Overview

Το end-to-end argument υποστηρίζει ότι κάποιες από τις λειτουργίες που επιτελούνται κατά την διάρκεια μιας επικοινωνίας μεταξύ συστημάτων θα πρέπει να μην υλοποιούνται σε χαμηλό επίπεδο (επίπεδο συστήματος επικοινωνιών) αλλά σε υψηλότερο επίπεδο, δηλαδή στις εφαρμογές που επικοινωνούν. Η γνώση που έχουν οι εφαρμογές αυτές είναι αρκετή για να υλοποιηθούν αυτές οι εξειδικευμένες λειτουργίες.

End-to-end επικοινωνία χωρίς σφάλματα.

Στόχος είναι να μεταφέρουμε ένα αρχείο χωρίς σφάλματα.

Ο έλεγχος στο επίπεδο συστήματος επικοινωνιών εξασφαλίζει ότι έχουμε μεταφορά στο δίκτυο χωρίς λάθη.

Όμως δεν εγγυάται για

- Λάθη μεταφοράς από τον σκληρό στο λειτουργικό σύστημα και αντίστροφα
- Λάθη που οφείλονται σε σφάλμα του σκληρού δίσκου.
- Λάθη hardware κατά την μεταφορά των δεδομένων πριν αυτά βγουν στο δίκτυο.
- Retries για αυτά τα λάθη.

Λύση : Πρέπει να φυλάσσουμε κάποιο checksum μαζί με το αρχείο στον σκληρό.
Ο παραλήπτης αφού λάβει το αρχείο θα υπολογίζει το checksum και αν είναι τα δύο checksum είναι ίδια, η μετάδοση θα είναι επιτυχής. Αυτό αναγκαστικά γίνεται από την εφαρμογή.

Επιχείρημα : *Αφού έτσι και αλλιώς έχουμε δικό μας checksum και ελέγχουμε μόνοι μας για την ακεραιότητα των δεδομένων δεν χρειάζεται το communications system να κάνει και αυτό έλεγχο για σωστή μεταφορά δεδομένων.*

Πρόβλημα : Πλέον χρειαζόμαστε και δική μας συνάρτηση που να κάνει retry δεδομένα που ήρθαν με λάθη. Αυτή που παρέχει το communication system δεν επαρκεί γιατί κάνει retry μόνο δεδομένα που προήρθαν από λάθη κατά την μετάδοση στο δίκτυο.

Επιχείρημα : *Έτσι δεν χρειάζεται πλέον να κάνει retries το communication system ! Αφού έτσι και αλλιώς υλοποιούμε δικιά μας συνάρτηση που κάνει retry σε περίπτωση λάθους.*

Συμπέρασμα : Για να πετύχουμε αλάνθαστη και προσεκτική μεταφορά αρχείου, η εφαρμογή πρέπει να παρέχει έναν ΕΞΕΙΔΙΚΕΥΜΕΝΟ και έμπιστο τρόπο μεταφοράς. Αυτό αφορά την εισαγωγή ενός checksum και έναν σχεδιασμό για retry/commit. Το επίπεδο δικτύου δεν μας απαλλάσσει από το να υλοποιήσουμε έτσι και αλλιώς αυτές τις λειτουργίες και στις end-to-end εφαρμογές μας.

Άρα οι λειτουργίες αυτές μπορούν να φύγουν από το επίπεδο δικτύου και να «ανεβούν» στο επίπεδο εφαρμογών.

Performance aspects.

Το επίπεδο δικτύου σπάει το αρχείο σε πολλά μικρά πακέτα.
Αν ένα από αυτό δεν μεταδοθεί σωστά τότε μόνο ένα μικρό πακέτο θα επαναμεταδοθεί.
Ενώ με την end-to-end μέθοδο θα πρέπει να μεταδοθεί ξανά ολόκληρο το αρχείο.

Έτσι εκ πρώτης όψεως φαίνεται ότι το να βάλουμε την λειτουργία αυτή σε χαμηλό επίπεδο θα επιφέρει βελτίωση στην ταχύτητα.

Αυτό όμως δεν είναι αλήθεια.
Μπορούμε να επιτύχουμε την ίδια ή και καλύτερη απόδοση αν μεταφέρουμε αυτή την λειτουργία ψηλά για δύο λόγους

1. Αν η λειτουργία είναι ενσωματωμένη στο δίκτυο και οι επικοινωνίες που δεν την χρειάζονται θα είναι αναγκασμένες να την χρησιμοποιούν.
2. Το κατώτερο σύστημα δεν μπορεί να έχει αρκετή πληροφορία για την μετάδοση έτσι δεν μπορεί να μεταφέρει τα δεδομένα όσο το δυνατόν πιο αποτελεσματικά.

Στο παράδειγμα της μεταφοράς αρχείων που αναφέραμε πριν

- Πρέπει έτσι και αλλιώς να χρησιμοποιήσουμε αυτούς του ελέγχους/retries
- Μπορούμε από μόνοι μας να σπάσουμε το αρχείο σε μικρά πακέτα με checksums (στην εφαρμογή) για να πετύχουμε ίδια απόδοση.

Πρόβλημα: Το να αφαιρέσουμε αυτές τις λειτουργίες από το επίπεδο δικτύου μπορεί να δημιουργήσει σε αύξηση κόστους αφού οι υπόλοιπες εφαρμογές πρέπει να «υλοποιήσουν» και αυτές δικούς τους ελέγχους (δεν προσφέρονται πλέον από το δίκτυο).

Αλλά παραδείγματα end-to-end argument

Delivery guarantees

- Το να ξέρουμε ότι ένα μήνυμα παραδόθηκε δεν είναι σημαντικό. Θέλουμε να ξέρουμε ότι η εφαρμογή έλαβε το μήνυμα και το ότι *το* **χειρίστηκε σωστά !**
- Έτσι χρειάζεται επιβεβαίωση και από την εφαρμογή ότι το μήνυμα λήφθηκε υπόψη (για παράδειγμα μπορούσε να ληφθεί σωστά αλλά να απορριφθεί από την εφαρμογή η αίτηση.)
- Οπότε και η επιβεβαίωση δεν χρειάζεται να γίνει σε χαμηλά επίπεδα

Secure transmission of data

- Θέλουμε να έχουμε κρυπτογράφηση σε επίπεδο εφαρμογής και όχι σε επίπεδο δικτύου για τρεις λόγους
  - Δεν εμπιστευόμαστε την κρυπτογράφηση του δικτύου
  - Η πληροφορία δεν θα είναι κρυπτογραφημένη μέχρι να φτάσει στο δίκτυο (εσωτερικά του συστήματος. π.χ. στην μνήμη)
  - Η αυθεντικότητα ενός μηνύματος πρέπει έτσι και αλλιώς να ελεγχθεί από την εφαρμογή.

- **Πρόβλημα**: Με το να καταργήσουμε την κρυπτογράφηση στο δίκτυο και να χρησιμοποιούμε κρυπτογράφηση μόνο στην εφαρμογή: Κάποιος χρήστης μπορεί να στείλει μέσω κάποιας δικιάς του εφαρμογής πληροφορίες από το σύστημα που δεν είναι κρυπτογραφημένες, ίσως και χωρίς να το ξέρει. Άλλα αυτό αποτελεί διαφορετική απαίτηση.

Duplicate message suppression

- Παρόλο που το δίκτυο ξεχωρίζει τα «δικά του» διπλά μηνύματα, και η ίδια η εφαρμογή μπορεί να προκαλέσει διπλά μηνύματα λόγω των δικών της λειτουργιών failure/retry. Τα μηνύματα αυτά είναι διαφορετικά όσον αφορά το δίκτυο αλλά ίδια για την εφαρμογή.
- Οπότε αφού έτσι και αλλιώς η εφαρμογή θα αντιμετωπίσει διπλά μηνύματα μπορεί να αντιμετωπίσει και τα διπλά μηνύματα που προέρχονται από το δίκτυο.

Guaranteed FIFO message delivery.

Θέλουμε να εξασφαλίσουμε ότι τα μηνύματα φτάνουν με την σωστή σειρά. Αυτό γίνεται ως τώρα στο επίπεδο δικτύου.

- Μία κατανεμημένη εφαρμογή μπορεί με ένα request να προκαλέσει μία σειρά από ενέργειες σε διάφορες άλλες μηχανές (όχι μια) που πρέπει να γίνουν με κάποια σειρά.
- Το επίπεδο δικτύου δεν έχει τέτοια πληροφορία (να βάλει σε σειρά λειτουργίες σε διαφορετικές μηχανές) οπότε αυτή η περίπτωση πρέπει να αντιμετωπιστεί από το υψηλότερο επίπεδο της εφαρμογής.

Transaction management

Όταν θέλουμε να προσπελάσουμε ένα δεδομένο στέλνουμε ένα μήνυμα με την διεύθυνση του δεδομένου, τον τύπο της προσπέλασης (ανάγνωση / εγγραφή), και έναν αύξων αριθμό.

Δεν χρειαζόμαστε να ελέγχουμε πάντα για διπλά μηνύματα στο επίπεδο δικτύου

- Στην περίπτωση της εγγραφής ο αύξων αριθμός και ο παραλήπτης είναι αρκετά στοιχεία για την ανίχνευση.
- Στην αίτηση για ανάγνωση δεν χρειάζεται έλεγχος (απλά θα ξανά-στείλουμε κάτι...).

Επίσης δεν χρειάζεται πάντα και επιβεβαίωση παραλαβής από το επίπεδο δικτύου

- Στην περίπτωση αίτησης ανάγνωσης το ίδιο το δεδομένο που ζητήθηκε είναι η απάντηση.

Identifying the ends

Χρειάζεται ανάλυση απαιτήσεων της εφαρμογής για να χρησιμοποιηθεί το end-to-end argument.
Για παράδειγμα

- Voice packet communication
  - Χρειάζεται μεταφορά ενός αρχείου χωρίς καθυστέρηση.
  - Θέλουμε FIFO μεταφορά.
  - Αν ένα μήνυμα δεν φτάσει σωστά προτιμάμε να το δεχθούμε ΛΑΝΘΑΣΜΕΝΟ παρά να περιμένουμε να γίνει επαναμετάδωση καθυστερώντας και όλα τα υπόλοιπα πακέτα (FIFO). Αυτό γιατί σε μία ζωντανή συνομιλία είναι πολύ πιθανό ο συνομιλητής αν δεν κατάλαβε κάτι να ξανά-ρωτήσει.
- Voice mail
  - Εδώ έχουμε χρόνο για επαναμεταδώσεις.
  - Εδώ απαιτούμε τα δεδομένα να είναι ακέραια γιατί ο ομιλητής που άφησε το μήνυμα δεν θα το επαναλάβει.

Conclusions

- Ο σχεδιαστής ενός συστήματος μπορεί να μπει στον πειρασμό να απαλλάξει τον χρήστη από κόπο, με το να υλοποιήσει πολλές, γενικές λειτουργίες, σε χαμηλό επίπεδο. Το τελικό όμως σύστημα θα είναι καλύτερο αν χρησιμοποιηθούν εξειδικευμένες λειτουργίες που θα προκύψουν από την γνώση των end-to-end arguments.
- Μπορούν να χρησιμοποιηθούν τα λεγόμενα layers πρωτόκολλα δικτύου. Ως τώρα δεν έχουν οριστεί τα κριτήρια για την ανάθεση λειτουργιών σε κάθε layer. Τα end-to-end arguments μπορούν να θεωρηθούν σαν ένα σύνολο από αρχές για την οργάνωση αυτών των layer έτσι ώστε να αυξηθεί η ευελιξία του επίπεδου επικοινωνιών.

Arguments against

- Υποχρεώνουμε όλες τις εφαρμογές και όλους τους χρήστες να φτιάχνουν συναρτήσεις ελέγχου ακόμα και αν τους έκανε ο μη εξειδικευμένος έλεγχος που προσέφερε το επίπεδο δικτύου.
- Το επιχείρημα που αναφέρεται και από τους συγγραφείς που αφορά την ασφάλεια. Όταν χρησιμοποιεί μόνο η εφαρμογή encryption ότι φεύγει από άλλες εφαρμογές είναι unencrypted. Οπότε μπορούν να φύγουν από άλλες εφαρμογές στοιχεία για το σύστημα που δεν θα θέλαμε να διαρρεύσουν.
- Χρειάζεται πολύ προσεκτικός σχεδιασμός ώστε να αναλυθούν πλήρως οι απαιτήσεις που έχει το δικό μας σύστημα για end-to-end argument (κόστος σχεδιασμού αυξάνεται).
- Υπάρχει πολύ μεγαλύτερο κόστος υλοποίησης της κάθε εφαρμογής.
- Χρειάζεται πολύ κόπος για να γίνει η εφαρμογή το ίδιο γρήγορη με αυτή χωρίς end-to-end argument.
- Είναι δύσκολο να εντοπίσουμε που ακριβώς γίνονται τα σφάλματα (δίκτυο, λειτουργικό, εφαρμογή)

Παράδειγμα συστημάτων που παραβιάζουν το end-to-end argument.

- TCP based applications.
- Mail.

Το NFS είναι ΣΥΜΒΑΤΟ με end-to-end αφού χρησιμοποιεί UDP και έχει δικές του μεθόδους για error corrections κτλ...