



# ΟΠΤΙΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ

Προγραμματιστική άσκηση 2019–2020

Διδάσκων: Γεώργιος Παπαδημητρίου, Καθηγητής  
Βοηθός Μαθήματος: Αναστάσιος Βαλκάνης, Υπ. Διδάκτορας

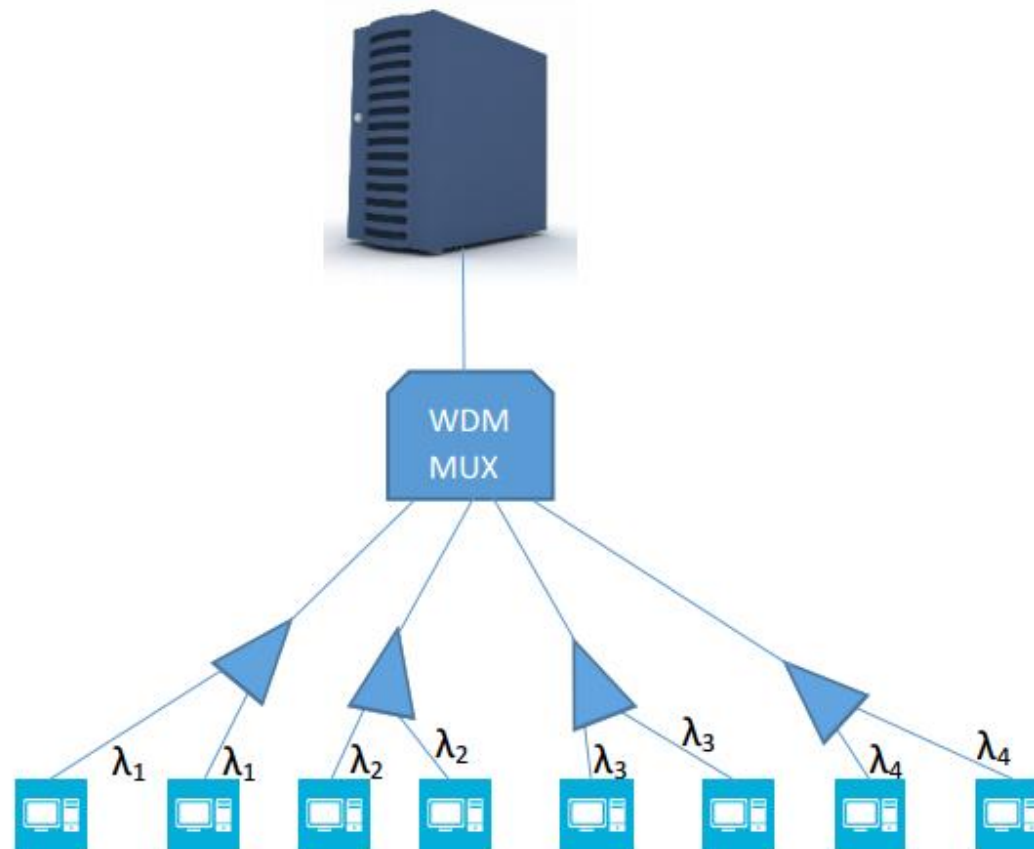
# Παραδοτέα εργασίας

- ▶ Πηγαίος κώδικας (ελεύθερη επιλογή γλώσσας προγραμματισμού)
- ▶ Εκτενή σχόλια
- ▶ Αποτελέσματα προσομοίωσης σε αρχείο κειμένου με σύντομο σχολιασμό
- ▶ Δε χρειάζεται γραφικό περιβάλλον

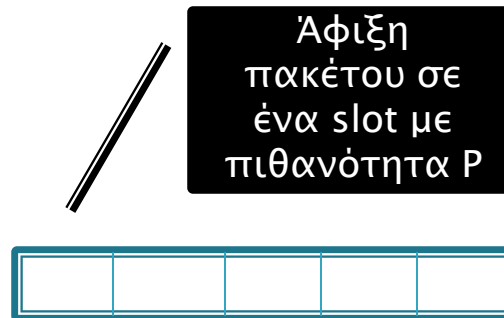
# Προγραμματιστική εργασία

- ▶ Οπτικό δίκτυο με 8 υπολογιστές & 1 server
- ▶ Χρήση 4 μηκών κύματος  $\lambda_1$ ,  $\lambda_2$ ,  $\lambda_3$  και  $\lambda_4$
- ▶ Υπολογιστές ανά 2 μεταδίδουν πακέτα στο ίδιο μήκος κύματος
- ▶ Χρόνος διαιρείται σε slot
- ▶ Διάρκεια μετάδοσης πακέτου = 1 slot

# Τοπολογία οπτικού δικτύου



- ▶ Κάθε υπολογιστής διαθέτει ουρά αναμονής 5 πακέτων



# Άφιξη & Μετάδοση

- ▶ Πιθανότητα άφιξης πακέτων σε ένα slot
  - $P = 0.1, 0.2, 0.3, \dots 1.0$  (arriving prob.)
- ▶ Πιθανότητα μετάδοσης πακέτου προς τον server σε ένα slot
  - $\pi = 0.5$  (transmitting prob.)

# Προγραμματιστική προσέγγιση εργασίας

- ▶ Δημιουργία stations (όνομα, id, packets in buffer, μ.κ. μετάδοσης κλπ.)
- ▶ Δημιουργία packets (χρόνος άφιξης, id, όνομα κλπ.)
- ▶ Γέμισμα των ουρών στα stations με βάση την είσοδο του χρήστη (arriving prob.)
- ▶ Μετάδοση πακέτων προς τον server  $\pi = 0.5$  (transmitting prob.)

# Προγραμματιστική προσέγγιση εργασίας

- ▶ Υπολογισμός συνολικής καθυστέρησης καθ'όλη τη διάρκεια της προσομοίωσης
  - Καθυστέρηση πακέτου → timeslot μετάδοσης – slot άφιξης
- ▶ Υπολογισμός συνολικών πακέτων που στάλθηκαν
- ▶ Αριθμός timeslots (π.χ. 500000 timeslots)
- ▶ Τιμές των μεταβλητών που μας ενδιαφέρουν σαν έξοδο του προγράμματος



# Ζητούμενα εργασίας

- ▶ Μέση καθυστέρηση πακέτου (σε slots) στο δίκτυο
  - *Average delay*
    - $Totaldelay/totalpacketsent$
- ▶ Μέσο αριθμό επιτυχών μεταδόσεων σε ένα slot throughput
  - *Throughput*
    - $Totalpacketsent/timeslots$
- ▶ Ρυθμό χαμένων πακέτων
  - *Packet loss rate*
    - $Totalpacketlost/Totalpacketscreated$

# Υπερχείλιση ουράς αναμονής

- ▶ Όταν ένα πακέτο βρει την ουρά γεμάτη σε κάποιο σταθμό, τότε το πακέτο χάνεται

```
if (packetlist.size() < 5){  
    packetlist.addLast(p);  
}  
else{  
    System.out.println("Buffer full");  
}
```

# Συγκρούσεις πακέτων κατά την μετάδοση

- ▶ Όταν 2 σταθμοί που μοιράζονται το ίδιο μ.κ. μεταδώσουν στο ίδιο slot, τότε τα πακέτα (και τα 2) καταστρέφονται και παραμένουν στις ουρές των σταθμών ώστε να μεταδοθούν ξανά

```
System.out.println("Stations 1 & 2 are trying to transmit COLLISION");
```

# Μοντελοποίηση λειτουργίας προσομοιωτή

