

Δίκτυα Υπολογιστών I

Αξιόπιστη επικοινωνία μέσα από ένα σύνδεσμο



Ευάγγελος Παπαπέτρου

Τμ. Μηχ. Η/Υ & Πληροφορικής, Παν. Ιωαννίνων

Διάρθρωση

- 1 Λογική σύνδεση
- 2 Πλαισίωση
- 3 Ανίχνευση, διόρθωση και έλεγχος σφαλμάτων
- 4 Πρωτόκολλα ARQ
 - Alternating Bit Protocol
 - ARQ με κυλιόμενο παράθυρο
 - Go-Back-N
 - Selective Repeat
- 5 Έλεγχος ροής
- 6 Παραδείγματα Τεχνολογιών



Διάρθρωση

- 1 Λογική σύνδεση
- 2 Πλαισίωση
- 3 Ανίχνευση, διόρθωση και έλεγχος σφαλμάτων
- 4 Πρωτόκολλα ARQ
 - Alternating Bit Protocol
 - ARQ με κυλιόμενο παράθυρο
 - Go-Back-N
 - Selective Repeat
- 5 Έλεγχος ροής
- 6 Παραδείγματα Τεχνολογιών



Αναγκαιότητα και ορισμός λογικής σύνδεσης

Η μετάδοση δεδομένων μέσα από ένα σύνδεσμο δεν εξασφαλίζει την **αξιόπιστη ανταλλαγή δεδομένων** μεταξύ δύο κόμβων του δικτύου

- ▶ π.χ. δεδομένα μπορεί να καταστραφούν στο κανάλι

Λογική σύνδεση

Η αξιόπιστη ροή δεδομένων μεταξύ δύο κόμβων που συνδέονται με ένα σύνδεσμο

Η υλοποίηση μιας λογικής σύνδεσης απαιτεί την επίλυση των παρακάτω ζητημάτων:

- ▶ καθορισμός των κόμβων που επικοινωνούν (όταν πολλοί κόμβοι χρησιμοποιούν τον ίδιο σύνδεσμο)
- ▶ τμηματοποίηση της πληροφορίας (για δίκτυα μεταγωγής πακέτου)
- ▶ αξιόπιστη μετάδοση των δεδομένων



Μηχανισμοί Λογικής σύνδεσης

Μια λογική σύνδεση αποτελείται από τους παρακάτω μηχανισμούς:

- ▶ πλαισίωση (framing)
- ▶ διόρθωση σφαλμάτων (error correction)
- ▶ έλεγχο σφαλμάτων (error control)
- ▶ έλεγχο ροής (flow control)

Οι μηχανισμοί μιας λογικής σύνδεσης κατατάσσονται στο επίπεδο συνδέσμου μετάδοσης δεδομένων (data link layer) σύμφωνα με το μοντέλο OSI

- ▶ συνήθως αναφέρουμε ότι οι μηχανισμοί της λογικής σύνδεσης αποτελούν ένα υποεπίπεδο που ονομάζεται **υποεπίπεδο ελέγχου λογικής σύνδεσης** (Logical Link Control, LLC)



Διάρθρωση

1. Λογική σύνδεση
2. Πλαισίωση
3. Ανίχνευση, διόρθωση και έλεγχος σφαλμάτων
4. Πρωτόκολλα ARQ
 - Alternating Bit Protocol
 - ARQ με κυλιόμενο παράθυρο
 - Go-Back-N
 - Selective Repeat
5. Έλεγχος ροής
6. Παραδείγματα Τεχνολογιών



Δομή πλαισίου

Στα δίκτυα μεταγωγής πακέτου η πληροφορία μεταδίδεται κατά τμήματα

- ▶ στο επίπεδο συνδέσμου μετάδοσης δεδομένων τα τμήματα αυτά ονομάζονται **πλαίσια (frames)**

Ένα πλαίσιο αποτελείται από:

- ▶ την **κεφαλίδα**: περιέχει δεδομένα απαραίτητα για την υλοποίηση των μηχανισμών της λογικής σύνδεσης αλλά και του επιπέδου συνδέσμου μετάδοσης δεδομένων, π.χ. διευθύνσεις κόμβων που επικοινωνούν
- ▶ το **ωφέλιμο φορτίο**: περιέχει το πακέτο που προέρχεται από το επίπεδο δικτύου
- ▶ το **επίμετρο**: περιέχει δεδομένα για την αξιόπιστη μετάδοση του πλαισίου



Καθορισμός ορίων πλαισίου

Ένα σημαντικό ζήτημα στην πλαισίωση είναι ο καθορισμός των ορίων (αρχής και τέλους) ενός πλαισίου δεδομένων

Οι γνωστότερες τεχνικές για την επίλυση του προβλήματος είναι:

- ▶ μετρητής χαρακτήρων
- ▶ δείκτες συνόρων με συμπλήρωση byte
- ▶ δείκτες συνόρων με συμπλήρωση bit
- ▶ χρονισμός με χαρακτήρες που δε χρησιμοποιούνται από την κωδικοποίηση φυσικού επιπέδου



Τεχνικές οριοθέτησης πλαισίου (1/2)

Μετρητής χαρακτήρων

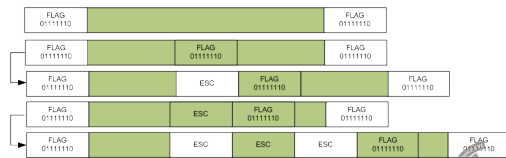
- ▶ κάθε πλαίσιο περιλαμβάνει στην κεφαλίδα ένα πεδίο (μεγέθους ενός ή περισσότερων bytes) που περιέχει το μήκος του πλαισίου (σε bytes)
- ▶ δημιουργείται πρόβλημα αν ο μετρητής χαρακτήρων καταστραφεί από κάποιο σφάλμα κατά τη μετάδοση

Δείκτες συνόρων με συμπλήρωση bit

- ▶ η αρχή και η λήξη του πλαισίου σηματοδοτείται με έναν ειδικό χαρακτήρα μήκους ενός byte

ο χαρακτήρας συνήθως ονομάζεται FLAG (01111110)

- ▶ αν ο χαρακτήρας FLAG εμφανίζεται στα δεδομένα τότε πριν από αυτόν τοποθετείται ένας ειδικός χαρακτήρας (ESC) (**byte stuffing**)



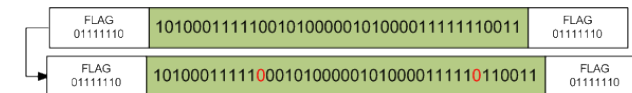
- ▶ το ίδιο συμβαίνει και αν ο χαρακτήρας ESC εμφανίζεται στα δεδομένα

Τεχνικές οριοθέτησης πλαισίου (2/2)

Δείκτες συνόρων με συμπλήρωση bit

- ▶ η αρχή και η λήξη του πλαισίου σηματοδοτείται με τη χρήση του χαρακτήρα FLAG (01111110)
- ▶ η εμφάνιση πέντε συνεχόμενων "1" στα δεδομένα οδηγεί στην συμπλήρωση ενός bit "0" (**bit stuffing**)

στόχος είναι να μην γίνεται σύγχυση των δεδομένων με το χαρακτήρα FLAG



Χρονισμός με χαρακτήρες που δε χρησιμοποιούνται από την κωδικοποίηση φυσικού επιπέδου

- ▶ η αρχή και η λήξη του πλαισίου σηματοδοτείται με χαρακτήρες που δε χρησιμοποιούνται για την αναπαράσταση δεδομένων

Διάρθρωση

1. Λογική σύνδεση
2. Πλαισίωση
3. Ανίχνευση, διόρθωση και έλεγχος σφαλμάτων
4. Πρωτόκολλα ARQ
 - Alternating Bit Protocol
 - ARQ με κυλιόμενο παράθυρο
 - Go-Back-N
 - Selective Repeat
5. Έλεγχος ροής
6. Παραδείγματα Τεχνολογιών

Ανίχνευση και διόρθωση σφαλμάτων: κεντρική ιδέα (1/2)

Ο θόρυβος μπορεί να προκαλέσει **αλλοίωση ενός ή περισσότερων bits** κατά τη μετάδοση δεδομένων μέσα από ένα σύνδεσμο

Αντιμετώπιση προβλήματος: **πλεονάζουσα (redundant)** πληροφορία

- ▶ ο παραλήπτης μπορεί να χρησιμοποιήσει την πλεονάζουσα πληροφορία για να **ανιχνεύσει** ή να **διορθώσει** τα σφάλματα

Συνηθέστερη προσέγγιση για τη χρήση πλεονάζουσας πληροφορίας στα δίκτυα υπολογιστών

- ▶ η πλεονάζουσα πληροφορία κατασκευάζεται με βάση τα δεδομένα και προστίθεται στο τέλος ενός πλαισίου (**επίμετρο**)

Κώδικας ανίχνευσης/διόρθωσης σφαλμάτων

Στα δίκτυα υπολογιστών **κώδικας ανίχνευσης/διόρθωσης σφαλμάτων** ονομάζεται ο αλγόριθμος σύμφωνα με τον οποίο κατασκευάζεται το επίμετρο

Ανίχνευση και διόρθωση σφαλμάτων: κεντρική ιδέα (2/2)

Οι κώδικες διακρίνονται σε δύο κατηγορίες ανάλογα με την ικανότητά τους να **ανιχνεύουν** ή να **διορθώνουν** σφάλματα:

- ▶ κώδικες ανίχνευσης σφαλμάτων (Error Detecting Codes)
- ▶ κώδικες διόρθωσης σφαλμάτων (Error Correcting Codes, ECC)
- ▶ οι κώδικες διόρθωσης σφαλμάτων έχουν μεγαλύτερο μήκος από αυτούς που ανιχνεύουν σφάλματα

Η αξιόπιστη μετάδοση των δεδομένων επιτυγχάνεται με μια από τις παρακάτω προσεγγίσεις:

- ▶ ευθεία διόρθωση σφαλμάτων (Forward Error Correction, FEC) με την χρήση ενός κώδικα διόρθωσης σφαλμάτων
- ▶ ανίχνευση των σφαλμάτων (με την χρήση ενός κώδικα ανίχνευσης σφαλμάτων) και επανεκπομπή του πλαισίου (έλεγχος σφαλμάτων)



Διόρθωση vs Ανίχνευση (1/2)

Η ευθεία διόρθωση σφαλμάτων **δεν είναι πάντα η καλύτερη λύση**

- ▶ το **μέγεθος** της απαιτούμενης πλεονάζουσας πληροφορίας είναι συνήθως **μεγάλο**
- ▶ η πλεονάζουσα πληροφορία μεταδίδεται ακόμα και αν **δεν υπάρχουν σφάλματα**

Η ευθεία διόρθωση σφαλμάτων είναι προτιμότερη όταν:

- ▶ τα σφάλματα είναι πολύ συχνά (π.χ. ασύρματη μετάδοση)
- ▶ το κόστος της επανεκπομπής ενός πλαισίου είναι μεγάλο (π.χ. μεγάλη καθυστέρηση διάδοσης)

Έστω ότι σε ένα σύνδεσμο:

☞ χρησιμοποιούμε ευθεία διόρθωση σφαλμάτων με n bits πλεονάζουσας πληροφορίας για κάθε πλαίσιο με συνολικό μέγεθος l bits

☞ η διεκπεραιωτική ικανότητα που επιτυγχάνουμε στο κανάλι είναι R

Η ωφέλιμη για το χρήστη διεκπεραιωτική ικανότητα είναι $R' = \frac{l-n}{l} R$

Διόρθωση vs Ανίχνευση (2/2)

Η ευθεία διόρθωση σφαλμάτων:

- ▶ **αποτυγχάνει** όταν το κανάλι **καταστρέφει πλήρως τα πλαίσια** (erasure channel)
- ▶ **αποτυγχάνει** όταν το πλήθος των σφαλμάτων σε ένα πλαίσιο είναι τέτοιο ώστε ακόμα και ο κώδικας ευθείας διόρθωσης **αποτυγχάνει**
- ▶ **δεν μπορεί να αντιμετωπίσει** απώλειες πλαισίων εξαιτίας του ότι ο παραλήπτης **δεν μπορεί να παραλάβει τα πλαίσια με το ρυθμό που του αποστέλλονται**

Στις περισσότερες περιπτώσεις δικτύων υπολογιστών χρησιμοποιείται η ανίχνευση σφαλμάτων και η επανεκπομπή των πλαισίων

- ▶ Τα γνωστότερα πρωτόκολλα που υλοποιούν αυτή την προσέγγιση ονομάζονται πρωτόκολλα **αυτόματης αίτησης επανάληψης** (Automatic Repeat reQuest, ARQ)



Διάρθρωση

Λογική σύνδεση

1 Πλαισίωση

2 Ανίχνευση, διόρθωση και έλεγχος σφαλμάτων

4 Πρωτόκολλα ARQ

Alternating Bit Protocol

ARQ με κυλιόμενο παράθυρο

Go-Back-N

Selective Repeat

Έλεγχος ροής

Παραδείγματα Τεχνολογιών



Εισαγωγή (1/2)

Κεντρική ιδέα: **ανίχνευση της απώλειας** ενός πλαισίου και **επανεκπομπή** του από τον αποστολέα

Βασικοί μηχανισμοί για την ανίχνευση της απώλειας ενός πλαισίου:

- ▶ επιβεβαίωση της λήψης από τον παραλήπτη με την αποστολή ενός **πλαίσιου επιβεβαίωσης (ACK)**
- ▶ αναμονή για τη λήψη της επιβεβαίωσης από τον αποστολέα (**χρόνος αναμονής (timeout)**)
 - αν δεν ληφθεί επιβεβαίωση τότε η εκπομπή επαναλαμβάνεται
- ▶ αρίθμηση των πλαισίων με **ακολουθιακούς αριθμούς (sequence numbers)**

⇒ Υπό συνθήκες, είναι εφικτό ο αποστολέας να **ειδοποιείται για την αλλοίωση και όχι για την ορθή λήψη** ενός πλαισίου (**αρνητικές επιβεβαιώσεις (Negative ACKs, NACKs)**)

⇒ Ένα κόμβος μπορεί να "φορτώσει" την επιβεβαίωση σε ένα πλαίσιο δεδομένων που πρόκειται να αποστείλει (**piggyback ACK**)

Εισαγωγή (2/2)

Τα πρωτόκολλα ARQ διαφέρουν ως προς την **πολυπλοκότητα** και την **αξιοποίηση του συνδέσμου** που επιτυγχάνουν

- ▶ η πολυπλοκότητα εξαρτάται από τον **αποθηκευτικό χώρο** που απαιτείται αλλά και το **πλήθος των ακολουθιακών αριθμών**

Η επιλογή του κατάλληλου πρωτοκόλλου ARQ εξαρτάται από τα χαρακτηριστικά του συνδέσμου

- ▶ πρωτίστως από το **γινόμενο καθυστέρησης-εύρους ζώνης**
- ▶ το **ρυθμό σφαλμάτων**

Τα σημαντικότερα πρωτόκολλα της κατηγορίας ARQ είναι τα:

- ▶ Alternating Bit Protocol (ABP)
- ▶ Go-Back-N (GBN)
- ▶ Selective Repeat (SR)



Ανίχνευση σφαλμάτων:Κυκλικός Έλεγχος Πλεονασμού (1/2)

Ο **κυκλικός έλεγχος πλεονασμού (Cyclic Redundancy Check, CRC)** χρησιμοποιείται για την ανίχνευση σφαλμάτων

- ▶ λίγα **πλεονάζοντα bits** αρκούν για τον έλεγχο σφαλμάτων σε χιλιάδες byte πληροφορίας
- ▶ στηρίζεται στη θεωρία των **πεπερασμένων πεδίων (finite fields)**

Σε κάθε πλαίσιο πληροφορίας μήκους k bits προστίθενται n πλεονάζοντα bits

- ▶ τα n bits ονομάζονται **άθροισμα ελέγχου (checksum)**
- ▶ στο δίκτυο μεταδίδεται το συνολικό πλαίσιο των $k + n$ bits

Το **άθροισμα ελέγχου** υπολογίζεται με τη χρήση μιας ακολουθίας από bits που ονομάζεται **ακολουθία ελέγχου, C**

- ▶ η ακολουθία ελέγχου θα πρέπει να διαιρεί χωρίς υπόλοιπο τα $k + n$ bits που προκύπτουν από τα δεδομένα και το άθροισμα ελέγχου
- ▶ το μήκος της ακολουθίας C είναι $n + 1$ bits



Ανίχνευση σφαλμάτων:Κυκλικός Έλεγχος Πλεονασμού (2/2)

Ανίχνευση σφάλματος: η διαίρεση του λαμβανόμενου πλαισίου ($k + n$ bits) με την ακολουθία ελέγχου δίνει **μη μηδενικό υπόλοιπο**

- ▶ ο παραλήπτης πρέπει να γνωρίζει την ακολουθία ελέγχου
- ▶ μηδενικό υπόλοιπο σημαίνει **μεγάλη πιθανότητα** απουσία σφάλματος

Η επιλογή του **μήκους** αλλά και της **ίδιας της ακολουθίας ελέγχου** είναι σημαντική για τη δυνατότητα ανίχνευσης σφαλμάτων (πλήθος και τύπος σφαλμάτων)

- ▶ μόνο λίγες ακολουθίες είναι κατάλληλες για ανίχνευση σφαλμάτων
- ▶ η επιλογή κατάλληλης ακολουθίας είναι συμβιβασμός μεταξύ των **δυνατοτήτων ανίχνευσης σφαλμάτων** και του **μήκους της ακολουθίας**
- ▶ σημαντικότερες ακολουθίες:

Τύπος CRC	ακ. ελέγχου	Τύπος CRC	ακ. ελέγχου
CRC-8-ITU	100000111	CRC-16-IBM	1100000000000101
CRC-10	11000110011	CRC-CCITT	10001000000100001
CRC-12	1100000001111	CRC-32	10000010011000001000111011010111



Διάρθρωση

- 1. Λογική σύνδεση
- 2. Πλαισίωση
- 3. Ανίχνευση, διόρθωση και έλεγχος σφαλμάτων
- 4. Πρωτόκολλα ARQ
 - Alternating Bit Protocol
 - ARQ με κυλιόμενο παράθυρο
 - Go-Back-N
 - Selective Repeat
- Έλεγχος ροής
- Παραδείγματα Τεχνολογιών



Alternating Bit Protocol: Εισαγωγή

Το πρωτόκολλο **Alternating Bit Protocol (ABP)** είναι το απλούστερο πρωτόκολλο για τον έλεγχο σφαλμάτων

- ▶ ονομάζεται επίσης και **Stop-and-Wait ARQ**

Κεντρική ιδέα: αποστολή ενός πλαισίου κάθε φορά και αναμονή για την επιβεβαίωσή του από τον παραλήπτη

Είναι ένα πρωτόκολλο με **χαμηλή πολυπλοκότητα** αλλά και **χαμηλή αξιοποίηση** του συνδέσμου

- ▶ ο αποστολέας πρέπει να έχει τη δυνατότητα αποθήκευσης ενός πλαισίου
- ▶ χρησιμοποιούνται μόνο **δύο αριθμοί ακολουθίας** (οι 0 και 1)

⇒ Χρειάζεται μόνο ένα bit στην κεφαλίδα του πλαισίου, το οποίο **εναλλάσσεται** μεταξύ των τιμών 0 και 1, για την αναπαράσταση των αριθμών ακολουθίας



Λειτουργία ABP (1/2)

Αποστολέας:

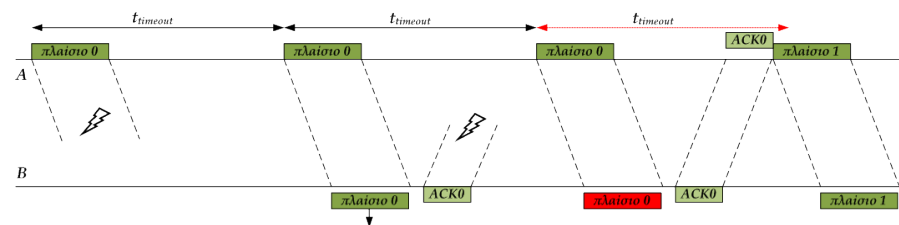
- ▶ εκπέμπει ένα πλαίσιο τη φορά και κρατά αντίγραφο του πλαισίου μέχρι αυτό να επιβεβαιωθεί από τον παραλήπτη
- ▶ για την αναμονή της επιβεβαίωσης χρησιμοποιείται χρονόμετρο που λήγει μετά από κάποιο προκαθορισμένο χρόνο ($t_{timeout}$)
 - ο χρόνος $t_{timeout}$ πρέπει να είναι τέτοιος ώστε να επιτρέπει τη λήψη μιας επιβεβαίωσης και από τον πλέον απομακρυσμένο κόμβο
- ▶ τα πλαίσια αριθμούνται εναλλάξ με 0 ή 1
- ▶ αν δεν ληφθεί επιβεβαίωση το πλαίσιο επανεκπέμπεται, διαφορετικά εκπέμπεται το επόμενο πλαίσιο

Παραλήπτης:

- ▶ με την ορθή λήψη του πλαισίου:
 - παραδίδει το πλαίσιο στο ανώτερο επίπεδο
 - επιβεβαιώνει την ορθή λήψη ενός πλαισίου με ένα ACK
- ▶ το πλαίσιο ACK αριθμείται όπως και το πλαίσιο που επιβεβαιώνει



Λειτουργία ABP (2/2)



Οι αριθμοί ακολουθίας χρησιμεύουν στην **αναγνώριση και απόρριψη διπλοτύπων** από τον παραλήπτη

Δημιουργία διπλοτύπων

⇒ Διπλότυπα δημιουργούνται όταν ο αποστολέας επαναλαμβάνει την αποστολή ενός πλαισίου επειδή δεν γνωρίζει ότι το πλαίσιο παραδόθηκε σωστά



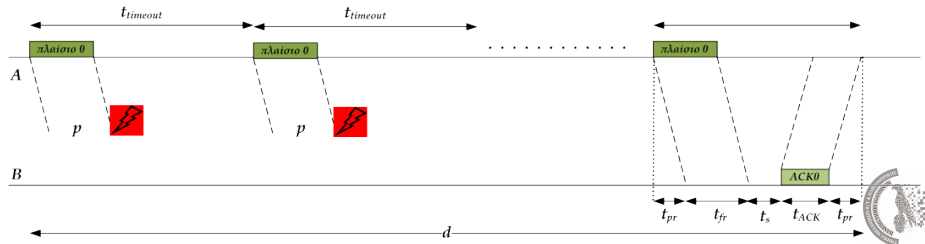
Απόδοση ABP (1/3)

Η μετάδοση ενός πλαισίου στο ABP αποτελείται από k συνολικά προσπάθειες ($k - 1$ αποτυχίες και 1 επιτυχία)

- ▶ σε κάθε αποτυχία δαπανάται χρόνος $t_{timeout}$
- ▶ στην επιτυχία δαπανάται χρόνος $2t_{pr} + t_{fr} + t_s + t_{ack}$

$$d = (k - 1)t_{timeout} + 2t_{pr} + t_{fr} + t_s + t_{ack}$$

$$E\{d\} = (E\{k\} - 1)t_{timeout} + 2t_{pr} + t_{fr} + t_s + t_{ack}$$



Απόδοση ABP (2/3)

Το μέσο πλήθος προσπαθειών (και αποτυχιών) εξαρτάται από την πιθανότητα αποτυχίας μιας προσπάθειας (p)

$$E\{k\} = \frac{1}{1-p}$$

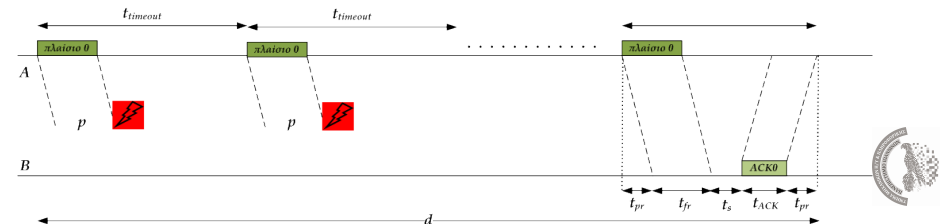
$$E\{k - 1\} = \frac{p}{1-p}$$

Η αξιοποίηση του συνδέσμου μετράται με τη ρυθμαπόδοση και την απόδοση

- ▶ C : εύρος ζώνης του συνδέσμου, l_{fr} : μέγεθος του πλαισίου

$$R = \frac{l_{fr}}{(\frac{p}{1-p})t_{timeout} + 2t_{pr} + t_{fr} + t_s + t_{ack}}$$

$$n = \frac{t_{fr}}{E\{d\}} = \frac{R}{C}$$



Απόδοση ABP (3/3)

Στην ειδική περίπτωση που $t_{timeout} = 2t_{pr} + t_{fr} + t_s + t_{ack}$

$$R = \frac{(1-p)l_{fr}}{2t_{pr} + t_{fr} + t_s + t_{ack}}$$

Η ρυθμαπόδοση του ABP είναι χαμηλή σε συνδέσμους με **υψηλό ρυθμό σφαλμάτων**

$$R \xrightarrow{p \rightarrow 1} 0$$

Η ρυθμαπόδοση του ABP είναι χαμηλή σε συνδέσμους με **μεγάλο γινόμενο καθυστέρησης-εύρους ζώνης** ακόμα και όταν ο ρυθμός σφαλμάτων είναι μικρός

$$R \xrightarrow{C \rightarrow \infty, p \rightarrow 0} \frac{l_{fr}}{2t_{pr} + t_s + t_{ack}}$$

$$R \xrightarrow{t_{pr} \gg t_{fr}, p \rightarrow 0} C \frac{(1-p) \frac{t_{fr}}{t_{pr}}}{2 + \frac{t_{fr} + t_s + t_{ack}}{t_{pr}}} \xrightarrow{\frac{t_{fr}}{t_{pr}} \ll 1} 0$$

Διάρθρωση

- 1 Λογική σύνδεση
- 2 Πλαισίωση
- 3 Ανίχνευση, διόρθωση και έλεγχος σφαλμάτων
- 4 Πρωτόκολλα ARQ
 - Alternating Bit Protocol
 - ARQ με κυλιόμενο παράθυρο
 - Go-Back-N
 - Selective Repeat
- 5 Έλεγχος ροής
- 6 Παραδείγματα Τεχνολογιών

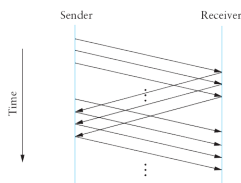
Σκοπιμότητα και κεντρική ιδέα

Η μικρή ρυθμαπόδοση του ABP οφείλεται στο ότι υπάρχει **μόνο ένα πλαίσιο υπό μετάδοση** κάθε φορά

- ▶ μπορεί να μεταδοθεί κατά μέγιστο ένα πλαίσιο κάθε $\sim 2t_{pr} \text{ sec}$

Κεντρική ιδέα: **περισσότερα από ένα πλαίσια υπό μετάδοση ταυτόχρονα στο σύνδεσμο**

- ▶ ο αποστολέας επιτρέπεται να μεταδώσει **παραπάνω από ένα πλαίσια πριν λάβει κάποια επιβεβαίωση**



Τα ARQ πρωτόκολλα αυτής της κατηγορίας ονομάζονται **κλιόμενου παραθύρου (Sliding Window ARQ)**



Βασικοί μηχανισμοί (1/4)

Αποστολέας και παραλήπτης υλοποιούν τους μηχανισμούς ARQ:

- ▶ ο αποστολέας αριθμεί τα πλαίσια δεδομένων με **διαφορετικούς ακολουθιακούς αριθμούς**
- ▶ ο παραλήπτης επιβεβαιώνει τη λήψη ενός πλαισίου με μια επιβεβαίωση που έχει τον ίδιο ακολουθιακό αριθμό
- ▶ για την αναμονή των επιβεβαιώσεων, ο αποστολέας χρησιμοποιεί **ένα χρονόμετρο για κάθε πλαίσιο δεδομένων**

Αποστολέας: μπορεί να εκπέμπει μέχρι ***SWS*** πλαίσια χωρίς να έχει λάβει κάποια επιβεβαίωση

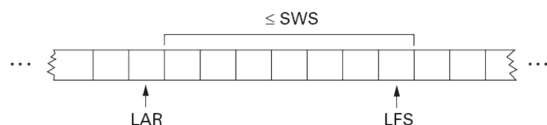
- ▶ ο αριθμός *SWS* ονομάζεται (**μέγεθος παραθύρου αποστολέα, Sender Window Size**)
- ▶ μέχρι *SWS* πλαίσια βρίσκονται υπό μετάδοση στο σύνδεσμο
- ▶ ο αποστολέας πρέπει να έχει τη δυνατότητα να **αποθηκεύει μέχρι *SWS* πλαίσια** μέχρι τη λήψη της επιβεβαίωσης



Βασικοί μηχανισμοί (2/4)

Αποστολέας: διατηρεί δύο μετρητές

- ▶ **LAR, Last Acknowledgement Received:** περιέχει τον αριθμό ακολουθίας της τελευταίας επιβεβαίωσης που ελήφθη
 ενημερώνεται με βάση τις επιβεβαιώσεις που λαμβάνονται
- ▶ **LFS, Last Frame Sent:** περιέχει τον αριθμό ακολουθίας του τελευταίου πλαισίου δεδομένων που στάλθηκε



Ο αποστολέας μπορεί να μεταδώσει ένα πλαίσιο με αριθμό ακολουθίας k (δηλαδή $LFS \leftarrow k$) μόνο αν ισχύει:

$$LFS - LAR \leq SWS$$



Βασικοί μηχανισμοί (3/4)

Ο παραλήπτης ενδέχεται να λάβει πλαίσια δεδομένων **εκτός σειράς (out of order)**

- ▶ το φαινόμενο εμφανίζεται όταν κάποιο πακέτο που μεταδόθηκε πρώτο καταστρέφεται, ενώ ένα επόμενο παραδίδεται κανονικά

Παραλήπτης: μπορεί να παραλάβει και να επιβεβαιώσει μέχρι ***RWS*** πλαίσια

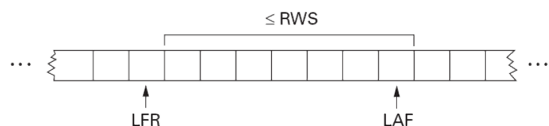
- ▶ ο αριθμός *RWS* ονομάζεται (**μέγεθος παραθύρου παραλήπτη, Receiver Window Size**)
- ▶ ο παραλήπτης πρέπει να έχει τη δυνατότητα να **αποθηκεύει μέχρι $RWS - 1$ πλαίσια** πριν τα παραδώσει στο ανώτερο επίπεδο
 μέχρι *RWS* πλαίσια μπορούν να φτάσουν στον παραλήπτη εκτός σειράς



Βασικοί μηχανισμοί (4/4)

Παραλήπτης: διατηρεί επίσης δύο μετρητές

- ▶ **LFR, Last Frame Received:** περιέχει τον αριθμό ακολουθίας του τελευταίου πλαισίου δεδομένων που ελήφθη
- ▶ **LAF, Largest Acceptable Frame:** ο μέγιστος αριθμός ακολουθίας ενός πλαισίου ώστε να γίνει αποδεκτό



Ένα πλαίσιο δεδομένων με ακολουθιακό αριθμό k :

- ▶ απορρίπτεται αν $k \leq LFR$ ή $k > LAF$
- ▶ γίνεται αποδεκτό (δηλαδή $LFR \leftarrow k$) και επιβεβαιώνεται αν $LFR < k \leq LAF$



Αριθμοί ακολουθίας

Το πλήθος των ακολουθιακών αριθμών που χρησιμοποιούνται (N) δεν μπορεί να είναι ανεξέλεγκτα μεγάλο

- ▶ χρειάζονται $\log_2 N$ bits για την αναπαράστασή τους, τα οποία ενσωματώνονται στο πλαίσιο (δεδομένων ή επιβεβαίωσης)

Κεντρική ιδέα: χρειάζονται μόνο **τόσοι ακολουθιακοί αριθμοί όσα τα πλαίσια σε εκκρεμότητα**

- ▶ είναι εφικτή η διάκριση των πλαισίων με βάση τον ακολουθιακό αριθμό
- ▶ επόμενα πλαίσια μεταδίδονται με επαναχρησιμοποίηση των ακολουθιακών αριθμών (sequence number rollover)

Πλήθος ακολουθιακών αριθμών

⇒ Ένα πρωτόκολλο ARQ χρειάζεται συνολικά $SWS + RWS$ διαφορετικούς ακολουθιακούς αριθμούς

- ❑ SWS πλαίσια υπό μετάδοση
- ❑ RWS πλαίσια εκτός σειράς που είναι αποθηκευμένα στον παραλήπτη και η παράδοσή τους στο ανώτερο επίπεδο εκκρεμεί

Μέγεθος παραθύρου

Η επιλογή του SWS σχετίζεται με την **αξιοποίηση του συνδέσμου**

- ▶ η βέλτιστη επιλογή προσδιορίζεται από το **γινόμενο καθυστέρησης-εύρους ζώνης**

Η επιλογή των SWS και RWS επηρεάζει την πολυπλοκότητα του μηχανισμού ARQ

- ▶ μέγεθος χώρου αποθήκευσης στον αποστολέα: SWS
- ▶ μέγεθος χώρου αποθήκευσης στον παραλήπτη: $RWS - 1$
- ▶ αναπαράσταση αριθμών ακολουθίας με $\log_2(SWS + RWS)$ bits

Συνηθέστερες επιλογές για SWS και RWS

- ▶ $RWS = 1 \Rightarrow$ πρωτόκολλο Go-Back-N
- ▶ $RWS = SWS \Rightarrow$ πρωτόκολλο Selective Repeat

- ⇒ Το ABP είναι ένα πρωτόκολλο κυλιόμενου παραθύρου με $SWS = 1$ και $RWS = 1$
- ⇒ Όταν αναφέρουμε ότι το μέγεθος του παραθύρου είναι $W \Rightarrow SWS = RWS = W$
- ⇒ Δεν έχει νόημα να ισχύει $RWS > SWS$ (δεν υπάρχει περίπτωση να φτάσουν στον παραλήπτη περισσότερα από τα πλαίσια που μεταδίδει ο αποστολέας)

Διάρθρωση

Λογική σύνδεση

1 Πλαισίωση

2 Ανίχνευση, διόρθωση και έλεγχος σφαλμάτων

4 Πρωτόκολλα ARQ

Alternating Bit Protocol

ARQ με κυλιόμενο παράθυρο

Go-Back-N

Selective Repeat

Έλεγχος ροής

Παραδείγματα Τεχνολογιών



Go-Back-N: Εισαγωγή

Το πρωτόκολλο **Go-Back-N (GBN)** ανήκει στην κατηγορία Sliding Window ARQ

- ▶ μέγεθος παραθύρου αποστολέα $SW_S = W$
- ▶ μέγεθος παραθύρου παραλήπτη $RWS = 1$

Επιτυγχάνει **μέτρια αξιοποίηση** του συνδέσμου και χαρακτηρίζεται από **μέτρια πολυπλοκότητα**

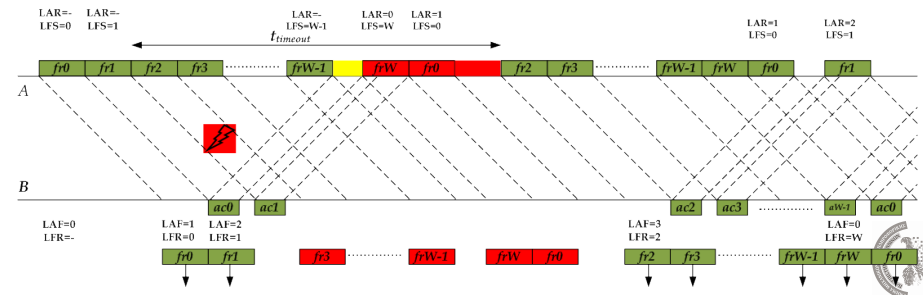
- ▶ δεν επιτυγχάνει υψηλή ρυθμαπόδοση σε συνδέσμους με υψηλό ρυθμό σφαλμάτων
- ▶ απαιτείται δυνατότητα αποθήκευσης μέχρι W πλαισίων στον αποστολέα
- ▶ δεν απαιτείται δυνατότητα αποθήκευσης στον παραλήπτη
- ▶ απαιτείται η χρήση $W + 1$ αριθμών ακολουθίας

⇒ Για την αναπαράσταση των αριθμών ακολουθίας χρειάζονται $\log_2(W + 1)$ bits στην κεφαλίδα ενός πλαισίου
 ⇒ Χρειάζονται $W + 1$ αριθμοί ακολουθίας ενώ το μέγιστο πλήθος των υπό μετάδοση πλαισίων είναι W

Λειτουργία Go-Back-N (1/2)

Αποστολέας:

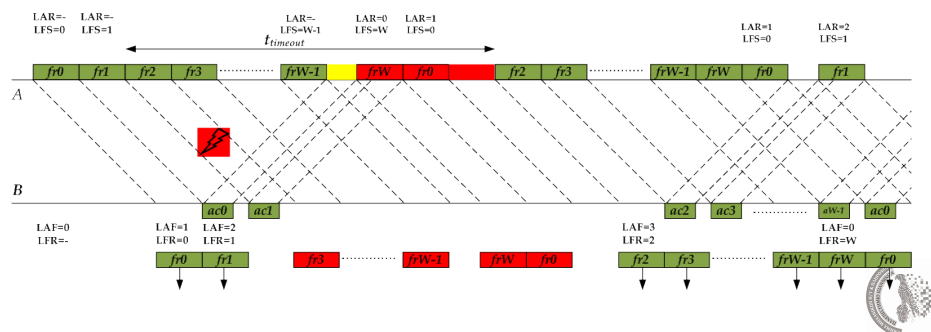
- ▶ αριθμεί τα προς μετάδοση πλαίσια χρησιμοποιώντας τους ακολουθιακούς αριθμούς $0, \dots, W$ **κυκλικά**
- ▶ εκπέμπει μέχρι W μη επιβεβαιωμένα πλαίσια
- ▶ αν δεν ληφθεί επιβεβαίωση για ένα πλαίσιο με αριθμό ακολουθίας k τότε επανεκπέμπονται όλα τα πλαίσια από το k μέχρι την εξάντληση του παραθύρου



Λειτουργία Go-Back-N (2/2)

Παραλήπτης:

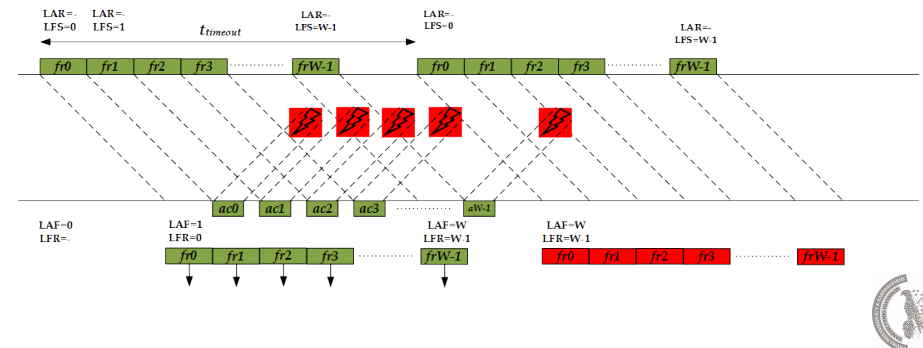
- ▶ ένα πλαίσιο γίνεται αποδεκτό και επιβεβαιώνεται (με ένα πλαίσιο ACK με τον ίδιο αριθμό ακολουθίας) μόνο αν ληφθεί στη σωστή σειρά
- ▶ τα πλαίσια με σωστή σειρά παραδίδονται στο ανώτερο επίπεδο
- ▶ απορρίπτει όλα τα πλαίσια που λαμβάνονται **εκτός σειράς**



Αριθμοί ακολουθίας στο Go-Back-N

Απαιτούνται $W + 1$ διαφορετικοί αριθμοί ακολουθίας για να γίνεται διάκριση των **διπλότυπων πλαισίων**

- ▶ ακόμα και όταν υπάρχουν W ανεπιβεβαιωμένα πλαίσια



Διάρθρωση

- 1. Λογική σύνδεση
- 2. Πλαισίωση
- 3. Ανίχνευση, διόρθωση και έλεγχος σφαλμάτων
- 4. Πρωτόκολλα ARQ
 - Alternating Bit Protocol
 - ARQ με κυλιόμενο παράθυρο
 - Go-Back-N
 - Selective Repeat
- Έλεγχος ροής
- Παραδείγματα Τεχνολογιών



Selective Repeat: Εισαγωγή

Το πρωτόκολλο **Selective Repeat (SR)** ανήκει στην κατηγορία Sliding Window ARQ

- ▶ μέγεθος παραθύρου αποστολέα $SWS = W$
- ▶ μέγεθος παραθύρου παραλήπτη $RWS = SWS$

Επιτυγχάνει **καλή αξιοποίηση** του συνδέσμου και χαρακτηρίζεται από **υψηλή πολυπλοκότητα**

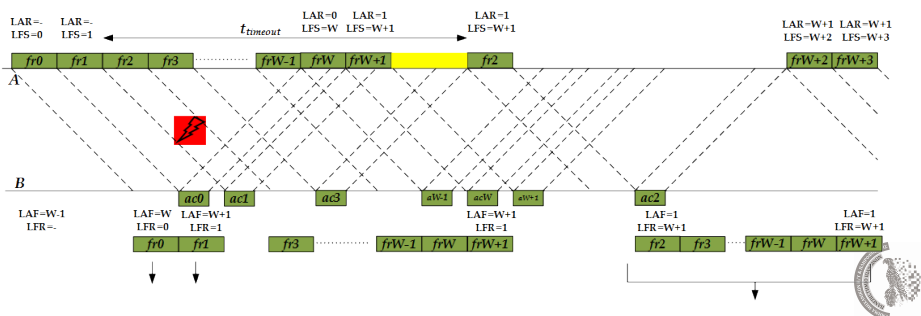
- ▶ απαιτείται δυνατότητα αποθήκευσης μέχρι W πλαισίων στον αποστολέα
- ▶ απαιτείται δυνατότητα αποθήκευσης μέχρι $W - 1$ πλαισίων στον παραλήπτη
- ▶ απαιτείται η χρήση $2W$ αριθμών ακολουθίας

⇒ Για την αναπαράσταση των αριθμών ακολουθίας χρειάζονται $\log_2(2W)$ bits στην κεφαλίδα ενός πλαισίου
 ⇒ Χρειάζονται $2W$ αριθμοί ακολουθίας ενώ το μέγιστο πλήθος των υπό μετάδοση πλαισίων είναι W

Λειτουργία Selective Repeat (1/2)

Αποστολέας:

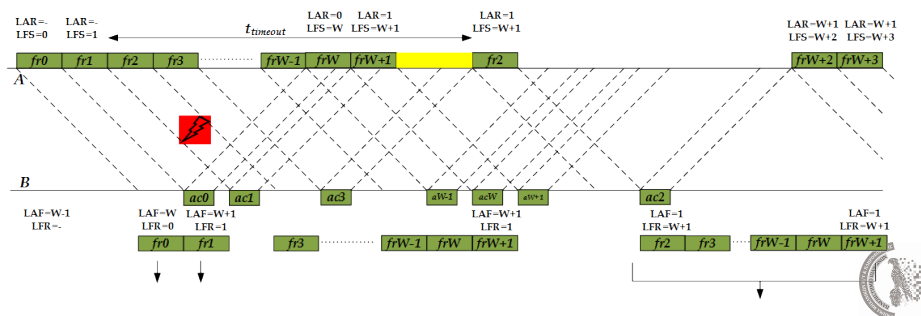
- ▶ αριθμεί τα προς μετάδοση πλαίσια χρησιμοποιώντας τους ακολουθιακούς αριθμούς $0, \dots, 2W - 1$ **κυκλικά**
- ▶ εκπέμπει μέχρι W μη επιβεβαιωμένα πλαίσια
- ▶ αν δεν ληφθεί επιβεβαίωση για ένα πλαίσιο με αριθμό ακολουθίας k τότε επανεκπέμπεται μόνο το πλαίσιο αυτό



Λειτουργία Selective Repeat (2/2)

Παραλήπτης:

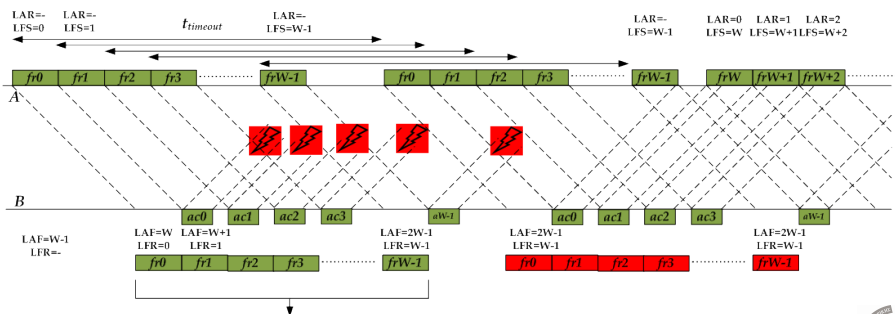
- ▶ επιβεβαιώνει την ορθή λήψη ενός πλαισίου με ένα πλαίσιο ACK με τον ίδιο αριθμό ακολουθίας
- ▶ αποθηκεύει τα πλαίσια που φτάνουν **εκτός σειράς**
- ▶ όταν είναι δυνατή η ανάκτηση της σειράς μιας ομάδας πλαισίων τότε αυτά προωθούνται στο ανώτερο επίπεδο



Αριθμοί ακολουθίας στο Selective Repeat

Απαιτούνται $2W$ διαφορετικοί αριθμοί ακολουθίας για να γίνεται διάκριση των διπλότυπων πλαισίων

▶ ακόμα και όταν χαθούν οι επιβεβαιώσεις για W πλαίσια



Διάρθρωση

1. Λογική σύνδεση
2. Πλαισίωση
3. Ανίχνευση, διόρθωση και έλεγχος σφαλμάτων
4. Πρωτόκολλα ARQ
 - Alternating Bit Protocol
 - ARQ με κυλιόμενο παράθυρο
 - Go-Back-N
 - Selective Repeat
5. Έλεγχος ροής
6. Παραδείγματα Τεχνολογιών

Αναγκαιότητα αλγόριθμων ελέγχου ροής

Πρόβλημα **ελέγχου ροής**: ο παραλήπτης δεν μπορεί να λάβει τα δεδομένα με το ρυθμό που του αποστέλλονται

- ▶ π.χ. λόγω αυξημένου φόρτου επεξεργασίας
- ▶ συνέπεια: απώλεια πλαισίων λόγω του πεπερασμένου αποθηκευτικού χώρου

Οι αλγόριθμοι ARQ αντιμετωπίζουν το πρόβλημα ως μέρος του γενικότερου προβλήματος της απώλειας πλαισίων

Αναγκαιότητα: αλγόριθμοι **μειωμένης πολυπλοκότητας** που επιλύουν μόνο το πρόβλημα του ελέγχου ροής

- ▶ χρήσιμοι σε περιπτώσεις που τα σφάλματα στο κανάλι **δεν υπάρχουν** ή διορθώνονται με ένα κώδικα ευθείας διόρθωσης
- ▶ ονομάζονται **αλγόριθμοι ελέγχου ροής (flow control algorithms)**

Κατηγορίες αλγόριθμων ελέγχου ροής

Η τεχνική του **κυλιόμενου παραθύρου** χρησιμοποιείται και στους αλγόριθμους ελέγχου ροής

- ▶ οι αλγόριθμοι αυτοί ανήκουν στην κατηγορία αλγορίθμων **ελέγχου ροής με ανάδραση (feedback-based flow control)**

Οι σημαντικότεροι αλγόριθμοι είναι οι:

- ▶ Stop-and-Wait
- ▶ Sliding Window flow control

- ⇒ Ο έλεγχος ροής με ανάδραση ονομάζεται και **έλεγχος ροής κλειστού βρόχου (closed loop flow control)**
- ⇒ Η άλλη κατηγορία ελέγχου ροής ονομάζεται **έλεγχος ροής ανοιχτού βρόχου (open loop flow control)**

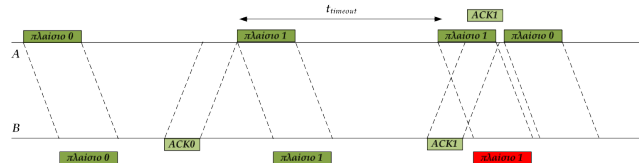
Stop-and-Wait

Ο αποστολέας μπορεί να μεταδώσει μόνο ένα πλαίσιο αναμένοντας την επιβεβαίωσή του

- ▶ το πλαίσιο επανεκπέμπεται αν δεν επιβεβαιωθεί μέσα στο χρόνο αναμονής

Ο παραλήπτης επιβεβαιώνει ένα πλαίσιο μόνο αν διαθέτει αποθηκευτικό χώρο για να λάβει και επόμενο πλαίσιο

- ▶ με τον τρόπο αυτό μπορεί να ελέγξει την αποστολή δεδομένων



Μειονέκτημα: χαμηλή ρυθμαπόδοση ακόμα και αν ο παραλήπτης μπορεί να λαμβάνει πλαίσια χωρίς διακοπές

⇒ Οι αλγόριθμοι ελέγχου ροής υποθέτουν ότι δεν υπάρχουν σφάλματα στο κανάλι

Sliding Window (1/2)

Αποτελεί γενίκευση του Stop-and-Wait με τη βοήθεια της τεχνικής Sliding Window

- ▶ ο αποστολέας μπορεί να μεταδώσει μέχρι SWS πλαίσια
- ▶ ο παραλήπτης μπορεί να αποθηκεύει μέχρι RWS λαμβανόμενα πλαίσια

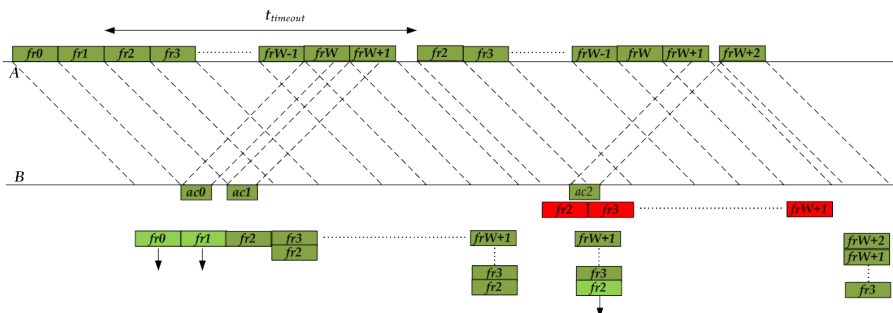
Ισχύει $SWS = RWS$

- ▶ δεν έχει νόημα η επιλογή $SWS < RWS$ (ο παραλήπτης δεν θα λάβει περισσότερα πλαίσια από αυτά που στέλνει ο αποστολέας)
- ▶ επίσης δεν έχει νόημα η επιλογή $SWS > RWS$ (ο αποστολέας στέλνει πλαίσια τα οποία ενδέχεται να μην μπορούν να αποθηκευτούν)

Ο παραλήπτης επιβεβαιώνει ένα πλαίσιο μόνο αν διαθέτει αποθηκευτικό χώρο για τουλάχιστον RWS πλαίσια

- ▶ με τον τρόπο αυτό μπορεί να περιορίσει το ρυθμό αποστολής

Sliding Window (2/2)



⇒ Σε μια εναλλακτική υλοποίηση ο παραλήπτης μπορεί να **διαμορφώνει δυναμικά** το μέγεθος του παραθύρου και να **ενημερώνει (μέσω των επιβεβαιώσεων)** τον αποστολέα για το νέο μέγεθος

Διάρθρωση

- 1 Λογική σύνδεση
- 2 Πλαισίωση
- 3 Ανίχνευση, διόρθωση και έλεγχος σφαλμάτων
- 4 Πρωτόκολλα ARQ
 - Alternating Bit Protocol
 - ARQ με κυλιόμενο παράθυρο
 - Go-Back-N
 - Selective Repeat
- 5 Έλεγχος ροής
- 6 Παραδείγματα Τεχνολογιών

High-level Data Link Control (1/2)

Το πρωτόκολλο **High-level Data Link Control, HDLC** είναι ένα πρωτόκολλο του επιπέδου συνδέσμου μετάδοσης δεδομένων (data link layer)

- ▶ προτάθηκε από τον οργανισμό ISO
- ▶ η αρχική του έκδοση (**Synchronous Data Link Control, SDLC**) προτάθηκε από την IBM
- ▶ μετεξέλιξη του αποτελεί το **Link Access Procedure, LAP** που προτάθηκε από την CCITT και χρησιμοποιείται στα δίκτυα X.25

Η οριοθέτηση του πλαισίου γίνεται με **δείκτες συνόρων και συμπλήρωση bit**

- ▶ ο χαρακτήρας που σηματοδοτεί την έναρξη και λήξη του πλαισίου είναι ο 01111110
- ▶ η πληροφορία αυτή μεταδίδεται και όταν δεν υπάρχουν δεδομένα ώστε να διευκολύνεται ο συγχρονισμός των κόμβων που επικοινωνούν



High-level Data Link Control (2/2)

Πεδία πλαισίου HDLC:

- ▶ address: περιέχει τη διεύθυνση του παραλήπτη
- ▶ control: χρησιμοποιείται για να ορίσει διαφορετικούς τύπους πλαισίων (ελέγχου και δεδομένων)
- ▶ checksum: χρησιμοποιείται για ανίχνευση σφαλμάτων (CRC κώδικας με ακολουθία αναφοράς 1000100000010001, CRC-16-CCITT)



Για τον έλεγχο σφαλμάτων το HDLC χρησιμοποιεί την τεχνική sliding window

- ▶ το μέγεθος του παραθύρου αποστολέα είναι 7 (3 bit για ακολουθιακούς αριθμούς)



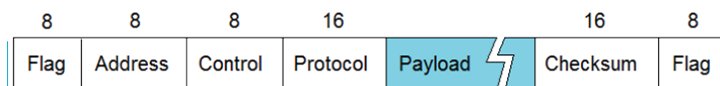
Point-to-Point Protocol (1/2)

Το **Point-to-Point Protocol, PPP** χρησιμοποιείται για απευθείας συνδέσεις δύο κόμβων

- ▶ συνήθως χρησιμοποιείται σε **dial-up** συνδέσεις

Η οριοθέτηση του πλαισίου γίνεται με **δείκτες συνόρων και συμπλήρωση byte**

- ▶ ο χαρακτήρας που σηματοδοτεί την έναρξη και λήξη του πλαισίου (flag) είναι ο 01111110
- ▶ το μέγεθος του πλαισίου είναι συνήθως 1500 bytes αλλά μπορεί να καθοριστεί διαφορετικά



Point-to-Point Protocol (2/2)

Πεδία πλαισίου PPP:

- ▶ address: περιέχει τη διεύθυνση του παραλήπτη
- ▶ protocol: χρησιμοποιείται ως **κλειδί αποπολύπλεξης** (προσδιορίζει το πρωτόκολλο ανώτερου επιπέδου στο οποίο πρέπει να παραδοθεί ή από το οποίο προέρχεται το ωφέλιμο φορτίο)
- ▶ checksum: χρησιμοποιείται για ανίχνευση σφαλμάτων (CRC κώδικας με ακολουθία αναφοράς 1000100000010001)

Το μέγεθος πολλών πεδίων του πλαισίου PPP είναι διαπραγματεύσιμο

- ▶ το πρωτόκολλο **Link Control Protocol (LCP)** χρησιμοποιείται για τον καθορισμό του μεγέθους των πεδίων

Το πρωτόκολλο LCP είναι επίσης υπεύθυνο για την έναρξη της επικοινωνίας μεταξύ δύο κόμβων

- ▶ τα πλαίσια που χρησιμοποιεί το LCP ονομάζονται **πλαίσια ελέγχου**
- ▶ κάθε είδος πλαισίου χαρακτηρίζεται από διαφορετική τιμή στο **πεδίο control**

