

## Άσκηση 2

### Ανάλυση του paper

#### α) Περίληψη και βασικά σημεία

##### Θέμα

Το θέμα του paper είναι ο αποτελεσματικός και αποδοτικός σχεδιασμός ενός καταναμημένου συστήματος. Συγκεκριμένα η ανάλυση των συγγραφέων επικεντρώνεται στη κατάλληλη τοποθέτηση και ανάθεση των απαιτούμενων λειτουργιών στα διάφορα τμήματα του συστήματος. Θεωρώντας την ύπαρξη μιας ιεραρχίας επιπέδων είναι ιδιαίτερα σημαντικό για το σχεδιαστή του συστήματος να επιλέξει έναν κατάλληλο διαμοιρασμό των επιθυμητών λειτουργιών έτσι ώστε να στη πράξη να επιτυγχάνεται το μέγιστο της απόδοσης του συστήματος.

##### Ιδέα

Για την αντιμετώπιση αυτού του προβλήματος οι συγγραφείς παραθέτουν και αναλύουν μια γενική σχεδιαστική αρχή με το όνομα “End – to – End”. Το επιχείρημα αυτό για ένα σύστημα όπου υπάρχει επικοινωνία δύο άκρων λέει τα εξής:

*Σε πολλές περιπτώσεις για να διεκπεραιωθεί η ζητούμενη για το σύστημα λειτουργία απαιτείται απαραίτητως η συνδρομή των δύο ακρων (επίπεδο εφαρμογής). Αυτό σημαίνει ότι δεν μπορεί η εκάστοτε λειτουργία να παρέχεται από πιο χαμηλό επίπεδο (επίπεδο μεταφοράς δεδομένων). Είναι όμως δυνατό σε κάποιες περιπτώσεις μια παραλλαγή της συγκεκριμένης λειτουργίας να έχει τοποθετηθεί από το σχεδιαστή στο χαμηλότερο επίπεδο για τη βελτίωση της αποτελεσματικότητας ή/και της απόδοσης.*

Προκειται στην ουσία για ένα γνωστό επιχείρημα (ήταν γνωστό την εποχή που γράφτηκε το paper) το οποίο είχε ήδη χρησιμοποιηθεί χωρίς όμως να έχει γίνει γενικά αποδεκτή η ορθότητά του. Στο συγκεκριμένο paper οι συγγραφείς επιδιώκουν να αποδείξουν την ορθότητα και την αναγκαιότητα χρήσης αυτής της αρχής, αναλύοντας τις απαιτήσεις ενός συστήματος μεταφοράς δεδομένων και δίνοντας αξιόλογα επιχειρήματα για το πως αυτές μπορούν να αντιμετωπιστούν με βάση την End-to-End αρχή. Η επιχειρηματολογία που παραθέτουν οι συγγραφείς αποτελείται από μία σειρά παραδειγμάτων, τα οποία καλύπτουν σε μεγάλο βαθμό τις βασικές απαιτήσεις ενός συστήματος επικοινωνίας.

## Τεκμηρίωση

Η ανάλυση και τεκμηρίωση της ιδέας των συγγραφέων γίνεται ακολουθώντας τρία βήματα:

- για κάθε ζητούμενη εφαρμογή παράθεση των απαιτήσεων και των πιθανών προβλημάτων
- ο στόχος του σχεδιαστή είναι η μείωση στο ελάχιστο της πιθανότητας αποτυχίας της λειτουργίας
- χρήση κατάλληλων λειτουργιών και ανάλυση της αποτελεσματικότητάς τους ανάλογα με το ποιο επίπεδο θα τοποθετηθούν

Τα βασικά σημεία στα οποία στηρίζουν την επιχειρηματολογία τους οι συγγραφείς είναι τα εξής:

### **1. Μεταφορά αρχείων από ένα κόμβο A σε έναν B**

Περιγράφεται το πρόβλημα της *προσεκτικής* μεταφοράς αρχείων από το δίσκο του κόμβου A στο δίσκο του B, θεωρώντας ότι υπάρχει ένα πρόγραμμα μεταφοράς αρχείων που ανήκει στο επίπεδο εφαρμογής και ένα δίκτυο επικοινωνίας και μεταφοράς δεδομένων που αποτελεί το πιο κάτω επίπεδο του συστήματος.

Μία προσέγγιση για την υλοποίηση ενός τέτοιου συστήματος είναι προσθήκη αθροίσματος ελέγχου (checksum) και επαναμετάδοσης (retry) στα πακέτα δεδομένων που ανταλλάσσονται μεταξύ των δύο κόμβων (end-to-end check and retry). Η προσέγγιση αυτή χρησιμοποιεί ένα άθροισμα ελέγχου, για κάθε αρχείο, το οποίο αποθηκεύεται μαζί με το αρχείο. Κάθε φορά που ο κόμβος B λαμβάνει ένα αρχείο από τον A η εφαρμογή που βρίσκεται στο B υπολογίζει (με βάση ένα προκαθορισμένο αλγόριθμο) το checksum και στέλνει την τιμή πίσω στον A, όπου η εφαρμογή τη συγκρίνει με την αρχική τιμή που είχε υπολογιστεί. Εάν η σύγκριση αποτύχει σημαίνει ότι κάπου έγινε λάθος και η διαδικασία μεταφοράς επαναλαμβάνεται.

Για το συγκεκριμένο σύστημα η εφαρμογή του End-to-End argument υποδεικνύει ότι όλες οι απαραίτητες λειτουργίες για την ασφαλή και αποτελεσματική μεταφορά δεδομένων μεταξύ των δύο ακρών μπορούν να επιτευχθούν μόνο από το επίπεδο εφαρμογής. Ακόμα και αν γίνει προσπάθεια για ενσωμάτωση κάποιων λειτουργιών στο communication network layer είναι άσκοπη όσον αφορά την αξιοπιστία στη μεταφορά αφού αυτή παράχεται εξ ολοκλήρου στο επίπεδο εφαρμογής. Το μόνο που θα οφελούσε μια τέτοια απόπειρα, θα ήταν η μείωση της συχνότητας επαναμεταδόσεων, δηλαδή στην απόδοση του συστήματος.

### **2. Εγγύηση παράδοσης (delivery guarantees)**

Εξετάζεται το πρόβλημα της ύπαρξης σαφών εγγυήσεων για τη παραλαβή δεδομένων μεταξύ δύο κόμβων. Η πιο προφανής και ευρέως χρησιμοποιούμενη προσέγγιση για αυτό το πρόβλημα είναι, για κάθε πακέτο που παραλαμβάνεται, η αποστολή από το κόμβο που παρέλαβε το πακέτο, ενός πακέτου επιβεβαίωσης (acknowledge) στον κόμβο που το

έστειλε. Η λειτουργία αυτή μπορεί πολύ εύκολα να υλοποιηθεί στο επίπεδο μεταφοράς, ειδοποιώντας τον αποστολέα ότι τα δεδομένα που έστειλε έφτασαν στο προορισμό.

Η εφαρμογή του End-to-End argument σε αυτή τη περίπτωση έρχεται να επισημάνει ότι για την αποτελεσματικότητα της λειτουργίας σημασία έχει η εφαρμογή να γνωρίζει όχι μόνο ότι το πακέτο έφτασε, αλλά και αν ο κόμβος που το παρέλαβε επέδρασε σε αυτό ή όχι. Δηλαδή η αναγκαία επιβεβαίωση θα πρέπει να προέρχεται από το κομμάτι εφαρμογής του κόμβου προορισμού και θα δηλώνει ότι έκανε τις απαραίτητες ενέργειες πάνω στο πακέτο δεδομένων που παρέλαβε. Ένα σημείο που κάνει ακόμα πιο ισχυρή την ανάγκη για εφαρμογή του End-to-End argument είναι ότι η εφαρμογή του κόμβου προορισμού μπορεί για κάποιο λόγο να παραλάβει το πακέτο και στη συνέχεια να το απορρίψει. Τότε, αντί για επιβεβαίωση, στον αποστολέα θα πρέπει να σταλεί ένα πακέτο που δηλώνει αποτυχία, κάτι που μπορεί να πραγματοποιηθεί μόνο σε επίπεδο εφαρμογής.

### **3. Ασφάλεια στη μεταφορά δεδομένων**

Η ασφάλεια στη μεταφορά δεδομένων μεταξύ δύο κόμβων εκφράζεται σχεδόν εξολοκλήρου με τη χρήση κρυπτογράφησης (ή πιο σωστά αυτό ίσχυε όταν γράφτηκε το paper, καθώς στις μέρες μας υπάρχουν και άλλοι τρόποι για τη παροχή ασφάλειας στα δίκτυα π.χ. Network Intrusion Detection Systems). Η επιχειρηματολογία για εφαρμογή του End-to-End argument για την ικανοποίηση αυτής της απαίτησης περιλαμβάνει τρία βασικά σημεία:

- Έαν η κρυπτογράφηση και αποκρυπτογράφηση πραγματοποιείται στο επίπεδο μεταφοράς θα πρέπει εκεί να γίνεται και η ασφαλής διαχείριση των κλειδιών public ή private.
- Τα δεδομένα θα είναι σε clear μορφή και συνεπώς ευάλωτα καθώς μεταφέρονται από το επίπεδο μεταφοράς στο επίπεδο εφαρμογής του κόμβου προορισμού.
- Η αυθεντικότητα του πακέτου μπορεί να ελεγχθεί μόνο από την εφαρμογή.

Με τη προσθήκη της κρυπτογράφησης στο επίπεδο εφαρμογής αυτά τα τρία σημεία ικανοποιούνται απόλυτα.

### **4. Αποφυγή διπλής αποστολής του ίδιου μηνύματος (duplicate message suppression)**

Στην επικοινωνία μεταξύ δύο κόμβων είναι πολύ πιθανό κάποιο πακέτο μπορεί να σταλεί δύο φορές σαν αποτέλεσμα της λήξης κάποιου χρονικού περιορισμού ή αποτυχίας που έχουν σαν αποτέλεσμα την επαναμετάδοση του πακέτου. Για αυτή τη περίπτωση οι σχεδιαστές κατανεμημένων συστημάτων χρησιμοποιούν μηχανισμούς εντοπισμού και καταστολής ή απλά προώθησης αυτών των πακέτων. Οι λόγοι για τους οποίους οι μηχανισμοί αυτοί πρέπει να ανατεθούν στην εφαρμογή μεταφοράς δεδομένων και όχι στο επίπεδο δικτύου είναι οι εξής:

Εάν οι λειτουργίες αυτές ανατεθούν στο επίπεδο δικτύου, η εφαρμογή ανταλλαγής δεδομένων μεταξύ των δύο κόμβων θα έχει πρόβλημα να λειτουργήσει αποτελεσματικά. Αυτό γιατί, πρώτον εάν το δίκτυο αγνοεί τα διπλά πακέτα και τα προωθεί κανονικά στο προορισμό τους η εφαρμογή θα δέχεται τα διπλά πακέτα χωρίς να μπορεί να τα απορρίψει. Δεύτερον, ακόμα και αν το δίκτυο καταστέλλει τα διπλά πακέτα, υπάρχει μεγάλη πιθανότητα η ίδια η εφαρμογή να δημιουργήσει (για κάποιο λόγο) δύο ίδια

πακέτα. Σε αυτή τη περίπτωση το επίπεδο μεταφοράς δεν μπορεί να το καταλάβει και τα θεωρεί σαν δύο ξεχωριστά πακέτα.

Επιπλέον στα καταναμημένα συστήματα υπάρχει ακόμα και σήμερα ένα πολύ μεγάλο και δυσεπίλυτο πρόβλημα. Το πρόβλημα είναι ότι είναι δύσκολο το ένα άκρο μιας σύνδεσης να καταλάβει αν το άλλο άκρο έχει τεθεί εκτός λειτουργίας ή απλά αργεί πολύ να απαντήσει, δεδομένου ότι για κάποιο χρονικό διάστημα δεν υπάρχει απόκριση.

Αυτό μπορεί να έχει σαν αποτέλεσμα τη δημιουργία πολλών διπλών πακέτων από επαναποστολές που μόνο το επίπεδο εφαρμογής μπορεί να εντοπίσει.

### **5. FIFO παραλαβή πακέτων**

Πρόκειται για την απαίτηση σε ορισμένες εφαρμογές να λαμβάνεται από το παραλήπτη πρώτα το πακέτο που στάλθηκε πρώτο (π.χ. multimedia εφαρμογές). Και σε αυτή τη περίπτωση το End-to-End argument επιβεβαιώνεται διότι ένας FIFO μηχανισμός δεν μπορεί να λειτουργήσει αποτελεσματικά σε επίπεδο δικτύου. Αυτό συμβαίνει γιατί ένα πακέτο είναι δυνατό να μην πάει απευθείας από τον ένα κόμβο στον άλλο, αλλά να πάει μέσω άλλων ανεξάρτητων υποδικτύων και ενδιάμεσων διεργασιών, φτάνοντας έτσι αργότερα στο προορισμό του από ότι κάποιο που έφυγε μετά από αυτό. Τη λύση μπορεί να τη δώσει μόνο η ενσωμάτωση της FIFO λειτουργίας στη καταναμημένη εφαρμογή.

### **6. Διαχείριση ανταλλαγής δεδομένων (transaction management)**

Την ορθότητα του End-to-End argument σε αυτή τη περίπτωση οι συγγραφείς την αποδεικνύουν με τη περιγραφή του SWALLOW, ενός καταναμημένου συστήματος αποθήκευσης δεδομένων. Από την ανάλυση των βασικών λειτουργιών διαχείρισης μεταφοράς δεδομένων του συστήματος προκύπτει ότι δεν υπάρχει σε καμία περίπτωση η ανάγκη για προσάρτηση των λειτουργιών αυτών στο επίπεδο δικτύου αφού είτε αυτό μπορεί να γίνει στο επίπεδο εφαρμογής, είτε είναι αναγκαίο να γίνει εκεί.

## **Επιλογή των σημείων εφαρμογής του End-to-End argument**

Ένα βασικό σημείο που τονίζεται από τους συγγραφείς είναι η προσεκτική και σωστή εφαρμογή της End-to-End αρχής για το σχεδιασμό ενός καταναμημένου συστήματος.

Αυτό προϋποθέτει σαφή κατανόηση της End-to-End φιλοσοφίας, με την έννοια ότι δεν είναι απαραίτητο για κάθε ζητούμενη απαίτηση η αυστηρή εφαρμογή της. Το ζητούμενο είναι η εφαρμογή να γίνει υπό τις κατάλληλες (ελαστικές ίσως) προϋποθέσεις για την επίτευξη της μέγιστης αποτελεσματικότητας.

Ενά χαρακτηριστικό παράδειγμα, στο οποίο End-to-End argument δεν εφαρμόζεται σαν αυστηρός κανόνας αλλά σαν γενικός οδηγός σχεδίασης, είναι οι real-time εφαρμογές στις οποίες μια αυστηρή εκδοχή της γενικής αρχής θα ήταν και δύσκολη αλλά ίσως και μη αναγκαία.

## **Θέματα απόδοσης**

Ένα βασικό στοιχείο του End-to-End argument όπως παρουσιάζεται από τους συγγραφείς είναι η παραδοχή ότι ένα κομμάτι ή μια απλή εκδοχή της απαιτούμενης

λειτουργίας θα μπορούσε να ενσωματωθεί και στο επίπεδο δικτύου με σκοπό την αύξηση της απόδοσης του συστήματος. Βέβαια τα ποσοστά κατά τα οποία μια λειτουργία εκκρεμεί μεταξύ των δύο επιπέδων εξαρτώνται από την εφαρμογή και το γεγονός αυτό δίνει τη δυνατότητα ευελιξίας στο σχεδιασμό του συστήματος.

Αυτομάτως γεννιέται το ερώτημα πόσο προγραμματιστικός κόπος αξίζει να καταναλωθεί για την υλοποίηση στο επίπεδο μεταφοράς δεδομένων και πόση αύξηση της απόδοσης αυτό θα επιφέρει. Το End-to-End argument δηλαδή περιλαμβάνει το trade-off απόδοσης και υλοποίησης στο επίπεδο μεταφοράς.

Είναι επίσης σημαντικό να αναφερθεί ότι σε ορισμένες περιπτώσεις η τοποθέτηση κάποιας λειτουργίας στο επίπεδο μεταφοράς δεν αυξάνει αναγκαία την απόδοση. Πρώτον διότι το επίπεδο μεταφοράς είναι κοινό σε πολλές εφαρμογές και κάποιες από αυτές αναγκάζονται να εκτελούν λειτουργίες που δεν χρειάζονται. Δεύτερον διότι το επίπεδο μεταφοράς ίσως να μην έχει όλη την απαραίτητη πληροφορία που χρειάζεται για να γίνει αποτελεσματικά η λειτουργία.

## β) Επιχειρήματα κατά του End-to-End argument

Οι συγγραφείς δίνουν αξιολογικά επιχειρήματα, με βάση παραδείγματα, για να αποδείξουν την ορθότητα του End-to-End argument στο σχεδιασμό οποιουδήποτε κατανεμημένου συστήματος. Αυτό που στην ουσία βγαίνει σαν συμπέρασμα είναι ότι οποιαδήποτε λειτουργία ενός συστήματος μπορεί να υλοποιηθεί θεωρητικά στο επίπεδο εφαρμογής αφήνοντας στο χαμηλότερο επίπεδο μεταφοράς μόνο τις βασικές λειτουργίες. Το θέμα είναι όμως πότε αυτό μπορεί να υποστηριχθεί και στη πράξη.

Το μεγαλύτερο πρόβλημα σε αυτή τη περίπτωση είναι ο υπερβολικός φόρτος που μπορεί να βαρύνει το δίκτυο από τις πολλαπλές λειτουργίες του επιπέδου εφαρμογής. Για παράδειγμα λειτουργίες όπως συμπίεση δεδομένων, τεμαχισμός (fragmentation) και επανασύνθεση πακέτων, μηνύματα λάθους, επαναμεταδόσεις κ.α μπορεί να είναι δύσκολο (ή και αδύνατο) να υλοποιηθούν μόνο από το επίπεδο μεταφοράς αλλά είναι επίσης δύσκολη και μη-πρακτική η υλοποίηση τους αποκλειστικά σε επίπεδο εφαρμογής. Επιπλέον η προσθήκη για κάθε επιθυμητή κατανεμημένη εφαρμογή λειτουργιών στο επίπεδο εφαρμογής συνεπάγεται, συνεχώς, τη συγγραφή μεγάλης ποσότητας νέου λογισμικού. Αυτό είναι σχετικά ανούσιο από την άποψη ότι θα ήταν καλύτερο να υπήρχαν υλοποιημένες κάποιες βασικές λειτουργίες για κατανεμημένες εφαρμογές και ενσωματωμένες στο επίπεδο μεταφοράς ώστε να μη χρειάζεται κάθε φορά να υλοποιούνται ξεχωριστά για κάθε εφαρμογή.

Από την άλλη η συγγραφή συνεχώς νέου κώδικα για κατανεμημένες εφαρμογές εγκυμονεί κινδύνους για την ασφάλεια του συστήματος, καθώς όσο μεγαλύτερη είναι η ποσότητα νέου λογισμικού που αναπτύσσεται, τόσο πιο πιθανό είναι αυτό να περιέχει προγραμματιστικές ατέλειες (vulnerabilities) τις οποίες μπορούν να εκμεταλλευτούν οι επίδοξοι hackers για να επιτεθούν στο σύστημα, πριν ακόμα τις αντιληφθούν οι χρήστες του συστήματος.

Επιπρόσθετα η συμβατότητα των κατανεμημένων εφαρμογών που τρέχουν σε μηχανές με διαφορετικά λειτουργικά συστήματα οφείλεται σε μεγάλο ποσοστό στο κοινό επίπεδο μεταφοράς (πρωτόκολλα επικοινωνίας). Η ανεξέλεγκτη μεταφορά λειτουργιών από το

επίπεδο μεταφοράς στο επίπεδο εφαρμογής μπορεί να έχει σαν αποτέλεσμα τη μείωση τη συμβατότητας μεταξύ των κόμβων του συστήματος.

Όσον αφορά την ασφάλεια της μεταφοράς δεδομένων του συστήματος, οι συγγραφείς παραθέτουν κάποιους λόγους για την ανάγκη κρυπτογράφησης στο επίπεδο εφαρμογής.

Παρόλο που τα επιχειρήματα αυτά είναι σωστά υπάρχει και αντίλογος που επιβάλλει να υπάρχει κρυπτογράφηση στο επίπεδο μεταφοράς. Ο αντίλογος είναι ότι αν η κρυπτογράφηση γίνει στο επίπεδο εφαρμογής δεν κρυπτογραφεί τα δεδομένα (προθήματα ή επιθήματα) που προστίθενται στο πακέτο στο επίπεδο μεταφοράς. Όπως για παράδειγμα τον TCP ή τον UDP header. Επιπλέον θα υπήρχε και σε αυτή τη περίπτωση πρόβλημα από την υπερφόρτωση του δικτύου, λόγω της χρονοβόρας διαδικασίας κρυπτογράφησης στο επίπεδο εφαρμογής. Ένα άλλο σημαντικό θέμα για την ασφάλεια του End-to-End argument είναι ότι τα δεδομένα στην ασφάλεια των δικτύων έχουν αλλάξει τραγικά από το 1984 που δημοσιεύτηκε το paper. Οι συγγραφείς σε πολλές περιπτώσεις θεωρούν δεδομένη την συμβολή και των δύο άκρων στην επίτευξη αρκετών λειτουργιών στο επίπεδο εφαρμογής. Θεωρούν δηλαδή και τα δύο άκρα έμπιστα πέρα πάσης αμφιβολίας. Σήμερα αυτή η θεώρηση αποτελεί μεγάλο λάθος καθώς είναι πολύ εύκολο και συχνό φαινόμενο κάποιος κακόβουλος χρήστης του Internet να προσποιείται ότι είναι το ένα ή/και το άλλο άκρο της σύνδεσης.

Στις μέρες μας αρχίζει να γίνεται ιδιαίτερα δημοφιλής η ύπαρξη του third-party host για την επικοινωνία δύο κόμβων. Ο τρίτος αυτός έμπιστος (κατά κανόνα) host είναι αναγκαίος για την εγκατάσταση της σύνδεσης μεταξύ των δύο άλλων και χρησιμοποιείται μόνο για αυτό το σκοπό. Η τεχνική αυτή είναι πολύ νέα και χρησιμοποιείται για λόγους ασφάλειας στην επικοινωνία. Το End-to-End argument δεν παρέχει πλαίσιο λειτουργίας για αυτή την απαίτηση.

Κάτι ακόμα πολύ σημαντικό που αντιτίθεται στην ιδέα του End-to-End argument είναι τάσεις που κυριαρχούν και οι απαιτήσεις που υπάρχουν σήμερα στη παγκόσμια αγορά και στη βιομηχανία τις πληροφορικής. Αυτές επιτάσσουν όσο το δυνατόν μεγαλύτερη συμπύκνωση λειτουργιών στα χαμηλότερα στρώματα των επικοινωνιακών δικτύων, ώστε να είναι στη συνέχεια ευκολότερη και οικονομικότερη η ανάπτυξη νέου εμπορεύσιμου λογισμικού, που θα τρέχει πάνω από το επίπεδο μεταφοράς.

Συμπερασματικά γίνεται κατανοητό ότι για ένα καταναμημένο σύστημα υπάρχει η ανάγκη ύπαρξης ενός κοινού πρωτοκόλλου δικτυακής επικοινωνίας όπως το TCP/IP, το οποίο θα περιλαμβάνει ένα πλήθος βασικών λειτουργιών στη μεταφορά δεδομένων.

Τα παραδείγματα και τα σενάρια που επικαλέστηκαν οι συγγραφείς για να υποστηρίξουν τα επιχειρήματά τους ταίριαζαν όλα άψογα στην End-to-End προσέγγιση. Δεν έγινε όμως καθόλου αναφορά σε περιπτώσεις όπου ένα σύστημα θα πρέπει να έχει πολύ συγκεκριμένες απαιτήσεις. Όπως ένα υψηλής ταχύτητας και αξιοπιστίας καταναμημένο σύστημα για στρατιωτική χρήση. Το σύστημα αυτό περιλαμβάνει αυξημένων απαιτήσεων λειτουργίες (π.χ. ασφάλειας και απόδοσης) οι οποίες μπορούν να υλοποιηθούν αποτελεσματικά μόνο σε χαμηλό επίπεδο.

Πιο γενικά θα μπορούσαμε να πούμε ότι το End-to-End argument είναι κατάλληλο για γενικής χρήσεως καταναμημένα συστήματα, όπου οι σχεδιαστές αυτών δεν ενδιαφέρονται να προσδώσουν στο σύστημά τους συγκεκριμένα υψηλών απαιτήσεων χαρακτηριστικά. Στην υλοποίηση όμως συγκεκριμένων συστημάτων, όπου υπάρχουν εξειδικευμένες απαιτήσεις η εφαρμογή αυτής της αρχής δεν αποδίδει πάντα. Δηλαδή το

δίκτυο θα πρέπει να έχει γνώση για το ποιοι είναι στα άκρα, ώστε να τους προσφέρει τις κατάλληλες υπηρεσίες.

Ο τρόπος ανάλυσης του προβλήματος και η επιχειρηματολογία από τους συγγραφείς σε κάποια σημεία γεννάει ενστάσεις σχετικά με τη γενικότερη ορθότητα του End-to-End argument. Η αιτία είναι η πολύ αφαιρετική προσέγγιση του προβλήματος. Αν και πρόκειται για θετική έως και μηχανική επιχειρηματολογία, παραμένει στη γενικότητα χρησιμοποιώντας ως μέσω μεμονωμένα παραδείγματα. Συγκεκριμένα ορίζεται πολύ γενικά την ιεραρχία των επιπέδων του δικτύου και του ρόλου που αυτά κατέχουν στη μεταφορά των δεδομένων. Γίνεται λόγος για επίτευξη απόδοσης, χωρίς όμως να γίνεται κάποιος ποσοτικός προσδιορισμός της ικανοποιητικής απόδοσης ή να οριστεί κάποια συνάρτηση κόστους η οποία ελαχιστοποιείται. Η ιδέα που μένει στον αναγνώστη αυτού του επιχειρήματος είναι ότι πρόκειται για μια πολύ γενική αρχή, η αποτελεσματικότητα της οποίας τελικά εξαρτάται από την εφαρμογή στην οποία χρησιμοποιείται. Γενικότερα είναι δύσκολο να γίνει κατανοητή και αποδεκτή μια σχεδιαστική επιχειρηματολογία, εάν δεν συνοδεύεται από κάποια, αν όχι μαθηματική, τουλάχιστον ποσοτική μοντελοποίηση.

Ο David P. Reed (ένας από τους συγγραφείς του paper) σε ένα paper που δημοσίευσε τον Απρίλιο του 2000 με τίτλο “The end of the End-to-End argument” φιλοσοφεί γύρω από τη τάση που είχε αρχίσει να αναπτύσσεται τότε για την ανάπτυξη κατανεμημένων συστημάτων και η οποία ερχόταν σε αντίθεση με το End-to-End argument. Ο λόγος ήταν το γνωστό trade-off που έθετε το ερώτημα:

να βελτιωθούν οι εφαρμογές αναπτύσσοντας λογισμικό στο επίπεδο εφαρμογής ή σε πιο χαμηλό επίπεδο;

Η απάντηση σε αυτό το ερώτημα είναι αυτή που έχουν υιοθετήσει πολλές κατανεμημένες εφαρμογές σήμερα και θα που θα καθορίζει τα κατανεμημένα συστήματα του μέλλοντος. Και αυτή (όπως άλλωστε και τα περισσότερα θέματα στη ζωή) χαρακτηρίζεται από την αρχή της απροσδιοριστίας του Heisenberg και η χρυσή τομή για το σχεδιασμό βρίσκεται κάπου ενδιάμεσα. Δηλαδή το End-to-End argument θα πρέπει να αποτελεί έναν γενικό οδηγό σχεδιασμού, αλλά και με τη προσθήκη και άλλων πιο ευέλικτων και συγκεκριμένων αρχών ανάλογα με τη ζητούμενη εφαρμογή.

### γ) Παραδείγματα κατανεμημένων συστημάτων που αντιτίθενται στο End-to-End argument

Τα περισσότερα γνωστά συστήματα σήμερα δεν έχουν υιοθετήσει εξολοκλήρου το End-to-End argument. Το πιο ευρέως διαδεδομένο λογισμικό για δικτυακή επικοινωνία τα πρωτόκολλα TCP/IP παραβιάζουν το End-to-End argument καθώς λειτουργίες όπως ο υπολογισμός των checksums, η επαναμετάδοση πακέτων, ο τεμαχισμός των πακέτων, τα acknowledgment πακέτα, τα μηνύματα λάθους και η κρυπτογράφηση των δεδομένων πραγματοποιούνται κάτω από το επίπεδο εφαρμογής.

Όλες τα κατανεμημένα συστήματα που τρέχουν πάνω από TCP/IP, συμπεριλαμβανομένου και του Internet, παραβιάζουν το End-to-End argument αφού όλες οι παραπάνω λειτουργίες πραγματοποιούνται κάτω από το επίπεδο εφαρμογής (να σημειωθεί βέβαια ότι πολλές από τις λειτουργίες του Internet όπως το http, το e-mail κ.α., έχουν υλοποιηθεί σύμφωνα με το End-to-End argument).

Επίσης το πρότυπο του Ethernet για μεταφορά δεδομένων σε τοπικό δίκτυο περιλαμβάνει αρκετές λειτουργίες σε χαμηλό επίπεδο.