ΣΧΕΔΙΑΣΗ ΠΡΩΤΟΚΟΛΛΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧ/ΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

Δίκτυα Δημόσιας Χρήσης και Διασύνδεση Δικτύων 2008-2009

ΚΑΒΟΥΡΓΙΑΣ ΓΕΩΡΓΙΟΣ Α.Μ 3659
ΚΑΣΤΑΝΗΣ ΔΗΜΗΤΡΙΟΣ Α.Μ 3871

ΠΕΡΙΕΧΟΜΕΝΑ

Σελ.

1) Εισαγωγή

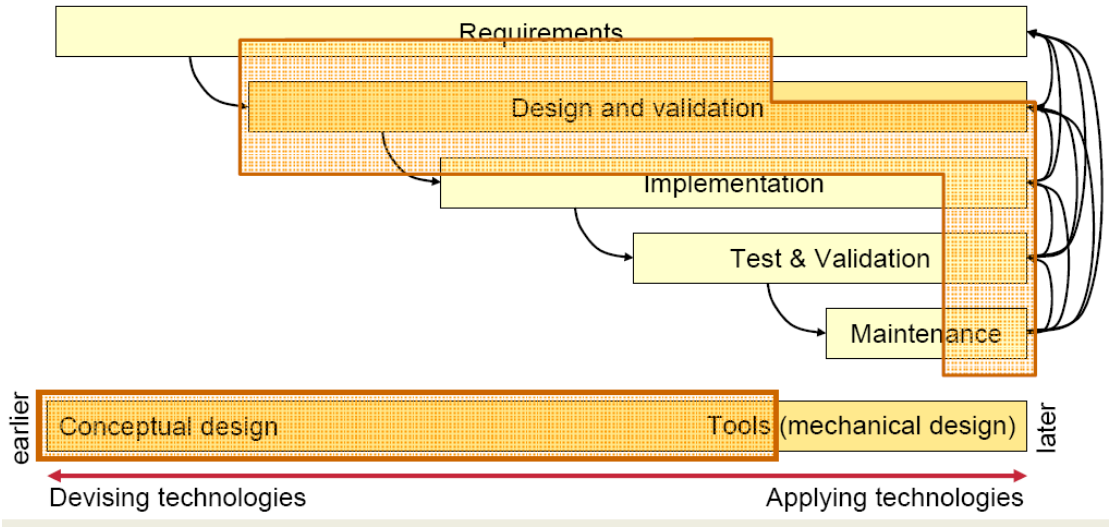
Τα πρώτα δίκτυα υπολογιστών προορίζονταν για να προσφέρουν περιορισμένες υπηρεσίες και να επιτυγχάνουν συνδέσεις σε μικρή κλίμακα. Επίσης σε αυτό συνέτεινε και το μεγάλο κόστος των υπολογιστών (δεν υπήρχαν ακόμη οι σημερινοί πανίσχυροι microprocessors, οι μνήμες ήσαν ακριβές, οι χωρητικότητές τους μικρές, κ.λ.π.), σε βαθμό που να είναι ακριβή η ανάπτυξη συστημάτων δικτύων τα οποία να κάνουν αποτελεσματικό routing ή switching και γενικότερη διαχείριση δικτύου. Η λύση ήταν επομένως η αγορά και εγκατάσταση αποκλειστικών κυκλωμάτων/ γραμμών σύνδεσης με συγκεκριμένα μεγάλα mainframe συστήματα. Η έλλειψη όμως κάποιων standards έκανε απαγορευτική την επέκταση τέτοιων συστημάτων προκειμένου να συνδεθούν με άλλα διαφορετικά συστήματα.

Η ανάγκη για διασύνδεση διαφορετικών συστημάτων, που εν γένει είναι πολύ απομακρυσμένα μεταξύ τους, έκανε επιτακτική την ανάγκη χρήσης των δημοσίων δικτύων επικοινωνιών (παλαιότερα χρησιμοποιούνταν ιδιωτικές μισθωμένες γραμμές επικοινωνίας). Έτσι, προκειμένου να διασυνδέονται πολλά ετερογενή περιβάλλοντα πάνω από ένα κοινό μέσο (δημόσια δίκτυα), έγινε απαραίτητη η σύσταση κάποιων standards που ορίζουν ακριβώς τον τρόπο σύνδεσης (interface) μεταξύ των συνδρομητών και του δικτύου.Δηλαδή ένα σύνολο κανόνων που είναι συμφωνημένοι και από τα δυο επικοινωνούντα μέρη και ελέγχει ή ενεργοποιεί τη σύνδεση,την επικοινωνία και τη μεταφορά δεδομένων μεταξύ τους. Αυτό είναι ένα πρωτόκολλο επικοινωνίας και στην πιο απλή του μορφή, μπορεί να οριστεί σαν οι κανόνες που διέπουν το συντακτικό, τη σημασιολογία και το συγχρονισμό της επικοινωνίας.Τα πρωτόκολλα μπορεί να υλοποιούνται από το υλικό, το λογισμικό ή και από ένα συνδυασμό των δύο.

Πέρα όμως από τα ήδη υπάρχοντα πρωτόκολλα, υπάρχουν στοιχεία στην καθημερινότητά μας που υποδεικνύουν την αναγκαίοτητα είτε να διασαφηνιστούν κάποια θέματα που πραγματεύονται τα ήδη υπάρχοντα πρώτοκολλα, είτε να δημιουργηθούν νέα πρωτόκολλα για να λύσουν διάφορα προβλήματα που υπάρχουν στις επικοινωνίες.Νέες εφαρμογές εμφανίζονται συνεχώς και όλο και περισσότερες σπό αυτές σχετίζονται με δίκτυα επομένως η αποσύνθεση της εφαρμογής σε λειτουργικά τμήματα και η διανομή αυτών των τμημάτων κάνει τη σχεδίαση πρωτοκόλλων ένα αναπόσπαστο κομμάτι της σχεδίασης του συστήματος.Επίσης υπάρχουν συνεχώς νέες απαιτήσεις εξαίτας της εξέλιξης της τεχνολογίας της επικοινωνίας.Τέλος ακόμα και ο τρόπος ζωής μας οδηγεί στην ανάγκη για νέα πρωτόκολλα με απτό παράδειγμα την ανάγκη για περισσότερες IP διευθύνσεις και τη δημιουργία του IPv6 πρωτοκόλλου.Σαφέστατα, τα προαναφερθέντα αποτελούν μία καλή βάση για να στηρίξουμε την άποψη περί ανάγκης εξέλιξης των πρωτοκόλλων είτε αυτή εμφανίζεται με τη μορφή της βελτιστοποίησης των ήδη υπάρχοντων όπως η διόρθωση των bugs, η διαγραφή μη αναγκαίων περιορισμών και η εξέλιξη(όσο αυτό είναι δυνατό)σύμφωνα με διαφοροποιημένες ή ακόμα και νέες απαιτήσεις είτε με τη δημιουργία νέων πρωτοκόλλων.

Ήρθε η ώρα λοιπόν να αναφερθούμε στον όρο ‘Σχεδίαση Πρωτοκόλλων’!Υπάρχουν διάφορες οπτικές γωνίες όπως η μαθηματική όπου ασχολείται με την σχεδίαση και την απόδειξη ορθότητας, από τη σκοπιά του μηχανικού που πραγματεύεται τη διαχείρηση της σχεδίασης πρωτοκόλλου και από πλευράς εργαλείων για ορισμό και επαλήθευση πρωτοκόλλων, ωστόσο εδώ θα ασχοληθούμε με άλλα θέματα όπως γιατί κάποιοι σχεδιασμοί λειτουργούν καλύτερα από κάποιους αλλους, την κατανόηση της σχέσης μεταξύ λειτουργικών και μη απόψεων στη σχεδίαση πρωτοκόλλων και βέβαια θα αναφερθούμε σε κάποια χαρακτηριστικά που πρέπει να διαθέτει ένα πρωτόκολλο(όπως για παράδειγμα η ανθεκτικότητα), αλλά και σε στοιχεία που λογικά θα χαρακτηρίζουν κάποια πρωτόκολλα στο μέλλον.

Το παρακάτω σχήμα δίνει μία καλή άποψη όσο αφορά στη διαδικασία σχεδίασης πρωτοκόλλου:



Σχήμα 1: διαδικασία σχεδίασης πρωτοκόλλου

Κατ’αρχάς υπάρχουν κάποιες προυποθέσεις πριν αρχίσει η σχεδίαση του πρωτοκόλλου.Θέματα όπως η κατανόηση του προβλήματος που προσπαθούμε να λύσουμε, η κατανόηση των απαιτήσεων(λειτουργικές και μη), η κατανόηση των περιορισμών, η λήψη αποφάσεων για μεγαλύτερη ακρίβεια σε θέματα που μας ενδιαφέρουν με κόστος όμως σε άλλους τομείς(τρανό παράδειγμα η ταχύτητα έναντι του κόστους)και η προσπάθεια για γενίκευση του σχεδιασμού μας, είναι νευραλγικής σημασίας για την επιτυχία του έργου μας.Επίσης υπάρχει μία σχεδιαστική απόφαση που χρήζει ειδικής μνείας, αφού αυτό θα καθορίσει σε μεγάλο βαθμό την πορεία μας κατά τη διαδικασία που θα ακολουθήσει.Αυτή δεν είναι άλλη από την ‘να φτιάξεις ή να πάρεις’ (ατυχώς παραφρασμένο από την σαφώς πιο εύηχη αγγλική εκδοχή:’make or take’!!!!) και αφορά την απόφαση της χρήσης της ήδη υπάρχουσας τεχνολογίας ή τη δημιουργία νέας από την αρχή.Στην πρώτη περίπτωση έχουμε οφέλη που σχετίζονται με την πείρα και τον υπάρχοντα κώδικα, όμως θέματα όπως το κατά πόσο ταιριάζει απόλυτα στις απαιτήσεις μας και το χρονικό διάστημα που θα

υποστηρίζεται ακόμα αυτή η τεχνολογία, ίσως μας δημιουργήσουν προβλήματα στο μέλλον.Στη δεύτερη θα έχουμε μεν μεγαλύτερο ρίσκο και δυσκολία, όμως θα είναι σαφώς περισσότερες και οι πιθανότητες να παραμείνει για μεγαλύτερο χρονικό διάστημα στην αγορά.

Όσο αφορά στη διάρκεια της διαδικασίας της σχεδίασης ίσως χρειαστεί να γίνουν κάποιες υποθέσεις σε σχέση με τις υπηρεσίες και τα χαρακτηριστικά του κατώτερου επιπέδου πάνω στο οποίο θα στηριχθεί το πρωτόκολλο αλλά και σε σχέση με τις εφαρμογές του ανώτερου επιπέδου οι οποίες θα το χρησιμοποιούν παρ'όλα αυτά υπάρχουν και εδώ στοιχεία που αν τύχουν ιδιαίτερης προσοχής είναι σίγουρο ότι θα κάνουν πιο εύκολο το έργο μας!Η επιπεδοποίηση θα ανεξαρτητοποιήσει τα κατώτερα επίπεδα από τα ανώτερα και παράλληλα θα μας βοηθήσει να κατασκεύασουμε ένα πρωτόκολλο που να μην επηρεάζεται ριζικά από μικρές αλλαγές και να είναι εύκολη η πρόσθεση νέων στοιχείων, ωστόσο η αυστηρή επιπεδοποίηση δεν είναι πάντα εφαρμόσιμη και θα υπάρξουν στιγμές που δε θα είναι δυνατό να 'αποκρύψουμε΄κάποια στοιχεία.Επίσης, όχι μόνο πριν, αλλά και κατά τη διάρκεια της σχεδίασης πρέπει να είμαστε έτοιμοι να θυσιάσουμε κάποια πράγματα για χάρη κάποιον χαρακτηριστικών που θέλουμε να πληρεί το πρωτόκολλο μας(λειτουργικότητα vs απλότητα, αξιοπιστία vs καθυστέρηση είναι μόνο μερικά παραδείγματα μίας ιδιαίτερα μακροσκελούς λίστας….).Μερικοί χρήσιμοι 'κανόνες' είναι να είναι ευέλικτο, να είναι επεκτάσιμο, να είναι αποτελεσματικό, να είναι επαρκές και να μην είναι πολύπλοκο.

Κλείνοντας αυτή την εισαγωγική ενότητα, μία παρουσίαση ατόμων που ασχολούνται με τη σχεδίαση πρωτοκόλλων ίσως να έκανε ακόμα πιο οικεία την ένοια στον αναγνώστη.Σε μία βιομηχανία με στόχο τη διεύρυνση της αγοράς, σε μία επιχείρηση είτε δουλεύοντας για ένα πελάτη είτε για τις ανάγκες τις ίδιας τις εταιρίας και βέβαια ερευνητές και επιστήμονες !

2) Αξιοπιστία (reliability)

2.1) Εισαγωγή

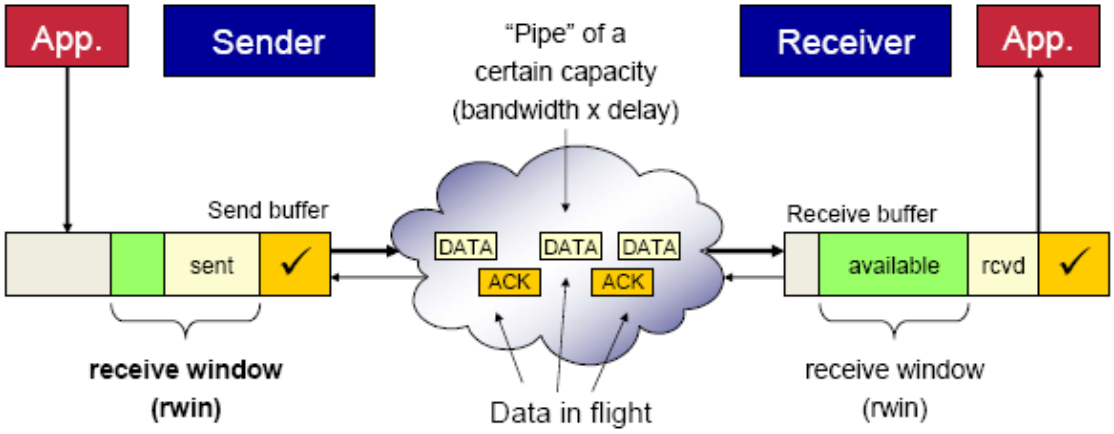
Σίγουρα θα περιμένετε να ακούσετε κάποιο ορισμό που να καθορίζει την ένοια της αξιοπιστίας!Αντιθέτως ας ξεκινήσουμε αναφέροντας το βασικό στόχο ενός πρωτοκόλλου και μέσου αυτού θα γίνει εμφανές και το περιεχόμενο της ένοιας ‘αξιοπιστία’ που είναι και το αντικείμενο με το οποίο θα ασχοληθούμε σε αυτή την ενότητα.Ο βασικός στόχος ενός πρωτοκόλλου είναι ο συγχρονισμός (synchronization) της κατάστασης μεταξύ δύο ή περισσοτέρων κόμβων.Αυτή η κατάσταση μπορεί να είναι οτιδήποτε, όπως κάποια δεδομένα, κάποια σχέση επικοινωνίας, το αποτέλεσμα κάποιας ενέργειας ή ακόμα τα περιεχόμενα μίας βάσης δεδομένων ή ενός φακέλου.Για να είναι αξιόπιστος ο συγχρονισμός, πρέπει να μπορέι να επιτευχθεί με τον ελάχιστο αριθμό ανταλλαγής μηνυμάτων.

Ας δούμε τώρα τι πρόβλημα μπορεί να δημιουργηθεί κατά την επικοινωνία δύο κόμβων, κάτι που μειώνει βέβαια την αξιοπιστία του συστήματός μας.Η πλειοψηφεία των προβλημάτων αυτών αποτελεί γνωστές έννοιες, οπότε θα αρκεστούμε σε απλή αναφορά κάποιων από αυτά.Υπάρχει πιθανότητα να δημιουργηθεί πρόβλημα στην επικοινωνία από την επιδραση ενός συνδέσμου (λάθη στα bits, απώλειες πλαισίων (frame), καθυστρήσεις κ.ά) ή σε ένα router (όπως απώλεια πακέτων, καταστροφή πακέτων (packet), καθυστέρηση ενός πακέτου κ.ά).Επίσης μπορεί να δημιουργηθούν σφάλματα στο δίκτυο (συγκρούσεις (congestion), routing loops, μη διαθεσιμότητα δικτύου κ.ά) αλλά και στον παραλήπτη (απώλεια πακέτων εξαιτίας buffer overflow, αποτυχία κάποιας εφαρμογής, δυσλειτουργίες γενικώς κ.ά).

Ενοείται ότι και σε αυτή την οπτική της σχεδιάσης πρωτοκόλλων (αξιοπιστία) πρέπει να λάβουμε κάποιες αποφάσεις που θα μειώσουν την απόδοση του πρωτοκόλλου μας σε άλλους τομείς, αλλά θα αυξήσουν την αξιοπιστία του. Η κυριότερη είναι ότι έχουμε επιβάρυνση στην καθυστέρηση προκειμένου να διευθετηθούν εργασιές που χρησιμοποιούν το πρωτόκολλο και αυτό οφείλεται σε δίαφορες διεργασίες.Οι πιο σημαντικές είναι η κωδικοποίηση και απόκωδικοποίηση της αρχικής πληροφορίας για τον έλεγχο των λαθών και παράλληλα την αύξηση της αξιοπιστίας του συστήματός μας και έλεγχοι που επιβάλλουν κάποιες τεχνικές ανίχνευσης ή διόρθωσης λαθών (για τεχνικές ανίχνευσης ή/και διόρθωσης λαθών θα μιλήσουμε στη συνέχεια), δηλαδή απαιτείται μεγαλύτερος φόρτος επεξεργασίας.Επίσης προκειμένου να γίνει έλεγχος λαθών μεταφέρεται μεγαλύτερο μέγεθος πληροφορίας, επομένως χρειάζεται μεγαλύτερος χρόνος για να γίνει η μετάδοση.Εξίσου σημαντική είναι και η απόφαση που αφορά στην επιλογή του μηχανισμού για τον έλεγχο της αξιοπιστίας, κάτι που επηρεάζει όχι μόνο την αποτελεσματικότητα του πρωτοκόλλου,αλλά και την πιθανότητά του να είναι αξιόπιστο.

2.2) Μηχανισμοί ελέγχου αξιοπιστίας

Ο μηχανισμός ελέγχου αξιοπιστίας επιλέγεται σε σχέση με την εφαρμογή και τη σημασία της, το περιβάλλον λειτουργίας (είδη λαθών, συχνότητα σφαλμάτων κ.ά) και την εγκατάσταση της επικοινωνίας (για παράδειγμα ο αριθμός των παραληπτών).Εδώ θα μας απασχολήσουν θέματα που αφορούν το είδος (για παράδειγμα έχουμε περισσότερη επιπρόσθετη πληροφορία ανά πακέτο ή περισσότερα πακέτα;), το ποσοστό και τις χρονικές στιγμές (συμβαίνουν συνεχώς ή μόνο σε περίπτωση σφάλματος;) της επιβάρυνσης που προκύπτει, τις πληροφορίες που πρέπει να ξέρει ο αποστολέας για τον παραλήπτη, το είδος και τον αριθμό των παραληπτών και τέλος τις παραμέτρους που επηρεάζουν την μέγιστη δυνατή απόδοση.Θα καταφύγουμε και πάλι σε μία εγνωσμένη μέθοδο για να γινουν πιο κατανοητά όλα αυτά και δεν είναι άλλη από αυτή της παράθεσης παραδείγματος! Παρακάτω παρατίθεται ένα σχήμα που απεικονίζει την TCP επικοινωνία και πανω σε αυτή θα μελετήσουμε συγκεκριμένους τρόπους με τους οποίους εμφανίζονται τα προβλήματα που προαναφέρθηκαν και βέβαια μεθόδους αντιμετώπισης.



Σχήμα 2: TCP επικοινωνία

Ας μελετήσουμε το θέμα της υπερφόρτωσης (overload) και της 'παραγκελίας' (ordering).Ο όρος ordering αναφέρεται στους αριθμούς ακολουθίας (sequence numbers SN) που μετράνε μηνύματα, πακέτα ή bytes και σαν στόχο έχουν την αποφυγή αναδιπλώσεων (wrap around) σε γρήγορα δίκτυα.Ο όρος overload έχει να κάνει τόσο με τον παραλήπτη όσο και με το δίκτυο.Στην περίπτωση του παραλήπτη έχουμε έλεγχο ροής (τυπικά μηνύματα παραθύρων με χρήση SN και με τον παραλήπτη να δηλώνει το διαθέσιμο μέγεθος buffer) με στόχο την ανανέωση της συχνότητας και της ικανότητας να είναι γεμάτος ο αγωγός και έλεγχο συχνότητας ο οποίος είναι προσυμφωνημένος μεταξύ του αποστολέα και του παραλήπτη και μπορεί να αλλάζει.

2.2.1) Μηχανισμοί εύρεσης λαθών (error detection)

Υπάρχουν διάφοροι τρόποι για να αντιμετωπίσουμε τα προβλήματα που μοπορεί να δημιουργηθούν στην προσπάθειά μας να επιτύχουμε όσο το δυνατό μεγαλύτερη αξιοπιστία τόσο σε επίπεδο bit όσο και σε επίπεδο πακέτων.Μπορούμε να

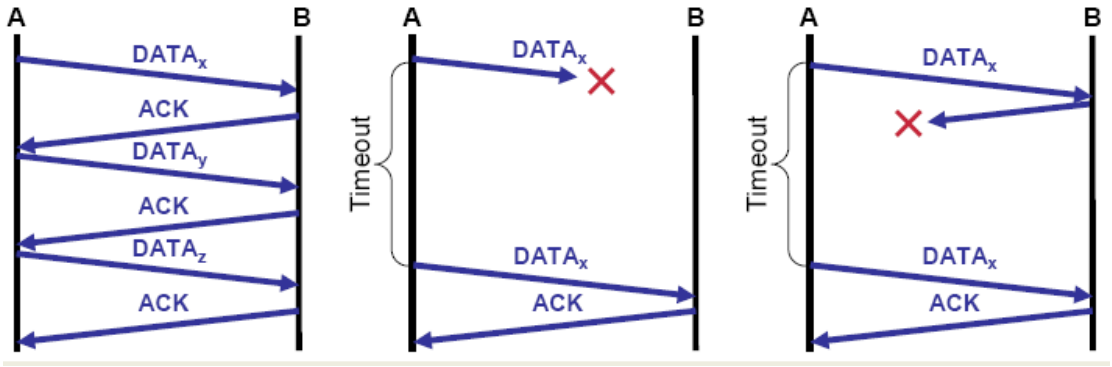
χρησιμοποιήσουμε Checksums, CRCs (Cyclic Redundancy Check Code) και MACs (Medium Access Control) για να βρούμε λάθη σε επίπεδο bit ή σε επίπεδο πλαισίου (frame) σε ένα πακέτο και Sequence numbers (SN) για να βρούμε πακέτα που έχουν χαθεί (με τον όρο χαθεί περιλαμβάνουμε και τις αποτυχημένες αποστολες).

2.2.2) Μηχανισμοί διόρθωσης λαθών (error correction)

Ακολουθούν μερικοι τρόποι διόρθωσης λαθών με γραφική αναπαράσταση και σύντομη περιγραφή του καθενός:

2.2.2.1) Απλό Lock-Step Πρωτόκολλο (simple Lock-Step protocol)

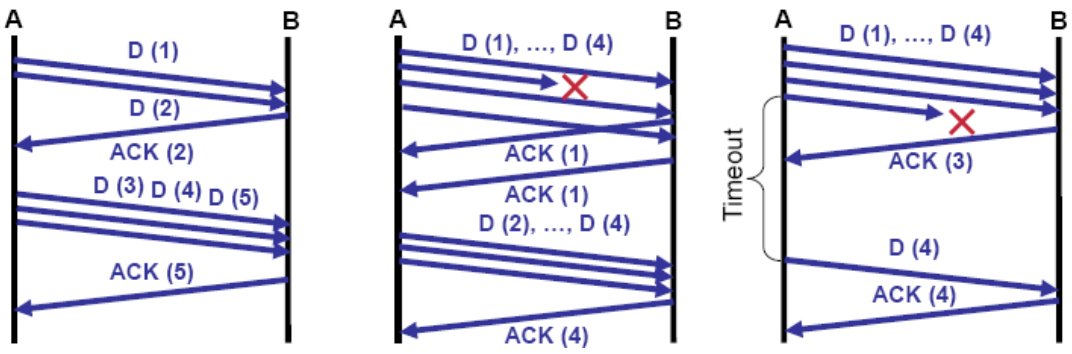
- Τα δεδομένα αποστέλονται και περιμένουμε για επιβεβαίωση (acknowledgement(ACK))
- Timeout στη μετάδωση trigger (trigger retransmission)
- Ασήμαντο αλλά και πολύ περιορισμένο
- Παράδειγμα: Trivial File Transfer Protocol (TFTP)



Σχήμα 3: simple Lock-Step protocol

2.2.2.2) Αθροιστικό ACK με Go-back-N (Cumulative ACK with Go-back-N)

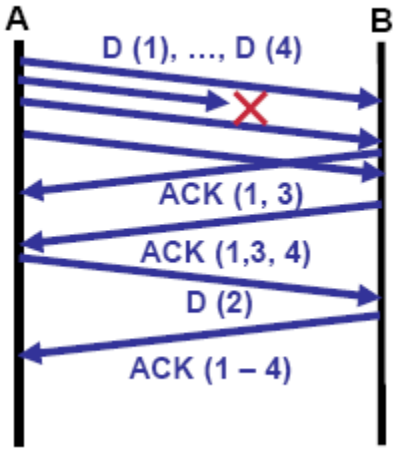
- Μηχανισός βασισμένος σε παράθυρα που επιτρέπει πολλαπλά σημαντικά πακέτα (με αυστηρό SN εύρος και μέγεθος buffer)
- Timeouts ή trigger αναμεταδώσεις (trigger retransmissions) των εκτός αίτησης ληφθέντων πακέτων
- Παραλλαγές: HDLC (LAPB/D/F), X.25 layer 3, plain old TCP



Σχήμα 4: Cumulative ACK with Go-back-N

2.2.2.3) Επιλεκτικά ACKs (Selective Acknowledgements)

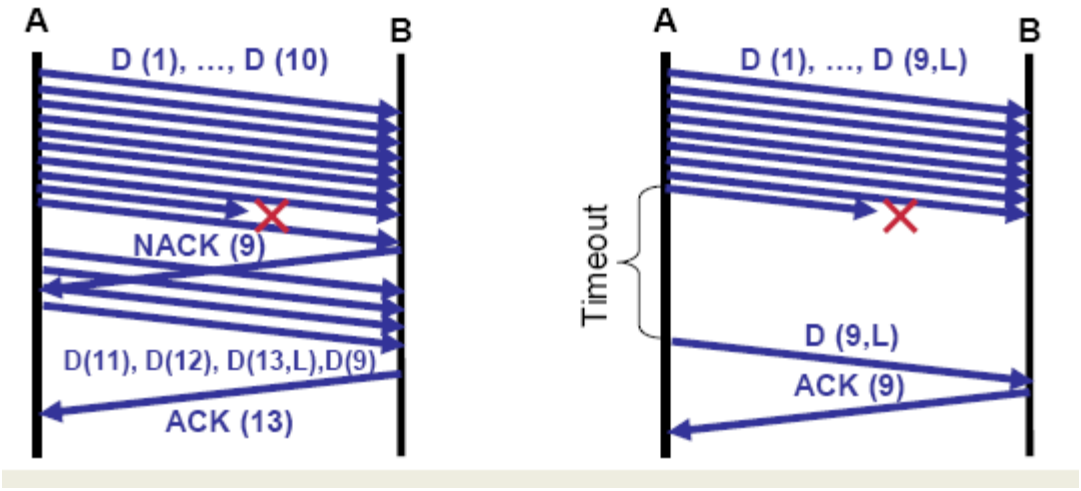
- Βασισμένη σε παράθυρο αλλά ρητή επιβεβαίωση των ληφθέντων πακέτων
- Ο παραλήπτης κρατάει τα εκτός αίτησης πακέτα (out-of-order packets)



Σχήμα 5: Selective Acknowledgements

2.2.2.4) Απλό NACK Πρωτόκολλο (Simple NACK (Negative ACK) Protocol)

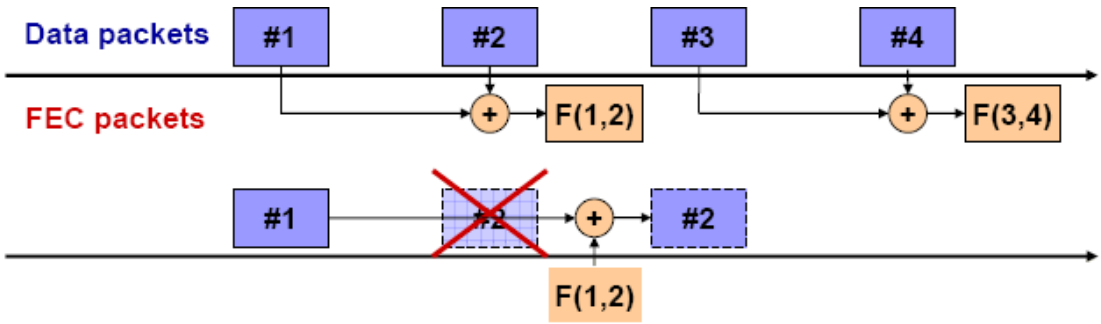
- Αισιόδοξη υπόθεση:τα πακέτα θα φθάσουν,άρα αναφέρουμε μόνο τις αποτυχίες (Negative ACK)
- Ειδικοί μηχανισμοί απαραίτητοι για το τελευταίο πακέτο
- Ειδικοί μηχανισμοί απαραίτητοι γιαέλεγχο ροής



Σχήμα 6: Simple NACK Protocol

2.2.2.5) Forward Error Correction (FEC)

- Βασική υπόθεση:θα συμβούν λάθη (αυξάνουμε την πιθανότητα σωστής λήψης στέλνοντας πακέτα μαζί με πακέτα ισοτιμίας (parity packets))
- Απλό XOR-based FEC (P_fec=P1 XOR P2 XOR P3… XOR Pn)
- Αυξάνονται οι απαιτήσεις σε bandwidth
- Αυξάνεται το overhead μειώνεται όμως η καθυστέρηση (δε χρείαζεται να περιμένουμε για NACK ή timeout)
- Υπάρχουν και πιο περίπλοκα FEC



Σχήμα 7: Forward Error Correction

Συνοψίζοντας υπάρχουν κάποια προβλήματα όσο αφορά στην αξιοπιστία. Κάποια στοιχεία πρέπει να είναι γνωστά μεταξύ αποστολέα και παραλήπτη όπως το μέγεθος του παραθύρου του παραλήπτη, τα SN και το τελευταίο ACK, ωστόσο η αρχικοποίηση είναι πιθανό να προκαλέσει Denial-of-Service (DoS) προβλήματα οπότε το ιδανικό είναι όλα αυτά να προσαρμόζονται δυναμικά στο περιβάλλον που πιθανότατα να είναι συχνά μεταβαλλόμενο.Αυτά όμως δεν ισχύουν στην περίπτωση που έχουμε

επικοινωνία μεταξύ ομάδων κόμβων.Ενδεικτικά ναφέρουμε ότι ακόμα κ ο ορισμός της σημασίας της σύνδεσης αλλάζει εδώ.

2.3) Χαλάρωση των απαιτήσεων για αξιοπιστία

Όπως είδαμε υπάρχουν πολλά ζητήματα που πρέπει να ικανοποιηθούν προκειμένου να έχουμε αξιοπιστία σε ικανοποιητικά επίπεδα.Το θέμα που θα σχολιάσουμε εδώ είναι τα περιθώρια που έχουμε ώστε να μειθούν αυτές οι απαιτήσεις και έτσι με το μικρότερο δυνατό κόστος σε απώλειες να έχουμε αύξηση του επιπέδου της αξιοπιστίας.Αρχικά ας σχολιάσουμε για τους κόμβους.Δεν είναι απαραίτητο για όλους τους κόμβους να παίρνουν όλες τις πληροφορίες, όμως υπάρχουν κόμβοι που οι λειτουργίες που εκτελούν απαιτούν όλες το μέγεθος της πληροφορίας και κάποιοι άλλοι που είναι πιθανό να καταρεύσουν αν δεν λάβουν όλες τις πληροφορίες.Συνεχίζοντας να αναφερθούμε στην πληροφορία και τη σημασία της.Πιθανότατα όλο το εύρος της πληροφορίας δεν είναι εξίσου σημαντικό, επομένως η ορθότητα και τα χρονικά περιθώρια για την αποστολή τους δεν είναι το ίδιο σημαντικά για όλα τα δεδομένα κάτι που ευνοεί την καλύτερη κάλυχη κάποιων τμημάτων της πληροφορίας έναντι άλλων.Για παράδειγμα μπορούμε να μειώσουμε τις απώλειες ανά πλαίσιο αν παρέχουμε CRC και/ή FEC μόνο σε τμήματα του πακέτου αφήνοντας τα λιγότερο σημαντικά τμήματα να περιέχουν σφάλματα, αλλά προστατεύοντας τα τμήματα που είναι απαραίτητα για την ανακατασκευή.Επίσης μπορούμε να έχουμε χαλάρωση σε θέματα που αφορούν την ακολουθιακή μετάδωση των δεδομένων (αξιόπιστη,αλλά όχι σειριακή μετάδωση όλων των δεδομένων διαχωρίζοντας πολλαπλά και ανεξάρτητα σειριακά ρεύματα μετάδοσης) και στις συγκρούσεις.

3) Ανθεκτικότητα (Robustness)

3.1) Εισαγωγή

Ενοείται ότι τα πρωτόκολλα πρέπει να είναι ανθεκτικά σε κάθε είδους δυσλειτουργία.Εδώ θα παρουσιάσουμε κάποιες γενικές κατηγορίες και θα σχολιάσουμε την ανθεκτικότητα των πρωτοκόλλων απέναντί τους.Θα μιλήσουμε για προβλήματα κατά τη διάρκεια της κανονικής λειτουργίας, απλές αποτυχίες (failure)(π.χ απώλειες πακέτων, κατάρευση συνδέσμων και κατάρευση κόμβων)και τις σαφώς μεγαλύτερου βαθμού δυσκολίας δυσλειτουργίες (π.χ προβλήματα στο υλικό, στις εφαρμογές και στη δομή) και προβλήματα από εξωτερικούς παράγοντες.

3.2) Ανθεκτικότητα κατά των απλών αποτυχιών (Simple Failures)

Πέρα από τα θέματα που αφορούν τις απώλειες πακέτων έχουμε προβλήματα που σχετίζονται με τα timeouts.Κατ'ρχάς τα timeouts των παραληπτών που εμπλέκονται πρέπει να ταιριάζουν, μία διαφορά που μπορεί να προκύψει και από κακό υπολογισμό των timeouts, αλλά και να προσαρμόζονται έτσι ώστε να ανταπεξέρχονται στις διαφορετικές μετρήσεις.Επίσης, τα timeouts πρέπει να υπολογίζουν και επαναλαμβνόμενες απώλειες πακέτων και να διαχειρίζονται μόνο από τη μία πλευρά, ειδάλλως σε περίπτωση που συμβεί ένα timeout, δεν έχει χρηστικά αξία!

3.3) Ανθεκτικότητα κατά των δυσλειτουργιών (Malfunctions)

Εδώ θα σχολιάσουμε προβλήματα στο υλικό που προκαλούν λανθασμένες λειτουργίες (για παράδειγμα έχουμε συχνή εμφάνιση προβλημάτων στους συνδέσμους και γιαυτό χρησιμοποιούμε checksums για να μειώσουμε αυτή την πιθανότητα), λάθη στις εφαρμογές όπως bugs και ακόμα λάθη στη δομή.Αρχικά ένα σύστημα πρέπει να 'αυτό-σταθεροποιείται' (self-stabilizing) που σημαίνει ότι όταν αφαιρείται κάποια δυσλειτουργία, το σύστημα πρέπει να επανέρχεται σε κανονική λειτουργία και σε περίπτωση που κάποια λανθασμένη πληροφορία μπαίνει στο σύστημα δεν πρέπει να παραμένει επ'αόριστο.Επίσης πρέπει να διασαφηνιστεί ότι κάποια πράγματα που θεωρούνται δεδομένα στη μία πλευρά ενός συστήματος δεν είναι σίγουρο ότι θα μεταφερφθούν στα άλλα άκρα.Όπως προαναφέρθηκε,ένα ακόμα στοιχείο που είναι πολύ σημαντικό είναι η λανθασμένη δομή.Αν δύο συστήματα πρέπει να έχουν μία συγκελριμένη δομή για να επικοινωνούν σωστά, τότε βρισκόμαστε σε μία πολύ εύθραυστη κατάσταση και η πιθανότητα λάθους είναι εμφανής.Ένας τρόπος για να το διαπιστώσουμε αυτό, είναι ότι τα συστήματα που δεν είναι συμβατά δεν παρουσιάζουν κάποια επικοινωνία.Κλείνοντας αυτή την υποενότητα αξίζει να ανφερθούμε στο γεγονός ότι κάποιες οφρές ένας σύνδεσμος που έχει καταρεύσει είναι προς όφελός μας!Ένας 'κακός' (bad) συνδεσμος μπορεί να είναι χειρότερος από ένα κατεστραμένο σύνδεσμο, αφού υπάρχει πιθανότητα το routing πρωτόκολλο να έχει κάποια καλύτερη εναλλακτική λύση.

3.4) Ασφάλεια (Robustness against Malice)

Μία επίθεση χωρίζεται σε τέσσερα στάδια:

1. αναγνώριση (Reconnaissance)
2. εισβολή(Intrusion)
3. εδραίωση και κάλυψη(Consolidation)
4. εύρεση αντικειμενικού στόχου

Η καλή ασφάλεια αντιμετωπίζει όλες αυτές τις φάσεις με πρόληψη, εντοπισμό και απομόνωση.
Στην περίπτωση του spoofing, δεν προσβλέπει στο να αποκτήσει άδεια εισόδου, παραμόνο να προσποιηθεί ότι έχει.Έτσι στην περίπτωση του UDP πλύ εύκολα μπορεί να ξεγελαστεί, ενώ σε αυτή του TCP είναι μεν πιο δύσκολο, αλλά και πάλι είανι εφικτό.Για πραγματική προστασία απιτείται κρυπτογραφία.Επίσης, σε γενικές γραμμές αυτοί που προσπαθούν να δημιουργήσουν πρόβλημα, πλήττουν περισσότερους πόρους από αυτούς που απαιτούνται για να επιτύχουν το στόχο τους.Εδώ η λύση είναι το σύστημά μας να δουλεύει σε πολλές διαφορετικές λειτουργίες που να σχετίζονται με ελέγχους ασφάλειας.

3.5) Θέματα ασφάλειας στην εφαρμογή πρωτοκόλλων

Σε αυτό το σημείο θα σχολιάσουμε για τα θέματα ασφάλειας που σχετίζονται με την υλοποίησή μας.Ο πιο δημοφιλής τρόπος για να πλήξουν το σύστημά μας χρησιμοποιώντας τα λάθη στον κώδικά μας είναι μέσω της υπερχείλησης του buffer (buffer overflow).Τα περισσότερα λάθη οφείλονται στη χρήση συναρτήσεων που παρέχονται από τις βιβλιοθήκες της γλώσσας που χρησιμοποιούμε και δεν έχουν έλεγχο ορίων (π.χ strcpy()) και η λύση είναι η χρήση παρόμοιων συναρτήσεων που να παρέχουν όμως τέτοιου είδους ελέγχους (π.χ strncpy()).Άλλος ένας τρόπος είναι μέσω των βάσεων δεδομένων οι οποίες διατηρούν στοιχεία για τους χρήστεςκαι η ενημέρωση τους από την είσοδο που δίνει ο χρήστης είναι ιδιαίτερα ευάλωτη.Συνοψίζοντας, μπορούμε να συμπεράνουμε το πόσο ευάλωτοι είμαστε στις επιθέσεις αν σκεφτούμε το εξής, τα πραγματικα συστήματα είναι πολύ περίπλοκα για να μην έχουν λάθη και επίσης με το πέρασμα των χρόνων γίνονται και πιο περίπλοκα προκειμένου να καλύψουν τις ανάγκες μας.Αυτό σημαίνει ότι υπάρχουν όλο και περισσότερα κενά στην άμυνά μας (security holes).Ενώ όμως σαν αμυνώμενοι πρέπει να βρούμε και να καλύψουμε όλα τα κενά, για τον επιτιθέμενο ένα και μόνο ένα κενό είναι αρκετό…

4) Ασφλάλεια (Security)

4.1) Εισαγωγή

Όπως αναφέρθηκε ήδη το σύστημά μας διατρέχει πολλούς κινδύνους είτε από εσωτερικούς παράγοντες (malfunctions και failures) είτε από εξωτερικούς (malice).Προκειμένου να το προφυλάξουμε πρέπει να πργανώσουμε την άμυνά μας και αυτό θα γίνει με την ασφάλεια πρωτοκόλλου.Ωστόσο κάθε σύστημα έχει κάποιες αδυναμίες και όντας τρωτό δίνεται η δυνατότητα να γίνει παράκαμψη των μηχανισμών ασφαλείας.Κάθε επίθεση μπορεί να προκαλέσει ζημιά.

Τα συστήματα ασφάλειας ελέγχουν τις επιθέσεις κυρίως μέσω της πρόληψης, αλλά εκτός αυτού ενοείται ότι πρέπει να είναι σε θέση και για αναγνώριση και βεβαίως για περιορισμό του κινδύνου.

4.2) Επιτιθέμενοι και στόχοι της ασφάλειας

Σε αυτή την υποενότητητα θα παρουσιάσουμε τις δύο ένοιες που συγκρούονται.Από τη μία πλευρά οι επιτιθέμενοι που στόχο έχουν να βλαψουν το σύστημά μας και από την άλλη οι στόχοι του συστήματος ασφαλείας προς προφύλαξη.Επίσης θα γίνει σύντομη αναφορά σε κάποιες σχεδιαστικές αποφάσεις για την ασφάλεια των συστημάτων.

4.2.1) Επιτιθέμενοι

Σε μία προσπάθεια παρουσίασής τους όχι τόσο αυξανόμενων ικανοτήτων όσο αυξανόμενης βαρύτητας των πράξεών τους (χωρίς βέβαια αυτό να είναι απόλυτο), κάποιες κατηγορίες που περιγράφουν σε γενικές γραμμές τους επιτιθέμενους είναι ‘άτομα που προέρχονται από μέσα’ (insiders), hackers, επαγγελματίες και μέλη οργανωμένου εγκλήματος.Τη στιγμή που οι hackers δρουν με βασικό γνώμονα την περιέργια, τη διασκέδασή τους και αποσκοπούν στην αναγνώριση, έχουμε από την άλλη πλευρά άτομα που η δουλειά τους είναι η κατάριψη των συστημάτων ασφαλείας όπως άτομα που εργάζονται σε μυστικές υπηρεσίες.Τέλος,έχουμε αυτούς που ανήκουν στο οργανωμένο έγκλημα, με κάθε άλλο παρά αγνές προθέσεις και συνήθως αποσκοπούν σε εκβιασμούς και στο να καταστρέψουν τους ανταγωνιστές τους.

4.2.2) Στόχοι των συστημάτων ασφαλείας

Οι έννοιες που ακολουθούν έιναι ο στόχος των επιτιθέμενων, επομένως είναι και ο βασικός στόχος προς προστασία από τα συστήματα ασφαλείας.Πρόκειται για την εμπιστευτικότητα (confidentiality),την ακεραιότητα (integrity),την υπευθυνότητα (accountability) και τη διαθεσιμότητα (availability).Κάθε μία από αυτές ανλύεται σε επιμέρους ορολογία.

4.2.2.1) Confidentiality

Περιλαμβάνει έννοιες όπως η μυστικότητα (secrecy) (περιορίζει την πρόσβαση σε εξουσιοδοτημένα άτομα) , η εχεμύθεια (ως προς την ανάγκη για φύλαξη μυστικών) και η μυστικότητα (privacy) (το δικαίωμα της φύλαξης προσωπικών δεδομένων), δηλαδή ενέργειες που σχετίζονται με το διάβασμα (read).Ειδική περίπτωση αποτελεί η ανωνυμία (anonymity) όντας ικανός να δράσεις χώρις να δώσεις ταυτότητα (παρά μόνο ψευδώνυμο).

4.2.2.2) Integrity/Authenticity

Στην προηγούμενη κατηγορία είχαμε ενέργειες σχετικές με το διάβασμα, τώρα σχετίζονται με την εγγραφή (write).Πρόκειται για την ακεραιότητα των δεδομένων (integrity of data) (προστασία κατά της μη εξουσιοδοτημένης τροποποίησης) και την αυθεντικότητα (οι πληροφορίες είναι προστατευόμενες και συσχετιζόμενες με την ταυτότητα της αρχής).

4.2.2.3) Accountability/Non-repudiability

Η υπευθυνότητα (accountability) αναφέρεται στο γεγονός ότι μία πράξη μπορεί να είναι υπεύθυνα συσχετισμένη με την ταυτότητα της αρχής που είναι υπεύθυνη γι'αυτή την πράξη και η μη απάρνηση (non-repudiability) στο ότι μία πράξη δε μπορεί πλέον να απαγορευθεί αν είναι απαραίτητη για κάποιο συμβόλαιο ή κάποιου είδους αλληλεπίδραση με τις κυβερνητικές αρχές.

4.2.2.4) Availability

Τέλος, η διαθεσιμότητα αναφέρεται στην προστασία του συστήματος ενάντια σε μη εξουσιοδοτημένες βλάβες της λειτουργίας και ένας συνδυασμός της διαθεσιμότητας με την ορθότητα εκφράζει τη αξιοπιστία στην παροχή υπηρεσιών.

4.2.3) Αδυναμίες συστήματος

Όσο αφορά στις αδυναμίες του συστήματος αυτές εντοπίζονται στον κακό σχεδιασμό με την έλλειψη μηχανισμών ασφαλείας και στην κακή implementation (εφαρμογή) όπως για παράδειγμα η υπερχείληση των buffers.Επίσης, ερνητικό στοιχείο είναι η κακή διεύθυνση (administration) όπως είναι οι λογαριασμοί με σταθερούς κωδικούς και η ελειπής χρήση του firewall και η κακή διαχείρηση (management) που σχετίζεται με την πολιτική προστασίας και την ελλειπή προσφορά σε αυτή είτε σε χρόνο είτε σε χρήματα.

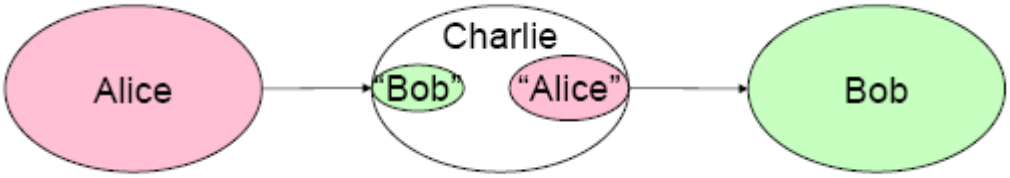
4.3) Τρόποι επίθεσης

Οι βασικοί τρόποι επίθεσης είναι δύο, ωστόσο αξάνονται αν συνυπολογίσουμε κάποιες ειδικές περιπτώσεις.Ο ένας είναι η παθητική επίθεση (passive attacks) όπου ο επιτιθέμενος απλώς διαβάζει (read) πακέτα (sniffing) κάτι που είναι εξαιρετικά εύκολο σε ασύρματα δίκτυα (wireless) και σχετικά εύκολο σε διαμοιραζόμενα δεδμένα όπως το Ethenret.Ο μόνος τρόπος να αντιμετωπιστεί είναι με κρυπτογραφία.Ο άλλος είανι ο ενεργητικός τρόπος (active attacks) όπου ο επιτιθέμενος προσθέτει νέα πακέτα μέσα στο δίκτυο και σε αυτή την περίπτωση η

διεύθυνση της πηγής μπορεί να κλαπεί (ο ακριβής όρος είναι spoofed).Κάνει είτε τυφλές επιθέσεις (blind attacks) όπου ο επιτιθέμενος μπορεί μόνο να διαβάσει και όχι να γράψει είτε επιθέσεις επανάληψης (replay attacks) όπου ο επιτιθέμενος εισάγει στο σύστημα αντίγραφα προήγούμενων πακέτων.

4.3.1) Ειδικές περπτώσεις

Ειδικές περιπτώσεις επιθέσεων αποτελούν οι συνδυασμοι των δύο βασικών περιπτώσεων και η Man-in-the-middle (middleperson) attack.Στην πρώτη περίπτωση έχουμε είτε παθητικη ακολουθούμενη από ενεργητική όπου έχουμε sniffing του κωδικού (passive) και εισαγωγή στο σύστημα χρησιμοποιώντας τον κωδικό (active) είτε ενεργητική επίθεση να διευκολύνει την παθητική με τον επιτιθέμενο να υπονομεύει το σύστημα προώθησης με στόχο να εκτρέψει τηνκυκλοφορία, κάτι που είναι ιδαίτερα εύκολο στο δεύτερο επίπεδο (layer 2), αλλά κάπως δυσκολότερο στο τρίτο (layer 3) και ένας συμβατός router/switch μπορεί να χρησιμοποιηθεί σαν εργαλείο.Στη δεύτερη περίπτωση που αποτελεί μία ειδική περίπτωση της ενεργής επίθεσης, ο man-in-the-middle δημιουργεί την ψευδαίσθηση για κάθε συμμετέχοντα στην επικοινωνία ότι είναι ο άλλος συμμετέχοντας και έτσι τα μηνύματα μπορούν να αντιγραφούν και να τροποποιηθούν.Και εδώ η λύση είναι η κρυπτογραφία.



Σχήμα 8: Man-in-the-middle attack

4.4) Σχεδιαστικές αρχές για ασφαλή συστήματα

Έχοντας παρουσιάσει τους κινδύνους που έχουμε να αντιμετωπίσουμε, είναι πιο εύκολο να σχεδιάσουμε την άμυνά μας γνωρίζοντας το στόχο τους και τον τρόπο δράσης τους.Υπάρχει ένας αριθμός από συμβουλές των οποίων η τήρηση θα οδηγήσει στο σχεδιασμό ασφαλών συστημάτων.Ακολουθεί η ονομαστική αναφορά τους κυρίως για λόγους πληρότητας:

- Λιτότητα του μηχανισμού (Economy of Mechanism) (ο μηχανισμός πρέπει να έχει απλό και μικρό μηχανισμό)
- Fail-safe defaults (ο μηχανισμός προστασίας πρέπει να αρνείται την αποδοχή by default και να την επιτρέπει μόνο όταν υπάρχει ρητή άδεια)
- Πλήρης μεσολάβηση (Complete Mediation) (ο μηχανισμός προστασίας πρέπει να ελέγχει κάθε πρόσβαση σε κάθε αντικείμενο)
- Ανοικτός σχεδιασμός (Open Design) (ο μηχανισμός προστασίας δεν πρέπει να στηρίζεται στην άγνοια των επιτιθέμενων για την αρχιτεκτονική του, αλλά στην άγνοια τους για συγκεκριμένες πληροφορίες (π.χ κωδικοί))

- Διαχωρισμός προνομίων (Separation of Privilege) (ο μηχανισμός προστασίας πρέπει να επιτρέπει την είσοδο μόνο βασιζόμενος σε κάποιο πλήθος πληροφοριών)
- Λιγότερα προνόμια(Least Privilege) (ο μηχανισμός προστασίας πρέπει να ανγκάζει κάθε διεργασία να λειτουργεί με τα λιγότερα δυνατά προνόμια)
- Λιγότερο γνωστός μηχανισμός(Least Common Mechanism) (ο μηχανισμός προστασίας πρέπει να μοιράζεται όσο το δυνατό λιγότερο μεταξύ των χρηστών)
- Άμυνα σε βάθος(Defense in Depth) (πρέπει να υπάρχουν πολλά επίπεδα άμυνας)
- Ασφαλισμός του πιο αδύναμου συνδέσμου(Securing the Weakest Link) (ο μηχανισμός προστασίας δεν πρέπει να έχει αδύναμα σημεία που να θέτουν σε κίνδυνο τα πιο καλά ασφαλισμένα μέρη)
- Απροθυμία πίστης(Reluctance to Trust) (ο μηχανισμός προστασίας δεν πρέπει να δίνει σε κανένα μηχανισμό την άδεια εισόδου χωρίς τα κατάλληλα διαπιστευτήρια)

5) Interoperability και Evolvability

5.1) Interoperability (διαχρηστικότητα)

Μεταξύ μίας εφαρμογής από διαφορετικές πηγές υπάρχουν διαφορές τόσο στην ποιότητα όσο και στην πολυπλοκότητα και βέβαια τίθεται και το θέμα του ελέγχου και της δυνατότητας για αποσφαλμάτωση.Μεταξύ μίας περισσότερο και μίας λιγότερο περίπλοκης εφαρμογής υπάρχουν κάποιες προαιρετικές συναρτήσεις και μεταξύ μίας παλαιότερης και μίας νεότερης τίθεται το θέμα της αξιοπιστίας!Επομένως, μεταξύ μίας υλοποίησης V1 και V2 έχουμε εξελιξιμότητα (evolvability).

Τώρα όσο αφορά στην επεκτασιμότητα για να ενεργοποιήσουμε το V2 πρέπει να είναι ήδη δημιουργημένο σε V1.Για τον τρόπο προσέγγισης έχουμε τόσο τη βασική (standard approaches), όσο και τη διαφορετικη (alternative) με την οποία με μετα-δεδομένα ελέγχουμε την κατάλληλη εκδοχή (πχ. Configuration και directory information)

5.2) Evolvability

Εξελιξιμότητα είναι η ικανότητα να εξελίσεσαι εύκολα.Σχεδιάζοντας να γίνεις μέρος από κάτι άλλο είτε μέσω της εξέλιξης η οποία παροτρίνεται από το εξελισσόμενο περιβάλλον, είτε αφομοιώνοντας μη προβλέψιμες απαιτήσεις.Όσο αφορά στον τρόπο με τον οποίο θα επιτύχουμε την εξελιξιμότητα, δεν υπάρχουν εύκολες ή δύσκολες απαντήσεις.Το μόνο σίγουρο είναι ότι σχεδόν πάντα θα είσαι λανθασμένος αν βελτιστοποιείς τη δουλεία σου με βάση το παρών (αφού τα πρωτόκολλα χρειάζονται μερικά χρόνια μέχρι να φθάσουν στην αγορά), αλλα και στην περίπτωση που βελτιστοποιείς με βάση το άγνωστο μέλλον.

5.3)Εξέλιξη στο IP πρωτόκολλο

Στην πραγματικότητα το IP δεν έχει εξελιχθεί ιδιαίτερα, παρ'όλα αυτά ας δούμε μερικά στοιχεία για διάφορες καινοτομίες που δοκιμάστηκαν πάνω σε αυτό το πρωτόκολλο.Ποιες δούλεψαν και ποιες όχι.

Πρόκειται για μία αρχιτεκτονική διευθυνσιοδότησης δύο διαστάσεων (net/interface) σε 32 bit.Αρχικά ήταν 8+24 και με μία σχετική διακύμανση ανάλογα με τις διάφοορες κλάσεις(7+24 A, 14+16 B, 21+8 C) και στη συνέχεια είχαμε το IPv6 και πλήρη ανασχεδιασμό.

Μία καινοτομία που εφαρμόστηκε στο IP ήταν το IP multicast το οποίο χρησιμοποιεί το εύρος διυθύνσεων της κλάσης D που πριν δεν χρησιμοποιούνταν και έχει σαν νέο host-to-router πρωτόκολλο το IGMP το οποίο απαιτεί αλλαγές μεταξύ host και router

και βέβαια είχε τεράστια επιρροή στην υποδομή της δρομολόγησης.Απέτυχε μεν σε παγκόσμιο επίπεδο ανάπτυξης, δουλεύει δε σε εταιρικά δίκτυα ή σε ειδικά περιβάλλοντα (π.χ ακαδημαικό).

Ένα άλλο πρωτόκολλο σήμανσης, το RSVP απαιτεί και αυτό αλλαγές μεταξύ host και router.Επίσης είναι αρκετά ανεπτυγμένο αλλά δεν είναι σε λειτουργία και έτσι οι εφαρμογές δεν ξέρουν πώς να το χρησιμοποιούν.Απέτυχε, αφού κανείς δεν πλήρωσε για διαγύλαξη πόρων.

Τέλος έχουμε το ECN το οποίο όμως ήταν κακώς καθοδηγημένο (misguided) (έστελνε παραπάνω σήματα στις συγκρούσεις σημάτων) και ποτέ δεν καθορίστικε σαφώς (όπως το πότε να σταλλούν τα σήματα και τι να κάνουν στους hosts).Ενώ στο TCP έχουμε ένα σήμα για τον έλεγχο συγκρούσεων (packet drop), στο ECN έχουμε ένα ακόμα bit για τις πληροφορίες από το router στο host.Και σε αυτό το πρωτόκολλο όμως έχουμε αργή εξέλιξη και υπάρχουν διάφορα προβλήματα, ωστόσο είναι ακόμα νωρίς να το χαρακτηρίσουμε σαν αποτυχία.

Σε γενικές γραμμές εκτός από το TTL όλα τα μεγέθη παιδίων ήταν λανθασμένα (κάτι όμως που δικαιολογείται από τη ραγδαία εξέλιξη της τεχνολογίας στο μεσοδιάστημα)και σχεδόν όλες οι καινοτομίες για το IP μέχρι το 1990 απέτυχαν.Το IPv6 είναι καλύτερο πρωτόκολλο, αλλά δυστυχώς δεν είναι ξεκάθαρο σε όλες τις αγορές το κίνητρο για την εξέλιξή του.

6) Υποθέσεις για πρωτόκολλα του μέλλοντος

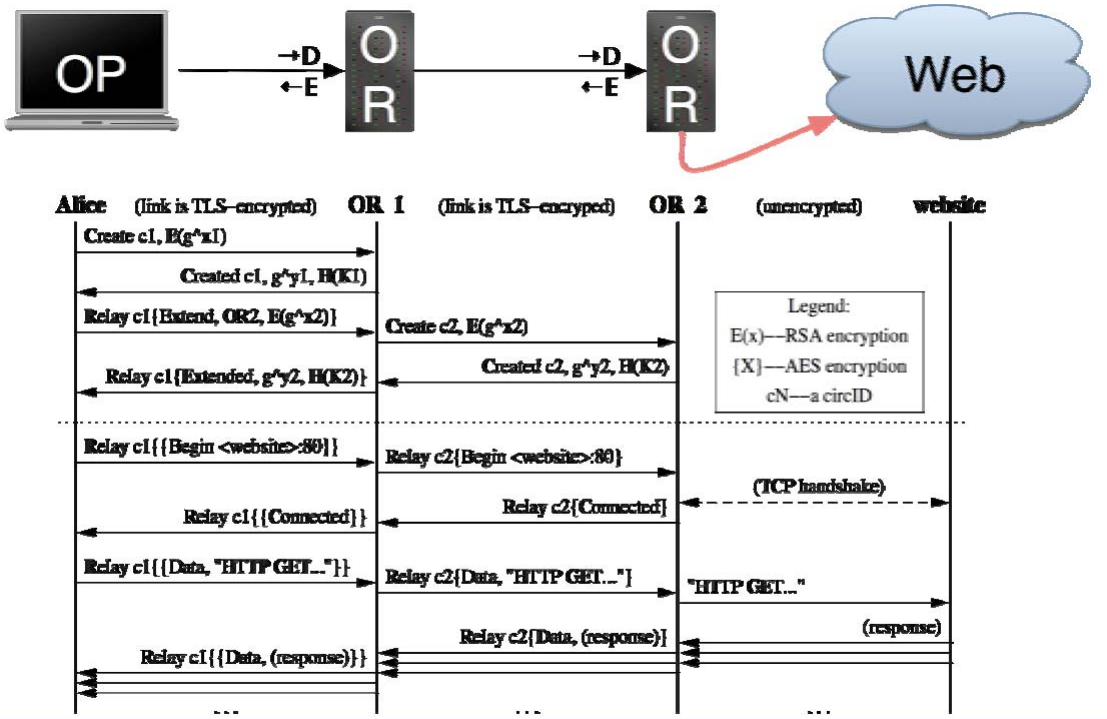
Στο τέλος αυτής της αναφοράς σε πρωτόκολλα, σχεδιασμό πρωτοκόλλων, προβλήματα που προκύπτουν κατά το σχεδιασμό και τη λειτουργία τους και βέβαια μετά τη σύντομη αναφορά για την εξέλιξή τους μέσα στα χρόνια, νομίζω ότι θα ήταν ενδιαφέρον να παρουσιάσουμε χαρακτηριστικά από ένα πρωτόκολλο που πιθανό να μας απασχολήσει στο μέλλον.

6.1) Anonymity in the Internet (ανωνυμία στο Internet)

Οι χρήστες του Internet μπορεί να θέλουν αν μείνουν ανώνυμοι είτε εξαιτίας αυτών που παρέχουν κάποια υπηρεσία (να αποφύγουν πολύ μεγάλα μεγέθη πληροφορίας, να παρακάμψουν κάποιους περιορισμούς της χώρας τους), είτε εξαιτίας άγνωστων αντιπαλοτήτων (να προστατευθεί κάποιος από μυστικές υπηρεσίες, να προστατευθεί λενα θύμα από εγκληματική επίθεση).Στην ουσία η ένοια της ανωνυμίας είναι το αντίθετο της accountability, αφού στόχος είναι οι ενέργειες μας να μην μπορούν να ‘επιστρέψουν σε εμάς’ (be tracked back to you), θα μπορούν βέβαια να επιστρέψουν στο set ανωνυμίας σου (anonymity set).Το πρόβλημα όμως που προκύπτει είναι όταν θέλουμε να μπορούμε να λάβουμε αντίστροφη επικοινωνία.Αυτό λύνεται από τη βασική ιδέα η οποία είναι η εξής:Ένας χρήστης μιλάει σε ένα ενδιάμεσο,ο ενδιάμεσος μιλάει στον άλλο χρήστη και έτσι ο πρώτος χρήστης ‘κρύβεται’ πίσω από τον ενδιάμεσο.Βέβαια υπάρχουν διάφορα θέματα που καθυστερούν την ανάπτυξή του.Κατ’αρχάς η ανωνυμία δε μπορεί να δημιουργηθεί από τον αποστολέα ή από τον παραλήπτη, επομένως κάποιος άλλος πρέπει να παράγει κυκλοφορία (traffic) και να καλύψει τον ‘ανωνυμο αποστολέα’.Έτσι προκύπτουν θέματα χρηστικότητας, αποτελεσματικότητας και κόστους.

6.2)The Onion Router (TOR)

TOR διευθύνσεις παρέχουν ανωνυμία μέσω μίας αλυσίδας από anonymizers, τους onion routers.Αυτοί επιλέγονται από την πηγή (onion proxy OP) και για κάθε κύκλωμα κάθε OR γνωρίζει μόνο από που πήρε τα δεδομένα και που θα τα στείλει.Επίσης όλη η κυκλοφορία (traffic) είναι 512-bytes κελιά, επομένως η ανάλυσή της είναι δυσκολότερη.Ακολουθεί σχηματική αναπαράσταση του TOR:



Σχήμα 9: The Onion Router

Όσο αφορά στην ανάπτυξή του , το σχέδιο πρέπει να υλοποιηθεί και να χρησιμοποιηθεί στον πραγματικό κόσμο, έτσι μόνο δεν θα έιναι ακριβό στην εκτέλεσή του.Σε καμία περίπτωση δεν πρέπει να επιρηφθεί βάρος υποχρεώσεων στους χρήστες, για παράδειγμα επιτρέποντας εμπλοκές των onion routers σε παράνομες δραστηριότητες και επισης δεν πρέπει να έιναι ακριβό στην εφαρμογή.Τέλος δεν πρέπει να απαιτεί μη-ανώνυμες ομάδες να ‘τρέχουν’ (run) το TOR και τέλος η πλευρά του πελάτη πρέπει να είναι εύκολα εφαρμόσιμη σε κάθε πλατφόρμα, για παράδειγμα δε θα ήταν λογικό να απαιτείται από τους χρήστες να αλλάξουν το λειτουργικό τους σύστημα προκειμένου να είναι ανώνυμοι.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1 http://www.netlab.tkk.fi/~jo/teaching/pd/

2 http://en.wikipedia.org/wiki/Main_Page

3 Δίκτυα Δημόσιας Χρήσης και Διασύνδεση Δικτύων (Χρήστος Ι. Μπούρας)

4 Δίκτυα Υπλογιστών (Andrew S. Tanenbaum)