

Σχεδιασμός Πρωτοκόλλων

Χειμερινό εξάμηνο

nancy@di.uoa.gr

Ιστορική αναδρομή

– Η ανάγκη για γρήγορη μεταφορά πληροφορίας σε μεγάλες αποστάσεις είναι τόσο παλιά όσο και ο άνθρωπος

– Δρομείς, ιππείς και ταχυδρομικά περιστέρια ήταν οι πρώτες λύσεις στο πρόβλημα

– Ανάγκη για βελτίωση --> δημιουργία συστημάτων και πρωτοκόλλων που λίγο διαφέρουν στις αρχές λειτουργίας τους από τις σημερινές λύσεις

Φρυκτωρίες

Στην τραγωδία «Αγαμέμνων», πού γράφτηκε εδώ και δύο χιλιάδες τετρακόσια χρόνια από τον ποιητή Αισχύλο, η βασίλισσα Κλυταιμνήστρα πληροφορεί τον χορό ότι η Τροία έπεσε στα χέρια των Ελλήνων. Ξαφνιασμένος ο χορός ρωτάει : «Ποιος μπόρεσε να φέρει τόσο γρήγορα τα νέα» ; Κι η Κλυταιμνήστρα απαντά: «Ο Ήφαιστος! Δυνατή φωτιά άναψαν στην κορυφή της Ίδας, και πολλές φωτιές μετά μάς έφεραν διαδοχικά το μήνυμα: από την Ίδα το μήνυμα της φλόγας πήγε στον κάβο Ερμή της Λήμνου, και τρίτη η ψηλή κορυφή του 'Αθω δέχτηκε το φως. Με δύναμη η φλόγα γεφύρωσε το πέλαγος, κι έφτασε τις Βίγλες του Μακίστου, στην Εύβοια. Οι φρουροί προώθησαν το μήνυμα στον Εύριπο, στους ανθρώπους πάνω στο Μεσσάπιο. Κι αυτοί έβαλαν φωτιά σ' ένα σωρό με ρείκια, και πάνω από τούς κάμπους του Αισώπου έστειλαν το μήνυμα στην κορυφή του Κιθαιρώνα. Κι από κει πάλι με ζήλο πολύ προώθησαν το μήνυμα, και το έστειλαν πάνω από την Γοργώπιδα λίμνη στο Αιγίπλαγκτο όρος. Και η φλόγα πέρασε τον Σαρωνικό και έφτασε σαν κεραυνός στις κορυφές του Αραχναίου, και τέλος εδώ το λάβαμε, των Ατρείδών τις στέγες, το φως αυτό, πού προπάππος του είναι η φωτιά της Ίδας»

Τηλέγραφος του Αινεία (350 π.χ.)



Τηλέγραφος του Πολύβιου

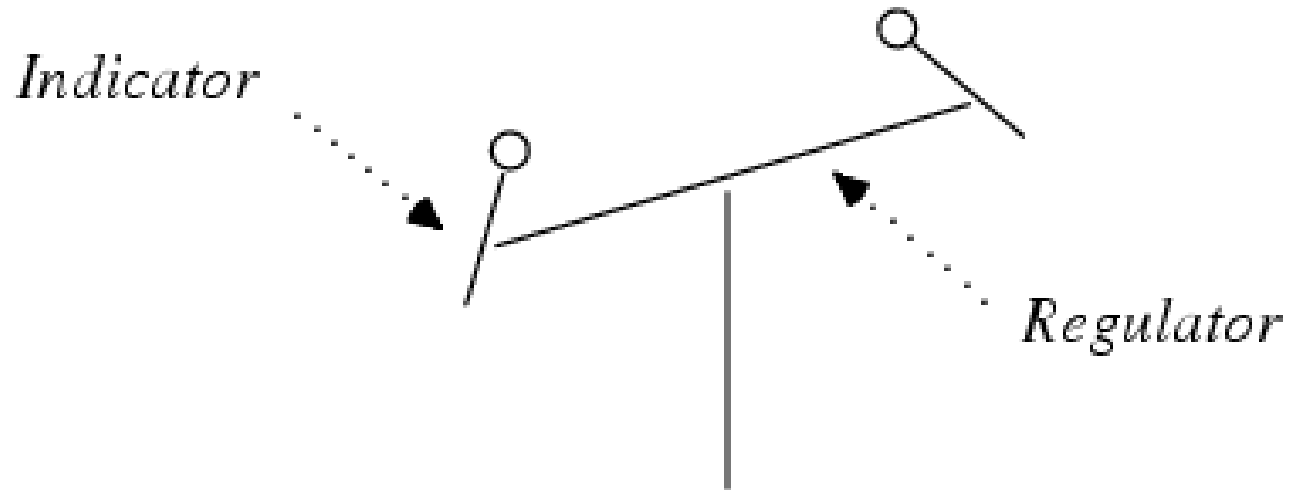
	1	2	3	4	5
1	A	B	Γ	Δ	E
2	Z	H	Θ	I	K
3	Λ	M	N	Ξ	O
4	Π	P	Σ	T	Υ
5	Φ	Χ	Ψ	Ω	

Ιστορική αναδρομή

- 1684 - Οπτικό σύστημα του Hooke (ξύλινα γράμματα και τηλεσκόπιο)
- 1796 – Huth «τηλέφωνο» (Παρόμοια ιδέα από Μ. Αλέξανδρο?)
- Πόλεμος Αμερικάνικης ανεξαρτησίας (σημαία, καλάθι, βαρέλι)

Ιστορική αναδρομή

- 1793 C. Chappe: $8 \times 8 \times 4 = 256$ συνδυασμοί
556 σταθμοί \rightarrow κάλυψη 3000 μίλια



Ιστορική αναδρομή

- Σύστημα με έξι κλείστρα (εξαδικό σύστημα επικοινωνίας) G. Murray Βρετανία
- Σύστημα με δέκα κλείστρα - Σουηδία:
session control (start, stop), error control,
flow control (slower, faster), negative ack
(cannot see)
- Ηλεκτρικός τηλέγραφος κτλ

Πρωτόκολλα επικοινωνιών

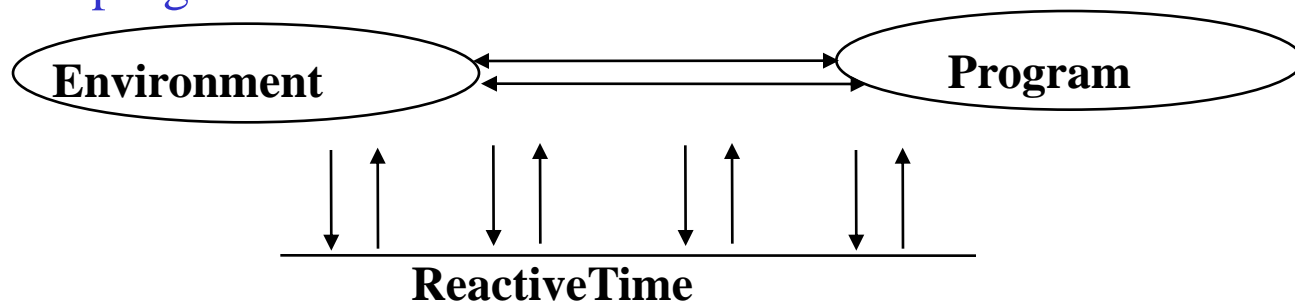
- Ανεξάρτητα από το τι προσπαθούν να αντιμετωπίσουν, τα πρωτόκολλα έχουν να αντιμετωπίσουν παρόμοια προβλήματα
- Σε αυτό το μάθημα θα μάθουμε ποια είναι αυτά τα προβλήματα και πώς να τα αντιμετωπίζουμε

Εισαγωγή

- **Πρωτόκολλο:** σύνολο κανόνων που ορίζει το πώς θα εκτελεστεί μία διεργασία
- Φορμαλιστική περιγραφή & επαλήθευση πρωτοκόλλων (SDL - Specification & Description Language, Estelle, LOTOS)
- complex, concurrent (parallel processing), quality critical (operating systems), safety critical (railway signaling), security critical, standardised systems
- Transformational programs:
initial state --f---> final state



- Reactive programs:

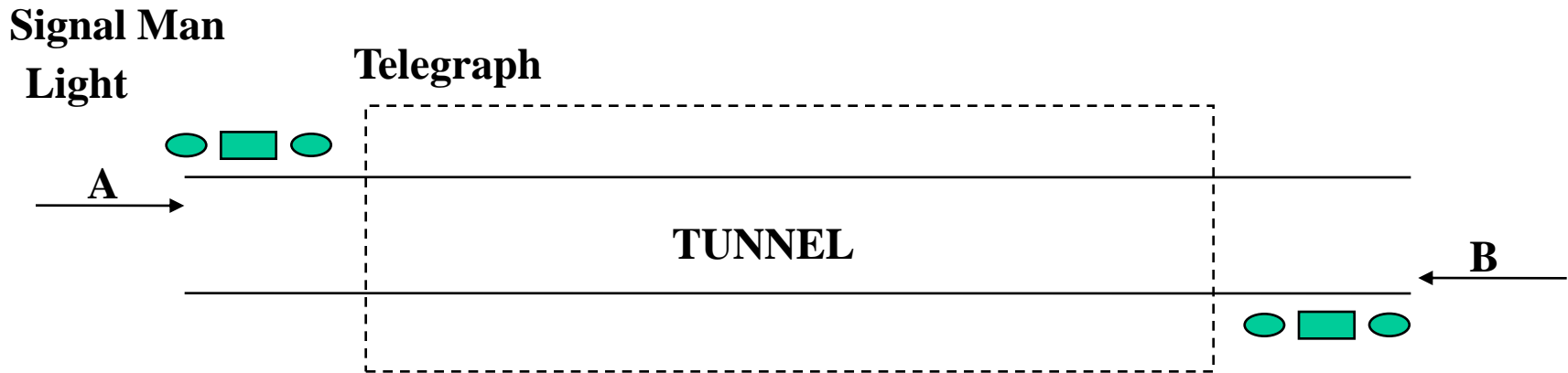


Πρωτόκολλο Επικοινωνίας

- Ορίζει την ακριβή μορφοποίηση των μηνυμάτων που ανταλλάσσονται (συντακτικό)
- Ορίζει τους διαδικαστικούς κανόνες για την ανταλλαγή των δεδομένων (γραμματική)
- Ορίζει το λεξιλόγιο των μηνυμάτων που μπορούν να ανταλλάξουν και τη σημασία τους (σημασιολογικά)

Παράδειγμα

Σηματοδοσία τραίνων (1861 - Αγγλία)



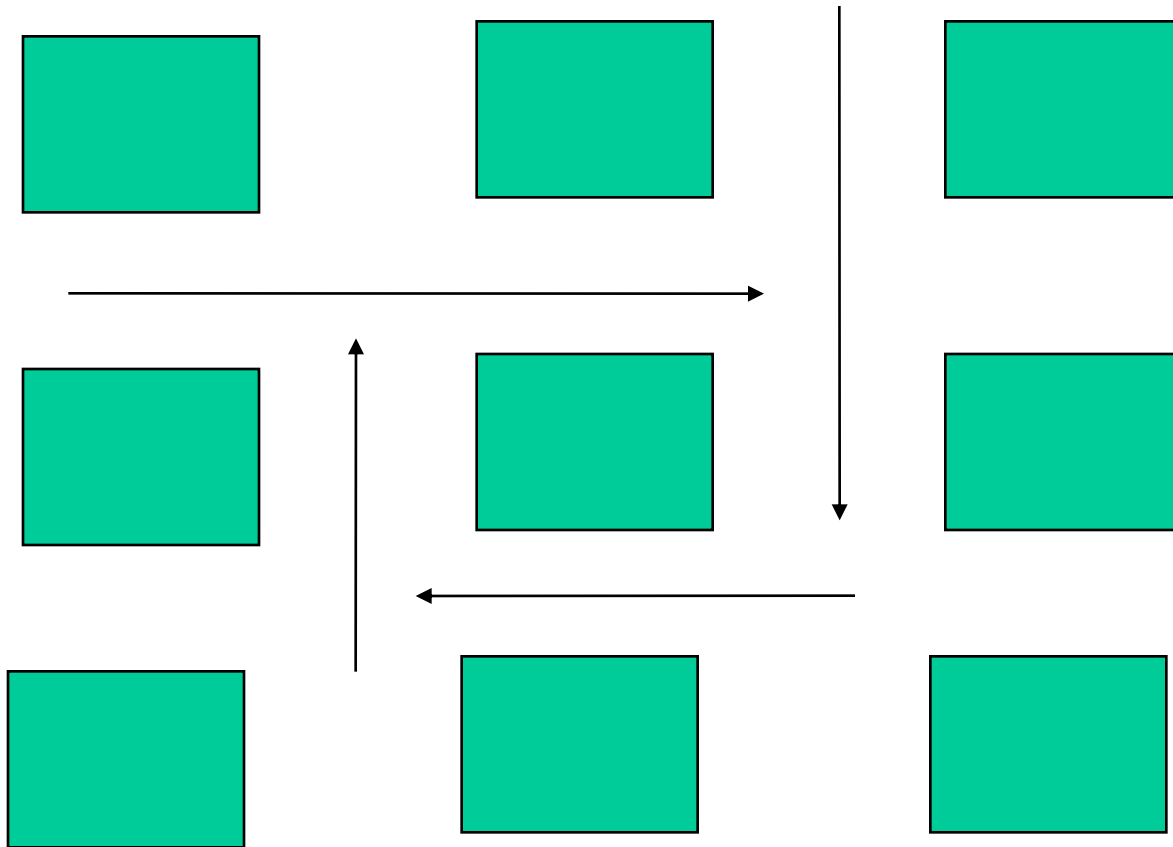
- Telegraph Signals:**
- 1. train in tunnel*
 - 2. tunnel is free*
 - 3. has train left tunnel?*

Τραίνο 1: Περνάει και ο σηματοδότης δεν δουλεύει
Τραίνο 2: Μπαίνει στη σήραγγα αλλά βλέπει σημαία
Τραίνο 3: Σταματάει πριν τη σήραγγα
Αποτέλεσμα: 26 νεκροί & 177 τραυματίες

Αναζήτηση λαθών

- Γιατί είναι εύκολο να συμβούν λάθη σε συστήματα παράλληλων διεργασιών;
- Συμβαίνει μόνο στο σχεδιασμό λογισμικού; (π.χ., σουπερμάρκετ, κυκλική πορεία)
- Ένας συνδυασμός γεγονότων μπορεί να έχει απρόβλεπτες (διασκεδαστικές ή δραματικές συνέπειες)

Κυκλική πορεία



Τι γίνεται με τις κυκλικές πορείες;

Μπορεί να λυθεί δίνοντας προτεραιότητα; Δημιουργεί νέα προβλήματα;

Άλλα προβλήματα

- Αδιέξοδα σε λειτουργικά συστήματα. Πώς αντιμετωπίζονται;

Άλλα προβλήματα

- Λάθος υποθέσεις. Όταν φτιάχνουμε μεγάλα συστήματα συνθέτουμε κομμάτια
- Υπάρχουν προβλήματα που παρουσιάζονται μόνο όταν έχουμε συνθέσει όλα τα κομμάτια
- Παράδειγμα Lufthansa Airbus 1993 (thrust reverser): α) τροχοί κάτω, β) οι τροχοί να γυρίζουν γ) το βάρος του αεροπλάνου να είναι στους τροχούς). Τι γίνεται όμως αν πατινάρουν οι τροχοί λόγω υγρασίας και υπάρχουν άνεμοι;

Τηλεφωνικό σύστημα

- Ανάγκη πρόσβασης τουλάχιστον σε δύο διαμοιραζόμενους πόρους (γραμμή καλούντος και γραμμή καλούμενου)
- Σειρά εκχώρησης πόρων;
- Τι γίνεται αν και οι δύο προσπαθήσουν παράλληλα να καλέσουν;
- Επιτυγχάνεται έτσι ο στόχος τους;

Πρωτόκολλα Επικοινωνίας

- Μέχρι πριν λίγα (συγκριτικά χρόνια) παρεμβαλλόταν άνθρωπος
- Αύξηση στην ταχύτητα αλλά και στο ρυθμό των λαθών!
- Επιπλέον, ανάγκη για τη δημιουργία των οργανισμών προτυποποίησης

Παραδοχές για το σχεδιασμό πρωτοκόλων

Το πρωτόκολλο περιλαμβάνει παραδοχές και συμφωνίες για τις μεθόδους που θα χρησιμοποιηθούν για:

- Initiation and termination of data exchanges
- Synchronization of senders and receivers
- Detection and correction of transmission errors
- Formatting and encoding of data

Προδιαγραφή πρωτοκόλλου – 5 διακριτά σημεία

- The service (υπηρεσία) to be provided by the protocol
- The assumptions (υποθέσεις) about the environment in which the protocol is executed
- The vocabulary (λεξιλόγιο) of messages used to implement the protocol
- The encoding (format) (κωδικοποίηση) of each message in the vocabulary
- The procedure rules (κανονες) guarding the consistency of message exchanges

Σχεδιασμός

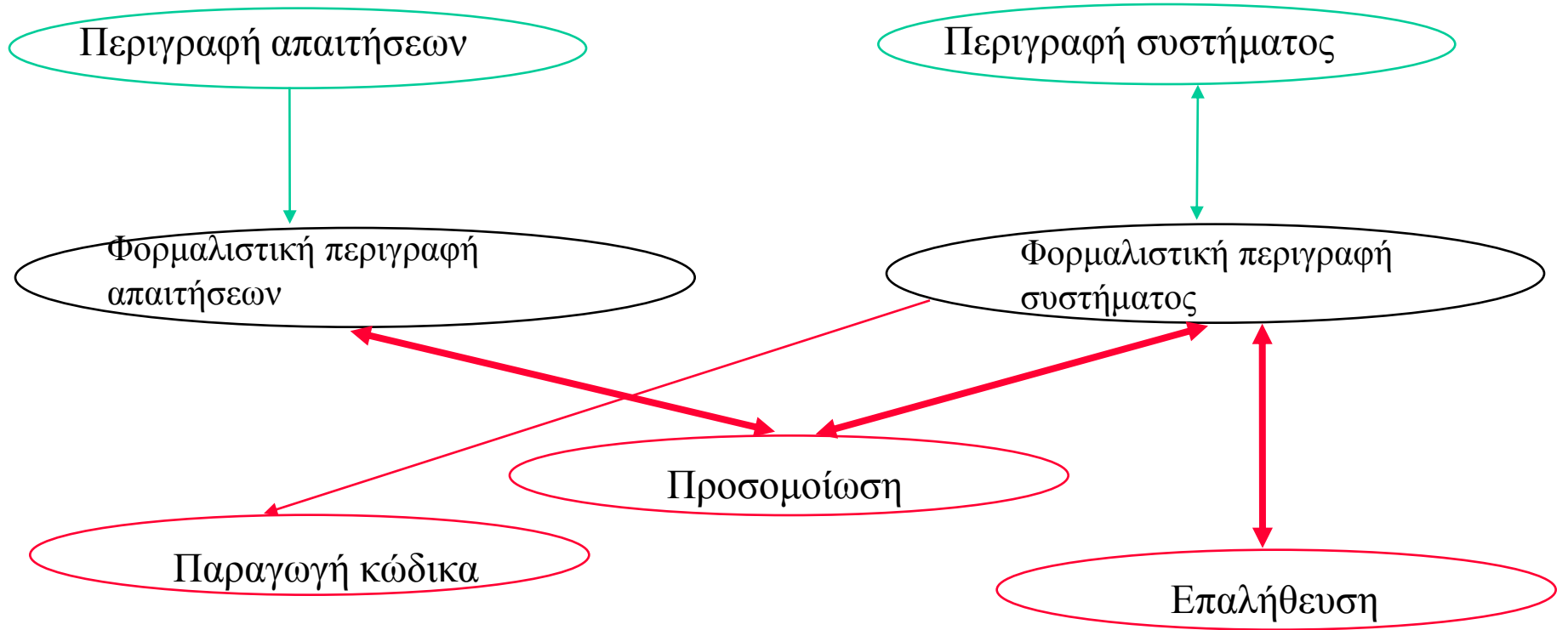
SERVICE SPECIFICATION

The purpose of the protocol is to transfer text files as sequences of characters across a telephone line while protecting against transmission errors, assuming that all transmission errors can in fact be detected. The protocol is defined for full-duplex file transfer, that is, it should allow for transfers in two directions simultaneously. Positive and negative acknowledgments for traffic from A to B are sent on the channel from B to A, and vice versa. Every message contains two parts: a message part, and a control part that applies to traffic on the reverse channel.

ASSUMPTIONS ABOUT THE ENVIRONMENT

The "environment" in which the protocol is to be executed consists minimally of two users of the file transfer service and a transmission channel. The users can be assumed to simply submit a request for file transfer and await its completion. The transmission channel is assumed to cause arbitrary message distortions, but not to lose, duplicate, insert, or reorder messages. We will assume here that a lower-level module is used to catch all distortions and change them into undistorted messages of type err.

Μεθοδολογία



Μηχανές Πεπερασμένων Καταστάσεων

Ορισμός: $M = (K, \Sigma, g, s, F)$

1. K = πεπερασμένο σύνολο καταστάσεων
2. Σ = σύνολο σημάτων εισόδου
3. F (υποσύνολο του K) = τελικές καταστάσεις
4. s = Αρχική κατάσταση
5. g = συνάρτηση μετάβασης

Αν $\sigma \in \Sigma$ & $q \in K \implies g(q, \sigma) \in K$

Προδιαγραφή Πρωτοκόλλων

1. *Αποστολέας*: στέλνει πακέτο στον παραλήπτη, και κρατάει αντίγραφο
2. *Παραλήπτης*: ειδοποιεί το αποστολέα αν το πακέτο έφτασε σωστά ή όχι
3. Τα πακέτα αν έχουν καταστραφεί αγνοούνται
4. Ο αποστολέας χρησιμοποιεί timer

Καταστάσεις διεργασιών:

Αποστολέας:

AE =Ack Expected

NAE =No Ack Expected

Παραλήπτης

MA = must ack

NA = No ack

Καταστάσεις συστήματος: $\chi = (s, r)$

$X_0 = \{NAE, NA\}$ <--- αρχική κατάσταση

$X_1 = \{AE, MA\}$

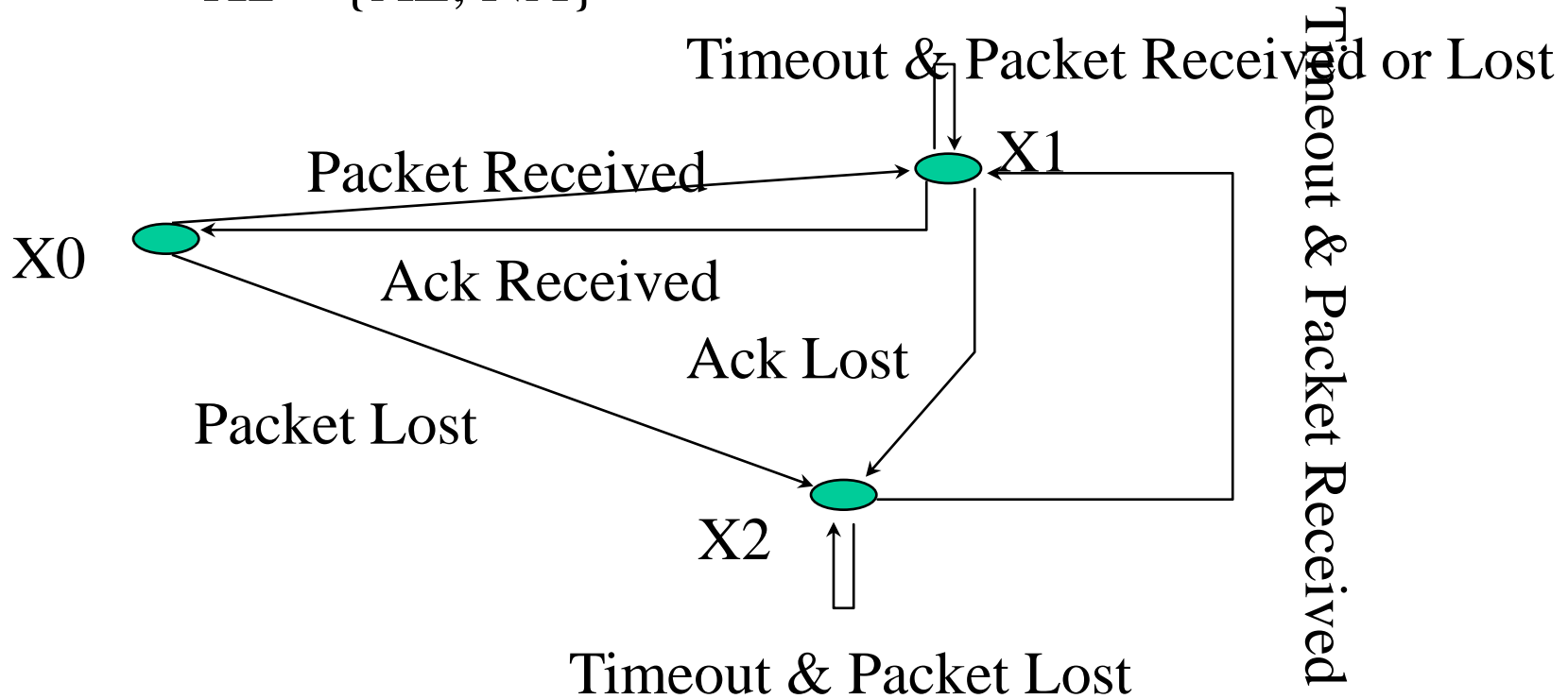
$X_2 = \{AE, NA\}$

State Transition

$X0 = \{NAE, NA\}$ <--- αρχική κατάσταση

$X1 = \{AE, MA\}$

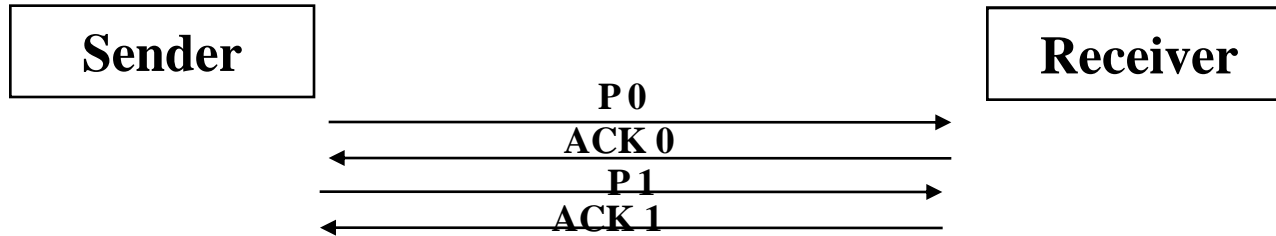
$X2 = \{AE, NA\}$



Χωρίς λάθη: X0, X1, X0, X1

X0, X1, X2, X1 ?

Alternating Bit Protocol



Καταστάσεις διεργασιών:

Αποστολέας:

0=Ack 0 Expected

1=Ack 1 Expected

- = No Ack Expected

Παραλήπτης

0 =P 0 expected

1 =P1 expected

Αρχική Κατάσταση: (- , 0)

Εκτέλεση χωρίς Λάθη:

(-, 0)-->(0, 1)-->(-, 1)-->(1, 0) --> (-, 0) --> (0, 1)

Alternating Bit Protocol

Αποστολέας:

0 = Ack 0 Expected

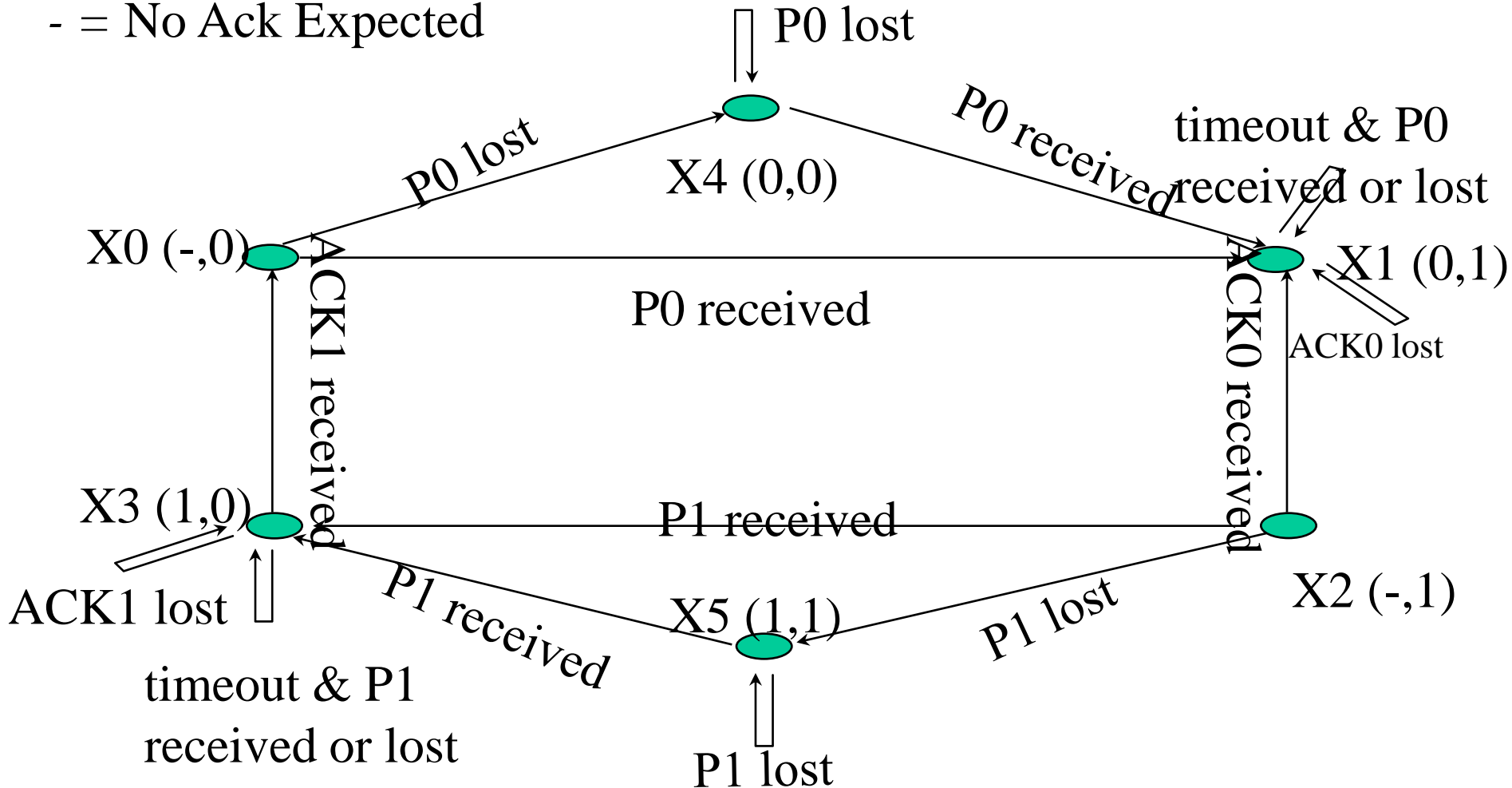
1 = Ack 1 Expected

- = No Ack Expected

Παραλήπτης

0 = P 0 expected

1 = P 1 expected



Επαλήθευση

monitor variable y with values 0, 1, θ

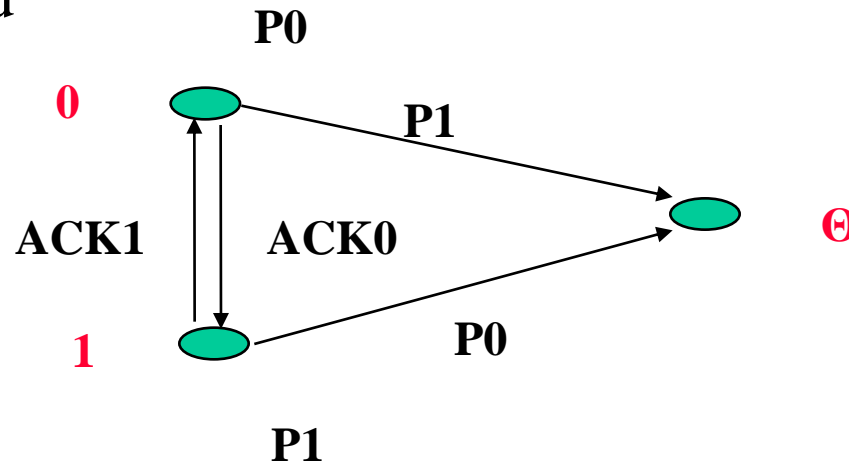
0 --> 1 and vice versa when $A1, A0$

0 --> 0 P0 received

1 --> 1 P1 received

0 --> θ P1 received

1 --> θ P0 received



Επαλήθευση

- **Σύνθεση των δύο μηχανών:**

$(X_0, 0) \dashrightarrow (X_1, 0)$

- Θέλουμε το σύνολο R όλων των καταστάσεων που μπορούμε να επισκεφτούμε

- **Αλγόριθμος:**

1. $X_0 \in R$

2. αν μπορούμε να πάμε στην κατάσταση X_i ,

\Rightarrow μπορούμε να πάμε σε όλες τις
καταστάσεις για τις οποίες έχουν οριστεί
μεταβάσεις

3. **$V(n)$:** οι καταστάσεις των οποίων οι

επόμενες καταστάσεις δεν έχουν

διερευνηθεί. **$R(n)$:** κρατάμε τις

καταστάσεις που έχουμε ήδη πάει

Βήμα 1: $n = 0, V(n) = R(n) = \{x_0\}$

Βήμα 2: $V(n+1): \{g(x,u), x \in V(n), u \in U\} - R(n)$

if $V(n+1) == \text{null}$ then $R=R(n)$

else goto last step

Επαλήθευση του ABP

- $R(0) = V(0) = \{x_0, 0\}$
- $V(1) = \{(x_1, 0), (x_4, 0)\},$
 $R(1) = \{(x_0, 0), (x_1, 0), (x_4, 0)\}$
- $V(2) = \{(x_2, 1)\}$
 $R(2) = \{(x_0, 0), (x_1, 0), (x_4, 0), (x_2, 1)\}$
- $V(3) = \{(x_3, 1), (x_5, 1)\}$
 $R(3) = \{(x_0, 0), (x_1, 0), (x_4, 0), (x_2, 1), (x_3, 1), (x_5, 1)\}$
- $V(4) = \{\text{null}\}$
 $R(4) = R(3) \text{ STOP}$

Αποτέλεσμα: Δεν υπάρχει περίπτωση $(\chi_i, \theta) \implies$

το ABP είναι σωστό

Περιορισμοί

- Μεγάλος αριθμός καταστάσεων του συστήματος σημαίνει και μεγάλο χρόνο επαλήθευσης

- 10 διεργασίες, με 100 καταστάσεις η κάθε μία.

Η μηχανή του συστήματος μπορεί να έχει μέχρι

$$100^{10} = 10^{20} \text{ καταστάσεις}$$

- Η μηχανή που ορίζει την απαίτηση ορθότητας έχει 100 καταστάσεις

- Αν μόνο το 0.01% των καταστάσεων μπορούμε να επισκεφθούμε \implies
 $0.0001 \times 10^{22} = 10^{18}$ εντολές

- Έστω ότι τρέχει σ' ένα υπολογιστή 100 Mips (100^6 εντολές το δευτερόλεπτο) ο αλγόριθμος χρειάζεται

$$(10^{18})/10^8 = 10^{10} \text{ sec δηλ.}$$

> 300 χρόνια

Μηχανές Πεπερασμένων Καταστάσεων που επικοινωνούν

1. Οι μηχανές επικοινωνούν μέσω πεπερασμένων FIFO ουρών
2. Τα σήματα εξόδου της μίας μηχανής είναι σήματα εισόδου της άλλης μηχανής
3. Αν η ουρά είναι άδεια, οι μεταβάσεις που χρειάζονται είσοδο δεν εκτελούνται
4. Αν η ουρά είναι γεμάτη, οι μεταβάσεις που χρειάζονται έξοδο δεν εκτελούνται

Παράδειγμα

Sender

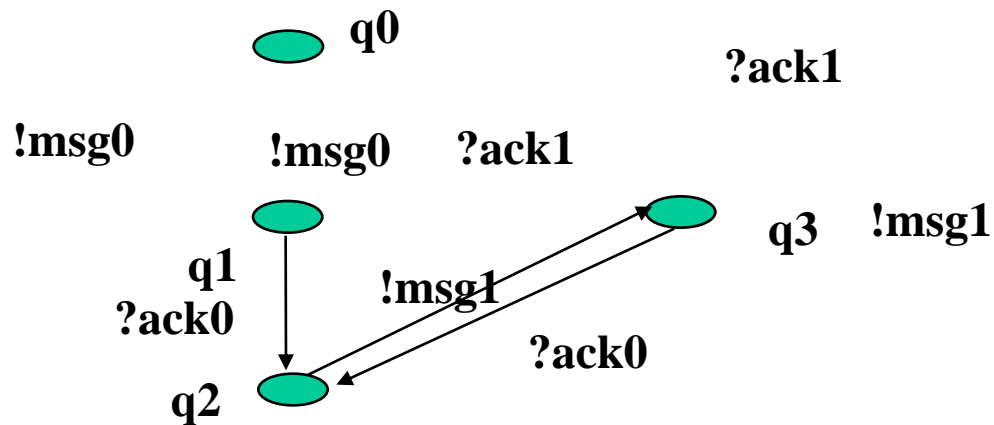
State	Input	Output	Next State
q0	-	msg0	q1
q1	ack1	-	q0
q1	ack0	-	q2
q2	-	msg1	q3
q3	ack0	-	q2
q3	ack1	-	q0

Receiver

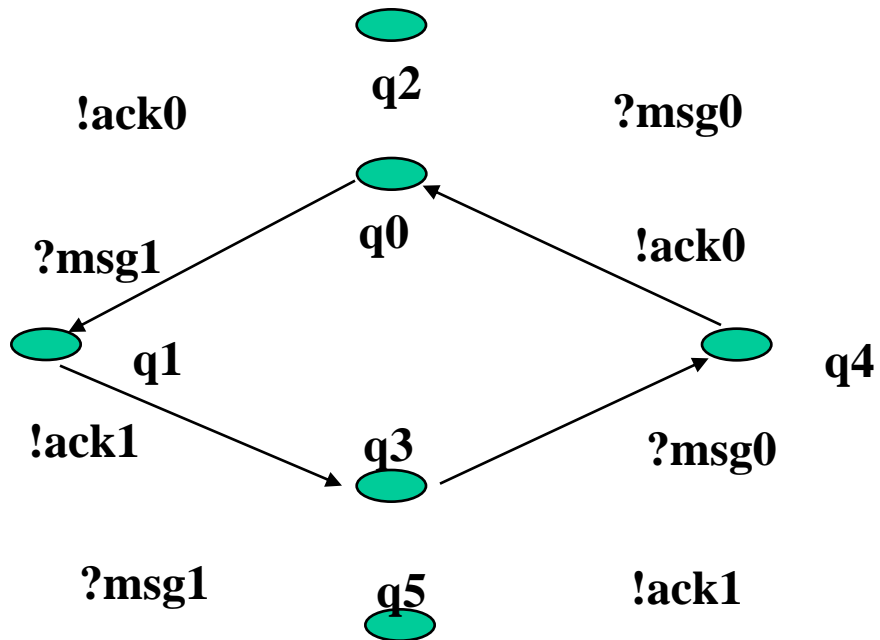
State	Input	Output	Next State
q0	msg1	-	q1
q0	msg0	-	q2
q1	-	ack1	q3
q2	-	ack0	q0
q3	msg0	-	q4
q3	msg1	-	q5
q4	-	ack0	q0
q5	-	ack1	q3

Παράδειγμα

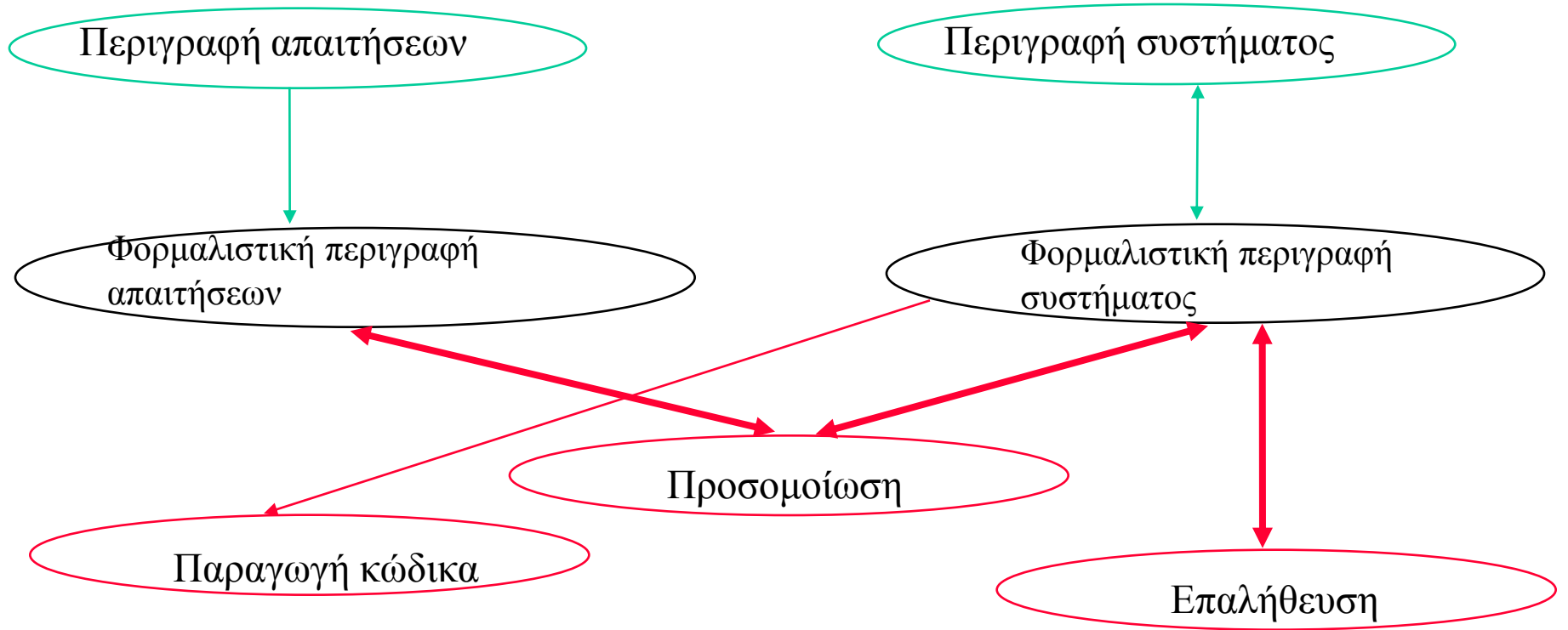
Sender



Receiver



Μεθοδολογία



Δομή Πρωτοκόλλων

Σύνολα κανόνων που ορίζουν την αλληλεπίδραση μεταξύ παράλληλων διαδικασιών

- Αρχή ανταλλαγής δεδομένων
- Συγχρονισμός διεργασιών
- Μορφοποίηση και κωδικοποίηση
- Ανακάλυψη & διόρθωση λαθών
- Λήξη ανταλλαγής δεδομένων