

3 ΕΠΙΠΕΔΟ ΣΥΝΔΕΣΗΣ ΔΕΔΟΜΕΝΩΝ

3.1 ΕΙΣΑΓΩΓΗ

Το επίπεδο σύνδεσης δεδομένων (Data Link Layer) ασχολείται με θέματα όπως η πλαισίωση των δεδομένων (Data Framing), ο έλεγχος (πειθαρχίας) της γραμμής (Line Discipline), ο έλεγχος ροής (Flow Control) και ο έλεγχος σφαλμάτων (Error Control).

3.2 ΠΛΑΙΣΙΩΣΗ ΔΕΔΟΜΕΝΩΝ

Μια από τις βασικότερες λειτουργίες του επιπέδου σύνδεσης δεδομένων είναι η πλαισίωση των δεδομένων (Data Framing).

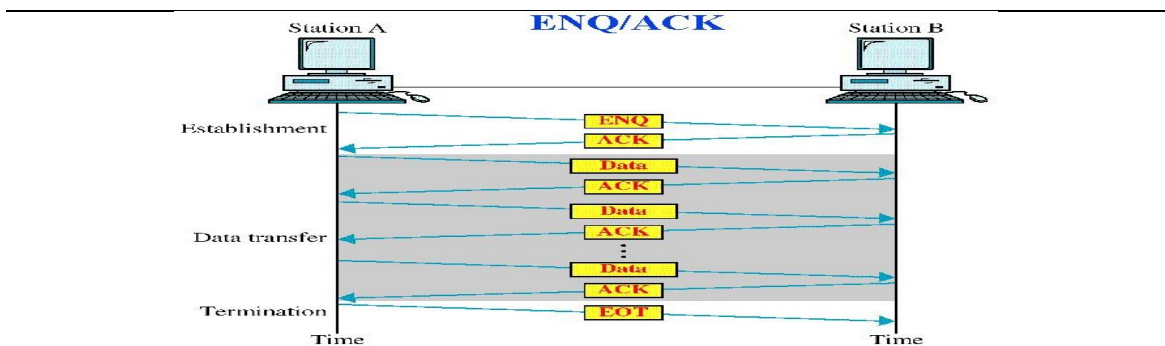
Τα δεδομένα ομαδοποιούνται σε πλαίσια (frames) των οποίων το μέγεθος είναι κυμαινόμενο και εξαρτάται από τις ιδιαιτερότητες του πρωτοκόλλου που χρησιμοποιείται. Τα πλαίσια χωρίζονται σε πεδία (fields), τα bits ή bytes των οποίων είτε χρησιμεύουν για τις λειτουργικές ανάγκες του καναλιού επικοινωνίας (ή γενικότερα του συστήματος επικοινωνίας), είτε απλά μεταφέρουν τα δεδομένα των χρηστών. Έτσι, υπάρχουν τα πεδία διευθύνσεων (Address Field), ελέγχου (Control Field), πληροφορίας (Information Field) και άλλα, ανάλογα με το πρωτόκολλο που χρησιμοποιείται.

3.3 ΕΛΕΓΧΟΣ ΓΡΑΜΜΗΣ

Η λειτουργία αυτή ελέγχει τη σωστή χρήση της γραμμής ώστε η κάθε συσκευή να μπορεί να μεταδίδει ή να δέχεται δεδομένα σε καθορισμένο χρονικό διάστημα. Υπάρχουν 2 τεχνικές:

3.3.1 ENQUIRY / ACKNOWLEDGEMENT

Χρησιμοποιείται για επικοινωνία μεταξύ 2 ισοτίμων (peer-to-peer) συσκευών. Η επικοινωνία πραγματοποιείται είτε σε full duplex, είτε σε half duplex μορφή. Η συσκευή που θέλει να στείλει δεδομένα, στέλνει στην άλλη ένα ENQ frame. Αν ο υποψήφιος δέκτης είναι σε θέση να δεχθεί τα δεδομένα, επιστρέφει ένα ACK frame, αλλιώς στέλνει ένα NACK (Negative Acknowledgement) frame. Αν ο δέκτης δεν αποκριθεί μέσα σε κάποιο χρονικό διάστημα, ο πομπός κάνει μέχρι 3 προσπάθειες, στέλνοντας ENQ frames. Αν δεν λάβει απάντηση, αποσυνδέεται και περιμένει κάποια άλλη χρονική στιγμή. Το τέλος αποστολή δεδομένων δηλώνεται με ένα πλαίσιο EOT (End of Transmission).



Εικόνα 3.1: Τεχνική ENQ / ACK για τον έλεγχο της γραμμής

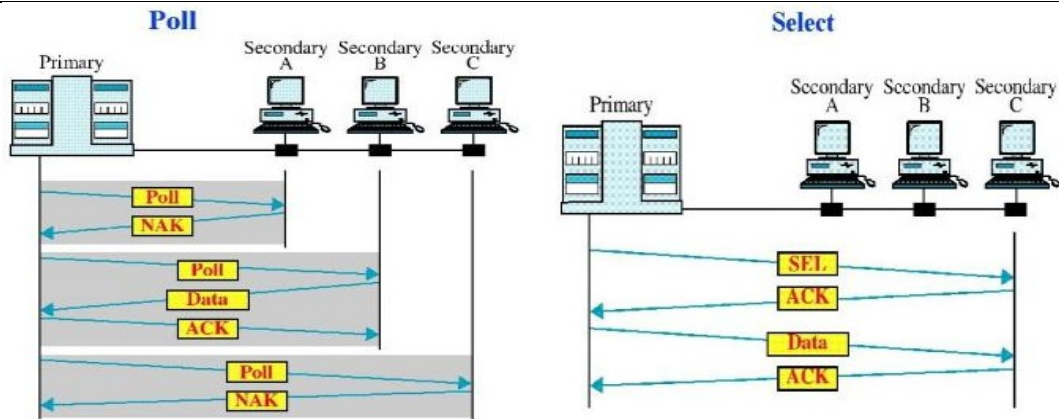
3.3.2 POLL / SELECT

Χρησιμοποιείται όταν στη γραμμή υπάρχουν πολλές συσκευές (multipoint τοπολογίες). Η μία από αυτές είναι η πρωτεύουσα (primary) συσκευή, ενώ οι άλλες είναι οι δευτερεύουσες (secondary). Όλες οι επικοινωνίες μπορούν να γίνουν διαμέσου της πρωτεύουσας συσκευής, καθώς αυτή ελέγχει τη γραμμή (line).

Με την 'poll' διαδικασία, η πρωτεύουσα "ρωτάει" την κάθε δευτερεύουσα αν έχει δεδομένα να στείλει. Αν ναι, η δευτερεύουσα απαντάει στέλνοντας frames δεδομένων. Αν όχι, στέλνει ένα NAK frame και η primary συσκευή περνάει στη επόμενη secondary. Ανάλογα με το πρωτόκολλο που χρησιμοποιείται, η πρωτεύουσα συσκευή, είτε επιβεβαιώνει (ACK) το κάθε frame δεδομένων, είτε αφήνει τη δευτερεύουσα να στέλνει συνεχώς πλαίσια. Ο τερματισμός αποστολής γίνεται είτε με ένα πλαίσιο EOT, είτε η πρωτεύουσα συσκευή διακόπτει τη σύνδεση μετά από κάποιο χρονικό διάστημα.

Με τη διαδικασία 'select', η πρωτεύουσα συσκευή επιλέγει τη δευτερεύουσα, στην οποία θα στείλει δεδομένα. Πρώτα, στέλνει σε αυτή ένα πλαίσιο SEL (Select). Αυτό περνάει από όλες τις δευτερεύουσες συσκευές, αλλά μόνο αυτή που αναγνωρίζει τη διεύθυνσή της μπορεί να διαβάσει το πλαίσιο. Αν η δευτερεύουσα συσκευή είναι σε θέση να δεχθεί δεδομένα, επιστρέφει ένα πλαίσιο ACK.

Από τα όσα ειπώθηκαν παραπάνω, είναι φανερό ότι η κάθε συσκευή διαθέτει μια μοναδική φυσική διεύθυνση που την ξεχωρίζει από τις άλλες μέσα στην multipoint τοπολογία.



Εικόνα 3.2: Διαδικασίες Poll και Select για τον έλεγχο της γραμμής επικοινωνίας

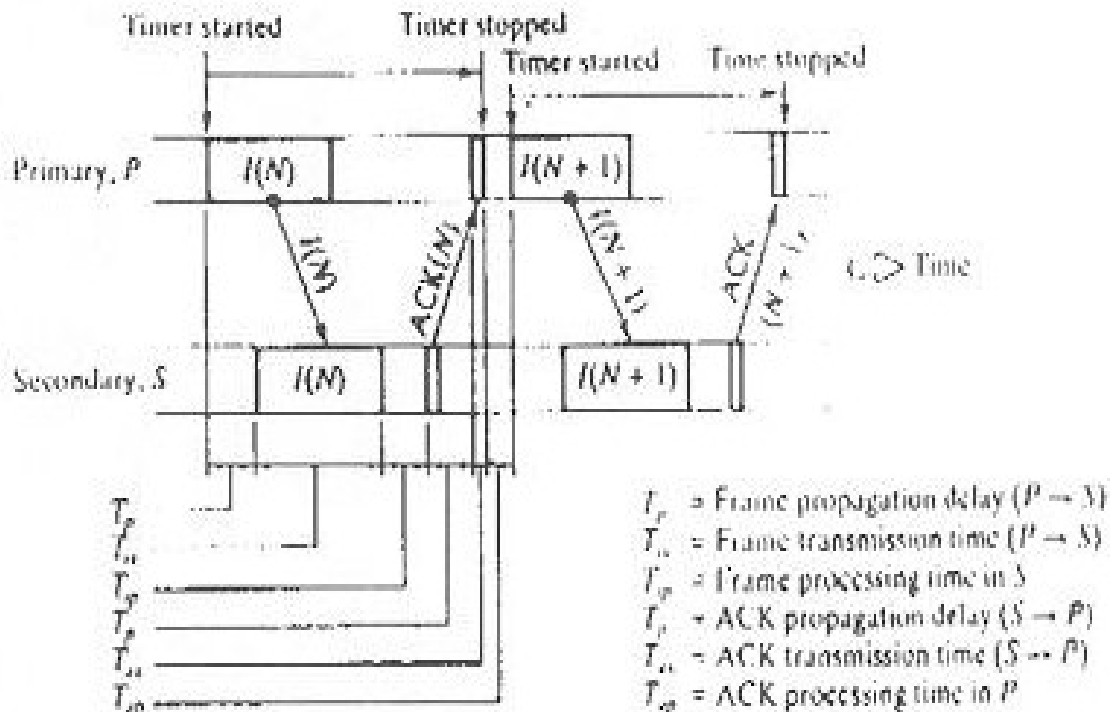
3.4 ΕΛΕΓΧΟΣ ΡΟΗΣ

Υπάρχουν 2 τύποι πρωτοκόλλων για τον έλεγχο ροής, έτσι ώστε η συσκευή λήψης των πλαισίων να μην πλημμυρίζει, λόγω καταγισμού δεδομένων.

3.4.1 ΠΡΩΤΟΚΟΛΛΑ ΠΑΥΣΗΣ / ΑΝΑΜΟΝΗΣ (STOP – AND – WAIT PROTOCOLS)

Ο πομπός στέλνει το κάθε πλαίσιο χωριστά και περιμένει την επιβεβαίωση λήψης (ACK) για να συνεχίσει με την αποστολή ενός επόμενου πλαισίου. Τα πρωτόκολλα αυτά

παρουσιάζουν σημαντικό μειονέκτημα, αφού είναι αργά και χρονοβόρα, ενώ δεν γίνεται χρήση όλου του εύρους ζώνης.



Εικόνα 3. 3: Το πρωτόκολλο παύσης – αναμονής χωρίς διόρθωση σφαλμάτων

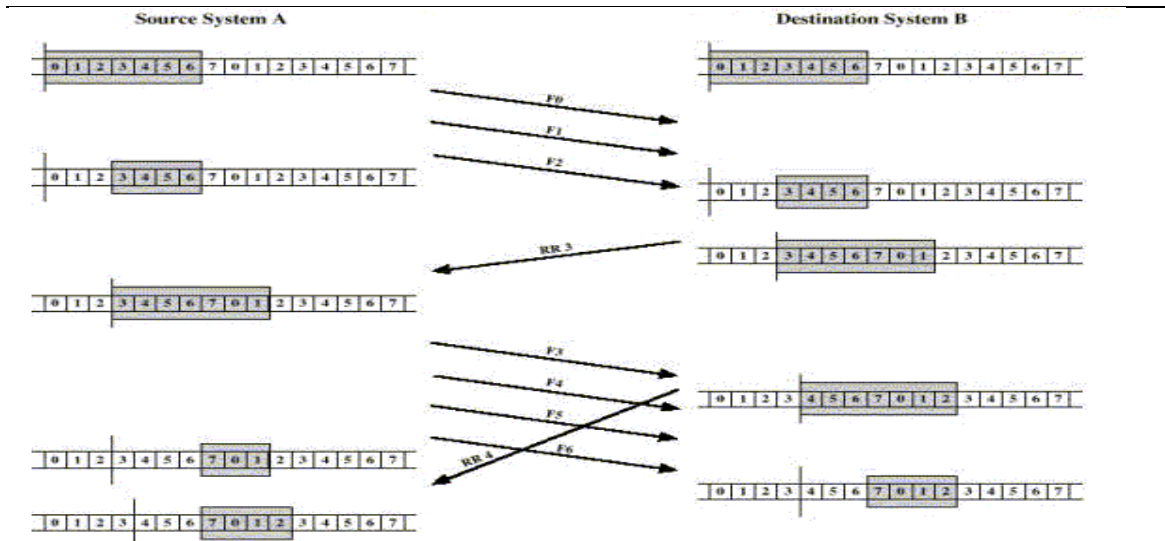
3.4.2 ΠΡΩΤΟΚΟΛΛΑ ΣΥΡΟΜΕΝΟΥ ΠΑΡΑΘΥΡΟΥ (SLIDING – WINDOW PROTOCOLS)

Ο όρος συρόμενο παράθυρο (Sliding Window) αναφέρεται στη μέγιστη ομάδα από frames που μπορούν να περάσουν διαμέσου της γραμμής μετάδοσης. Τα frames φέρουν αύξοντα αριθμό από 0 έως n-1, οπότε το συρόμενο παράθυρο περιλαμβάνει n frames.

Στον αποστολέα, το συρόμενο παράθυρο εκφράζει ένα τμήμα των, προς αποστολή, frames, τα οποία βρίσκονται στον ενταμιευτή του (buffer). Καθώς τα frames φεύγουν, το παράθυρο συρρικνώνεται, ενώ όταν ληφθεί κάποιο ACK τότε το παράθυρο μεγαλώνει για να συμπεριλάβει τόσα frames, όσα επιβεβαιώθηκαν στον παραλήπτη.

Στον δέκτη, το συρόμενο παράθυρο εκφράζει άδειες θέσεις στον ενταμιευτή. Καθώς λαμβάνονται πακέτα, αυτό συρρικνώνεται. Όταν στέλνεται στον πομπό κάποιο ACK frame, το παράθυρο αυξάνει κατά τόσες κενές θέσεις, όσα είναι τα frames που επιβεβαιώθηκαν.

Τα πρωτόκολλα συρόμενου παραθύρου κάνουν σχεδόν πλήρη εκμετάλλευση του εύρους ζώνης και είναι ταχύτερα από τα πρωτόκολλα παύσης / αναμονής.



Εικόνα 3.4: Έλεγχος ροής με την μέθοδο συρόμενου παραθύρου

Σημείωση: Όταν ο δέκτης στείλει ACK για το k frame ενός παραθύρου των n frames, έχει επιβεβαιώσει τη λήψη όλων των frames ως το k, αλλά όχι και του k.

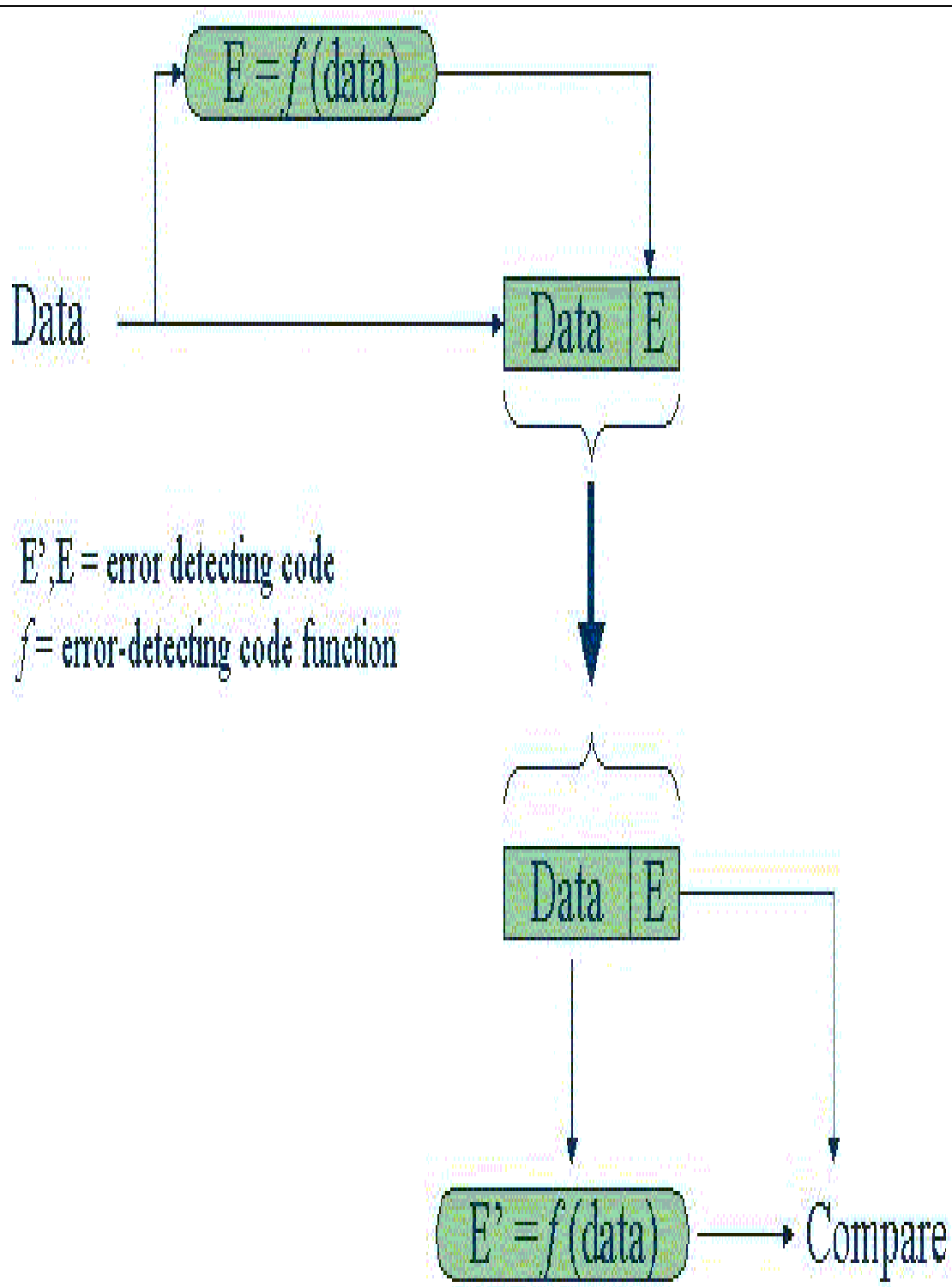
3.5 ΣΦΑΛΜΑΤΑ ΜΕΤΑΔΟΣΗΣ

Κατά τη μετάδοση των αναλογικών, κυρίως, σημάτων είναι δυνατό να πραγματοποιηθούν αλλαγές στη μορφή τους και χρονικές καθυστερήσεις, εξαιτίας φαινομένων θορύβου. Αυτό οδηγεί σε αλλοίωση των δεδομένων και ο δέκτης λαμβάνει εσφαλμένες πληροφορίες. Υπάρχουν 2 τύποι σφαλμάτων μετάδοσης:

- **Σφάλματα απλού Bit (Single-Bit Errors):** Μόνο ένα bit έχει αλλάξει στην μονάδα των δεδομένων (Data Unit, π.χ. Byte, ...), από '0' σε '1' ή από '1' σε '0'. Τέτοιου είδους σφάλματα συναντώνται, κυρίως, σε περιπτώσεις παράλληλης μετάδοσης. Για να συμβούν σε σειριακή μετάδοση, πρέπει ο θόρυβος να είναι στιγμιαίος (πολύ μικρός σε διάρκεια), πράγμα εξαιρετικά απίθανο.
- **Σφάλματα Καταιγισμού (Burst Errors):** Περισσότερα από ένα bit έχουν αλλάξει τιμή. Ο αριθμός των bit από το πρώτο ως το τελευταίο εσφαλμένο bit αποτελεί το μήκος του (σφάλματος) καταιγισμού. Αυτού του είδους τα σφάλματα είναι συνηθέστερα και πραγματοποιούνται στη σειριακή μετάδοση.

3.5.1 ΑΝΙΧΝΕΥΣΗ ΣΦΑΛΜΑΤΩΝ

Οι τεχνικές που χρησιμοποιούνται για την ανίχνευση των σφαλμάτων σε μια μονάδα δεδομένων είναι οι ακόλουθες:



Εικόνα 3.5: Διαδικασία ανίχνευσης σφαλμάτων

3.5.1.1 Vertical Redundancy Check (VRC)

Η μέθοδος αυτή είναι γνωστή και ως parity check (έλεγχος ισοτιμίας). Σύμφωνα με αυτή, στο τέλος του κάθε Data Unit εισάγεται ένα bit ισοτιμίας (parity bit). Αν ενδιαφέρει ο έλεγχος άρτιας ισοτιμίας, τότε:

1. Αν ο αριθμός των '1' είναι περιττός, το bit ισοτιμίας είναι το '1' ώστε ο αριθμός να γίνει άρτιος.
2. Αν ο αριθμός των '1' είναι άρτιος, το bit ισοτιμίας είναι το '0' ώστε ο αριθμός να παραμείνει άρτιος.

Όταν το data unit φθάσει στο δέκτη, αυτός μετράει τους άσσους και αν τους βρει περιττούς, αποβάλλει το data unit. Η μέθοδος αυτή παρουσιάζει αδυναμία όταν ένας άρτιος αριθμός από bits αλλάζει τιμή, λόγω θορύβου.

Τα αντίθετα συμβαίνουν σε περιπτώσεις όπου ενδιαφέρει η περιττή ισοτιμία.

3.5.1.2 Longitudinal Redundancy Check (LRC)

Σύμφωνα με την τεχνική LRC, στο τέλος του κάθε frame προστίθεται ένα byte (8 bits). Το πρώτο bit αποτελεί το bit ισοτιμίας όλων των πρώτων bits που υπάρχουν στα bytes δεδομένων του frame, το δεύτερο bit είναι bit ισοτιμίας όλων των δεύτερων bits, κ.ο.κ. Σε σύγκριση με τη μέθοδο VRC, η LRC αυξάνει την πιθανότητα εύρεσης σφαλμάτων, αλλά μειονεκτεί όταν πραγματοποιηθεί άρτιος αριθμός αλλαγών σε bits που βρίσκονται σε ίδια θέση μέσα στα bytes των frames (περίπτωση άρτιας ισοτιμίας).

3.5.1.3 Cyclic Redundancy Check (CRC)

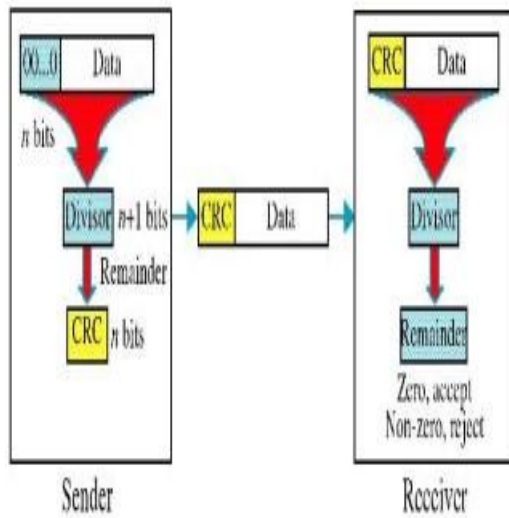
Η τεχνική αυτή είναι πιο αποτελεσματική στην ανίχνευση σφαλμάτων. Βασίζεται στη δυαδική διαίρεση. Ο πομπός προσαρμόζει μια ακολουθία από $n-1$ '0' στο τέλος των δεδομένων και διαιρεί το σύνολο με ένα διαιρέτη (CRC Generator των n bits). Το υπόλοιπο (δηλαδή, το CRC των $n-1$ bits) αντικαθιστά την ακολουθία των '0' στο τέλος των δεδομένων, ώστε η διαίρεση να είναι τέλεια. Ο δέκτης επαναλαμβάνει τη δυαδική διαίρεση και αν είναι τέλεια, δέχεται τα δεδομένα, αλλιώς απορρίπτει το frame.

Ο γεννήτορας CRC μπορεί να αναπαρασταθεί και με μορφή πολυωνύμων, για συντομία και για τη μαθηματική τεκμηρίωση της τεχνικής. Με βάση τα πολυώνυμα, ο γεννήτορας CRC επιλέγεται έτσι ώστε να μην διαιρείται από το x και να διαιρείται από το $(x + 1)$.

Η μέθοδος αυτή μπορεί να ανιχνεύει σε περιττού αριθμού bits, τα λάθη μήκους μικρότερου ή ίσου του βαθμού του πολυωνύμου, και με μεγάλη πιθανότητα τα λάθη καταιγισμού, μήκους μεγαλύτερου του βαθμού του πολυωνύμου.

Τα πιο γνωστά πρωτόκολλα είναι τα: CRC – 12, CRC – 16, CRC – 32 και CRC – ITU – T.

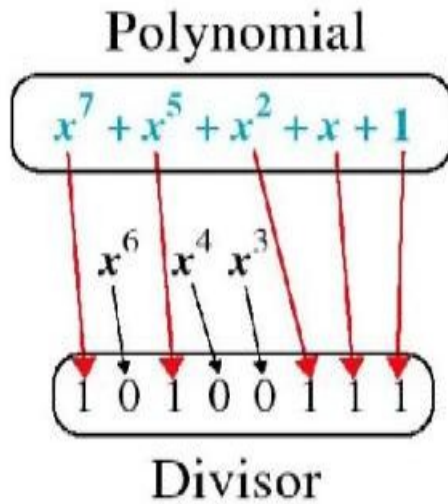
CRC



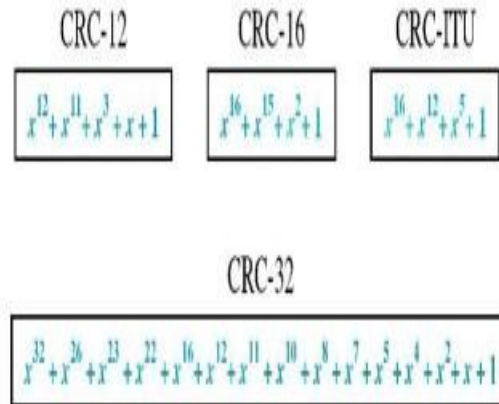
Binary Division



Polynomial and Divisor



Standard Polynomials

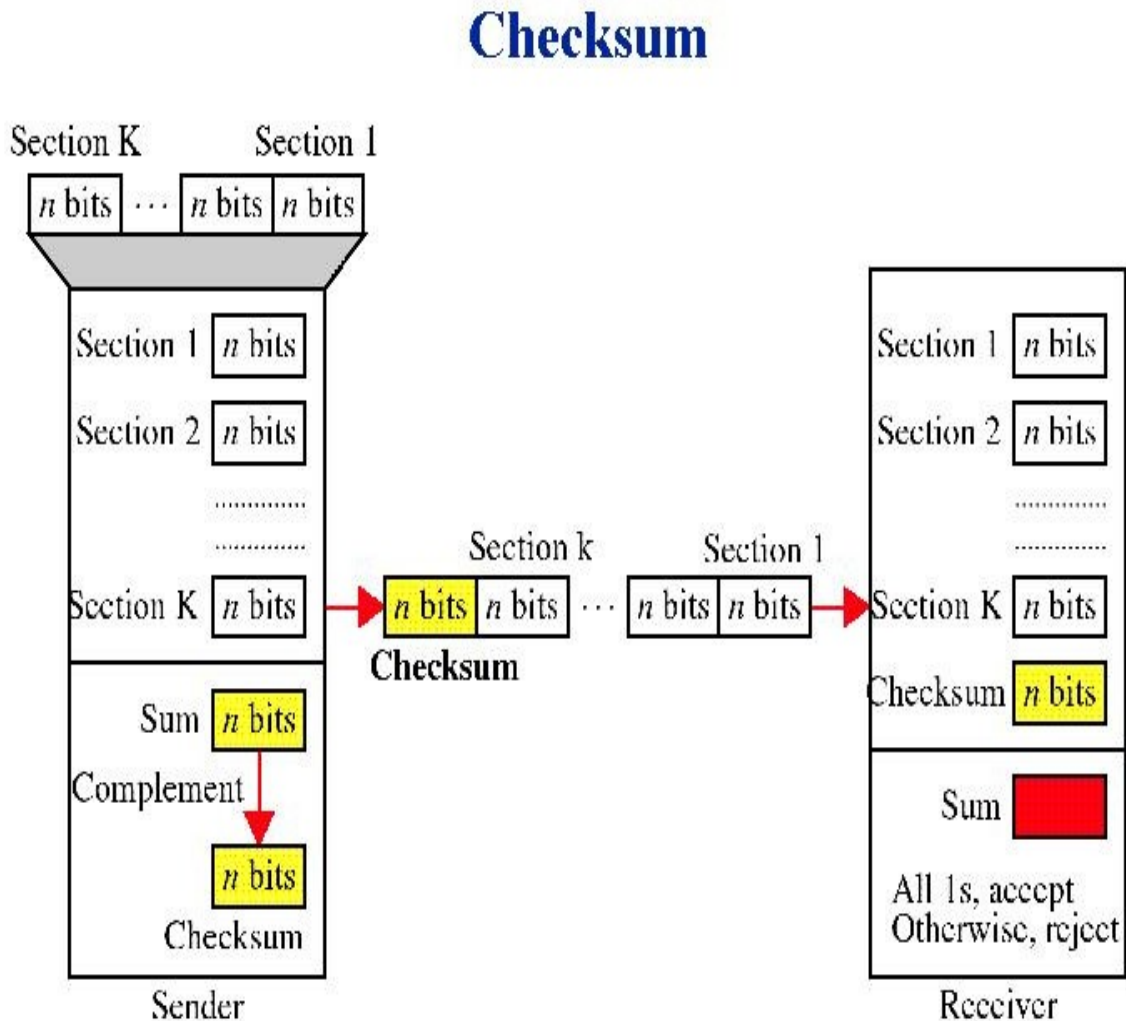


Εικόνα 3.6: Διαδικασία ανίχνευσης σφαλμάτων CRC. Παράδειγμα δυαδικής διαίρεσης. Πολυωνυμική μετατροπή του Διαιρέτη CRC. Τα πρότυπα CRC

3.5.1.4 Έλεγχος Αθροίσματος (Checksum)

Η μέθοδος αυτή χρησιμοποιείται σε επίπεδο ανώτερο από το επίπεδο σύνδεσης δεδομένων. Τα μεγάλα πακέτα δεδομένων (data segments) διασπώνται σε μικρότερα και αθροίζονται με χρήση αριθμητικής συμπληρώματος του 1. Το άθροισμα ρυθμίζεται έτσι ώστε να είναι συγκεκριμένου μήκους (π.χ. n bits). Στο τέλος του πακέτου δεδομένων (Data Segment) προσαρτάται το συμπληρωματικό του αθροίσματος αυτού (Checksum Field). Ο δέκτης επαναλαμβάνει την παραπάνω διαδικασία κάνοντας χρήση και του checksum. Έτσι, αν το ολικό άθροισμα είναι μηδέν, το πακέτο λαμβάνεται ως σωστό. Στην αντίθετη περίπτωση, κάποιο σφάλμα έχει συμβεί και το πακέτο απορρίπτεται.

Η παραπάνω μέθοδος είναι αποτελεσματική για όλα του περιττού αριθμού λάθη και για τα περισσότερα λάθη αρτίου αριθμού. Όμως, παρουσιάζει αδυναμία όταν αλλάζουν κάποια bits ενός πακέτου και τα αντίστοιχα bits αντίθετων τιμών ενός άλλου, οπότε κατά το άθροισμα στον δέκτη, το ολικό πακέτο γίνεται δεκτό, ενώ είναι λανθασμένο.

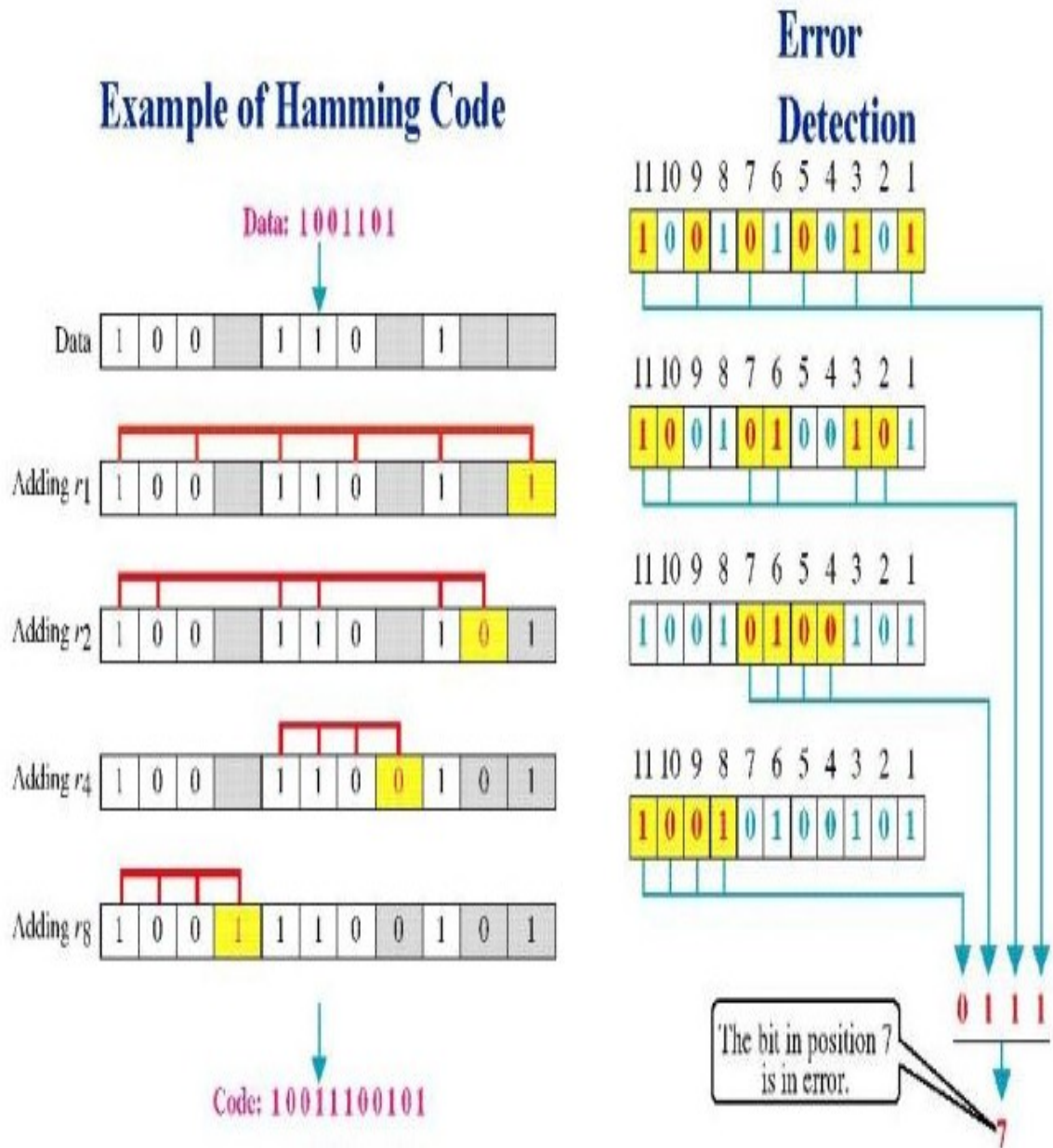


Εικόνα 3.7: Η διαδικασία ανίχνευσης σφαλμάτων CheckSum

3.5.2 ΔΙΟΡΘΩΣΗ ΣΦΑΛΜΑΤΩΝ

Η διαδικασία διόρθωσης σφαλμάτων είναι πιο πολύπλοκη από αυτή της ανίχνευσης και για αυτό με μεγαλύτερο κόστος. Δημιουργεί την ανάγκη για περισσότερα bits ελέγχου, τόσο για την ανίχνευση, όσο και για τη διόρθωση των λαθών.

Όσον αφορά την ανεύρεση και διόρθωση σφαλμάτων ενός bit, έχει αποδειχθεί ότι αν r είναι ο αριθμός των bits ελέγχου (redundancy bits) και m είναι ο αριθμός των bits των δεδομένων, τότε πρέπει $2^r \geq m + r + 1$ (δηλαδή, το r πρέπει να μπορεί να ξεχωρίζει $m+r+1$ καταστάσεις).



Εικόνα 3.8: Παράδειγμα λειτουργίας του κώδικα Hamming για την διόρθωση σφαλμάτων

Για τη διόρθωση σφάλματος απλού bit έχει αναπτυχθεί ο κώδικας Hamming. Το κάθε bit ελέγχου αποτελεί bit άρτιας ισοτιμίας (parity bit) συγκεκριμένων bits μέσα στο frame των δεδομένων. Το πρώτο bit ελέγχου κατέχει την 0...001 θέση μέσα στο frame και ελέγχει την άρτια ισοτιμία με όλα τα bits που καταλαμβάνουν θέσεις, των οποίων ο αύξων αριθμός είναι της μορφής xxx...xx1, όπου $x \in \{0,1\}$, κ.ο.κ.

Όταν το frame των δεδομένων φθάσει στον δέκτη, αυτός κάνει τον έλεγχο άρτιας ισοτιμίας για τις διάφορες ομάδες με το αντίστοιχο bit ελέγχου και σχηματίζει ένα δυαδικό αριθμό με βάση τις καινούργιες τιμές ισοτιμίας. Ο σχηματιζόμενος δυαδικός αριθμός δείχνει τη θέση μέσα στο frame, όπου βρίσκεται το λανθασμένο bit. Αν το frame είναι σωστό, ο δυαδικός αριθμός περιλαμβάνει μηδενικά ψηφία (π.χ. 000...0).

Για τη διόρθωση λαθών καταγισμού, ο κώδικας Hamming μπορεί να μετασκευαστεί, αλλά τα bit ελέγχου είναι υπερβολικά περισσότερα από αυτά για τη διόρθωση απλού λάθους.

3.5.3 ΑΝΑΛΥΣΗ ΣΦΑΛΜΑΤΩΝ ΜΕ ΧΡΗΣΗ ΘΕΩΡΙΑΣ ΠΙΘΑΝΟΤΗΤΩΝ

Αν p είναι η πιθανότητα ένα bit να είναι λανθασμένο και N είναι το μήκος του πλαισίου σε αριθμό bit, τότε η πιθανότητα να υπάρχουν ακριβώς k λανθασμένα bit σε ένα πλαίσιο δίνεται

$$\text{από τη σχέση: } P_k = \sum \binom{N}{k} p^k (1-p)^{N-k}, \text{ όπου } \binom{N}{k} = \frac{N!}{k!(N-k)!}.$$

Από τον παραπάνω τύπο προκύπτει ότι: $P_0 = (1-p)^N$, $P_N = p^N$ και $P_1 = Np(1-p)^{N-1}$.

Εφαρμογή: Αν $p = 10^{-3}$ και $N = 10$, να βρεθεί η πιθανότητα να υπάρχει τουλάχιστον ένα λανθασμένο bit σε ένα μεταδιδόμενο πλαίσιο.

Έχουμε: $P(\text{τουλάχιστον ένα λανθασμένο bit}) = P_1 + P_2 + \dots + P_N = 1 - P_0 = 1 - (1 - p)^N \approx 10^{-2}$.

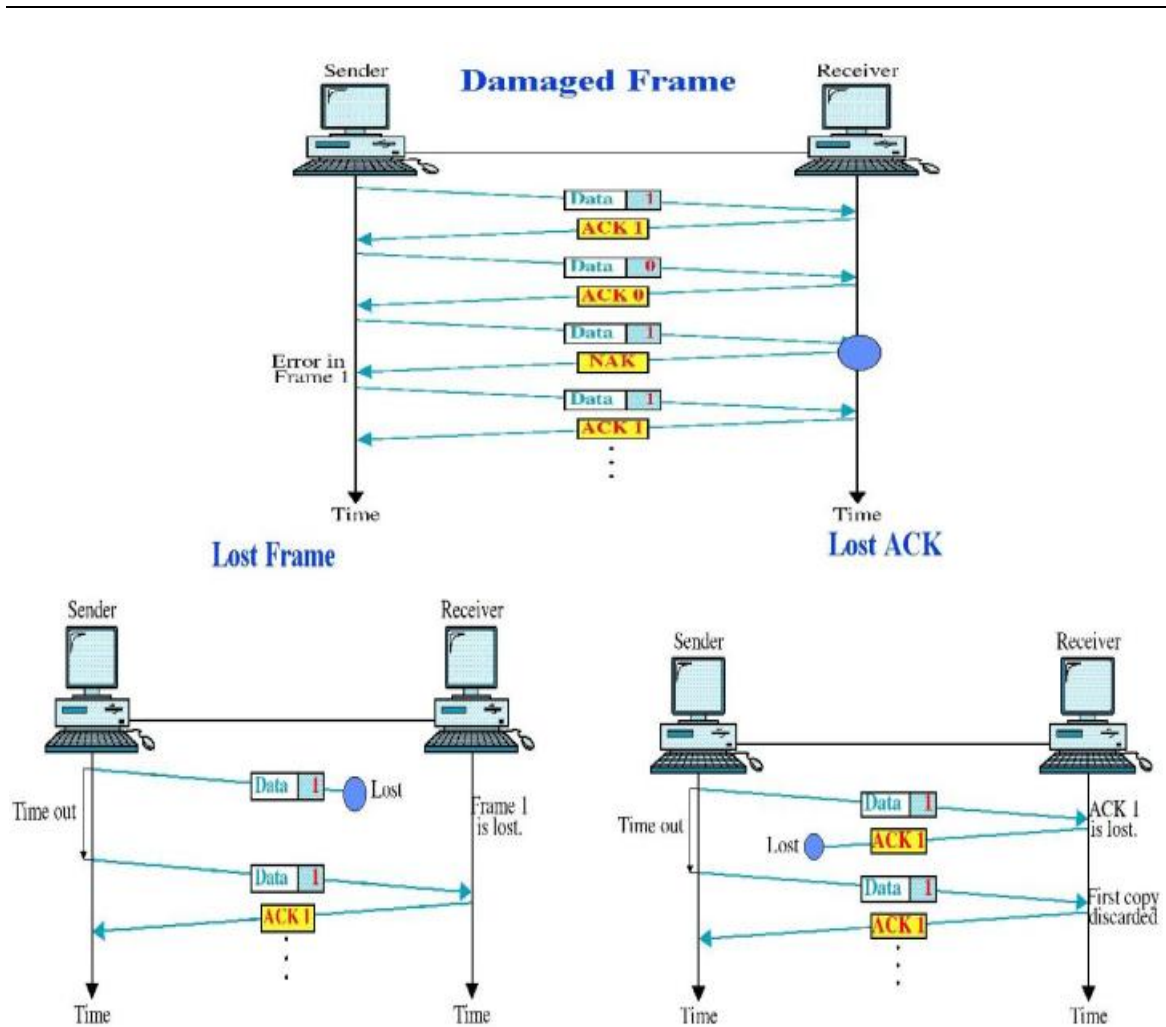
3.5.4 ΕΛΕΓΧΟΣ ΣΦΑΛΜΑΤΩΝ

Υπάρχουν 2 περιπτώσεις λαθών στο επίπεδο σύνδεσης δεδομένων: κατεστραμμένο πλαίσιο, χαμένο πλαίσιο (δεδομένων ή ACK/NACK). Η διαδικασία ελέγχου των σφαλμάτων είναι γνωστή ως ARQ (Automatic Repeat Request) και είναι ενσωματωμένη στα πρωτόκολλα ελέγχου ροής.

3.5.4.1 ARQ σε πρωτόκολλο παύσης – αναμονής

Στην περίπτωση των πρωτοκόλλων παύσης / αναμονής, ο μηχανισμός ARQ έχει ως εξής:

- Κρατιέται από τον αποστολέα το κάθε πλαίσιο που μεταδίδεται, ώσπου να ληφθεί ένα σήμα ACK για το frame αυτό.
- Τα πλαίσια αριθμούνται διαδοχικά με τους αριθμούς '0' και '1'.
- Αν ο δέκτης ανακαλύψει λάθος σε κάποιο πλαίσιο, στέλνει ένα πλαίσιο NACK στον αποστολέα.
- Ο αποστολέας διαθέτει ένα χρονόμετρο, ώστε να θέτει ένα ανώτατο χρονικό όριο ως τη λήψη κάποιου ACK. Μετά το πέρας του ορίου, ξαναστέλνει το τελευταίο πλαίσιο δεδομένων.



Εικόνα 3.9: Έλεγχος σφαλμάτων με χρήση του Stop-and-Wait ARQ για τις περιπτώσεις κατεστραμμένου πακέτου και χαμένου πακέτου (δεδομένων ή επιβεβαίωσης)

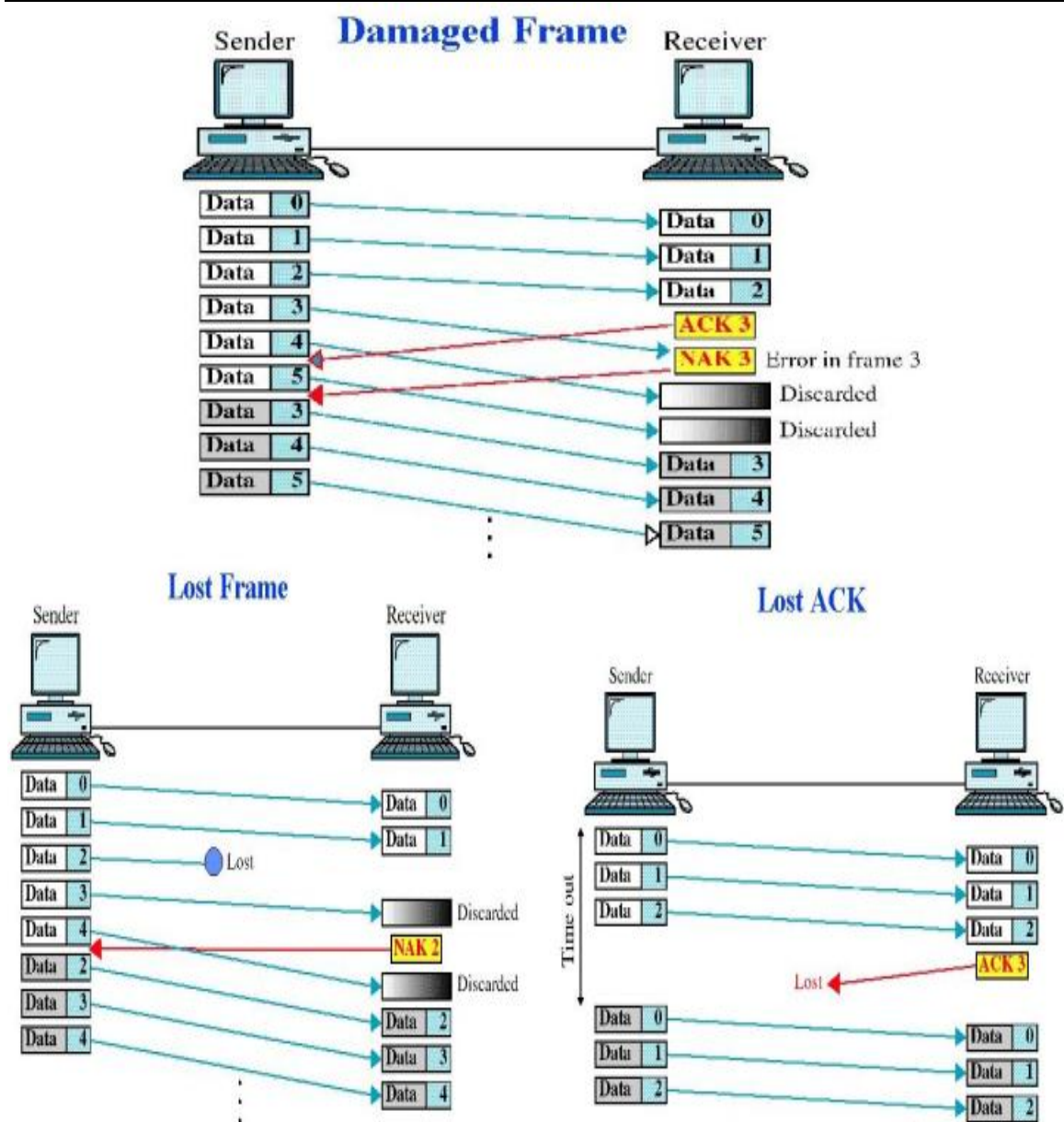
3.5.4.2 ARQ σε πρωτόκολλο συρόμενου παραθύρου

Στην περίπτωση των πρωτοκόλλων συρόμενου παραθύρου, ο μηχανισμός ARQ έχει ως εξής:

- Ο αποστολέας κρατάει αντίγραφα των πλαισίων που έστειλε και δεν έχουν επιβεβαιωθεί από τον παραλήπτη.
- Ο δέκτης έχει τη δυνατότητα αποστολής ενός πλαισίου NACK με τον αύξοντα αριθμό του κατεστραμμένου πλαισίου.
- Ο αποστολέας διαθέτει χρονόμετρο, ώστε να χειρίζεται τις περιπτώσεις των χαμένων πλαισίων (δεδομένων ή ACK/NACK)

Σε περιπτώσεις σφαλμάτων, υπάρχουν 2 διαφορετικοί τρόποι αντιμετώπισης. Αυτοί είναι οι εξής:

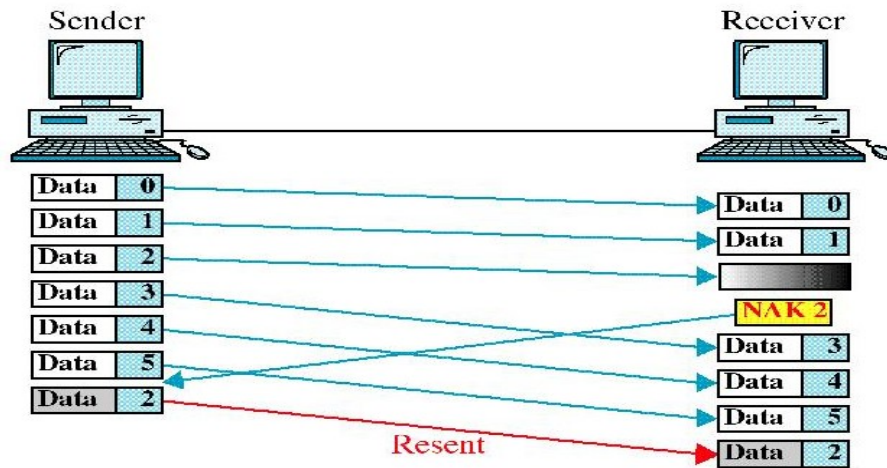
- **Go – Back – n ARQ:** Όταν κάποιο πλαίσιο χαθεί ή καταστραφεί, όλα τα πλαίσια που μεταδόθηκαν μετά από το τελευταίο επιβεβαιωμένο αναμεταδίδονται.



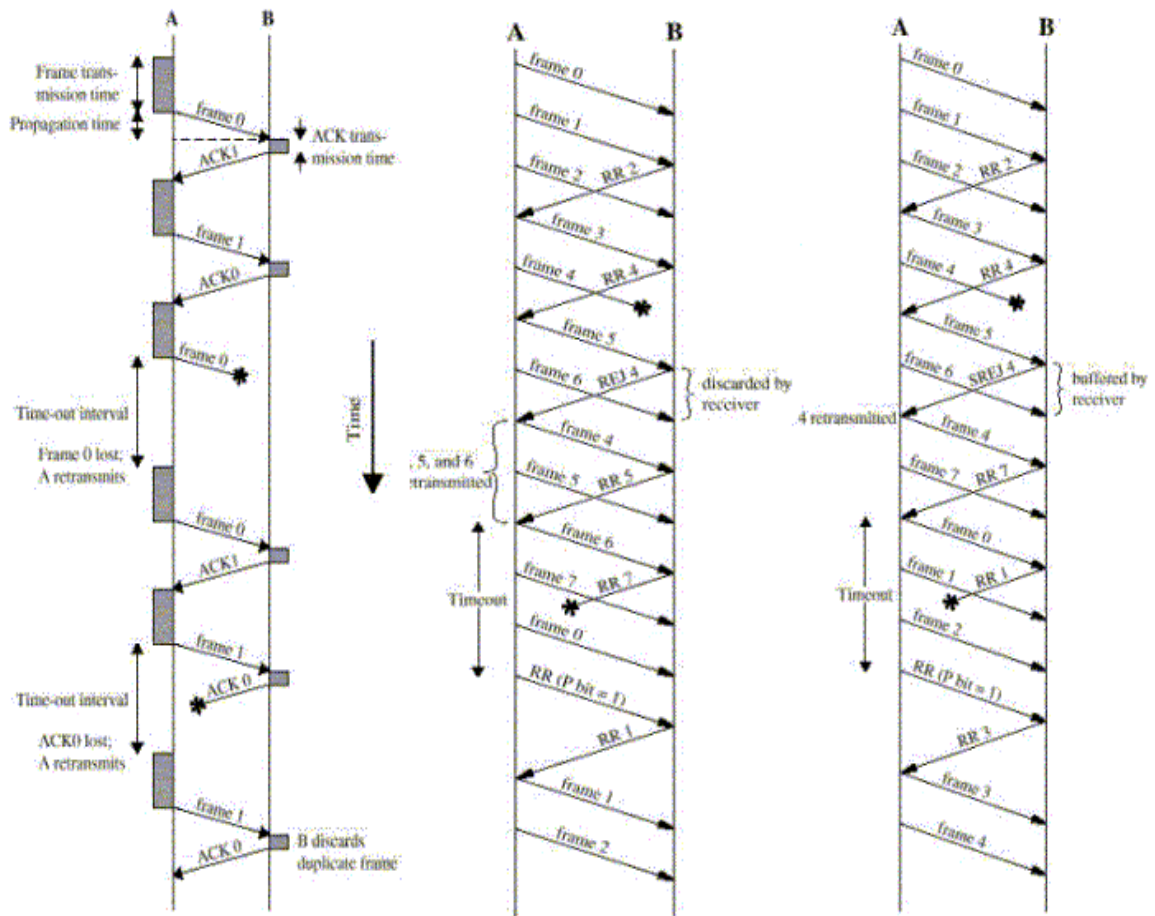
Εικόνα 3.10: Έλεγχος σφαλμάτων με χρήση του Go-Back-n ARQ για τις περιπτώσεις κατεστραμμένου πλαισίου και χαμένου πακέτου (δεδομένων ή επιβεβαιώσης)

- Selective – Reject ARQ:** Αναμεταδίδεται μόνο το συγκεκριμένο χαμένο ή κατεστραμμένο πλαίσιο. ο δέκτης πρέπει να διαθέτει πιο εξελιγμένο λογισμικό, από ό,τι στην περίπτωση Go – Back – n. Συγκεκριμένα, πρέπει να διαθέτει λογισμικό ταξινόμησης των frames που λήφθηκαν εκτός σειράς και να ξεχωρίζει τα πλαίσια πριν και μετά κάποιο NAK. Ο αποστολέας πρέπει να είναι σε θέση να αναγνωρίζει και να βρίσκει τα πλαίσια που του ζητούνται μετά τη λήψη ενός NACK. Επιπλέον, δημιουργείται η ανάγκη για μικρότερα μεγέθη παραθύρου από ό,τι στην περίπτωση Go – Back – n. Η αρίθμηση των πλαισίων ACK δηλώνει το ληφθέν ή χαμένο πλαίσιο και όχι το επόμενο προς αναμονή πλαίσιο, για να αυξηθεί η επιλεκτικότητα.

Selective Reject



Εικόνα 3.11: Παράδειγμα ελέγχου σφαλμάτων με χρήση του Selective – Reject ARQ



Εικόνα 3.12: Σύγκριση των τριών τεχνικών ελέγχου σφαλμάτων (με σειρά: Stop-and-Wait ARQ, Go-Back-N ARQ και Selective-Retry ARQ)

3.6 ΜΕΛΕΤΗ ΕΠΙΔΟΣΕΩΝ

3.6.1 ΠΡΩΤΟΚΟΛΛΟ STOP-AND-WAIT

3.6.1.1 Χωρίς Σφάλματα

Υποθέτουμε ότι T_{ix} (Transmission Time) είναι ο χρόνος μετάδοσης ενός πλαισίου, T_p είναι ο χρόνος διάδοσης του πλαισίου (Propagation Time). Ο χρόνος μετάδοσης εξαρτάται από το μήκος του πλαισίου και από την ταχύτητα μετάδοσης. Ο χρόνος διάδοσης εξαρτάται από την απόσταση μεταξύ του πομπού και του δέκτη και από την ταχύτητα του φωτός (περίπου 300.000 Km/sec).

Πιο συγκεκριμένα, όσον αφορά το χρόνο μετάδοσης, αυτός δίνεται από τη σχέση: $T_{ix} = \frac{N}{R}$, όπου N: ο αριθμός των bit προς μετάδοση και R: ο ρυθμός μετάδοσης των bit από τη γραμμή μετάδοσης (σε bps).

Επιπλέον, η καθυστέρηση διάδοσης του σήματος δίνεται από τη σχέση: $T_p = \frac{S}{V}$, όπου S: η απόσταση (σε μέτρα) ανάμεσα στον πομπό και το δέκτη και V: η ταχύτητα διάδοσης (σε μέτρα/δευτερόλεπτο).

Επιπλέον, T_{ack} είναι ο χρόνος μετάδοσης ενός πλαισίου ACK, T_{nack} είναι ο χρόνος μετάδοσης ενός πλαισίου τύπου NACK και T_{proc} είναι ο χρόνος επεξεργασίας ενός πλαισίου. Για ευκολία θεωρούμε ότι αυτοί οι τρεις αυτοί χρόνοι είναι ίσοι.

Στην περίπτωση όπου δεν υπάρχουν σφάλματα η απόδοση (u) ενός πρωτοκόλλου stop-and-wait έχει ως εξής: $u = \frac{\text{Ωφέλιμος Χρόνος}}{\text{Ολικός Χρόνος}} = \frac{T_{ix}}{T_{ix} + 2T_p + 3T_{proc}}$. Αν

θεωρηθεί αμελητέο το T_{proc} , η απόδοση του συστήματος δίνεται από τη σχέση: $u = \frac{T_{ix}}{T_{ix} + 2T_p}$.

Αν θέσουμε $a = \frac{T_p}{T_{ix}}$, παίρνουμε: $u = \frac{1}{1 + 2a}$. Για περιπτώσεις όπου το a είναι πολύ μικρό, η απόδοση του συστήματος είναι πολύ κοντά στη μονάδα ($u = 1$).

3.6.1.2 Με σφάλματα

Για την περίπτωση όπου το πρωτόκολλο stop-and-wait λαμβάνει αντιμετωπίζει περιπτώσεις σφαλμάτων, ακολουθείται η παρακάτω προσέγγιση.

Υποθέτουμε ότι N_i είναι αριθμός των bit ενός πλαισίου i . Αν συμβολίσουμε με p την πιθανότητα ένα bit να είναι λανθασμένο, τότε έχουμε:

$P_f = P(\text{ένα πλαίσιο να είναι κατεστραμμένο}) = P(\text{τουλάχιστον ένα λανθασμένο bit}) = 1 - P(\text{κανένα λάθος}) = 1 - (1 - p)^{N_i} \approx N_i \cdot p \ll 1$.

Υποθέτουμε ότι N_r είναι ο αριθμός των μεταδόσεων για ένα πακέτο. Ισχύει το εξής:

$$\begin{aligned} N_r &= (1 - P_f) + 2P_f(1 - P_f) + 3P_f^2(1 - P_f) + \dots + kP_f^{k-1}(1 - P_f) = \\ &= (1 - P_f)(1 + 2P_f + 3P_f^2 + \dots + kP_f^{k-1} + \dots) = \frac{1 - P_f}{(1 - P_f)^2} = \frac{1}{1 - P_f}. \end{aligned}$$

Ο τύπος της απόδοσης της προηγούμενης ενότητας τροποποιείται ως εξής:

$$u = \frac{T_{ix}}{N_r T_{ix} + 2N_r T_p} = \frac{1}{N_r(1+2a)}.$$

Λαμβάνοντας υπόψη τις 2 τελευταίες σχέσεις προκύπτει ότι: $u = \frac{1-P_f}{1+2a}$.

3.6.2 ΠΡΩΤΟΚΟΛΛΟ SELECTIVE – REJECT

3.6.2.1 Χωρίς Λάθη

Υποθέτουμε ότι το μέγεθος παραθύρου είναι k . Η απόδοση του πρωτοκόλλου εξαρτάται από το μέγεθος του παραθύρου.

Στην περίπτωση όπου το k είναι μεγαλύτερο από τον αριθμό των πακέτων που πρέπει να περάσουν πριν πάρουμε επιβεβαίωση, έχουμε συνεχή ροή, οπότε $u = 100\%$. Για να ισχύει αυτό θα πρέπει: $kT_{ix} \geq T_{ix} + 2T_p \Rightarrow k \geq 1 + 2\frac{T_p}{T_{ix}} \Rightarrow k \geq 1 + 2a$.

Στην περίπτωση όπου $k \leq 1 + 2a$, η απόδοση του πρωτοκόλλου έχει ως εξής:

$$u = \frac{kT_{ix}}{T_{ix} + 2T_p} = \frac{k}{1+2a}.$$

3.6.2.2 Με Λάθη

Για την περίπτωση όπου $k \leq 1 + 2a$, $u = 1 - P_f$.

Αν $k \geq 1 + 2a$, $u = \frac{kT_{ix}}{N_r(T_{ix} + 2T_p)} = \frac{k(1-P_f)}{1+2a}$. Υπενθυμίζεται ότι $N_r = \frac{1}{1-P_f}$.

3.6.3 ΠΡΩΤΟΚΟΛΛΟ GO-BACK-N

3.6.3.1 Χωρίς Λάθη

Η περίπτωση αυτή είναι ακριβώς η ίδια με την αντίστοιχη στο πρωτόκολλο Selective-Reject.

3.6.3.2 Με Λάθη

$$\text{Αν } k \leq 1 + 2a, u = \frac{k(1-P_f)}{(1+2a)(1+P_f(k-1))}.$$

$$\text{Αν } k \geq 1 + 2a, u = \frac{1+2a(1-P_f)}{(1+2a)(1+P_f(k-1))}.$$

3.7 ΠΡΩΤΟΚΟΛΛΑ ΣΥΝΔΕΣΗΣ ΔΕΔΟΜΕΝΩΝ

Τα πρωτόκολλα του επιπέδου σύνδεσης δεδομένων, που επιφορτίζονται με τα όσα ειπώθηκαν προηγουμένως, χωρίζονται σε 2 κύριες κατηγορίες: τα ασύγχρονα και τα σύγχρονα πρωτόκολλα.

3.7.1 ΑΣΥΓΧΡΟΝΑ ΠΡΩΤΟΚΟΛΛΑ

Πρόκειται για πρωτόκολλα τα οποία πραγματοποιούν ασύγχρονη επικοινωνία (βλέπε κεφάλαιο 1) μεταξύ του πομπού και του δέκτη. Είναι χαμηλού κόστους και όχι ιδιαίτερα πολύπλοκα, αλλά ταυτόχρονα πολύ αργά ώστε να μην έχουν πολλές πρακτικές εφαρμογές σήμερα.

Τα κυριότερα ασύγχρονα πρωτόκολλα είναι: XMODEM, YMODEM, ZMODEM, BLAST (Blocked Asynchronous Transmission) και Kermit.

3.7.2 ΣΥΓΧΡΟΝΑ ΠΡΩΤΟΚΟΛΛΑ

Πρόκειται για πρωτόκολλα τα οποία πραγματοποιούν σύγχρονη επικοινωνία (βλέπε κεφάλαιο 1) μεταξύ του πομπού και του δέκτη. Χωρίζονται σε 2 κατηγορίες: τα πρωτόκολλα μετάδοσης χαρακτήρων (character-oriented) και τα πρωτόκολλα μετάδοσης ψηφίων (bit-oriented).

3.7.2.1 Πρωτόκολλα μετάδοσης χαρακτήρων (Character-oriented protocols)

Τα frames που μεταφέρονται αποτελούνται από μια αλληλουχία χαρακτήρων του ενός byte (= 8 bit). Έτσι, όλες οι πληροφορίες διαβάζονται ως χαρακτήρες, με τη βοήθεια κάποιου συστήματος κωδικοποίησης, π.χ. ASCII (American Standard Code for Information Interchange – βλέπε εικόνα 3.13), EBCDIC (Extended Binary Coded Decimal Interchange Code – βλέπε εικόνα 3.14)...

Bit positions	7	0	0	0	0	1	1	1	1
	6	0	0	1	1	0	0	1	1
	5	0	1	0	1	0	1	0	1
4 3 2 1									
0 0 0 0	NUL	DLE	SP	0	@	P	\	p	
0 0 0 1	SOH	DC1	!	1	A	Q	a	q	
0 0 1 0	STX	DC2	"	2	B	R	b	r	
0 0 1 1	ETX	DC3	#	3	C	S	c	s	
0 1 0 0	EOT	DC4	\$	4	D	T	d	t	
0 1 0 1	ENQ	NAK	%	5	E	U	e	u	
0 1 1 0	ACK	SYN	&	6	F	V	f	v	
0 1 1 1	BEL	ETB	'	7	G	W	g	w	
1 0 0 0	BS	CAN	(8	H	X	h	x	
1 0 0 1	HT	EM)	9	I	Y	i	y	
1 0 1 0	LF	SUB	*	:	J	Z	j	z	
1 0 1 1	VT	ESC	+	:	K	[k	{	
1 1 0 0	FF	FS	.	<	L	\	l		
1 1 0 1	CR	GS	-	=	M]	m		
1 1 1 0	SO	RS	.	>	N	^	n	~	
1 1 1 1	SI	US	:	?	O	_	o	DEL	

Εικόνα 3.13: Ο κώδικας ASCII

Bit positions	4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	
	3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	
	2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	
	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
8	7	6	5																
0	0	0	0	NUL	SOH	STX	ETX	PF	HT	LC	DEL			SMM	VT	FF	CR	SO	SI
0	0	0	1	DLE	DC ₁	DC ₂	DC ₃	RES	NL	BS	IL	CAN	EM	CC		IFS	IGS	IRS	IUS
0	0	1	0	DS	SOS	FS		BYP	LF	EOB	PRE			SM			ENQ	ACK	BEL
0	0	1	1			SYN		PN	RS	UC	EOT					DC ₄	NAK		SUB
0	1	0	0	SP										e	.	<	(+	
0	1	0	1	&										!	\$	*)	;	~
0	1	1	0	-	/									'	%	-	>	?	
0	1	1	1											:	#	@	,	=	"
1	0	0	0		a	b	c	d	e	f	g	h	i						
1	0	0	1		j	k	l	m	n	o	p	q	r						
1	0	1	0			s	t	u	v	w	x	y	z						
1	0	1	1																
1	1	0	0		A	B	C	D	E	F	G	H	I						
1	1	0	1		J	K	L	M	N	O	P	Q	R						
1	1	1	0			S	T	U	V	W	X	Y	Z						
1	1	1	1		0	1	2	3	4	5	6	7	8	9					□

Εικόνα 3.14: Ο κώδικας EBCDIC

Τα πρωτόκολλα αυτά είναι γνωστά και ως byte-oriented protocols. Το αντιπροσωπευτικότερο είναι το πρωτόκολλο BSC (Binary Synchronous Communication), το οποίο αναπτύχθηκε από την IBM το 1964. Αυτό υποστηρίζει half-duplex, point-to-point και multipoint επικοινωνία, ενώ για τον έλεγχο ροής και σφαλμάτων χρησιμοποιεί την τεχνική stop-and-wait ARQ. Τα frames που χρησιμοποιεί, είτε μεταφέρουν πληροφορίες ελέγχου της μετάδοσης (Control Frames), είτε μεταφέρουν τόσο πληροφορίες ελέγχου όσο και δεδομένα (Data Frames).

Το κυριότερο πρόβλημα, που αντιμετωπίζουν τα πρωτόκολλα χαρακτήρων, είναι γνωστό ως πρόβλημα διαφάνειας δεδομένων (Data Transparency). Το γεγονός ότι, οποιαδήποτε ακολουθία ψηφίων μπορεί να είναι δεδομένα, είναι δυνατό να οδηγήσει σε αδυναμία του δέκτη να ξεχωρίσει ανάμεσα σε χαρακτήρες ελέγχου και χαρακτήρες δεδομένων.

3.7.2.2 Πρωτόκολλα μετάδοσης ψηφίων (Bit-oriented protocols)

Αυτά αποτελούν εξέλιξη των πρωτοκόλλων χαρακτήρων. Μπορούν να μεταφέρουν τις πληροφορίες σε μικρότερα πλαίσια και γενικά αποφεύγουν προβλήματα διαφάνειας. Έχουν αναπτυχθεί πολλά τέτοια πρωτόκολλα, όπως το SDLC (Synchronous Data Link Control – IBM, 1975), το HDLC (High-level Data Link Control – ISO, 1979), η σειρά LAP (Link Access Protocols, π.χ. LAPB, LAPD, LAPM, LAPX,...) από την ITU-T κ.α.

3.7.2.2.1 Το πρωτόκολλο HDLC

Το HDLC αποτελεί τη βάση όλων των μεταγενέστερων αυτού πρωτοκόλλων μετάδοσης bit. Υποστηρίζει half-duplex ή full-duplex, point-to-point ή multipoint επικοινωνία. Υπάρχουν τρεις δυνατοί σχηματισμοί (configurations), τους οποίους υποστηρίζει το πρωτόκολλο HDLC.

Ο πρώτος (Unbalanced Configuration) μπορεί να είναι είτε point-to-point είτε multipoint και περιλαμβάνει μια πρωτεύουσα συσκευή (primary device) για τον έλεγχο της γραμμής, και μία ή περισσότερες δευτερεύουσες (secondary devices). Ο δεύτερος (Symmetrical Configuration) περιλαμβάνει 2 σταθμούς που ο καθένας περιλαμβάνει μια primary και μια secondary συσκευή. Η primary του ενός συνδέεται με τη secondary το άλλου, με μια ξεχωριστή γραμμή. Τέλος, ο τρίτος σχηματισμός (Balanced Configuration) αποτελεί μια point-to-point τοπολογία μεταξύ δύο συσκευών, οι οποίες έχουν διπλή ιδιότητα (combined = primary + secondary) με αποτέλεσμα η γραμμή να ελέγχεται και από τις 2 συσκευές.

Η επικοινωνία μεταξύ των συσκευών είναι τριών καταστάσεων (Communication Modes):

- NRM (Normal Response Mode): Σε αυτού του είδους την επικοινωνία, η δευτερεύουσα συσκευή πρέπει να ζητήσει την άδεια της πρωτεύουσας για να μεταδώσει.
- ARM (Asynchronous Response Mode): Η δευτερεύουσα συσκευή μπορεί να μεταδίδει frames χωρίς την άδεια της πρωτεύουσας, όταν το κανάλι είναι άδειο (idle).
- ABM (Asynchronous Balance Mode): Εκφράζει τη σχέση μεταξύ των συσκευών για την περίπτωση του balance configuration, όπου οποιαδήποτε από τις δύο συσκευές μπορεί να ελέγχει τη γραμμή χωρίς την άδεια της άλλης.

Το πρωτόκολλο HDLC χρησιμοποιεί 3 τύπων frames:

- I – Frames (Information Frames): Χρησιμοποιούνται στη μεταφορά των δεδομένων των χρηστών, αλλά μεταφέρουν ταυτόχρονα και πληροφορίες ελέγχου γραμμής, ροής, σφαλμάτων, ...
- S – Frames (Supervisory Frames): Χρησιμοποιούνται μόνο στην αποστολή πληροφοριών ελέγχου της γραμμής επικοινωνίας (έλεγχος ροής και σφαλμάτων).
- U – Frames (Unnumbered Frames): Μεταφέρουν πληροφορίες για την διαχείριση και εποπτεία του συστήματος.