείτε να δείτε ότι το «ζιγκ-ζαγκ» ανάμεσα στο ρυθμό που είναι πολύ επιθετικός (και οδηγεί σε συμφόρηση) και σε αυτόν που είναι υπερβολικά συντηρητικός (και οδηγεί στην υπο-αξιοποίηση των δυνατοτήτων των συνδέσεων) μπορεί να μειωθεί, σε σύγκριση με το TCP Reno. Η χρήση της καθυστέρησης ως ένα συνεχές σήμα της συμφόρησης είναι καλύτερη από τη χρήση μόνο της απώλειας ως ένα δυαδικό σήμα και θα δούμε διάφορα επιχειρήματα για αυτή την παρατήρηση στην επόμενη ενότητα.



Σχήμα 14.5 Τυπικές εξελίξεις των τιμών του $cwnd$ στο TCP Reno στα αριστερά και στο TCP Vegas στα δεξιά. Το TCP Reno χρησιμοποιεί την απώλεια ως σήμα συμφόρησης ενώ το TCP Vegas χρησιμοποιεί την καθυστέρηση ως σήμα συμφόρησης. Τα ζιγκ-ζαγκ ανάμεσα στην υπερφόρτωση και στην υπο-αξιοποίηση της χωρητικότητας τείνουν να είναι μικρότερα στο Vegas, αν οι παράμετροι είναι κατάλληλα συντονισμένοι.

14.2 Μία εκτενής απάντηση

Είτε κατανεμημένα, όπως το TCP, είτε μέσω ενός κεντροκοποιημένου συστήματος εντολών, κάθε πρωτόκολλο που προσπαθεί να ελέγξει τη συμφόρηση πρέπει να λάβει υπόψη του αυτό το βασικό θέμα: κάθε σταθερή χωρητικότητα c_i της ζεύξης l διαμοιράζεται σε πολλαπλές συνόδους και κάθε μία από αυτές τις από άκρο-σε-άκρο συνόδους διασχίζει πολλαπλές συνδέσεις. Υποθέτουμε ότι κάθε πηγή i έχει μία σύνοδο και χρησιμοποιεί δρομολόγηση μονής διαδρομής. Έτσι, οι «ροές», οι «συνεδρίες», οι «σύνοδοι» και οι «πηγές» είναι εναλλάξιμοι όροι σε αυτό το κεφάλαιο. Κάθε σύνδεση l διαμοιράζεται από ένα σύνολο συνόδων $S(l)$ και κάθε σύνοδος i χρησιμοποιεί ένα σύνολο από συνδέσεις $L(i)$ για τη διαδρομή της που αποφασίζεται από τη δρομολόγηση IP.



Σχήμα 14.6 Ένα απλό δίκτυο με τέσσερις συνδέσεις και τρεις συνόδους. Οι συνόδοι *A* και *B* μοιράζονται τη σύνδεση 1 και οι συνόδοι *A* και *C* μοιράζονται τη σύνδεση 4. Περιορισμένοι από τις σταθερές χωρητικότητες στις τέσσερις συνδέσεις, δεν είναι εύκολο να σχεδιάσουμε έναν κατανομημένο αλγόριθμο που να κατανέμει τις χωρητικότητες με έναν αποτελεσματικό και δίκαιο τρόπο ανάμεσα στις τρεις συνόδους.

Θεωρήστε το απλό παράδειγμα του Σχήματος 14.6. Όπως θα δούμε στο Κεφάλαιο 19, συχνά συνοδεύεται από σηματοδότηση ελέγχου που διασχίζει και άλλες διαδρομές. Μερικές φορές τα περιεχόμενα μίας συνόδου βρίσκονται επίσης και σε άλλες τοποθεσίες, π.χ. διαφημίσεις σε μία ιστοσελίδα πρέπει να μεταφορτωθούν από έναν διαφορετικό εξυπηρετητή από ότι το πραγματικό περιεχόμενο της ιστοσελίδας. Εδώ αγνοούμε αυτούς τους παράγοντες.

Σε αυτό το γράφο, η σύνοδος *A*, που προέρχεται από τον κόμβο 1, διασχίζει τις συνδέσεις 1, 3 και 4. Και η σύνδεση 1 διαμοιράζεται από τις συνόδους *A* και *B*. Ακόμα και εάν η σύνδεση 3 δεν αξιοποιείται πλήρως, δεν μπορούμε να αυξήσουμε το ρυθμό της συνόδου *A*, καθώς η σύνδεση με συμφόρηση για αυτή τη σύνοδο μπορεί να είναι η σύνδεση 1.

Πώς μπορούμε να κατανειμούμε τη χωρητικότητα κάθε σύνδεσης, έτσι ώστε οι συνόδοι να χρησιμοποιούν όσο το δυνατόν περισσότερη συνολική χωρητικότητα, χωρίς να προκαλούν συμφόρηση και ο ανταγωνισμός τους να συγχρονίζεται δίκαια; Μία κατανομή χωρητικότητας πρέπει κατ' αρχάς να είναι εφικτή μέσα στους περιορισμούς χωρητικότητας των συνδέσεων και κατόπιν να είναι επίσης αποτελεσματική και δίκαιη.

Στο Σχήμα 14.6, θεωρήστε ότι η χωρητικότητα κάθε σύνδεσης είναι 1 Mbps. Μία λύση που ικανοποιεί όλους τους περιορισμούς της χωρητικότητας όλων των τεσσάρων συνδέσεων, και συνεπώς αποτελεί εφικτή λύση, είναι η $(0,5, 0,5, 0,5)$ για τις τρεις συνόδους *A*, *B* και *C*, σε Mbps. Σε αυτή την ίση κατανομή των από άκρο-σε-άκρο ρυθμών, από τις χωρητικότητες κατά μήκος όλων των συνδέσεων χρησιμοποιούνται 3 Mbps. Για την ίδια αξιοποίηση της χωρητικότητας, μία άλλη εφικτή λύση είναι η $(1, 0, 0)$, η οποία αφήνει τις συνόδους *B* και *C* χωρίς εξυπηρέτηση και ίσως να μη φαίνεται ως δίκαιη κατανομή. Αποδεικνύεται ότι μία τυπική γνώμη της ίσης

μεταχείρισης θα έδινε την κατανομή $(1/3, 2/3, 2/3)$ στις τρεις ανταγωνιστικές συνόδους. Αυτό μπορεί να κάνει αίσθηση σε κάποιους ανθρώπους, καθώς η σύνοδος Α διασχίζει δύο συνδέσεις, που είναι πιθανά σημεία συμφόρησης, ενώ οι σύνοδοι Β και C διασχίζουν μόνο μία τέτοια σύνδεση.

Τώρα πρέπει να γράψουμε τη διατύπωση του προβλήματος ακριβέστερα. Θα ονομάσουμε αυτή τη βελτιστοποίηση βασικό πρόβλημα Μεγιστοποίησης Χρησιμότητας Δικτύου (Network Utility Maximization – NUM). Είναι η δικτυακή έκδοση του προβλήματος κοινωνικής πρόνοιας του Κεφαλαίου 11. Θα παρουσιάσουμε σύντομα μία κατανομημένη λύση σε αυτό το πρόβλημα σε αυτό το μέρος, αφήνοντας τα βήματα της παραγωγής της για το Προχωρημένο Υλικό. Μετά θα δείξουμε ότι το TCP Reno και το TCP Vegas στην πραγματικότητα μπορεί να είναι λύσεις αντίστροφης μηχανικής σε συγκεκριμένα προβλήματα NUM.

14.2.1 Διατύπωση του προβλήματος NUM

Πρέπει να διευθετήσουμε δύο θέματα στη μοντελοποίηση του ελέγχου συμφόρησης: (1) πώς να μετρήσουμε την επίδοση και τη δικαιοσύνη, και (2) πώς να συλλάβουμε τον περιορισμό της χωρητικότητας.

(1) Πώς μετράμε την επίδοση και τη δικαιοσύνη; Χρησιμοποιούμε τις συναρτήσεις ωφελείας που εισαγάγαμε στο Κεφάλαιο 11 και αθροίζουμε τη χρησιμότητα κάθε μεμονωμένης συνόδου TCP στον τελικό χρήστη. Μοντελοποιούμε την ωφέλεια ως μία συνάρτηση του ρυθμού μετάδοσης από άκρο-προς-άκρο μίας συνόδου TCP εδώ, καθώς προσαρμόζουμε μόνο αυτούς τους ρυθμούς υποθέτοντας ότι η επίδοση της εφαρμογής εξαρτάται μόνο από αυτό το ρυθμό. Η δικαιοσύνη μπορεί επίσης να συλληφθεί μέσω κάποιων από αυτές τις συναρτήσεων ωφελείας, όπως οι α -δίκαιες συναρτήσεις ωφελείας.

(2) Πώς αναπαριστούμε τον περιορισμό της χωρητικότητας της σύνδεσης; Για κάθε σύνδεση l υπάρχει μία περιορισμένη χωρητικότητα c_l σε bps. Το φορτίο θα πρέπει να είναι μικρότερο από το c_l . Υπάρχουν αρκετοί τρόποι για να εκφράσουμε το φορτίο σε μία σύνδεση, από την άποψη των μεταβλητών ρυθμών μετάδοσης στις πηγές και τις δεδομένες αποφάσεις δρομολόγησης.

Μπορούμε να γράψουμε το φορτίο στη σύνδεση l ως το άθροισμα των πηγαίων ρυθμών x_i διαμέσου αυτών των πηγών χρησιμοποιώντας αυτή τη σύνδεση: $\sum_{i \in S(l)} x_i$. Η

μπορούμε να χρησιμοποιήσουμε το R_{li} ως ένα δυαδικό δείκτη, έτσι ώστε $R_{li}=0$, εάν η σύνοδος της πηγής i διασχίζει τη σύνδεση l , και $R_{li}=1$ διαφορετικά. Σε αυτή την παράσταση, μπορείτε εύκολα να δείτε ότι οι περιορισμοί

$$\sum_i R_{li} x_i \leq c_l, \forall l$$

είναι ισοδύναμοι με τις παρακάτω γραμμικές ανισότητες σε σημειολογία πίνακα:

$$\mathbf{R}\mathbf{x} \leq \mathbf{c},$$

όπου το \leq ανάμεσα σε δύο διανύσματα συμβολίζει μία ανισότητα στοιχείων ανάμεσα στα αντίστοιχα στοιχεία των διανυσμάτων.

Για παράδειγμα, στη τοπολογία δικτύου στο Σχήμα 14.6, ο περιορισμός της χωρητικότητας της σύνδεσης γίνεται σε μορφή πίνακα

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_A \\ X_B \\ X_C \end{bmatrix} \leq \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}.$$

14.2.2 Κατανεμημένος αλγόριθμος που λύνει το NUM

Τώρα έχουμε προσδιορίσει πλήρως το πρόβλημα κατανομής της χωρητικότητας της σύνδεσης, που θεσπίζει τι θα πρέπει να επιλύει ο έλεγχος συμμόρφωσης:

$$\begin{array}{ll} \text{μεγιστοποιήστε} & \sum_i U_i(x_i) \\ \text{υπό τον περιορισμό} & \mathbf{R}\mathbf{x} \leq \mathbf{c} \\ \text{με μεταβλητές} & x_i \geq 0, \forall i \end{array} \quad (14.1)$$

$R\mathbf{x} \leq \mathbf{c}$ $R\mathbf{x} \leq \mathbf{c}$. Αναφερόμαστε σε αυτό το πρόβλημα ως το βασικό πρόβλημα NUM. Το πρόβλημα (14.1) είναι εύκολο να λυθεί για διάφορους λόγους.

- Είναι ένα κυρτό πρόβλημα βελτιστοποίησης, όπως ορίστηκε στο Κεφάλαιο 4. Ακριβέστερα, αυτή τη φορά μεγιστοποιείται μία κοίλη συνάρτηση οφέλους (το άθροισμα των ωφελειών όλων των συνόδων) από το να ελαχιστοποιείται μία κυρτή συνάρτηση κόστους. Συνεπώς, απολαμβάνει όλα τα προνόμια της κυρτής βελτιστοποίησης: μία τοπικά βέλτιστη λύση είναι επίσης και καθολικά βέλτιστη, το χάσμα δυαδικότητας (μία έννοια που θα την εισαγάγουμε σύντομα) είναι μηδενικό και μπορεί να λυθεί πολύ αποτελεσματικά, τουλάχιστον σε έναν κεντρικό υπολογιστή.
- Είναι επίσης διασπασίμο. Η διάσπαση εδώ αναφέρεται στη διάσπαση ενός προβλήματος βελτιστοποίησης σε πολλά μικρότερα, κάπως συντονισμένα,

έτσι ώστε η λύση τους να είναι ισότιμη με τη λύση του αρχικού προβλήματος. Γιατί να ενδιαφερόμαστε να έχουμε πολλά προβλήματα αντί για ένα; Γιατί μία τέτοια διάσπαση οδηγεί σε έναν καταναμημένο αλγόριθμο. Κάθε ένα από αυτά τα μικρότερα προβλήματα είναι ευκολότερο να λυθεί, συχνά τοπικά σε κάθε κόμβο του δικτύου. Και αν ο συντονισμός τους μπορεί να επιτευχθεί χωρίς την αποστολή μηνυμάτων, θα έχουμε ένα πραγματικά καταναμημένο τρόπο για να λύσουμε το πρόβλημα.

Αφήνοντας την παραγωγή της διάσπασης (μέσω του δυϊκού προβλήματος Lagrange) για το Προχωρημένο Υλικό, έχουμε την παρακάτω επαναληπτική λύση για το (14.1), που αποτελείται από ενέργειες της πηγής και ενέργειες του δρομολογητή.

Σε κάθε ένα από τα διακριτά χρονικά διαστήματα $[t]$ η πηγή κάθε συνόδου απλά αποφασίζει το ρυθμό μετάδοσής της από τη συνάρτηση ζήτησής της, σε σχέση με την τρέχουσα τιμή κατά μήκος της διαδρομής της. Αυτή η τιμή της διαδρομής q_i είναι το άθροισμα των τιμών συνδέσεων P_l όλων των συνδέσεων που διασχίζει η διαδρομή: $q_i = \sum_{l \in L(i)} p_l$. Ο ρυθμός μεταφοράς της πηγής i είναι

$$x_i[t] = D_i(q_i[t]) = U_i^{t-1}(q_i[t]), \quad (14.2)$$

όπου D είναι η συνάρτηση ζήτησης όπως προσδιορίστηκε στο Κεφάλαιο 11.

Φυσικά, σε μία εφαρμογή βάσει συρόμενου παραθύρου, η πηγή προσαρμόζει το μέγεθος παραθύρου $cwnd$ αντί για το x_i απευθείας. Η τιμή της διαδρομής υπηρετεί ως το σήμα ανάδρασης συμφόρησης από το δίκτυο. Ελπίζουμε ότι μπορεί να επιτευχθεί χωρίς τη μετάδοση μηνυμάτων σε πραγματικές εφαρμογές. Την ίδια στιγμή, ο δρομολογητής σε κάθε σύνδεση l ανανεώνει την «τιμή» σε αυτή τη σύνδεση:

$$p_l[t] = \{p_l[t-1] + \beta(y_l[t] - c_l)\}^+ \quad (14.3)$$

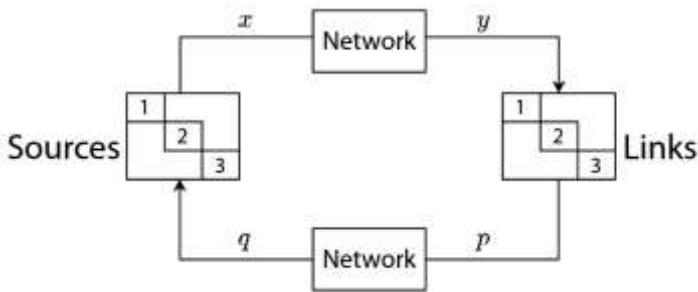
όπου y_l είναι το συνολικό φορτίο στη σύνδεση l : $y_l[t] = \sum_{i \in S(l)} x_i[t]$. Όπου το $\{\dots\}^+$

απλά λέει ότι, αν η έκφραση μέσα στις αγκύλες παίρνει αρνητική τιμή, τότε αυτή επιστρέφει την τιμή 0. Σε αυτή την περίπτωση, αυτό σημαίνει ότι η τιμή της σύνδεσης δεν επιτρέπεται ποτέ να πάρει αρνητική τιμή.

Αυτό αποτελεί μία ερμηνεία στη γλώσσα της τιμολόγησης, όχι πραγματικά χρήματα που αλλάζουν χέρια ανάμεσα στις οντότητες του δικτύου, όπως στα Κεφάλαια 11 και 12. Η παράμετρος $\beta \geq 0$ είναι το μέγεθος βήματος που ελέγχει το συμβιβασμό ανάμεσα στην εγγύηση ύπαρξης σύγκλισης και στην ταχύτητα σύγκλισης. Σκεφτείτε ότι παίζετε γκολφ: αν χτυπήσετε τη μπάλα πολύ δυνατά, ακόμα και στη σωστή κατεύθυνση, θα περάσει από την τρύπα και θα πρέπει να τη χτυπήσετε προς τα πίσω πάλι. Το μέγεθος του βήματος είναι κάτι σαν μέγεθος της δύναμής σας. Αν είναι

επαρκώς μικρή, ο παραπάνω αλγόριθμος είναι εγγυημένος ότι θα συγκλίνει και θα συγκλίνει προς τη σωστή λύση στον τύπο (14.1). Αν είναι πολύ μικρή, η εγγυημένη σύγκλιση γίνεται πολύ αργή. Σε πραγματικά συστήματα, ο συντονισμός αυτής της παραμέτρου δεν είναι εύκολος.

Ο βρόχος ανάδρασης στο ζεύγος των εξισώσεων (14.2) και (14.3) απεικονίζεται στο Σχήμα 14.7. Έχει νόημα από μία οικονομική οπτική γωνία. Αν, στον χρόνο t , υπάρχει περισσότερο φορτίο από τη χωρητικότητα στη σύνδεση l , τότε η τιμή p_l θα αυξηθεί σύμφωνα με την (14.3) και οι τιμές όλων των διαδρομών, που περιλαμβάνουν τη σύνδεση l , θα αυξηθούν στην επόμενη χρονοθυρίδα $t+1$.



Σχήμα 14.7 Ο βρόχος ανάδρασης ελέγχου σε μία κατανεμημένη λύση του NUM. Κάθε πηγή i υιοθετεί αυτόνομα το μέγεθος του παραθύρου της (ή το ρυθμό μετάδοσης x_i) με βάση την τιμή της ανάδρασης συμφόρησης της διαδρομής q_i , ενώ κάθε σύνδεση l αυτόνομα υιοθετεί την τιμή συμφόρησης p_l με βάση το φορτίο y_l .

Μία υψηλότερη τιμή θα μειώσει τη ζήτηση σύμφωνα με την (14.2) και το x_i θα πέσει σε όλες τις πηγές, που χρησιμοποιούν τη σύνδεση l , βοηθώντας να αποκατασταθεί η ισορροπία ανάμεσα στη ζήτηση και στην προσφορά στη σύνδεση l . Αυτό το σήμα τιμολόγησης εξισορροπεί την ελαστική ζήτηση και τη σταθερή προσφορά της χωρητικότητας. Αυτό που είναι ενδιαφέρον είναι ότι εκπληρώνει αυτό το έργο κατανεμημένα μέσω ενός δικτύου που αποτελείται από πολλές συνδέσεις και συνόδους. Μαθηματικά, αυτές οι τιμές σύνδεσης αποδεικνύεται ότι είναι οι μεταβλητές σε μία «εικόνα καθρέφτη» του βασικού προβλήματος NUM: οι μεταβλητές στο δυϊκό πρόβλημα Lagrange του NUM.

Ο παραπάνω αλγόριθμος δεν λύνει μόνο το βασικό πρόβλημα NUM (για κατάλληλο β), αλλά, επίσης, το λύνει και με έναν πολύ ωραίο τρόπο: πλήρως κατανεμημένα και προφανή διαισθητικά.

- Όπως φαίνεται καθαρά στην (14.3), κάθε σύνδεση πρέπει να μετρήσει μόνο το δικό της συνολικό φορτίο. Δεν χρειάζεται να γνωρίζει την κατάσταση καμίας άλλης σύνδεσης ούτε ακόμα και το φορτίο που έρχεται από κάθε πηγή που τη χρησιμοποιεί.

- Όπως φαίνεται καθαρά στην (14.2), κάθε πηγή χρειάζεται να γνωρίζει μόνο τη συνολική τιμή κατά μήκος της διαδρομής που θα χρησιμοποιήσει. Δεν χρειάζεται να γνωρίζει την κατάσταση καμίας άλλης διαδρομής ή πηγής ή ακόμα την τιμή ανά σύνδεση κατά μήκος της διαδρομής που χρησιμοποιεί. Αν η τιμή της διαδρομής q_i μπορεί να μετρηθεί τοπικά σε κάθε πηγή i χωρίς μία ρητή αποστολή μηνυμάτων, αυτή θα ήταν η ολοκληρωμένα κατανεμημένη λύση. Αυτή είναι η περίπτωση της χρήσης απωλειών πακέτων στην ανάδραση με τιμολόγηση στο TCP Reno και των καθυστερήσεων πακέτων στην ανάδραση με τιμολόγηση στο TCP Vegas.

14.2.3 Αντίστροφη μηχανική

Σε αυτό το σημείο, μπορεί να υπάρχει η αίσθηση ότι το πρώτο μέρος αυτού του κεφαλαίου και το τρέχον μέρος είναι κάπως ασύνδετα. Είναι τα πρωτόκολλα ελέγχου συμφόρησης του TCP, που εφαρμόζονται στο Διαδίκτυο, σχετικά με την κατανεμημένη λύση του βασικού προβλήματος NUM; Περίπου μία δεκαετία μετά τη δημιουργία του πρώτου πρωτόκολλου ελέγχου συμφόρησης του TCP οι ερευνητές χρησιμοποίησαν αντίστροφη μηχανική σε αυτά τα πρωτόκολλα και έδειξαν ότι είναι δυνατόν στην πραγματικότητα να ερμηνευτούν, περίπου, ως λύσεις του NUM. Αν μου περιγράψετε ένα πρωτόκολλο, μπορώ να σας πω ποια είναι η συνάρτηση οφέλους που εν δυνάμει μεγιστοποιείται, και ποιες είναι οι μεταβλητές τιμολόγησης.

Για παράδειγμα, αποδεικνύεται ότι το TCP Reno μεγιστοποιεί έμμεσα τα οφέλη που ορίζονται μέσω μίας συνάρτησης του τόξου της εφαπτομένης με τιμή τις απώλειες πακέτων. Και το TCP Vegas μεγιστοποιεί έμμεσα τα λογαριθμικά οφέλη, με τιμή τις καθυστερήσεις των πακέτων. Στο Προχωρημένο Υλικό, θα παρουσιάσουμε την παραγωγή της περίπτωσης του TCP Reno και μία παρόμοια παραγωγή για το TCP Vegas δίνεται ως άσκηση.

Η αντίστροφη μηχανική παρουσιάζει μία ιδιόρρυθμη άποψη: δώστε μου τη λύση και θα σας πω ποιο είναι το πρόβλημα που λύνεται από αυτή τη λύση. Θα αναρωτιέστε, γιατί να ενδιαφέρεστε για το πρόβλημα, αν έχετε ήδη τη λύση; Λοιπόν, η ανακάλυψη του υποκείμενου προβλήματος που λύνεται παρέχει μία ενδελεχή κατανόηση, του γιατί αυτή η λύση λειτουργεί και πώς να την κάνουμε να λειτουργήσει καλύτερα. Επίσης, οδηγεί σε νέους σχεδιασμούς: σε εμπρόσθια μηχανική που χρησιμοποιεί τη βαθιά γνώση της αντίστροφης μηχανικής.

Εδώ συναντάμε ένα παράδειγμα των επιπτώσεων των ιδιοτήτων του TCP που προέρχεται από την αντίστροφη μηχανική. Καθώς η εφαρμογή του TCP Reno λύνει το NUM με οφέλη ως συναρτήσεις τόξου εφαπτομένης, γνωρίζουμε ότι ο εξισορροπημένος ρυθμός καθυστέρησης πακέτων, δηλαδή οι βέλτιστες δυαδικές μεταβλητές, δεν μπορεί να εξαρτώνται από παραμέτρους που δεν εμφανίζονται καν στο NUM.

Συγκεκριμένα, αν διπλασιάσουμε το μέγεθος της ενδιάμεσης μνήμης, δεν θα βοηθήσει στη μείωση του ρυθμού εξισορρόπησης καθυστέρησης πακέτων, καθώς το μέγεθος της ενδιάμεσης μνήμης δεν εμφανίζεται στο πρόβλημα NUM (14.1). Αυτό που συμβαίνει είναι ότι αυξάνοντας το μέγεθος της ενδιάμεσης μνήμης απλά αναβάλλεται η έναρξη της συμφόρησης και της απώλειας πακέτων, αλλά ουσιαστικά δεν βοηθάει με την εξισορρόπηση.

Ορίστε ένα άλλο παράδειγμα: καθώς στο TCP Vegas έχει εφαρμοστεί αντίστροφη μηχανική, ως λύση για τη μεγιστοποίηση της λογαριθμικής χρησιμότητας, οδηγούμαστε σε μία αναλογικά δίκαιη κατανομή των χωρητικοτήτων των συνδέσεων. Φυσικά, αυτό δεν εγγυάται ότι το TCP Vegas τελικά θα συγκλίνει. Αλλά, αν συγκλίνει, θα επιτύχουμε αναλογική δικαιοσύνη.

Επιπρόσθετα στη νοοτροπία της αντίστροφης μηχανικής, έχουμε εισαγάγει δύο σημαντικά θέματα σε αυτό το κεφάλαιο, θέματα που δεν διατρέχουν μόνο τον έλεγχο συμφόρησης του TCP στο Διαδίκτυο.

- Ένα σήμα ανάδρασης μπορεί να παραχθεί και να χρησιμοποιηθεί σε ένα δίκτυο για κατανεμημένο συγχρονισμό. Τα προσωπικά συμφέροντα των χρηστών μπορεί να παραταχθούν με τα σήματα τιμολόγησης για να επιτευχθεί η μεγιστοποίηση της γενικής ευημερίας. Σε ορισμένες περιπτώσεις, τα σήματα τιμολόγησης δεν απαιτούν καν ρητή μετάδοση μηνυμάτων, ένα επιπλέον προνόμιο που γενικά δεν συναντάται συχνά.
- Ένα πρωτόκολλο δικτύου μπορεί να αναλυθεί και να σχεδιαστεί ως ένας νόμος ελέγχου. Οι ιδιότητες των πρωτοκόλλων του Διαδικτύου μπορεί να αναλυθούν μέσα από τροχιές του αντίστοιχου νόμου ελέγχου. Αυτό μπορεί να ακούγεται προφανές όταν εκφράζεται, αλλά ήταν μία καινοτόμος άποψη όταν πρωτο-αναπτύχθηκε στα τέλη της δεκαετίας του 1990. Άνοιξε το δρόμο στον τρόπο σκέψης για τα πρωτόκολλα δικτύου όχι μόνο ως προσεγγίσεις δοκιμής-και-λάθους με προσέγγιση από τη βάση προς την κορυφή, αλλά επίσης και ως αποτέλεσμα μίας σχεδιαστικής μεθοδολογίας βασισμένης σε αρχές με προσέγγιση από τη βάση προς την κορυφή.

14.3 Παραδείγματα

Θεωρήστε ξανά το δίκτυο που φαίνεται στο Σχήμα 14.6. Ο πίνακας δρομολόγησης, όπου οι γραμμές είναι οι συνδέσεις και οι στήλες είναι οι πηγές, είναι

$$R = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

Η δρομολόγηση μπορεί επίσης να παρασταθεί ως $\{S(l)\}$ ή $\{L(i)\}$. Σε αυτό το παράδειγμα, έχουμε $S(1) = \{A, B\}$, $S(2) = \{B\}$, $S(3) = \{A\}$, $S(4) = \{A, C\}$ και $L(A) = \{1, 3, 4\}$, $L(B) = \{1, 2\}$, $L(C) = 4$.

Υποθέστε ότι η χωρητικότητα όλων των συνδέσεων είναι 1 Mbps και η συνάρτηση ωφέλειας είναι μία λογαριθμική συνάρτηση για όλες τις πηγές, δηλαδή $U_i(x_i) = \log x_i$. Ο σκοπός μας είναι να βρούμε το ρυθμό αποστολής κάθε πηγής: x_A , x_B και x_C . Το πρόβλημα NUM παίρνει τη μορφή:

$$\begin{aligned} & \text{μεγιστοποιήστε} && \log(x_A) + \log(x_B) + \log(x_C) \\ & \text{υπό τον περιορισμό} && \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_A \\ X_B \\ X_C \end{bmatrix} \leq \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}, && (14.4) \\ & && x_i \geq 0, \forall i. \end{aligned}$$

Θυμηθείτε ότι οι ρυθμοί των πηγών και οι τιμές των συνδέσεων συγκλίνουν σε μία κατανομημένη λύση στο πρόβλημα NUM μέσω των παρακάτω επαναληπτικών αναεώσεων:

$$\begin{aligned} x_i[t] &= U_i^{-1}(q_i[t]) = \frac{1}{q_i[t]}, \\ p_l[t] &= \left\{ p_l[t-1] + \beta(y_l[t] - c_l) \right\}^+, \end{aligned}$$

όπου q_i είναι η τιμή της διαδρομής που βλέπει η πηγή i (προσοχή στη μετάβαση του δείκτη χρόνου από το $t-1$ στο t)

$$\begin{aligned} q_i[t] &= \sum_{l \in L(i)} p_l(t-1), \\ y_l[t] &= \sum_{i \in L(l)} x_i(t). \end{aligned}$$

Ας αρχικοποιήσουμε τους ρυθμούς των πηγών στο 0 και το κόστος των συνδέσεων στο 1, δηλαδή $x_A[0] = x_B[0] = x_C[0] = 0$ και $p_1[0] = p_2[0] = p_3[0] = p_4[0] = 1$. Ας είναι το μέγεθος βήματος $\beta = 1$.

Στο $t = 1$, ανανεώνουμε πρώτα τους ρυθμούς των πηγών. Αφού

$$q_A[1] = p_1[0] + p_3[0] + p_4[0] = 1 + 1 + 1 = 3\sqrt{a^2 + b^2},$$

$$q_B[1] = p_1[0] + p_2[0] = 1 + 1 = 2,$$

$$q_C[1] = p_4[0] = 1,$$

έχουμε, σε Mbps,

$$x_A[1] = \frac{1}{q_A[1]} = 0,333$$

$$x_B[1] = \frac{1}{q_B[1]} = 0,5$$

$$x_C[1] = \frac{1}{q_C[1]} = 1$$

Μετά, ανανεώνουμε τις τιμές συνδέσεων. Καθώς τα φορτία των συνδέσεων είναι, σε Mbps,

$$y_1[1] = x_A[1] + x_B[1] = 0,33 + 0,5 = 0,833.$$

$$y_2[1] = x_B[1] = 0,5,$$

$$y_3[1] = x_A[1] = 0,333,$$

$$y_4[1] = x_A[1] + x_C[1] = 0,33 + 1 = 1,33.$$

έχουμε

$$p_1[1] = [p_1[0] + y_1[1] - c]^+ = [1 + 0,83 - 1]^+ = 0,833,$$

$$p_2[1] = [p_2[0] + y_2[1] - c]^+ = [1 + 0,5 - 1]^+ = 0,5,$$

$$p_3[1] = [p_3[0] + y_3[1] - c]^+ = [1 + 0,33 - 1]^+ = 0,333,$$

$$p_4[1] = [p_4[0] + y_4[1] - c]^+ = [1 + 1,33 - 1]^+ = 1,33.$$

Στο $t=2$, ανανεώνουμε τους ρυθμούς των πηγών. Αφού

$$q_A[2] = p_1[1] + p_3[1] + p_4[1] = 0,83 + 0,33 + 1,33 = 2,5$$

$$q_B[2] = p_1[1] + p_2[1] = 0,83 + 0,5 = 1,33$$

$$q_C[2] = p_4[1] = 1,33$$

έχουμε, σε Mbps,

$$x_A[2] = \frac{1}{q_A[2]} = 0,4,$$

$$x_B[2] = \frac{1}{q_B[2]} = 0,75,$$

$$x_C[2] = \frac{1}{q_C[2]} = 0,75.$$

Μετά, ανανεώνουμε τις τιμές συνδέσεων. Καθώς τα φορτία των συνδέσεων είναι, σε Mbps,

$$y_1[2] = x_A[2] + x_B[2] = 0,4 + 0,75 = 1,15,$$

$$y_2[2] = x_B[2] = 0,75,$$

$$y_3[2] = x_A[2] = 0,4,$$

$$y_4[2] = x_A[2] + x_C[2] = 0,4 + 0,75 = 1,15,$$

έχουμε

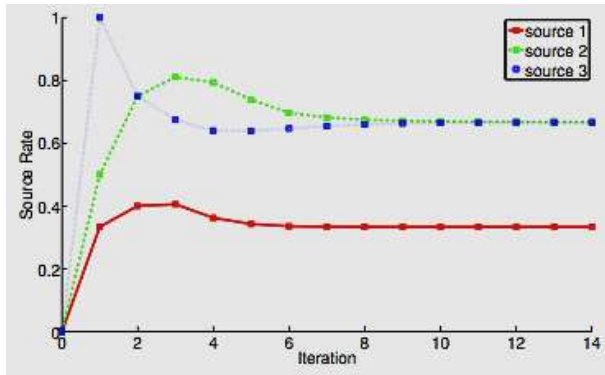
$$p_1[2] = [p_1[1] + y_1[2] - c]^+ = [0,83 + 1,15 - 1]^+ = 0,983,$$

$$p_2[2] = [p_2[1] + y_2[2] - c]^+ = [0,5 + 0,75 - 1]^+ = 0,25,$$

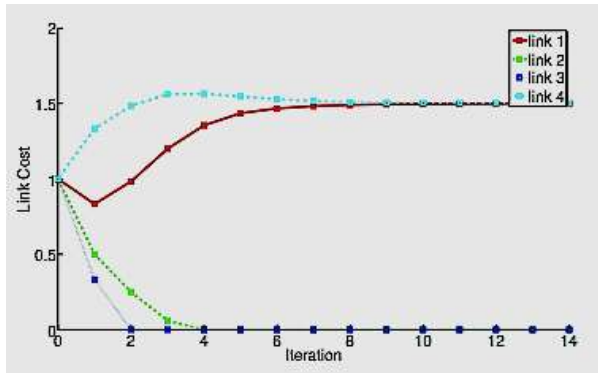
$$p_3[2] = [p_3[1] + y_3[2] - c]^+ = [0,33 + 0,4 - 1]^+ = 0,$$

$$p_4[2] = [p_4[1] + y_4[2] - c]^+ = [1,33 + 1,15 - 1]^+ = 1,48.$$

Οι επαναλήψεις συνεχίζονται. Απεικονίζουμε την εξέλιξή τους κατά τη διάρκεια του χρόνου στα Σχήματα 14.8 και 14.9.



Σχήμα 14.8 Σε ένα παράδειγμα ελέγχου συμφόρησης, οι ρυθμοί των πηγών κατά τη διάρκεια του χρόνου συγκλίνουν.



Σχήμα 14.9 Σε ένα παράδειγμα ελέγχου συμφόρησης, οι ρυθμοί των πηγών κατά τη διάρκεια του χρόνου.

Βλέπουμε ότι επιτυγχάνεται μία ισορροπία μετά από περίπου επτά επαναλήψεις. Σε αυτό το σημείο οι ρυθμοί των πηγών είναι

$$x_A^* = 0,33Mbps ,$$

$$x_B^* = 0,67Mbps ,$$

$$x_C^* = 0,67Mbps .$$

Είναι λογικό ότι, για αναλογική δικαιοσύνη, η σύνοδος που καταλαμβάνει περισσότερους πόρους του δικτύου παίρνει χαμηλότερο ρυθμό: η σύνοδος 1 διασχίζει διπλάσιες συνδέσεις με πιθανότητα συμφόρησης και παίρνει το μισό από τον καταναεμμένο ρυθμό.

Ως έλεγχο λογικότητας, ας σιγουρευτούμε ότι οι τιμές εξισορρόπησης ικανοποιούν του περιορισμούς στην (14.4). Προφανώς, όλα τα x_i^* είναι μη αρνητικά. Για να ελέγξουμε τους περιορισμούς της χωρητικότητας των συνδέσεων, βλέπουμε ότι

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_A \\ x_B \\ x_C \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0,33 \\ 0,67 \\ 0,67 \end{bmatrix} = \begin{bmatrix} 1 \\ 0,67 \\ 0,33 \\ 1 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Στην ισορροπία, οι τιμές των συνδέσεων είναι

$$p_1^* = 1,5,$$

$$p_2^* = 0,$$

$$p_3^* = 0,$$

$$p_4^* = 1,5.$$

Πάλι, είναι λογικό ότι, οι συνδέσεις 2 και 3 έχουν τιμή 0 στην ισορροπία, καθώς η χωρητικότητά τους δεν αξιοποιείται πλήρως στην ισορροπία, όπως περιορίζεται από τον τρόπο που δρομολογούνται οι σύνοδοι. Αντίθετα, οι συνδέσεις 1 και 4 έχουν αυστηρά θετικές τιμές στην ισορροπία και, λόγω της ιδιότητας της συμπληρωματικής χαλαρότητας που παρουσιάζεται στο Προχωρημένο Υλικό, γνωρίζουμε ότι οι χωρητικότητες των συνδέσεών τους πρέπει να αξιοποιούνται πλήρως στην ισορροπία, δηλαδή οι συνδέσεις 1 και 4 είναι σε συμφόρηση.

14.4 Προχωρημένο Υλικό

14.4.1 Διάσπαση του NUM

Σε αυτή την υποενότητα, θα αναγάγουμε τη λύση των (14.2) και (14.3) στο βασικό πρόβλημα NUM της (14.1). Η αντικειμενική συνάρτηση έχει ήδη διαχωριστεί σε όλες τις συνόδους i : κάθε όρος U_i στην αθροιστική ωφέλεια εξαρτάται μόνο από το ρυθμό x_i για αυτή τη σύνοδο i . Έτσι θα πρέπει μόνο να διαχωρίσουμε τους περιορισμούς. Είναι ακριβώς αυτό το σύνολο των γραμμικών παραμέτρων χωρητικότητας, που συνενώνει μαζί τις συνόδους μέσω του δεδομένου πίνακα δρομολόγησης \mathbf{R} .

Η μέθοδος αποσύνθεσης που θα χρησιμοποιήσουμε λέγεται **δυϊκή διάσπαση**, καθώς στην πραγματικότητα λύνει το **δυϊκό πρόβλημα Lagrange** του NUM. Με δεδομένο οποιοδήποτε πρόβλημα βελτιστοποίησης, μπορούμε να αντλήσουμε ένα

«κατοπτρικό» πρόβλημα που ονομάζεται το δυϊκό πρόβλημα. Μερικές φορές η βελτιστοποιημένη αντικειμενική τιμή του δυϊκού προβλήματος είναι ισότιμη με αυτή του πρωτότυπου αρχικού προβλήματος. Και κάθε φορά, παρέχει ένα όριο απόδοσης σε αυτήν του πρωτότυπου προβλήματος. Το δυϊκό πρόβλημα μπορεί μερικές φορές να λυθεί πολύ γρηγορότερα και, στην περίπτωση μας, να λυθεί με έναν καταναμημένο τρόπο.

Το πρώτο βήμα για την άντληση του δυϊκού προβλήματος Lagrange είναι να γράψουμε τη Λαγκραζιανή: το άθροισμα την πρωτότυπης αντικειμενικής συνάρτησης και ενός σταθμισμένου βάρους περιορισμών $c_l - \sum_{i \in S(l)} x_i \geq 0$. Τα θετικά βάρη ονομάζονται **πολλαπλασιαστές Lagrange** \mathbf{p} , και μπορεί να ερμηνευτούν ως τιμές των συνδέσεων:

$$L(\mathbf{x}, \mathbf{p}) = \sum_i U_i(x_i) + \sum_l p_l \left(c_l - \sum_{i \in S(l)} x_i \right). \quad (14.5)$$

Η ιδέα είναι ότι αλλάξαμε μία βελτιστοποίηση με πολλούς περιορισμούς σε μία πολύ ευκολότερη, χωρίς περιορισμούς, με το να μετακινήσουμε τους περιορισμούς ώστε να επαυξήσουμε την αντικειμενική συνάρτηση. Η προσδοκία είναι ότι αν θέσουμε τα σωστά βάρη \mathbf{p} , μπορούμε ακόμα να φτάσουμε και στη λύση του αρχικού προβλήματος.

Κατόπιν, ομαδοποιούμε ό,τι είναι σχετικό με τις μεταβλητές x , για να προσπαθήσουμε να εξαγάγουμε κάποια δομή από την Λαγκραζιανή: \mathbf{p}

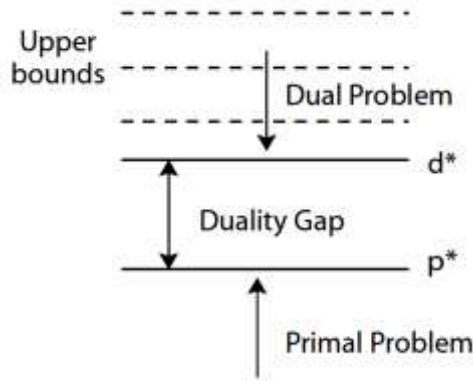
$$L(\mathbf{x}, \mathbf{p}) = \sum_i U_i(x_i) - \sum_l \sum_{i \in S(l)} p_l x_i + \sum_l c_l p_l.$$

Τώρα συμβαίνει κάτι σχεδόν μαγικό: μπορούμε να ξαναγράψουμε την παραπάνω διπλή άθροιση, με το να αντιστρέψουμε τη σειρά της άθροισης: πρώτα το άθροισμα ως προς το l με δεδομένο το i και μετά το άθροισμα ως προς τα i :

$$\sum_l \sum_{i \in S(l)} p_l x_i,$$

που μας επιτρέπει να ξαναγράψουμε το μέρος του L που περιλαμβάνει τις μεταβλητές ρυθμού x όπως παρακάτω:

$$L(\mathbf{x}, \mathbf{p}) = \sum_i \left[U_i(x_i) - \left(\sum_{l \in L(i)} p_l \right) x_i \right] + \sum_l c_l p_l.$$



Σχήμα 14.10 Υποθέστε ότι έχουμε ένα πρόβλημα μεγιστοποίησης, το οποίο μπορούμε να αποκαλούμε το αρχικό πρόβλημα, με μία βελτιστοποιημένη τιμή αντικειμενικής συνάρτησης p^* . Υπάρχει ένα αντίστοιχο δυϊκό πρόβλημα Lagrange, το οποίο είναι ένα πρόβλημα ελαχιστοποίησης, με μία βελτιστοποιημένη τιμή αντικειμενικής συνάρτησης d^* . Κάθε εφικτή λύση στο δυϊκό πρόβλημα παράγει ένα άνω όριο p^* του αρχικού προβλήματος. Αυτό λέγεται ασθενής δυϊκότητα. Το πιο στενό όριο είναι το d^* , που μπορεί ακόμα να έχει ένα κενό (χάσμα) από το p^* . Αν δεν υπάρχει κενό, όπως είναι η περίπτωση όταν το αρχικό πρόβλημα είναι μία κυρτή βελτιστοποίηση (και ικανοποιεί κάποιους τεχνικούς όρους), λέμε ότι ισχύει η ισχυρή ιδιότητα δυϊκότητας.

Για παράδειγμα, για το δίκτυο του Σχήματος 14.6, έχουμε

$$(x_A, x_B, x_C, p_1, p_2, p_3, p_4) = U_A(x_A) - (p_1 + p_3 + p_4)x_A + U_B(x_B) - (p_1 + p_2)x_B + U_C(x_C) - p_4x_C + c_1p_1 + c_2p_2 + c_3p_3 + c_4p_4$$

$$(x_A, x_B, x_C, p_1, p_2, p_3, p_4) = U_A(x_A) - (p_1 + p_3 + p_4)x_A + U_B(x_B) - (p_1 + p_2)x_B + U_C(x_C) - p_4x_C + c_1p_1 + c_2p_2 + c_3p_3 + c_4p_4$$

$$L(x_A, x_B, x_C, p_1, p_2, p_3, p_4) = U_A(x_A) - (p_1 + p_3 + p_4)x_A + U_B(x_B) - (p_1 + p_2)x_B + U_C(x_C) - p_4x_C + c_1p_1 + c_2p_2 + c_3p_3 + c_4p_4.$$

Σημειώνουμε την τιμή της διαδρομής (το άθροισμα των τιμών των συνδέσεων που χρησιμοποιούνται από τη σύνοδο i , ως q_i , έτσι $q_i = \sum_{l \in L(i)} p_l$. Κατόπιν έχουμε μία

αποσυντιθέμενη Λαγκραζιανή: η μεγιστοποίηση ως προς το \mathbf{x} μπορεί να εξαχθεί ανεξάρτητα από κάθε πηγή i (δείτε την τετράγωνη παρένθεση μέσα στο παρακάτω):

$$L(\mathbf{x}, \mathbf{p}) = \sum_i [U_i(x_i) - q_i x_i] + \sum_l c_l p_l.$$

Υποθέστε τώρα ότι μεγιστοποιούμε τη Λαγκραζιανή πάνω στις πρωτότυπες μεταβλητές \mathbf{x} . Αυτό ήταν το σχέδιό μας από την αρχή: να μετατρέψουμε μία περιορισμένη βελτιστοποίηση σε μία χωρίς περιορισμούς. Φυσικά ο μεγιστοποιητής και η μεγιστοποιημένη τιμή L εξαρτώνται από το ποιους πολλαπλασιαστές Lagrange \mathbf{p} χρησιμοποιήσαμε. Έτσι, πρέπει να σημειώσουμε την τιμή του αποτελέσματος ως συνάρτηση του \mathbf{p} :

$$g(\mathbf{p}) \max_{\mathbf{x}} L(\mathbf{x}, \mathbf{p}).$$

Αυτή η συνάρτηση $g(\mathbf{p})$ ονομάζεται δυϊκή συνάρτηση Lagrange.

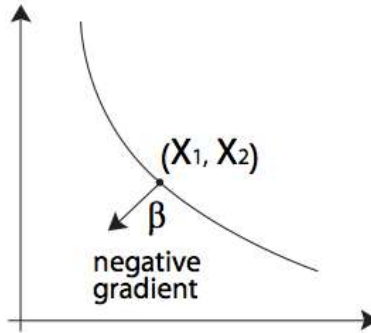
Αποδεικνύεται ότι όποιο \mathbf{p} και να χρησιμοποιήσουμε (αρκεί να μην είναι αρνητικό), η $g(\mathbf{p})$ παρέχει πάντα ένα όριο στην επίδοση. Αυτό είναι ένα άνω όριο στο μέγιστο $U^* = \sum_i U_i(x_i^*)$ του πρωτότυπου προβλήματος NUM. Αυτό είναι εύκολο να το

δούμε. Θεωρήστε το μεγιστοποιητή του προβλήματος NUM \mathbf{x}^* . Πρέπει να είναι ένα εφικτό διάνυσμα και να ικανοποιεί τον περιορισμό ανισότητας. Επίσης το \mathbf{p} είναι μη αρνητικό. Έτσι η Λαγκραζιανή $L(14.5)$ πρέπει να είναι τουλάχιστον μεγάλη όσο το U^* , όταν $\mathbf{x} = \mathbf{x}^*$. Καθώς η δυϊκή συνάρτηση Lagrange g είναι η μεγαλύτερη Λαγκραζιανή από όλα τα \mathbf{x} , πρέπει επίσης να μην είναι μικρότερη από το U^* . Αυτό ονομάζεται **ασθενής ιδιότητα της δυϊκότητας**, που στην πραγματικότητα ισχύει για όλα τα προβλήματα βελτιστοποίησης.

Τι θα λέγατε να στενέψουμε αυτό το όριο $g(\mathbf{p})$, με το να επιλέξουμε το καλύτερο \mathbf{p} ; Ονομάζουμε το πρόβλημα που προκύπτει πρόβλημα δυϊκότητας Lagrange και ονομάζουμε τις δυϊκές μεταβλητές Lagrange τώρα στο \mathbf{p} :

$$\text{ελαχιστοποιήστε}_{\mathbf{p}} g(\mathbf{p}).$$

Όπως φαίνεται στο Σχήμα 14.10, αν αυτός ο περιορισμός των ορίων παραγάγει την ακριβή απάντηση στην αρχική βελτιστοποίηση, λέμε ότι το βέλτιστο **χάσμα δυϊκότητας** είναι μηδέν και ότι διατηρείται η ιδιότητα της **ισχυρής δυϊκότητας**. Η ισχυρή δυϊκότητα δεν είναι πάντα αληθινή. Μαζί με κάποιους τεχνικούς όρους το αρχικό πρόβλημα, που είναι ένα κυρτό πρόβλημα βελτιστοποίησης, παρέχει μία επαρκή συνθήκη για να διατηρείται η ισχυρή δυϊκότητα. Αυτός είναι ένας άλλος λόγος, γιατί η κυρτή βελτιστοποίηση είναι εύκολη.



Σχήμα 14.11 Υποθέστε ότι θέλουμε να ελαχιστοποιήσουμε μία συνάρτηση δύο μεταβλητών, όπως φαίνεται εδώ. Ο αλγόριθμος κλίσης πηγαίνει από το τρέχον σημείο (x_1, x_2) κατά μήκος της κατεύθυνσης της αρνητικής κλίσης (negative gradient) της συνάρτησης, με ένα βέβαιο μήκος βήματος β . Μερικές φορές η κλίση μπορεί να υπολογιστεί κατανεμημένα.

Εφαρμόζοντας την παραπάνω δυϊκή αποσύνθεση για να διασπάσουμε ένα πρόβλημα σε πολλά μικρότερα στο NUM, βλέπουμε ότι το πρώτο βήμα είναι να μεγιστοποιήσουμε πάνω στο x για ένα ορισμένο \mathbf{p} . Αυτό δεν είναι τίποτα άλλο παρά η μεγιστοποίηση της χρησιμότητας του δικτύου (net utility) που είδαμε στο Κεφάλαιο 11, εγαιστικά και κατανεμημένα εξαγόμενη σε κάθε πηγή τώρα:

$$x_i^*(\mathbf{p}) = \arg \max [U_i(x_i) - q_i x_i].$$

Λαμβάνουμε ακριβώς την (14.2).

Το δεύτερο βήμα, που είναι η ελαχιστοποίηση της αντικειμενικής συνάρτησης του δυϊκού προβλήματος Lagrange $g(\mathbf{p}) = L(x^*(\mathbf{p}), \mathbf{p})$, ως προς το \mathbf{p} , μπορεί να επιτευχθεί με χρήση της μεθόδου κλίσης (gradient method). Πηγαίνοντας προς τα κάτω κατά μήκος της διεύθυνσης της αρνητικής κλίσης με ένα βήμα μεγέθους β , όπως φαίνεται στο Σχήμα 14.11, έχουμε

$$\mathbf{p}[t+1] = \mathbf{p}[t] - \beta \text{ (η κλίση του } g(\mathbf{p}) \text{ στο } \mathbf{p}[t]).$$

Αποδεικνύεται ότι για ένα γραμμικά περιορισμένο, κοίλο πρόβλημα μεγιστοποίησης όπως το NUM, η ίδια η συνάρτηση περιορισμού $c_i - \sum_{i \in S(t)} x_i = c_i - y_i$ είναι η κλίση

για κάθε \mathbf{p}_i . Έτσι το μόνο που χρειάζεται είναι να πολλαπλασιάσουμε την κλίση με ένα βήμα μεγέθους β (και μετά να σιγουρευτούμε ότι δεν είναι ποτέ αρνητική):

$$p_l[t] = \left\{ p_l[t-1] - \beta \left(c_l - \sum_{i \in S(l)} (x_i^*(\mathbf{p})) \right) \right\}^+$$

που είναι ακριβώς η (14.3).

Καθώς η ισχυρή δυϊκότητα διατηρείται στο NUM, η λύση του δυϊκού προβλήματος Lagrange είναι ισοδύναμη με τη λύση του αρχικού προβλήματος. Αυτό ολοκληρώνει την παραγωγή των (14.2) και (14.3) ως ένα κατανεμημένο αλγόριθμο λύσης για το (14.1).

Αυτές οι βελτιστοποιημένες αρχικές μεταβλητές (το διάνυσμα ρυθμού \mathbf{x}^*) και οι δυϊκές μεταβλητές (το διάνυσμα τιμών απόστασης \mathbf{p}^*) ικανοποιούν επίσης χρήσιμες ιδιότητες, συμπεριλαμβανομένης της παρακάτω συμπληρωματικής ιδιότητας χαλαρότητας. Αν ο αρχικός περιορισμός l είναι χαλαρός, δηλαδή $\sum_{i \in S(l)} x_i^* < c_l$, η αντί-

στοιχη βέλτιστη δυϊκή μεταβλητή p_l^* (η βέλτιστη τιμή συμφόρησης συνδέσεων) πρέπει να είναι 0, δηλαδή ο περιορισμός της μη-αρνητικότητας στο δυϊκό πρόβλημα δεν είναι χαλαρός. Αντίθετα, αν η βέλτιστη τιμή συμφόρησης συνδέσεων $p_l > 0$ για κάποια σύνδεση l , πρέπει να έχουμε $\sum_{i \in S(l)} x_i^* = c_l$, δηλαδή η σύνδεση l είναι ένα πιθανό

σημείο συμφόρησης στην ισορροπία.

14.4.2 Αντίστροφη μηχανική

Αναφέραμε πρωτότερα ότι εάν μου δώσετε ένα πρωτόκολλο ελέγχου συμφόρησης του TCP, μπορώ να σας επιστρέψω ένα αντίστοιχο πρόβλημα NUM, που λύνεται έμμεσα από αυτό, με τη συνάρτηση ωφέλειας να ικανοποιείται πλήρως, όπου οι ρυθμοί των πηγών (ή τα μεγέθη των παραθύρων) είναι οι μεταβλητές και τα σήματα τιμολόγησης είναι οι δυϊκές μεταβλητές Lagrange. Τώρα θα απεικονίσουμε αυτά τα κύρια βήματα σε αυτή την προσέγγιση αντίστροφης μηχανικής στο TCP Reno.

Το πρώτο βήμα στην παραγωγή είναι να γράψουμε την εξέλιξη του μεγέθους του παραθύρου συμφόρησης, σημειωμένο εδώ με το w , όπως καθορίζεται από το ορισμένο πρωτόκολλο. Κάθε φορά που λαμβάνεται μία επιβεβαίωση με τη σωστή σειρά από την πηγή, το μέγεθος του παραθύρου μεγαλώνει κατά $\frac{1}{w}$, σε σχέση με το τρέχον μέγεθός του. Συνεπώς, αν κάθε πακέτο παραλαμβάνεται κανονικά, το μέγεθος του παραθύρου αυξάνεται κατά 1 μετά από ένα RTT. Αλλά κάθε φορά που εντοπίζεται μία απώλεια πακέτου, το μέγεθος του παραθύρου μειώνεται στο μισό. Επομένως, η συνολική αλλαγή στο μέγεθος του παραθύρου $w[t]$ είναι

$$x[t](1-q[t])\frac{1}{w[t]} - x[t]q[t]\frac{w[t]}{2}, \quad (14.6)$$

όπου παραλείψαμε το δείκτη i για απλότητα.

Τώρα, ας πούμε ότι το RTT είναι d και ας υποθέσουμε ότι είναι σταθερό (αν και προφανώς διαφέρει στο χρόνο ανάλογα με την κατάσταση συμφόρησης). Καθώς

$x = \frac{w}{d}$ (14.6), αυτό οδηγεί στην παρακάτω διαφορο-ισότητα:

$$x[t+1] = x[t] + \frac{1-q[t]}{d^2} - \frac{1}{2}q[t]x^2[t].$$

Εξ' ορισμού, το x δεν αλλάζει πλέον σε ένα σημείο ισορροπίας, που σημαίνει ότι $\frac{1-q}{d^2} = \frac{1}{2}qx^2$. Αυτή η συνθήκη ισορροπίας μας δίνει μία ισότητα που συνδέει το q με το x :

$$q = \frac{2}{x^2d^2 + 2}.$$

Από τον ορισμό της συνάρτησης ζήτησης, γνωρίζουμε ότι $U_i(x_i) = q_i$. Έτσι, αν ενσωματώσουμε την παραπάνω έκφραση στο x , ανακτούμε τη συνάρτηση ωφέλειας:

$$U(x) = \frac{\sqrt{2}}{d} \arctan\left(\sqrt{\frac{1}{2}x_d}\right).$$

Συνοψίζοντας, η εξισορρόπηση του TCP Reno λύνει το NUM με ωφέλεια του τύπου του τόξου εφαπτομένης, με τη βοήθεια της απώλειας πακέτων, όπως οι μεταβλητές Lagrange (δηλαδή η ανάδραση τιμολόγησης). Μπορούμε επίσης να επιβεβαιώσουμε ότι ικανοποιείται η συμπληρωματική χαλαρότητα: εάν ένας αρχικός περιορισμός είναι χαλαρός, δηλαδή η ζήτηση είναι αυστηρώς μικρότερη από τη χωρητικότητα σε μία σύνδεση στην ισορροπία, δεν θα υπάρχει ούτε απώλεια, ούτε καθυστέρηση ουράς. Αντίθετα, αν ένας δυϊκός περιορισμός είναι χαλαρός, δηλαδή υπάρχει απώλεια ή καθυστέρηση ουράς σε μία σύνδεση, η χωρητικότητά του πρέπει να αξιοποιείται πλήρως.

Τώρα, ο ρυθμός απώλειας πακέτων q κατά μήκος μίας διαδρομής δεν είναι πραγματικά ίσος με το άθροισμα των ρυθμών απώλειας πακέτων στις συνδέσεις κατά μήκος της διαδρομής. Είναι $1 - \prod_i (1 - p_i)$, δηλαδή 1 μείον την πιθανότητα να μην

έχει χαθεί κανένα πακέτο σε καμία σύνδεση. Αλλά όταν ο ρυθμός απώλειας είναι μικρός, η $q_i = 1 - \prod_l (1 - p_l) \approx \sum_{l \in L(i)} p_l$ λειτουργεί πολύ καλά ως προσέγγιση.

- Εστίασαμε μόνο στη συμπεριφορά στην ισορροπία. Αλλά ένα πρωτόκολλο μπορεί να μην συγκλίνει. Η συμπεριφορά του στην ισορροπία μπορεί να είναι επιθυμητή, αλλά μία ισορροπία μπορεί ποτέ να μην επιτευχθεί.
- Αγνοήσαμε την πραγματική δυναμική των ουρών αναμονής.
- Αγνοήσαμε την καθυστέρηση της μετάδοσης που χρειάζονται τα πακέτα και οι επιβεβαιώσεις για να ταξιδέψουν μέσα στο δίκτυο.
- Υποθέσαμε ότι υπάρχει ένα σταθερό σύνολο συνόδων που μοιράζονται τις δυναμικότητες του δικτύου, οι οποίες υπάρχουν για πάντα.

Πολλές από αυτές τις παραδοχές αφαιρέθηκαν και επιτεύχθηκαν με την πάροδο του χρόνου ισχυρότερα αποτελέσματα. Αυτό που είναι κάπως αναπάντεχο είναι ότι, ακόμα και με μερικές από αυτές τις υποθέσεις, η θεωρητική πρόβλεψη από την ανάλυση του NUM δουλεύει αρκετά καλά σε σύγκριση με τις πραγματικές λειτουργίες του TCP.

Επιπλέον, το μοντέλο βελτιστοποίησης του ελέγχου συμφόρησης οδήγησε σε εμπρόσθια μηχανική νέων παραλλαγών του TCP, που είναι πιθανόν ευσταθείς. Η ευστάθεια εδώ σημαίνει ότι η τροχιά των μεταβλητών ενός πρωτόκολλου συγκλίνει προς την επιθυμητή ισορροπία, όπως η λύση σε ένα NUM με κατάλληλα επιλεγμένη συνάρτηση ωφέλειας. Μερικές από αυτές τις παραλλαγές έχουν εκτεθεί σε εργαστηριακά πειράματα και μετά εμπορευματοποιήθηκαν, συμπεριλαμβανομένων των FAST TCP για μεταδόσεις μεγάλων αποστάσεων και μεγάλου όγκου και του CUBIC, του προεπιλεγμένου πυρήνα του Linux.

Σύνοψη

Πλαίσιο 14 Κατανεμημένος έλεγχος συμφόρησης

Το TCP χρησιμοποιεί έναν προσανατολισμένο προς τη σύνδεση, με από-άκρο-προς-άκρο έλεγχο στο επίπεδο μεταφοράς και περιλαμβάνει έναν έλεγχο συμφόρησης βασισμένο σε συρόμενα παράθυρα. Τα σήματα ανάδρασης συμφόρησης στέλνονται έμμεσα από τις συνδέσεις στις πηγές ως απώλεια πακέτων ή καθυστέρηση. Αυτή η αρνητική ανάδραση μπορεί να ιδωθεί ως σήματα τιμολόγησης για τον κατανεμημένο συγχρονισμό της ζήτησης χωρητικότητας. Σε κάθε πρωτόκολλο ελέγχου συμφόρησης μπορεί να εφαρμοστεί αντίστροφη μηχανική ως μία έμμεση λύση ενός προβλήματος Μεγιστοποίησης Χρησιμότητας Δικτύου (NUM) και ενός δυϊκού προβλήματος Lagrange μέσω ενός κατανεμημένου αλγόριθμου κλίσης.

Περαιτέρω Μελέτη

Η μοντελοποίηση και ο σχεδιασμός του ελέγχου συμφόρησης είναι ένας τομέας ενεργούς έρευνας στη δικτύωση για περισσότερα από 20 χρόνια.

1. Ένα τυπικό βιβλίο αναφοράς στο TCP/IP είναι:

W. R. Stevens, TCP/IP Illustrated, Vol. 1: The Protocols, Addison Wesley, 1994

2. Η άποψη για το σύστημα ελέγχου δυναμικής, το μοντέλο βελτιστοποίησης και η τιμολογιακή ερμηνεία του TCP ήταν πρωτοπορία των Kelly και άλλων στα τέλη της δεκαετίας του '90:

F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: Shadow prices, proportional fairness, and stability," Journal of the Operational Research Society, vol. 49, pp. 237-252, March 1998

3. Τα πρωτόκολλα αντίστροφης μηχανικής του TCP σε μοντέλα NUM συνοψίστηκαν στο παρακάτω άρθρο:

S. H. Low, "A duality model of TCP and queue management system," IEEE/ACM Transactions on Networking, vol. 11, no. 4, pp. 525-536, July 2003

4. Η παρακάτω μονογραφία συνόψισε τα κύρια αποτελέσματα της μοντελοποίησης του ελέγχου συμφόρησης μέχρι το 2004, συμπεριλαμβανομένων και των στοχαστικών αφίξεων και αναχωρήσεων των συνόδων:

R. Srikant, The Mathematics of Internet Congestion Control, Berkhauser, 2004

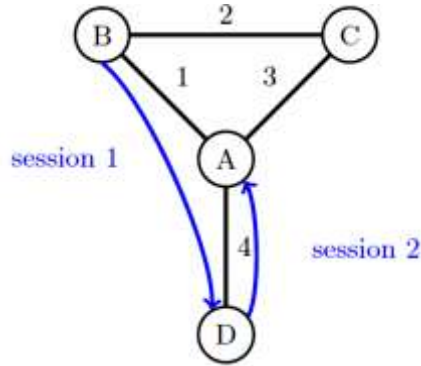
5. Το παρακάτω άρθρο ερευνήσε την γενικοποίηση της προσέγγισης των μοντέλων σε αυτό το κεφάλαιο από την άποψη της «στρωματοποίησης ως μία αποσύνθεση της βελτιστοποίησης»:

M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, “Layering as optimization decomposition: A mathematical theory of network architecture,” Proceedings of the IEEE, vol. 95, no. 1, pp. 255-312, January 2007.

Ασκήσεις

14.1 Ένα αριθμητικό παράδειγμα του NUM **

Υποθέστε ότι έχουμε ένα δίκτυο του οποίου η τοπολογία φαίνεται στο Σχήμα 14.12.



Σχήμα 14.12 Ένα μικρό δίκτυο για ένα αριθμητικό παράδειγμα του NUM.

- Γράψτε τον πίνακα δρομολόγησης A.
- Τρέξτε ένα αριθμητικό παράδειγμα για δέκα βήματα χρόνου για να λύσετε το πρόβλημα NUM, χρησιμοποιώντας τις ανανεώσεις των τιμών των συνδέσεων και των ρυθμών πηγών στις (14.2) και (14.3). Η συνάρτηση χρησιμότητας είναι μία λογαριθμική συνάρτηση του ρυθμού πηγής. Αρχικοποιήστε τις τιμές συνδέσεων στο 1 και τρέξτε το πρώτο βήμα της ανανέωσης του ρυθμού πηγών. Θέστε το μέγεθος βήματος $\beta=1$. Σχεδιάστε τους ρυθμούς πηγής και τις τιμές συνδέσεων κατά τη διάρκεια του χρόνου. Ποιες είναι οι τιμές ισορροπίας;
- Αλλάξτε το μέγεθος βήματος β και παρατηρήστε την επίδραση στη σύγκλιση του αλγόριθμου.

14.2 Αργή εκκίνηση στο TCP*

Μάθαμε τον αρχικό τρόπο λειτουργίας του TCP Reno, όπου το μέγεθος του παραθύρου συμφόρησης, που δίνεται ως w στις ασκήσεις, αυξάνεται κατά $\frac{1}{w}$ για κάθε

RTT. Αντίστοιχα, το w αυξάνεται κατά $\frac{1}{w}$ για κάθε ACK που λαμβάνεται. Αυτός ο τρόπος λειτουργίας ονομάζεται αποφυγή συμφόρησης.

Ωστόσο, αυτό έχει ως αποτέλεσμα μία γραμμική αύξηση του w κατά τη διάρκεια του χρόνου. Στο ξεκίνημα μίας σύνδεσης TCP, θα θέλαμε να αυξήσουμε γρήγορα το w , πριν μεταφερθούμε σε κατάσταση αποφυγής συμφόρησης. Τα περισσότερα πρωτόκολλα TCP έχουν μία (κάπως μπερδεμένα ονομασμένη) φάση αργής εκκίνησης, που το επιτυγχάνει. Σε αυτή την κατάσταση, το w αυξάνεται κατά 1 για κάθε ACK που λαμβάνεται.

- Αν σχεδιάσουμε το w ως συνάρτηση του χρόνου, αντί να έχουμε μία γραμμική αύξηση σε μία αποφυγή συμφόρησης, τι είδους αύξηση βλέπουμε στην αργή εκκίνηση;
- Υποθέστε ότι το w ξεκινάει από το 1. Σχεδιάστε ένα διάγραμμα χώρου-χρόνου για τη φάση της αργής εκκίνησης για τέσσερα RTT.

14.3 Παράθυρο συμφόρησης του TCP Reno**

Θυμηθείτε ότι το μέγεθος του παραθύρου συμφόρησης, κατά τη διάρκεια της φάσης αποφυγής συμφόρησης του TCP Reno, αλλάζει με το χρόνο όπως παρακάτω.

- Αν ένα ACK παραληφθεί, τότε αυξήστε το w κατά $\frac{1}{w}$.
- Αν εντοπιστεί συμφόρηση, τότε μειώστε το w κατά $\frac{w}{2}$.

Υποθέστε ότι η πιθανότητα αποτυχίας μίας μετάδοσης είναι p , η πιθανότητα μίας επιτυχημένης μετάδοσης είναι τότε $1-p$. Ο ρυθμός μετάδοσης είναι $x = \frac{w}{RTT}$ πακέτα ανά δευτερόλεπτο.

(α) Γράψτε την εξίσωση για την αναμενόμενη αλλαγή του w ανά χρονικό βήμα.

(β) Στην ισορροπία, η αναμενόμενη αλλαγή είναι 0. Με τη χρήση του (α) δείξτε ότι

$$x_r = \frac{1}{RTT} \sqrt{\frac{2(1-p)}{p}}.$$

14.4 TCP Vegas**

Το TCP Vegas προσπαθεί να προβλέψει τη συμφόρηση από την εκτιμώμενη καθυστέρηση. Σε αυτό το πρωτόκολλο, το μέγεθος του παραθύρου συμφόρησης αλλάζει με βάση το χρονισμό των ACK που παραλαμβάνονται. Συγκεκριμένα, έχουμε:

$$w[t+1] = \left\{ \begin{array}{l} w[t] + 1 \text{ αν } \frac{w[t]}{d} - \frac{w[t]}{D[t]} < \beta \\ w[t] - 1 \text{ αν } \frac{w[t]}{d} - \frac{w[t]}{D[t]} < \beta \\ w[t] \text{ διαφορετικά} \end{array} \right\},$$

όπου d είναι το ελάχιστο RTT που έχει παρατηρηθεί ιστορικά, $D[t]$ είναι το RTT που παρατηρήθηκε στο χρόνο t και β είναι ένα κατώφλι. Έτσι, $\frac{w}{d}$ είναι ο αναμενόμενος ρυθμός και $\frac{w}{D}$ είναι ο ρυθμός που έχει παρατηρηθεί και το w μειώνεται (αυξάνεται) αν ο αναμενόμενος ρυθμός είναι μεγαλύτερος (μικρότερος) από τον πραγματικό ρυθμό ανά β .

Υποθέστε ότι:

$$D[t] = \left\{ \begin{array}{l} t \text{ αν } t \leq 4 \\ 4 \text{ διαφορετικά} \end{array} \right\}$$

και ότι $w[1]=4$, $\beta=3$ και $d=1$. Σχεδιάστε την εξέλιξη του w ως συνάρτηση του χρόνου για δέκα βήματα χρόνου.

14.5 Αντίστροφη μηχανική του TCP Vegas

Μπορεί να αποδειχθεί ότι το TCP Vegas σχεδόν λύνει το παρακάτω σταθμισμένο λογαριθμικό πρόβλημα μεγιστοποίησης ωφέλειας, όπου β_i είναι η παράμετρος του πρωτόκολλου στην τελευταία άσκηση και d_i είναι το RTT χωρίς συμφόρηση για τη σύνοδο i :

$$\begin{array}{ll} \text{μεγιστοποιήστε} & \sum_i \beta_i d_i \log x_i \\ \text{υπό τον περιορισμό} & \sum_{i \in S(l)} x_i \leq c_l, \forall l \\ \text{με μεταβλητές} & x_i \geq 0, \forall i \end{array} \quad (14.7)$$

Οι συνδέσεις ανανεώνουν τις τιμές τους όπως παρακάτω:

$$p_l[t+1] = \left\{ p_l[t] + y_l (y_l[t] - c_l) \right\}^+ \quad (14.8)$$

Όπως πριν,

$L(i)$ είναι το σύνολο των συνδέσεων που χρησιμοποιούνται από τη σύνοδο i , $S(l)$ είναι το σύνολο των συνόδων που χρησιμοποιούν τη σύνδεση l ,

$$y_i[t] = \sum_{i \in S(l)} x_i[t], \quad (14.9)$$

$$q_l[t] = \sum_{l \in L(i)} p_i[t].$$

Θα το δείξουμε αυτό μέσω αρκετών βημάτων.

α. Προσδιορίστε τα ανεκτέλεστα υπόλοιπα (backlog) στη σύνδεση l από όλες τις συνόδους $b_l[t]$. Κατόπιν, κάθε σύνδεση ανανεώνει τις εκκρεμότητές της σε κάθε

βήμα χρόνου κατά $b_l[t+1] = \{b_l[t] + \beta(y_i[t] - c_l)\}^+$. Δείξτε ότι, αν θέσουμε

$$p_l[t] = \frac{b_l[t]}{c_l}, \text{ οι συνδέσεις ανανεώνουν τις τιμές τους όπως στην (14.8)}$$

β. Αν ένα δίκτυο προσπαθεί να λύσει την (14.7), ποιος θα πρέπει να είναι ο κανόνας της ανανέωσης της πηγής; Θυμηθείτε ότι $x_i[t] = U_i^{-1}(q_i[t])$.

γ. Θυμηθείτε ότι το μέγεθος κάθε παραθύρου στο TCP Vegas ανανεώνεται ως ακολούθως:

$$w[t+1] = \begin{cases} w_i[t] + 1 & \text{αν } \frac{w_i[t]}{d} - \frac{w_i[t]}{D_i[t]} < \beta_i \\ w_i[t] - 1 & \text{αν } \frac{w_i[t]}{d} - \frac{w_i[t]}{D_i[t]} > \beta_i \\ w_i[t] & \text{διαφορετικά} \end{cases}.$$

Γνωρίζουμε επίσης ότι οι εκκρεμότητες στη σύνδεση l από τη σύνοδο i είναι $\frac{x_i[t]}{c_l} b_l[t]$. Το μέγεθος του παραθύρου συμφόρησης, w_i , είναι το άθροισμα των

συνολικών εκκρεμοτήτων στη διαδρομή της i και του γινομένου της καθυστέρησης επί το εύρος ζώνης, δηλαδή $w_i[t] = \sum_{l \in L(i)} \frac{x_i[t]}{c_l} b_l[t] + d_i x_i[t]$. Δείξτε ότι ο

κανόνας ανανέωσης του ρυθμού της πηγής ταιριάζει με την απάντηση του μέρους (β).