



Περιεχόμενα

1	Εργασία	3
2	Σκοπός	3
3	Προγράμματα	3
4	Υλοποίηση	4
5	Αποτελέσματα	8
6	Συμπεράσματα	27

1 Εργασία

Κατανεμημένα Δικτυοκεντρικά Συστήματα
Απαλλακτική Εργασία
Έτος: 2019
Εξάμηνο: Εαρινό
Διδάσκοντες : Κ.Οικονόμου,Ε.Χριστοπούλου

Φοιτητής : Φάμπιαν Χαίγκερ
Έτος Φοίτησης: 3ό
ΑΜ: 2016187

2 Σκοπός

Σκοπός της εργασίας είναι χρησιμοποιώντας το OMNeT++ να δημιουργήσουμε ένα τυχαίο γεωμετρικό γράφο εκατό κόμβων και μεταβλητής ακτίνας συνδεσιμότητας rc . Στον γράφο αυτό πρέπει για δέκα διαφορετικές τιμές της rc , μέσω της αποστολής και λήψης μηνυμάτων μεταξύ των κόμβων να γίνει εύρεση του κόμβου με τον μεγαλύτερο αριθμό γειτόνων ο οποίος να καθιερωθεί ως αρχηγός(sink) και στη συνέχεια να ξεκινήσουν να στέλνουν όλοι οι κόμβοι μηνύματα προς αυτόν.

Κάθε χρονική στιγμή κάθε κόμβος συγκρίνει μια τυχαία μεταβλητή με το $\lambda(0 \leq \lambda \leq 1)$ και αν είναι μικρότερη δημιουργεί ένα νέο μήνυμα το οποίο καταχωρεί σε μια ουρά. Όποιο μήνυμα δεχθεί με προορισμό τον sink επίσης προστίθεται στην ουρά. Κάθε κόμβος σε κάθε χρονική στιγμή μπορεί προωθεί ένα μήνυμα από την ουρά.

Η διαδικασία αυτή συνεχίζεται μέχρι περάσουν ένα εκατομμύριο χρονικές στιγμές και επαναλαμβάνεται για δέκα διαφορετικές τιμές του λ για κάθε rc . Ο λόγος που κάνουμε αυτή την διαδικασία είναι για να δείξουμε ένα τρόπο συλλογής πληροφοριών σε ένα συγκεκριμένο κόμβο σε κατανεμημένο δίκτυο ασύρματων αισθητήρων.

3 Προγράμματα

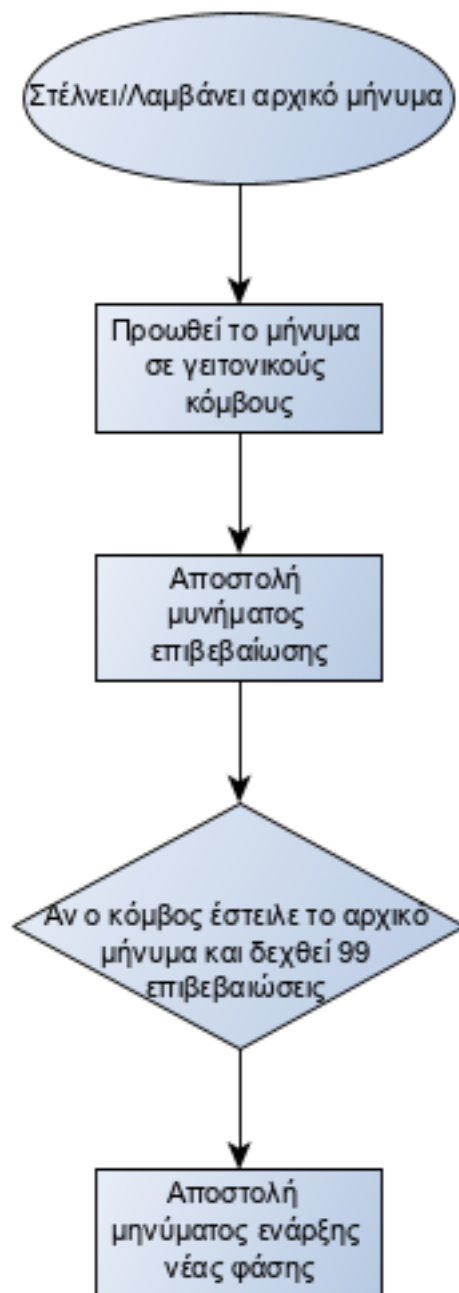
Για την εργασία χρησιμοποιήθηκε το omnet++ 5.4.1. Για την μεταφορά πληροφοριών (πχ. id,αριθμός γειτόνων ,θέση στο δίκτυο) χρησιμοποιήθηκαν μηνύματα κλάσης cPackage το οποίο αρχικοποιείται σε αρχείο τύπου (.msg). Για την απεικόνιση των πληροφοριών που συλλέξαμε χρησιμοποιήθηκε το gnuplot 5.2.

4 Υλοποίηση

Για την προώθηση των μηνυμάτων χρησιμοποιήθηκε ο αλγόριθμος πλημμυρίδας (flooding/broadcasting) με κάποιους ελέγχους για την βελτιστοποίηση του συστήματος. Σε πρώτη φάση ελέγχουμε ότι το δίκτυο είναι συνεκτικό, δηλαδή ότι όλοι οι κόμβοι είναι δυνατό να στείλουν μήνυμα έμμεσα ή άμεσα προς του άλλους. Αρχικά ο κόμβος με id ίσο με ένα(1) στέλνει ένα μήνυμα πλημμυρίδας προς τους γειτονικούς του κόμβους και εκείνοι το προωθούν περαιτέρω. Κάθε κόμβος που λαμβάνει το μήνυμα στέλνει ένα μήνυμα επιβεβαίωσης. Αν ο κόμβος με id ίσο με 1 λάβει ενενήντα εννιά μηνύματα επιβεβαίωσης σημαίνει ότι το δίκτυο είναι συνεκτικό και ξαναστέλνει μήνυμα πλημμυρίδας για την έναρξη της επόμενης φάσης.

Αλγόριθμος Γ'.1 Flooding

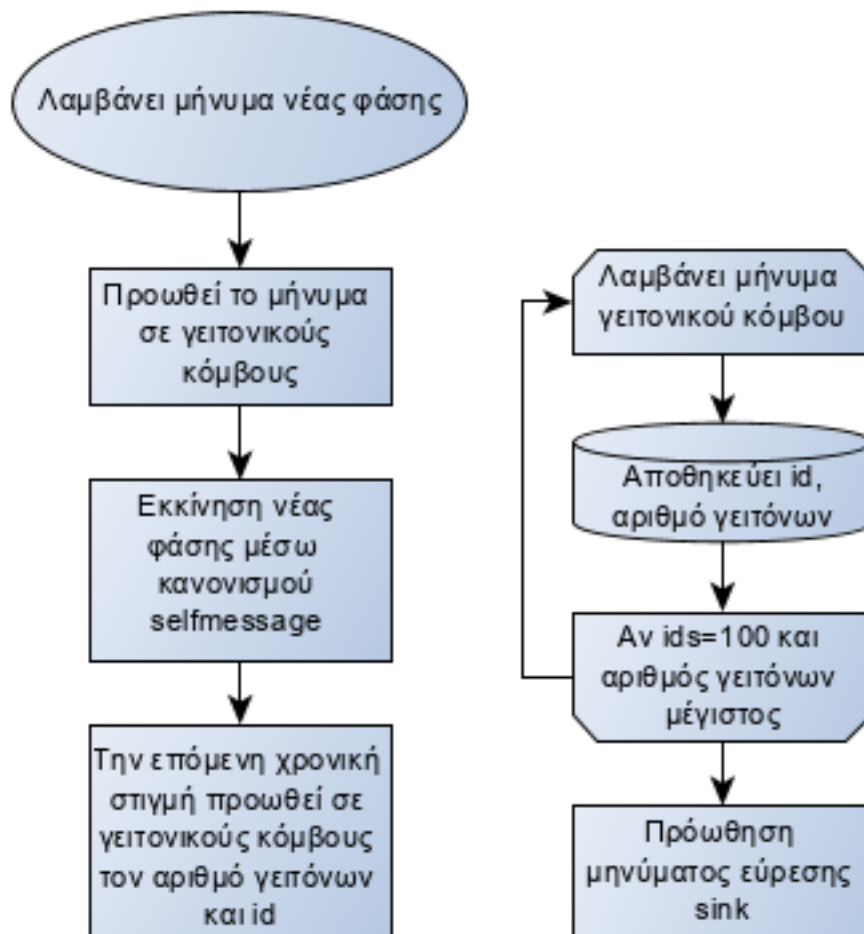
```
1: Objective: Ζητούμενο Γ'.1;  
2: Environment:  $A, \mathcal{B}$ ;  
3: Status:  $State = \{INITIATE, DONE\}$ ;  $State_0 = \{INITIATE\}$ ;  $State_T = \{DONE\}$ ;  
4: Consts: node  $u$ ; SetOfNodes  $S_u$ ;  
5:   string  $msg$ ; ▷ Η προς μετάδοση πληροφορία  
6: Vars:  $State\ state_u = INITIATE$ ; node  $v, z$ ;  
7: Events: WakeUp; receive< $msg, v$ >;  
  
8:  $INITIATE \leftarrow Wakeup$  ▷ Πρόκειται για το γεγονός που σκανδαλίζει (μόνο) τον κόμ-  
   βο που έχει την αρχική πληροφορία προς μετάδοση  
9:   for  $\forall v \in S_u$  do  
10:     send< $v:msg, u$ >;  
11:    $state_u = DONE$ ;  
  
12:  $INITIATE \leftarrow receive<msg, z>$  ▷ Λαμβάνεται μήνυμα από τους υπόλοιπους κόμβους  
   (πλην του αρχικού που προκάλεσε την πλημμυρίδα)  
  
13:   for  $\forall v \in S_u \setminus \{z\}$  do Λαμβάνεται μέριμνα να μην προωθηθεί το μήνυμα ξανά  
   ▷ προς τον κόμβο  $z$  ο οποίος το απέστειλε στον προκείμενο  
   κόμβο  $u$   
14:     send< $v:msg, u$ >;  
15:    $state_u = DONE$ ;
```



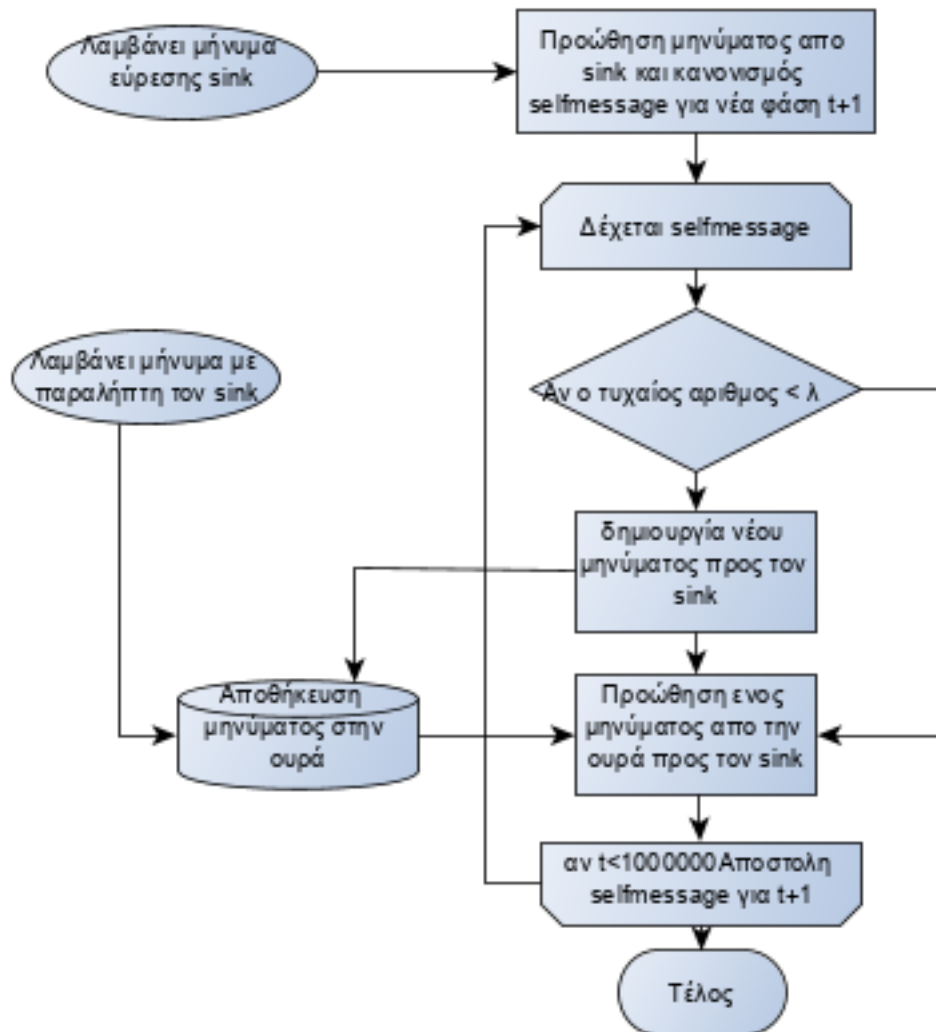
Κάθε κόμβος που λαμβάνει το δεύτερο μήνυμα πλημμυρίδας κανονίζει ένα σελφμεσάγε για την επόμενη χρονική στιγμή και προωθεί το μήνυμα στους γειτονικούς του κόμβους. Όταν έρθει η επόμενη χρονική στιγμή ξεκινά η διαδικασία εύρεσης του αρχηγού.

Τότε όλοι στέλνουν ένα μήνυμα προς κάθε γείτονα τους με το id και τον αριθμό γειτόνων. Το id χρησιμοποιείται σαν δεύτερο κριτήριο στην εύρεση αρχηγού, σε περίπτωση ίσου αριθμού γειτόνων επιλέγεται εκείνος με μικρότερο id.

Όταν λαμβάνονται τα μηνύματα αυτά αποθηκεύονται οι πληροφορίες και προωθούνται. Αν ένας κόμβος έχει λάβει όλα τα ids και τηρεί τις προϋποθέσεις επικυρώνει τον εαυτό του σαν αρχηγό στέλνοντας μήνυμα στους υπόλοιπους.



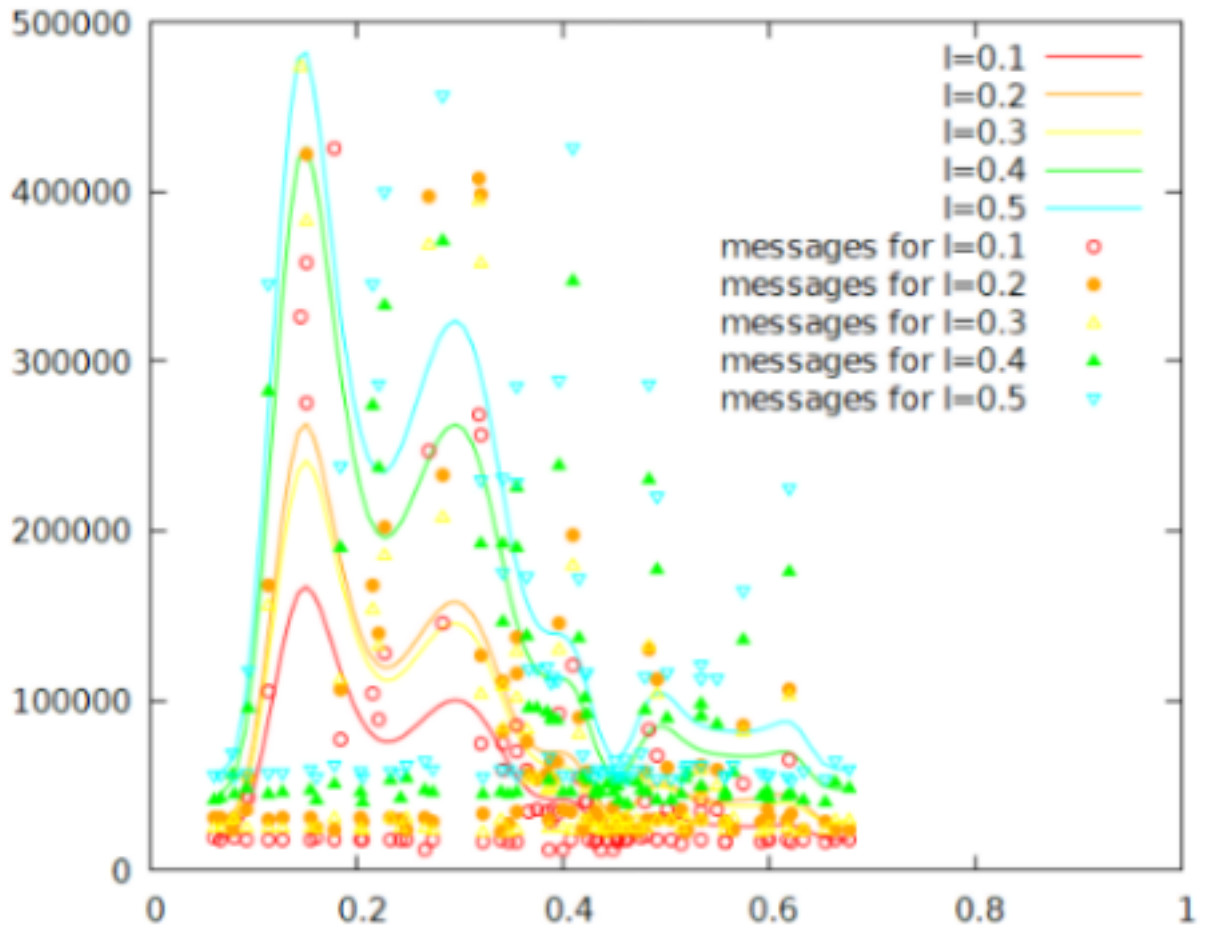
Στην τελευταία και μεγαλύτερη φάση , μετα τη λήψη του μηνύματος εύρεσης sink με ένα μήνυμα σελφμεσσαζε την επόμενη χρονική στιγμή ξεκινά οι προώθηση μηνυμάτων προς αυτον . Είναι σημαντικό να αναφέρουμε ότι το “μονοπάτι” από κάθε κόμβο στον sink έχει βρεθεί αποθηκεύοντας τη θύρα προέλευσης του μηνύματος εύρεσης sink .Κάθε χρονική στιγμή προωθείται ένα μήνυμα από την ουρά. Οι πληροφορίες που συλλέγουμε για κάθε εκτέλεση είναι ο αριθμός μηνυμάτων που προώθησε και η απόσταση του απο τον κόμβο sink.

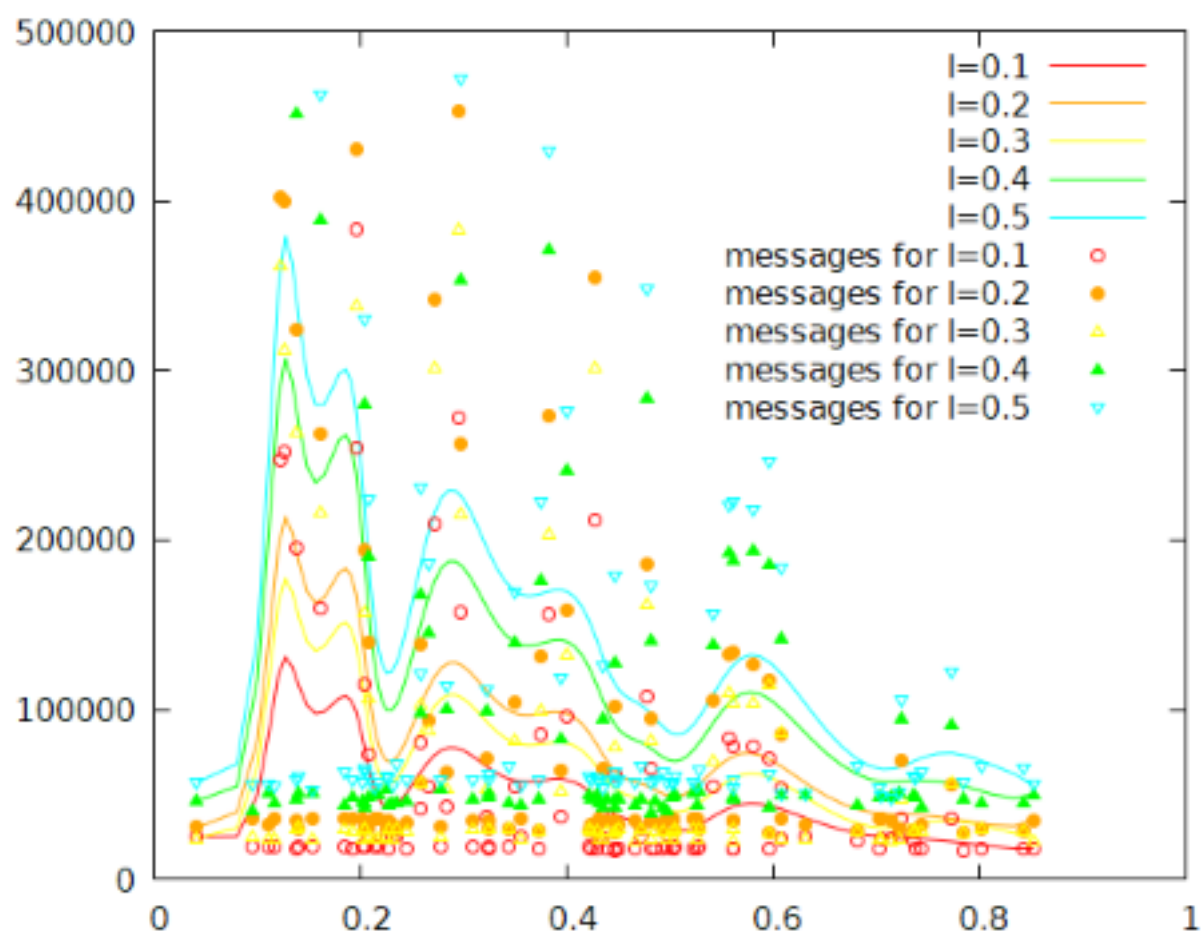


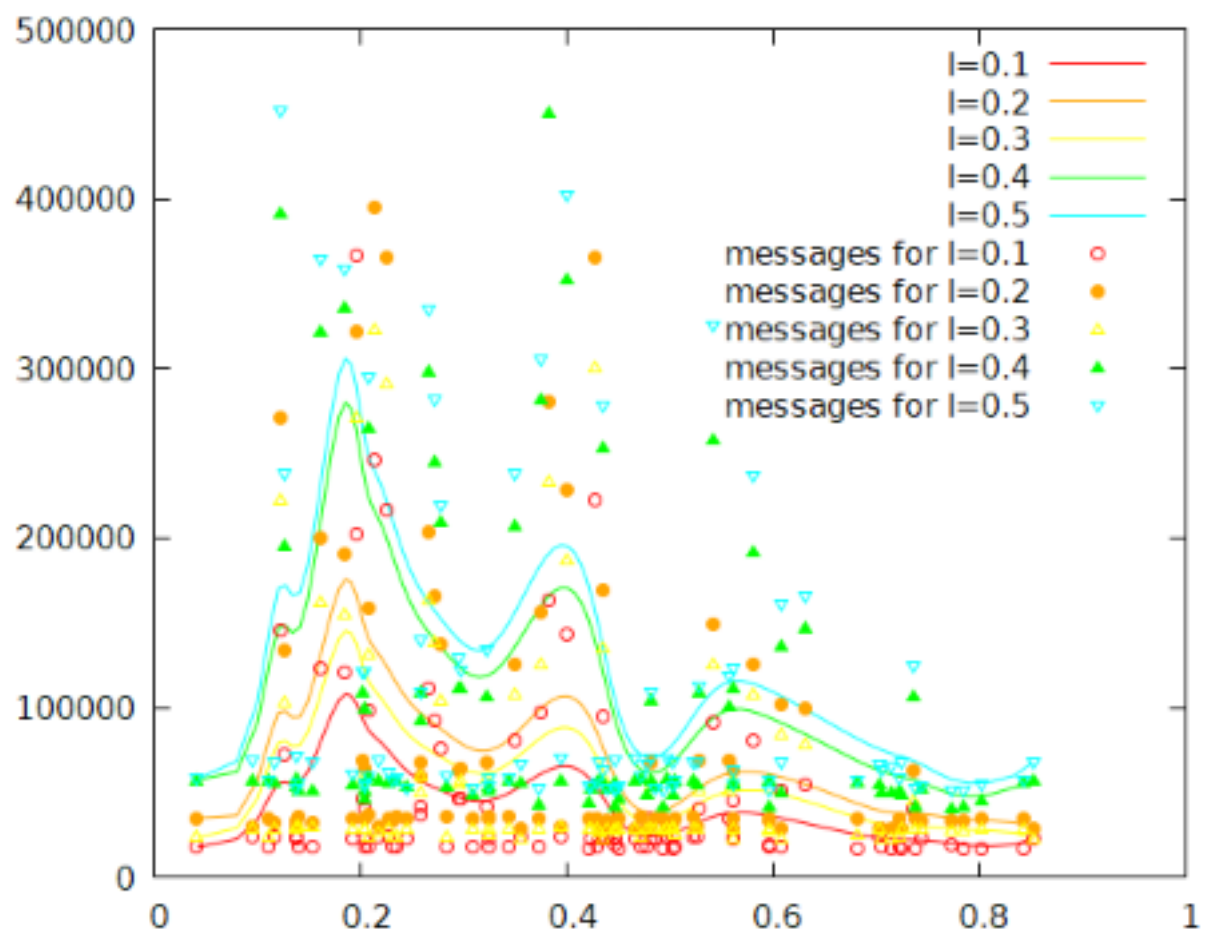
5 Αποτελέσματα

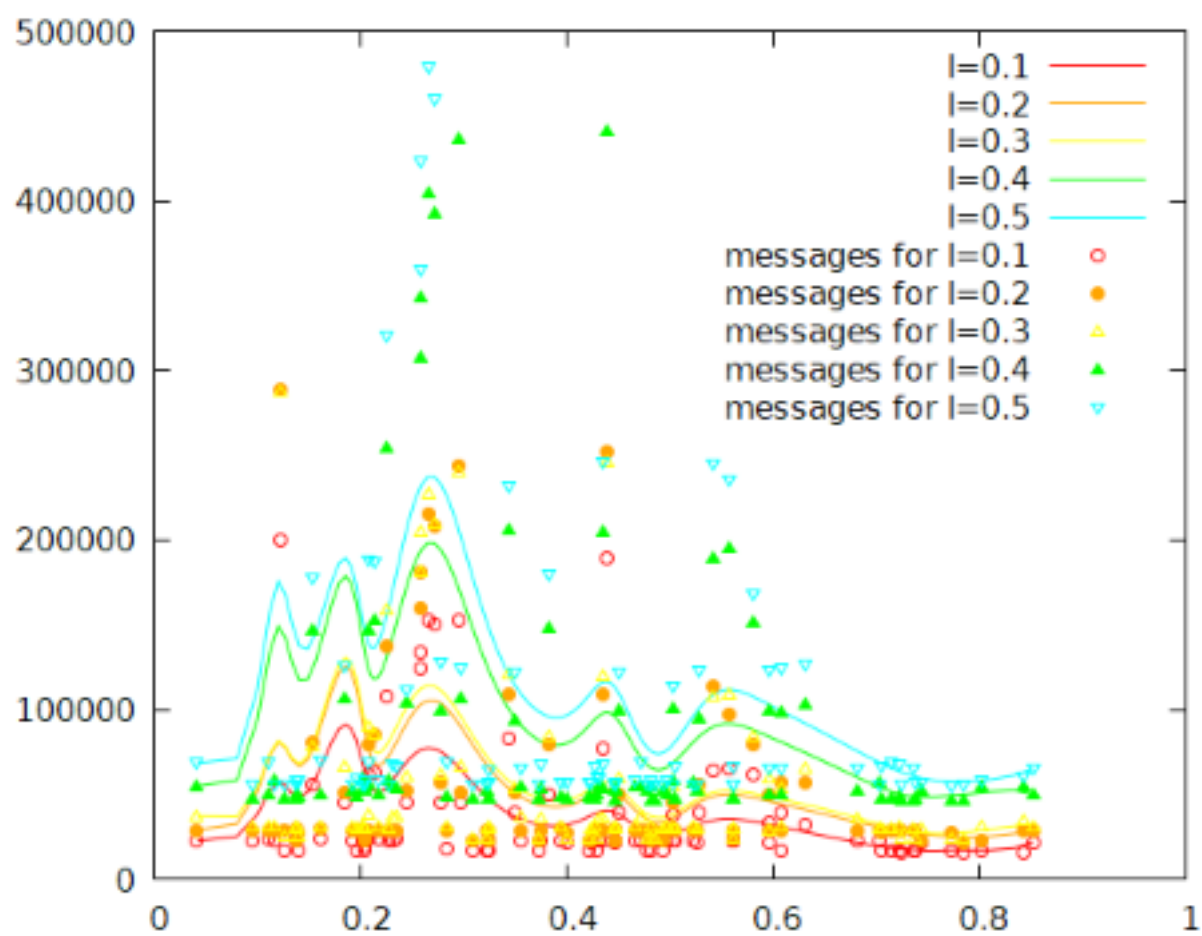
Παρακάτω βρίσκεται ο πίνακας με τα rc και τα λ που επιλέχθηκαν. Για την επιλογή των rc χρησιμοποιήθηκε σαν κριτήριο ο μέσος αριθμός γειτόνων.

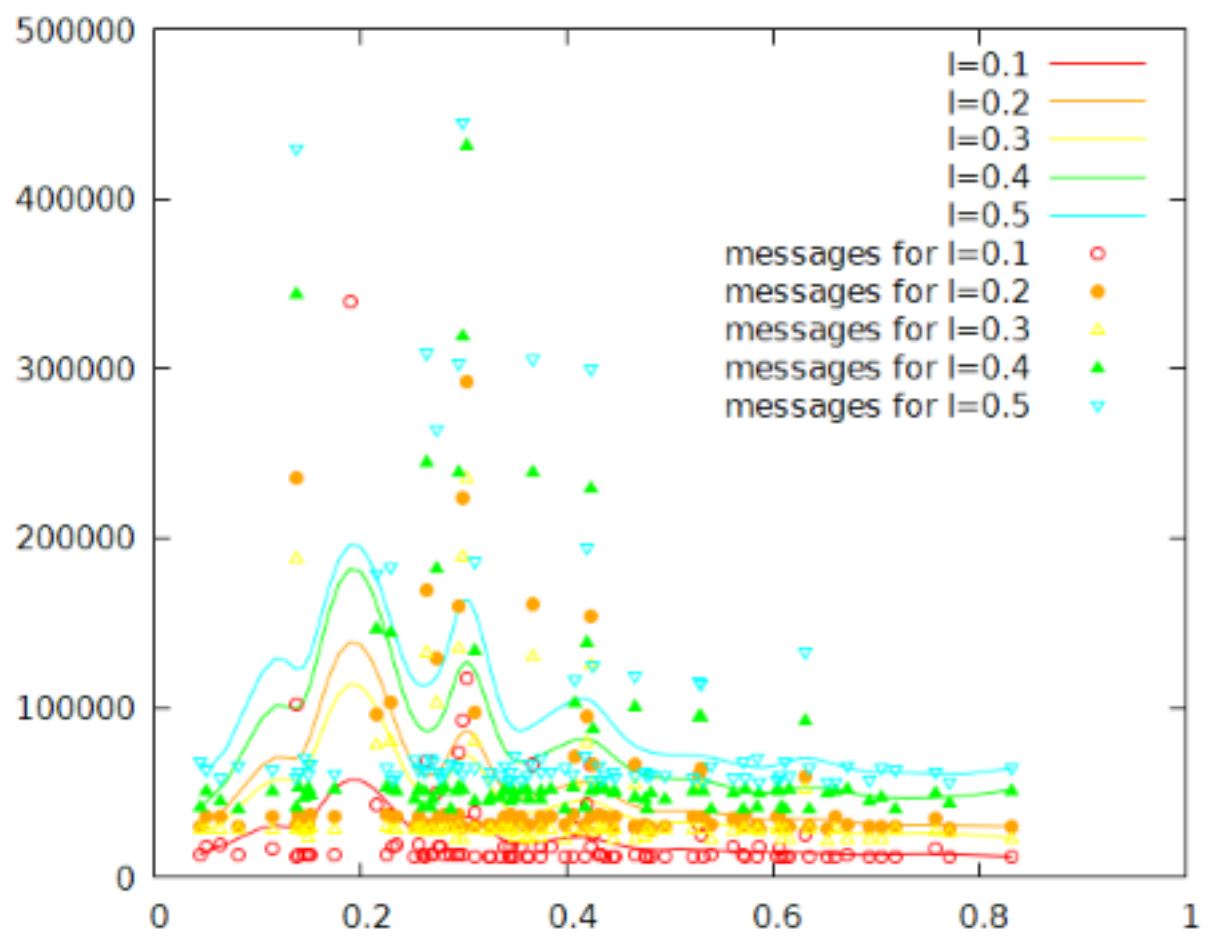
	rc		λ
1	0.1851	9	1
2	0.21	11	2
3	0.23	13	3
4	0.27	17	4
5	0.31	22	5
6	0.35	28	6
7	0.39	33	7
8	0.45	42	8
9	0.50	49	9
10	0.60	62	10

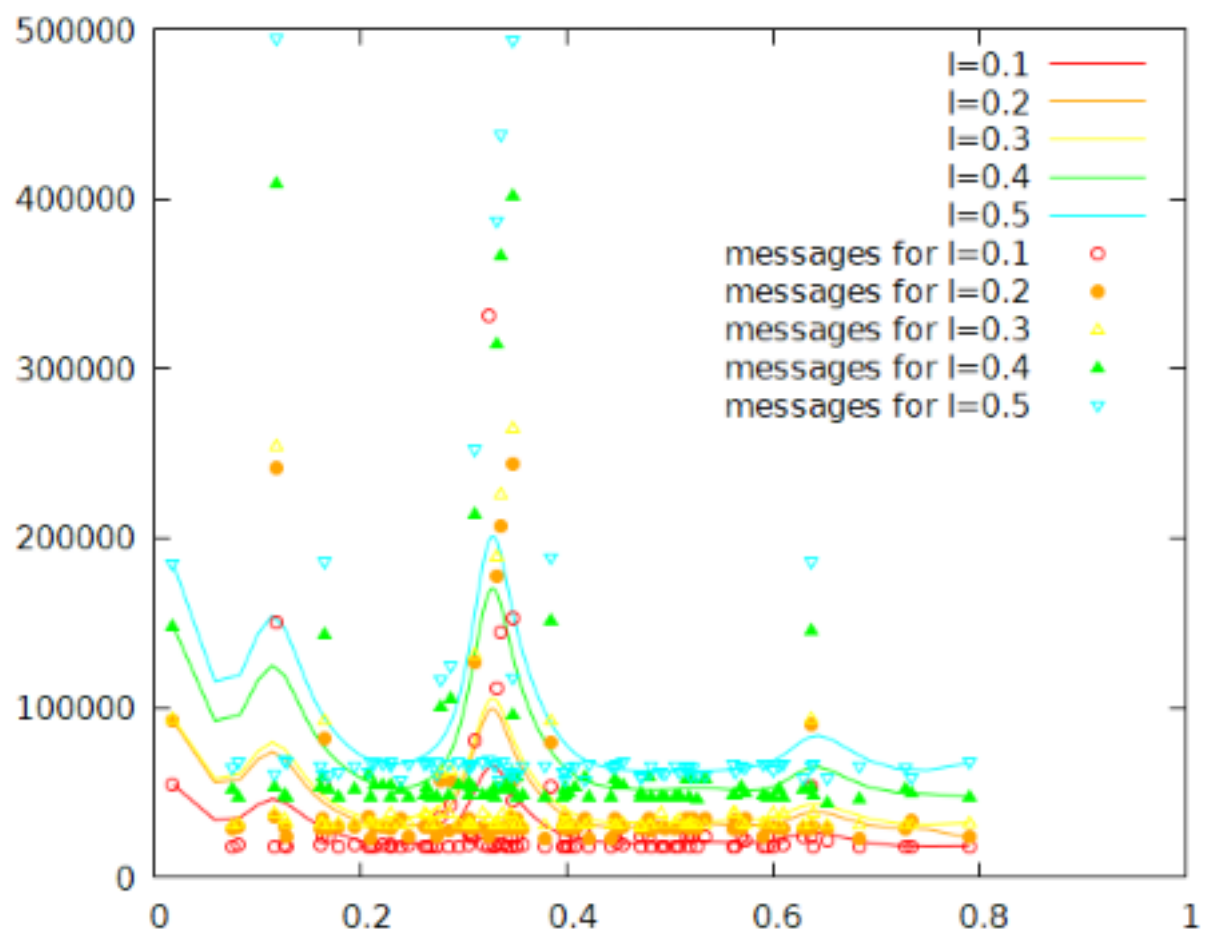


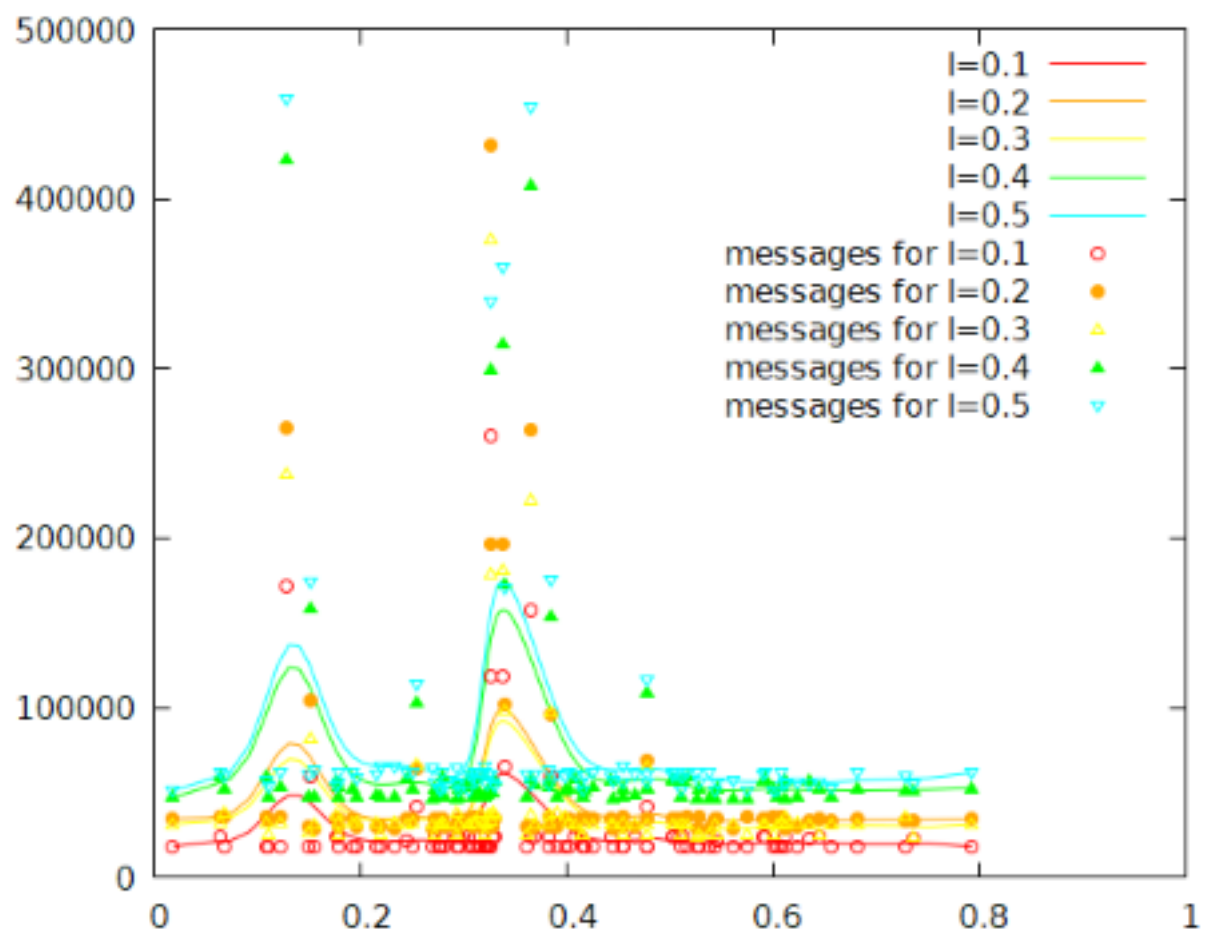


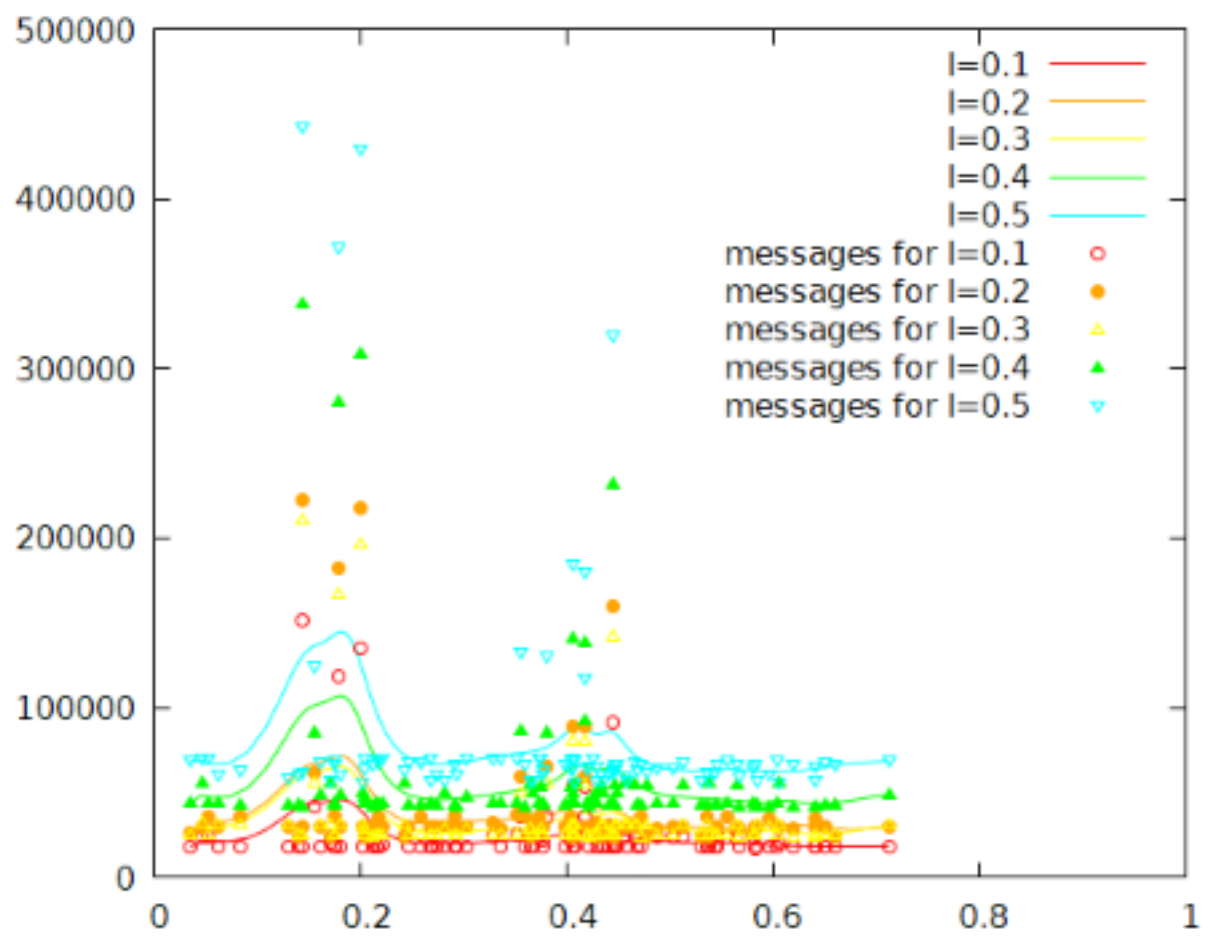


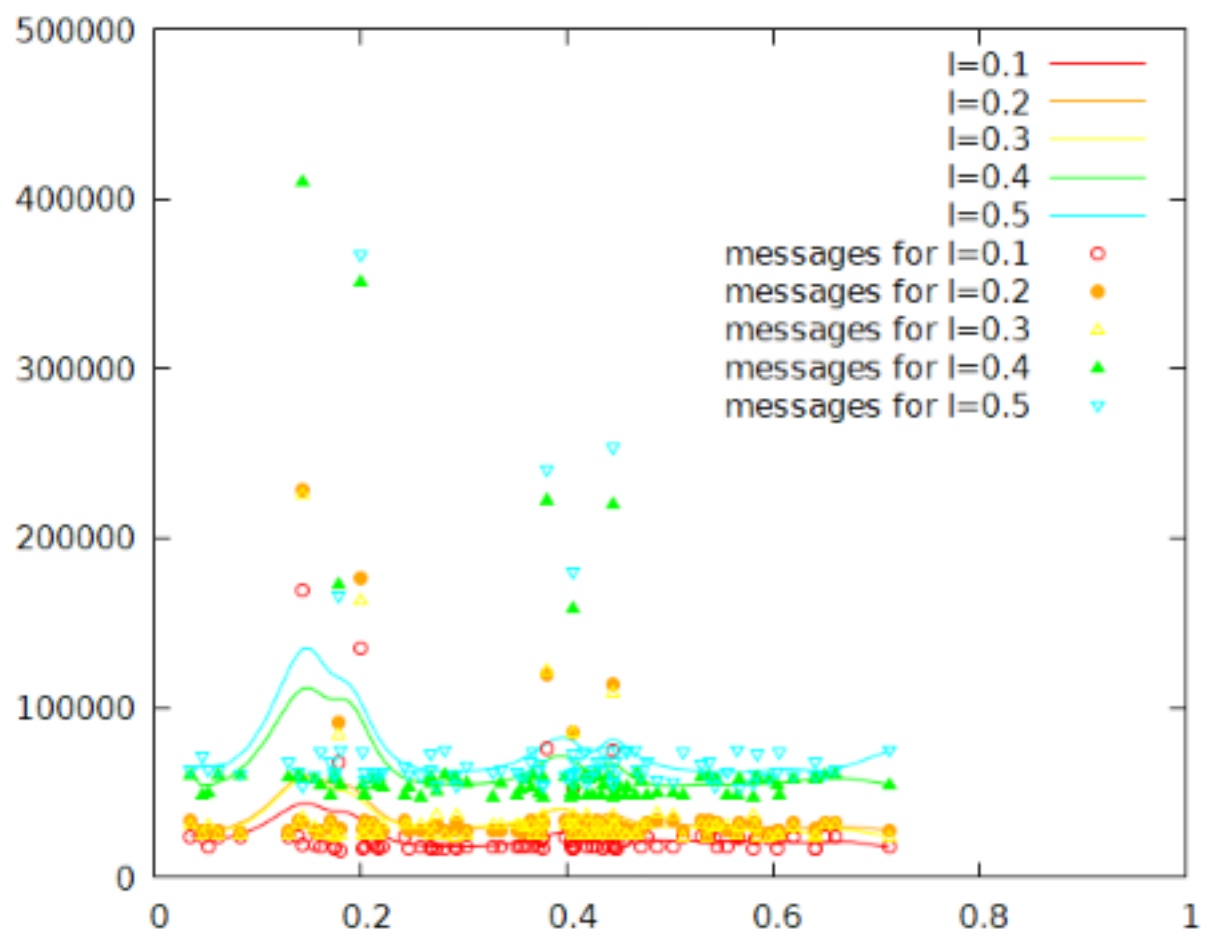


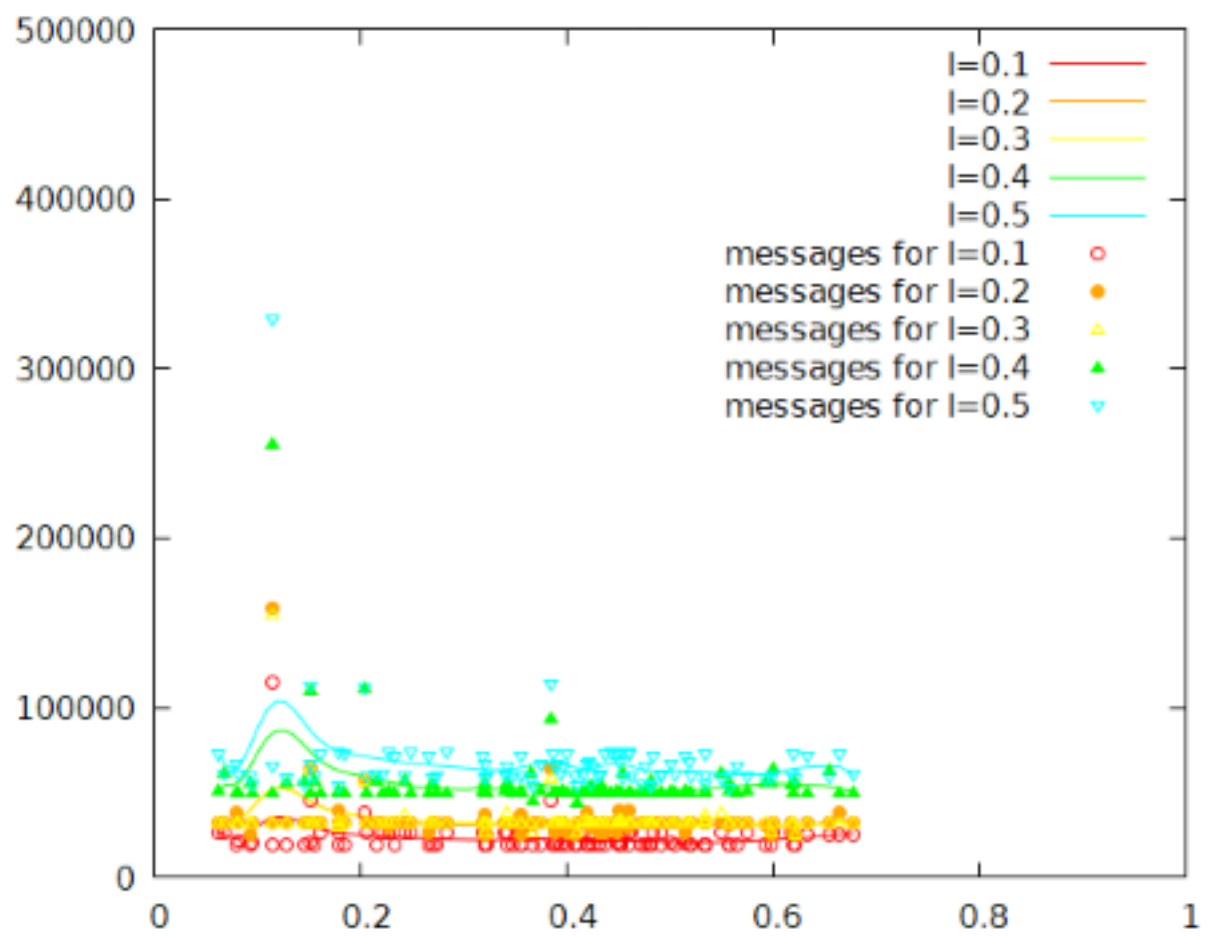


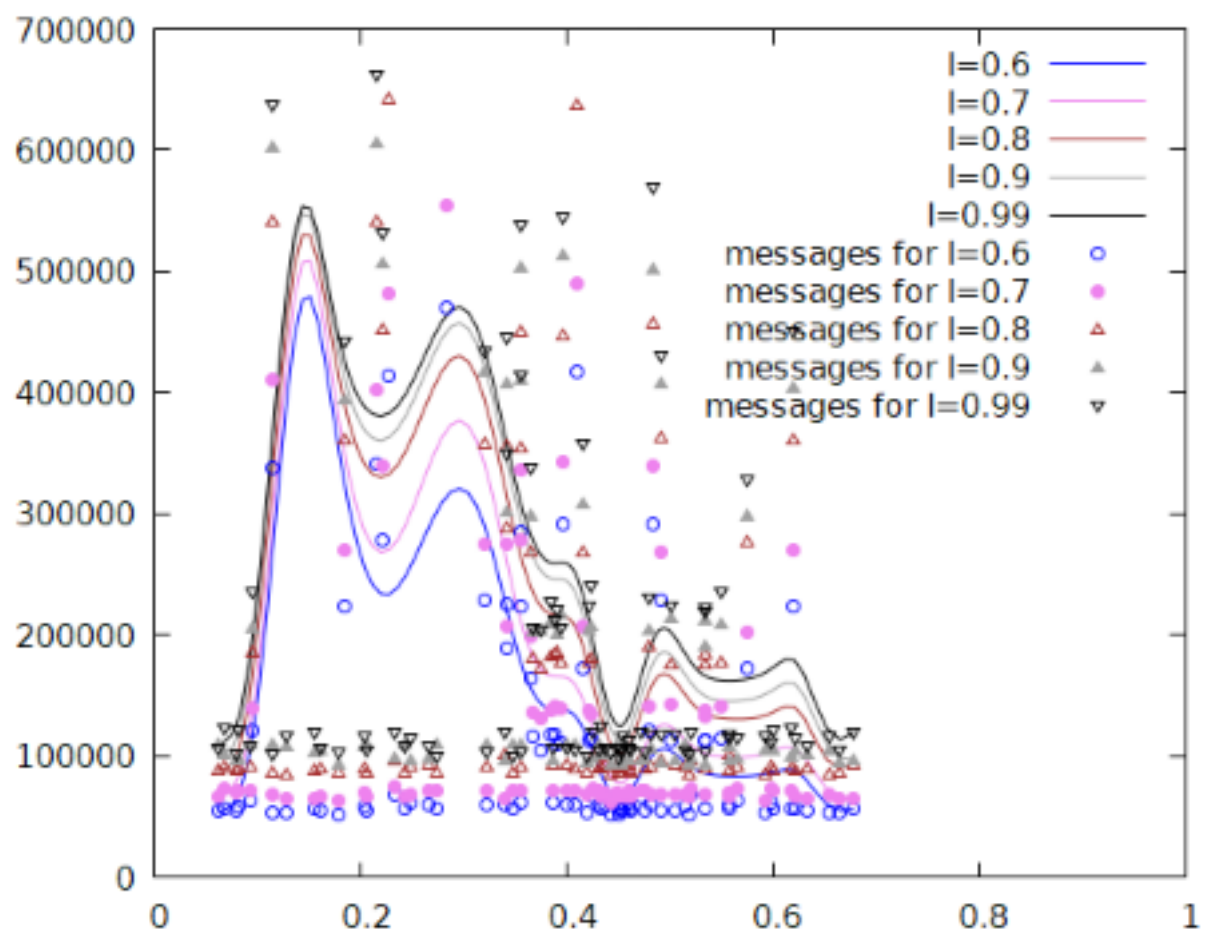


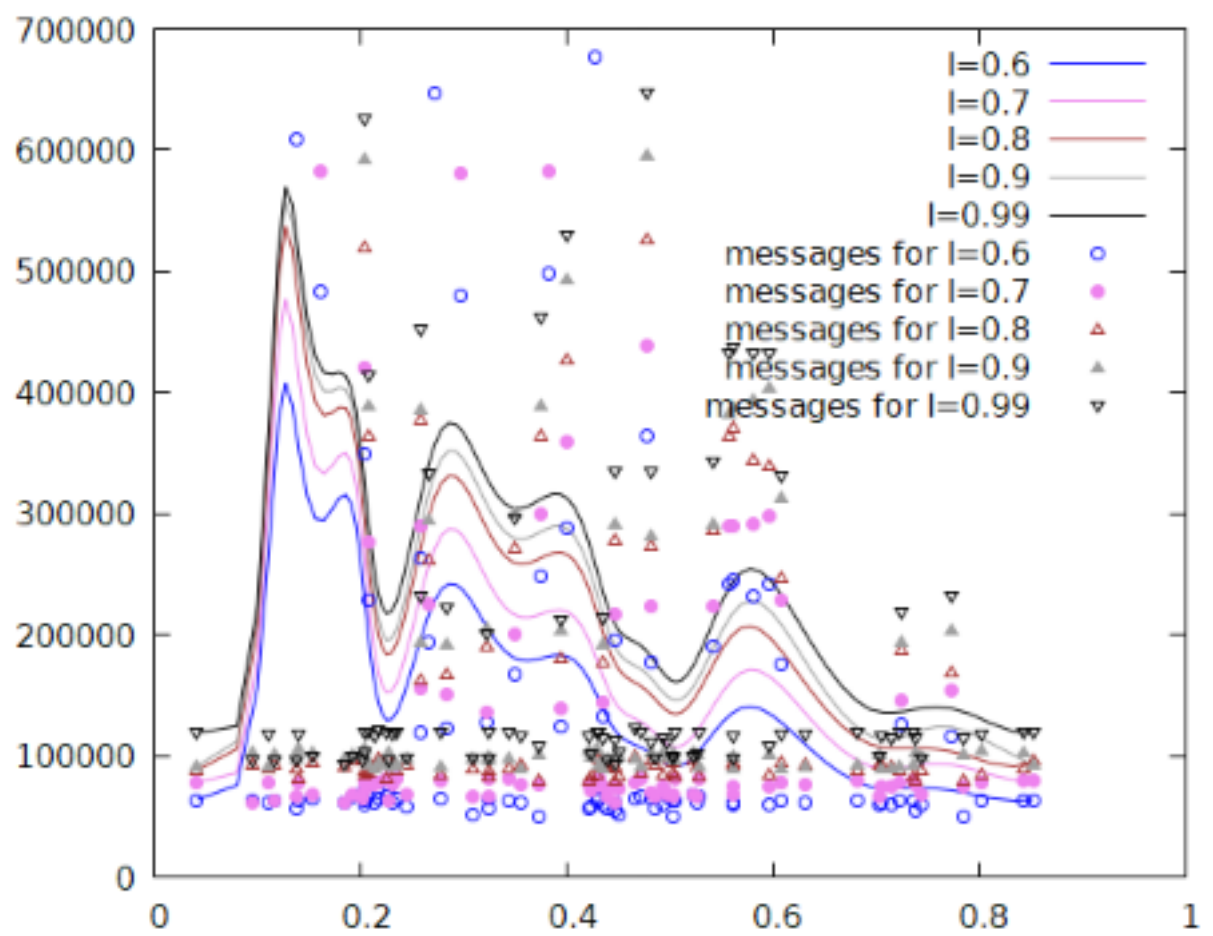


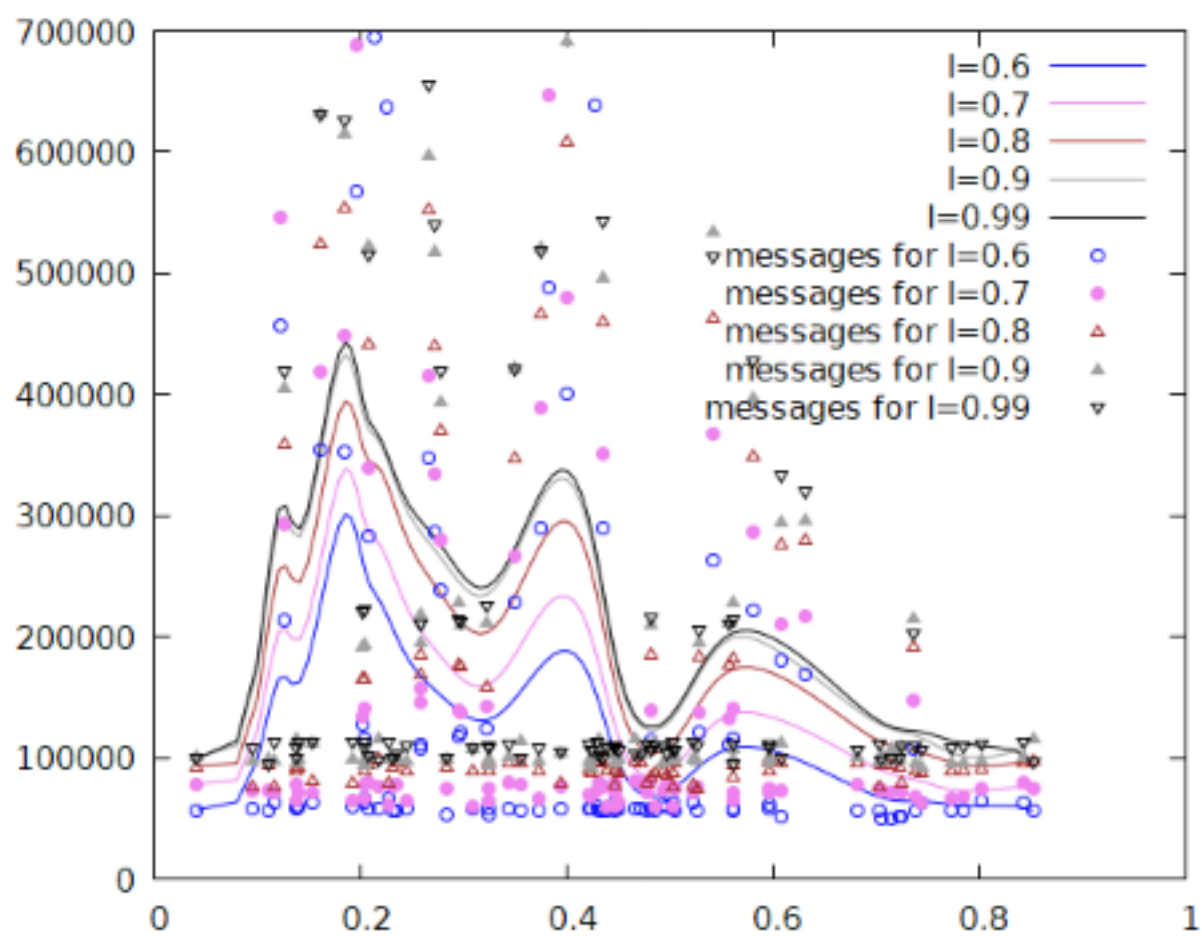


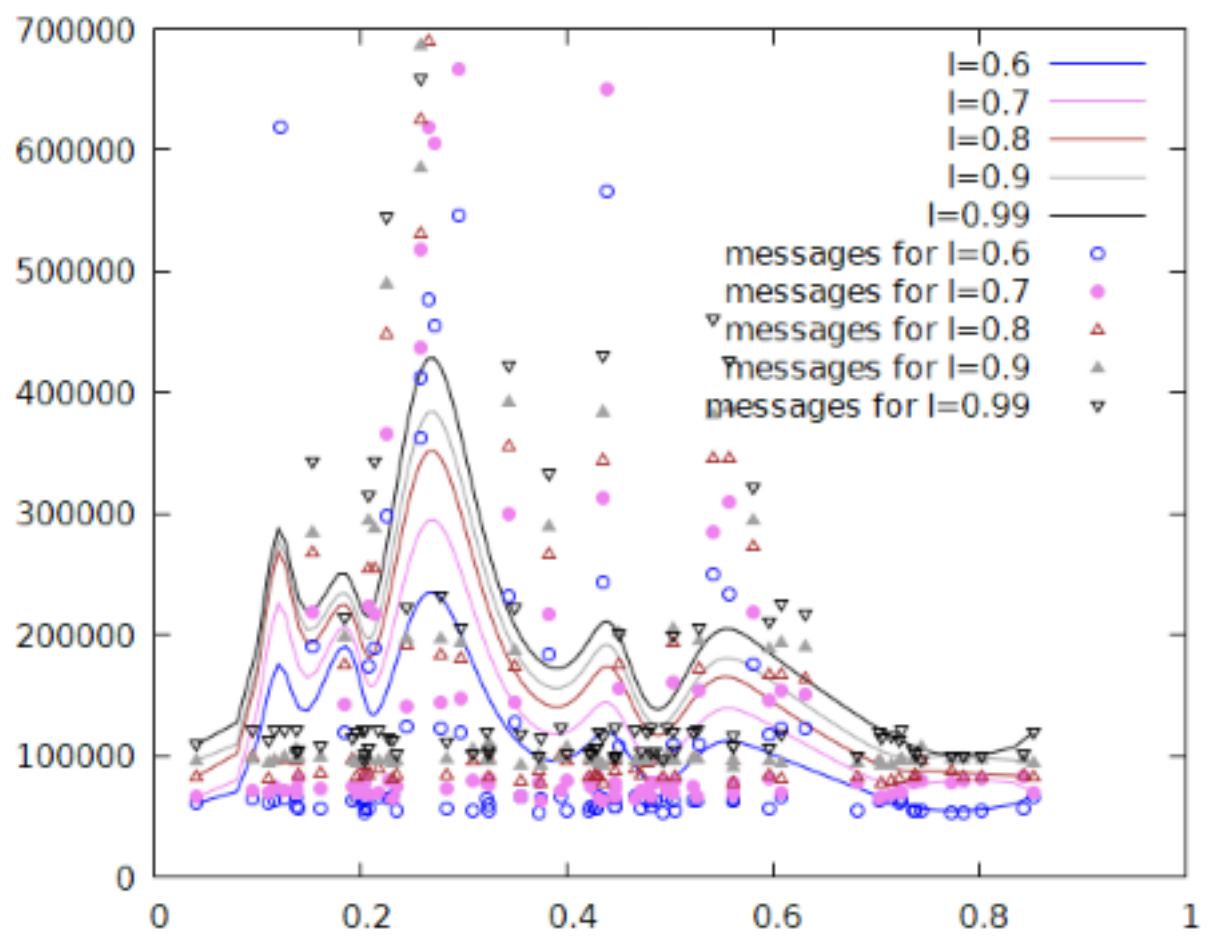


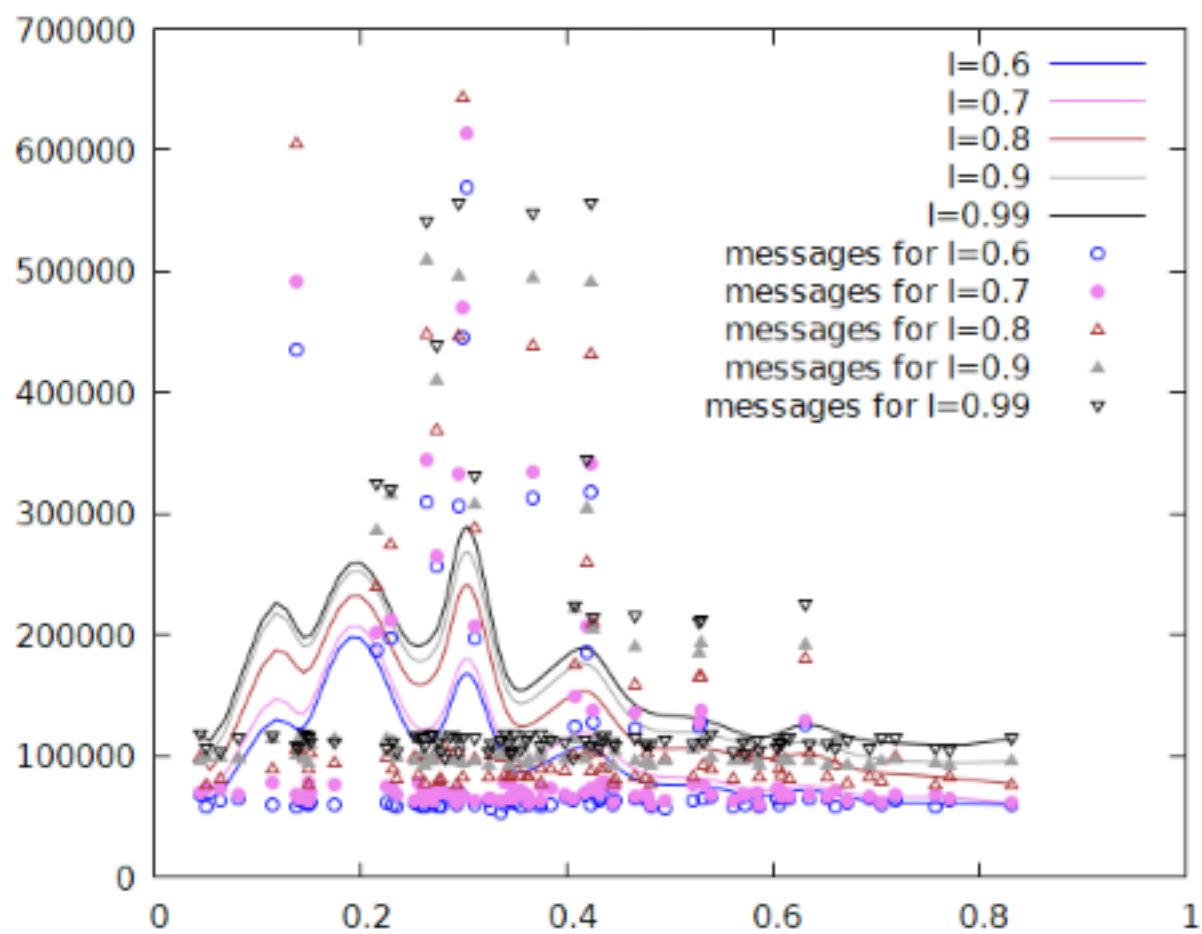


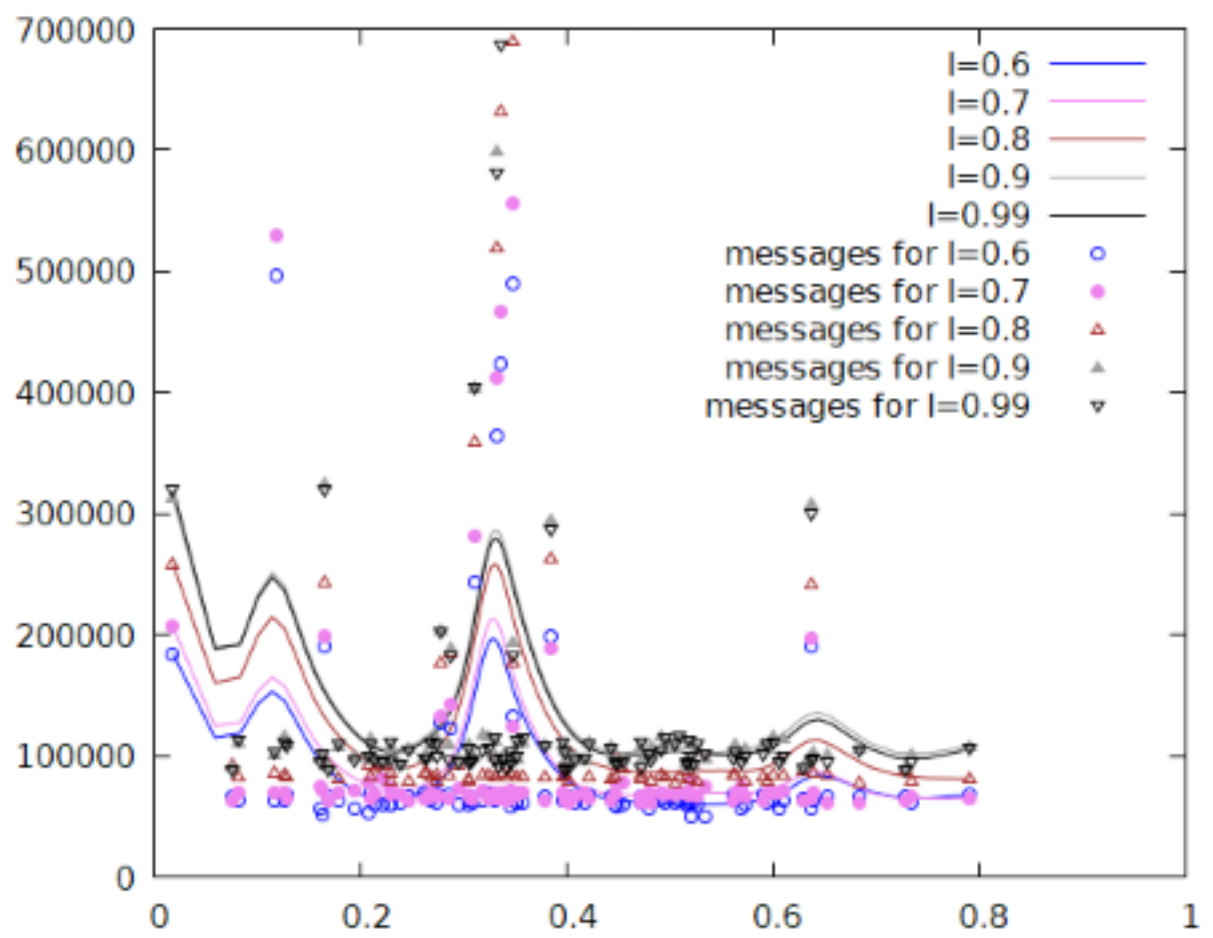


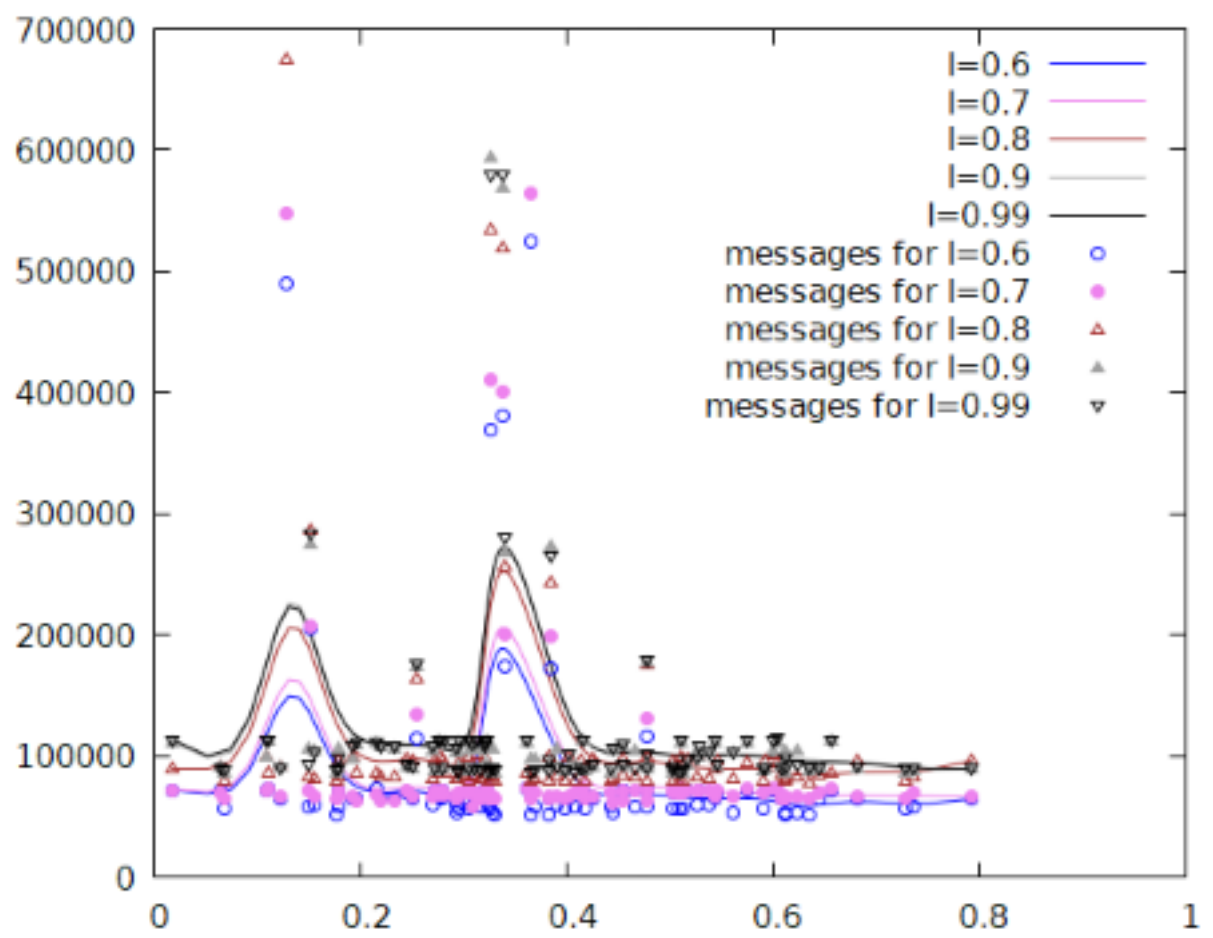


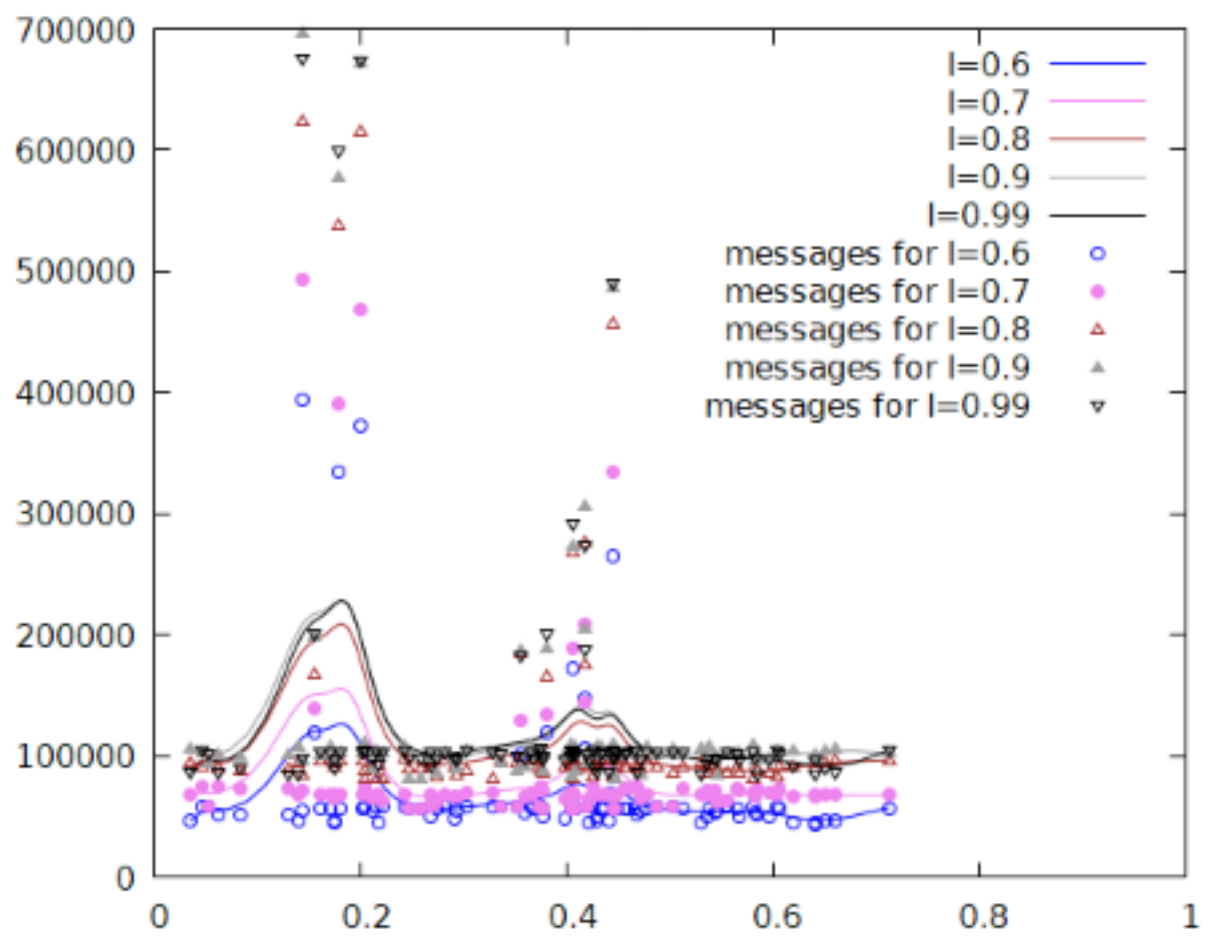


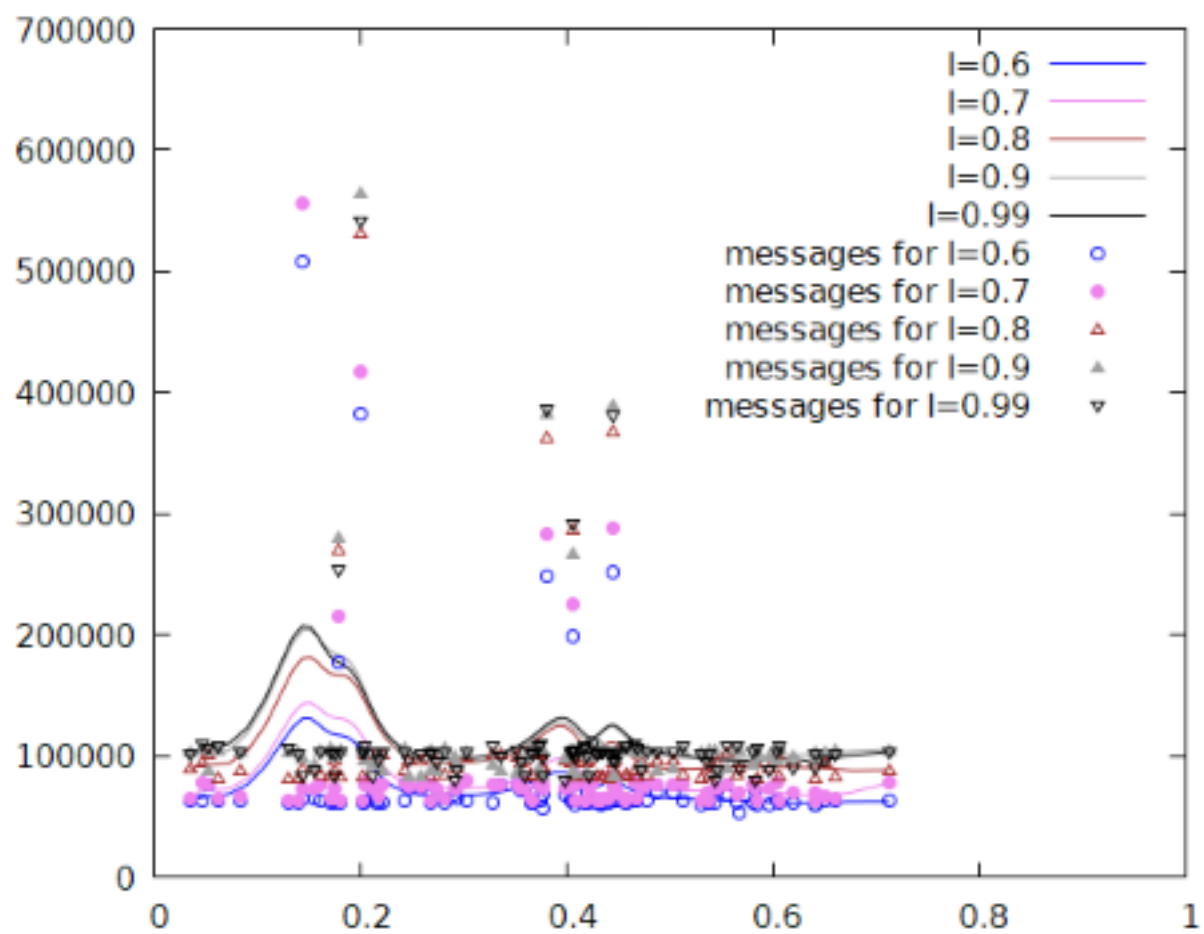


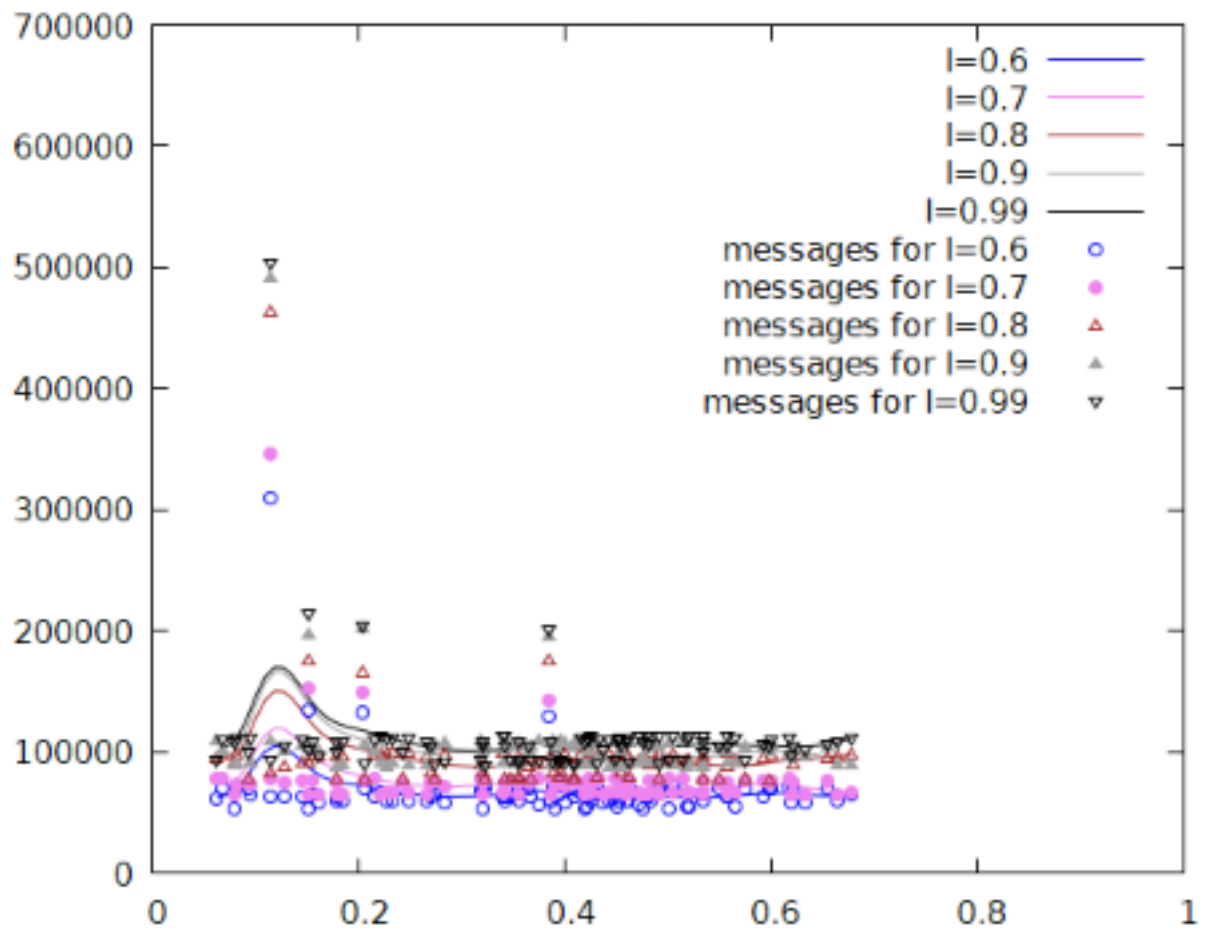










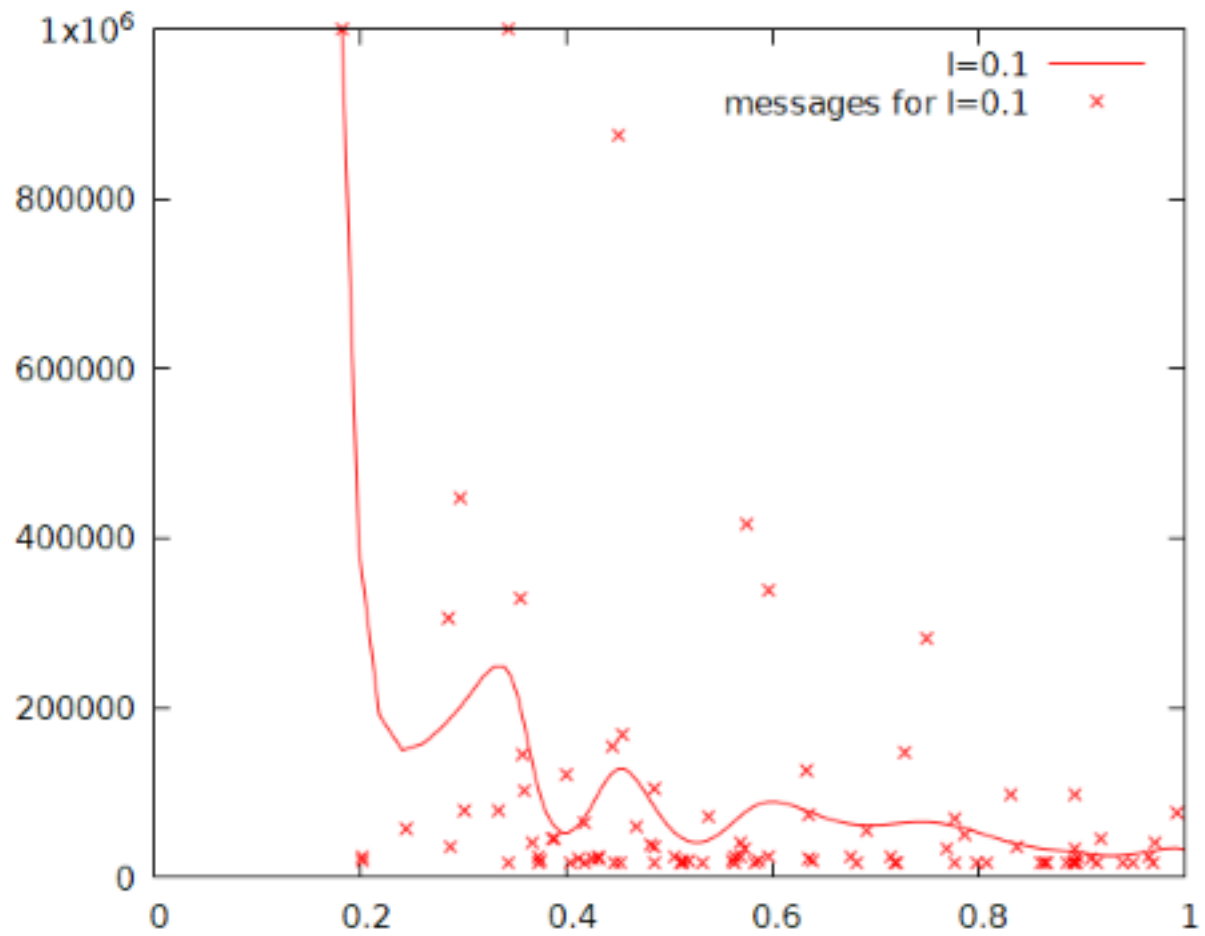


6 Συμπεράσματα

Όπως βλέπουμε, όσο το δίκτυο έχει πολύ μικρό rc ο όγκος των μηνυμάτων συγκεντρώνεται σε μερικούς κοντινότερους κόμβους του κόμβου sink και όσο η απόσταση από τον sink αυξάνεται ο αριθμός μηνυμάτων μειώνεται εκθετικά έως ένα σημείο όπου η συσχέτιση απόστασης με αριθμό μηνυμάτων που προωθούνται να μην είναι πλέον ισχυρή.

Με την αύξηση του rc η συσχέτιση απόστασης με αριθμό μηνυμάτων που προωθούνται γίνεται όλο και πιο ομαλή και ο όγκος των μηνυμάτων μοιράζεται σε περισσότερους κόμβους. Συγκεκριμένα για rc ίσο με 0.6 τα ο αριθμός μηνυμάτων που προωθεί ο κάθε κόμβος είναι παρόμοιος. Κάτι τέτοιο είναι λογικό γιατί πλέον κάθε κόμβος έχει μεγάλο αριθμό γειτόνων και ο sink επίσης δέχεται μηνύματα άμεσα από μεγάλο αριθμό κόμβων.

Με μια αλλαγή στον κώδικα επιλέγεται ο κόμβος με τα μικρότερο αριθμό γειτόνων ως sink.



Όπως ήταν αναμενόμενο για sink ίσο με 0.1851(ελάχιστο δυνατό για συνεκτικό δίκτυο) ο αριθμός μηνυμάτων του κοντινότερου κόμβου αυξάνεται ραγδαία καθώς δεν υπάρχει άλλο "μονοπάτι" προς τον sink.