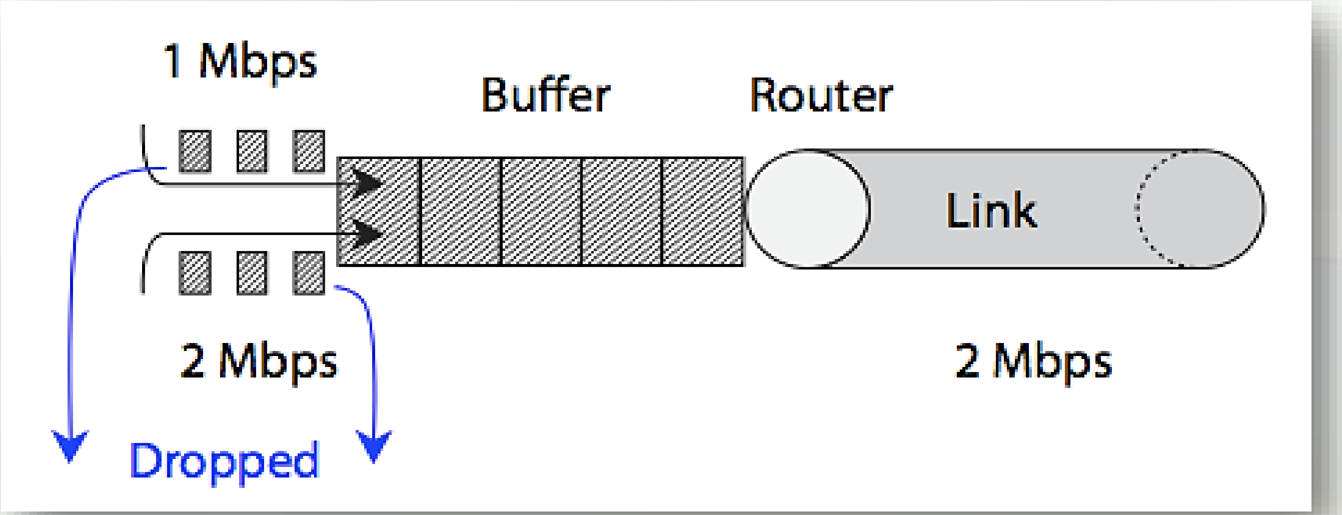
Κεφάλαιο 14: Γιατί το Διαδίκτυο δεν καταρρέει λόγω συμφόρησης;

Καθηγητής Χρήστος Δουληγέρης
E-mail: cdoulig@unipi.gr
Γραφείο: 302

Αρχές του κατανεμημένου ελέγχου συμφόρησης

- Όταν η ζήτηση υπερβαίνει την προσφορά, έχουμε συμφόρηση.
- Τον Οκτώβριο του 1986 το Διαδίκτυο είχε την πρώτη του κατάρρευση από συμφόρηση.
- **Ερώτηση: Γιατί έχουμε συμφόρηση;**
  Απάντηση:
  - Όταν οι χρήστες στέλνουν πάρα πολλά bit/sec, 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, περισσότερα εισερχόμενα πακέτα συσσωρεύονται στην ενδιάμεση μνήμη, μέχρι αυτή να υπερχειλίσει και να απορριφθούν τα πακέτα.
  - 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.
  - ο αποστολέας πρέπει να ξαναστείλει τα μη επιβεβαιωμένα πακέτα. Αυτό οδηγεί σε έναν φαύλο κύκλο, έναν **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**.

Αρχές του κατανεμημένου ελέγχου συμφόρησης



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.

Αρχές του κατανεμημένου ελέγχου συμφόρησης (συνέχεια)

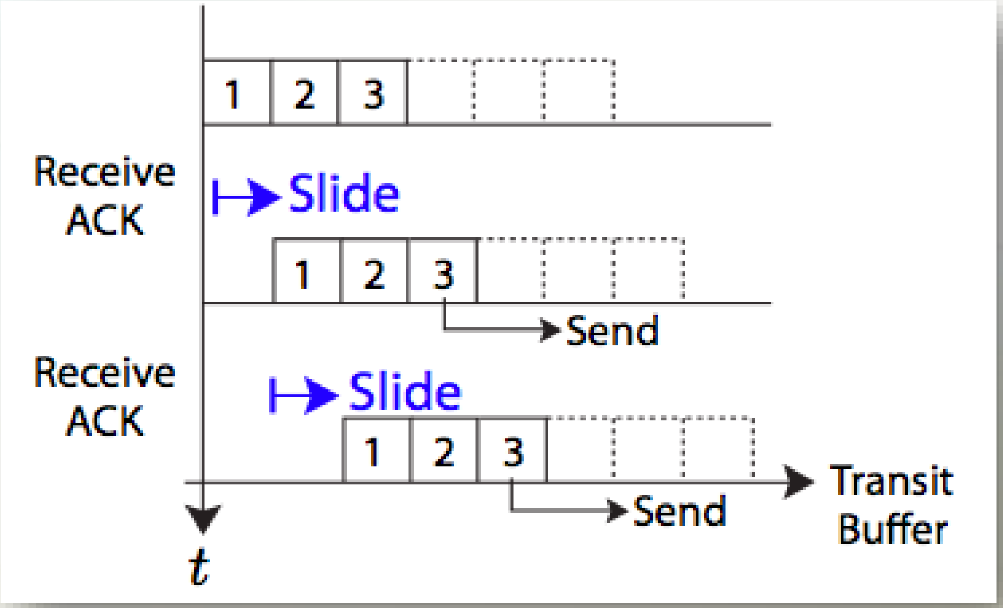
Από-άκρο-προς-άκρο έλεγχος μέσω αρνητικής ανάδρασης:

– 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.
– Ο έλεγχος συμφόρησης του TCP υιοθετεί μία εναλλακτική προσέγγιση, να έχει ένα ευφυές δίκτυο άκρων και ένα χαζό δίκτυο κορμού (dumb core network).
– 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.

Αρχές του κατανεμημένου ελέγχου συμφόρησης (συνέχεια)

Έλεγχος βάσει συρόμενου 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**.
- 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.



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.

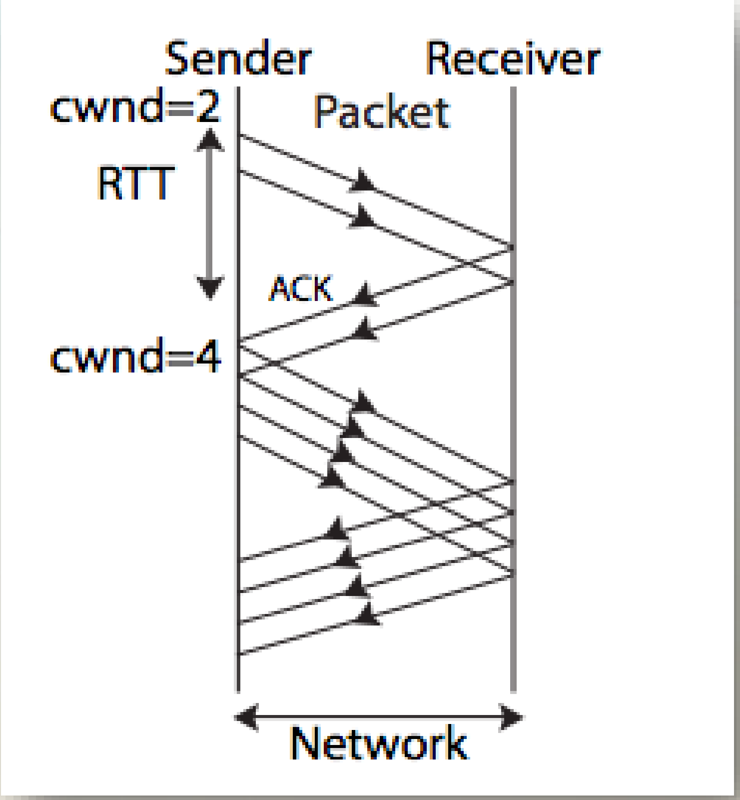
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Δικτυωμένη Ζωή

Αρχές του κατανεμημένου ελέγχου συμφόρησης (συνέχεια)

Προσθετική αύξηση και πολλαπλασιαστική μείωση:

- Στο TCP, όταν το cwnd μεγαλώνει, μεγαλώνει γραμμικά: το cwnd μεγαλώνει κατά 1/cwnd μετά την παραλαβή κάθε επιβεβαίωσης. Αυτό σημαίνει ότι μετά από έναν γύρο ταξίδι, το cwnd μεγαλώνει κατά 1, αν όλα τα ACK παρελήφθησαν κανονικά. Αλλά αν 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 ελεγχόμενος. Θα ήταν πολύ πιο επιθετικός αν συνέβαινε το αντίθετο: **πολλαπλασιαστική αύξηση** και **προσθετική μείωση**.



Ένα διάγραμμα χώρου-χρόνου των πακέτων TCP που στέλνονται και επιβεβαιώνονται. Καθώς δύο επιβεβαιώσεις παραλαμβάνονται από τον αποστολέα, το παράθυρο συμφόρησης δεν σύρεται μόνο, αλλά επίσης και αυξάνει κατά 1.

Αρχές του κατανεμημένου ελέγχου συμφόρησης (συνέχεια)

Εξαγωγή συμπεράσματος συμφόρησης λόγω απώλειας πακέτων ή καθυστέρησης:

Ερωτήσεις:

- Αλλά πώς γνωρίζουμε αν **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;**

Απαντήσεις:

- Οι αρχικές εκδόσεις του ελέγχου συμφόρησης του TCP έκαναν μία σημαντική υπόθεση:
- αν υπάρχει απώλεια πακέτων, υπάρχει συμφόρηση, αλλά μερικές φορές η απώλεια πακέτων οφείλεται σε ένα κακό κανάλι, όπως στις ασύρματες συνδέσεις, και όχι στην 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 συμφόρησης ή όχι συμφόρηση, μία τιμή καθυστέρησης υπονοεί το βαθμό της συμφόρησης.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Δικτυωμένη Ζωή

Αρχές του κατανεμημένου ελέγχου συμφόρησης (συνέχεια)

Εκτίμηση απώλειας πακέτων και καθυστέρησης από χρονοδιακόπτες (1/2):

Ερώτηση: Ας πούμε ότι συμφωνείτε ότι η απώλεια πακέτων ή η καθυστέρηση υπονοεί συμφόρηση, πώς μπορείτε να πείτε αν ένα πακέτο έχει χαθεί και πώς υπολογίζετε την καθυστέρηση;

Απάντηση: Το TCP χρησιμοποιεί **2 προσεγγίσεις κοινής λογικής**:

1η προσέγγιση:

Ερώτηση: Αν ο αποστολέας περιμένει αρκετό χρόνο και δεν έρχεται επιβεβαίωση, πιθανόν το πακέτο να έχει χαθεί. Πόσος είναι ο «αρκετός χρόνος»;

Απάντηση: Ας πούμε είναι τρεις φορές **ο κανονικός χρόνος μετ' επιστροφής** (**round trip time – RTT**) ανάμεσα στον αποστολέα και στον παραλήπτη.

Ερώτηση: Και ποιος είναι ο κανονικός **RTT**;

Απάντηση: 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 ένα κυλιόμενο μέσο όρο του RTT. Ο μικρότερος RTT σε μία περίοδο του χρόνου είναι κατά προσέγγιση ο «κανονικός», χωρίς συμφόρηση RTT.

Αρχές του κατανεμημένου ελέγχου συμφόρησης (συνέχεια)

Εκτίμηση απώλειας πακέτων και καθυστέρησης από χρονοδιακόπτες (2/2):

Ερώτηση: Ας πούμε ότι συμφωνείτε ότι η απώλεια πακέτων ή η καθυστέρηση υπονοεί συμφόρηση, πώς μπορείτε να πείτε αν ένα πακέτο έχει χαθεί και πώς υπολογίζετε την καθυστέρηση;

Απάντηση: Το TCP χρησιμοποιεί **δύο προσεγγίσεις κοινής λογικής**:

2η προσέγγιση:

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**, αλλά αν τρία πακέτα έχουν ήδη παραληφθεί, δεν είναι απλά καθυστερημένο, αλλά απολεσθέν.

Αρχές του κατανεμημένου ελέγχου συμφόρησης (συνέχεια)

Παραλλαγές του ελέγχου συμφόρησης

- Το γεγονός ότι το Διαδίκτυο δεν έχει καταρρεύσει, παρ' όλη την εκτόξευση της ζήτησης, μπορεί εν μέρει να αποδοθεί στις δυνατότητες του ελέγχου συμφόρησης.
- Ο πρώτος μηχανισμός ελέγχου συμφόρησης που προστέθηκε στο TCP το **1998** ονομαζόταν **TCP Tahoe**.
- Το **1990** προστέθηκε ο μηχανισμός **TCP Reno**.
- το **TCP Vegas** το **1995** άλλαξε από τα σήματα συμφόρησης βασισμένα στην απώλεια σε σήματα συμφόρησης βασισμένα στην καθυστέρηση.
- Το **FAST TCP** το **2002** σταθεροποίησε τον έλεγχο συμφόρησης για να επιτύχει υ-ψηλή αξιοποίηση της χωρητικότητας της σύνδεσης.
- Το **CUBIC** το **2005** συνδύασε σήματα συμφόρησης βασισμένα στην απώλεια με σήματα βασισμένα στην καθυστέρηση και τώρα είναι το **προεπιλεγμένο TCP στον πυρήνα του Linux**.
- Υπάρχουν επίσης πολλές άλλες παραλλαγές του ελέγχου συμφόρησης του TCP, που προτάθηκαν τις τελευταίες δύο δεκαετίες.

Εξαγωγή συμπεράσματος συμφόρησης βάσει απώλειας

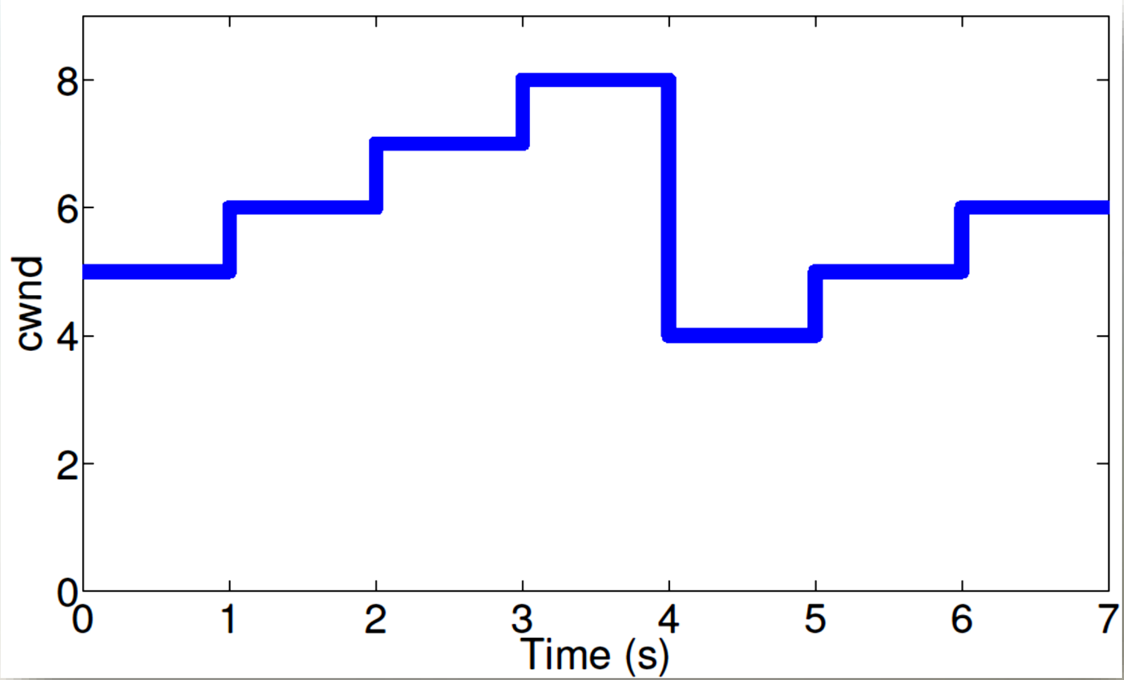
Έλεγχος συμφόρησης βάσει καθυστέρησης, όπως γίνεται στο **TCP Reno**.

- Για από άκρο-σε-άκρο έλεγχο συμφόρησης χωρίς την 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.
- Για τον έλεγχο συμφόρησης βάσει απώλειας, όπως το **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), τότε *αύξησε το **cwnd** κατά 1 για κάθε **RTT***.
  - Αλλιώς, *μείωσέ το κόβοντάς το στη μέση.*
- Υπάρχουν επίσης και άλλα λεπτά χαρακτηριστικά, όπως η **Γρήγορη Αναμετάδοση** και η **Γρήγορη Επαναφορά**

Εξαγωγή συμπεράσματος συμφόρησης βάσει απώλειας (συνέχεια)

Παράδειγμα - TCP Reno

Ας πούμε ότι ***RTT*** = 1 και αρχικοποιούμε το cwnd = 5. Υποθέστε ότι όλα τα πακέτα παραλαμβάνονται επιτυχώς και επιβεβαιώνονται (ACK) κατά τη διάρκεια κάθε RTT, εκτός στο t=4 , όπου συμβαίνει μία απώλεια πακέτου.



Μεγεθυσμένη προβολή της εξέλιξης του cwnd για το TCP Reno με RTT=1 μονάδα χρόνου.

Παρατηρούμε ότι:

– Όταν δεν υπάρχει απώλεια πακέτου (t = 0, 1, 2, 3), το cwnd αυξάνεται γραμμικά.

– Όταν υπάρχει απώλεια πακέτου (t = 4), το cwnd μειώνεται απότομα, μετά ξεκινά πάλι να αυξάνεται γραμμικά (t = 5, 6).

Εξαγωγή συμπεράσματος συμφόρησης βάσει καθυστέρησης

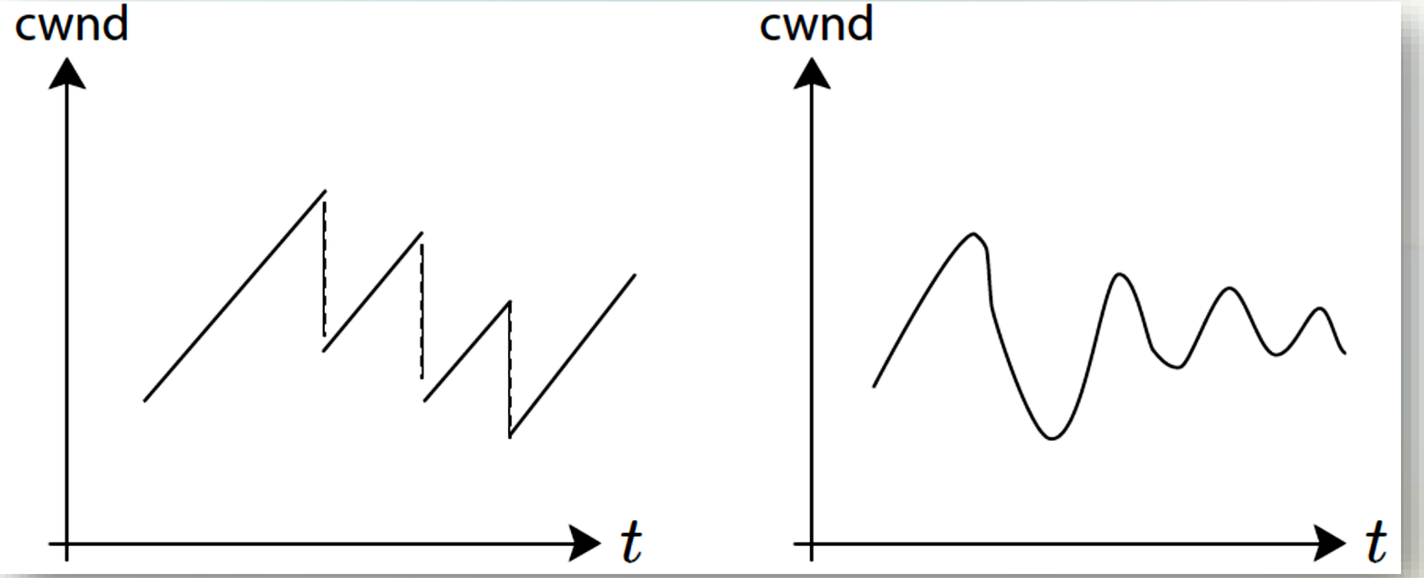
Έλεγχος συμφόρησης βάσει καθυστέρησης, όπως γίνεται στο **TCP Vegas**.

- Το συνολικό RTT αποτελείται κυρίως από την καθυστέρηση διάδοσης.
- Όσο πιο μεγαλύτερη είναι η συμφόρηση, τόσο μεγαλύτερη είναι η αναμονή. Έτσι ο αποστολέας πρέπει να εκτιμήσει το $\boldsymbol{RTT_{min}}$, το ελάχιστο ***RTT*** που λέει στον αποστολέα ποια θα είναι η τιμή της καθυστέρησης αν (σχεδόν) δεν υπάρχει συμφόρηση.
- Μετά, αφού λάβει κάθε μία επιβεβαίωση, ο αποστολέας κοιτάει τη διαφορά μεταξύ $\frac{\boldsymbol{cwnd}}{\boldsymbol{RTT_{min}}}$ και $\frac{\boldsymbol{cwnd}}{\boldsymbol{RTT_{max}}}$. 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.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Δικτυωμένη Ζωή

Σύγκριση των 2 ελέγχων συμφόρησης

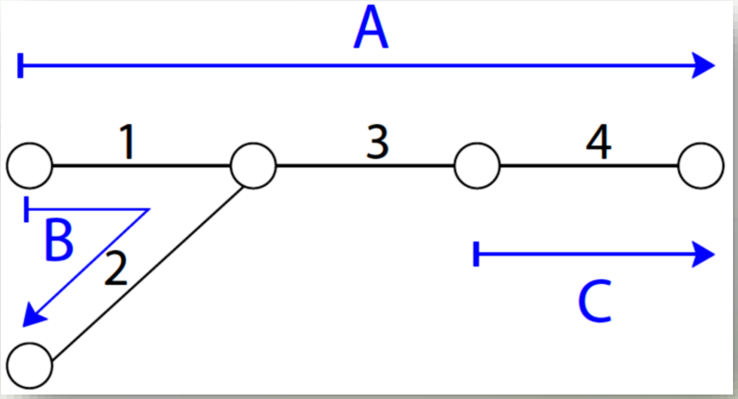


*Τυπικές εξελίξεις των τιμών του cwnd στο **TCP Reno** στα αριστερά και στο **TCP Vegas** στα δεξιά.*

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 τείνει να είναι μικρότερη στο Vegas, αν οι παράμετροι είναι κατάλληλα συντονισμένοι.

Το πρόβλημα της κατανομής χωρητικότητας

Ερώτηση: Πώς μπορούμε να κατανέμουμε τη χωρητικότητα κάθε σύνδεσης, έτσι ώστε οι σύνοδοι να χρησιμοποιούν όσο το δυνατόν περισσότερη συνολική χωρητικότητα, χωρίς να προκαλούν συμφόρηση και ο ανταγωνισμός τους να συγχρονίζεται δίκαια;



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

- 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 (0.5, 0.5, 0.5) για τις τρεις συνόδους A, B και C, σε Mbps. Σε αυτή την ίση κατανομή των απο 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 το (1, 0, 0), η οποία αφήνει τις συνόδους B και C χωρίς εξυπηρέτηση και ίσως να μη φαίνεται σαν δίκαιη κατανομή. Αποδεικνύεται ότι μία τυπική γνώμη της ίσης μεταχείρισης θα έδινε την κατανομή (1/3, 2/3, 2/3) στις τρεις ανταγωνιστικές συνόδους, καθώς η σύνοδος A διασχίζει δύο συνδέσεις, που είναι πιθανά σημεία συμφόρησης, ενώ οι σύνοδοι B και C διασχίζουν μόνο μία τέτοια σύνδεση.

Διατύπωση του προβλήματος NUM

Το πρόβλημα **NUM** (**Network Utility Maximization**) είναι το **Βασικό Πρόβλημα Μεγιστοποίησης Χρησιμότητας Δικτύου**. Στη **μοντελοποίηση του ελέγχου συμφόρησης** χρειάζεται να διευθετηθούν 2 θέματα:

– **Θέμα 1ο: Πώς μετράμε την επίδοση και τη δικαιοσύνη**;

Απάντηση:

- Χρησιμοποιούμε τις συναρτήσεις ωφελείας 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 από τις συναρτήσεων ωφέλειας, όπως οι αδίκαιες συναρτήσεις ωφέλειας.

– **Θέμα 2ο: Πώς αναπαριστούμε τον περιορισμό της χωρητικότητας της σύνδεσης**;

Απάντηση:

- Για κάθε σύνδεση *l* υπάρχει μία περιορισμένη χωρητικότητα c_l σε bps.
- Το φορτίο θα πρέπει να είναι μικρότερο από το c_l.

Διατύπωση του προβλήματος NUM (συνέχεια)

Μπορούμε να γράψουμε το φορτίο στη σύνδεση ***l*** ως το άθροισμα των πηγαίων ρυθμών $\boldsymbol{x_i}$ διαμέσου αυτών των πηγών χρησιμοποιώντας αυτή τη σύνδεση:

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

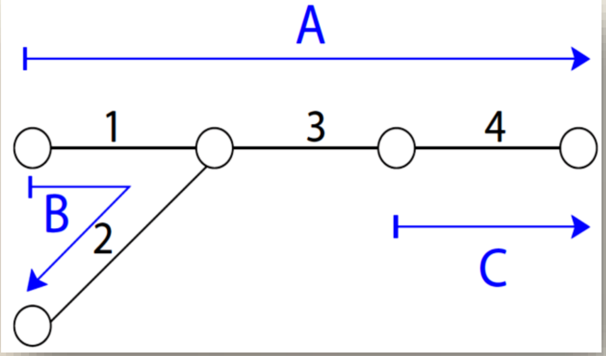
Ή μπορούμε να χρησιμοποιήσουμε το R_{li} ως ένα δυαδικό δείκτη, έτσι ώστε R_{li}=0, εάν η σύνοδος της πηγής *i* διασχίζει τη σύνδεση *l*, και R_{li}=0 διαφορετικά.

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

Σε αυτή την παράσταση, μπορείτε εύκολα να δείτε ότι οι περιορισμοί είναι ισοδύναμοι με τις παρακάτω γραμμικές ανισότητες σε σημειολογία πίνακα: $\mathbf{R_X} \leq \mathbf{c}$,

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

Για παράδειγμα, στη τοπολογία του δικτύου στο σχήμα, ο περιορισμός της χωρητικότητας της σύνδεσης γίνεται σε μορφή πίνακα:



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

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 συμφόρησης:

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

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.
- Μπορεί να διασπαστεί σε πολλά μικρότερα, με τέτοιο τρόπο ώστε η λύση τους να είναι ισότιμη με τη λύση του αρχικού προβλήματος.

Κατανεμημένος αλγόριθμος που λύνει το NUM (συνέχεια)

- 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

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

D: η συνάρτηση ζήτησης

- Σε μία εφαρμογή βάσει συρόμενου παραθύρου, η πηγή προσαρμόζει το μέγεθος παραθύρου συμφόρησης αντί για το x_i απευθείας.
- Η τιμή της διαδρομής υπηρετεί ως το ***σήμα ανάδρασης συμφόρησης*** από το δίκτυο.
- Την ίδια στιγμή, ο δρομολογητής σε κάθε σύνδεση *l* ανανεώνει την «τιμή» σε αυτή τη σύνδεση:

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

όπου y_l είναι το συνολικό φορτίο στη σύνδεση *l*: $y_l[t] = \sum_{l \in S(l)} x_l[t]$

το $\{\ldots\}^+$ σημαίνει ότι, αν η έκφραση μέσα στις αγκύλες παίρνει αρνητική τιμή, τότε αυτή επιστρέφει την τιμή 0.

Κατανεμημένος αλγόριθμος που λύνει το NUM (συνέχεια)

- 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

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

D: η συνάρτηση ζήτησης

- Σε μία εφαρμογή βάσει συρόμενου παραθύρου, η πηγή προσαρμόζει το μέγεθος παραθύρου συμφόρησης αντί για το x_i απευθείας.
- Η τιμή της διαδρομής υπηρετεί ως το ***σήμα ανάδρασης συμφόρησης*** από το δίκτυο.
- Την ίδια στιγμή, ο δρομολογητής σε κάθε σύνδεση *l* ανανεώνει την «τιμή» σε αυτή τη σύνδεση:

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

όπου y_l είναι το συνολικό φορτίο στη σύνδεση *l*: $y_l[t] = \sum_{l \in S(l)} x_l[t]$

το $\{\ldots\}^+$ σημαίνει ότι, αν η έκφραση μέσα στις αγκύλες παίρνει αρνητική τιμή, τότε αυτή επιστρέφει την τιμή 0.

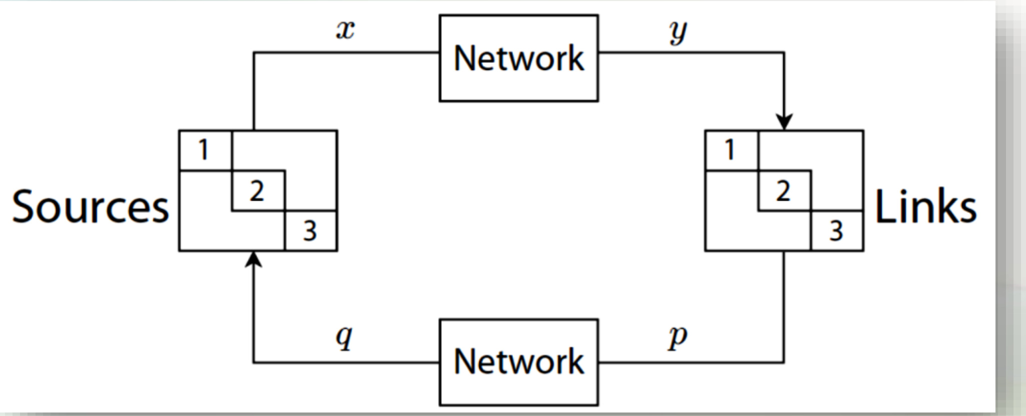
Κατανεμημένος αλγόριθμος που λύνει το NUM (συνέχεια)

Ο βρόχος ανάδρασης ελέγχου σε μία κατανεμημένη λύση του NUM στο ζεύγος των εξισώσεων

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

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

απεικονίζεται στο διπλανό σχήμα:



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.

- Αν, στον χρόνο t, υπάρχει περισσότερο φορτίο από τη χωρητικότητα στη σύνδεση *l*, τότε η τιμή p_l θα αυξηθεί (14.3) και οι τιμές όλων των διαδρομών, που περιλαμβάνουν τη σύνδεση *l*, θα αυξηθούν στην επόμενη χρονοθυρίδα *t*+1.
- Μία υψηλότερη τιμή θα μειώσει τη ζήτηση (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.

Κατανεμημένος αλγόριθμος που λύνει το NUM (συνέχεια)

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

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

Όπως φαίνεται 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.

Όπως φαίνεται 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.

Αντίστροφη Μηχανική

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***”.

1ο Παράδειγμα αντίστροφης μηχανικής:

- Καθώς η εφαρμογή του TCP Reno λύνει το Βασικό Πρόβλημα Μεγιστοποίησης Χρησιμότητας Δικτύου (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 NUM. 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.

Αντίστροφη Μηχανική (συνέχεια)

2ο Παράδειγμα αντίστροφης μηχανικής:

- 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.

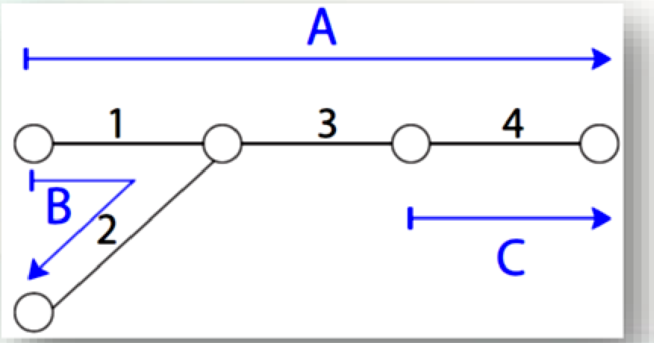
2ο Παράδειγμα αντίστροφης μηχανικής:

- 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.

Παραδείγματα

Θεωρήστε το δίκτυο που φαίνεται στο διπλανό σχήμα. 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 .

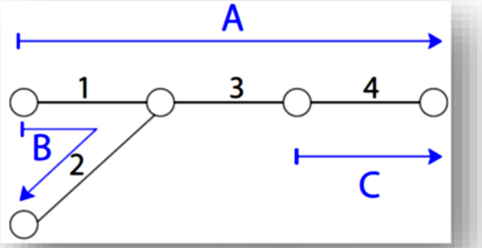
Παραδείγματα (συνέχεια)

Το πρόβλημα NUM παίρνει τη μορφή:

$$\text{μεγιστοποιήστε} \quad \log(x_A)+\log(x_B)+\log(x_C)$$

$$\text{υπό τον περιορισμό} \quad \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_A \\ X_B \\ X_C \end{bmatrix} \leq \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}, \qquad (14.4)$$

$$x_i \geq 0, \forall i.$$



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

$$x_i[t] = U'^{-1}(q_i[t]) = \frac{1}{q_i[t]},$$

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

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*)

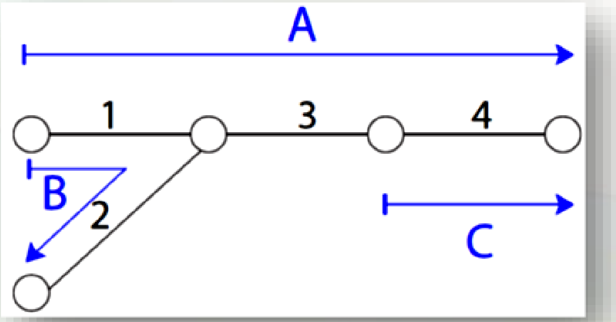
$$q_i[t] = \sum_{l \in L(i)} p_l(t-1),$$

$$y_l[t] = \sum_{l \in L(l)} x_i(t).$$

Παραδείγματα (συνέχεια)

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 *β* = 1.



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

$$q_A[1] = p_1[0] + p_3[0] + p_4[0] = 1+1+1 = 3$$

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

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

έχουμε, σε Mbps,

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

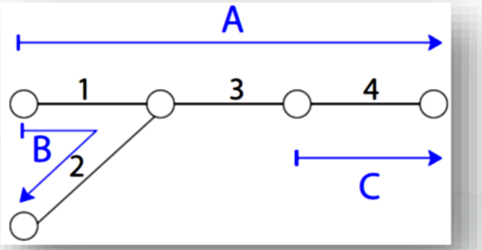
$$x_B[1] = \frac{1}{q_B[1]} = 0,5$$

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

Παραδείγματα (συνέχεια)

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

$$p_1[1] = [p_1[0] + y_1[1] - c]^+ = [1 + 0,83 - 1]^+ = 0,833,$$
$$p_2[1] = [p_2[0] + y_2[1] - c]^+ = [1 + 0,5 - 1]^+ = 0,5,$$
$$p_3[1] = [p_3[0] + y_3[1] - c]^+ = [1 + 0,33 - 1]^+ = 0,333,$$
$$p_4[1] = [p_4[0] + y_4[1] - c]^+ = [1 + 1,33 - 1]^+ = 1,33.$$

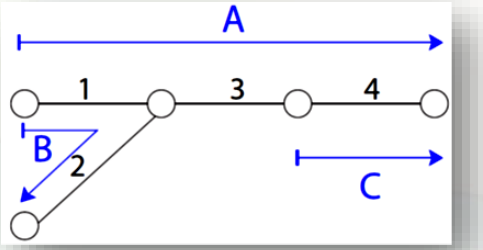
Παραδείγματα (συνέχεια)

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,

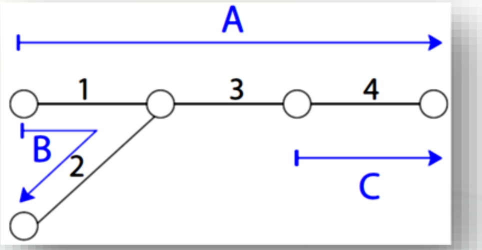
$$x_A[2] = \frac{1}{q_A[2]} = 0,4\,,$$

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

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

Παραδείγματα (συνέχεια)

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

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

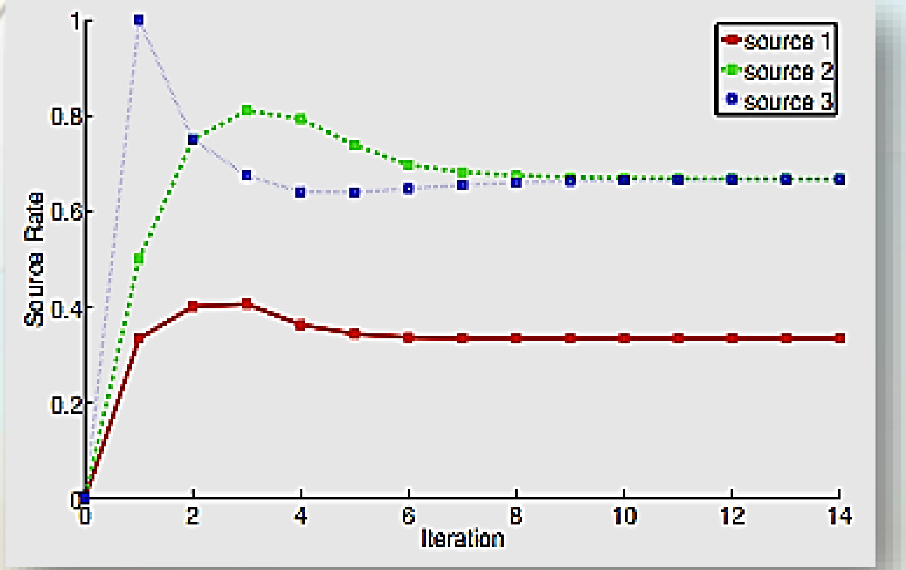
$$p_2[2] = [p_2[1] + y_2[2] - c]^+ = [0,5 + 0,75 - 1]^+ = 0,25\,,$$

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

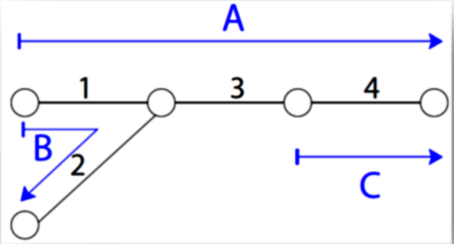
$$p_4[2] = [p_4[1] + y_4[2] - c]^+ = [1,33 + 1,15 - 1]^+ = 1,48\,.$$

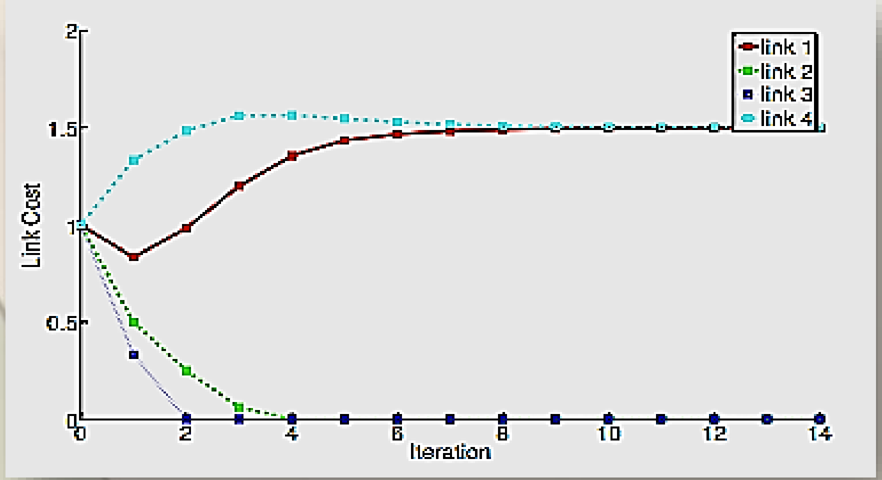
Παραδείγματα (συνέχεια)

Οι επαναλήψεις συνεχίζονται. Απεικονίζουμε την εξέλιξή τους κατά τη διάρκεια του χρόνου στα παρακάτω σχήματα:



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





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

Παραδείγματα (συνέχεια)

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

$$x_A^* = 0,33 Mbps\,,$$
$$x_B^* = 0,67 Mbps\,,$$
$$x_C^* = 0,67 Mbps\,.$$

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 κατανεμημένο 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

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

Στην ισορροπία, οι τιμές των συνδέσεων είναι

$$p_1^* = 1,5\,, \quad p_2^* = 0\,, \quad p_3^* = 0\,, \quad p_4^* = 1,5\,.$$

Προχωρημένο Υλικό

Διάσπαση του NUM

- 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 **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 περίπτωσή μας, να λυθεί με έναν κατανεμημένο τρόπο.

Προχωρημένο Υλικό

Διάσπαση του NUM (συνέχεια)

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 p**, και μπορεί να ερμηνευτούν ως τιμές των συνδέσεων:

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

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 **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: **p**

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

Προχωρημένο Υλικό

Διάσπαση του NUM (συνέχεια)

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*:

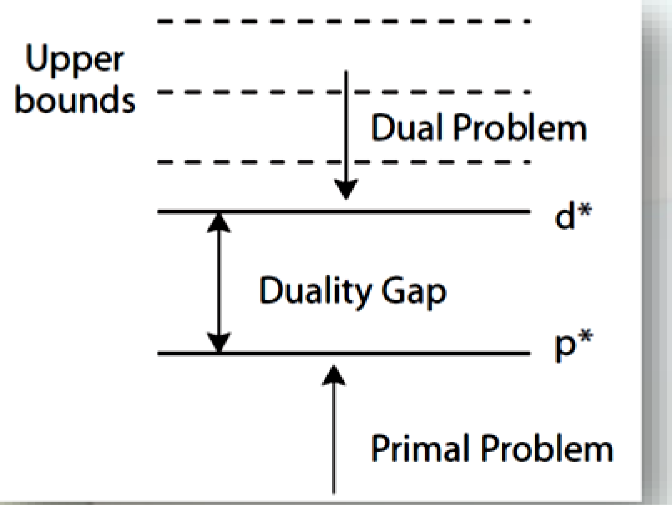
$$\sum_{l} \sum_{i \in S(l)} p_i x_i ,$$

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

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

Προχωρημένο Υλικό

Διάσπαση του NUM (συνέχεια)



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 παραπάνω σχήματος έχουμε

$$
\begin{aligned}
&(x_A, x_B, x_C, p_1, p_2, p_3, p_4) = U_A(x_A) - (p_1 + p_3 + p_4)x_A + U_B(x_B) - (p_1 + p_2)x_B \\
&+U_C(x_C) - p_4 x_C + c_1 p_1 + c_2 p_2 + c_3 p_3 + c_4 p_4 \\
&(x_A, x_B, x_C, p_1, p_2, p_3, p_4) = U_A(x_A) - (p_1 + p_3 + p_4)x_A + U_B(x_B) - (p_1 + p_2)x_B \\
&+U_C(x_C) - p_4 x_C + c_1 p_1 + c_2 p_2 + c_3 p_3 + c_4 p_4 \\
&L(x_A, x_B, x_C, p_1, p_2, p_3, p_4) = U_A(x_A) - (p_1 + p_3 + p_4)x_A + U_B(x_B) - (p_1 + p_2)x_B \\
&+U_C(x_C) - p_4 x_C + c_1 p_1 + c_2 p_2 + c_3 p_3 + c_4 p_4 .
\end{aligned}
$$

Προχωρημένο Υλικό

Διάσπαση του NUM (συνέχεια)

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 **x** μπορεί να εξαχθεί ανεξάρτητα από κάθε πηγή *i* (δείτε την τετράγωνη παρένθεση μέσα στο παρακάτω):

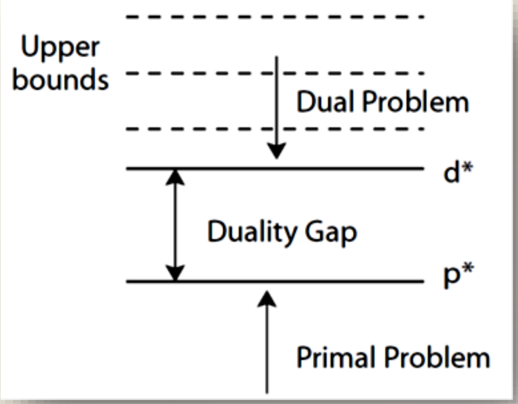
$$L(\mathbf{x},\mathbf{p}) = \sum_i \left[U_i(x_i) - q_i x_i \right] + \sum_l c_l p_l \, .$$

Υποθέστε τώρα ότι μεγιστοποιούμε τη Λαγκραζιανή πάνω στις πρωτότυπες μεταβλητές **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 **p** χρησιμοποιήσαμε.

Έτσι, πρέπει να σημειώσουμε την τιμή του αποτελέσματος ως συνάρτηση του **p**:

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



Αυτή η συνάρτηση *g(p)* ονομάζεται **δυϊκή συνάρτηση Lagrange**.

Προχωρημένο Υλικό

Διάσπαση του NUM (συνέχεια)

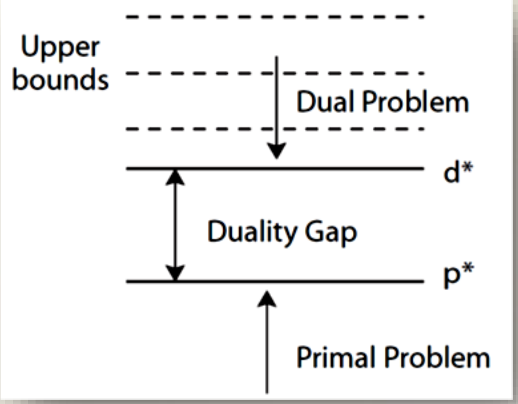
Αποδεικνύεται ότι όποιο **p** και να χρησιμοποιήσουμε (αρκεί να μην είναι αρνητικό), η *g*(**p**) παρέχει πάντα ένα όριο στην επίδοση. Αυτό είναι ένα άνω όριο στο μέγιστο $U^* = \sum_i U_i\left(x_i^*\right)$ του πρωτότυπου προβλήματος NUM. Αυτό είναι εύκολο να το δούμε.

Θεωρήστε το μεγιστοποιητή του προβλήματος NUM **x***. Πρέπει να είναι ένα εφικτό διάνυσμα και να ικανοποιεί τον περιορισμό ανισότητας. Επίσης το **p** είναι μη αρνητικό. Έτσι η Λαγκραζιανή L(14.5) πρέπει να είναι τουλάχιστον μεγάλη όσο το *U**, όταν **x** = **x***. Καθώς η δυική συνάρτηση Lagrange *g* είναι η μεγαλύτερη Λαγκραζιανή από όλα τα **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*(**p**), με το να επιλέξουμε το καλύτερο **p**; Ονομάζουμε το πρόβλημα που προκύπτει πρόβλημα δυϊκότητας Lagrange και ονομάζουμε τις δυϊκές μεταβλητές Lagrange τώρα στο **p**:

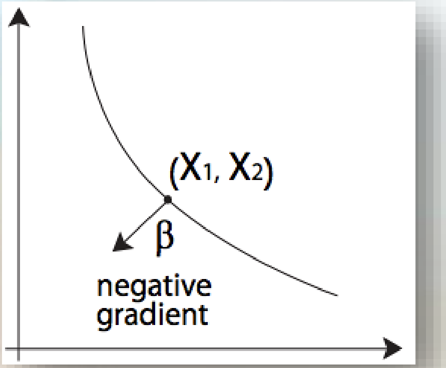
$$\varepsilon\lambda\alpha\chi\iota\sigma\tau o\pi o\iota\acute{\eta}\sigma\tau\varepsilon_{\mathbf{p}}\, g\left(\mathbf{p}\right).$$



Προχωρημένο Υλικό

Διάσπαση του NUM (συνέχεια)

- Όπως φαίνεται στο παρακάτω σχήμα, 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.



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 **p** (δηλ. η μεγιστοποίηση της χρησιμότητας του δικτύου (net utility) κατανεμημένα εξαγόμενη σε κάθε πηγή:

$$x_i^*(\mathbf{p}) = \mathrm{argmax}[U_i(x_i) - q_i x_i]$$

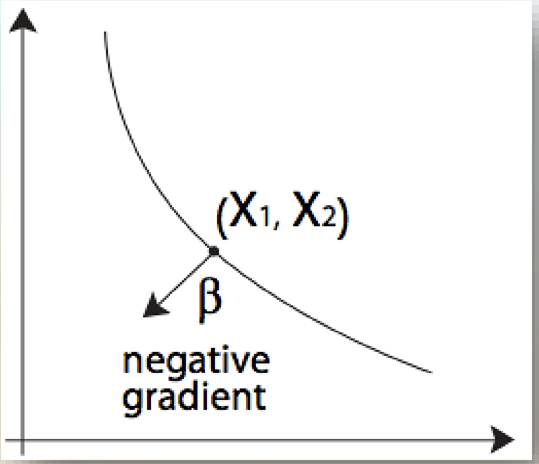
Λαμβάνουμε ακριβώς την (14.2).

Προχωρημένο Υλικό

Διάσπαση του NUM (συνέχεια)

Όπως φαίνεται στο παρακάτω σχήμα, 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.



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 **p**.

Προχωρημένο Υλικό

Διάσπαση του NUM (συνέχεια)

Το δεύτερο βήμα, που είναι η ελαχιστοποίηση της αντικειμενικής συνάρτησης του δυϊκού προβλήματος Lagrange $g(\mathbf{p}) = L\left(x^*(\mathbf{p}), \mathbf{p}\right)$, ως προς το **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 3, όπως φαίνεται στο παραπάνω σχήμα, έχουμε

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

Αποδεικνύεται ότι για ένα γραμμικά περιορισμένο, κοίλο πρόβλημα μεγιστοποίησης όπως το NUM, η ίδια η συνάρτηση περιορισμού $c_l - \sum_{i \in S(l)} x_i = c_l - y_l$ είναι η κλίση για κάθε p_i. Έτσι το μόνο που χρειάζεται είναι να πολλαπλασιάσουμε την κλίση με ένα βήμα μεγέθους 3 (και μετά να σιγουρευτούμε ότι δεν είναι ποτέ αρνητική):

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

που είναι ακριβώς η (14.3).

Προχωρημένο Υλικό

Διάσπαση του NUM (συνέχεια)

Καθώς η ισχυρή δυϊκότητα διατηρείται στο NUM, η λύση του δυικού 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 **x***) και οι δυϊκές μεταβλητές (το διάνυσμα τιμών απόστασης **p***) ικανοποιούν επίσης χρήσιμες ιδιότητες, συμπεριλαμβανομένης της παρακάτω συμπληρωματικής ιδιότητας χαλαρότητας. Αν ο αρχικός περιορισμός *l* είναι χαλαρός,

$$\text{δηλαδή } \sum_{i \in S(l)} x_i^* < c_l \,,$$

η αντίστοιχη βέλτιστη δυική μεταβλητή p_l^* (η βέλτιστη τιμή συμφόρησης συνδέσεων) πρέπει να είναι 0, δηλαδή ο περιορισμός της μη-αρνητικότητας στο δυικό 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.

Προχωρημένο Υλικό

Αντίστροφη Μηχανική

- Αναφέρθηκε πρωτύτερα ότι εάν δωθεί ένα πρωτόκολλο ελέγχου συμφόρησης του TCP, μπορεί να σας επιστραφεί 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.
- Παρακάτω θα απεικονιστεί αυτά τα κύρια βήματα σε αυτή την προσέγγιση αντίστροφης μηχανικής στο TCP Reno.
- Το **πρώτο βήμα** 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 (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:

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

Προχωρημένο Υλικό

Αντίστροφη Μηχανική (συνέχεια)

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*:

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

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:

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

Προχωρημένο Υλικό

Αντίστροφη Μηχανική (συνέχεια)

- 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

$$1-\prod_{l}(1-p_l)$$

δηλαδή 1 μείον την πιθανότητα να μην έχει χαθεί κανένα πακέτο σε καμία σύνδεση.

Αλλά όταν ο ρυθμός απώλειας είναι μικρός, η $q_i = 1-\prod_{l}\left(1-p_l\right) \approx \sum_{l\in L(i)} p_l$ λειτουργεί πολύ καλά ως προσέγγιση.

Προχωρημένο Υλικό

Αντίστροφη Μηχανική (συνέχεια)

- Εστιάσαμε 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.
- 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

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.

Τέλος Κεφαλαίου 14
Ερωτήσεις;;;