



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

**Δυναμική Ανάθεση Εύρους Ζώνης
σε Παθητικά Οπτικά Δίκτυα**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μιχαήλ Γ. Χειμώνας

Επιβλέπων : Ιάκωβος Βενιέρης

Καθηγητής Ε.Μ.Π.

Αθήνα, Μάιος 2011



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

**Δυναμική Ανάθεση Εύρους Ζώνης
σε Παθητικά Οπτικά Δίκτυα**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μιχαήλ Γ. Χειμώνας

Επιβλέπων : Ιάκωβος Βενιέρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 31^η Μαΐου 2011

.....
Ι. Στ. Βενιέρης
Καθηγητής Ε.Μ.Π.

.....
Δ.-Θ. Ι. Κακλαμάνη
Καθηγήτρια Ε.Μ.Π.

.....
Ν. Κ. Ουζούνoglou
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάιος 2011

.....
Μιχαήλ Γ. Χειμώνας

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Μιχαήλ Γ. Χειμώνας 2011

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Η διαρκής αύξηση των αναγκών για εύρος ζώνης σε παγκόσμιο επίπεδο έχει οδηγήσει την τελευταία δεκαετία στην εγκατάλειψη του χαλκού και τη σταδιακή μετάβαση σε δίκτυα πρόσβασης οπτικών ινών. Από τις διαθέσιμες τεχνολογίες οπτικών δικτύων πρόσβασης, τα παθητικά οπτικά δίκτυα εμφανίζουν σαφή πλεονεκτήματα, γεγονός που αποδεικνύεται από τον υψηλό βαθμό διείσδυσής τους στη πλειοψηφία των αγορών. Το πρωτόκολλο G-PON, προτυποποιημένο από την ITU-T στις συστάσεις G.984.x, προσφέρει αυξημένη απόδοση και εγγυήσεις ποιότητας για ένα ευρύ φάσμα υποστηριζόμενων υπηρεσιών.

Η ανάθεση του εύρους ζώνης στα δίκτυα G-PON πραγματοποιείται με δυναμικό τρόπο, ωστόσο ο αλγόριθμος δυναμικής ανάθεσης εύρους ζώνης δεν προσδιορίζεται στις συστάσεις της ITU-T και αφήνεται στην κρίση του κατασκευαστή του δικτυακού εξοπλισμού. Η επίδραση του αλγορίθμου δυναμικής ανάθεσης εύρους ζώνης που χρησιμοποιείται στην υλοποίηση ενός δικτύου G-PON είναι κρίσιμη για την επίτευξη αποδοτικής και δίκαιης χρησιμοποίησης του εύρους ζώνης με έμφαση στη ροή ανόδου και παράλληλα τη δυνατότητα υποστήριξης των απαιτήσεων ποιότητας υπηρεσίας διάφορων εφαρμογών. Στο πλαίσιο της παρούσας διπλωματικής εργασίας μοντελοποιήθηκε το πρωτόκολλο G-PON με χρήση του περιβάλλοντος προσομοίωσης OPNET και εκτελέστηκαν προσομοιώσεις με στόχο το προσδιορισμό του σημείου βέλτιστης λειτουργίας, εξετάζοντας το αποτέλεσμα εύρους ρυθμίσεων και αλγορίθμων δυναμικής ανάθεσης εύρους ζώνης.

Λέξεις Κλειδιά

Παθητικά Οπτικά Δίκτυα, Δυναμική Ανάθεση Εύρους ζώνης, Δίκτυα Πρόσβασης Οπτικών Ινών, Πρωτόκολλο G-PON, Έλεγχος Πρόσβασης στο Μέσο

Abstract

A global, constant rise in demand for bandwidth has led in the ultimate decade to the gradual transition from copper to optical access networks. Among the available optical access network technologies, Passive Optical Networks feature significant advantages, a fact proven by the high level of market penetration. The G-PON protocol, standardized by ITU-T in the G.984.x series of recommendations, offers increased performance and quality guarantees for a wide spectrum of supported services.

Dynamic Bandwidth Allocation in G-PON networks is performed in a dynamic fashion, however the Dynamic Bandwidth Allocation algorithm is not specified within the ITU-T recommendations and is left for the network vendors to implement. The effect of the DBA algorithm in a G-PON network implementation proves crucial, regarding effective and fair bandwidth utilization especially in the upstream direction. In the same time the DBA algorithm provides support for the Quality of Service requirements of various applications. Within the scope of this thesis, the G-PON protocol was modeled using the OPNET modeler tool and a series of simulations were executed, with the aim of defining the optimal function parameters and studying the effect of different DBA algorithms.

Keywords

Passive Optical Networks, Dynamic Bandwidth Allocation, Optical Access Networks, G-PON protocol, Medium Access Control

ΕΥΧΑΡΙΣΤΙΕΣ

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας, η οποία σηματοδοτεί το τέλος της φοίτησής μου στο Εθνικό Μετσόβιο Πολυτεχνείο και την έναρξη μιας νέας περιόδου, αισθάνομαι την ανάγκη να ευχαριστήσω τους ανθρώπους χωρίς την βοήθεια των οποίων δεν θα μπορούσα να φτάσω ως εδώ.

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της διπλωματικής εργασίας, Ιάκωβο Βενιέρη, για την ανάθεση της εργασίας, την ευκαιρία για μία πρώτη ερευνητική εμπειρία καθώς και τις γνώσεις που αποκόμισα κατά την εκπόνησή της. Η συγκεκριμένη εργασία δεν θα ήταν φυσικά δυνατό να ολοκληρωθεί χωρίς την πολύτιμη καθοδήγηση και βοήθεια που έλαβα από το διδάκτορα Ευάγγελο Κοσμάτο, τον οποίο αισθάνομαι την υποχρέωση να ευχαριστήσω θερμά για τη συμβολή του και την υπομονή που έδειξε απέναντι στα ερωτήματά μου.

Δεν θα μπορούσα να μην αναφέρω στο τέλος της διαδρομής αυτής, τους φίλους και συμφοιτητές μου, οι οποίοι στάθηκαν δίπλα μου καθ' όλη τη διάρκεια της φοίτησής μου και μοιράστηκαν τις καλές αλλά και τις άσχημες στιγμές: Μαρία, Γιώτα, Βαγγέλη, Νάνσυ, Γιώργο, Δημήτρη, Αλέξανδρο, Γιώργο, Βασιλική.

Τέλος, αλλά περισσότερο από κάθε άλλον, επιθυμώ να ευχαριστήσω τους γονείς μου Γιάννη και Ασπασία, καθώς και την αδερφή μου Σόνια, για την αγάπη και την αμέριστη στήριξη που μου προσέφεραν σε κάθε περίοδο της ζωής μου.

Μιχαήλ Γ. Χειμώνας

Μάιος 2011

“Be patient, for the world is broad and wide”

William Shakespeare

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	11
ΑΚΡΩΝΥΜΙΑ	13
ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ	15
ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ	17
ΚΕΦΑΛΑΙΟ 1^ο : ΕΙΣΑΓΩΓΗ	19
1.1 ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ ΚΑΙ Ο ΣΤΟΧΟΣ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ	20
1.2 Η ΔΟΜΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ	20
ΚΕΦΑΛΑΙΟ 2^ο: ΠΑΘΗΤΙΚΑ ΟΠΤΙΚΑ ΔΙΚΤΥΑ & ΠΡΟΔΙΑΓΡΑΦΕΣ ΛΕΙΤΟΥΡΓΙΑΣ ΠΡΩΤΟΚΟΛΛΟΥ G-PON	21
2.1 ΓΕΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΟΠΤΙΚΩΝ ΔΙΚΤΥΩΝ ΠΡΟΣΒΑΣΗΣ & ΔΙΚΤΥΩΝ G-PON.....	21
2.1.1 Αρχιτεκτονική οπτικών δικτύων πρόσβασης.....	21
2.1.2 Γενικά χαρακτηριστικά δικτύων G-PON	23
2.2 ΤΟ ΣΤΡΩΜΑ ΣΥΓΚΛΙΣΗΣ ΜΕΤΑΔΟΣΗΣ ΤΟΥ ΔΙΚΤΥΟΥ G-PON	24
2.2.1 Αρχιτεκτονική Πολυπλεξίας Διαίρεσης Χρόνου στο δίκτυο G-PON	24
2.2.2 Τύποι αναγνωριστικών και αντικειμένων στο δίκτυο G-PON	26
2.2.3 Έλεγχος πρόσβασης στο μέσο	26
2.2.4 Δυναμική ανάθεση εύρους ζώνης.....	27
2.2.4.1. Λειτουργικές απαιτήσεις & μέθοδοι δυναμικής ανάθεσης εύρους ζώνης	29
2.2.4.2. Μαθηματικό μοντέλο δυναμικής ανάθεσης εύρους ζώνης	29
2.2.4.3 Περιγραφείς Κίνησης.....	30
2.2.4.4 Συνιστώσες εύρους ζώνης	30
2.2.5 Περιγραφείς κίνησης και τύποι κλάσεων T-Cont	33
2.2.6 Πλαισίωση στο στρώμα σύγκλισης μετάδοσης του πρωτοκόλλου G-PON	34
2.2.6.1 Δομή πλαισίου σύγκλισης μετάδοσης του G-PON στη ροή καθόδου	34
2.2.6.2 Δομή πλαισίου σύγκλισης μετάδοσης του G-PON στη ροή ανόδου	36
2.2.6.3 Χαρτογράφηση πλαισίων GEM στο τμήμα φόρτου δεδομένων	39
ΚΕΦΑΛΑΙΟ 3^ο: ΠΕΡΙΒΑΛΛΟΝ ΠΡΟΣΟΜΟΙΩΣΗΣ ΟΡΝΕΤ & ΤΟ ΜΟΝΤΕΛΟ ΤΟΥ ΔΙΚΤΥΟΥ G-PON	41
3.1 ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΠΡΟΣΟΜΟΙΩΣΗΣ ΟΡΝΕΤ	41
3.1.1 Επίπεδα μοντελοποίησης	41
3.1.2 Αντικείμενα και μοντέλα	42
3.1.3 Γεγονότα και Χρόνος προσομοίωσης	43
3.1.4 Η λειτουργία του επιπέδου διεργασίας	44
3.1.4.1 Διαγράμματα Μετάβασης Καταστάσεων	45
3.1.4.2 Μεταβλητές	47
3.1.5 Η λειτουργία του επιπέδου κόμβου	48
3.1.5.1 Κατηγορίες δομικών μονάδων	48
3.1.5.2 Σύνδεση δομικών μονάδων.....	52
3.1.6 Η λειτουργία του επιπέδου δικτύου	53
3.1.6.1 Τύποι αντικειμένων στο επίπεδο δικτύου.....	53
3.2 ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΟΥ ΔΙΚΤΥΟΥ G-PON	55
3.2.1 Ο κόμβος τερματικού OLT του δικτύου G-PON	57
3.2.1.1 Χαρακτηριστικά του τερματικού OLT	58
3.2.1.2 Η διεργασία του επεξεργαστή του τερματικού OLT	60

3.2.2 Ο κόμβος μονάδας ONU του δικτύου G-PON	63
3.2.2.1 Χαρακτηριστικά μονάδας ONU	63
3.2.2.2 Η διεργασία του επεξεργαστή της μονάδας ONU	65
3.2.3 Η διεργασία της πηγής κίνησης “gpon_bursty_source”	68
3.2.4 Η διεργασία του ενταμιευτή “pc_fifo_enhanced”	70
3.2.5 Η μορφή της παραγόμενης κίνησης	72
ΚΕΦΑΛΑΙΟ 4^ο : ΕΚΤΕΛΕΣΗ ΠΡΟΣΟΜΙΩΣΕΩΝ, ΑΠΟΤΕΛΕΣΜΑΤΑ & ΣΥΜΠΕΡΑΣΜΑΤΑ.....	75
4.1 ΡΥΘΜΙΣΕΙΣ ΛΕΙΤΟΥΡΓΙΑΣ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ	75
4.1.1 Ρυθμίσεις κίνησης	76
4.1.2 Ρυθμίσεις περιγραφέων κίνησης	76
4.2 ΓΡΑΦΙΚΕΣ ΠΑΡΑΣΤΑΣΕΙΣ	77
ΠΑΡΑΡΤΗΜΑ Α : ΔΙΑΓΡΑΜΜΑΤΑ ΜΕΤΑΒΑΣΗΣ ΚΑΤΑΣΤΑΣΕΩΝ & ΚΩΔΙΚΑΣ ΜΟΝΤΕΛΟΥ	
ΠΡΟΣΟΜΙΩΣΗΣ	87
A.1 Ο ΕΠΕΞΕΡΓΑΣΤΗΣ ΤΟΥ ΤΕΡΜΑΤΙΚΟΥ OLT	87
A.1.1 Εκτελέσεις εισόδου κατάστασης “init”	87
A.1.2 Συνάρτηση “self_interrupt()”	90
A.1.3 Συνάρτηση “endsim_stat_write()”	91
A.1.4 Συνάρτηση “process_up_burst()”	92
A.1.5 Συνάρτηση “send_gtc_d_frame()”	97
A.2 Ο ΕΠΕΞΕΡΓΑΣΤΗΣ ΤΗΣ ΜΟΝΑΔΑΣ ONU	102
A.2.1 Εκτελέσεις εισόδου κατάστασης “init”	103
A.2.1 Συνάρτηση “process_gtc_d()”	103
A.2.3 Συνάρτηση “send_upstream_burst()”	104
A.3 Η ΠΗΓΗ ΚΙΝΗΣΗΣ “GPN_BURSTY_SOURCE”	109
A.3.1 Εκτελέσεις εισόδου κατάστασης “init”	109
A.3.2 Εκτελέσεις εισόδου κατάστασης “off”	110
A.3.3 Εκτελέσεις εξόδου κατάστασης “off”	110
A.3.5 Εκτελέσεις εξόδου κατάστασης “on”	112
A.4 Ο ΕΝΤΑΜΙΕΥΤΗΣ “PC_FIFO_ENHANCED”	115
A.4.1 Εκτελέσεις εισόδου κατάστασης “init”	115
A.4.2 Εκτελέσεις εισόδου κατάστασης “SEND_HEAD”	116
A.4.3 Εκτελέσεις εισόδου κατάστασης “INS_TAIL”	117
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	118

ΑΚΡΩΝΥΜΙΑ

A/A	Ακρωνύμιο	Περιγραφή Ακρωνυμίου
1	BER	Bit Error Ratio
2	B-PON	Broadband - Passive Optical Network
3	BWmap	BandWidth mapping
4	DBA	Dynamic Bandwidth Allocation
5	DBRu	Dynamic Bandwidth Report upstream
6	E-PON	Ethernet - Passive Optical Network
7	FSAN	Full Service Access Network
8	FSM	Finite State Machine
9	FTTB/C	Fiber To The Building/Curb
10	FTTCab	Fiber To The Cabinet
11	FTTH	Fiber To The Home
12	GEM	G-PON Encapsulation Method
13	G-PON	Gigabit capable - Passive Optical Network
14	GTC	G-PON Transmission Convergence
15	GTCd	G-PON Transmission Convergence downstream
16	HEC	Header Error Control
17	IP	Internet Protocol
18	ISDN	Integrated Services Digital Network
19	MAC	Medium Access Control
20	NRZ	Non Return to Zero
21	ODN	Optical Distribution Network
22	OLT	Optical Line Termination

23	ONT	Optical Network Termination
24	ONU	Optical Network Unit
25	PCBd	Physical Control Block downstream
26	PLOAMd / u	Physical Layer Operations, Administrations and Maintenance downstream / upstream
27	PLOu	Physical Layer Overhead upstream
28	PON	Passive Optical Network
29	PSTN	Public Switched Telephone Network
30	QoS	Quality of Service
31	SNI	Service Node Interface
32	STD	State Transition Diagram
33	T-CONT	Transmission - CONTainer
34	TDM	Time Division Multiplexing
35	TDMA	Time Division Multiple Access
36	UNI	User Network Interface
37	VoIP	Voice over Internet Protocol

ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Παθητικό Οπτικό Δίκτυο τοπολογίας δέντρου.....	21
Σχήμα 2: Αρχιτεκτονικές Οπτικών Δικτύων Πρόσβασης.....	22
Σχήμα 3: Η διαδικασία πολυπλεξίας ροής καθόδου.....	25
Σχήμα 4: Η διαδικασία πολυπλεξίας ροής ανόδου.....	25
Σχήμα 5: Το πλαίσιο σύγκλισης μετάδοσης G-PON ροής καθόδου.....	27
Σχήμα 6: Ο μηχανισμός Δυναμικής Ανάθεσης Εύρους ζώνης.....	28
Σχήμα 7: Οι συνιστώσες του συνολικά εκχωρημένου εύρους ζώνης.....	31
Σχήμα 8: Η δομή του πλαισίου σύγκλισης μετάδοσης G-PON ροής ανόδου και καθόδου.....	34
Σχήμα 9: Τα ενθυλακωμένα πλαίσια GEM στο τμήμα φόρτου δεδομένων GTC.....	35
Σχήμα 10: Τα πεδία του τμήματος PCBd.....	35
Σχήμα 11: Η δομή του μηνύματος PLOAMd.....	36
Σχήμα 12: Η δομή της χαρτογράφησης εύρους ζώνης ροής ανόδου.....	36
Σχήμα 13: Η δομή μιας έκρηξης μετάδοσης ροής ανόδου.....	37
Σχήμα 14: Τα ενθυλακωμένα πλαίσια GEM σε μια έκρηξη μετάδοσης ροής ανόδου.....	39
Σχήμα 15: Η δομή ενός ενθυλακωμένου πλαισίου GEM.....	39
Σχήμα 16: Τα πεδία της επικεφαλίδας του πλαισίου GEM.....	40
Σχήμα 17: Τεμαχισμένα πλαίσια δεδομένων χρήστη.....	40
Σχήμα 18: Τα επίπεδα μοντελοποίησης του περιβάλλοντος OPNET.....	42
Σχήμα 19: Γεγονότα & Χρόνος προσομοίωσης.....	44
Σχήμα 20: Συμβολισμός καταστάσεων στο περιβάλλον προσομοίωσης OPNET.....	45
Σχήμα 21: Εκτελέσεις εισόδου & εξόδου κατά την κλήση μιας διεργασίας.....	46
Σχήμα 22: Η αρχική κατάσταση.....	46
Σχήμα 23: Συμβολισμός ενός επεξεργαστή στο περιβάλλον OPNET.....	49
Σχήμα 24: Συμβολισμός μιας ουράς στο περιβάλλον OPNET.....	49
Σχήμα 25: Υποουρές στο εσωτερικό μιας δομικής μονάδας ουράς.....	50
Σχήμα 26: Συμβολισμός πομπών στο περιβάλλον OPNET.....	50
Σχήμα 27: Λειτουργία ενός πομπού για την επικοινωνία μεταξύ δύο κόμβων.....	51
Σχήμα 28: Συμβολισμός δεκτών στο περιβάλλον OPNET.....	51
Σχήμα 29: Λειτουργία ενός δέκτη για την επικοινωνία μεταξύ δύο κόμβων.....	52
Σχήμα 30: Συμβολισμός ροών πακέτων στο περιβάλλον OPNET.....	52
Σχήμα 31: Μοντελοποίηση ενός δικτύου τοπολογίας αστέρα.....	53
Σχήμα 32: Ιεραρχία υποδικτύων.....	54
Σχήμα 33: Το εσωτερικό μοντέλο λειτουργίας ενός κόμβου επικοινωνιών.....	54
Σχήμα 34: Κανάλια επικοινωνίας στο εσωτερικό μιας ζεύξης.....	55
Σχήμα 35: Η τοπολογία παθητικού δέντρου.....	55
Σχήμα 36: Το μοντέλο του δικτύου G-PON στο περιβάλλον OPNET.....	56
Σχήμα 37: Η διαδικασία αντιστάθμισης απόστασης.....	57
Σχήμα 38: Δομικές μονάδες στο εσωτερικό του κόμβου τερματικού OLT.....	57
Σχήμα 39: Ρυθμίσεις των χαρακτηριστικών του τερματικού OLT.....	59
Σχήμα 40: Διάγραμμα Μετάβασης Καταστάσεων του επεξεργαστή του τερματικού OLT.....	60
Σχήμα 41: Μοντελοποίηση της δομής του τμήματος PCBd.....	61
Σχήμα 42: Μοντελοποίηση της δομής της χαρτογράφησης εύρους ζώνης ροής ανόδου.....	61

Σχήμα 43: Μοντελοποίηση της δομής ενός πλαισίου GEM	62
Σχήμα 44: Δομικές μονάδες στο εσωτερικό ενός κόμβου μονάδας ONU	63
Σχήμα 45: Ρυθμίσεις των χαρακτηριστικών μιας μονάδας ONU	64
Σχήμα 46: Η εκρηκτική λειτουργία των πηγών κίνησης	65
Σχήμα 47: Διάγραμμα Μετάβασης Καταστάσεων του επεξεργαστή μιας μονάδας ONU	66
Σχήμα 48: Μοντελοποίηση της δομής μιας έκρηξης μετάδοσης ροής ανόδου	67
Σχήμα 49: Μοντελοποίηση του τμήματος φόρτου δεδομένων χρήστη	67
Σχήμα 50: Μοντελοποίηση των πλαισίων GEM	67
Σχήμα 51: Ρυθμίσεις χαρακτηριστικών μιας πηγής κίνησης	68
Σχήμα 52: Διάγραμμα Μετάβασης Καταστάσεων του επεξεργαστή μιας πηγής κίνησης	69
Σχήμα 53: Ρυθμίσεις χαρακτηριστικών ενός ενταμιευτή δεδομένων	70
Σχήμα 54: Διάγραμμα Μετάβασης Καταστάσεων του επεξεργαστή ενός ενταμιευτή	71
Σχήμα 56: Αποδοτικότητα ροής ανόδου	77
Σχήμα 57: Μέση καθυστέρηση πακέτων κλάσης T-Cont 1	78
Σχήμα 58: Jitter πακέτων κλάσης T-Cont 1	79
Σχήμα 59: Μέση καθυστέρηση πακέτων κλάσης T-Cont 2	79
Σχήμα 60: Μέση καθυστέρηση πακέτων κλάσης T-Cont 3	80
Σχήμα 61: Jitter πακέτων κλάσης T-Cont 2	80
Σχήμα 62: Jitter πακέτων κλάσης T-Cont 3	81
Σχήμα 63: Μέση καθυστέρηση πακέτων κλάσης T-Cont 4	81
Σχήμα 64: Jitter πακέτων κλάσης T-Cont 4	82
Σχήμα 65: Πυκνότητα πιθανότητας καθυστέρησης,	83
για επίπεδο φόρτου 30% και κύκλο δειγματοληψίας 125 μs	83
Σχήμα 66: Πυκνότητα πιθανότητας καθυστέρησης,	83
για επίπεδο φόρτου 70% και κύκλο δειγματοληψίας 125 μs	83
Σχήμα 67: Πυκνότητα πιθανότητας καθυστέρησης,	84
για επίπεδο φόρτου 90% και κύκλο δειγματοληψίας 125 μs	84
Σχήμα 68: Πυκνότητα πιθανότητας καθυστέρησης,	84
για επίπεδο φόρτου 90% και κύκλο δειγματοληψίας 250 μs	84
Σχήμα 699: Πυκνότητα πιθανότητας καθυστέρησης,	85
για επίπεδο φόρτου 90% και κύκλο δειγματοληψίας 500 μs	85
Σχήμα 70: Πυκνότητα πιθανότητας καθυστέρησης,	85
για επίπεδο φόρτου 90% και κύκλο δειγματοληψίας 1000 μs	85
Σχήμα 71: Πυκνότητα πιθανότητας καθυστέρησης,	86
για επίπεδο φόρτου 90% και κύκλο δειγματοληψίας 2000 μs	86

ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ

Πίνακας 1: Περιγραφείς κίνησης κλάσεων T-Cont.....	33
Πίνακας 2: Κατανομή πιθανοτήτων των μηκών των πακέτων δεδομένων χρήστη.....	73
Πίνακας 3: Μορφή παραγόμενης κίνησης.....	76
Πίνακας 4: Τιμές περιγραφέων κίνησης για φόρτο δικτύου 30% / 70% / 90%	77

ΚΕΦΑΛΑΙΟ 1^ο : Εισαγωγή

Ως απάντηση στη διαρκώς αυξανόμενη ζήτηση για εύρος ζώνης και δικτυακές υπηρεσίες τόσο από τους οικιακούς χρήστες όσο και από τους επιχειρησιακούς πελάτες, τα Παθητικά Οπτικά Δίκτυα (Passive Optical Networks) έχουν αναδειχθεί την τελευταία δεκαετία ως μια υποσχόμενη τεχνολογία πρόσβασης που προσφέρει ευελιξία, δυνατότητες ευρείας κάλυψης και οικονομικώς αποδοτικό διαμοιρασμό των ακριβών οπτικών ζεύξεων συγκριτικά με τις συμβατικές λύσεις σημείου-προς-σημείο. Τα συγκεκριμένα πλεονεκτήματα των Παθητικών Οπτικών Δικτύων έχουν πυροδοτήσει τα τελευταία χρόνια σημαντική εμπορική δραστηριότητα, η οποία αντικατοπτρίζεται και στις εργασίες αρκετών αρχών προτυποποίησης.

Το 1998 ο τομέας προτυποποίησης της Διεθνούς Ένωσης Τηλεπικοινωνιών (ITU-T) δημοσίευσε τις πρώτες συστάσεις της σειράς G.983.x, οι οποίες αναφέρονται συχνά ως πρωτόκολλο B-PON και είχαν συνταχθεί από την ομάδα Full Service Access Network (FSAN). Το πρωτόκολλο B-PON χρησιμοποιεί το ATM ως στρώμα μεταφοράς και επιδεικνύει ορισμένα ισχυρά χαρακτηριστικά, γεγονός που επαληθεύεται από το εύρος των υλοποιήσεων βασισμένες στο πρωτόκολλο B-PON που έχουν πραγματοποιηθεί μέχρι σήμερα. Το 2004 δύο νέες τεχνολογίες Παθητικών Οπτικών Δικτύων προτυποποιήθηκαν: το πρωτόκολλο E-PON από το ινστιτούτο IEEE και το πρωτόκολλο G-PON από τον τομέα ITU-T, το οποίο συντάχθηκε από την ομάδα FSAN και επικυρώθηκε με τις συστάσεις G.984.x. Και οι δύο τεχνολογίες προσφέρουν μεταφορά δεδομένων βασισμένη σε λειτουργίες πακέτων, με ταχύτητες τάξεως Gigabit ανά δευτερόλεπτο.

Παρόλο που οι δύο τεχνολογίες αναπτύχθηκαν και προτυποποιήθηκαν σε παρόμοιο χρονικό πλαίσιο και υπήρχε επικοινωνία μεταξύ των δύο ομάδων με στόχο την ανταλλαγή πληροφοριών και την κατάληξη σε κοινές προδιαγραφές, στην τελική μορφή τους τα δύο πρωτόκολλα εμφανίζουν σημαντικές διαφοροποιήσεις. Σε τεχνικό επίπεδο, το πρωτόκολλο E-PON σχεδιάστηκε με γνώμονα την απλουστευμένη λειτουργία, επιτυγχάνοντας συμβατότητα με τα πρότυπα 802.3. Η ομάδα FSAN αντίθετα, επιχείρησε τη δημιουργία του πρωτοκόλλου G-PON από την αρχή με βάση συγκεκριμένες απαιτήσεις απόδοσης και υποστήριξης υπηρεσιών. Στην πράξη τελικά υιοθετήθηκαν αρκετά χαρακτηριστικά από το πρωτόκολλο B-PON, ωστόσο η τελική μορφή του πρωτοκόλλου G-PON απαιτούσε τη χρησιμοποίηση πιο πολύπλοκου εξοπλισμού επιτυγχάνοντας ταυτόχρονα αυξημένη απόδοση.

Παράλληλα με την οικονομική ανάπτυξη και λειτουργία που επιτυγχάνεται με τα Παθητικά Οπτικά Δίκτυα, η οποία είναι εξαιρετικά σημαντική για τους διαχειριστές και τους παρόχους υπηρεσιών, εξίσου σημαντική αποδεικνύεται η υψηλή και δίκαιη χρησιμοποίηση των διαθέσιμων πόρων. Για την επίτευξη αυτής, ο μηχανισμός κατανομής εύρους ζώνης πρέπει να σχεδιασθεί ώστε να επιτυγχάνει βέλτιστη ισορροπία μεταξύ αποδοτικότητας και συγκεκριμένων εγγυήσεων επίδοσης, ώστε να υποστηρίζονται υπηρεσίες και εφαρμογές με διαφορετικές απαιτήσεις. Η υποστήριξη ξεχωριστών επιπέδων ποιότητας υπηρεσίας (Quality of Service) είναι κρίσιμη για την επιτυχία μιας τεχνολογίας πρόσβασης, αφού σχετίζεται άμεσα με τη προσφορά υπηρεσιών όπως είναι η τηλεφωνία, το διαδίκτυο και τα πολυμέσα κατά παραγγελία.

1.1 Το αντικείμενο και ο στόχος της διπλωματικής εργασίας

Η παρούσα διπλωματική εργασία έχει ως στόχο τη μοντελοποίηση ενός δικτύου πρόσβασης βασισμένο σε Παθητικά Οπτικά Δίκτυα, καθώς και τη μελέτη της συμπεριφοράς του υπό διάφορες συνθήκες λειτουργίας. Αναλυτικά:

- Επιλέχθηκε να μοντελοποιηθεί ένα δίκτυο πρόσβασης βασισμένο στο πρωτόκολλο G-PON, το οποίο θα ακολουθεί πλήρως τις προδιαγραφές λειτουργίας που καθορίζονται στις συστάσεις G.984.x του τομέα ITU-T. Το μοντέλο του πρωτοκόλλου σχεδιάστηκε με χρήση του περιβάλλοντος προσομοίωσης OPNET Modeler, ενσωματώνοντας κάθε απαραίτητο χαρακτηριστικό και λειτουργία του πρωτοκόλλου για τις ανάγκες της προσομοίωσης της λειτουργίας του.
- Με την εκτέλεση προσομοιώσεων λειτουργίας του μοντέλου υπό διάφορες συνθήκες φόρτου κίνησης και ρυθμίσεις λειτουργίας, συγκεντρώθηκαν επιλεγμένα στατιστικά τα οποία σχετίζονται με τις παραμέτρους απόδοσης του δικτύου.
- Με βάση τις γραφικές παραστάσεις που συνοψίζουν τα αποτελέσματα των προσομοιώσεων, προέκυψαν ορισμένα συμπεράσματα σχετικά με τις βέλτιστες ρυθμίσεις λειτουργίας του δικτύου G-PON.

1.2 Η δομή της διπλωματικής εργασίας

Το περιεχόμενο της διπλωματικής εργασίας, εξαιρουμένου του παρόντος εισαγωγικού κεφαλαίου, αναλύεται σε τρία κεφάλαια και ένα συμπληρωματικό παράρτημα:

- Στο 2^ο Κεφάλαιο πραγματοποιείται μία εισαγωγή στην ορολογία και την αρχιτεκτονική των οπτικών δικτύων πρόσβασης. Στη συνέχεια αναλύονται τα γενικά χαρακτηριστικά και οι προδιαγραφές λειτουργίας του πρωτοκόλλου G-PON, με έμφαση στο στρώμα σύγκλισης μετάδοσης του πρωτοκόλλου (GTC layer).
- Στο 3^ο Κεφάλαιο περιλαμβάνεται μία συνοπτική παρουσίαση του περιβάλλοντος προσομοίωσης OPNET και των 4 προσφερόμενων επιπέδων μοντελοποίησης. Στο δεύτερο μέρος του κεφαλαίου περιγράφεται η διαδικασία σχεδιασμού του μοντέλου του πρωτοκόλλου G-PON, με αναλυτική περιγραφή της λειτουργίας των δομικών του μονάδων.
- Στο 4^ο Κεφάλαιο εκτίθενται τα αποτελέσματα των προσομοιώσεων της λειτουργίας του μοντέλου, ανάλογα με τις ρυθμίσεις λειτουργίας και τις συνθήκες φόρτου του δικτύου. Με βάση τα στατιστικά των προσομοιώσεων και τις γραφικές παραστάσεις εξάγονται συμπεράσματα σχετικά με τις ρυθμίσεις βέλτιστης λειτουργίας του δικτύου.
- Στο Παράρτημα Α συγκεντρώνεται ο κώδικας που εκτελείται για τις προσομοιώσεις της λειτουργίας του μοντέλου G-PON.

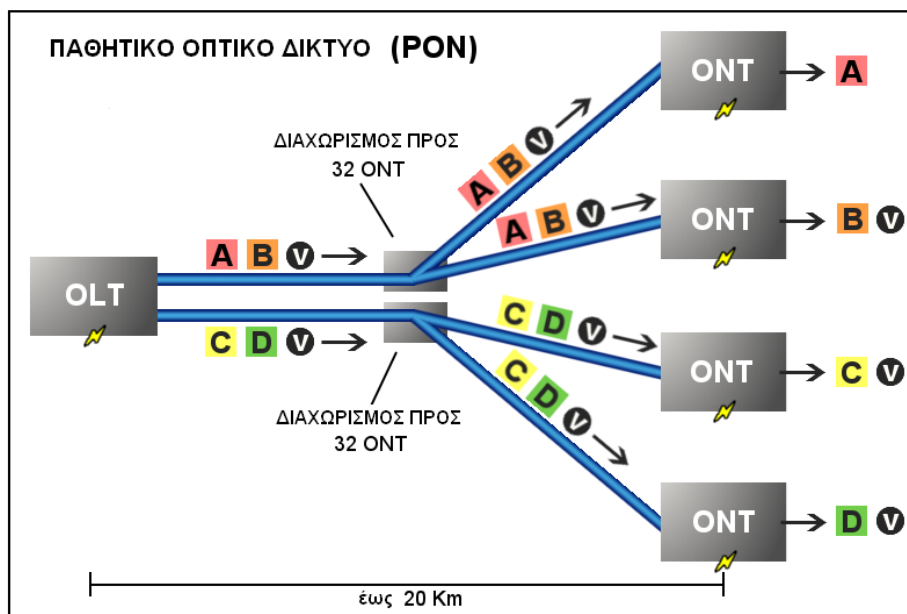
ΚΕΦΑΛΑΙΟ 2^ο: Παθητικά Οπτικά Δίκτυα & Προδιαγραφές λειτουργίας πρωτοκόλλου G-PON

2.1 Γενικά χαρακτηριστικά οπτικών δικτύων πρόσβασης & δικτύων G-PON

Το οπτικό τμήμα ενός δικτύου πρόσβασης μπορεί να είναι ενεργό ή παθητικό, όπως επίσης και τύπου σημείο προς σημείο (point-to-point) ή σημείο προς πολλαπλά σημεία (point-to-multipoint). Τα Παθητικά Οπτικά Δίκτυα είναι οπτικά δίκτυα σημείου-προς-πολλαπλά σημεία, τα οποία δεν περιέχουν ενεργά στοιχεία, δηλαδή δεν πραγματοποιείται μετατροπή του οπτικού σήματος σε ηλεκτρικό από την πηγή μέχρι το προορισμό.

2.1.1 Αρχιτεκτονική οπτικών δικτύων πρόσβασης

Η ανταλλαγή δεδομένων στα Παθητικά Οπτικά Δίκτυα πραγματοποιείται μεταξύ του Οπτικού Τερματιστή Γραμμής (Optical Line Termination, OLT) και των Οπτικών Μονάδων Δικτύου (Optical Network Unit, ONU) ή των Οπτικών Τερματιστών Δικτύου (Optical Network Termination, ONT), μέσω του Οπτικού Δικτύου Διανομής (Optical Distribution Network, ODN) που τα διασυνδέει. Το τερματικό OLT αποτελεί το σημείο προσαρμογής μεταξύ του δικτύου PON και του δικτύου κορμού στο κέντρο μεταγωγής.

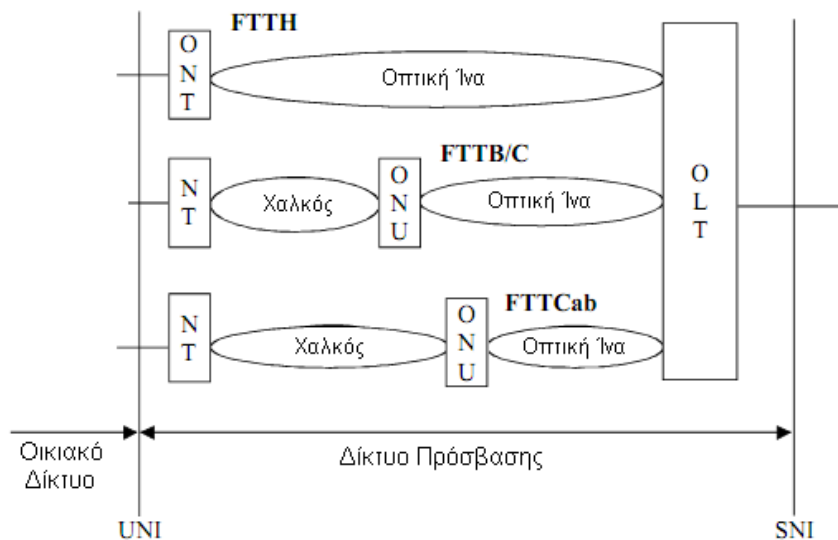


Σχήμα 1: Παθητικό Οπτικό Δίκτυο τοπολογίας δέντρου

Σημείωση σχετικά με τους όρους ONU/ONT: το δικτυακό στοιχείο που λειτουργεί ως διεπαφή για το Οπτικό Δίκτυο Διανομής και τις εγκαταστάσεις πρόσβασης του τελικού χρήστη, θα αναφέρεται από αυτό το σημείο ως Οπτική Μονάδα Δικτύου (ONU). Σε ορισμένες συστάσεις της ITU-T πραγματοποιείται διάκριση των εννοιών ONU και ONT, θεωρώντας το τερματικό ONT ως μία ειδική περίπτωση (που αφορά ένα μόνο χρήστη) μιας μονάδας ONU. Ωστόσο, δεδομένου ότι από τη σκοπιά της λειτουργίας του στρώματος σύγκλισης μετάδοσης του δικτύου G-PON οι δύο οντότητες θεωρούνται ταυτόσημες, ο όρος μονάδα ONU θα αναφέρεται εφεξής και στις δύο έννοιες.

Οι αρχιτεκτονικές Οπτικών Δικτύων Πρόσβασης που έχουν επικρατήσει μέχρι σήμερα είναι οι:

- Οπτική Ίνα μέχρι το Σπίτι (Fiber To The Home, FTTH)
- Οπτική Ίνα μέχρι το Κτίριο/Πεζοδρόμιο (Fiber To The Building/Curb, FTTB/C)
- Οπτική Ίνα μέχρι το Καφάο (Fiber To The Cabinet, FTTCab)



Σχήμα 2: Αρχιτεκτονικές Οπτικών Δικτύων Πρόσβασης

Οι διαφορές των επιλογών που σχετίζονται με τις αρχιτεκτονικές FTTB, FTTC, FTTCab και FTTH οφείλονται κατά κύριο λόγο στις διαφορές μεταξύ των υπηρεσιών που υποστηρίζονται, επομένως για τις ανάγκες της ανάλυσης της λειτουργίας των δικτύων G-PON αντιμετωπίζονται ως μία ενιαία αρχιτεκτονική. Τα δίκτυα G-PON σχεδιάστηκαν με την απαίτηση για υποστήριξη όλων των γνωστών υπηρεσιών αλλά και νέες υπηρεσίες υπό συζήτηση, τόσο για οικιακούς συνδρομητές όσο και για τους επιχειρησιακούς πελάτες των παρόχων. Η επιλογή των προσφερόμενων υπηρεσιών εξαρτάται από τις ρυθμιστικές συνθήκες της αγοράς που δραστηριοποιούνται οι πάροχοι αλλά και από τις ανάγκες και δυνατότητες της αγοράς. Η προσφορά των υπηρεσιών με οικονομικός αποδοτικό τρόπο είναι συνάρτηση όχι μόνο νομικών πλαισίων, αλλά επιπλέον παραγόντων όπως οι υπάρχουσες τηλεπικοινωνιακές υποδομές, η οικιακή διανομή και η αναλογία οικιακών πελατών - επιχειρήσεων. Ορισμένες από τις υπηρεσίες που υποστηρίζονται από τα δίκτυα G-PON είναι:

- Υπηρεσίες δεδομένων (Ethernet: IP, VoIP, ροές video με κωδικοποίηση MPEG-2/-4)
- Υπηρεσίες PSTN (POTS, ISDN BRI/PRI)
- Υπηρεσίες μισθωμένης γραμμής (T1, E1, DS3, E3)
- Υπηρεσίες video (ψηφιακό video)

2.1.2 Γενικά χαρακτηριστικά δικτύων G-PON

Με βάση την προτυποποίηση του τομέα ITU-T ([4], [5]), για τα δίκτυα G-PON ισχύουν τα ακόλουθα:

Ρυθμός μετάδοσης (citrade): τα δίκτυα G-PON στοχεύουν σε ρυθμούς μετάδοσης μεγαλύτερους ή ίσους με 1.24416 Gbit/s. Οι επιτρεπτοί συνδυασμοί ρυθμών μετάδοσης στην πράξη είναι:

- 1.24416 Gbit/s στην άνοδο, 2.48832 Gbit/s στην κάθοδο
- 2.48832 Gbit/s στην άνοδο, 2.48832 Gbit/s στην κάθοδο

Ο πιο συνηθισμένος συνδυασμός ρυθμών μετάδοσης, ο οποίο συνιστά την πλειοψηφία των εγκατεστημένων και προγραμματισμένων προς εγκατάσταση συστημάτων G-PON είναι ο 1.24416 Gbit/s στην άνοδο, 2.48832 Gbit/s στην κάθοδο.

Λογική Εμβέλεια (logical reach): η λογική εμβέλεια ορίζεται ως η μέγιστη απόσταση μεταξύ των μονάδων ONU και του τερματικού OLT, μη συμπεριλαμβανομένων των περιορισμών του φυσικού στρώματος. Για τα δίκτυα G-PON, η μέγιστη λογική εμβέλεια ορίζεται στα 60 km.

Φυσική Εμβέλεια (physical reach): η φυσική εμβέλεια ορίζεται ως η μέγιστη φυσική απόσταση μεταξύ των μονάδων ONU και του τερματικού OLT. Για Οπτικά Δίκτυα Διανομής κλάσης C, η φυσική εμβέλεια είναι ορισμένη για τα δίκτυα G-PON στα 20 km.

Διαφορική απόσταση οπτικών ινών (differential fiber distance): για τα δίκτυα G-PON η μέγιστη διαφορική απόσταση οπτικών ινών ορίζεται στα 20 km.

Μέγιστη μέση καθυστέρηση μεταφοράς σήματος (maximum mean signal transfer delay): τα δίκτυα G-PON οφείλουν να υποστηρίζουν υπηρεσίες που απαιτούν μέγιστη μέση καθυστέρηση μεταφοράς σήματος 1.5 ms.

Λόγος διαχωρισμού (split ratio): όσο μεγαλύτερος είναι ο λόγος διαχωρισμού που υποστηρίζεται από το δίκτυο G-PON, τόσο πιο ελκυστικό αποδεικνύεται για τον πάροχο. Ωστόσο ένας υψηλός λόγος διαχωρισμού μεταφράζεται σε υψηλότερο οπτικό διαχωρισμό, για τον οποίο απαιτείται αυξημένος προϋπολογισμός ισχύος για την κάλυψη της φυσικής εμβέλειας. Λόγοι διαχωρισμού μέχρι 1:64 αποδεικνύονται ρεαλιστικοί για το φυσικό στρώμα (βασισμένο σε Οπτικά Δίκτυα Διανομής κλάσης C), με βάση την υπάρχουσα τεχνολογία. Ωστόσο με τη διαρκή εξέλιξη των οπτικών μονάδων, το στρώμα σύγκλισης μετάδοσης οφείλει να υποστηρίζει λόγο διαχωρισμού μέχρι 1:128.

Μήκος Κύματος λειτουργίας: στην κατεύθυνση ροής καθόδου, το εύρος μήκους κύματος λειτουργίας για συστήματα μονής οπτικής ίνας είναι 1480-1500 nm. Στην κατεύθυνση ροής ανόδου το αντίστοιχο εύρος είναι 1260-1360 nm.

Κωδικοποίηση γραμμής: κωδικοποίηση NRZ, στην άνοδο και τη κάθοδο.

2.2 Το στρώμα σύγκλισης μετάδοσης του δικτύου G-PON

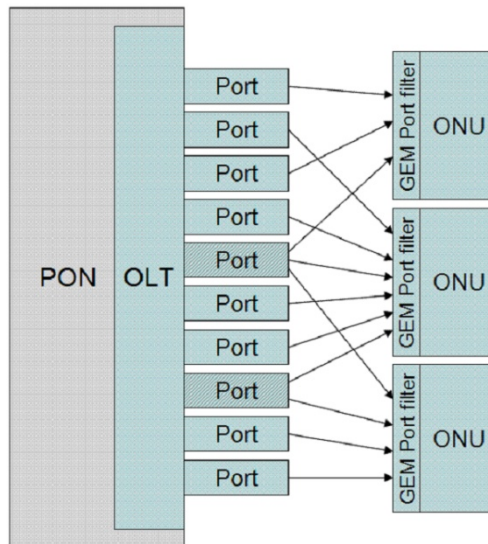
Στην παρούσα ενότητα αναλύονται οι προδιαγραφές και οι συστάσεις που αναφέρονται στο στρώμα σύγκλισης μετάδοσης του δικτύου G-PON (G-PON Transmission Convergence layer, GTC layer), ανάμεσα στη διεπαφή κόμβου υπηρεσίας (SNI) και τη διεπαφή χρήστη-δικτύου (UNI). Αναφέρονται μεταξύ άλλων οι προδιαγραφές σχετικά με τη μορφή των πλαισίων (frame format) και η μέθοδος που καθορίζει τον έλεγχο της πρόσβασης των μονάδων ONU στο μέσο (Medium Access Control).

Σημείωση: η σύσταση του τομέα ITU-T G.984.3, η οποία καθορίζει το στρώμα σύγκλισης μετάδοσης του δικτύου G-PON, αναγνωρίζει ως μοναδικό σχήμα μεταφοράς δεδομένων τη μέθοδο ενθυλάκωσης G-PON (G-PON Encapsulation Method, GEM). Η μέθοδος GEM παρέχει ένα μηχανισμό πλαισίωσης μεταβλητού μήκους προσανατολισμένο στη λειτουργία με σύνδεση, για τη μεταφορά των δεδομένων των χρηστών στο δίκτυο G-PON. Ο συγκεκριμένος μηχανισμός λειτουργεί ανεξάρτητα από τον τύπο της διεπαφής SNI του τερματικού OLT και τους τύπους των διεπαφών UNI στις μονάδες ONU.

Αν και στην αρχική της μορφή η σύσταση ITU-T G.984.3 προέβλεπε δύο τρόπους λειτουργίας για υποστήριξη τόσο μεταφοράς πλαισίου (GEM) όσο και μεταφοράς κελιού (ATM), πλέον η υποστήριξη μεταφοράς ATM από το στρώμα GTC έχουν εγκαταλειφθεί. Ο τρόπος μεταφοράς ATM δεν απαιτείται πλέον από καμία υπηρεσία που παρουσιάζει ενδιαφέρον, ενώ λίγα είναι τα συστήματα που τον υποστήριξαν στη πάροδο του χρόνου.

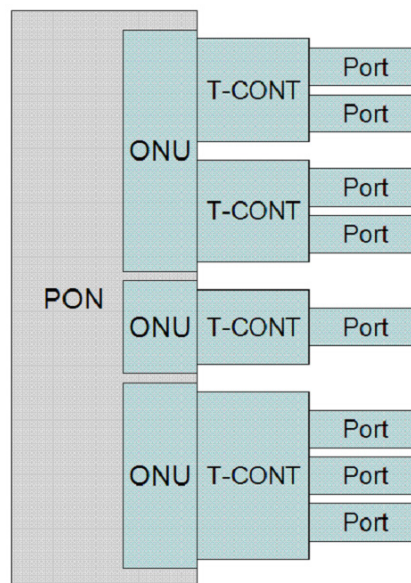
2.2.1 Αρχιτεκτονική Πολυπλεξίας Διαίρεσης Χρόνου στο δίκτυο G-PON

Στην κατεύθυνση ροής καθόδου (downstream direction), η λειτουργία πολυπλεξίας κίνησης είναι συγκεντρωτική και πραγματοποιείται από το τερματικό OLT. Τα πλαίσια GEM πολυπλέκονται στο μέσο μετάδοσης χρησιμοποιώντας το αναγνωριστικό θύρας GEM (GEM Port-ID) ως μοναδικό κλειδί για να αναγνωρισθούν τα πλαίσια που ανήκουν σε ξεχωριστές λογικές συνδέσεις της ροής καθόδου. Κάθε μονάδα ONU φιλτράρει τα πλαίσια GEM της ροής καθόδου ανάλογα με το αναγνωριστικό GEM Port-ID και επεξεργάζεται μόνο εκείνα που ανήκουν στην ίδια.



Σχήμα 3: Η διαδικασία πολυπλεξίας ροής καθόδου

Στην κατεύθυνση ροής ανόδου (upstream direction), η λειτουργία πολυπλεξίας κίνησης είναι κατανομημένη και πραγματοποιείται ξεχωριστά από κάθε μονάδα ONU. Το τερματικό OLT παραχωρεί τη δυνατότητα για μετάδοση στη ροή ανόδου, ή ισοδύναμα «κατανομές εύρους ζώνης ροής ανόδου» (upstream bandwidth allocations), σε κάθε οντότητα που φέρει κίνηση εντός των μονάδων ONU που διαχειρίζεται. Οι οντότητες που παράγουν κίνηση σε κάθε μονάδα ONU αναγνωρίζονται μοναδικά από τα «αναγνωριστικά κατανομών» (Alloc-IDs) που τους έχουν ανατεθεί. Οι κατανομές εύρους ζώνης ροής ανόδου των διαφόρων αναγνωριστικών Alloc-ID πολυπλέκονται στο χρόνο, με βάση τη «χαρτογράφηση εύρους ζώνης» (BWmap) που μεταδίδει το τερματικό OLT στη ροή καθόδου. Σε κάθε κατανομή εύρους ζώνης, η μονάδα ONU χρησιμοποιεί το αναγνωριστικό GEM Port-ID ως κλειδί πολυπλεξίας για να αναγνωρίσει τα πλαίσια GEM που ανήκουν σε ξεχωριστές λογικές συνδέσεις ροής ανόδου.



Σχήμα 4: Η διαδικασία πολυπλεξίας ροής ανόδου

2.2.2 Τύποι αναγνωριστικών και αντικειμένων στο δίκτυο G-PON

Αναγνωριστικό μονάδας ONU (ONU-ID): πρόκειται για ένα αναγνωριστικό μήκους 8 bit, το οποίο ανατίθεται από το τερματικό OLT σε κάθε μονάδα ONU κατά τη διαδικασία της ενεργοποίησής της και είναι μοναδικό εντός του δικτύου PON. Παραμένει έγκυρο μέχρι την απενεργοποίηση της μονάδας ONU από το τερματικό OLT ή τη μετάβασή της σε ανενεργή κατάσταση. Το εύρος τιμών που είναι διαθέσιμο προς ανάθεση στις μονάδες ONU είναι [0..253], η τιμή 254 είναι κρατημένη, ενώ η τιμή 255 αντιστοιχεί στην πολλαπλή εκπομπή προς όλες τις μονάδες ONU (στη ροή καθόδου) ή δεν χρησιμοποιείται (στη ροή ανόδου).

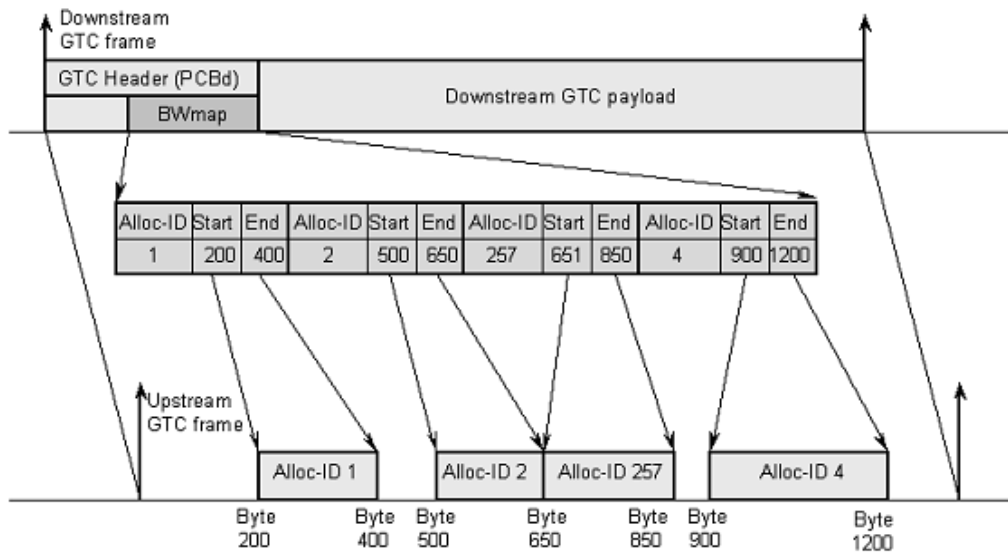
Αναγνωριστικό Κατανομής (Alloc-ID): το συγκεκριμένο αναγνωριστικό, μήκους 12 bit, ανατίθεται από το τερματικό OLT σε μία μονάδα ONU, για να αναγνωριστεί μία οντότητα εντός της μονάδας ONU η οποία φέρει κίνηση και είναι αποδέκτης κατανομών εύρους ζώνης ροής ανόδου. Μία τέτοια οντότητα αντιπροσωπεύεται από μία κλάση υπηρεσίας T-Cont. Σε κάθε μονάδα ONU ανατίθεται τουλάχιστον το προκαθορισμένο αναγνωριστικό Alloc-ID, κατά τη διάρκεια της ενεργοποίησης, το οποίο είναι αριθμητικά ίσο με το αναγνωριστικό ONU-ID. Ενδεχομένως να ανατεθούν περισσότερα αναγνωριστικά Alloc-ID, αν κριθεί απαραίτητο από το τερματικό OLT. Όλες οι αναθέσεις αναγνωριστικών Alloc-ID, συμπεριλαμβανομένου του προκαθορισμένου, μηδενίζονται όταν απενεργοποιηθεί η μονάδα ONU.

Κλάση υπηρεσίας T-Cont (Transmission container): ένα αντικείμενο της μονάδας ONU το οποίο αντιπροσωπεύει μια ομάδα λογικών συνδέσεων, οι οποίες εμφανίζονται ως μια μοναδική οντότητα στη λειτουργία ανάθεσης εύρους ζώνης ροής ανόδου. Για κάθε μονάδα ONU υπάρχει καθορισμένος αριθμός ξεχωριστών κλάσεων T-Cont που υποστηρίζονται. Το τερματικό OLT αναλαμβάνει να ανακαλύψει τον αριθμό των κλάσεων T-Cont που υποστηρίζουν οι μονάδες ONU και να δημιουργήσει μοναδικές αντιστοιχίες μεταξύ αυτών και των αναγνωριστικών Alloc-ID.

Αναγνωριστικό θύρας GEM (GEM Port-ID): ένας αριθμός μήκους 12 bit, ο οποίος ανατίθεται από το τερματικό OLT σε κάθε ξεχωριστή λογική σύνδεση δεδομένων με προορισμό τις μονάδες ONU.

2.2.3 Έλεγχος πρόσβασης στο μέσο

Το σύστημα σύγκλισης μετάδοσης του δικτύου G-PON παρέχει έλεγχο πρόσβασης στο μέσο μετάδοσης (Medium Access Control) για την κίνηση της ροής ανόδου: πλαίσια στη ροή καθόδου υποδεικνύουν τις επιτρεπτές θέσεις για την κίνηση στα πλαίσια GTC ανόδου, τα οποία μεταδίδονται σε συγχρονισμό με τα πλαίσια GTC καθόδου. Πιο αναλυτικά, το τερματικό OLT ορίζει δείκτες στο πεδίο χαρτογράφησης εύρους ζώνης ροής ανόδου (upstream BWmap) της επικεφαλίδας του πλαισίου GTC καθόδου. Οι δείκτες υποδεικνύουν τη χρονική στιγμή στην οποία κάθε μονάδα ONU οφείλει να αρχίσει και να σταματήσει τη μετάδοση στην άνοδο. Με αυτόν τον τρόπο μόνο μία μονάδα ONU είναι δυνατό να αποκτήσει πρόσβαση στο μέσο μετάδοσης σε μια δεδομένη χρονική στιγμή, επομένως στα πλαίσια της φυσιολογικής λειτουργίας δεν εμφανίζονται διαμάχες για τη πρόσβαση στο μέσο.

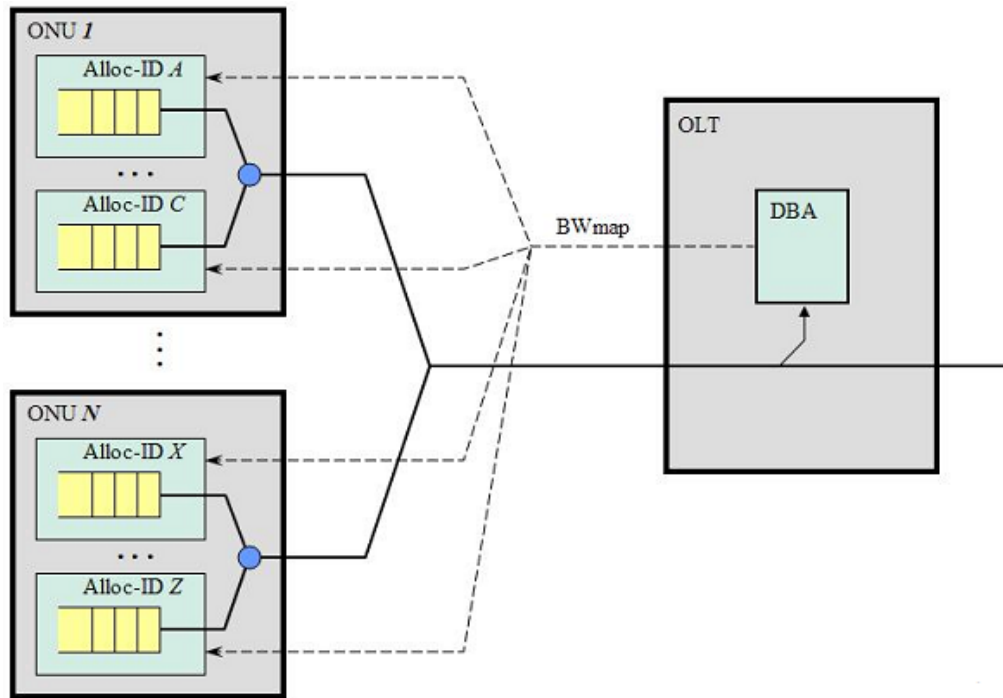


Σχήμα 5: Το πλαίσιο σύγκλισης μετάδοσης G-PON ροής καθόδου

2.2.4 Δυναμική ανάθεση εύρους ζώνης

Η δυναμική ανάθεση εύρους ζώνης (DBA) στα δίκτυα G-PON είναι η διαδικασία μέσω της οποίας το τερματικό OLT ανακατανέμει τις δυνατότητες μετάδοσης στη ροή ανόδου, στις οντότητες που φέρουν κίνηση εντός των μονάδων ONU που διαχειρίζεται. Η ανακατανομή πραγματοποιείται με βάση τη δυναμική υπόδειξη της κατάστασης λειτουργίας τους και τα καθορισμένα συμβόλαια κίνησης. Η υπόδειξη της κατάστασης λειτουργίας μπορεί να είναι είτε ρητή, μέσω αναφορών της κατάστασης των ενταμιευτών, είτε υπονοούμενη μέσω της μετάδοσης κενών πλαισίων GEM στη θέση των χορηγημένων δυνατοτήτων μετάδοσης στη ροή ανόδου.

Σε σύγκριση με τη στατική εκχώρηση εύρους ζώνης, ο μηχανισμός DBA βελτιώνει τη χρησιμοποίηση εύρους ζώνης ροής ανόδου στα δίκτυα G-PON, με το να προσαρμόζει τη λειτουργία του στην εκρηκτική φύση της κίνησης των μονάδων ONU. Τα πρακτικό όφελος του μηχανισμού DBA είναι διπλό: οι διαχειριστές του δικτύου έχουν τη δυνατότητα να προσθέσουν περισσότερους συνδρομητές στο δίκτυο PON λόγω της αποδοτικότερης χρήσης του εύρους ζώνης και επιπλέον οι συνδρομητές μπορούν να απολαμβάνουν πιο απαιτητικές υπηρεσίες μεταβλητού ρυθμού, οι οποίες δε θα ήταν δυνατές με τη στατική μέθοδο ανάθεσης.



Σχήμα 6: Ο μηχανισμός Δυναμικής Ανάθεσης Εύρους ζώνης

Στα δίκτυα G-PON, η οντότητα που γίνεται παραλήπτης μιας κατανομής εύρους ζώνης ροής ανόδου, εκπροσωπείται από ένα αναγνωριστικό Alloc-ID. Ανεξάρτητα από το πλήθος των αναγνωριστικών Alloc-ID που έχουν ανατεθεί σε κάθε μονάδα ONU, τον αριθμό των θυρών GEM που έχουν πολυπλεχθεί σε κάθε αναγνωριστικό Alloc-ID και τη φυσική και λογική δομή αναμονής που υλοποιείται από τη μονάδα ONU, το τερματικό OLT μοντελοποιεί το σύνολο της κίνησης που σχετίζεται με κάθε αναγνωριστικό Alloc-ID σαν ένα ξεχωριστό λογικό ενταμιευτή. Για τη λειτουργία της ανάθεσης εύρους ζώνης, το τερματικό OLT θεωρεί όλα τα αναγνωριστικά Alloc-IDs που έχουν προσδιορισθεί για το δίκτυο PON ως ανεξάρτητες ισάξιες οντότητες στο ίδιο επίπεδο λογικής ιεραρχίας.

Για τους λογικούς ενταμιευτές κάθε αναγνωριστικού Alloc-ID, η μονάδα λειτουργίας δυναμικής ανάθεσης εύρους ζώνης του τερματικού OLT εξάγει συμπέρασμα για τον όγκο των δεδομένων είτε μέσω της συλλογής των εισερχόμενων αναφορών κατάστασης είτε παρατηρώντας την άερνη μορφή της ροής ανόδου. Στη συνέχεια παρέχει τα δεδομένα στη μονάδα σχεδιασμού ροής ανόδου του τερματικού OLT, η οποία είναι υπεύθυνη για τη δημιουργία της χαρτογράφησης BWmap. Η χαρτογράφηση BWmap αποστέλλεται τελικά στις μονάδες ONU, ενσωματωμένη στην κίνηση της ροής καθόδου.

2.2.4.1. Λειτουργικές απαιτήσεις & μέθοδοι δυναμικής ανάθεσης εύρους ζώνης

Η δυναμική ανάθεση εύρους ζώνης στα δίκτυα G-PON περιλαμβάνει τις ακόλουθες λειτουργίες, οι οποίες εφαρμόζονται σε επίπεδο ξεχωριστού αναγνωριστικού Alloc-ID το οποίο έχει συσχετισθεί με κλάσεις υπηρεσιών T-CONT και παραμέτρους παρεχόμενων συνιστωσών εύρους ζώνης:

- Εξαγωγή συμπεράσματος για τον όγκο των δεδομένων προς μετάδοση στο λογικό ενταμιευτή.
- Ενημέρωση του στιγμιαίου εκχωρούμενου εύρους ζώνης που προκύπτει από το συναχθέν συμπέρασμα, στα πλαίσια των παραμέτρων παρεχόμενων συνιστωσών εύρους ζώνης.
- Έκδοση των κατανομών σύμφωνα με το ενημερωμένο στιγμιαίο εύρος ζώνης.
- Διαχείριση των λειτουργιών δυναμικής ανάθεσης εύρους ζώνης.

Το τερματικό OLT του δικτύου G-PON οφείλει υποχρεωτικά να υποστηρίζει δυναμική ανάθεση εύρους ζώνης. Ανάλογα με το μηχανισμό πληροφόρησης του όγκου δεδομένων των ενταμιευτών της μονάδας ONU, διακρίνονται δύο μέθοδοι δυναμικής ανάθεσης εύρους ζώνης :

- Δυναμική ανάθεση εύρους ζώνης με αναφορά κατάστασης, μέθοδος η οποία βασίζεται στη ρητή αναφορά της κατάστασης των ενταμιευτών που ζητείται από το τερματικό OLT και αποστέλλεται από τις μονάδες ONU.
- Δυναμική ανάθεση εύρους ζώνης με παρακολούθηση κίνησης, μέθοδος η οποία βασίζεται στην παρατήρηση από το τερματικό OLT των άεργων πλαισίων GEM και τη σύγκρισή τους με την αντίστοιχη χαρτογράφηση εύρους ζώνης.

2.2.4.2. Μαθηματικό μοντέλο δυναμικής ανάθεσης εύρους ζώνης

Συμβολισμοί μαθηματικού μοντέλου:

- **C (Mbps)**: η χωρητικότητα ροής ανόδου του παθητικού οπτικού δικτύου, η οποία ισούται με τον ονομαστικό ρυθμό μετάδοσης ζεύξης ανόδου αφαιρώντας τις επικεφαλίδες των πλαισίων και τις επιβαρύνσεις επικεφαλίδων του φυσικού στρώματος.
- **B (bits)**: φόρτος κίνησης, ο όγκος των δεδομένων εντός του ενταμιευτή, προσθέτοντας 40 bits που αναλογούν στην επικεφαλίδα πλαισίου GEM για κάθε πακέτο.
- **D**: περιγραφέας κίνησης κατανομής
- **R_F (Mbps)**: το πάγιο παρεχόμενο εύρος ζώνης
- **R_A (Mbps)**: το καθορισμένο παρεχόμενο εύρος ζώνης
- **R_M (Mbps)**: το μέγιστο παρεχόμενο εύρος ζώνης
- **R_{NA} (Mbps)**: το επιπρόσθετο, μη εγγυημένο εύρος ζώνης
- **R_{BE} (Mbps)**: το επιπρόσθετο εύρος ζώνης, καλύτερης προσπάθειας
- **X_{AB} (NA, BE, None)**: δείκτης εκλεξιμότητας για ανάθεση επιπρόσθετου εύρους ζώνης
- **S_{NA} (Mbps)**: το δυναμικό πλεόνασμα εύρους ζώνης, διαθέσιμο για ανάθεση ως επιπρόσθετο μη-εγγυημένο εύρος ζώνης
- **S_{BE} (Mbps)**: το δυναμικό πλεόνασμα εύρους ζώνης, διαθέσιμο για ανάθεση ως επιπρόσθετο εύρος ζώνης καλύτερης προσπάθειας

2.2.4.3 Περιγραφείς Κίνησης

Κάθε αναγνωριστικό Alloc-ID χαρακτηρίζεται από έναν περιγραφέα κίνησης, ο οποίος προσδιορίζει τις τρεις συνιστώσες παραμέτρους του εύρους ζώνης (πάγιο, καθορισμένο και μέγιστο εύρος ζώνης σε Mbps), καθώς και το δείκτη εκλεξιμότητας για ανάθεση επιπρόσθετου εύρους ζώνης (Μη εγγυημένο-NA, Καλύτερης προσπάθειας-BE, Καμία-None). Επομένως ο περιγραφέας κίνησης του αναγνωριστικού Alloc-ID i μπορεί να αναπαρασταθεί τυπικά:

$$D^i = \langle R_F^i, R_A^i, R_M^i, X_{AB}^i \rangle$$

- **Πάγιο Εύρος Ζώνης, $R_F \geq 0$:** αντιπροσωπεύει το μέρος της χωρητικότητας ζεύξης ανόδου που ανατίθεται στατικά σε ένα αναγνωριστικό Alloc-ID από το τερματικό OLT, ανεξάρτητα από την αναφορά κίνησής της ή το συνολικό φόρτο κίνησης στο δίκτυο.
- **Καθορισμένο Εύρος Ζώνης, $R_A \geq 0$:** αντιπροσωπεύει το μέρος της χωρητικότητας ζεύξης ανόδου που υποχρεούται να αναθέσει το τερματικό OLT σε ένα αναγνωριστικό Alloc-ID αν αυτό έχει κίνηση προς εξυπηρέτηση, ανεξάρτητα από το συνολικό φόρτο κίνησης στο δίκτυο. Αν ένα αναγνωριστικό Alloc-ID δεν έχει κίνηση προς εξυπηρέτηση, το τερματικό OLT δικαιούται να αναθέσει μερικά ή ολικά το συγκεκριμένο εύρος ζώνης σε άλλα υποψήφια αναγνωριστικά Alloc-ID.
- **Μέγιστο Εύρος Ζώνης, $R_M > 0$:** αντιπροσωπεύει το άνω όριο του συνολικού εύρους ζώνης που μπορεί να ανατεθεί στο συγκεκριμένο αναγνωριστικό Alloc-ID υπό οποιοσδήποτε συνθήκες κίνησης.

Ένας ορθά ορισμένος περιγραφέας κίνησης οφείλει να ικανοποιεί τους ακόλουθους περιορισμούς:

$$- R_M^i \geq R_F^i + R_A^i$$

$$- X_{AB}^i = NA, \text{μόνο αν } R_M^i > R_F^i + R_A^i > 0$$

$$- X_{AB}^i = BE, \text{μόνο αν } R_M^i > R_F^i + R_A^i \geq 0$$

Επιπλέον, η συνολική προδιαγραφή της κίνησης οφείλει να ικανοποιεί τη βασική συνθήκη ευστάθειας:

$$\sum_i (R_F^i + R_A^i) \leq C$$

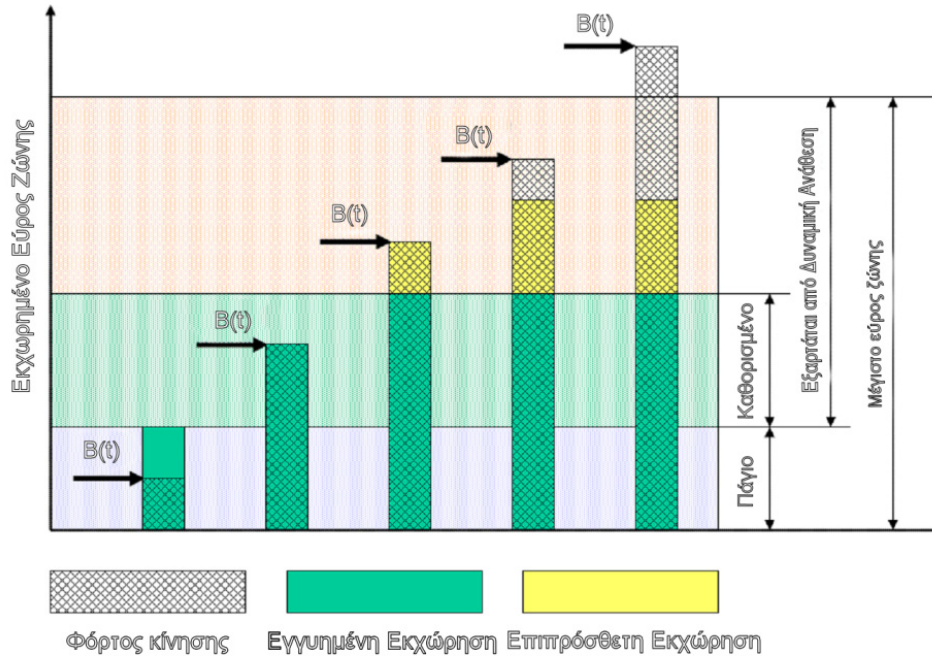
, όπου η άθροιση πραγματοποιείται στο σύνολο όλων των αναγνωριστικών Alloc-ID του δικτύου.

2.2.4.4 Συνιστώσες εύρους ζώνης

Το εύρος ζώνης $R^i(t) \geq 0$, το οποίο έχει ανατεθεί δυναμικά σε ένα αναγνωριστικό Alloc-ID υπό το παρόν μαθηματικό μοντέλο αποτελείται από τις συνιστώσες εγγυημένου (το άθροισμα

πάγιου και καθορισμένου) και επιπρόσθετου εύρους ζώνης. Το επιπρόσθετο εύρος ζώνης, αν υπάρχει, μπορεί να έχει μορφή μη εγγυημένου ή καλύτερης προσπάθειας. Επομένως:

- $R^i(t) = R_G^i(t) + R_{NA}^i(t)$ για τα αναγνωριστικά Alloc-ID που έχουν $X_{AB}^i = NA$
- $R^i(t) = R_G^i(t) + R_{BE}^i(t)$ για τα αναγνωριστικά Alloc-ID που έχουν $X_{AB}^i = BE$



Σχήμα 7: Οι συνιστώσες του συνολικά εκχωρημένου εύρους ζώνης

Ανάθεση εγγυημένου εύρους ζώνης: Εφ' όσον ικανοποιείται η βασική συνθήκη ευστάθειας $\sum_i (R_F^i + R_A^i) \leq C$, η συνιστώσα εγγυημένου εύρους ζώνης η οποία εκχωρείται δυναμικά είναι:

$$R_G^i(t) = \min \left\{ R_F^i + R_A^i; \max \{ R_F^i; B(t) \} \right\}$$

Η συγκεκριμένη συνιστώσα είναι διαθέσιμη για το αναγνωριστικό Alloc-ID i ανεξάρτητα από τη κατάσταση του συνολικού φόρτου κίνησης στο δίκτυο. Επομένως, το R_F^i είναι το κάτω όριο της συνιστώσας $R_G^i(t)$ και το $R_F^i + R_A^i$ το άνω όριο.

Ανάθεση μη-εγγυημένου επιπρόσθετου εύρους ζώνης: Το μη-εγγυημένο επιπρόσθετο εύρος ζώνης μπορεί να ανατεθεί δυναμικά από το τερματικό OLT σε ένα εκλέξιμο αναγνωριστικό Alloc-ID, κατ' αναλογία του αθροίσματος παγίου και εγγυημένου εύρους ζώνης του. Η ποσότητα του πλεονάσματος εύρους ζώνης, η οποία μπορεί να συμμετέχει στην ανάθεση μη-εγγυημένου επιπρόσθετου εύρους ζώνης είναι ίση με το μέρος της χωρητικότητας ροής ανόδου που παραμένει διαθέσιμη μετά το πέρας της ανάθεσης του εγγυημένου εύρους ζώνης σε όλα τα αναγνωριστικά Alloc-ID. Η ποσότητα δίνεται από την ακόλουθη έκφραση:

$$S_{NA}(t) = C - \sum_i \min \left\{ R_F^i + R_A^i; \max \{ R_F^i; B(t) \} \right\}$$

Το κριτήριο δίκαιης ανάθεσης μη-εγγυημένου επιπρόσθετου εύρους ζώνης: υπό συνθήκες συγκεκριμένου φόρτου κίνησης, οποιαδήποτε εκλέξιμα ($X_{AB}^i = NA$), μη κορεσμένα ($R_G(t) + R_{NA}(t) < \min\{R_M; B\}$) αναγνωριστικά Alloc-ID i και j θα λαμβάνουν επιπρόσθετο μη-εγγυημένο εύρος ζώνης που ικανοποιεί τη συνθήκη:

$$\frac{R_{NA}^i(t)}{R_F^i + R_A^i} = \frac{R_{NA}^j(t)}{R_F^j + R_A^j}$$

Για ένα εκλέξιμο αναγνωριστικό Alloc-ID του οποίου το δίκαιο μερίδιο μη-εγγυημένου επιπρόσθετου εύρους ζώνης οδηγεί σε κορεσμό ($R_G(t) + R_{NA}(t) \geq \min\{R_M; B(t)\}$), η δυναμική ανάθεση εύρους ζώνης θα ικανοποιεί τη συνθήκη $R(t) = \min\{R_M; B(t)\}$.

Ανάθεση επιπρόσθετου εύρους ζώνης καλύτερης προσπάθειας: Το επιπρόσθετο εύρος ζώνης καλύτερης προσπάθειας μπορεί να ανατεθεί δυναμικά από το τερματικό OLT σε ένα εκλέξιμο αναγνωριστικό Alloc-ID, κατ' αναλογία του μη εγγυημένου μέρους του μέγιστου καθορισμένου εύρους ζώνης του συγκεκριμένου Alloc-ID. Η ποσότητα του πλεονάσματος εύρους ζώνης, η οποία μπορεί να συμμετέχει στην ανάθεση επιπρόσθετου εύρους ζώνης καλύτερης προσπάθειας είναι ίση με το μέρος της χωρητικότητας ροής ανόδου που παραμένει διαθέσιμη μετά το πέρας της ανάθεσης του εγγυημένου και του μη-εγγυημένου επιπρόσθετου εύρους ζώνης σε όλα τα αναγνωριστικά Alloc-ID. Η ποσότητα δίνεται από την ακόλουθη έκφραση:

$$S_{BE}(t) = C - \sum_{i \in \{X_{AB}^i = NA\}} \min\{R_M^i; \max\{R_F^i; B(t)\}\} - \sum_{i \in \{X_{AB}^i \neq NA\}} \min\{R_F^i + R_A^i; \max\{R_F^i; B(t)\}\}$$

Το κριτήριο δίκαιης ανάθεσης επιπρόσθετου εύρους ζώνης καλύτερης προσπάθειας: υπό συνθήκες συγκεκριμένου φόρτου κίνησης, οποιαδήποτε εκλέξιμα ($X_{AB}^i = BE$), μη κορεσμένα ($R_G(t) + R_{BE}(t) < \min\{R_M; B\}$) αναγνωριστικά Alloc-ID i και j θα λαμβάνουν επιπρόσθετο εύρος ζώνης καλύτερης προσπάθειας που ικανοποιεί τη συνθήκη:

$$\frac{R_{BE}^i(t)}{R_M^i - (R_F^i + R_A^i)} = \frac{R_{BE}^j(t)}{R_M^j - (R_F^j + R_A^j)}$$

Για ένα εκλέξιμο αναγνωριστικό Alloc-ID, του οποίου το δίκαιο μερίδιο επιπρόσθετου εύρους ζώνης καλύτερης προσπάθειας οδηγεί σε κορεσμό ($R_G(t) + R_{BE}(t) \geq \min\{R_M; B(t)\}$), η δυναμική ανάθεση εύρους ζώνης θα ικανοποιεί τη συνθήκη $R(t) = \min\{R_M; B(t)\}$.

Ιεράρχηση συνιστωσών εκχωρημένου εύρους ζώνης: Το παρόν μαθηματικό μοντέλο εισάγει μία αυστηρή ιεραρχία προτεραιοτήτων για τις συνιστώσες του εκχωρημένου εύρους ζώνης:

- 1) Πάγιο εύρος ζώνης (υψηλότερη προτεραιότητα)
- 2) Καθορισμένο εύρος ζώνης
- 3) Μη-εγγυημένο επιπρόσθετο εύρος ζώνης
- 4) Επιπρόσθετο εύρος ζώνης καλύτερης προσπάθειας (χαμηλότερη προτεραιότητα)

Αρχικά, το τερματικό OLT εκχωρεί το πάγιο εύρος ζώνης σε όλα τα αναγνωριστικά Alloc-ID στο παθητικό οπτικό δίκτυο, ανεξάρτητα από το φόρτο και τις συνολικές συνθήκες κίνησης στο δίκτυο. Στη συνέχεια το τερματικό OLT ολοκληρώνει την εκχώρηση της συνιστώσας εγγυημένου εύρους ζώνης, αναθέτοντας το καθορισμένο εύρος ζώνης στα αναγνωριστικά Alloc-ID μέχρι το καθορισμένο επίπεδο R_A ή μέχρι να ικανοποιηθεί η ζήτηση. Ύστερα το τερματικό OLT αναθέτει τη συνιστώσα μη-εγγυημένου επιπρόσθετου εύρους ζώνης στα εκλέξιμα, μη κορεσμένα αναγνωριστικά Alloc-ID μέχρι είτε όλα τα αναγνωριστικά Alloc-ID να φτάσουν το επίπεδο κορεσμού τους ή το πλεόνασμα εύρους ζώνης $S_{NA}(t)$ να εξαντληθεί. Τέλος, το τερματικό OLT αναθέτει τη συνιστώσα επιπρόσθετου εύρους ζώνης καλύτερης προσπάθειας στα εκλέξιμα, μη κορεσμένα αναγνωριστικά Alloc-ID.

2.2.5 Περιγραφείς κίνησης και τύποι κλάσεων T-Cont

Εν γένει, οι συνιστώσες εύρους ζώνης και εκλεξιμότητας ενός περιγραφέα κίνησης ενός αναγνωριστικού Alloc-ID μπορεί να έχουν διαφορετική αλληλοσυσχέτιση μεταξύ τους. Οι τέσσερις συνδυασμοί παραμέτρων περιγραφέν κίνησης αναγνωριστικών Alloc-ID που στην πράξη αποδεικνύονται πιο σημαντικοί από τους υπόλοιπους, αναγνωρίζονται και συσχετίζονται με τους συγκεκριμένους τύπους κλάσης T-Cont. Ο ορισμός των 4 κλάσεων T-Cont συγκεντρώνεται στον ακόλουθο πίνακα:

	<i>T-Cont 1</i>	<i>T-Cont 2</i>	<i>T-Cont 3</i>	<i>T-Cont 4</i>
<i>Πάγιο εύρος ζώνης</i>	R_F	-	-	-
<i>Καθορισμένο εύρος ζώνης</i>	-	R_A	R_A'	-
<i>Μέγιστο εύρος ζώνης</i>	$R_M = R_F$	$R_M > R_A$	$R_M' > R_A'$	R_M
<i>Εκλεξιμότητα για επιπρόσθετο εύρος ζώνης</i>	Καμία	Μη εγγυημένο	Μη εγγυημένο	Καλύτερης προσπάθειας

Πίνακας 1: Περιγραφείς κίνησης κλάσεων T-Cont

Κλάση T-Cont 1: χαρακτηρίζεται μόνο από τη συνιστώσα πάγιου εύρους ζώνης R_F , ενώ δεν είναι εκλέξιμη για επιπρόσθετο εύρος ζώνης. Είναι κατάλληλη για κίνηση σταθερού ρυθμού (ή μεταβλητού ρυθμού με σχετικά χαμηλό όριο), η οποία είναι ευαίσθητη στην καθυστέρηση και τη διακύμανση καθυστέρησης.

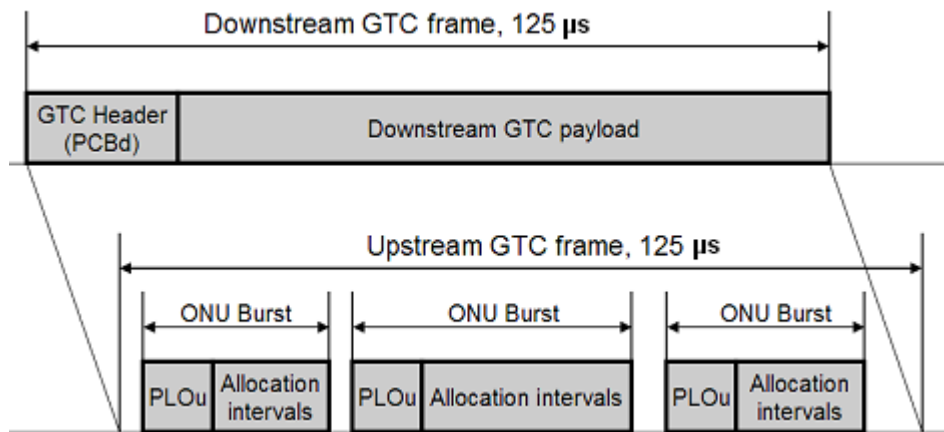
Κλάση T-Cont 2: χαρακτηρίζεται από τη συνιστώσα καθορισμένου εύρους ζώνης R_A και εκλεξιμότητα για μη εγγυημένο επιπρόσθετο εύρος ζώνης με όριο την τιμή $R_M > R_A$. Είναι κατάλληλη για εκρηκτική κίνηση μεταβλητού ρυθμού η οποία απαιτεί εγγύηση ενός ελάχιστου μέσου ρυθμού.

Κλάση T-Cont 3: παρόμοια με την κλάση T-Cont 2, αλλά με διαφορετικές τιμές R_A' και R_M' . Ο συνδυασμός των κλάσεων T-Cont 2 και 3 εισάγει τη δυνατότητα για υποστήριξη υπηρεσιών που παράγουν κίνηση με παρόμοιες απαιτήσεις, αλλά διαφορετική προτεραιότητα.

Κλάση T-Cont 4: χαρακτηρίζεται από εκλεξιμότητα για επιπρόσθετο εύρος ζώνης καλύτερης προσπάθειας, χωρίς παρεχόμενες συνιστώσες πάγιου ή καθορισμένου εύρους ζώνης. Είναι κατάλληλη για εκρηκτική κίνηση μεταβλητού ρυθμού που είναι ανεκτική στην καθυστέρηση.

2.2.6 Πλαισίωση στο στρώμα σύγκλισης μετάδοσης του πρωτοκόλλου G-PON

Τα πλαίσια σύγκλισης μετάδοσης G-PON ροής ανόδου και καθόδου εμφανίζουν αρκετές διαφορές αναφορικά με τη δομή τους:

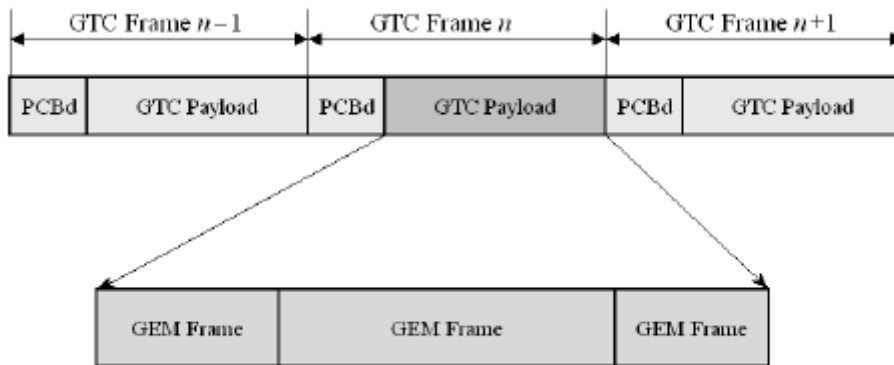


Σχήμα 8: Η δομή του πλαισίου σύγκλισης μετάδοσης G-PON ροής ανόδου και καθόδου

Το πλαίσιο σύγκλισης μετάδοσης του G-PON στη ροή καθόδου αποτελείται από το τμήμα ελέγχου φυσικού στρώματος καθόδου (Physical Control Block downstream, PCBd) και το τμήμα φόρτου δεδομένων (GTC payload). Επιπλέον παρέχει το κοινό σημείο χρονικής αναφοράς για ολόκληρο το δίκτυο PON και τη κοινή σηματοδότηση ελέγχου για τη ροή ανόδου. Το πλαίσιο σύγκλισης μετάδοσης του G-PON στη ροή ανόδου περιέχει πολλαπλές εκρήξεις μετάδοσης. Κάθε έκρηξη ροής ανόδου αποτελείται από την επιβάρυνση λόγω επικεφαλίδας φυσικού στρώματος ροής ανόδου (Physical Layer Overhead upstream, PLOu) και από ένα ή περισσότερα διαστήματα κατανομής εύρους ζώνης συσχετισμένα με ένα συγκεκριμένο αναγνωριστικό Alloc-ID.

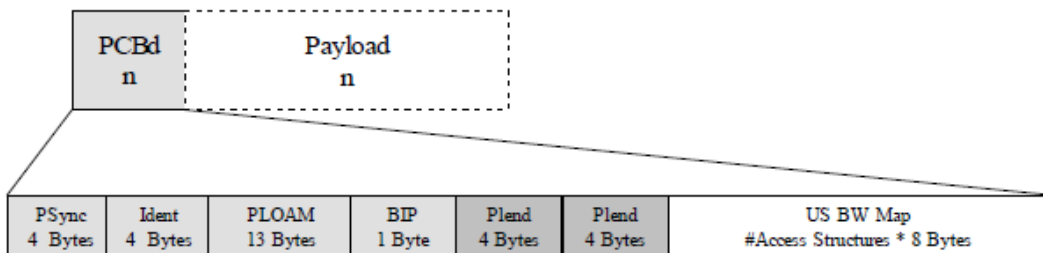
2.2.6.1 Δομή πλαισίου σύγκλισης μετάδοσης του G-PON στη ροή καθόδου

Το πλαίσιο ροής καθόδου έχει διάρκεια 125μs και μήκος 38880 bytes, το οποίο αντιστοιχεί σε ρυθμό μετάδοσης ροής καθόδου 2.48832 Gbit/s. Το μήκος του τμήματος PCBd εξαρτάται από το πλήθος των δομών κατανομής ανά πλαίσιο. Η δομή του πλαισίου σχηματικά:



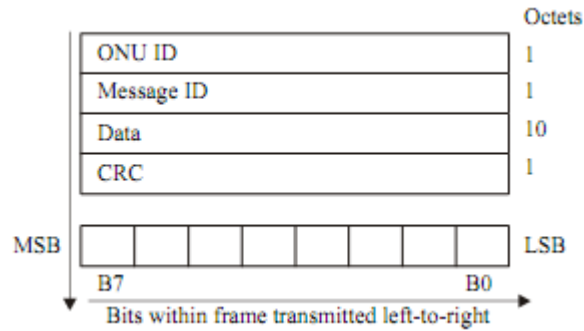
Σχήμα 9: Τα ενθυλακωμένα πλαίσια GEM στο τμήμα φόρτου δεδομένων GTC

Το πλήρες τμήμα PCBd αποστέλλεται από το τερματικό OLT με πολλαπλή εκπομπή προς όλες τις μονάδες ONU, οι οποίες στη συνέχεια ενεργούν με βάση τις σχετικές πληροφορίες που περιέχονται σε αυτό.



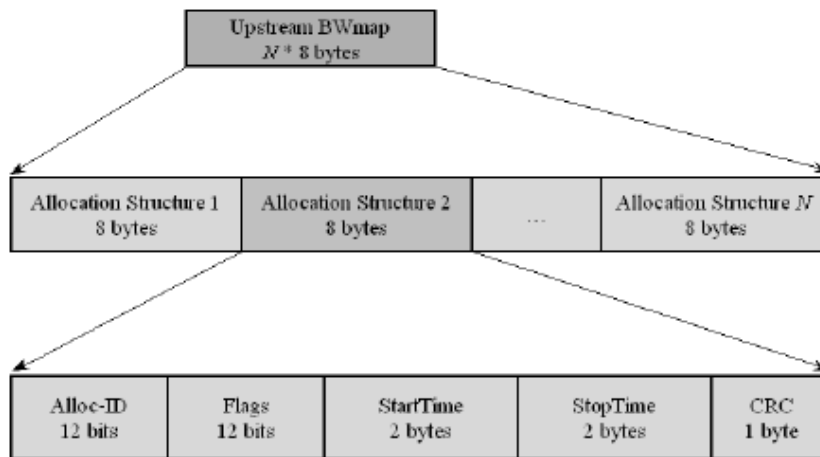
Σχήμα 10: Τα πεδία του τμήματος PCBd

- **Φυσικού συγχρονισμού (PSync):** πρόκειται για ένα καθορισμένο πεδίο μήκους 32 bit με το οποίο ξεκινάει κάθε τμήμα PCBd. Η μονάδα ONU μπορεί να χρησιμοποιήσει αυτό το πεδίο για να ανιχνεύσει την έναρξη του πλαισίου. Η κωδικοποίηση του πεδίου είναι 0xB6AB31E0.
- **Αναγνώρισης (Ident):** το συγκεκριμένο πεδίο μήκους 4 byte χρησιμοποιείται για την υπόδειξη μεγαλύτερων δομών πλαισίων.
- **Λειτουργίας, διαχείρισης και συντήρησης φυσικού στρώματος καθόδου (PLOAMd):** το συγκεκριμένο πεδίο μήκους 13 byte περιέχει το μήνυμα PLOAM, μέσω του οποίου πραγματοποιούνται λειτουργίες διαχείρισης του στρώματος σύγκλισης μετάδοσης, όπως είναι η ενεργοποίηση μίας μονάδας ONU, η διαμόρφωση της χρησιμοποιούμενης κρυπτογράφησης, η διαχείριση κλειδιών και η σηματοδότηση συναγερμών του συστήματος.



Σχήμα 11: Η δομή του μηνύματος PLOAMd

- **Ισότητας διαστρωμένων bit (Bit Interleaved Parity, BIP):** περιέχει την ισότητα των διαστρωμένων bit όλων των byte που έχουν μεταδοθεί από το τελευταίο πεδίο BIP, εξαιρουμένης της ισότητας FEC (αν υπάρχει).
- **Μήκους τμήματος φόρτου δεδομένων καθόδου (PLend) :** προσδιορίζει το μήκος του πεδίου χαρτογράφησης BWmap και αποστέλλεται δύο φορές για αποφυγή λαθών.
- **Χαρτογράφησης εύρους ζώνης (BWmap):** είναι μια βαθμωτή διάταξη αποτελούμενη από δομές κατανομής των 8 byte. Κάθε καταχώρηση σε αυτή τη διάταξη εκπροσωπεί μια ξεχωριστή κατανομή εύρους ζώνης σε μια συγκεκριμένη κλάση T-Cont και περιλαμβάνει το αναγνωριστικό Alloc-ID της οντότητας-παραλήπτη, ορισμένες υποδείξεις (flags) για τη λειτουργία της αντίστοιχης μετάδοσης ανόδου, τους χρόνους έναρξης και λήξης (StartTime / StopTime) της μετάδοσης και τέλος τον έλεγχο κυκλικού πλεονασμού (CRC):



Σχήμα 12: Η δομή της χαρτογράφησης εύρους ζώνης ροής ανόδου

2.2.6.2 Δομή πλαισίου σύγκλισης μετάδοσης του G-PON στη ροή ανόδου

Η διάρκεια του πλαισίου σύγκλισης μετάδοσης του G-PON στη ροή ανόδου είναι 125 μ s, διάρκεια που αντιστοιχεί σε μήκος πλαισίου 19440 byte για συστήματα G-PON με ρυθμό ανόδου 1.24416 Gbit/s και 38880 byte για ρυθμό ανόδου 2.48832 Gbit/s. Κάθε πλαίσιο ροής ανόδου περιέχει εκρήξεις μετάδοσης προερχόμενες από μία ή περισσότερες μονάδες ONU.

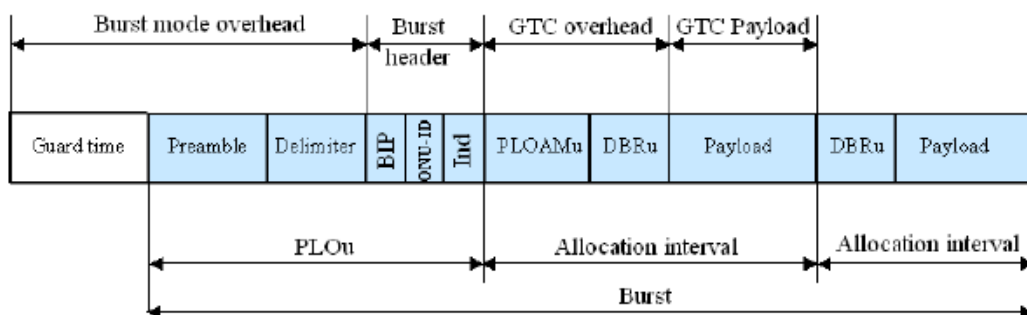
Κάθε έκρηξη ροής ανόδου περιέχει το τμήμα επιβάρυνσης λόγω επικεφαλίδας φυσικού στρώματος ροής ανόδου (Physical Layer Overhead upstream, PLOu) και ένα ή περισσότερα διαστήματα κατανομής εύρους ζώνης συσχετισμένα με ένα συγκεκριμένο αναγνωριστικό Alloc-ID. Η χαρτογράφηση BWmap υπαγορεύει τη διευθέτηση των εκρήξεων εντός του πλαισίου και τα διαστήματα κατανομής εύρους ζώνης σε κάθε έκρηξη. Κάθε διάστημα κατανομής εύρους ζώνης ελέγχεται από μία συγκεκριμένη δομή κατανομής της χαρτογράφησης BWmap.

Ένα διάστημα κατανομής εύρους ζώνης μπορεί να περιέχει δύο είδη πεδίων της επιβάρυνσης στρώματος σύγκλισης μετάδοσης:

- i. Μήνυμα λειτουργίας, διαχείρισης και συντήρησης φυσικού στρώματος ανόδου (PLOAMu, μόνο προς τα προκαθορισμένα αναγνωριστικά Alloc-ID)
- ii. Δυναμική αναφορά εύρους ζώνης ροής ανόδου (DBRu)

Το τερματικό OLT χρησιμοποιεί το πεδίο υποδείξεων (flags) της δομής κατανομής για να υποδείξει αν θα πρέπει το μήνυμα PLOAMu και το πεδίο DBRu να συμπεριληφθούν στο αντίστοιχο διάστημα κατανομής εύρους ζώνης. Η μετάδοση του μηνύματος PLOAMu θα πρέπει να ζητείται μόνο σε διαστήματα κατανομής που σχετίζονται με τα προκαθορισμένα αναγνωριστικά Alloc-ID των μονάδων ONU.

Στο ακόλουθο σχήμα απεικονίζονται οι λεπτομέρειες των επιβαρύνσεων του φυσικού και του στρώματος σύγκλισης μετάδοσης, οι οποίες σχετίζονται με μία έκρηξη μετάδοσης ροής ανόδου. Κάθε έκρηξη ξεκινά με το τμήμα επιβάρυνσης φυσικού στρώματος ροής ανόδου (PLOu) το οποίο αποτελείται από το προοίμιο (preamble), τον οριοθέτη (delimiter) και την επικεφαλίδα εκρήξεως (burst header) μήκους 3 byte.



Σχήμα 13: Η δομή μιας έκρηξης μετάδοσης ροής ανόδου

Οι επιβαρύνσεις PLOu και λειτουργίας έκρηξης (burst mode overhead) είναι διακεκριμένες αλλά και μερικώς επικαλυπτόμενες έννοιες. Σε αντίθεση με την επιβάρυνση PLOu που αναφέρεται στα πεδία που πραγματικά μεταδίδονται από τη μονάδα ONU, η επιβάρυνση λειτουργίας έκρηξης αναφέρεται στα χρονικά διαστήματα που απαιτούνται για να επιτευχθεί ορθή απομόνωση και οριοθέτηση των εκρήξεων. Συντίθεται από το χρόνο φύλαξης (guard time), το χρόνο προοιμίου (preamble time) και το χρόνο οριοθέτησης (delimiter time), οι οποίοι παραμένουν αμετάβλητοι για όλες τις μονάδες ONU στο δίκτυο PON, όσο αυτό παραμένει σε λειτουργία.

Όσο το τερματικό OLT παρέχει κατανομές εύρους ζώνης σε συγκεκριμένα αναγνωριστικά Alloc-ID, οφείλει να λαμβάνει υπόψιν τις απαιτήσεις σε εύρος ζώνης και καθυστέρηση που προκύπτουν από τις αντίστοιχες επιβαρύνσεις των πεδίων PLOAMu και DBRu. Όταν δημιουργείται μια χαρτογράφηση BWmap συνδυάζοντας τις ξεχωριστές δομές κατανομών, το τερματικό OLT οφείλει να παρέχει επαρκείς παραχωρήσεις για τους χρόνους επιβαρύνσεων λειτουργίας έκρηξης και τις επικεφαλίδες των εκρήξεων.

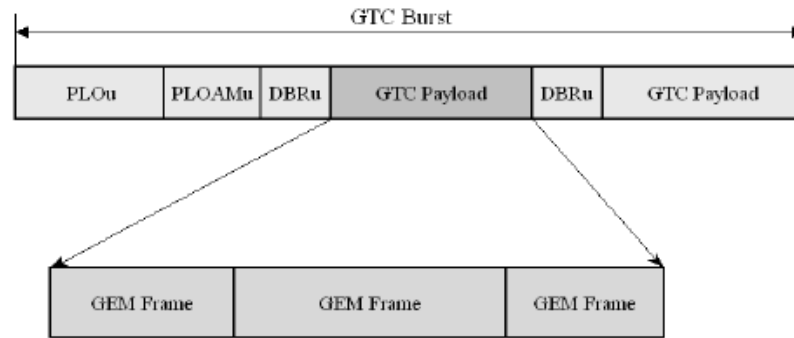
Η απαίτηση να εισάγεται το τμήμα PLOu στην αρχή μιας ακολουθίας διαστημάτων κατανομής υπονοείται στον κανονισμό μεταξύ των ίδιων των κατανομών. Ο κανόνας είναι ότι το τμήμα PLOu οφείλει να αποστέλλεται στην αρχή κάθε έκρηξης, ή ισοδύναμα, κάθε φορά που μια μονάδα ONU αναλαμβάνει το μέσο μετάδοσης του δικτύου PON από την προηγούμενη. Οι συνεχόμενες κατανομές (ο Χρόνος Λήξης της προηγούμενης δομής κατανομής είναι ακριβώς 1 byte μικρότερος από το Χρόνο Έναρξης της επόμενης) θεωρούνται πως ανήκουν στην ίδια έκρηξη και δεν απαιτείται η εισαγωγή του τμήματος PLOu ενδιάμεσα. Αντιθέτως, η απαίτηση για συνεχόμενες κατανομές απαγορεύει στο τερματικό OLT να ορίζει κενά ανάμεσα στις κατανομές της ίδιας μονάδας ONU. Προκειμένου να διατηρηθεί η συνδεσιμότητα με τις μονάδες ONU, το τερματικό OLT οφείλει να επιδιώκει την κατανομή μίας μετάδοσης στη ροή ανόδου σε κάθε ONU σε κάποιο ορισμένο χρονικό διάστημα, η διάρκεια του οποίου καθορίζεται από τις παραμέτρους λειτουργίας της μονάδας ONU.

Για ζεύξη ανόδου 1.24416 Gbit/s, οι ελάχιστοι χρόνοι φύλαξης, προοιμίου και οριοθέτησης είναι 32, 44 και 20 bit αντίστοιχα. Τα πεδία που περιέχονται στην επικεφαλίδα της έκρηξης είναι:

- **Πεδίο ισότητας διαστρωμένων bit (Bit Interleaved Parity, BIP):** περιέχει την ισότητα των διαστρωμένων bit όλων των byte που έχουν μεταδοθεί από το τελευταίο πεδίο BIP, εξαιρουμένων των byte του προοιμίου και της οριοθέτησης, καθώς και της ισότητας FEC (αν υπάρχει).
- **Πεδίο αναγνωριστικού ONU-ID:** έχει μήκος 8 bit και περιέχει το μοναδικό αναγνωριστικό της μονάδας ONU που αποστέλλει τη συγκεκριμένη μετάδοση.
- **Πεδίο ενδείξεων (Ind):** παρέχει στο τερματικό OLT αναφορές κατάστασης της μονάδας ONU σε πραγματικό χρόνο.

Την επικεφαλίδα κάθε έκρηξης ακολουθούν ένα ή περισσότερα διαστήματα κατανομών, τα οποία αποτελούνται από τα πεδία PLOAMu, DBRu και το τμήμα φόρτου δεδομένων (payload):

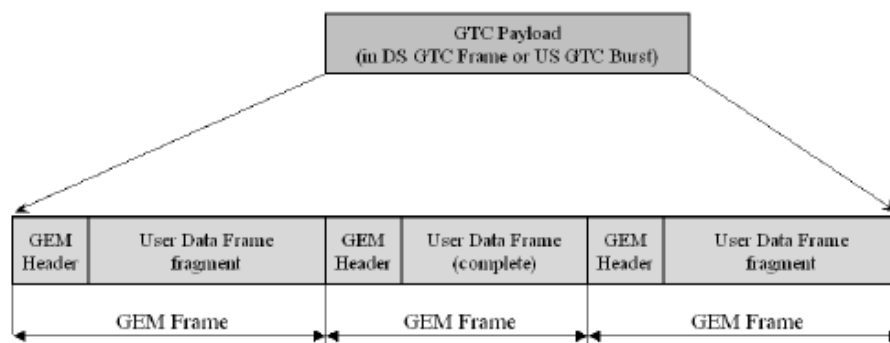
- **Πεδίο PLOAMu:** περιέχει το μήνυμα PLOAMu. Έχει μήκος 13 byte και αποστέλλεται όταν υποδεικνύεται από το πεδίο υποδείξεων (flags) της δομής κατανομής.
- **Πεδίο DBRu:** η συγκεκριμένη δομή μήκους 1 ή 2 byte περιέχει πληροφορίες συνδεδεμένες με μία οντότητα T-CONT και όχι με τη μονάδα ONU ως σύνολο. Πιο αναλυτικά, περιέχει την κατάσταση του όγκου κίνησης της συγκεκριμένης οντότητας T-CONT. Παρόμοια με το πεδίο PLOAMu, αποστέλλεται όταν υποδεικνύεται από το πεδίο υποδείξεων (flags) της χαρτογράφησης BWmap.
- **Τμήμα φόρτου δεδομένων (payload):** χρησιμοποιείται για τη μεταφορά των πλαισίων GEM. Το μήκος του τμήματος υπολογίζεται από τη διαφορά του μεγέθους των ζητούμενων επιβαρύνσεων από τη διάρκεια της κατανομής.



Σχήμα 14: Τα ενθυλακωμένα πλαίσια GEM σε μια έκρηξη μετάδοσης ροής ανόδου

2.2.6.3 Χαρτογράφηση πλαισίων GEM στο τμήμα φόρτου δεδομένων

Η κίνηση GEM μεταφέρεται από το πρωτόκολλο σύγκλισης μετάδοσης του G-PON με διαφανή τρόπο. Το πρωτόκολλο GEM έχει διπλή λειτουργία: παρέχει οριοθέτηση των πλαισίων δεδομένων του χρήστη (τόσο στη ροή καθόδου όσο και στη ροή ανόδου) και επιπλέον αναγνώριση θύρας για την πολυπλεξία.

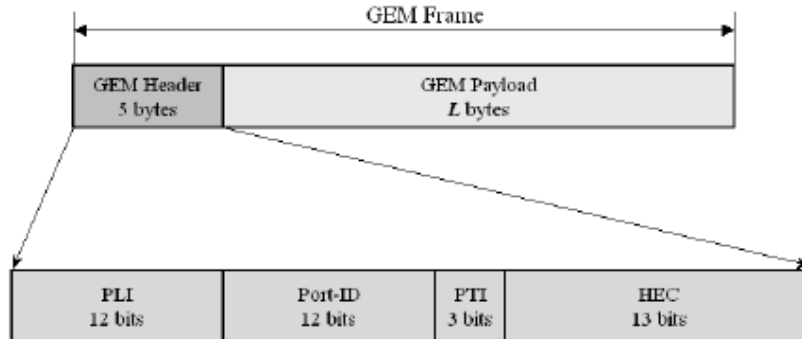


Σχήμα 15: Η δομή ενός ενθυλακωμένου πλαισίου GEM

Η επικεφαλίδα του πλαισίου GEM έχει μήκος 40 bit και περιλαμβάνει τα ακόλουθα πεδία:

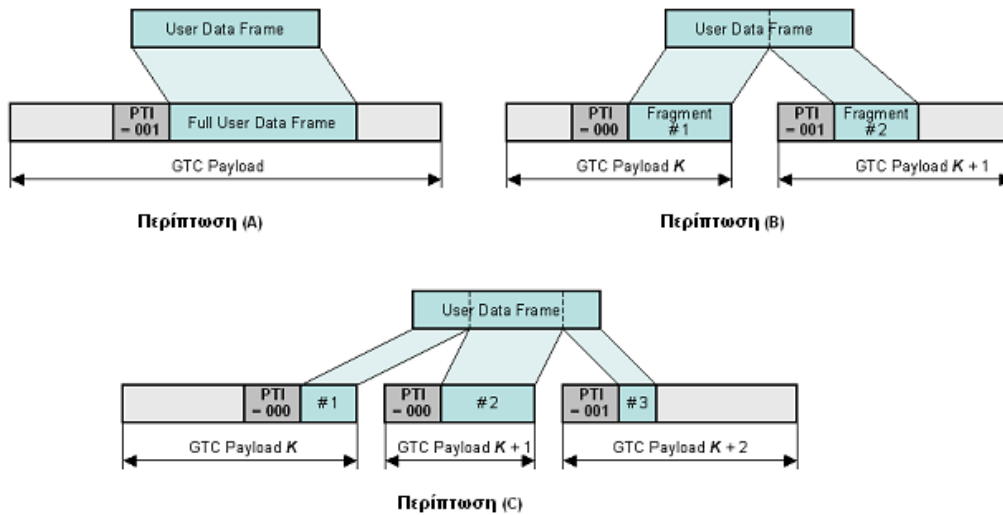
- **Ένδειξη μήκους τμήματος φόρτου δεδομένων (PLI):** υποδεικνύει το μήκος L, σε byte, του τμήματος φόρτου δεδομένων που ακολουθεί την επικεφαλίδα. Χρησιμοποιείται για την εύρεση της επόμενης επικεφαλίδας της ροής, στα πλαίσια της οριοθέτησης. Το μήκος 12 bit του πεδίου επιτρέπει τμήματα (fragments) μεγέθους μέχρι 4095 byte. Αν τα πλαίσια δεδομένων του χρήστη έχουν μέγεθος μεγαλύτερο από αυτό, απαιτείται τεμαχισμός τους σε τμήματα των 4095 byte ή μικρότερα.
- **Αναγνωριστικό θύρας (Port-ID):** χρησιμοποιείται για την παροχή 4096 μοναδικών αναγνωριστικών κίνησης στο δίκτυο PON, για τις ανάγκες της πολυπλεξίας κίνησης. Κάθε αναγνωριστικό θύρας περιέχει μία ροή δεδομένων του χρήστη. Είναι δυνατή η μετάδοση μίας ή περισσότερων αναγνωριστικών θύρας εντός ενός αναγνωριστικού Alloc-ID/ κλάσης T-CONT.

- **Ένδειξη τύπου τμήματος φόρτου δεδομένων (PTI):** υποδεικνύει τον τύπο του περιεχομένου του φόρτου δεδομένων και την κατάλληλη μεταχείριση.
- **Έλεγχος σφαλμάτων επικεφαλίδας (HEC):** παρέχει ανίχνευση σφαλμάτων και λειτουργίες διόρθωσης για την επικεφαλίδα.



Σχήμα 16: Τα πεδία της επικεφαλίδας του πλαισίου GEM

Τεμαχισμός πλαισίων χρήστη: Επειδή τα πλαίσια δεδομένων του χρήστη είναι τυχαίου μήκους, το πρωτόκολλο GEM οφείλει να υποστηρίζει τη λειτουργία τεμαχισμού των πλαισίων, ώστε να επιτρέπεται η εισαγωγή της επικεφαλίδας GEM στην αρχή κάθε τμήματος φόρτου δεδομένων. Ο τεμαχισμός είναι δυνατό να συμβεί τόσο στην κατεύθυνση καθόδου όσο και στην κατεύθυνση ανόδου.



Σχήμα 17: Τεμαχισμένα πλαίσια δεδομένων χρήστη

ΚΕΦΑΛΑΙΟ 3^ο: Περιβάλλον προσομοίωσης OPNET & το μοντέλο του δικτύου G-PON

3.1 Το περιβάλλον προσομοίωσης OPNET

Το περιβάλλον προσομοίωσης OPNET είναι ένα λογισμικό μοντελοποίησης, το οποίο προσφέρει τα απαραίτητα εργαλεία για το σχεδιασμό και την ανάλυση δικτύων επικοινωνιών, συσκευών, πρωτοκόλλων και εφαρμογών. Μέσω της ανάλυσης των προσομοιωμένων συστημάτων είναι δυνατό να συγκριθεί η επίδραση διαφορετικών τεχνολογιών στην ολική συμπεριφορά ενός δικτύου, να μελετηθεί η συμπεριφορά και να μετρηθεί η απόδοσή του. Με την ενσωμάτωση ενός σημαντικού εύρους πρωτοκόλλων και τεχνολογιών και ταυτόχρονα περιλαμβάνοντας ένα ισχυρό περιβάλλον ανάπτυξης, καθιστά δυνατή τη μοντελοποίηση οποιουδήποτε τύπου δικτύου και δικτυακής τεχνολογίας.

3.1.1 Επίπεδα μοντελοποίησης

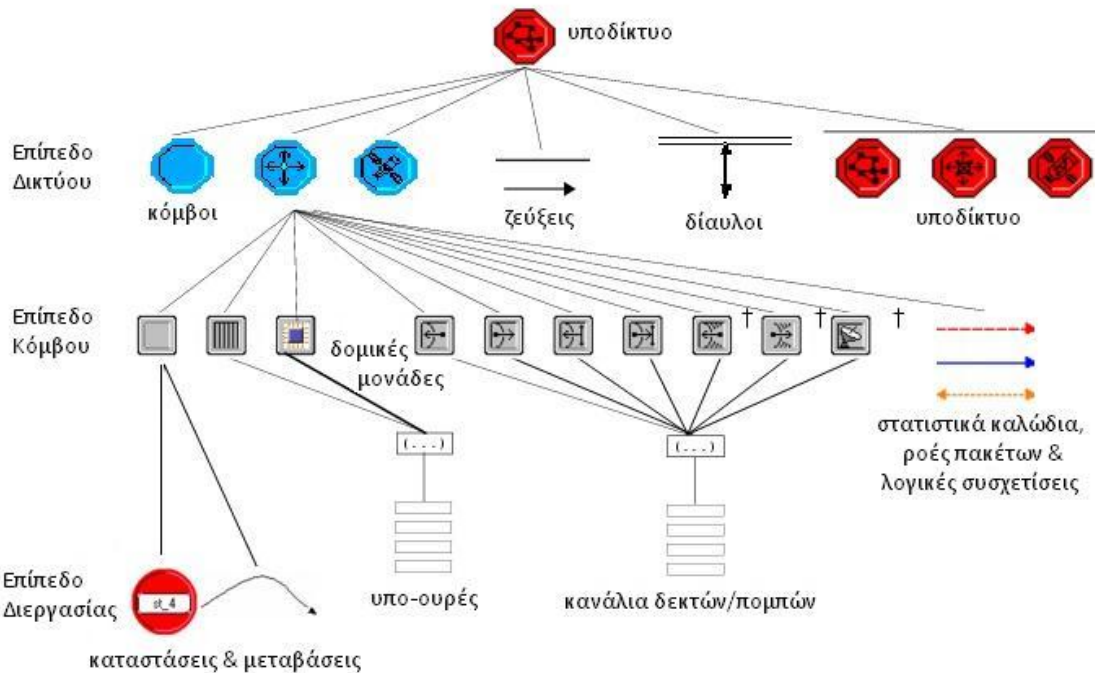
Το περιβάλλον προσομοίωσης OPNET διαχωρίζει την πλειοψηφία των προδιαγραφών των μοντέλων σε ένα σύνολο από τέσσερα περιβάλλοντα, τα οποία ονομάζονται επίπεδα μοντελοποίησης (modeling domains). Τα δίκτυα επικοινωνιών και τα καταμεμημένα συστήματα κατά κανόνα περιλαμβάνουν μεγάλο εύρος τεχνολογιών, που κυμαίνονται από υλικό χαμηλού επιπέδου έως λογισμικό υψηλού επιπέδου. Μία επιτυχημένη μοντελοποίηση ενός τέτοιου συστήματος οφείλει να αναπαριστά κάθε ένα από αυτά τα υποσυστήματα καθώς και την μεταξύ τους αλληλεπίδραση, με επίπεδο λεπτομέρειας επαρκές για την εξαγωγή των ζητούμενων συμπερασμάτων αναφορικά με τη συμπεριφορά και την απόδοση του συστήματος. Η προσέγγιση του περιβάλλοντος προσομοίωσης OPNET ως προς τη διαδικασία της μοντελοποίησης βασίζεται στα τρία τυπικά στοιχεία που αντιπροσωπεύουν τα διακριτά επίπεδα που εμφανίζονται στα δίκτυα επικοινωνιών. Ένα τέταρτο στοιχείο παρέχει τις απαραίτητες λειτουργίες επικοινωνίας για προσομοίωση σε συνεργασία με εξωτερικά εργαλεία. Τα τέσσερα επίπεδα μοντελοποίησης είναι:

i. Επίπεδο Δικτύου (network domain): περιέχει τις προδιαγραφές ενός συστήματος από τη σκοπιά συσκευών υψηλού επιπέδου, οι οποίες ονομάζονται κόμβοι, καθώς και των ζεύξεων επικοινωνιών μεταξύ τους.

ii. Επίπεδο Κόμβου (node domain): περιέχει τις προδιαγραφές των δυνατοτήτων των κόμβων, από τη σκοπιά των εφαρμογών, της επεξεργασίας, της αναμονής και των διεπαφών επικοινωνίας.

iii. Επίπεδο Διεργασίας (process domain): περιέχει τις προδιαγραφές της συμπεριφοράς των διεργασιών που εκτελούνται στο εσωτερικό των κόμβων του συστήματος. Είναι δυνατό να ορισθούν πλήρως παραμετροποιημένες διεργασίες και αλγόριθμοι λήψης αποφάσεων.

iv. Επίπεδο Εξωτερικού Συστήματος (external system domain): περιέχει τις προδιαγραφές των μηχανισμών επικοινωνίας που επιτρέπουν στο περιβάλλον προσομοίωσης OPNET να περιλαμβάνει μοντέλα εξωτερικού κώδικα ή συστήματα παράλληλης προσομοίωσης.



Σχήμα 18: Τα επίπεδα μοντελοποίησης του περιβάλλοντος OPNET

Για την περιγραφή ενός συστήματος το περιβάλλον προσομοίωσης OPNET χρησιμοποιεί δύο θεμελιώδεις μεθόδους:

- **Αντικείμενα (objects):** αντιπροσωπεύουν δομές συστημάτων μέσω της αποσύνθεσης ενός συστήματος σε μικρότερα δομικά κομμάτια και εκφράζοντας τις μεταξύ τους σχέσεις.
- **Διαδικασίες (procedures):** παρέχουν έναν τρόπο έκφρασης συμπεριφοράς, η οποία μπορεί να είναι δυναμική και προσαρμοστική.

Οι περιγραφές συμπεριφοράς οφείλουν να μεριμνούν για μεγάλο εύρος συνθηκών, συμπεριλαμβάνοντας μεταβλητές καταστάσεις του συστήματος και απρόβλεπτα γεγονότα. Επιπλέον, σύνθετες ενέργειες οφείλουν να προσδιορισθούν για την αντιμετώπιση τέτοιων συνθηκών. Η συγκεκριμένη προσέγγιση μοντελοποίησης παρέχει ένα φυσικό τρόπο ενσωμάτωσης των δύο προδιαγραφών σε ένα ενοποιημένο μοντέλο συστήματος.

3.1.2 Αντικείμενα και μοντέλα

Η σχέση μεταξύ ενός αντικειμένου και ενός μοντέλου είναι θεμελιώδης για όλες τις εργασίες προδιαγραφών που πραγματοποιούνται με το περιβάλλον προσομοίωσης OPNET. Όταν δημιουργείται ένα αντικείμενο, ένας ορισμένος αριθμός χαρακτηριστικών θεωρείται ότι ανήκουν σε

αυτό. Τα χαρακτηριστικά είναι δυνατό να σχετίζονται με την εσωτερική δομή του αντικειμένου, τις δυνατότητες και το σκοπό του, την παρουσίαση και τις διεπαφές του. Η σχετιζόμενη πληροφορία οφείλει να προέρχεται από κάποια προηγούμενη προδιαγραφή με βάση την οποία έχει δημιουργηθεί το αντικείμενο: η συγκεκριμένη προδιαγραφή ονομάζεται μοντέλο του αντικειμένου.

Συνεπώς, το αντικείμενο είναι ένα στιγμιότυπο (instance) του μοντέλου και κάθε αντικείμενο είναι δυνατό να προέρχεται μόνο από ένα μοντέλο. Η διαδικασία δημιουργίας του αντικειμένου φυσικά είναι δυνατό να επαναληφθεί, με αποτέλεσμα τη δημιουργία ενός νέου αντικειμένου το οποίο αποτελεί στιγμιότυπο του ίδιου μοντέλου. Επομένως, υπάρχει μία σχέση «πολλαπλά-προς-ένα» μεταξύ αντικειμένων και μοντέλων. Αν και τα αντικείμενα βασίζονται στο ίδιο μοντέλο για τη λήψη ορισμένων χαρακτηριστικών, συνυπάρχουν ανεξάρτητα το ένα από το άλλο. Ισοδύναμα, αλλαγές σε ένα από τα στιγμιότυπα του μοντέλου δεν επηρεάζουν τα άλλα στιγμιότυπα, μία αρχή που αντικατοπτρίζει ορθά τις πραγματικές καταστάσεις που αντιπροσωπεύουν τα αντικείμενα.

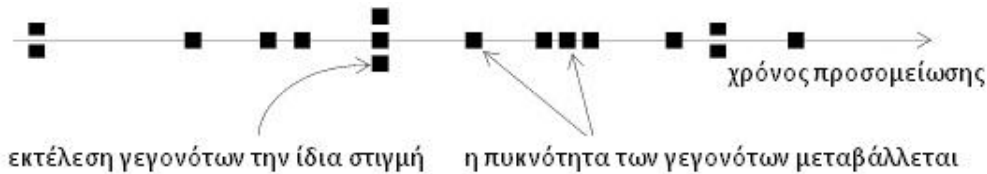
Τα χαρακτηριστικά (attributes) των αντικειμένων θεωρείται χρήσιμο να είναι διαθέσιμα στο χρήστη, για δύο βασικούς λόγους: για την πληροφόρηση των χρηστών σχετικά με τα επιλεγμένα χαρακτηριστικά του αντικειμένου και για τη δυνατότητα τροποποίησής τους για συγκεκριμένες εφαρμογές. Το είδος της πληροφορίας που μπορεί να προσδιορισθεί μέσω των χαρακτηριστικών είναι αρκετά γενικό. Περιλαμβάνει πληροφορίες σχετικές με τα γενικά χαρακτηριστικά, τη συμπεριφορά και την εσωτερική δομή του αντικειμένου. Τα χαρακτηριστικά είναι δυνατό να συσχετισθούν γενικά με τα μοντέλα των αντικειμένων, αλλά και ειδικά με τα αντικείμενα ως στιγμιότυπα.

3.1.3 Γεγονότα και Χρόνος προσομοίωσης

Η εκτέλεση των προσομοιώσεων που πραγματοποιούνται μέσω του περιβάλλοντος προσομοίωσης OPNET βασίζεται στην αρχή των διακεκριμένων γεγονότων (discrete event simulation), ή ισοδύναμα τη παραγωγή μιας ακολουθίας καταστάσεων για ένα μοντέλο συστήματος. Η λειτουργία του μοντέλου εξελίσσεται μέσω των συγκεκριμένων καταστάσεων ως συνάρτηση του χρόνου, με βάση τις προδιαγραφές της συμπεριφοράς των συνιστωσών του μοντέλου και τις μεταξύ τους αλληλεπιδράσεις. Αν οι προδιαγραφές του μοντέλου έχουν ορισθεί με ικανοποιητική ακρίβεια, η λειτουργία του αντιπροσωπεύει τη συμπεριφορά του πραγματικού συστήματος στη πάροδο του χρόνου. Η έννοια του χρόνου σε μία προσομοίωση δε σχετίζεται άμεσα με τον πραγματικό χρόνο που απαιτείται για την εκτέλεση της προσομοίωσης, αλλά με μία μεταβλητή που χρησιμοποιείται από το πρόγραμμα της προσομοίωσης. Η συγκεκριμένη μεταβλητή ονομάζεται χρόνος προσομοίωσης (simulation time) για να διακρίνεται με σαφή τρόπο από τον πραγματικό χρόνο (real time).

Με βάση τη προσέγγιση μοντελοποίησης με διακεκριμένα γεγονότα, η πρόοδος της λειτουργίας του μοντέλου στο χρόνο προσομοίωσης αποσυντίθεται σε ξεχωριστά χρονικά σημεία όπου πραγματοποιείται κάποια αλλαγή. Ο όρος που χρησιμοποιείται για κάθε τέτοιο χρονικό σημείο είναι ένα γεγονός (event). Κάθε γεγονός αντιπροσωπεύει την ανάγκη του μοντέλου να επιβληθεί μια αλλαγή στην κατάσταση του ή να ληφθεί μία απόφαση. Ορισμένα γεγονότα είναι δυνατό να μην επηρεάζουν την κατάσταση του συστήματος, λόγω του ότι οι ενέργειες που θα προκαλούσαν την αλλαγή εκφράζονται υπό συνθήκες.

Κάθε στιγμή που επέρχεται ένα γεγονός, εκτελείται από τη προσομοίωση και ο χρόνος προσομοίωσης αυξάνεται μονότονα. Καθώς το περιβάλλον προσομοίωσης OPNET υποστηρίζει τη μοντελοποίηση καταναμημένων συστημάτων, οφείλει να επιτρέπει την ταυτόχρονη (με την έννοια του χρόνου προσομοίωσης) εκτέλεση πολλαπλών γεγονότων τα οποία επηρεάζουν διαφορετικές συνιστώσες του συστήματος. Επομένως, οποιοσδήποτε αριθμός διακεκριμένων γεγονότων μπορεί να επέλθει στην ίδια ακριβώς χρονική στιγμή της προσομοίωσης.



Σχήμα 19: Γεγονότα & Χρόνος προσομοίωσης

Εκτός από τα γεγονότα, ορίζεται επιπλέον η σχετιζόμενη έννοια της διακοπής (interrupt). Τα γεγονότα είναι δυνατό να παρατηρηθούν ως αιτήσεις για πραγματοποίηση δραστηριοτήτων σε συγκεκριμένες χρονικές στιγμές. Σε αντίθεση, οι διακοπές ορίζονται ως οι επικλήσεις που προκύπτουν από την εκτέλεση των γεγονότων από τον πυρήνα της προσομοίωσης. Ισοδύναμα, τα γεγονότα που βρίσκονται στη λίστα γεγονότων προς εκτέλεση, μετατρέπονται σε διακοπές μόνο όταν παραδοθούν στις δομικές μονάδες για τις οποίες προορίζονται. Οι διακοπές ποτέ δεν αποτελούν μέρος της λίστας των γεγονότων προς εκτέλεση.

3.1.4 Η λειτουργία του επιπέδου διεργασίας

Τα μοντέλα διεργασιών χρησιμοποιούνται για τον προσδιορισμό της συμπεριφοράς των μονάδων επεξεργαστών και ουρών που περιέχονται στο επίπεδο κόμβου. Είναι δυνατό να χρησιμοποιηθούν για την υλοποίηση μεγάλου εύρους υποσυστημάτων υλικού και λογισμικού, όπως πρωτόκολλα επικοινωνίας, αλγόριθμοι, κοινόχρηστοι πόροι όπως μνήμη ή δίσκοι, λειτουργικά συστήματα, συστήματα αναμονής, ειδικευμένες πηγές κίνησης κ.ά.

Για την υλοποίηση ενός συγκεκριμένου έργου, ξεχωριστές διεργασίες ή ομάδες διεργασιών ορίζονται πριν τοποθετηθούν εντός μιας μονάδας επεξεργαστή ή ουράς. Μία διεργασία είναι ένα στιγμιότυπο ενός μοντέλου διεργασίας που έχει ορισθεί με το εργαλείο σύνταξης διεργασίας (process editor). Επομένως, κάθε μοντέλο διεργασίας μπορεί να ανατεθεί σε πολλές μονάδες επεξεργαστή ή ουράς σε ένα σύστημα. Κάθε ξεχωριστό στιγμιότυπο διεργασίας έχει ορισθεί με παρόμοια χαρακτηριστικά αλλά είναι δυνατό να λειτουργεί σε διαφορετικά περιβάλλοντα.

Οι ενέργειες που είναι απαραίτητο να υλοποιούνται από ένα μοντέλο διεργασίας είναι συχνά πολύπλοκες και ποικίλλουν σημαντικά ανάλογα την εφαρμογή. Επομένως η γλώσσα που θα χρησιμοποιηθεί για τον ορισμό των μοντέλων διεργασιών πρέπει να είναι εξαιρετικά ευέλικτη και ταυτόχρονα πολύπλευρη. Η γλώσσα που χρησιμοποιείται από το περιβάλλον προσομοίωσης OPNET για την ανάπτυξη των μοντέλων των διεργασιών ονομάζεται Proto-C και είναι ενσωματωμένη στο εργαλείο σύνταξης διεργασίας της εφαρμογής.

3.1.4.1 Διαγράμματα Μετάβασης Καταστάσεων

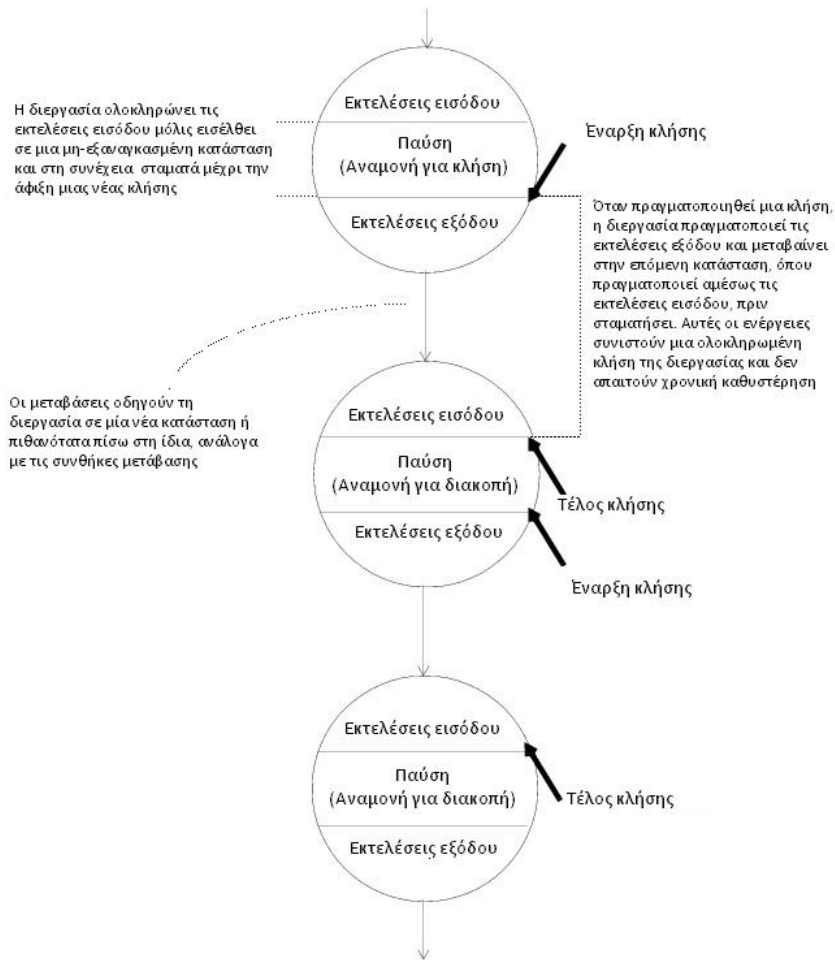
Η γλώσσα Proto-C βασίζεται στα διαγράμματα μετάβασης καταστάσεων (state transition diagrams) ή ισοδύναμα μηχανές πεπερασμένων καταστάσεων (finite state machines), το μηχανισμό που έχει επικρατήσει για την περιγραφή συστημάτων διακριτών γεγονότων που διατηρούν πληροφορίες σχετικά με τις καταστάσεις τους. Στα συγκεκριμένα διαγράμματα συνυπάρχουν καταστάσεις (states) που υποδηλώνουν τις τοποθεσίες υψηλού επιπέδου στις οποίες μπορεί να εισέλθει η διεργασία και μεταβάσεις (transitions), που προδιαγράφουν τις επιτρεπτές μεταβολές καταστάσεων.

Οι καταστάσεις είναι αμοιβαία αποκλειστικές και συμπληρωματικές, δηλαδή μία διεργασία βρίσκεται πάντα σε μία μόνο κατάσταση και μπορεί να μετακινηθεί σε μία άλλη ως απόκριση σε μία διακοπή που δέχεται. Οι μεταβάσεις που αναχωρούν από μία κατάσταση υποδεικνύουν σε ποιες καταστάσεις μπορεί να εισέλθει η διεργασία στη συνέχεια καθώς και τη συνθήκη που απαιτείται για την κάθε μετάβαση. Η γλώσσα Proto-C ορίζει δύο είδη καταστάσεων, τις εξαναγκασμένες (forced states, συμβολίζονται με πράσινο κύκλο) και τις μη-εξαναγκασμένες (unforced states, συμβολίζονται με κόκκινο κύκλο), οι οποίες διαφέρουν στο χρόνο εκτέλεσης.



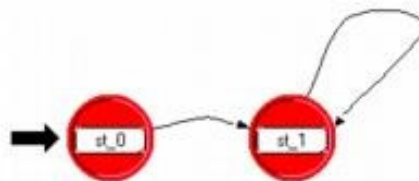
Σχήμα 20: Συμβολισμός καταστάσεων στο περιβάλλον προσομοίωσης OPNET

Οι ενέργειες που πραγματοποιούνται σε μία κατάσταση αποκαλούνται εκτελέσεις (executives). Οι εκτελέσεις μίας κατάστασης χωρίζονται στις εκτελέσεις εισόδου (enter executives) που πραγματοποιούνται κατά την είσοδο της διεργασίας στην κατάσταση και τις εκτελέσεις εξόδου (exit executives) που πραγματοποιούνται κατά την έξοδό της. Οι μη-εξαναγκασμένες καταστάσεις επιτρέπουν μία παύση μεταξύ των εκτελέσεων εισόδου και εξόδου, επομένως είναι σε θέση να μοντελοποιήσουν πραγματικές καταστάσεις ενός συστήματος. Αντίθετα στις εξαναγκασμένες καταστάσεις οι εκτελέσεις εισόδου και εξόδου πραγματοποιούνται διαδοχικά, χωρίς παύση μεταξύ τους.



Σχήμα 21: Εκτελέσεις εισόδου & εξόδου κατά την κλήση μιας διεργασίας

Μία ειδική περίπτωση κατάστασης, η αρχική κατάσταση (initial state), πρέπει να ορίζεται υποχρεωτικά σε κάθε μοντέλο διεργασίας. Γραφικά, η αρχική κατάσταση αναγνωρίζεται από ένα βέλος τοποθετημένο στα αριστερά της.



Σχήμα 22: Η αρχική κατάσταση

Η αρχική κατάσταση είναι το σημείο από το οποίο ξεκινάει η εκτέλεση κατά την πρώτη κλήση της διεργασίας. Δεν έχει κάποια ειδική σημασία για τις επόμενες κλήσεις και μπορεί να λειτουργεί ως μία συνηθισμένη κατάσταση. Ωστόσο, επειδή συνήθως η αρχική κατάσταση περιέχει εκτελέσεις που

αφορούν την αρχικοποίηση μεταβλητών που πρέπει να πραγματοποιηθούν μόνο μία φορά, τα περισσότερα μοντέλα δεν περιλαμβάνουν μεταβάσεις πίσω σε αυτήν.

Όπως ήδη αναφέρθηκε, οι μεταβάσεις περιγράφουν πιθανές κινήσεις μιας διεργασίας μεταξύ των καταστάσεων και τις συνθήκες υπό τις οποίες μπορεί να πραγματοποιηθεί η συγκεκριμένη μετάβαση. Υπάρχουν τέσσερις συνιστώσες στη προδιαγραφή μιας μετάβασης: η κατάσταση-αφετηρία, η κατάσταση-προορισμός, η έκφραση της συνθήκης μετάβασης και η έκφραση εκτέλεσης μετάβασης. Η προδιαγραφή διαβάζεται: «Όταν η διεργασία βρίσκεται στην κατάσταση-αφετηρία, αν η συνθήκη μετάβασης είναι αληθής, υλοποιείται η εκτέλεση μετάβασης και ο έλεγχος μεταφέρεται στην κατάσταση-προορισμό».

3.1.4.2 Μεταβλητές

Οι διεργασίες της γλώσσας Proto-C έχουν πρόσβαση σε αρκετές μορφές μνήμης για την αποθήκευση πληροφοριών. Κάθε μορφή μνήμης χαρακτηρίζεται από διαφορετικές ιδιότητες που την καθιστούν ορθή για συγκεκριμένες χρήσεις. Κάποιες από αυτές τις μορφές μνήμης επιτρέπουν τη δήλωση και τη διαχείριση πληροφορίας απευθείας σαν συνηθισμένα ονόματα: αυτές οι μορφές αποθήκευσης αναφέρονται ως μεταβλητές (variables). Η γλώσσα Proto-C χρησιμοποιεί τρεις ξεχωριστές κατηγορίες μεταβλητών: τις μεταβλητές κατάστασης (state variables), τις προσωρινές (temporary) και τις καθολικές (global) μεταβλητές.

Μεταβλητές κατάστασης: ονομάζονται οι μεταβλητές που χρησιμοποιούνται για την αναπαράσταση συσσωρευμένων πληροφοριών που διατηρούνται από μία διεργασία. Οφείλουν το όνομά τους στο γεγονός ότι οι συγκεκριμένες μεταβλητές, σε συνδυασμό με την παρούσα θέση μιας διεργασίας εντός του διαγράμματος μετάβασης καταστάσεων, αντιπροσωπεύουν την πλήρη κατάσταση μιας διεργασίας σε οποιαδήποτε χρονική στιγμή. Οι διεργασίες είναι δυνατό να παράγουν ή να έχουν πρόσβαση σε πλήθος πληροφοριών στην πάροδο του χρόνου, αλλά αυτές δεν κωδικοποιούνται στην κατάστασή τους. Επομένως ο όρος «κατάστασης» αναφέρεται μόνο στην πληροφορία που η ίδια η διεργασία αποφασίζει να διατηρήσει αποθηκεύοντάς την στις μεταβλητές κατάστασης, οι οποίες διατηρούν την τιμή τους στη πάροδο του χρόνου. Οι μεταβλητές κατάστασης δηλώνονται σε μία αφιερωμένη περιοχή του μοντέλου της γλώσσας Proto-C, η οποία αναφέρεται ως τμήμα μεταβλητών κατάστασης (state variable block).

Προσωρινές μεταβλητές: μία διεργασία μπορεί να δηλώσει προσωρινές μεταβλητές για να αποθηκεύσει πληροφορίες που δεν απαιτούν διατήρηση της τιμής τους στην πάροδο του χρόνου. Για τις προσωρινές μεταβλητές, δεν είναι εγγυημένο ότι θα έχουν κάποιο συγκεκριμένο σύνολο τιμών κατά την κλήση ενός μοντέλου διεργασίας. Πιο συγκεκριμένα, οι τιμές που είχαν ανατεθεί σε αυτό το είδος μεταβλητών από την ίδια διεργασία σε μία προηγούμενη κλήση της, πιθανότατα θα έχουν αλλάξει. Λόγω του ότι δεν διατηρούν την τιμή τους, οι προσωρινές μεταβλητές είναι χρήσιμες μόνο ως πρόχειρη αποθήκευση για τη διευκόλυνση υπολογισμών σύνθετων εκφράσεων ή για την αποθήκευση πληροφορίας μέσα στο χρονικό πλαίσιο μίας κλήσης της διεργασίας. Δηλώνονται στο τμήμα προσωρινών μεταβλητών (temporary variable block) του εργαλείου σύνταξης διεργασίας.

Καθολικές μεταβλητές: παρέχουν έναν τρόπο αποθήκευσης πληροφοριών σε ένα κοινό σημείο, προσβάσιμο από πολλαπλές διεργασίες που ανήκουν σε διαφορετικές δομικές μονάδες. Είναι ένα χαρακτηριστικό που παρέχεται απευθείας από την οικογένεια γλωσσών C και C++, οι οποίες επίσης αναφέρονται στις συγκεκριμένες μεταβλητές ως καθολικές. Οι δηλώσεις τους πραγματοποιούνται στο τμήμα επικεφαλίδας (header block) του μοντέλου διεργασίας.

Όλες οι μεταβλητές ανεξάρτητα από την κατηγορία τους έχουν ένα όνομα, έναν τύπο δεδομένων και μία τιμή. Το όνομα και ο τύπος δεδομένων μιας μεταβλητής παραμένουν αμετάβλητα για όλο το κύκλο ζωής μιας διεργασίας, ενώ η τιμή είναι η μόνη συνιστώσα της που επιτρέπεται να μεταβληθεί.

3.1.5 Η λειτουργία του επιπέδου κόμβου

Τα μοντέλα δικτύων του περιβάλλοντος προσομοίωσης OPNET κατασκευάζονται από δύο ευρείες κλάσεις αντικειμένων: κόμβους και ζεύξεις επικοινωνιών. Η εσωτερική δομή αυτών των αντικειμένων σε γενικές γραμμές δεν είναι ορατή από το επίπεδο δικτύου. Τόσο η πολυπλοκότητα της εσωτερικής δομής όσο και ο σκοπός της δραστηριότητάς τους μπορούν να ποικίλλουν σημαντικά, ανάλογα με το σύστημα που μοντελοποιείται.

Ένα μοντέλο κόμβου αποτελείται από μία σειρά διασυνδεδεμένων τμημάτων που ονομάζονται δομικές μονάδες (modules). Κάθε δομική μονάδα περιέχει ένα σύνολο εισόδων και εξόδων, μία ποσότητα μνήμης κατάσταση, καθώς και μία μέθοδο για τον υπολογισμό των εξόδων της από τις εισόδους και τη μνήμη της. Ο τρόπος με τον οποίο πραγματοποιείται ο συγκεκριμένος υπολογισμός εξαρτάται από τον τύπο της δομικής μονάδας. Ορισμένες χαρακτηρίζονται από προκαθορισμένη συμπεριφορά για την επίτευξη ενός συγκεκριμένου σκοπού, ενώ άλλες υποστηρίζουν τον καθορισμό της συμπεριφοράς τους από το σχεδιαστή του μοντέλου.

Τα μοντέλα κόμβων καθορίζουν τον τρόπο με τον οποίο συνδέονται οι εισοδοί και οι εξοδοί των διαφόρων δομικών μονάδων, μέσω αντικειμένων που ονομάζονται συνδέσεις. Μια δομική μονάδα μπορεί να αποστέλλει πακέτα δεδομένων μέσω των ρών πακέτων εξόδου (output packet streams) και να παραλάβει μέσω των ρών πακέτων εισόδου (input packet streams).

3.1.5.1 Κατηγορίες δομικών μονάδων

Οι δομικές μονάδες αντιπροσωπεύουν τα μέρη ενός κόμβου επικοινωνιών από τα οποία παράγονται, καταστρέφονται ή επεξεργάζονται δεδομένα. Υπάρχουν διάφορα είδη δομικών μονάδων, με βάση τη διάκριση της λειτουργίας τους εντός του κόμβου: επεξεργαστές, ουρές, δέκτες και πομποί. Οι λειτουργίες ορισμένων ειδών δομικών μονάδων όπως οι επεξεργαστές και οι ουρές, περιορίζονται αυστηρά στο εσωτερικό του κόμβου, ενώ άλλες, όπως οι πομποί και οι δέκτες διαθέτουν εξωτερικές συνδέσεις προς ζεύξεις μετάδοσης δεδομένων. Η συμπεριφορά των επεξεργαστών και των ουρών είναι δυνατό να προδιαγραφεί με ακρίβεια, μέσω μοντέλων διεργασιών δημιουργημένα από το χρήστη. Άλλες δομικές μονάδες έχουν προκαθορισμένη

συμπεριφορά η οποία είναι δυνατό να τροποποιηθεί μόνο με την αλλαγή των τιμών των χαρακτηριστικών τους.

Επεξεργαστές: οι δομικές μονάδες επεξεργαστών (processor modules) είναι οι πρωταρχικές μονάδες γενικού σκοπού, από τις οποίες κατασκευάζονται τα μοντέλα κόμβων. Η συμπεριφορά τους μπορεί να προδιαγραφεί πλήρως μέσω του χαρακτηριστικού μοντέλου διεργασίας που διαθέτουν, το οποίο καθορίζει το μοντέλο διεργασίας που εκτελείται από τη δομική μονάδα. Το μοντέλο διεργασίας μπορεί να ανταποκρίνεται σε εξωτερικά γεγονότα ή διακοπές, ανάλογα τις απαιτήσεις λειτουργίας του.



Σχήμα 23: Συμβολισμός ενός επεξεργαστή στο περιβάλλον OPNET

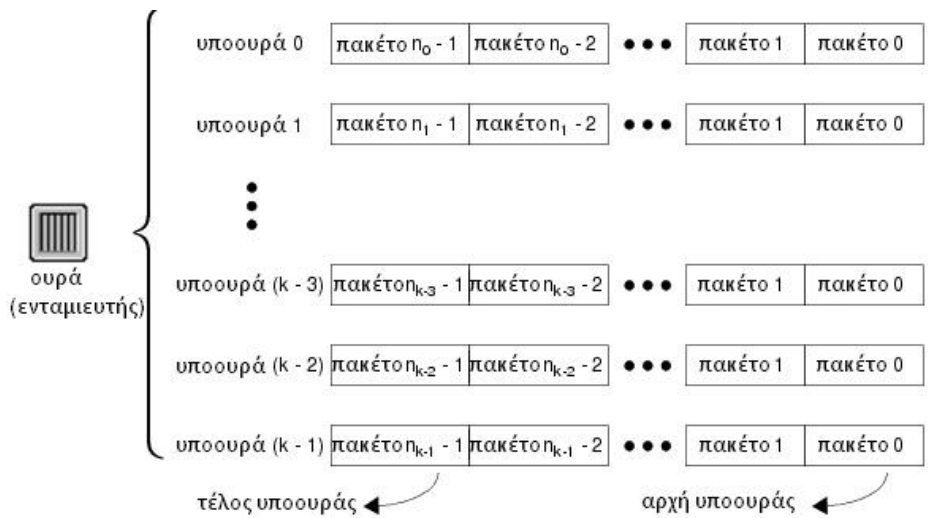
Οι επεξεργαστές μπορούν να συνδεθούν με άλλες δομικές μονάδες για την αποστολή και λήψη πακέτων μέσω οποιουδήποτε αριθμού ροών πακέτων. Χρησιμοποιούνται επίσης για την επεξεργασία πακέτων δεδομένων ή την παραγωγή τους.

Ουρές: οι δομικές μονάδες ουρών (queue modules) παρέχουν ένα υπερσύνολο της λειτουργικότητας των επεξεργαστών. Όπως και οι επεξεργαστές, μπορούν να εκτελέσουν ένα αυθαίρετο μοντέλο διαδικασίας που περιγράφει τη συμπεριφορά μιας συγκεκριμένης διαδικασίας ή πρωτοκόλλου, ή να συνδεθούν μέσω ροών πακέτων με άλλες δομικές μονάδες, επιτρέποντας τη λήψη και την αποστολή πακέτων δεδομένων.



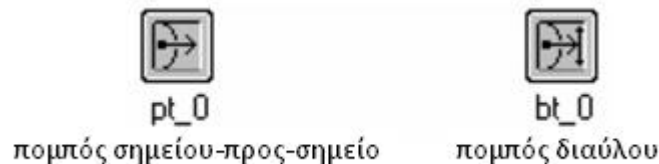
Σχήμα 24: Συμβολισμός μιας ουράς στο περιβάλλον OPNET

Η βασική διαφορά μεταξύ επεξεργαστών και ουρών έγκειται στο ότι οι ουρές περιέχουν επιπρόσθετους εσωτερικούς πόρους, τις υποουρές. Οι υποουρές διευκολύνουν την ενταμίευση (buffering) και τη διαχείριση μιας συλλογής πακέτων δεδομένων. Αν και είναι δυνατή να υλοποιηθεί η συγκεκριμένη λειτουργία με χρήση επεξεργαστών, η χρήση υποουρών σε συνδυασμό με τις διαδικασίες πυρήνα (kernel procedures) που τις χειρίζονται, παρέχει βελτιωμένη ευελιξία και ευκολία υλοποίησης μεγάλου εύρους αρχών αναμονής.



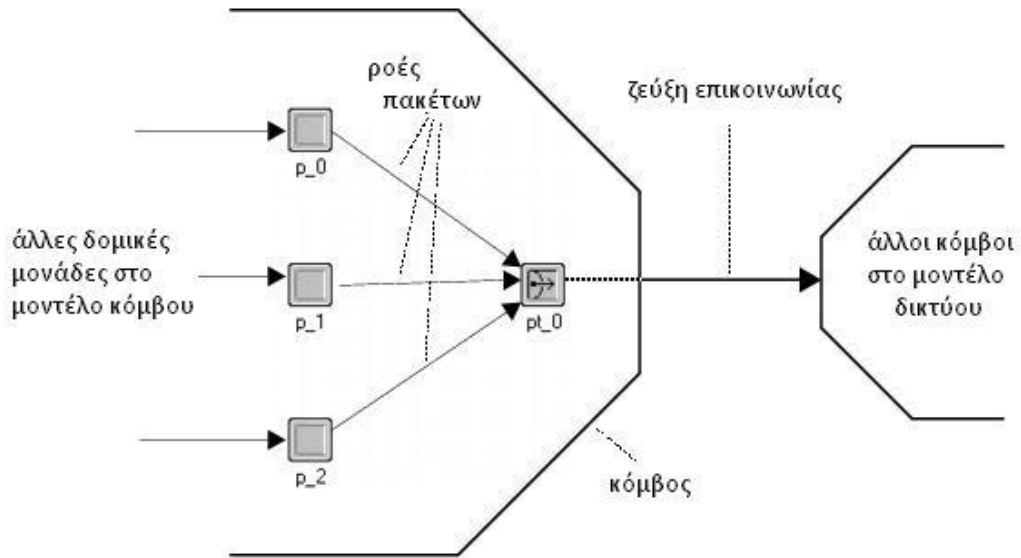
Σχήμα 25: Υποσούρες στο εσωτερικό μιας δομικής μονάδας σούρας

Πομποί: οι δομικές μονάδες πομπών (transmitter modules) λειτουργούν ως εξερχόμενες διεπαφές μεταξύ των ρών πακέτων στο εσωτερικό του κόμβου και τις ζεύξεις επικοινωνιών στο εξωτερικό του. Υπάρχουν δύο τύποι πομπών, σε αντιστοιχία με τους διαφορετικούς τύπους ζεύξεων επικοινωνιών: σημείου-προς-σημείο (point-to-point transmitter) και διαύλου (bus transmitter). Κάθε είδος πομπού παρουσιάζει κατ' ουσίαν την ίδια συμπεριφορά, ωστόσο υπάρχουν ορισμένες διαφορές που σχετίζονται με τους διαφορετικούς μηχανισμούς επικοινωνίας.



Σχήμα 26: Συμβολισμός πομπών στο περιβάλλον OPNET

Αρκετές από τις παραμέτρους που ελέγχουν τη μετάδοση των πακέτων από τους πομπούς σημείου-προς-σημείο και διαύλου, όπως είναι η πιθανότητα σφαλμάτων και η καθυστέρηση διάδοσης ορίζονται ως χαρακτηριστικά της ζεύξης.



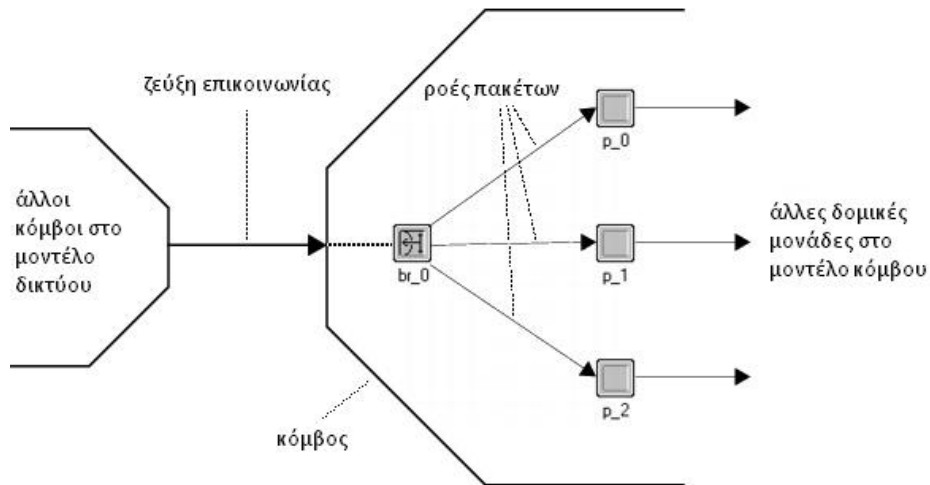
Σχήμα 27: Λειτουργία ενός πομπού για την επικοινωνία μεταξύ δύο κόμβων

Δέκτες: οι δομικές μονάδες δεκτών (receiver modules) λειτουργούν ως εισερχόμενες διεπαφές μεταξύ των ζεύξεων επικοινωνιών στο εξωτερικό ενός κόμβου και των ροών πακέτων στο εσωτερικό του. Όπως και με τους πομπούς, υπάρχουν δύο είδη δεκτών σε αντιστοιχία με τους διαφορετικούς τύπους ζεύξεων επικοινωνιών: σημείου-προς-σημείο (point-to-point receiver) και διαύλου (bus receiver).



Σχήμα 28: Συμβολισμός δεκτών στο περιβάλλον OPNET

Κάθε είδος δέκτη παρουσιάζει κατ' ουσίαν την ίδια συμπεριφορά, ωστόσο υπάρχουν ορισμένες διαφορές που σχετίζονται με τους διαφορετικούς μηχανισμούς επικοινωνίας.

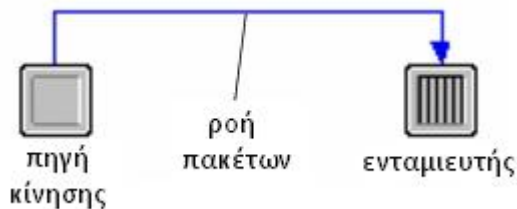


Σχήμα 29: Λειτουργία ενός δέκτη για την επικοινωνία μεταξύ δύο κόμβων

3.1.5.2 Σύνδεση δομικών μονάδων

Η σύνδεση των δομικών μονάδων πραγματοποιείται κατά κανόνα με ροές πακέτων (packet streams), συνδέσεις που μεταφέρουν πακέτα δεδομένων από μία δομική μονάδα-πηγή προς μία δομική μονάδα-προορισμό. Αντιπροσωπεύουν τη ροή των δεδομένων διαμέσου των διεπαφών του υλικού και του λογισμικού, στο εσωτερικό ενός κόμβου επικοινωνιών. Θεωρούνται απολύτως αξιόπιστες συνδέσεις, δηλαδή δεν εμφανίζεται απώλεια πακέτων ούτε εισαγωγή σφαλμάτων, ενώ διαθέτουν δυνητικά άπειρο εύρος ζώνης αφού ένα αυθαίρετα μεγάλο πακέτο μπορεί να διασχίσει μία ροή χωρίς καθυστέρηση. Επιπλέον οι ροές πακέτων διαθέτουν άπειρη χωρητικότητα ενταμίευσης πακέτων δεδομένων στη δομική μονάδα-προορισμό, διατηρώντας παράλληλα τη σειρά άφιξής τους.

Οι ροές πακέτων υποστηρίζουν την εισαγωγή καθυστέρησης μεταξύ της χρονικής στιγμής που ένα πακέτο τοποθετείται στη ροή από την πηγή και τη στιγμή που φτάνει στον προορισμό του. Η καθυστέρηση μπορεί να είναι επιθυμητή για τη μοντελοποίηση ενός σταθερού χρόνου επεξεργασίας, μιας καθυστέρησης διάδοσης ή ενός ρυθμού μεταφοράς.

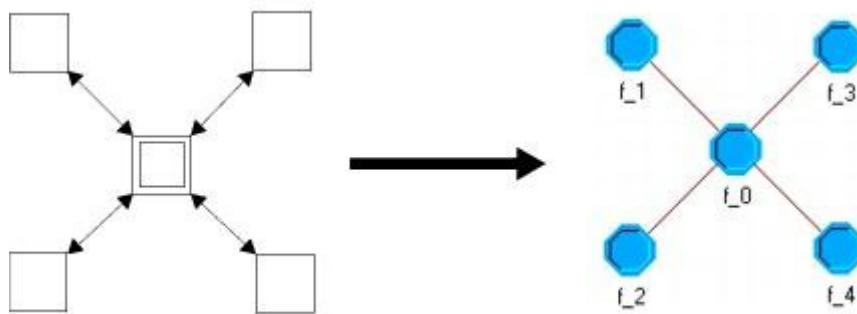


Σχήμα 30: Συμβολισμός ροών πακέτων στο περιβάλλον OPNET

3.1.6 Η λειτουργία του επιπέδου δικτύου

Ένα μοντέλο δικτύου ορίζει τη συνολική έκταση του συστήματος που θα προσομοιωθεί. Πρόκειται για μία περιγραφή υψηλού επιπέδου των αντικειμένων που περιέχονται στο σύστημα. Το μοντέλο δικτύου προδιαγράφει τα αντικείμενα που ανήκουν στο σύστημα, αλλά και τη φυσική τους θέση, τις μεταξύ τους συνδέσεις και ρυθμίσεις. Τα αντικείμενα στο επίπεδο δικτύου περιλαμβάνουν υποδίκτυα, κόμβους και ζεύξεις επικοινωνιών.

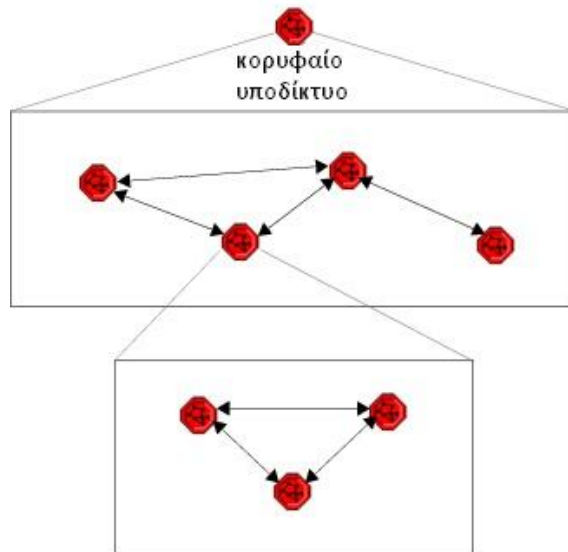
Το μέγεθος και η έκταση των μοντελοποιημένων δικτύων μπορεί να κυμαίνεται από έναν κόμβο ή ένα υποδίκτυο έως πολλούς διασυνδεδεμένους κόμβους και υποδίκτυα, αφού η δομή και η πολυπλοκότητα ενός μοντέλου δικτύου συνήθως ακολουθεί αυτή του συστήματος που μοντελοποιείται. Για παράδειγμα, ένα δίκτυο με τοπολογία αστέρα μοντελοποιείται με ένα κεντρικό κόμβο και ορισμένους περιφερειακούς κόμβους συνδεδεμένους με αυτό μέσω ζεύξεων σημείου-προς-σημείο.



Σχήμα 31: Μοντελοποίηση ενός δικτύου τοπολογίας αστέρα

3.1.6.1 Τύποι αντικειμένων στο επίπεδο δικτύου

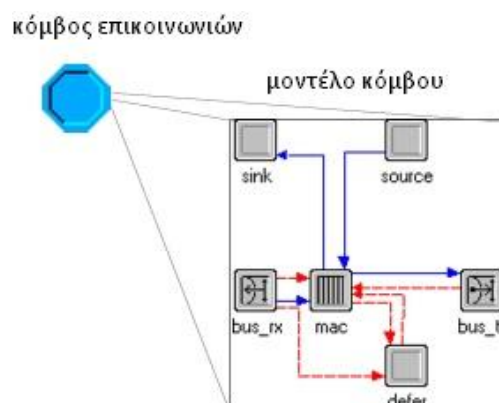
Υποδίκτυα: πρόκειται κατ' ουσίαν για οντότητες που διαχωρίζουν τα τμήματα του δικτύου που έχουν οριστεί να ανήκουν στο υποδίκτυο, σε ένα αντικείμενο. Ένα υποδίκτυο μπορεί να περιέχει ένα σύνολο κόμβων και ζεύξεων, συνήθως προκειμένου να αναπαραστήσει μία φυσική ή λογική ομαδοποίηση αντικειμένων όπως π.χ. ένα δίκτυο τοπικής περιοχής. Ένα υποδίκτυο μπορεί να περιέχει άλλα υποδίκτυα, τα οποία σχηματίζουν την ιεραρχία του μοντέλου δικτύου. Η ιεραρχία πρέπει να επεκτείνεται όσο είναι απαραίτητο για τις ανάγκες της μοντελοποίησης της δομής του δικτύου. Ένα υποδίκτυο και τα αντικείμενα που περιέχονται σε αυτό περιγράφονται στην ιεραρχία από μία σχέση γονέα - παιδιού (parent - child relationship). Το υποδίκτυο είναι ο γονέας των αντικειμένων στο εσωτερικό του, ενώ τα αντικείμενα είναι τα παιδιά του υποδικτύου.



Σχήμα 32: Ιεραρχία υποδικτύων

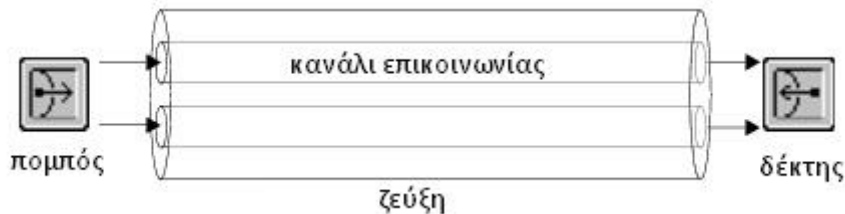
Ένα ξεχωριστό υποδίκτυο, το οποίο ονομάζεται κορυφαίο ή καθολικό υποδίκτυο, βρίσκεται στο υψηλότερο επίπεδο της ιεραρχίας του δικτύου. Το κορυφαίο υποδίκτυο δεν έχει αντικείμενο – γονέα. Τα συνηθισμένα υποδίκτυα δημιουργούνται και διασυνδέονται στο εσωτερικό του κορυφαίου υποδικτύου ή άλλων υποδικτύων.

Κόμβοι επικοινωνιών: αντιπροσωπεύουν δικτυακές συσκευές με μεγάλο εύρος λειτουργιών και τοποθετούνται στο εσωτερικό ενός υποδικτύου. Η λειτουργία και συμπεριφορά ενός κόμβου καθορίζεται από το μοντέλο κόμβου του, στο οποίο προδιαγράφεται η εσωτερική του δομή. Ένας κόμβος αποκαλείται στιγμιοτύπο του μοντέλου κόμβου του και ξεχωριστά στιγμιοτύπα του ίδιου μοντέλου κόμβου λειτουργούν ανεξάρτητα κατά τη διάρκεια μιας προσομοίωσης. Ένα μοντέλο δικτύου μπορεί να περιέχει αυθαίρετο αριθμό κόμβων επικοινωνιών, στιγμιοτύπα του ίδιου ή διαφορετικού μοντέλου κόμβου.



Σχήμα 33: Το εσωτερικό μοντέλο λειτουργίας ενός κόμβου επικοινωνιών

Ζεύξεις επικοινωνιών: οι ζεύξεις επιτρέπουν την ανταλλαγή πληροφοριών μεταξύ των κόμβων με τη μορφή πακέτων. Όταν ένα πακέτο υποβάλλεται σε έναν πομπό του κόμβου-πηγής, μεταφέρεται μέσω της ζεύξης στο δέκτη του κόμβου-προορισμού. Μία ζεύξη αποτελείται από ένα ή περισσότερα κανάλια επικοινωνίας, καθένα από τα οποία ορίζει μία σύνδεση ανάμεσα σε ένα κανάλι πομπού και ένα κανάλι δέκτη. Ένας πομπός μπορεί να υποστηρίζει πολλαπλά εξερχόμενα κανάλια προς μία ζεύξη και αντίστοιχα ένας δέκτης να υποστηρίζει πολλαπλά εισερχόμενα κανάλια από μία ζεύξη.

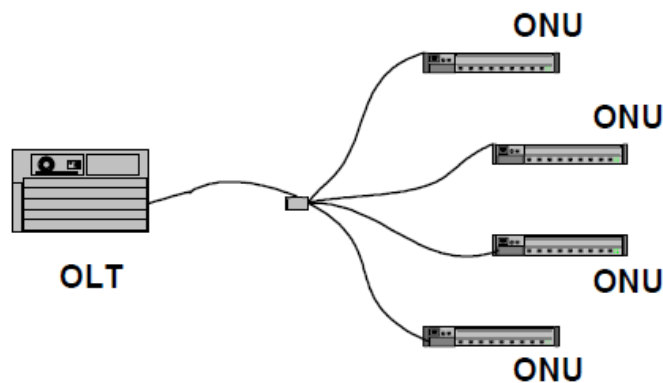


Σχήμα 34: Κανάλια επικοινωνίας στο εσωτερικό μιας ζεύξης

Το περιβάλλον προσομοίωσης OPNET υποστηρίζει δύο είδη ζεύξεων: σημείου-προς-σημείο και διαύλου. Κάθε είδος ζεύξης παρέχει θεμελιωδώς διαφορετικό τύπο συνδεσιμότητας: οι ζεύξεις σημείου-προς-σημείο συνδέουν ένα κόμβο-πηγή με ένα κόμβο-προορισμό, ενώ οι ζεύξεις διαύλου συνδέουν ένα σταθερό σύνολο κόμβων μεταξύ τους.

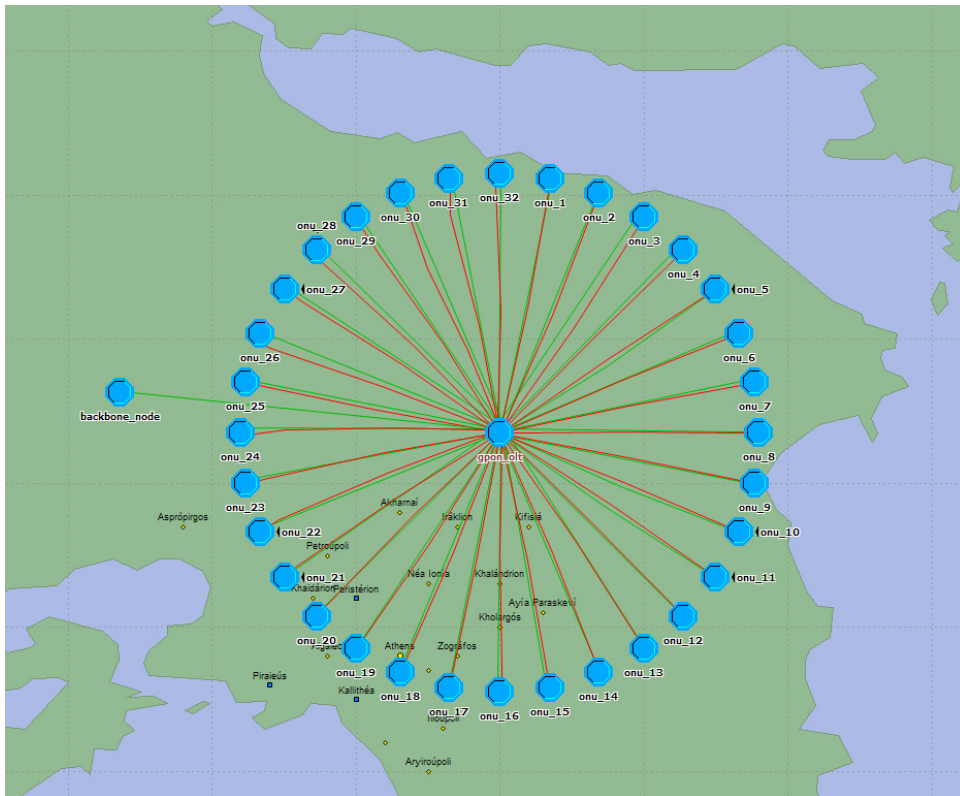
3.2 Μοντελοποίηση του δικτύου G-PON

Για τις ανάγκες της προσομοίωσης της λειτουργίας ενός δικτύου G-PON, δημιουργήθηκε ένα μοντέλο που ακολουθεί τις προδιαγραφές του πρωτοκόλλου. Η τοπολογία που επιλέχθηκε να προσομοιωθεί είναι το παθητικό δέντρο, όπου μία οπτική ίνα εξερχόμενη από το τερματικό OLT συνδέεται με πολλαπλά σημεία τερματισμού, μέσω της διαίρεσης του σήματος από έναν ή περισσότερους παθητικούς διαχωριστές.



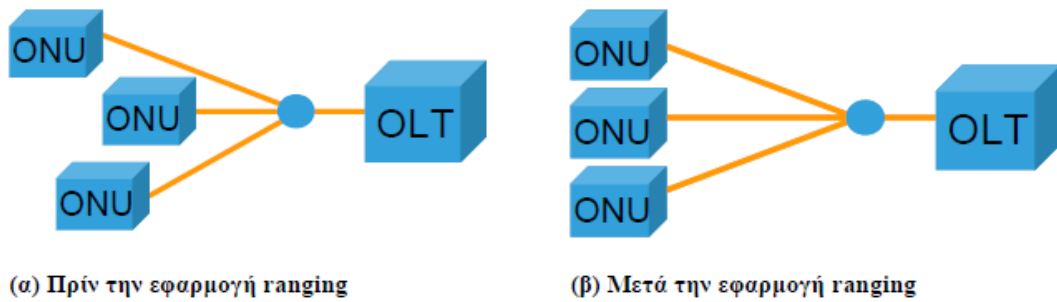
Σχήμα 35: Η τοπολογία παθητικού δέντρου

Για την υλοποίηση της τοπολογίας του παθητικού δέντρου στο περιβάλλον προσομοίωσης OPNET, λόγω απουσίας διαχωριστών και περιορισμού στο κόστος, στο επίπεδο δικτύου το τερματικό OLT συνδέεται μέσω δύο ζεύξεων σημείου-προς-σημείο με κάθε μία από τις 32 μονάδες ONU. Επιπλέον το τερματικό OLT συνδέεται με έναν κόμβο που προσομοιώνει τη λειτουργία του δικτύου κορμού.



Σχήμα 36: Το μοντέλο του δικτύου G-PON στο περιβάλλον OPNET

Δεδομένου ότι η λειτουργία ενός δικτύου G-PON περιλαμβάνει τη διαδικασία της αντιστάθμισης απόστασης (ranging) των μονάδων ONU, η οποία δεν εμπεριέχεται στο αντικείμενο της διπλωματικής εργασίας, τα μήκη όλων των ζεύξεων τέθηκαν ίσα με τη μέγιστη φυσική εμβέλεια των 20 χιλιομέτρων.

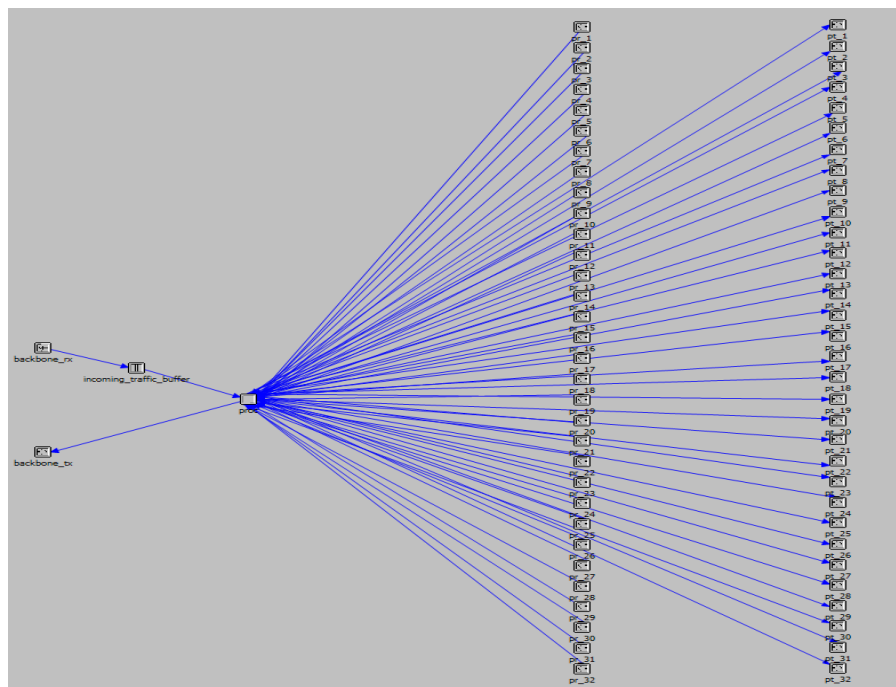


Σχήμα 37: Η διαδικασία αντιστάθμισης απόστασης

Οι ζεύξεις σχεδιάσθηκαν να υποστηρίζουν ρυθμό μετάδοσης 2.48832 Gbit/s στην κατεύθυνση ροής καθόδου (πράσινη ζεύξη) και 1.24416 Gbit/s στην κατεύθυνση ροής ανόδου (κόκκινη ζεύξη). Η αναλογία λαθών προς bit (bit error rate) τέθηκε ίση με 0, ενώ χρησιμοποιήθηκε μηδενική καθυστέρηση διάδοσης. Σημειώνεται ότι σε πραγματικές συνθήκες για απόσταση 20 χιλιομέτρων η καθυστέρηση διάδοσης προσεγγίζεται με μία σταθερή τιμή των 100μs [11].

Εμβαθύνοντας περαιτέρω στο μοντέλο του δικτύου G-PON, ακολουθεί η ανάλυση της λειτουργίας των δομικών μονάδων του: το τερματικό OLT και οι μονάδες ONU.

3.2.1 Ο κόμβος τερματικού OLT του δικτύου G-PON



Σχήμα 38: Δομικές μονάδες στο εσωτερικό του κόμβου τερματικού OLT

Εξετάζοντας το τερματικό OLT σε επίπεδο κόμβου, οι δομικές μονάδες από τις οποίες αποτελείται είναι:

- 32 ζεύγη δεκτών και πομπών σημείου-προς-σημείο, καθένα συνδεδεμένο με τις 2 ζεύξεις που εξυπηρετούν την επικοινωνία με τις 32 μονάδες ONU.
- Ένα επιπλέον ζεύγος πομπού και δέκτη σημείου-προς-σημείο, το οποίο εξυπηρετεί την επικοινωνία του τερματικού OLT με το δίκτυο κορμού.
- Μία ουρά (“gpon_olt.incoming_traffic_buffer”) σε ρόλο ενταμιευτή για τα πακέτα που προέρχονται από το δίκτυο κορμού με προορισμό το δίκτυο G-PON.
- Ένας επεξεργαστής (“gpon_olt.proc”), ο οποίος αποτελεί τη βασικότερη δομική μονάδα του τερματικού OLT και ρυθμίζει τον τρόπο λειτουργίας του.

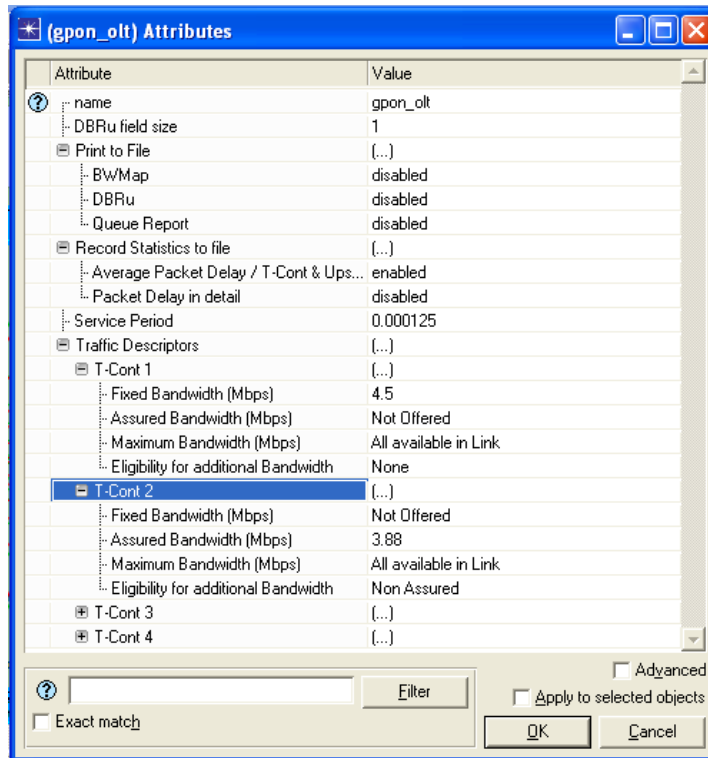
3.2.1.1 Χαρακτηριστικά του τερματικού OLT

Το τερματικό OLT διαθέτει μία σειρά από χαρακτηριστικά τα οποία καθορίζουν τον τρόπο λειτουργίας του μοντέλου G-PON ως σύνολο. Αναλυτικά τα διαθέσιμα χαρακτηριστικά και οι προσφερόμενες επιλογές:

Μήκος πεδίου αναφοράς DBRu (bytes): 1 ή 2. Καθορίζει το μήκος του πεδίου DBRu (δυναμική αναφορά εύρους ζώνης ροής ανόδου), που χρησιμοποιείται από τις μονάδες ONU για την αναφορά του όγκου των δεδομένων που περιμένουν να εξυπηρετηθούν στους ενταμιευτές τους.

Εκτύπωση σε αρχείο: BWmap / DBRu / QueueReport: Ενεργοποιημένο / Απενεργοποιημένο. Αν οριστεί ως ενεργοποιημένο, τυπώνονται κατά τη διάρκεια της προσομοίωσης σε αρχείο τα περιεχόμενα της χαρτογράφησης BWmap, η αναφορά DBRu ή ορισμένα στατιστικά που αφορούν τα δεδομένα κάθε ουράς (άφιξη και αποχώρηση πακέτων, συνολικό μέγεθος ουράς, μέγεθος πρώτου πακέτου) αντίστοιχα.

Καταγραφή στατιστικών σε αρχείο: Μέση καθυστέρηση πακέτων ανά κλάση T-Cont και αποδοτικότητα ροής ανόδου / Καθυστέρηση πακέτων λεπτομερώς: Ενεργοποιημένο / Απενεργοποιημένο. Αν οριστεί ως ενεργοποιημένο, τυπώνονται σε αρχείο τα επιλεγμένα στατιστικά της προσομοίωσης. Με την 1^η επιλογή τυπώνεται στο τέλος της προσομοίωσης η μέση καθυστέρηση από άκρο σε άκρο (end-to-end delay) των πακέτων για κάθε κλάση υπηρεσίας T-Cont και ο δείκτης αποδοτικότητας ροής ανόδου (για όλες τις μονάδες ONU). Με τη 2^η επιλογή τυπώνεται κατά τη διάρκεια της προσομοίωσης η καθυστέρηση από άκρο σε άκρο κάθε πακέτου, σε ξεχωριστό αρχείο για κάθε κλάση T-Cont.



Σχήμα 39: Ρυθμίσεις των χαρακτηριστικών του τερματικού OLT

Περίοδος δειγματοληψίας (δευτερόλεπτα): 125 / 250 / 500 / 1000 / 2000 μ s, άλλη τιμή (πολλαπλάσια των 125 μ s). Ορίζει τη χρονική περίοδο κατά την οποία θα ζητείται από κάθε μονάδα ONU να αποστέλλει την αναφορά DBRu ενθυλακωμένη στο πλαίσιο σύγκλισης μετάδοσης ροής ανόδου.

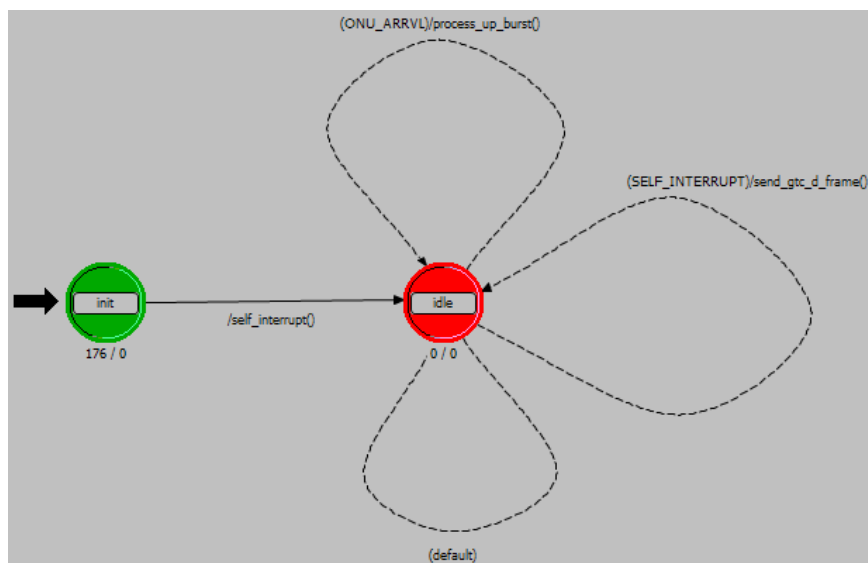
Περιγραφείς κίνησης, για κάθε T-Cont ξεχωριστά: Ισχύουν για κάθε μονάδα ONU και ορίζουν τις παραμέτρους των παρεχόμενων συνιστωσών εύρους ζώνης για κάθε κλάση υπηρεσίας T-Cont.

- **Πάγιο εύρος ζώνης (Mbps):** [0..38.88] / Δεν προσφέρεται. Η συνιστώσα του εύρους ζώνης που εκχωρείται σε κάθε περίπτωση, ανεξάρτητα από τις ανάγκες και το συνολικό φόρτο κίνησης στο δίκτυο G-PON.
- **Καθορισμένο εύρος ζώνης (Mbps):** [0..38.88] / Δεν προσφέρεται. Η συνιστώσα του εύρους ζώνης που οφείλει να εκχωρεί το τερματικό OLT αν ζητηθεί από τη κάθε μονάδα ONU.
- **Μέγιστο εύρος ζώνης (Mbps):** [0..38.88] / Συνολικά Διαθέσιμο στη ζεύξη. Η μέγιστη ποσότητα εύρους ζώνης που επιτρέπεται να εκχωρηθεί (συνυπολογίζοντας το πάγιο και το καθορισμένο εύρος ζώνης). Με την επιλογή "All available in link", δεν τίθεται άνω όριο και η κίνηση όλων των μονάδων ONU πολυπλέκεται στη ζεύξη για να αξιοποιηθεί πλήρως το συνολικά διαθέσιμο εύρος ζώνης ροής ανόδου.
- **Εκλεξιμότητα για επιπρόσθετο εύρος ζώνης:** Μη εγγυημένο / Καλύτερης προσπάθειας / Καμία. Οι επιλογές "Μη εγγυημένο" και "Καλύτερης προσπάθειας" κατανέμουν το πλεόνασμα εύρους ζώνης ροής ανόδου με βάση συγκεκριμένα

κριτήρια (μεγαλύτερη προτεραιότητα η επιλογή “Μη εγγυημένο”). Με την επιλογή “Καμία” δεν εκχωρείται επιπρόσθετο εύρος ζώνης, ακόμα και αν αυτό είναι διαθέσιμο στη ροή ανόδου.

3.2.1.2 Η διεργασία του επεξεργαστή του τερματικού OLT

Όπως ήδη αναφέρθηκε, ο επεξεργαστής του τερματικού OLT αποτελεί τη σημαντικότερη δομική μονάδα του τερματικού OLT και κατ’ επέκταση του μοντέλου G-PON συνολικά. Η διεργασία του επεξεργαστή που καθορίζει τη λειτουργία του περιγράφεται από το ακόλουθο διάγραμμα μετάβασης καταστάσεων (state transition diagram):



Σχήμα 40: Διάγραμμα Μετάβασης Καταστάσεων του επεξεργαστή του τερματικού OLT

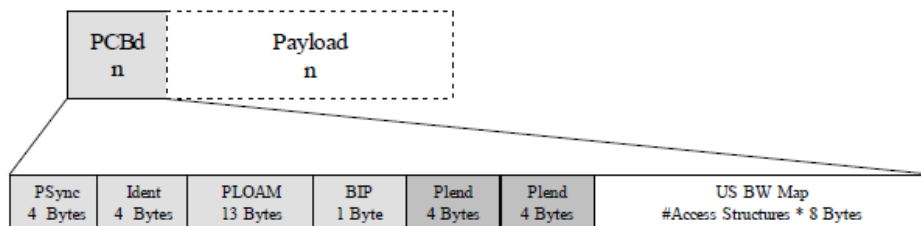
Κατάσταση “init”: Κατά την έναρξη της προσομοίωσης, η διεργασία βρίσκεται στην εξαναγκασμένη κατάσταση “init”, όπου πραγματοποιείται η αρχικοποίηση των τιμών των μεταβλητών που θα χρησιμοποιηθούν και λαμβάνονται τα χαρακτηριστικά που καθορίζουν τη λειτουργία του τερματικού OLT. Επίσης προγραμματίζεται μία διακοπή (τύπου self-interrupt) για την ώρα της λήξης της προσομοίωσης, ώστε να καταγραφούν τα ζητούμενα στατιστικά και αποτελέσματα. Αφού πραγματοποιηθούν οι παραπάνω ενέργειες, ξεκινάει η κυρίως λειτουργία του τερματικού OLT και η διεργασία μεταβαίνει αυτόματα στην μη-εξαναγκασμένη κατάσταση “idle”.

Κατάσταση “idle”: η διεργασία του τερματικού OLT για το υπόλοιπο της προσομοίωσης μεταβαίνει από και προς αυτήν την κατάσταση. Κάθε 125μs προγραμματίζεται νέα διακοπή self-interrupt και καλείται η συνάρτηση “send_gtc_d_frame()”, η οποία αναλαμβάνει τη δημιουργία και την αποστολή του πλαισίου σύγκλισης μετάδοσης G-PON ροής καθόδου (GTC downstream frame) με πολλαπλή εκπομπή προς όλες τις μονάδες ONU. Επιπλέον, κάθε φορά που το τερματικό OLT

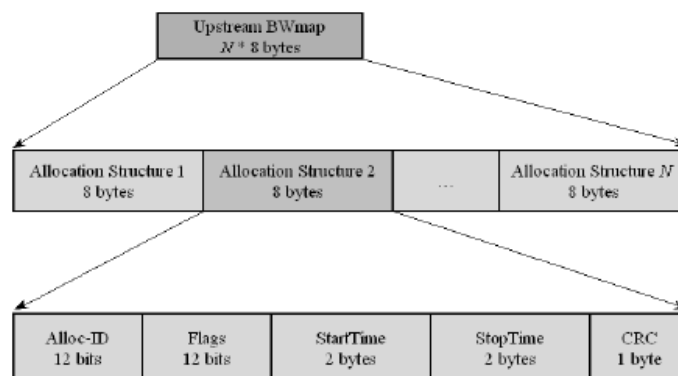
λαμβάνει το πλαίσιο σύγκλισης μετάδοσης G-PON ροής ανόδου (GTC upstream frame) από τις μονάδες ONU, καλείται η συνάρτηση “process_up_burst()”, η οποία είναι υπεύθυνη για την επεξεργασία του πλαισίου. Αναλυτικότερα η λειτουργία των 2 συναρτήσεων:

send_gtc_d_frame():

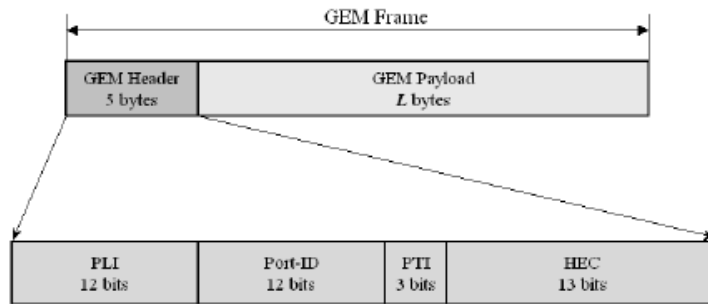
- Με βάση το χαρακτηριστικό “Service Period”, λαμβάνεται η απόφαση αν θα ζητηθεί από τις μονάδες ONU να ενημερώσουν το τερματικό OLT στον τρέχων κύκλο λειτουργίας, σχετικά με το περιεχόμενο των ουρών τους (αναφορά DBRu), τον όγκο δηλαδή των δεδομένων που περιμένουν να εξυπηρετηθούν στους ενταμιευτές τους.
- Με βάση την αποθηκευμένη πληροφορία σχετικά με την κατάσταση των ουρών των μονάδων ONU που διαθέτει το τερματικό OLT και τα χαρακτηριστικά των Περιγραφέων Κίνησης (Traffic Descriptors attributes), εκτελείται ο δυναμικός αλγόριθμος ανάθεσης εύρους ζώνης (DBA algorithm) και υπολογίζονται οι κατανομές εύρους ζώνης ροής ανόδου που εκχωρούνται σε κάθε αναγνωριστικό Alloc-ID, για όλες τις μονάδες ONU.
- Αφού πραγματοποιηθούν οι παραπάνω υπολογισμοί, δημιουργείται το πλαίσιο σύγκλισης μετάδοσης G-PON ροής καθόδου (GTCd frame), το οποίο αποτελείται από το τμήμα PCBd με όλα τα απαραίτητα πεδία που προβλέπονται από το πρωτόκολλο (PSync, Ident, PLOAM, BIP, Plend, Plend, BWmap) και φυσικά το τμήμα φόρτου δεδομένων (payload), με τα πλαίσια δεδομένων χρήση ενθυλακωμένα σε πλαίσια GEM:



Σχήμα 41: Μοντελοποίηση της δομής του τμήματος PCBd



Σχήμα 42: Μοντελοποίηση της δομής της χαρτογράφησης εύρους ζώνης ροής ανόδου



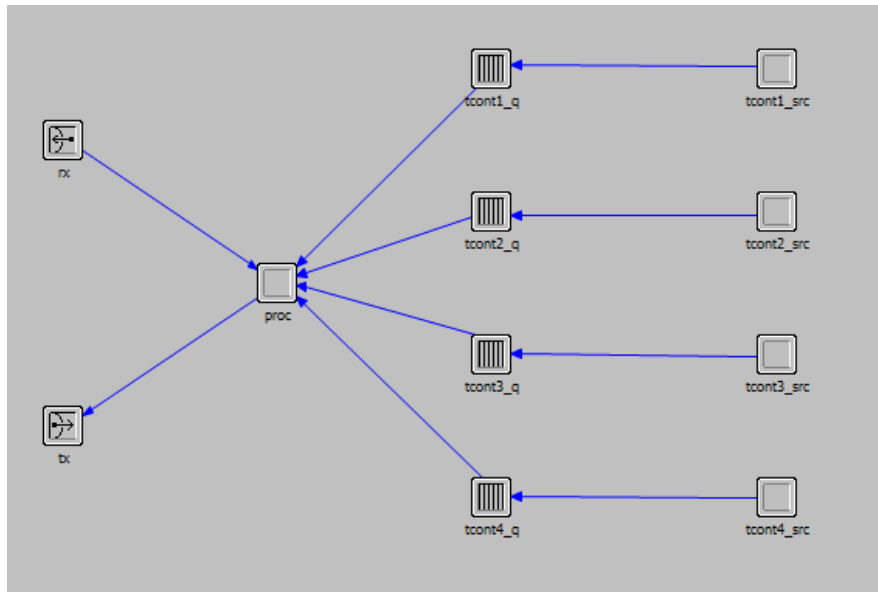
Σχήμα 43: Μοντελοποίηση της δομής ενός πλαισίου GEM

- Το ολοκληρωμένο πλαίσιο GTCd, μήκους 38880 byte, αποστέλλεται με πολλαπλή εκπομπή προς όλες τις μονάδες ONU.
- Τέλος, προγραμματίζεται η επόμενη διακοπή self-interrupt μετά από 125μs, προκειμένου να κληθεί εκ νέου η συνάρτηση send_gtc_d_frame() και να επαναληφθεί η διαδικασία για τον επόμενο κύκλο λειτουργίας.

process_up_burst():

- Αν περιλαμβάνεται στο πλαίσιο GTC ανόδου η αναφορά DBRu, ανανεώνονται οι πίνακες που διατηρεί το τερματικό OLT σχετικά με τον όγκο των δεδομένων που περιμένουν να εξυπηρετηθούν στους ενταμιευτές κάθε μονάδας ONU.
- Για κάθε ένα από τα αναγνωριστικά Alloc-ID των 4 κλάσεων T-Cont κάθε μονάδας ONU, πραγματοποιείται η επεξεργασία των πλαισίων GEM και η εξαγωγή των πλαισίων δεδομένων χρήστη που είναι ενθυλακωμένα σε αυτά. Αν διαπιστωθεί ότι ένα πακέτο είναι τμήμα (fragment) ενός τεμαχισμένου πακέτου, εισάγεται στο κατάλληλο ενταμιευτή επανασυναρμολόγησης (reassembly buffer), όπου επανασυνδέεται με τα υπόλοιπα τμήματα ή περιμένει μέχρι να ληφθούν τα υπόλοιπα τμήματα του αρχικού πακέτου.

3.2.2 Ο κόμβος μονάδας ONU του δικτύου G-PON



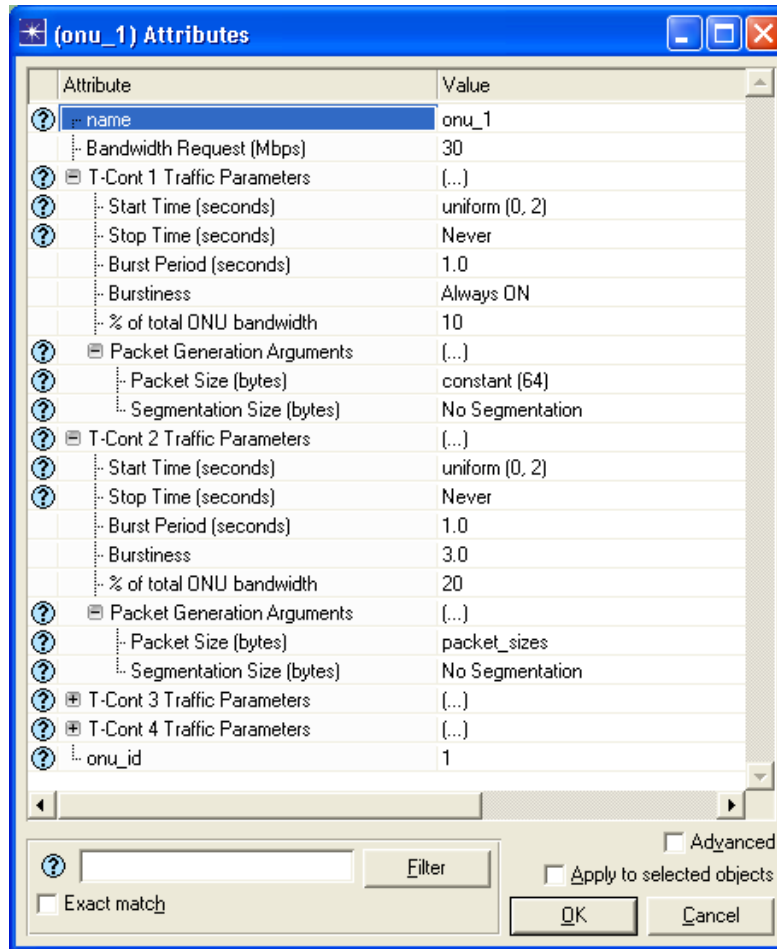
Σχήμα 44: Δομικές μονάδες στο εσωτερικό ενός κόμβου μονάδας ONU

Εξετάζοντας μία μονάδα ONU σε επίπεδο κόμβου, οι δομικές μονάδες από τις οποίες αποτελείται είναι:

- Ένα ζεύγος πομπού και δέκτη σημείου-προς-σημείο, συνδεδεμένους με τις 2 ζεύξεις που εξυπηρετούν την επικοινωνία της μονάδας ONU με το τερματικό OLT.
- 4 επεξεργαστές (“gpon_onu_node.tcont#_src”), σε ρόλο πηγών κίνησης για κάθε μία από τις 4 κλάσεις T-Cont.
- 4 ουρές (“gpon_onu_node.tcont#_q”), σε ρόλο ενταμιευτών για τα πακέτα που παράγονται από τις 4 πηγές και περιμένουν να εξυπηρετηθούν.
- Ένας επεξεργαστής (“gpon_onu_node.proc”), ο οποίος αποτελεί τη βασικότερη δομική μονάδα κάθε μονάδας ONU και ρυθμίζει τον τρόπο λειτουργίας της.

3.2.2.1 Χαρακτηριστικά μονάδας ONU

Κάθε μονάδα ONU διαθέτει χαρακτηριστικά που καθορίζουν τη ποσότητα και τις παραμέτρους της παραγόμενης κίνησης. Αναλυτικά τα διαθέσιμα χαρακτηριστικά και οι προσφερόμενες επιλογές:



Σχήμα 45: Ρυθμίσεις των χαρακτηριστικών μιας μονάδας ONU

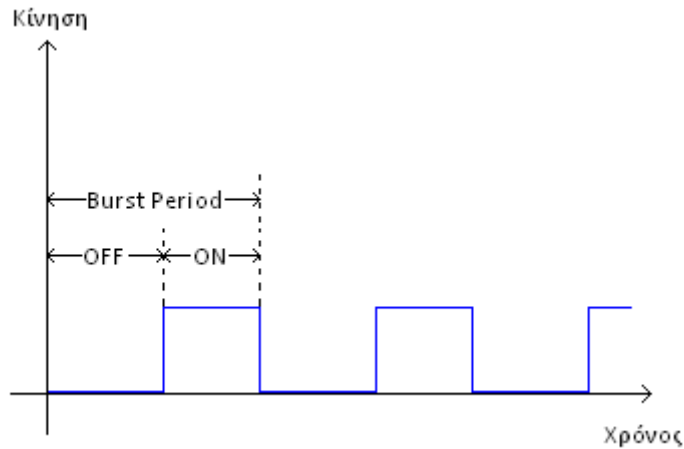
Ζητούμενο εύρος ζώνης (Mbit/s): [0..38.88]. Ορίζει το συνολικό ύψος της παραγόμενης κίνησης από τις 4 πηγές κίνησης της μονάδας ONU.

Παράμετροι Κίνησης, για κάθε κλάση T-Cont ξεχωριστά. Ορίζουν τα μεγέθη που αφορούν την παραγόμενη κίνηση, ή ισοδύναμα τις παραμέτρους λειτουργίας των πηγών.

- *Χρόνος έναρξης & λήξης (δευτερόλεπτα):* Σταθερή / Τυχαία / Άλλη κατανομή / Ποτέ. Ορίζουν τη χρονική στιγμή της έναρξης και της λήξης παραγωγής κίνησης από τις πηγές, η οποία μπορεί να ακολουθεί κάποια από τις συνήθεις κατανομές ή να μη συμβαίνει Ποτέ.
- *Περίοδος έκρηξης (δευτερόλεπτα):* Ορίζει την περίοδο της εκρηκτικής λειτουργίας της πηγής, ή ισοδύναμα τη διάρκεια που η πηγή βρίσκεται σε κατάσταση λειτουργίας και μη-λειτουργίας (κατάσταση ON+OFF).
- *Εκρηκτικότητα:* Θετική πραγματική τιμή / Μόνιμη λειτουργία / Μόνιμη παύση. Ορίζει τον παράγοντα εκρηκτικότητας, από τον οποίο προκύπτει η διάρκεια της κατάστασης λειτουργίας στην περίοδο της έκρηξης:

$$\text{Εκρηκτικότητα Πηγής} = \frac{\text{Διάρκεια Περιόδου Λειτουργίας}}{\text{Διάρκεια Περιόδου Μη - Λειτουργίας}} = \frac{\text{ON state period}}{\text{OFF state period}}$$

Με τις επιλογές “Μόνιμη λειτουργία” / “Μόνιμη παύση”, η πηγή παράγει συνεχόμενα ή καθόλου, αντίστοιχα.

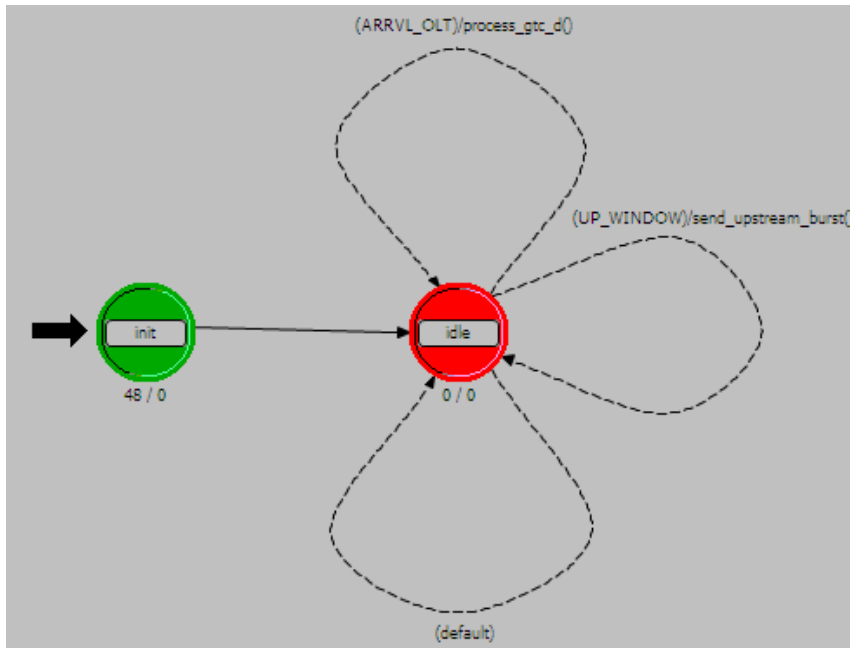


Σχήμα 46: Η εκρηκτική λειτουργία των πηγών κίνησης

- Ποσοστό συνολικού εύρους ζώνης μονάδας ONU (%): [0..100]. Ορίζει το ποσοστό του συνολικού εύρους ζώνης της μονάδας ONU, το οποίο θα αντιπροσωπεύει τη συγκεκριμένη κλάση T-Cont.
- Ορίσματα Παραγωγής Πακέτων: οι παράμετροι που ορίζουν τη μορφή των παραγόμενων πακέτων.
 - Μέγεθος πακέτου (byte): Σταθερή / Τυχαία / Άλλη κατανομή / packet_sizes. Ορίζεται το μέγεθος των πακέτων που θα παραχθούν με βάση κάποια από τις συνηθισμένες ή την κατανομή “packet_sizes”, η οποία προσεγγίζει τα πιο συνηθισμένα μεγέθη πακέτων IPv4.
 - Μέγεθος Τεμαχισμού πακέτων (byte): θετικός ακέραιος / Χωρίς Τεμαχισμό. Ορίζει το μέγεθος πέρα από το οποίο θα πραγματοποιείται τεμαχισμός στα πακέτα, αν αυτό είναι επιθυμητό.

3.2.2.2 Η διεργασία του επεξεργαστή της μονάδας ONU

Ο επεξεργαστής αποτελεί τη σημαντικότερη δομική μονάδα της μονάδας ONU. Η διεργασία του επεξεργαστή, η οποία καθορίζει τη λειτουργία του περιγράφεται από το ακόλουθο διάγραμμα μετάβασης καταστάσεων (state transition diagram):



Σχήμα 47: Διάγραμμα Μετάβασης Καταστάσεων του επεξεργαστή μιας μονάδας ONU

Κατάσταση “init”: Κατά την έναρξη της προσομοίωσης, η διεργασία βρίσκεται στην εξαναγκασμένη κατάσταση “init”, στην οποία πραγματοποιείται η αρχικοποίηση των τιμών των μεταβλητών που θα χρησιμοποιηθούν και λαμβάνονται τα χαρακτηριστικά που καθορίζουν την παραγόμενη κίνηση. Στη συνέχεια, ξεκινάει η κυρίως λειτουργία της μονάδας ONU και η διεργασία μεταβαίνει αυτόματα στη μη-εξαναγκασμένη κατάσταση “idle”.

Κατάσταση “idle”: η διεργασία του επεξεργαστή της μονάδας ONU για το υπόλοιπο της προσομοίωσης μεταβαίνει από και προς αυτήν την κατάσταση. Σε κάθε κύκλο λειτουργίας η μονάδα ONU λαμβάνει το πλαίσιο σύγκλισης μετάδοσης G-PON ροής καθόδου (GTCd frame) από το τερματικό OLT και καλείται η συνάρτηση “process_gtc_d()”, η οποία είναι υπεύθυνη για την επεξεργασία του πλαισίου και τον προσδιορισμό της χρονικής στιγμής που επιτρέπεται στη μονάδα ONU να χρησιμοποιήσει το μέσο για μετάδοση. Σε εκείνη τη χρονική στιγμή καλείται η συνάρτηση “send_upstream_burst”, η οποία είναι υπεύθυνη για τη δημιουργία και την αποστολή της έκρηξης ροής ανόδου, το τμήμα δηλαδή του πλαισίου σύγκλισης μετάδοσης G-PON ροής ανόδου που αντιστοιχεί στη συγκεκριμένη μονάδα ONU. Αναλυτικότερα η λειτουργία των 2 συναρτήσεων:

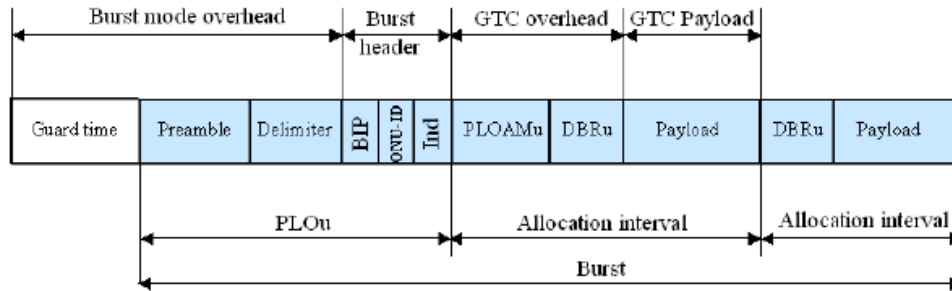
process_gtc_d():

- Εξάγεται η χαρτογράφηση BWmap από το πλαίσιο GTCd, με βάση την οποία υπολογίζονται οι κατανομές εύρους ζώνης που δέχεται κάθε ένα από τα αναγνωριστικά Alloc-ID της μονάδας ONU.
- Με βάση τη πληροφορία που περιέχεται στη χαρτογράφηση BWmap, προγραμματίζεται ένα self-interrupt για τη χρονική στιγμή που η μονάδα ONU δικαιούται πρόσβαση στο μέσο για μετάδοση.
- Με βάση το πεδίο ενδείξεων (flags), διαπιστώνεται αν το τερματικό OLT έχει αιτηθεί την αποστολή αναφοράς DBRu στο επόμενο πλαίσιο ανόδου.

- Πραγματοποιείται επιλογή και επεξεργασία των πακέτων που προορίζονται για τα αναγνωριστικά Alloc-ID της συγκεκριμένης μονάδας ONU.

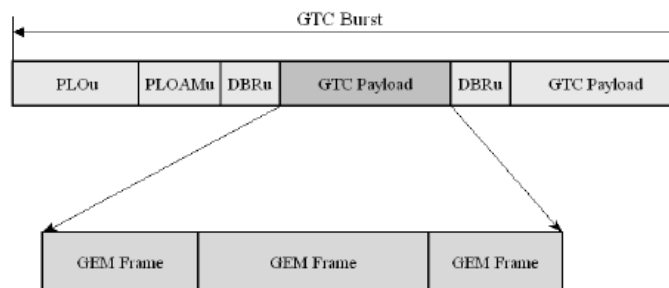
send_upstream_burst():

- Δημιουργείται η έκρηξη ροής ανόδου, με τους προβλεπόμενους χρόνους και πεδία (guard time, preamble, delimiter, BIP, ONU-Id, Ind).



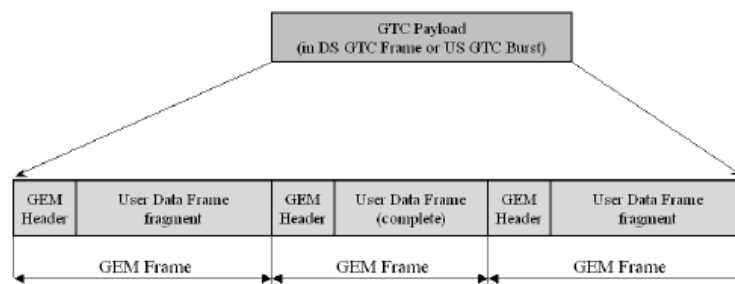
Σχήμα 48: Μοντελοποίηση της δομής μιας έκρηξης μετάδοσης ροής ανόδου

- Για κάθε μία από τις κατανομές εύρους ζώνης ροής ανόδου, ζητείται ο καθορισμένος στη χαρτογράφηση BWmap όγκος δεδομένων από τον αντίστοιχο ενταμιευτή. Τα πακέτα παραλαμβάνονται, ενθυλακώνονται σε πλαίσια GEM και ενσωματώνονται στο τμήμα φόρτου δεδομένων (payload) της κατάλληλης κλάσης T-Cont.



Σχήμα 49: Μοντελοποίηση του τμήματος φόρτου δεδομένων χρήστη

Όπως ισχύει και στα πλαίσια GEM που ενθυλακώνονται στο πλαίσιο GTCd, τα δεδομένα που περιέχονται σε αυτά μπορεί να είναι ολόκληρα πακέτα ή τεμάχια (segments). Επομένως αν είναι απαραίτητο πραγματοποιούνται οι κατάλληλες διαδικασίες τεμαχισμού.



Σχήμα 50: Μοντελοποίηση των πλαισίων GEM

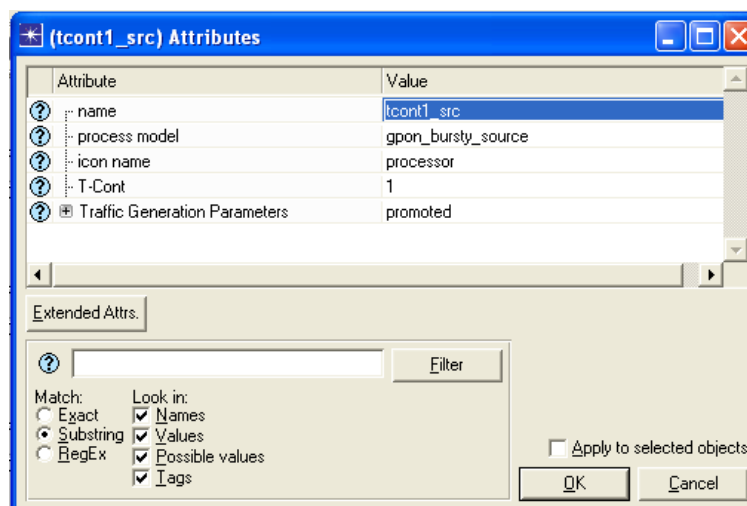
- Αν έχει ζητηθεί από το τερματικό OLT, λαμβάνεται από κάθε ενταμιευτή η απαιτούμενη πληροφορία σχετικά με τον όγκο των δεδομένων του και δημιουργείται η αναφορά DBRu, η οποία ενσωματώνεται στην κατανομή εύρους ζώνης της αντίστοιχης κλάσης T-Cont.
- Η έκρηξη ροής ανόδου μεταδίδεται στο μέσο, όπου θα σχηματίσει μαζί με τις εκρήξεις των άλλων μονάδων ONU το ολοκληρωμένο πλαίσιο σύγκλισης μετάδοσης G-PON ροής ανόδου πριν παραληφθεί από το τερματικό OLT.

3.2.3 Η διεργασία της πηγής κίνησης “gpon_bursty_source”

Κάθε μία από τις 4 πηγές κίνησης των μονάδων ONU λειτουργεί με βάση το μοντέλο διεργασίας “gpon_bursty_source”, το οποίο είναι βασισμένο στο μοντέλο “bursty_source” του περιβάλλοντος προσομοίωσης OPNET, με τις κατάλληλες τροποποιήσεις ώστε να ικανοποιεί τις ανάγκες της προσομοίωσης του δικτύου G-PON. Τα χαρακτηριστικά που ρυθμίζουν τον τρόπο λειτουργίας της πηγής κίνησης είναι:

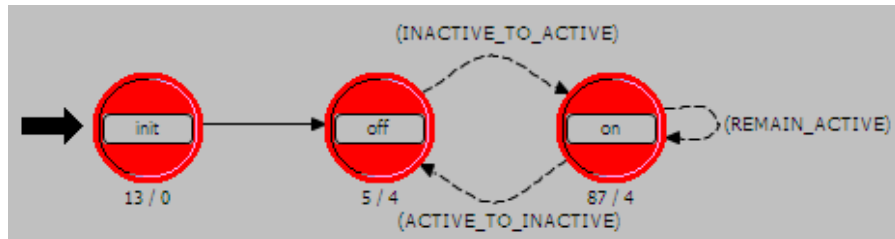
Κλάση υπηρεσίας T-Cont: [1..4]. Καθορίζει σε ποια από τις 4 κλάσεις υπηρεσίας θα ανήκουν τα παραγόμενα πακέτα.

Παράμετροι Παραγωγής Κίνησης: καθορίζουν ορισμένες από τις παραμέτρους της παραγόμενης κίνησης που αναλύθηκαν στην προηγούμενη ενότητα. Έχουν προαχθεί (promoted) σε ανώτερο επίπεδο, ώστε να ρυθμίζονται πιο άμεσα μέσω των χαρακτηριστικών του κόμβου της κάθε μονάδας ONU. Επιπλέον με αυτόν τον τρόπο είναι δυνατή η προσαρμοσμένη λειτουργία κάθε μονάδας ONU, ώστε να εμφανίζει μοναδική συμπεριφορά.



Σχήμα 51: Ρυθμίσεις χαρακτηριστικών μιας πηγής κίνησης

Η λειτουργία της διεργασίας των πηγών κίνησης περιγράφεται από το ακόλουθο διάγραμμα μετάβασης καταστάσεων:



Σχήμα 52: Διάγραμμα Μετάβασης Καταστάσεων του επεξεργαστή μιας πηγής κίνησης

Κατάσταση “init”: η διεργασία της πηγής κίνησης εκκινεί από αυτή τη μη-εξαναγκασμένη κατάσταση, όπου:

- Καλείται η συνάρτηση `bursty_source_sv_init ()`, υπεύθυνη για την αρχικοποίηση των `state variables` της διεργασίας και τον υπολογισμό ορισμένων μεγεθών. Αναλυτικά, λαμβάνονται τα απαραίτητα χαρακτηριστικά που καθορίζουν τη λειτουργία της πηγής κίνησης (κλάση `T-Cont`, παράμετροι παραγόμενης κίνησης, ζητούμενο εύρος ζώνης) και υπολογίζονται οι χρόνοι παραμονής στη κατάσταση λειτουργίας / μη-λειτουργίας (`ON/OFF state duration`). Ακόμα φορτώνεται η κατανομή από την οποία προκύπτει το μέγεθος των παραγόμενων πακέτων και σε συνδυασμό με το συνολικό ζητούμενο εύρος ζώνης υπολογίζεται ο χρόνος μεταξύ της άφιξης δύο πακέτων (`packet interarrival time`).
- Προγραμματίζεται μία διακοπή `self-interrupt` στο χρόνο έναρξης λειτουργίας της πηγής, όταν η διεργασία μεταβαίνει στη μη-εξαναγκασμένη κατάσταση “off”.

Κατάσταση “off”: η διεργασία παραμένει στη συγκεκριμένη κατάσταση, όσο βρίσκεται σε περίοδο μη παραγωγής πακέτων. Κάθε φορά που πραγματοποιείται μετάβαση προς αυτή την κατάσταση, προγραμματίζεται μία νέα διακοπή `self-interrupt` για τη χρονική στιγμή όταν πρέπει να επανεκκινήσει η παραγωγή πακέτων. Τότε η διεργασία μεταβαίνει στη μη-εξαναγκασμένη κατάσταση “on” (μετάβαση “`INACTIVE_TO_ACTIVE`”).

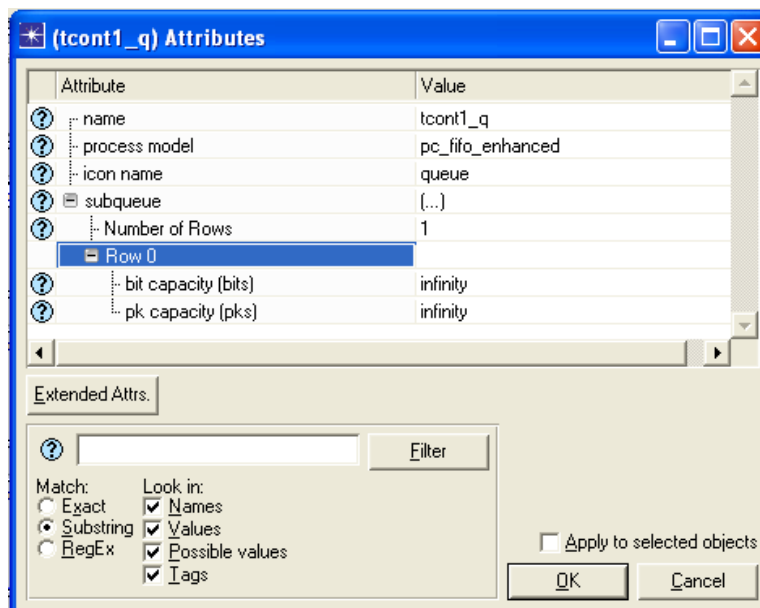
Κατάσταση “on”: η διεργασία παραμένει στη συγκεκριμένη κατάσταση, όσο βρίσκεται σε περίοδο παραγωγής πακέτων. Πραγματοποιούνται οι ακόλουθες ενέργειες:

- Αν η διεργασία προέρχεται από την κατάσταση “off”, προγραμματίζεται μία διακοπή `self-interrupt` για τη χρονική στιγμή όπου είναι επιθυμητό να σταματήσει η παραγωγή πακέτων και να γίνει εκ νέου μετάβαση στην κατάσταση “off” (μετάβαση “`ACTIVE_TO_INACTIVE`”).
- Δημιουργείται ένα νέο πακέτο, το μέγεθος του οποίου καθορίζεται με βάση την κατανομή που έχει επιλεγεί στα χαρακτηριστικά της πηγής. Δημιουργείται στο πακέτο ένα πεδίο όπου καταγράφεται η κλάση `T-Cont` της πηγής.

- Αν είναι επιθυμητό, το πακέτο τεμαχίζεται σε τμήματα με μέγεθος ορισμένο από τα χαρακτηριστικά της κίνησης.
- Το πακέτο (ή τα τεμάχια αυτού) αποστέλλεται στον ενταμιευτή που αντιστοιχεί στη συγκεκριμένη πηγή κίνησης.
- Ελέγχεται αν ο χρόνος παραγωγής του επόμενου πακέτου είναι εντός της τρέχουσας περιόδου λειτουργίας. Αν ναι, η διεργασία μεταβαίνει εκ νέου στην κατάσταση “on” (μετάβαση “REMAIN_ACTIVE”). Αλλιώς πραγματοποιείται μετάβαση στην κατάσταση “off”.

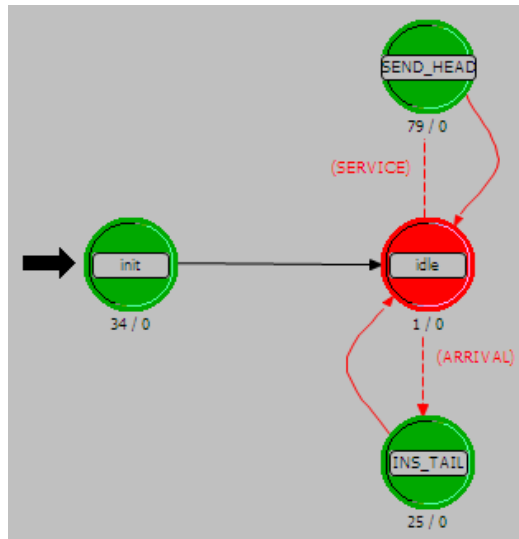
3.2.4 Η διεργασία του ενταμιευτή “pc_fifo_enhanced”

Κάθε μονάδα ONU διαθέτει για κάθε πηγή κίνησης έναν ενταμιευτή, στον οποίο αποθηκεύονται τα παραγόμενα πακέτα μέχρι να εξυπηρετηθούν. Οι ενταμιευτές λειτουργούν με βάση το μοντέλο διεργασίας “pc_fifo_enhanced”, μια παραλλαγή της FIRST IN – FIRST OUT ουράς “pc_fifo” του OPNET, με τις κατάλληλες προσθήκες ώστε να επαρκεί για τις ανάγκες της προσομοίωσης του δικτύου G-PON. Τα χαρακτηριστικά των ενταμιευτών που ορίζουν τη λειτουργία τους είναι η χωρητικότητα σε bits και πακέτα, η οποία ορίζει το μέγεθος της ουράς, πέρα από το οποίο επιπλέον πακέτα θα απορρίπτονται.



Σχήμα 53: Ρυθμίσεις χαρακτηριστικών ενός ενταμιευτή δεδομένων

Η λειτουργία της διεργασίας ενός ενταμιευτή περιγράφεται από το ακόλουθο διάγραμμα μετάβασης καταστάσεων:



Σχήμα 54: Διάγραμμα Μετάβασης Καταστάσεων του επεξεργαστή ενός ενταμιευτή

Κατάσταση “init”: η διεργασία του ενταμιευτή εκκινεί από αυτήν την εξαναγκασμένη κατάσταση, όπου αρχικοποιούνται οι τιμές των μεταβλητών κατάστασης και λαμβάνεται το χαρακτηριστικό “Print to File – Queue Report”, το οποίο καθορίζει αν θα τυπώνονται σε αρχείο οι δραστηριότητες της ουράς κατά τη διάρκεια της προσομοίωσης. Στη συνέχεια, ξεκινάει η κυρίως λειτουργία του ενταμιευτή και η διεργασία μεταβαίνει αυτόματα στη μη-εξαναγκασμένη κατάσταση “idle”.

Κατάσταση “idle”: για το υπόλοιπο της προσομοίωσης η διεργασία του ενταμιευτή μεταβαίνει από και προς αυτήν την κατάσταση. Κάθε φορά που ένα νέο πακέτο εισέρχεται στην ουρά, η διεργασία ακολουθεί τη μετάβαση “ARRIVAL” στην εξαναγκασμένη κατάσταση “INS_TAIL” και στη συνέχεια επανέρχεται στην κατάσταση “idle”. Παρόμοια, όταν η διεργασία δέχεται απομακρυσμένη διακοπή (remote interrupt) από τον επεξεργαστή της μονάδας ONU με αίτηση για συγκεκριμένο όγκο δεδομένων, ακολουθεί τη μετάβαση “SERVICE” στην εξαναγκασμένη κατάσταση “SEND_HEAD” και στη συνέχεια επιστρέφει στην κατάσταση “idle”.

Κατάσταση “INS_TAIL”: οι ενέργειες που πραγματοποιούνται σε αυτή την κατάσταση είναι:

- Παραλαμβάνεται το νέο πακέτο, το οποίο τοποθετείται στο τέλος της ουράς.
- Ανανεώνονται οι τιμές των μεταβλητών κατάστασης: μέγεθος πρώτου πακέτου στην ουρά (head_pk_size), συνολικός όγκος δεδομένων στην ουρά (ttl_queue_data) και συνολικός αριθμός πακέτων στην ουρά (ttl_queue_pkts).
- Αν έχει ζητηθεί, τυπώνεται σε αρχείο η τρέχουσα δραστηριότητα.

Κατάσταση “SEND_HEAD”: σε αυτή την κατάσταση:

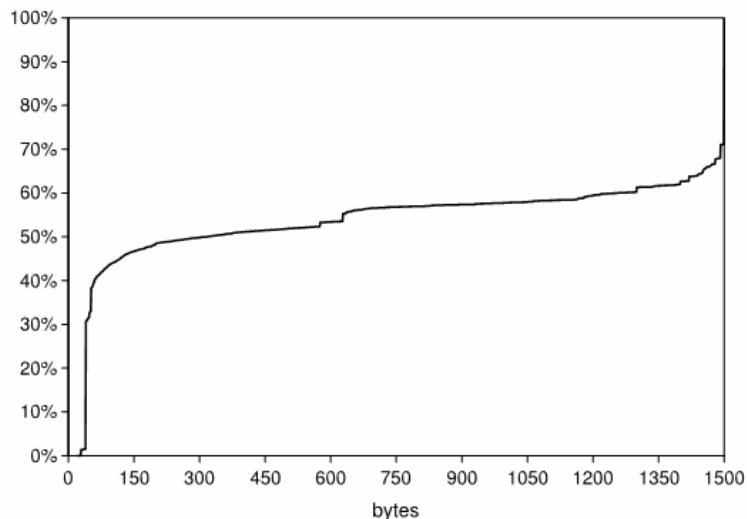
- Με βάση τον κωδικό της απομακρυσμένης διακοπής (remote interrupt) που δέχεται ο ενταμιευτής από τον επεξεργαστή της μονάδας ONU, υπολογίζεται ο όγκος των δεδομένων που αιτούνται για αποστολή.

- Εξάγονται από την αρχή της ουράς όσα πακέτα επαρκούν για την κάλυψη του αιτούμενου όγκου δεδομένου, εκτός αν τα διαθέσιμα πακέτα είναι αθροιστικά μικρότερα σε μέγεθος από τον αιτούμενο όγκο. Το τελευταίο πακέτο ενδέχεται να τεμαχιστεί ώστε να προκύψει ακριβής αντιστοιχία αιτούμενου και αποστελλόμενου όγκου δεδομένων. Λαμβάνεται υπόψιν ότι για κάθε πακέτο θα χρειαστούν 40 bit επιπλέον, για την επικεφαλίδα του πλαισίου GEM.
- Τα πακέτα αποστέλλονται στον επεξεργαστή της μονάδας ONU.
- Ανανεώνονται οι τιμές των μεταβλητών: μέγεθος πρώτου πακέτου στην ουρά (head_pk_size), συνολικός όγκος δεδομένων στην ουρά (ttl_queue_data) και συνολικός αριθμός πακέτων στην ουρά (ttl_queue_pkts).
- Αν έχει ζητηθεί, τυπώνεται σε αρχείο η τρέχουσα δραστηριότητα.

3.2.5 Η μορφή της παραγόμενης κίνησης

Ο ρόλος των 4 πηγών κίνησης στο εσωτερικό του κόμβου κάθε μονάδας ONU είναι η παραγωγή των πακέτων δεδομένων του χρήστη, τα οποία αποτελούν το μεγαλύτερο ποσοστό της συνολικής κίνησης στο δίκτυο G-PON.

Η κίνηση που ανήκει στην κλάση T-Cont 1 αποτελείται από πακέτα δεδομένων σταθερού μήκους, αφού η κίνηση είναι σταθερού ρυθμού. Για τις κλάσεις υπηρεσίας T-Cont 2, 3 και 4 τα παραγόμενα πακέτα είναι μεταβλητού μήκους, το οποίο ακολουθεί μία προσέγγιση της κατανομής πιθανότητας του μήκους των πακέτων IPv4 [12].



Σχήμα 55: Αθροιστική κατανομή πιθανότητας του μήκους των πακέτων IPv4

Για τις ανάγκες μοντελοποίησης της παραγόμενης κίνησης, η κατανομή των μηκών των πακέτων δεδομένων χρήστη προσεγγίστηκε με τις ακόλουθες τιμές:

Ποσοστό παραγόμενων πακέτων	Μήκος πακέτου (byte)
44 %	46
37 %	1488
9 %	278
7 %	980
1.76 %	628
1.1 %	1300
0.14 %	4470

Πίνακας 2: Κατανομή πιθανοτήτων των μηκών των πακέτων δεδομένων χρήστη

ΚΕΦΑΛΑΙΟ 4^ο : Εκτέλεση προσομοιώσεων, Αποτελέσματα & Συμπεράσματα

Στο παρόν κεφάλαιο εκτελούνται προσομοιώσεις της λειτουργίας του μοντέλου που δημιουργήθηκε στα πλαίσια της διπλωματικής εργασίας. Τα στατιστικά που προκύπτουν από τις προσομοιώσεις χρησιμοποιούνται για τον υπολογισμό των παραμέτρων απόδοσης του δικτύου, οι οποίες απεικονίζονται μέσω των γραφικών παραστάσεων. Οι παράμετροι απόδοσης που υπολογίστηκαν για το δίκτυο G-PON είναι:

- *Μέση καθυστέρηση πακέτων από άκρο σε άκρο*, ανά κλάση υπηρεσίας T-Cont: δεδομένου ότι η τιμή της καθυστέρησης διάδοσης στο μέσο είναι μηδενική, η συγκεκριμένη καθυστέρηση αναφέρεται στο μέσο χρόνο αναμονής των πακέτων στους ενταμιευτές των μονάδων ONU ως τη στιγμή που εξυπηρετείται και το τελευταίο τμήμα τους.
- *Τυπική απόκλιση της καθυστέρησης (jitter) των πακέτων*, ανά κλάση υπηρεσίας T-Cont
- *Συνάρτηση πυκνότητας πιθανότητας της καθυστέρησης των πακέτων*, ανά κλάση T-Cont
- *Αποδοτικότητα στη κατεύθυνση ροής ανόδου*: το συγκεκριμένο μέγεθος ορίζεται ως ο λόγος του συνολικού (για κάθε κύκλο λειτουργίας j) όγκου των δεδομένων που αντιστοιχεί στα πλαίσια δεδομένων του χρήστη, προς το συνολικό όγκο δεδομένων που μεταφέρθηκε στη ροή ανόδου, από κάθε μονάδα ONU i .

$$\text{Αποδοτικότητα ροής ανόδου} = \frac{\sum_{i,j}(B_{i,j}^{data})}{\sum_{i,j}(B_{i,j}^{data} + B_{i,j}^{DBRu} + B_{i,j}^{GEMheader} + B_{i,j}^{guard} + B_{i,j}^{PLOu})}$$

4.1 Ρυθμίσεις λειτουργίας και αποτελέσματα

Για τις ανάγκες της προσομοίωσης λειτουργίας του δικτύου υπό διάφορες συνθήκες φόρτου, πραγματοποιήθηκαν οι απαραίτητες ρυθμίσεις στις μονάδες ONU καθώς και το τερματικό OLT.

- Κάθε μονάδα ONU αντιμετωπίζεται από το μοντέλο του πρωτοκόλλου G-PON ως ισότιμη μονάδα, επομένως το συνολικά διαθέσιμο εύρος ζώνης ροής ανόδου των 1.24416 Gbit/s μοιράζεται ισομερώς στις 32 μονάδες ONU, αντιστοιχώντας σε εύρος ζώνης ίσο με 38.88 Mbit/s για κάθε μονάδα.
- Δεδομένου του κύκλου λειτουργίας των 125 μ s του πρωτοκόλλου G-PON, η διάρκεια κάθε προσομοίωσης ορίσθηκε στα 10 δευτερόλεπτα.
- Η επίδραση του κύκλου δειγματοληψίας (polling cycle), δηλαδή της χρονικής περιόδου κατά την οποία το τερματικό OLT αιτείται την αποστολή της αναφοράς DBRu από τις μονάδες ONU, εξετάσθηκε για τις τιμές 125, 250, 500, 1000 και 2000 μ s.
- Το μήκος του πεδίου αναφοράς DBRu ορίσθηκε σε 1 byte (για κάθε κλάση T-Cont).

4.1.1 Ρυθμίσεις κίνησης

Οι μονάδες ONU λειτουργούν στο δίκτυο G-PON ως πηγές κίνησης, επομένως από αυτές εξαρτάται το συνολικό επίπεδο φόρτου του δικτύου. Μέσω των χαρακτηριστικών κάθε μονάδας ONU, είναι δυνατό να ρυθμιστεί τόσο ο όγκος όσο και η μορφή της παραγόμενης κίνησης, όπως περιγράφηκε στο προηγούμενο κεφάλαιο.

Πραγματοποιήθηκαν συνολικά τρεις σειρές προσομοιώσεων, κάθε μία για διαφορετικά επίπεδα φόρτου κίνησης: 30%, 70% και 90% της χωρητικότητας της ροής ανόδου. Ως μέγιστο επίπεδο φόρτου κίνησης επιλέχθηκε το 90%, λόγω του ότι επιπλέον κίνηση στο δίκτυο οδηγεί σε συμφόρηση στους ενταμιευτές των πακέτων κλάσης T-Cont 4, η οποία έχει τη χαμηλότερη προτεραιότητα. Η μορφή της παραγόμενης κίνησης που ανήκει στις 4 κλάσεις T-Cont παρέμεινε ανεξάρτητη του όγκου της:

Κλάση Υπηρεσίας	Χρόνος Έναρξης (sec)	Χρόνος Λήξης (sec)	Περίοδος Έκρηξης (sec)	Εκρηκτικότητα Πηγής	Ποσοστό συνολικού εύρους ζώνης της μονάδας ONU (%)	Μέγεθος παραγόμενων πακέτων (byte)
T-Cont 1	[0..2]	Ποτέ	1	Πάντα Ενεργή	10	64
T-Cont 2	[0..2]	Ποτέ	1	3	20	Packet_sizes
T-Cont 3	[0..2]	Ποτέ	1	5	30	Packet_sizes
T-Cont 4	[0..2]	Ποτέ	1	8	40	Packet_sizes

Πίνακας 3: Μορφή παραγόμενης κίνησης

Διευκρινίσεις:

- Οι χρόνοι έναρξης των πηγών είναι τυχαίοι εντός του διαστήματος [0..2].
- Οι πηγές κίνησης κλάσης T-Cont 1 παραμένουν πάντα ενεργές (always on).
- Οι πηγές κίνησης κλάσης T-Cont 1 παράγουν πακέτα σταθερού μήκους 64 byte, ενώ οι υπόλοιπες πηγές παράγουν πακέτα μήκους που ακολουθεί την κατανομή "packet_sizes", η οποία αναλύθηκε στο προηγούμενο κεφάλαιο.

4.1.2 Ρυθμίσεις περιγραφών κίνησης

Ανάλογα με το φόρτο κίνησης του δικτύου, μεταβάλλονται και ορισμένες συνιστώσες των περιγραφών κίνησης των 4 κλάσεων υπηρεσίας:

- Πάγιο εύρος ζώνης: ανεξάρτητο του φόρτου κίνησης στο δίκτυο. Η κλάση T-Cont 1 λαμβάνει σε κάθε περίπτωση 4.5Mbit/s, ενώ οι κλάσεις T-Cont 2, 3 και 4 μηδενικό.
- Καθορισμένο εύρος ζώνης: οι κλάσεις T-Cont 1 και 4 δε λαμβάνουν καθορισμένο εύρος ζώνης. Οι κλάσεις T-Cont 2 και 3 λαμβάνουν 10% και 15% του συνολικού εύρους ζώνης που αιτείται η μονάδα ONU, αντίστοιχα.
- Μέγιστο εύρος ζώνης: ανεξάρτητο του φόρτου κίνησης στο δίκτυο. Η τιμή του μέγιστου εύρους ζώνης που δικαιούται η κλάση T-Cont 1 είναι ίση με το πάγιο εύρος ζώνης

- Εκλεξιμότητα για επιπρόσθετο εύρος ζώνης: ανεξάρτητη του φόρτου κίνησης στο δίκτυο. Η κλάση T-Cont 1 δεν δικαιούται επιπρόσθετο εύρος ζώνης, οι κλάσεις T-Cont 2 και 3 έχουν εκλεξιμότητα τύπου “Μη εγγυημένο”, ενώ η T-Cont 4 “Καλύτερης προσπάθειας”.

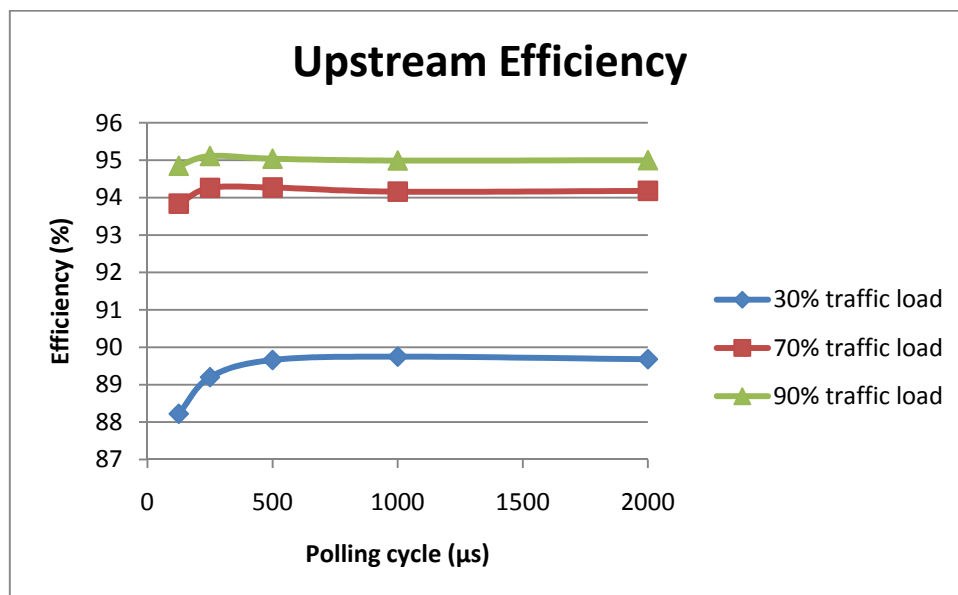
Οι τιμές των περιγραφών κίνησης για φόρτους κίνησης 30%, 70% και 90%, ή ισοδύναμα 11.6 Mbit/s, 27.2Mbit/s και 35.0Mbit/s από κάθε μονάδα ONU, συγκεντρώνονται στους ακόλουθους πίνακες:

Κλάση Υπηρεσίας	Πάγιο εύρος ζώνης (Mbit/s)	Καθορισμένο εύρος ζώνης(Mbit/s)	Μέγιστο εύρος ζώνης(Mbit/s)	Εκλεξιμότητα για επιπρόσθετο εύρος ζώνης
T-Cont 1	4.5	-	4.5	Καμία
T-Cont2	-	1.16/2.72/3.5	Συνολικό πλεόνασμα	Μη εγγυημένο
T-Cont 3	-	1.74/4.08/5.25	Συνολικό πλεόνασμα	Μη εγγυημένο
T-Cont 4	-	-	Συνολικό πλεόνασμα	Καλύτερης προσπάθειας

Πίνακας 4: Τιμές περιγραφών κίνησης για φόρτο δικτύου 30% / 70% / 90%

Διευκρίνιση: οι κλάσεις υπηρεσίας T-Cont 2, 3 και 4 δεν έχουν περιορισμό στο μέγιστο εύρος ζώνης που δικαιούνται, όσο υπάρχει πλεόνασμα εύρους ζώνης στη ζεύξη ανόδου.

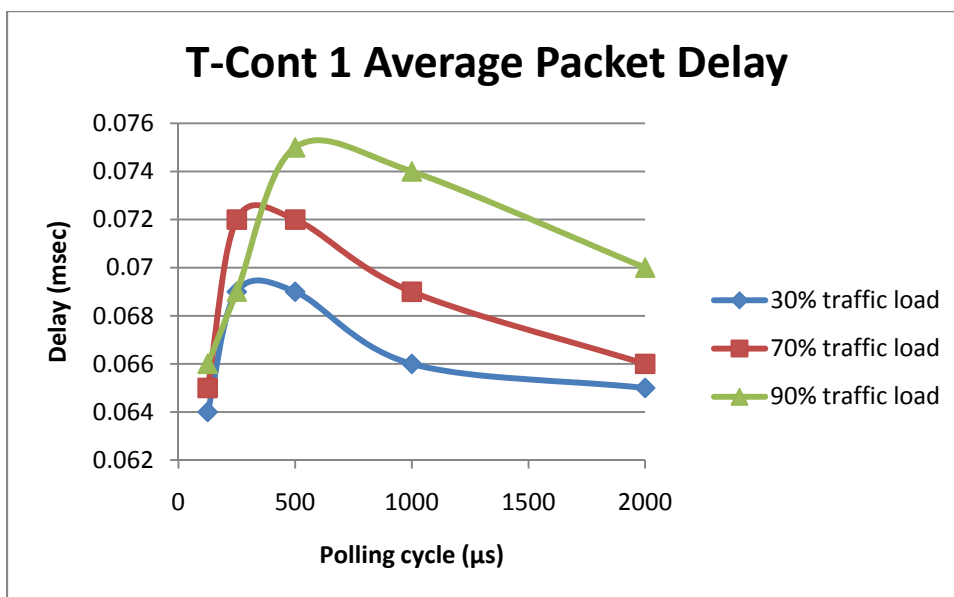
4.2 Γραφικές παραστάσεις



Σχήμα 56: Αποδοτικότητα ροής ανόδου

Όπως είναι αναμενόμενο, η αύξηση του κύκλου δειγματοληψίας επιφέρει μικρή αύξηση στην αποδοτικότητα ροής ανόδου, αφού αποστέλλονται λιγότερες αναφορές DBRu και το εύρος ζώνης που εξοικονομείται μπορεί να χρησιμοποιηθεί για τη μεταφορά μεγαλύτερου όγκου δεδομένων.

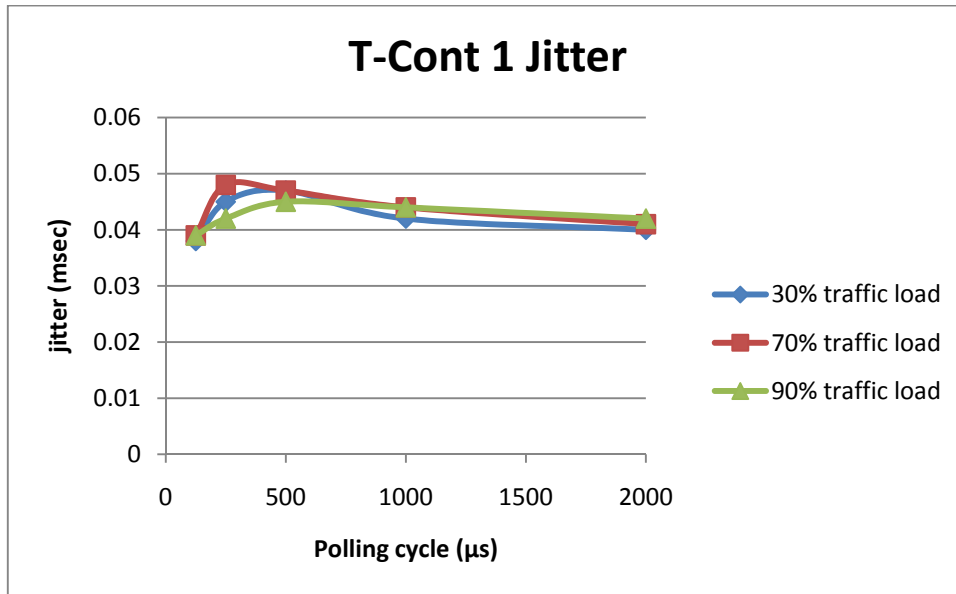
Υψηλότερη αύξηση παρατηρείται για επίπεδο φόρτου 30%, κατά την αύξηση του κύκλου δειγματοληψίας από 125 σε 500 μs , με ποσοστιαία μεταβολή 1.63%. Για επίπεδα φόρτου 70% και 90% η αντίστοιχη μεταβολή είναι μικρότερη, ίση με 0.45% και 0.2% αντίστοιχα. Επίσης είναι εμφανές ότι περαιτέρω αύξηση του κύκλου δειγματοληψίας δεν επιφέρει βελτίωση στην αποδοτικότητα ροής ανόδου, ωστόσο εισάγει σημαντική πρόσθετη καθυστέρηση όπως φαίνεται στις γραφικές παραστάσεις που ακολουθούν.



Σχήμα 57: Μέση καθυστέρηση πακέτων κλάσης T-Cont 1

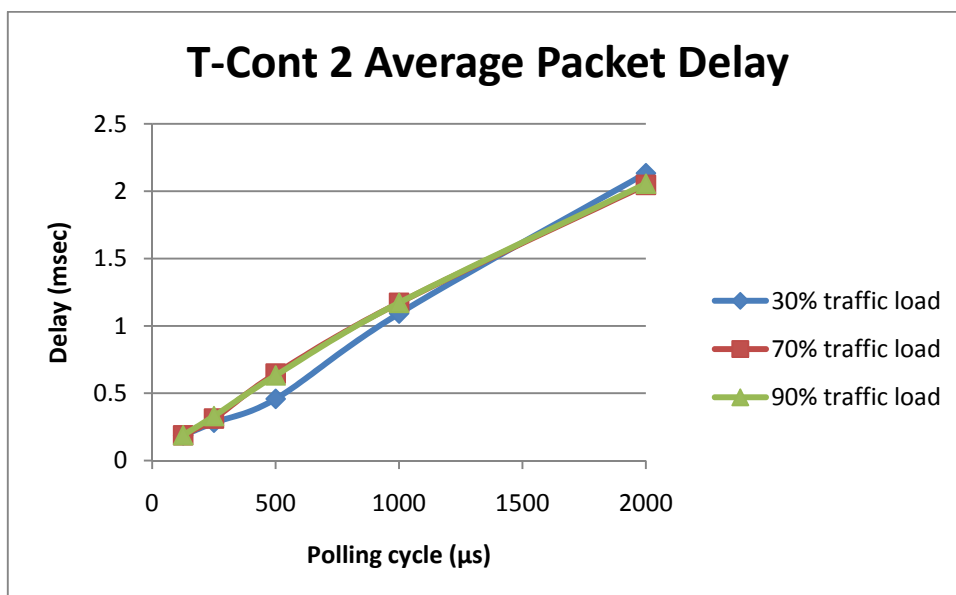
Η αύξηση του κύκλου δειγματοληψίας επιφέρει αμελητέα επίδραση στη μέση καθυστέρηση των πακέτων κλάσης T-Cont 1, η οποία για κάθε επίπεδο φόρτου περιορίζεται σε τιμές μικρότερες των 75 μs . Πράγματι, όπως ήταν αναμενόμενο από τη σχεδίαση του μοντέλου και τη συνιστώσα πάγιου εύρους ζώνης που καθορίζεται στους περιγραφείς κίνησης της κλάσης T-Cont 1, η συγκεκριμένη κλάση είναι κατάλληλη για υποστήριξη υπηρεσιών με απαιτητικές εγγυήσεις ποιότητας ως προς τη μέγιστη ανεκτή καθυστέρηση, υπό οποιεσδήποτε συνθήκες φόρτου στο δίκτυο.

Η απαίτηση για χαμηλά και φραγμένα επίπεδα καθυστέρησης ικανοποιείται όπως αποδεικνύεται και από τη γραφική παράσταση του jitter των πακέτων της κλάσης T-Cont 1:

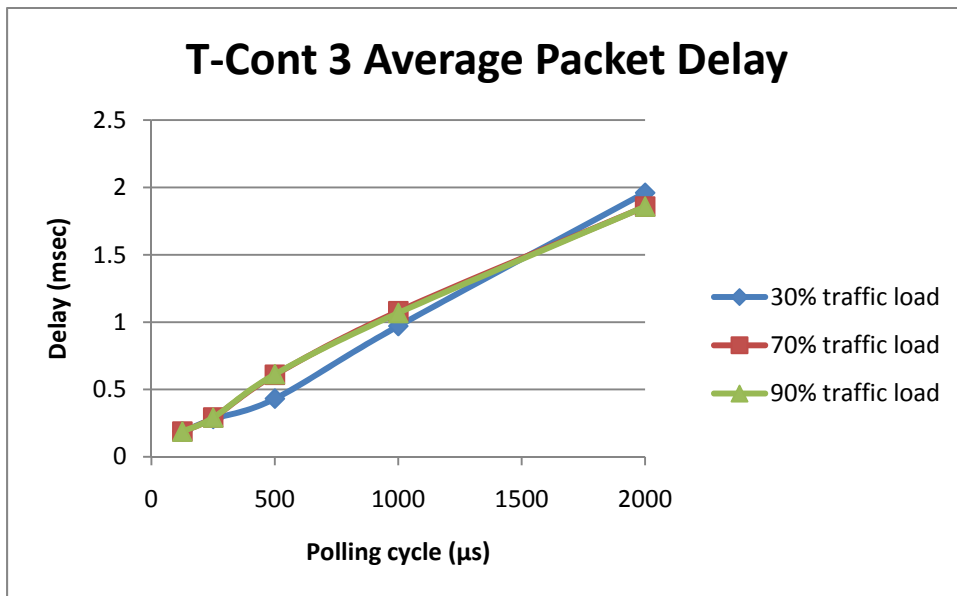


Σχήμα 58: Jitter πακέτων κλάσης T-Cont 1

Οι κλάσεις υπηρεσιών T-Cont 2 και 3 επηρεάζονται άμεσα από την αύξηση του κύκλου δειγματοληψίας, όπως είναι αναμενόμενο δεδομένου ότι δε διαθέτουν συνιστώσα πάγιου εύρους ζώνης. Οι προσεγγιστικά ευθείες γραμμές των γραφικών παραστάσεων δείχνουν ότι με την αύξηση της τιμής του κύκλου δειγματοληψίας, εισάγεται αναλογικά επιπρόσθετη καθυστέρηση στη μεταφορά των πακέτων που ανήκουν σε αυτές τις κλάσεις. Σημειώνεται ότι οι τιμές της μέσης καθυστέρησης των συγκεκριμένων κλάσεων δεν επηρεάζονται σχεδόν καθόλου από την αύξηση στο φόρτο του δικτύου, γεγονός το οποίο οφείλεται στη συνιστώσα καθορισμένου εύρους ζώνης, ενώ σε κάθε περίπτωση η μέση καθυστέρηση δεν υπερβαίνει τα 2.13 ms.

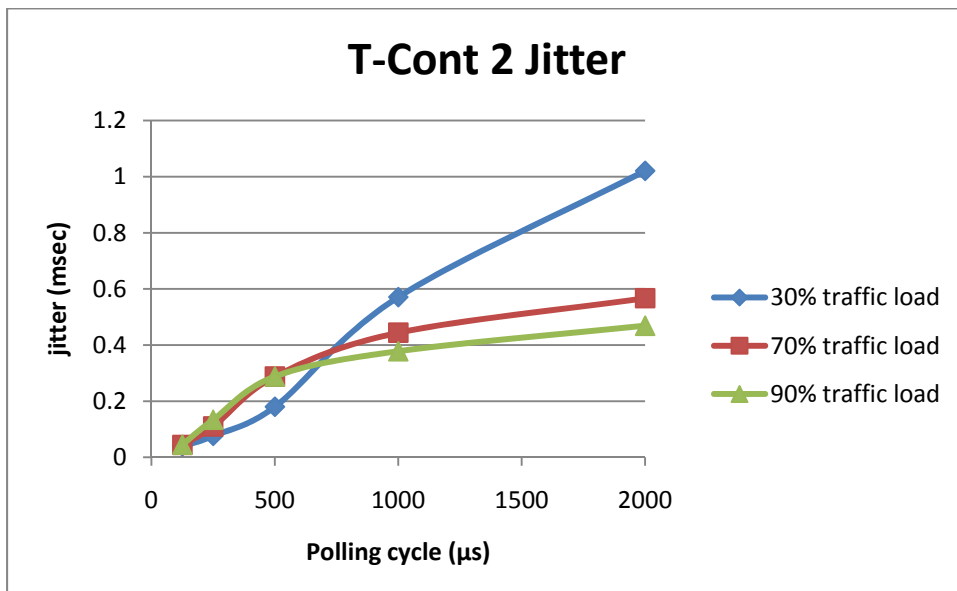


Σχήμα 59: Μέση καθυστέρηση πακέτων κλάσης T-Cont 2

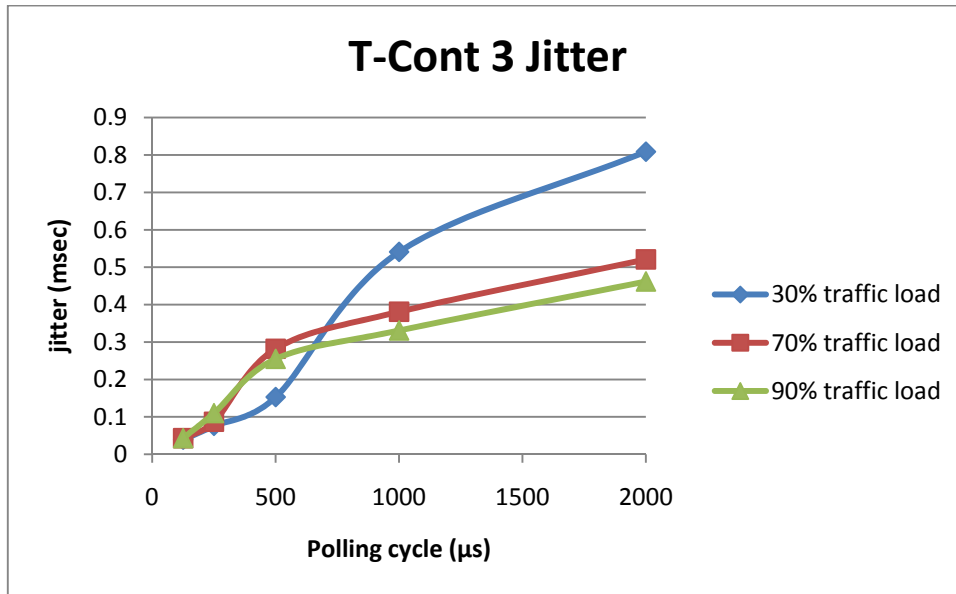


Σχήμα 60: Μέση καθυστέρηση πακέτων κλάσης T-Cont 3

Επιπλέον, τόσο η κλάση T-Cont 2 όσο και η 3 χαρακτηρίζονται από μικρές τιμές jitter, οι οποίες αν και αυξάνονται με την αύξηση του κύκλου δειγματοληψίας, σε καμία περίπτωση δεν υπερβαίνουν το 1 ms. Επομένως, σε συνδυασμό με τις αναμενόμενες τιμές της καθυστέρησης, είναι δυνατό να υποστηρίξουν ευρύ φάσμα απαιτητικών υπηρεσιών και εφαρμογών.

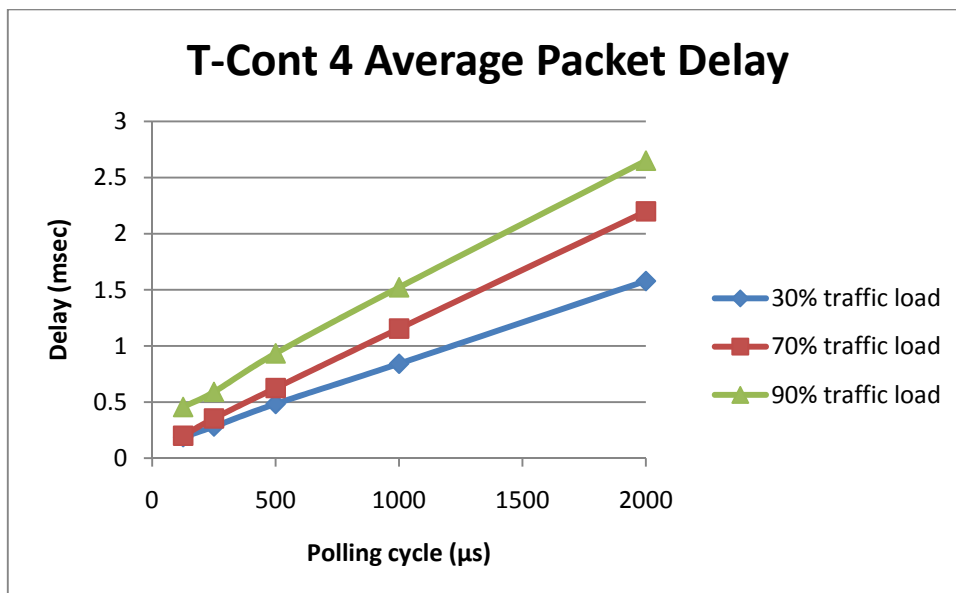


Σχήμα 61: Jitter πακέτων κλάσης T-Cont 2



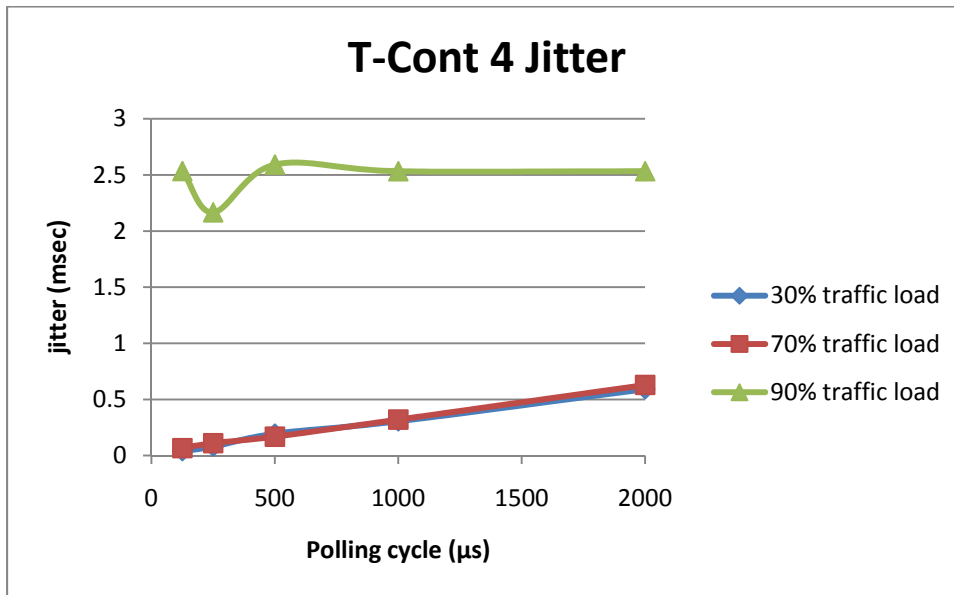
Σχήμα 62: Jitter πακέτων κλάσης T-Cont 3

Οι γραφικές παραστάσεις μέσης καθυστέρησης και jitter των πακέτων της κλάσης T-Cont 4 επιβεβαιώνουν τον προσανατολισμό «καλύτερης προσπάθειας» της συγκεκριμένης κλάσης, με βάση τον οποίο πραγματοποιήθηκε ο σχεδιασμός των περιγραφών κίνησής της. Όπως και με τις κλάσεις 2 & 3, είναι εμφανές ότι η μέση καθυστέρηση αυξάνεται αναλογικά με τον κύκλο δειγματοληψίας, ωστόσο πλέον παρατηρείται ότι αυξημένα επίπεδα φόρτου στο δίκτυο επιφέρουν επιπρόσθετη καθυστέρηση.



Σχήμα 63: Μέση καθυστέρηση πακέτων κλάσης T-Cont 4

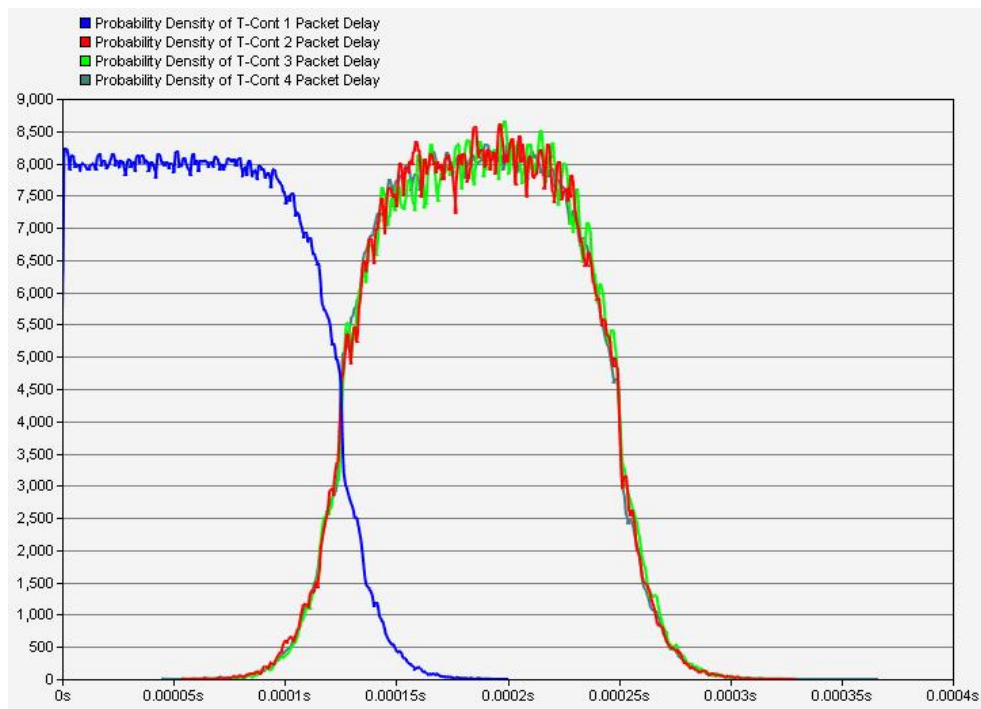
Το επίπεδο φόρτου κίνησης 90% αποτελεί το κρίσιμο σημείο, πέρα από το οποίο τα πακέτα κλάσης T-Cont 4 παράγονται με μεγαλύτερο ρυθμό από το ρυθμό εξυπηρέτησης, με αποτέλεσμα την αυξημένη συμφόρηση στους ενταμιευτές της κλάσης. Υπό συνθήκες φόρτου μεγαλύτερου του 90%, η μέση καθυστέρηση των πακέτων κλάσης T-Cont 4 αυξάνεται διαρκώς. Αντίστοιχη είναι η επίδραση και στο jitter των πακέτων, το οποίο ήδη στο 90% έχει αυξημένη τιμή σε σχέση με τις υπόλοιπες κλάσεις:



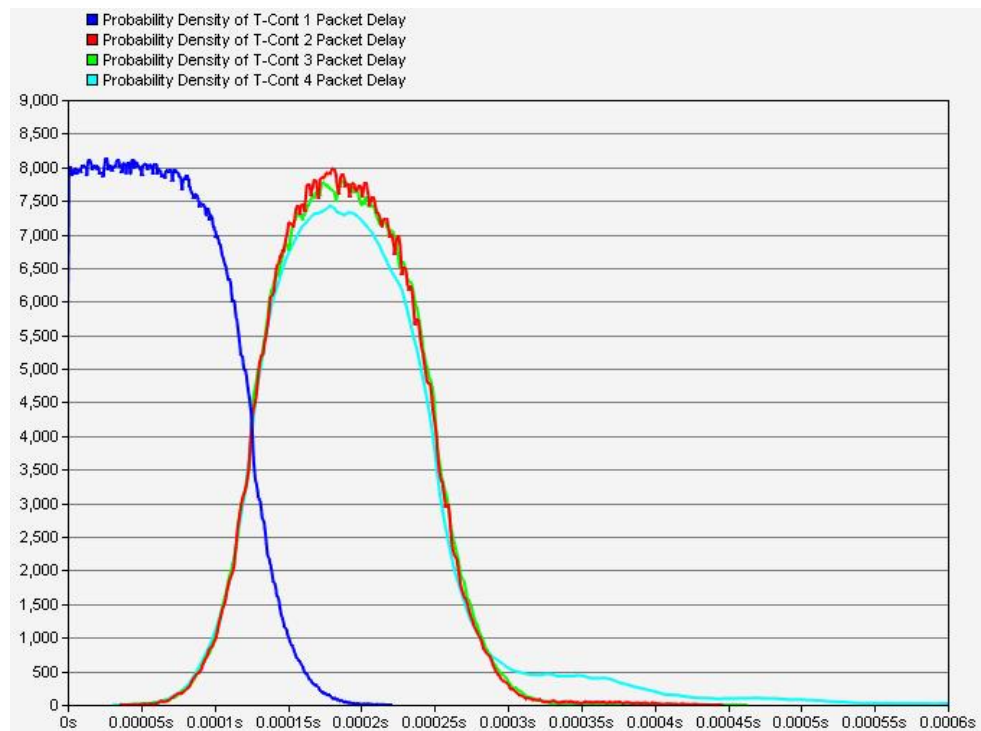
Σχήμα 64: Jitter πακέτων κλάσης T-Cont 4

Στη σειρά από γραφικές παραστάσεις που ακολουθεί, απεικονίζεται η πυκνότητα πιθανότητας της καθυστέρησης για τα πακέτα κάθε κλάσης T-Cont και ο τρόπος με τον οποίο αυτή μεταβάλλεται ανάλογα με το επίπεδο φόρτου κίνησης και το κύκλο δειγματοληψίας. Όπως είναι αναμενόμενο και από τις προηγούμενες γραφικές παραστάσεις, τα επίπεδα φόρτου 30% και 70% δεν εμφανίζουν σημαντικές διαφορές, η καθυστέρηση των πακέτων λαμβάνει χαμηλές τιμές χωρίς σημαντικές διακυμάνσεις.

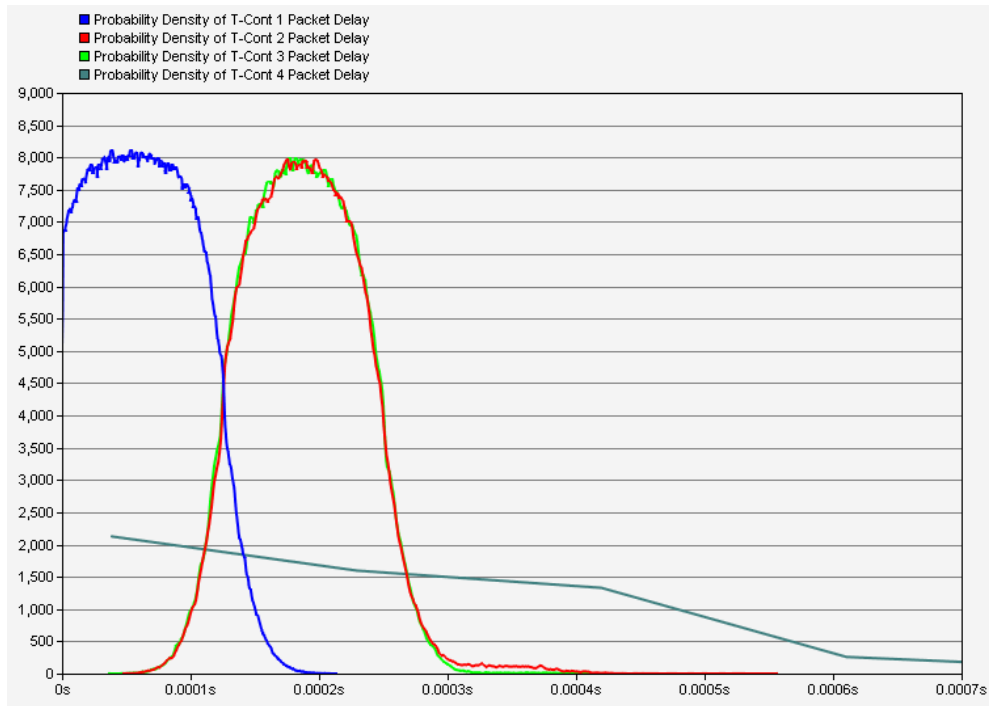
Για επίπεδο φόρτου κίνησης 90%, το οποίο παρουσιάζει και το μεγαλύτερο ενδιαφέρον λόγω του οριακού σημείου συμφόρησης που συνιστά, παρατίθενται οι γραφικές παραστάσεις πυκνότητας πιθανότητας της καθυστέρησης για ορισμένους κύκλους δειγματοληψίας.



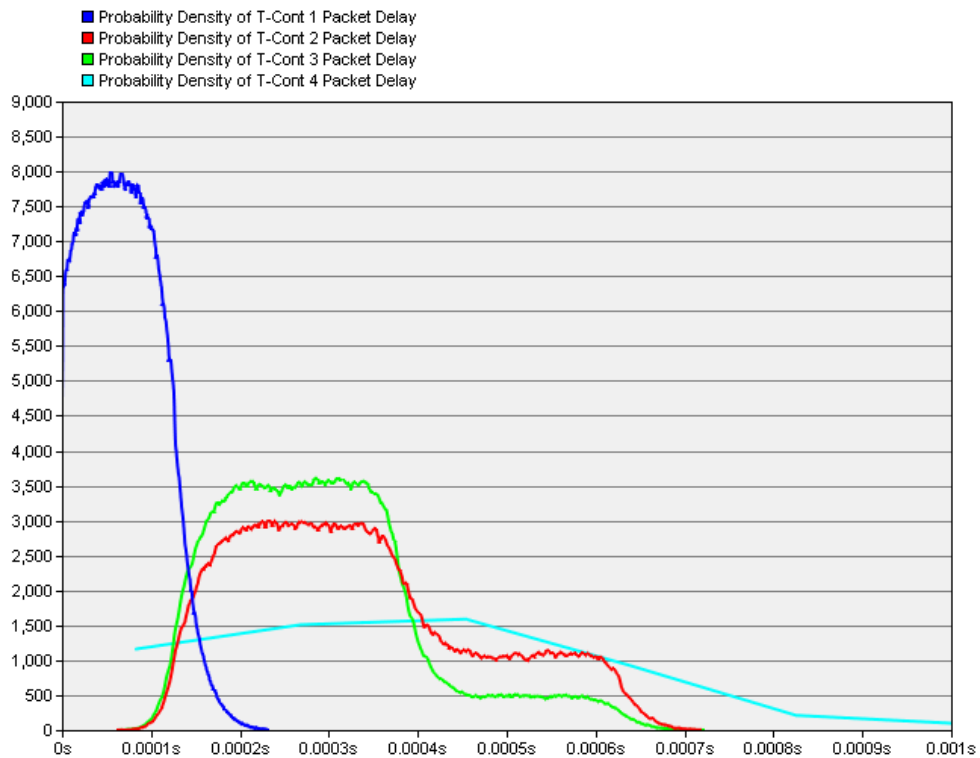
Σχήμα 65: Πυκνότητα πιθανότητας καθυστέρησης, για επίπεδο φόρτου 30% και κύκλο δειγματοληψίας 125 μ s



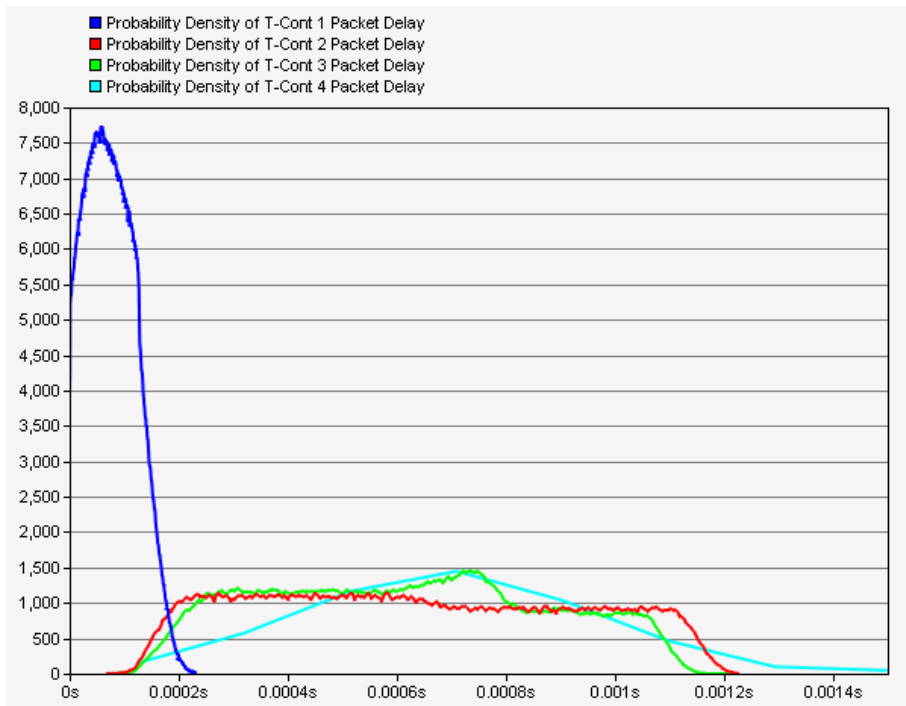
Σχήμα 66: Πυκνότητα πιθανότητας καθυστέρησης, για επίπεδο φόρτου 70% και κύκλο δειγματοληψίας 125 μ s



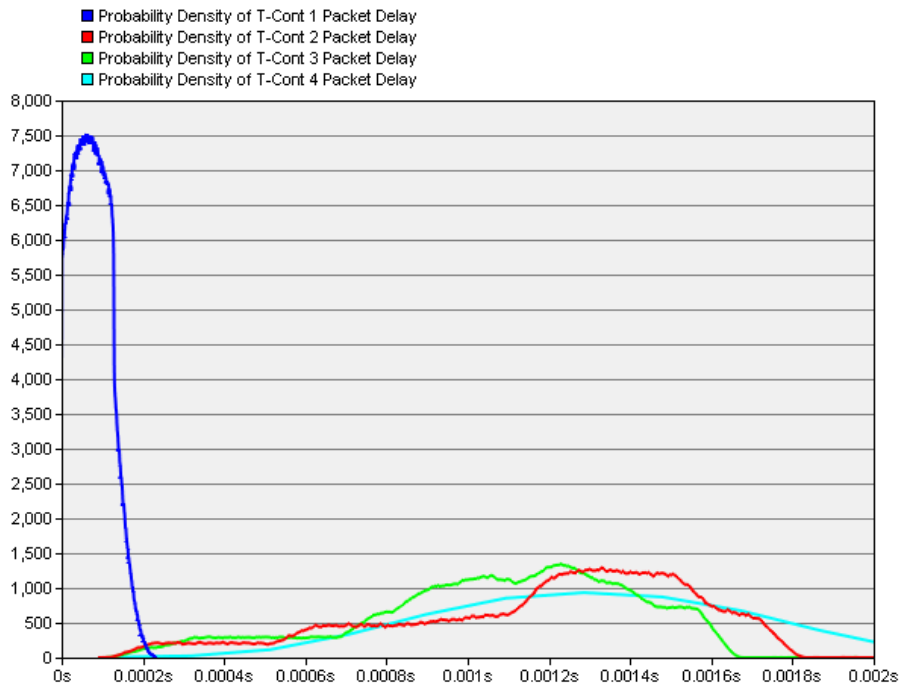
Σχήμα 67: Πυκνότητα πιθανότητας καθυστέρησης, για επίπεδο φόρτου 90% και κύκλο δειγματοληψίας 125 μs



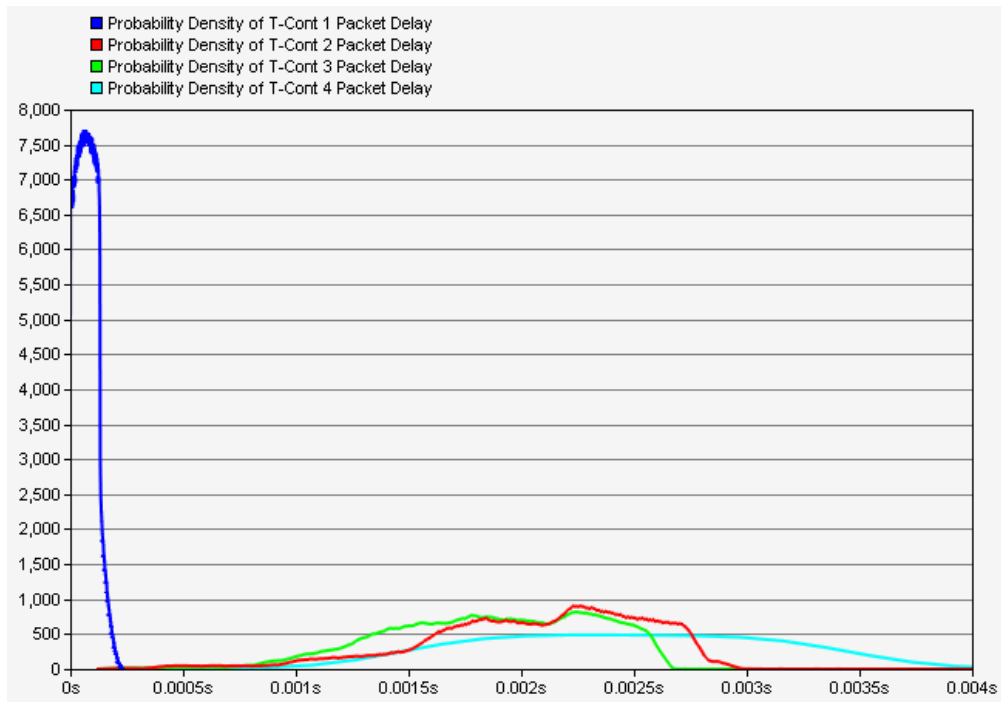
Σχήμα 68: Πυκνότητα πιθανότητας καθυστέρησης, για επίπεδο φόρτου 90% και κύκλο δειγματοληψίας 250 μs



Σχήμα 699: Πυκνότητα πιθανότητας καθυστέρησης, για επίπεδο φόρτου 90% και κύκλο δειγματοληψίας 500 μs



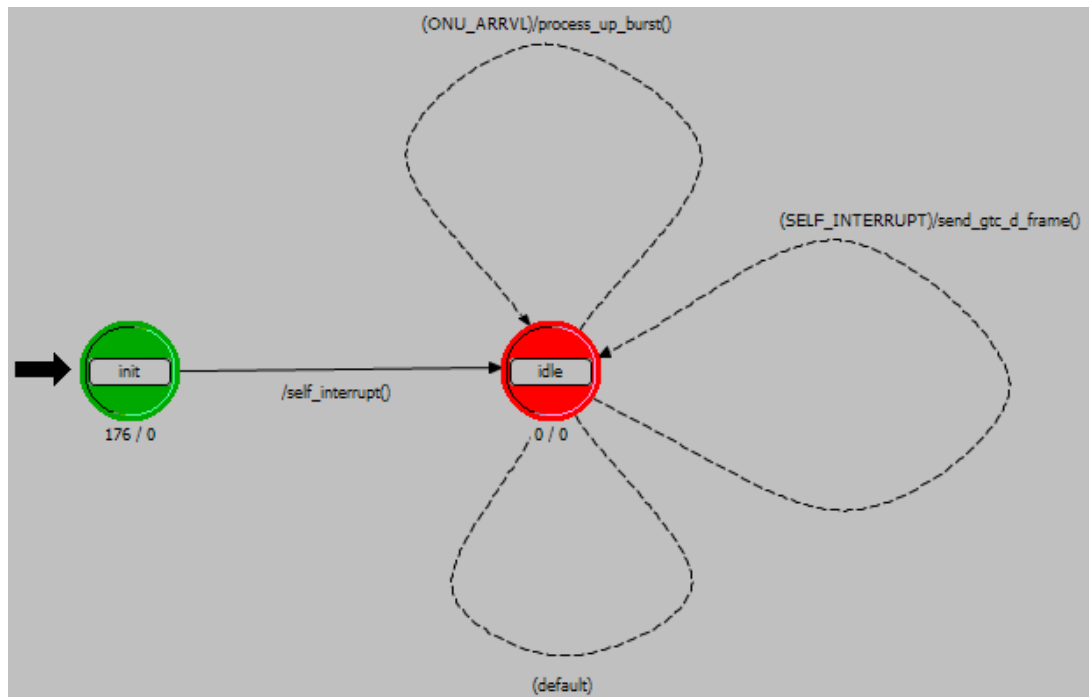
Σχήμα 70: Πυκνότητα πιθανότητας καθυστέρησης, για επίπεδο φόρτου 90% και κύκλο δειγματοληψίας 1000 μs



Σχήμα 71: Πυκνότητα πιθανότητας καθυστέρησης, για επίπεδο φόρτου 90% και κύκλο δειγματοληψίας 2000 μ s

Παράρτημα Α : Διαγράμματα Μετάβασης Καταστάσεων & Κώδικας μοντέλου προσομοίωσης

A.1 Ο επεξεργαστής του τερματικού OLT



A.1.1 Εκτελέσεις εισόδου κατάστασης "init"

```
Objid attr_comp_objid;
Objid attr_objid;
Objid t_cont_comp_objid;
Objid t_cont_objid;
Objid onu_id;
```

```
int i;
int j;
```

```
double guaranteed_bw;
```

```
/* Self-Interrupt frequency value */
self_interrupt_freq = 0.000125;
```

```
/* Obtain Service Period Attribute and initialize related variables */
```

```

op_ima_obj_attr_get (op_topo_parent(op_id_self ()), "Service Period", &service_period);
service_counter = 0;
dbru_rcvd = 0;

/* Obtain DBRu field size Attribute */
op_ima_obj_attr_get (op_topo_parent(op_id_self ()), "DBRu field size", &dbru_fd_size);
dbru_fd_size*=8;

/* Onu Bandwidth initializations */
total_onus= 32;
onu_up_bw = 19440*8;

/* Obtain incoming traffic buffer Objid */
incoming_queue = op_id_from_name (op_topo_parent (op_id_self()), OPC_OBJTYPE_QUEUE,
"incoming_traffic_buffer");

/* Initialize tables with zero values, create reassembly buffers */
for (i=0; i<4; i++) {
    for (j=0; j<2; j++) {
        pk_delay_table[i][j]=0;
        pk_data_table[j]=0;
    }
}

for (i=0; i<128; i++) {
    alloc_table[i]=0;
    bw_grants_table[i]=0;
    rsmbl_buf[i] = op_sar_buf_create (OPC_SAR_BUF_TYPE_REASSEMBLY,
OPC_SAR_BUF_OPT_DEFAULT);
    gem_pl[i]=0;
    seg_cnt[i]=0;
}

/*Obtain Traffic Descriptors compound attribute*/

op_ima_obj_attr_get (op_topo_parent(op_id_self ()), "Traffic Descriptors", &attr_comp_objid);
attr_objid = op_topo_child (attr_comp_objid, OPC_OBJTYPE_GENERIC, 0);

op_ima_obj_attr_get (attr_objid, "T-Cont 1", &t_cont_comp_objid);
t_cont_objid = op_topo_child (t_cont_comp_objid, OPC_OBJTYPE_GENERIC, 0);
op_ima_obj_attr_get (t_cont_objid, "Fixed Bandwidth", &fixed_bw[0]);
op_ima_obj_attr_get (t_cont_objid, "Assured Bandwidth", &assured_bw[0]);
op_ima_obj_attr_get (t_cont_objid, "Maximum Bandwidth", &max_bw[0]);
op_ima_obj_attr_get (t_cont_objid, "Eligibility for additional Bandwidth", &additional_bw[0]);

op_ima_obj_attr_get (attr_objid, "T-Cont 2", &t_cont_comp_objid);
t_cont_objid = op_topo_child (t_cont_comp_objid, OPC_OBJTYPE_GENERIC, 0);
op_ima_obj_attr_get (t_cont_objid, "Fixed Bandwidth", &fixed_bw[1]);
op_ima_obj_attr_get (t_cont_objid, "Assured Bandwidth", &assured_bw[1]);
op_ima_obj_attr_get (t_cont_objid, "Maximum Bandwidth", &max_bw[1]);
op_ima_obj_attr_get (t_cont_objid, "Eligibility for additional Bandwidth", &additional_bw[1]);

```



```

op_ima_obj_attr_get (attr_objid, "T-Cont 3", &t_cont_comp_objid);
t_cont_objid = op_topo_child (t_cont_comp_objid, OPC_OBJTYPE_GENERIC, 0);
op_ima_obj_attr_get (t_cont_objid, "Fixed Bandwidth", &fixed_bw[2]);
op_ima_obj_attr_get (t_cont_objid, "Assured Bandwidth", &assured_bw[2]);
op_ima_obj_attr_get (t_cont_objid, "Maximum Bandwidth", &max_bw[2]);
op_ima_obj_attr_get (t_cont_objid, "Eligibility for additional Bandwidth", &additional_bw[2]);

op_ima_obj_attr_get (attr_objid, "T-Cont 4", &t_cont_comp_objid);
t_cont_objid = op_topo_child (t_cont_comp_objid, OPC_OBJTYPE_GENERIC, 0);
op_ima_obj_attr_get (t_cont_objid, "Fixed Bandwidth", &fixed_bw[3]);
op_ima_obj_attr_get (t_cont_objid, "Assured Bandwidth", &assured_bw[3]);
op_ima_obj_attr_get (t_cont_objid, "Maximum Bandwidth", &max_bw[3]);
op_ima_obj_attr_get (t_cont_objid, "Eligibility for additional Bandwidth", &additional_bw[3]);

/* Perform Checks related to bandwidth components */

if ((fixed_bw[0]+fixed_bw[1]+fixed_bw[2]+fixed_bw[3]+assured_bw[0]+assured_bw[1]+
assured_bw[2]+assured_bw[3]) > 38.88) {
    op_sim_error (OPC_SIM_ERROR_ABORT, "Total Fixed & Assured Bandwidth exceeds ONU's
upstream capacity.", OPC_NIL);
}

for (i=0; i<4; i++) {
    if ((fixed_bw[i]==0) && (assured_bw[i]==0) && (additional_bw[i]==1)) {
        op_sim_error (OPC_SIM_ERROR_ABORT, "A T-Cont cannot be eligible for NA
additional BW, if it has zero guaranteed BW.", OPC_NIL);
    }
}

guaranteed_bw=0;
for (i=0; i<4; i++) {
    if (additional_bw[i]==1) guaranteed_bw+=fixed_bw[i] + assured_bw[i];
}
for (i=0; i<4; i++) {
    if (additional_bw[i]==1) {
        na_bw_mul[i]=fixed_bw[i] + assured_bw[i];
        na_bw_mul[i]/=guaranteed_bw;
    }
    else na_bw_mul[i]=0;
}

guaranteed_bw=0;
for (i=0; i<4; i++) {
    if (additional_bw[i]==2) guaranteed_bw+=max_bw[i]-fixed_bw[i]-assured_bw[i];
}
for (i=0; i<4; i++) {
    if (additional_bw[i]==2) {
        be_bw_mul[i]=max_bw[i] - fixed_bw[i] - assured_bw[i];
        be_bw_mul[i]/=guaranteed_bw;
    }
}

```

```

        else be_bw_mul[i]=0;
    }

for (i=0; i<4; i++) {
    if ((fixed_bw[i]+assured_bw[i])>max_bw[i]) {
        max_bw[i]=fixed_bw[i]+assured_bw[i];
        op_sim_error (OPC_SIM_ERROR_WARNING, "Maximum BW must be >= the sum of
Fixed & Assured BW", "Maximum BW value modified!");
    }
}

/* Obtain Print to file attribute */
op_ima_obj_attr_get (op_topo_parent(op_id_self ()), "Print to File", &attr_comp_objid);
attr_objid = op_topo_child (attr_comp_objid, OPC_OBJTYPE_GENERIC, 0);

op_ima_obj_attr_get (attr_objid, "DBRu", &fprint_dbru);
op_ima_obj_attr_get (attr_objid, "BWMap", &fprint_bwmap);

/* Obtain Record Statistics attribute */
op_ima_obj_attr_get (op_topo_parent(op_id_self ()), "Record Statistics to file", &attr_comp_objid);
attr_objid = op_topo_child (attr_comp_objid, OPC_OBJTYPE_GENERIC, 0);

op_ima_obj_attr_get (attr_objid, "Packet Delay in detail", &recstat_pkdelay);
op_ima_obj_attr_get (attr_objid, "Average Packet Delay / T-Cont & Upstream Efficiency",
&recstat_upeff);

/* Obtain ONU Bw request attribute */
onu_id = op_id_from_name (op_topo_parent(op_topo_parent (op_id_self ())),
OPC_OBJTYPE_NODE_FIX, "onu_1");
onu_id = op_id_from_name (onu_id, OPC_OBJTYPE_PROC, "proc");
op_ima_obj_attr_get (onu_id, "Bandwidth Request", &onu_bw_request);

/* Initilaize the global statistic handles */
tcont1_pk_delay = op_stat_reg ("T-Cont 1 Packet Delay",          OPC_STAT_INDEX_NONE,
OPC_STAT_GLOBAL);
tcont2_pk_delay = op_stat_reg ("T-Cont 2 Packet Delay",          OPC_STAT_INDEX_NONE,
OPC_STAT_GLOBAL);
tcont3_pk_delay = op_stat_reg ("T-Cont 3 Packet Delay",          OPC_STAT_INDEX_NONE,
OPC_STAT_GLOBAL);
tcont4_pk_delay = op_stat_reg ("T-Cont 4 Packet Delay",          OPC_STAT_INDEX_NONE,
OPC_STAT_GLOBAL);
/* Schedule endsim interrupt if required */
if (recstat_upeff==1) {
    op_intrpt_schedule_call (OPC_INTRPT_SCHED_CALL_ENDSIM , 0, endsim_stat_write,
OPC_NIL);
}

```

A.1.2 Συνάρτηση “self interrupt()”

```

/* schedules a self-interrupt at present time */

```

```
static void self_interrupt(void)
{
    op_intrpt_schedule_self (op_sim_time (), 0);
}
```

A.1.3 Συνάρτηση "endsim_stat_write()"

```
/* Actions performed at endsim interrupt */
static void endsim_stat_write (void * nil, int code)
{
    FILE * file;

    FIN (endsim_stat_write);

    file = fopen("sim_results.txt","a+");
    fprintf(file, "Simulation Parameters\n");
    fprintf(file, "=====\n");
    fprintf(file, "DBRu field size= %d bits. Service period= %.6lf microsec.\n", dbru_fd_size,
service_period);
    fprintf(file, "ONU Bandwidth request= %.2lf Mbps, translated to %.2lf%% load.\n",
onu_bw_request, 100*onu_bw_request/38.8);
    fprintf(file, "Traffic Descriptors: Tcont-1 | T-Cont2 | T-Cont 3 | T-Cont 4\n");
    fprintf(file, "Fixed BW (Mbps):   %.2lf   %.2lf   %.2lf   %.2lf\n", fixed_bw[0],
fixed_bw[1], fixed_bw[2], fixed_bw[3]);
    fprintf(file, "Assured BW (Mbps):  %.2lf   %.2lf   %.2lf   %.2lf\n", assured_bw[0],
assured_bw[1], assured_bw[2], assured_bw[3]);
    fprintf(file, "Maximum BW (Mbps):  %.2lf   %.2lf   %.2lf   %.2lf\n", max_bw[0],
max_bw[1], max_bw[2], max_bw[3]);
    fprintf(file, "Additional BW:    %d    %d    %d    %d\n", additional_bw[0],
additional_bw[1], additional_bw[2], additional_bw[3]);
    fprintf(file, "Simulation Results\n");
    fprintf(file, "=====\n");
    fprintf(file, "T-Cont 1 average packet delay = %lf sec.In total %lf packets.\n",
pk_delay_table[0][0]/pk_delay_table[0][1], pk_delay_table[0][1]);
    fprintf(file, "T-Cont 2 average packet delay = %lf sec.In total %lf packets.\n",
pk_delay_table[1][0]/pk_delay_table[1][1], pk_delay_table[1][1]);
    fprintf(file, "T-Cont 3 average packet delay = %lf sec.In total %lf packets.\n",
pk_delay_table[2][0]/pk_delay_table[2][1], pk_delay_table[2][1]);
    fprintf(file, "T-Cont 4 average packet delay = %lf sec.In total %lf packets.\n",
pk_delay_table[3][0]/pk_delay_table[3][1], pk_delay_table[3][1]);

    fprintf(file, "Total Upstream efficiency = %.2lf %%.\n\n",
100*pk_data_table[0]/(pk_data_table[0]+pk_data_table[1]));

    fclose(file);

    FOUT;
}
```

A.1.4 Συνάρτηση “process_up_burst()”

```

/* Processes the incoming Upstream GTC burst*/
static void process_up_burst(void) {

    Packet * up_burst;
    Packet * tcont_pl;
    Packet * gem_frame;
    Packet * gem_payload;
    Packet * comp_pk;

    int tcont_alloc;
    int onu_id;
    int i;
    int j;
    int num_gems;

    double pk_delay;

    FILE * file;

    FIN (process_up_burst);

    /* Obtain the incoming upstream burst and get the sender ONU-id */
    up_burst = op_pk_get (op_intrpt_strm ());
    op_pk_fd_get_int32 (up_burst, 4, &onu_id);

    /* if set, process DBRu and update alloc_table. Update dbru_received var and print if
requested */

    if (op_pk_fd_is_set (up_burst, 6)) {
        dbru_rcvd = 1;

        op_pk_fd_get_int32 (up_burst, 6, &tcont_alloc);
        alloc_table[(onu_id - 1)*4] = tcont_alloc*8 ;

        op_pk_fd_get_int32 (up_burst, 8, &tcont_alloc);
        alloc_table[(onu_id - 1)*4 + 1] = tcont_alloc*8 ;

        op_pk_fd_get_int32 (up_burst, 10, &tcont_alloc);
        alloc_table[(onu_id - 1)*4 + 2] = tcont_alloc*8 ;

        op_pk_fd_get_int32 (up_burst, 12, &tcont_alloc);
        alloc_table[(onu_id - 1)*4 + 3] = tcont_alloc*8 ;
    }
    else dbru_rcvd=0;

    if ((dbru_rcvd==1) && (fprint_dbu==1)) {
        file = fopen("dbru_rcvd.txt","a+");

        fprintf(file, "%lf %d %s %d", op_sim_time(), onu_id, "requests tcont1",
alloc_table[(onu_id - 1)*4]);

```

```

fprintf(file, "%s %d", " tcont2", alloc_table[(onu_id - 1)*4+1]);
fprintf(file, "%s %d", " tcont3", alloc_table[(onu_id - 1)*4+2]);
fprintf(file, "%s %d\n", " tcont4", alloc_table[(onu_id - 1)*4+3]);
fclose(file);
}

pk_data_table[1]+=120;
if (dbru_rcvd==1) pk_data_table[1]+=4*dbru_fd_size;

/*process tcont1 gems*/
if (recstat_pkdelay == 1) file = fopen("tcont1_ete_delay.txt","a+");
if (op_pk_fd_is_set (up_burst, 7) == OPC_TRUE) {
    op_pk_fd_get_pkt (up_burst, 7, &tcont_pl);

    num_gems = op_pk_num_fds (tcont_pl);
    pk_data_table[1]+=40*num_gems;

    for (i=0; i<num_gems; i++) {
        op_pk_fd_get_pkt(tcont_pl, i, &gem_frame);
        op_pk_fd_get_pkt(gem_frame, 4, &gem_payload);
        if (op_sar_pk_is_segment(gem_payload) == OPC_TRUE) {
            seg_cnt[(onu_id - 1)*4]++;
            op_sar_rsmbuf_seg_insert (rsmbf_buf[(onu_id - 1)*4], gem_payload);
            if (op_sar_rsmbuf_pk_count(rsmbf_buf[(onu_id - 1)*4]) >0 ) {
                for (j=0; j<(seg_cnt[(onu_id - 1)*4]-2); j++) {
                    comp_pk = op_sar_rsmbuf_pk_remove (rsmbf_buf[(onu_id - 1)*4]);
                    op_sar_rsmbuf_seg_insert (rsmbf_buf[(onu_id - 1)*4], comp_pk);
                }
                seg_cnt[(onu_id - 1)*4]=0;
                comp_pk = op_sar_rsmbuf_pk_remove (rsmbf_buf[(onu_id - 1)*4]) ;
                pk_delay= (op_sim_time() - op_pk_creation_time_get(comp_pk) ) ;
                pk_delay_table[0][0]+=pk_delay;
                pk_delay_table[0][1]++;
                pk_data_table[0]+=op_pk_total_size_get(comp_pk);
                if (recstat_pkdelay == 1) fprintf(file, "%lf\n", pk_delay);
                op_stat_write (tcont1_pk_delay, pk_delay);
                op_pk_destroy(comp_pk);
            }
        }
        else {
            pk_delay= (op_sim_time() - op_pk_creation_time_get(gem_payload)) ;
            pk_delay_table[0][0]+=pk_delay;
            pk_delay_table[0][1]++;
            pk_data_table[0]+=op_pk_total_size_get(gem_payload);
            if (recstat_pkdelay == 1) fprintf(file, "%lf\n", pk_delay);
                op_stat_write (tcont1_pk_delay,      pk_delay);
                op_pk_destroy (gem_payload);
            }
        }
        op_pk_destroy(gem_frame);
    }
    op_pk_destroy(tcont_pl);
}

```

```

}
if (recstat_pkdelay == 1) fclose(file);

/*process tcont2 gems*/

if (recstat_pkdelay == 1) file = fopen("tcont2_ete_delay.txt","a+");
if (op_pk_fd_is_set (up_burst, 9) == OPC_TRUE) {
    op_pk_fd_get_pkt (up_burst, 9, &tcont_pl);

    num_gems = op_pk_num_fds (tcont_pl);
    pk_data_table[1]+=40*num_gems;

    for (i=0; i<num_gems; i++) {
        op_pk_fd_get_pkt(tcont_pl, i, &gem_frame);
        op_pk_fd_get_pkt(gem_frame, 4, &gem_payload);
        if (op_sar_pk_is_segment(gem_payload) == OPC_TRUE) {
            seg_cnt[(onu_id - 1)*4+1]++;
            op_sar_rsmbuf_seg_insert (rsmb_buf[(onu_id - 1)*4+1], gem_payload);
            if (op_sar_rsmbuf_pk_count(rsmb_buf[(onu_id - 1)*4+1]) > 0 ) {
                for (j=0; j<(seg_cnt[(onu_id - 1)*4+1]-2); j++) {
                    comp_pk = op_sar_rsmbuf_pk_remove (rsmb_buf[(onu_id - 1)*4+1]) ;
                    op_sar_rsmbuf_seg_insert (rsmb_buf[(onu_id - 1)*4+1], comp_pk);
                }
                seg_cnt[(onu_id - 1)*4+1]=0;
                comp_pk = op_sar_rsmbuf_pk_remove (rsmb_buf[(onu_id - 1)*4+1]) ;
                pk_delay=( op_sim_time() - op_pk_creation_time_get(comp_pk) ) ;
                pk_delay_table[1][0]+=pk_delay;
                pk_delay_table[1][1]++;
            }
        }
        pk_data_table[0]+=op_pk_total_size_get(comp_pk);
        if (recstat_pkdelay == 1) fprintf(file, "%lf\n", pk_delay);
        op_stat_write (tcont2_pk_delay,bpk_delay);
        op_pk_destroy(comp_pk);
    }
}
else {
    pk_delay=( op_sim_time() - op_pk_creation_time_get(gem_payload) ) ;
    pk_delay_table[1][0]+=pk_delay;
    pk_delay_table[1][1]++;
    pk_data_table[0]+=op_pk_total_size_get(gem_payload);
    if (recstat_pkdelay == 1) fprintf(file, "%lf\n", pk_delay);
    op_stat_write (tcont2_pk_delay,          pk_delay);
    op_pk_destroy (gem_payload);
}
op_pk_destroy(gem_frame);
}
op_pk_destroy(tcont_pl);
}
if (recstat_pkdelay == 1) fclose(file);

/*process tcont3 gems*/

```

```

if (recstat_pkdelay == 1) file = fopen("tcont3_ete_delay.txt","a+");
if (op_pk_fd_is_set (up_burst, 11) == OPC_TRUE) {
    op_pk_fd_get_pkt (up_burst, 11, &tcont_pl);

    num_gems = op_pk_num_fds (tcont_pl);
    pk_data_table[1]+=40*num_gems;

    for (i=0; i<num_gems; i++) {
        op_pk_fd_get_pkt(tcont_pl, i, &gem_frame);
        op_pk_fd_get_pkt(gem_frame, 4, &gem_payload);

        if (op_sar_pk_is_segment(gem_payload) == OPC_TRUE) {

            seg_cnt[(onu_id - 1)*4+2]++;
            op_sar_rsmbuf_seg_insert (rsmbf_buf[(onu_id - 1)*4+2], gem_payload);
            if (op_sar_rsmbuf_pk_count(rsmbf_buf[(onu_id - 1)*4+2]) > 0 ) {
                for (j=0; j<(seg_cnt[(onu_id - 1)*4+2]-2); j++) {
                    comp_pk = op_sar_rsmbuf_pk_remove (rsmbf_buf[(onu_id - 1)*4+2]);
                    op_sar_rsmbuf_seg_insert (rsmbf_buf[(onu_id - 1)*4+2], comp_pk);
                }

                seg_cnt[(onu_id - 1)*4+2]=0;
                comp_pk = op_sar_rsmbuf_pk_remove (rsmbf_buf[(onu_id - 1)*4+2]);
                pk_delay=( op_sim_time() - op_pk_creation_time_get(comp_pk) );
                pk_delay_table[2][0]+=pk_delay;
                pk_delay_table[2][1]++;
                pk_data_table[0]+=op_pk_total_size_get(comp_pk);
                if (recstat_pkdelay == 1) fprintf(file, "%lf\n", pk_delay);
                op_stat_write (tcont3_pk_delay,      pk_delay);
                op_pk_destroy(comp_pk);
            }
        }
        else {
            pk_delay=( op_sim_time() - op_pk_creation_time_get(gem_payload));
            pk_delay_table[2][0]+=pk_delay;
            pk_delay_table[2][1]++;
            pk_data_table[0]+=op_pk_total_size_get(gem_payload);
            if (recstat_pkdelay == 1) fprintf(file, "%lf\n", pk_delay);
            op_stat_write (tcont3_pk_delay,      pk_delay);
            op_pk_destroy (gem_payload);
        }
        op_pk_destroy(gem_frame);
    }
    op_pk_destroy(tcont_pl);
}
if (recstat_pkdelay == 1) fclose(file);

/*process tcont4 gems*/

if (recstat_pkdelay == 1) file = fopen("tcont4_ete_delay.txt","a+");

```

```

if (op_pk_fd_is_set (up_burst, 13) == OPC_TRUE) {
    op_pk_fd_get_pkt (up_burst, 13, &tcont_pl);

    num_gems = op_pk_num_fds (tcont_pl);
    pk_data_table[1]+=40*num_gems;

    for (i=0; i<num_gems; i++) {
        op_pk_fd_get_pkt(tcont_pl, i, &gem_frame);
        op_pk_fd_get_pkt(gem_frame, 4, &gem_payload);

        if (op_sar_pk_is_segment(gem_payload) == OPC_TRUE) {

            seg_cnt[(onu_id - 1)*4+3]++;
            op_sar_rsmbuf_seg_insert (rsmbuf_buf[(onu_id - 1)*4+3], gem_payload);
            if (op_sar_rsmbuf_pk_count(rsmbuf_buf[(onu_id - 1)*4+3]) > 0) {
                for (j=0; j<(seg_cnt[(onu_id - 1)*4+3]-2); j++) {
                    comp_pk = op_sar_rsmbuf_pk_remove (rsmbuf_buf[(onu_id - 1)*4+3]) ;
                    op_sar_rsmbuf_seg_insert (rsmbuf_buf[(onu_id - 1)*4+3], comp_pk);
                }

                seg_cnt[(onu_id - 1)*4+3]=0;
                comp_pk = op_sar_rsmbuf_pk_remove (rsmbuf_buf[(onu_id - 1)*4+3]) ;
                pk_delay= (op_sim_time() - op_pk_creation_time_get(comp_pk) ) ;
                pk_delay_table[3][0]+=pk_delay;
                pk_delay_table[3][1]++;
                pk_data_table[0]+=op_pk_total_size_get(comp_pk);
                if (recstat_pkdelay == 1) fprintf(file, "%lf\n", pk_delay);
                    op_stat_write (tcont4_pk_delay,      pk_delay);
                    op_pk_destroy(comp_pk);
                }
            }
        }
        else {
            pk_delay= (op_sim_time() -
op_pk_creation_time_get(gem_payload)) ;
            pk_delay_table[3][0]+=pk_delay;
            pk_delay_table[3][1]++;
            pk_data_table[0]+=op_pk_total_size_get(gem_payload);
            if (recstat_pkdelay == 1) fprintf(file, "%lf\n", pk_delay);
                op_stat_write (tcont4_pk_delay,      pk_delay);
                op_pk_destroy (gem_payload);
            }
        }
        op_pk_destroy(gem_frame);
    }
    op_pk_destroy(tcont_pl);
}
if (recstat_pkdelay == 1) fclose(file);

op_pk_destroy (up_burst);

FOUT;
}

```


A.1.5 Συνάρτηση “send_gtc_d_frame()”

```
/* Create GTC downstream frame and broadcast it to all ONUs*/
```

```
static void send_gtc_d_frame(void)
{
    Packet * gtc_d;
    Packet * cp_pkptr;
    Packet * pcbd;
    Packet * ds_payload;
    Packet * gem_frame;
    Packet * ploamd;
    Packet * bwmap;
    Packet * bwmap_alloc;
    Packet * next_data_pk;

    int * head_pk_size_ptr;
    int dbru_req;
    int n_alloc;
    int i;
    int pcbd_size;
    int payload_space_left;
    int ds_payload_exists;
    int stream_empty;
    int up_bw_left;
    int start_bit_pointer;
    int n;
    int data_pk_size;
    int dest_onu;

    double temp;

    FILE * file;

    FIN (send_gtc_d_frame());

    /*Decide whether ONUs will be requested to send DBRu, according to service period
*/

    if (service_counter==service_period/0.000125) service_counter=0;
    if (service_counter==0) {
        service_counter++;
        dbru_req=1;
    }
    else {
        service_counter++;
        dbru_req=0;
    }

    /*calculate bw grants, dba algorithm:4 tconts for every onu in sequence*/
```

```

/*tcont1 receives 120 extra for guard+plou, each tcont 8/16 extra bits for dbu in
service period*/

```

```

up_bw_left = onu_up_bw - 32*120;
if (dbu_req==1) up_bw_left-=128*dbu_fd_size;
for(i=0; i<32; i++) {

```

```

/*Assign Fixed & Assured bandwidth*/

```

```

bw_grants_table[4*i]=min((fixed_bw[0]*125+assured_bw[0]*125),(max(fixed_bw[0]
*125,alloc_table[4*i])));
alloc_table[4*i]=bw_grants_table[4*i];
if (alloc_table[4*i]<0) alloc_table[4*i]=0;
up_bw_left -= bw_grants_table[4*i];

```

```

bw_grants_table[4*i+1]=min((fixed_bw[1]*125+assured_bw[1]*125),(max(fixed_bw[
1]*125,alloc_table[4*i+1])));
alloc_table[4*i+1]=bw_grants_table[4*i+1];
if (alloc_table[4*i+1]<0) alloc_table[4*i+1]=0;
up_bw_left -= bw_grants_table[4*i+1];

```

```

bw_grants_table[4*i+2]=min((fixed_bw[2]*125+assured_bw[2]*125),(max(fixed_bw[
2]*125,alloc_table[4*i+2])));
alloc_table[4*i+2]=bw_grants_table[4*i+2];
if (alloc_table[4*i+2]<0) alloc_table[4*i+2]=0;
up_bw_left -= bw_grants_table[4*i+2];

```

```

bw_grants_table[4*i+3]=min((fixed_bw[3]*125+assured_bw[3]*125),(max(fixed_bw[
3]*125,alloc_table[4*i+3])));
alloc_table[4*i+3]=bw_grants_table[4*i+3];
if (alloc_table[4*i+3]<0) alloc_table[4*i+3]=0;
up_bw_left -= bw_grants_table[4*i+3];

```

```

bw_grants_table[4*i] += 120;
if (dbu_req==1) {
    bw_grants_table[4*i] += dbu_fd_size;
    bw_grants_table[4*i+1] += dbu_fd_size;
    bw_grants_table[4*i+2] += dbu_fd_size;
    bw_grants_table[4*i+3] += dbu_fd_size;
}
}

```

```

/*Non Assured BW Assignment*/

```

```

n=0;
/*file = fopen("test.txt","a+"); */
for(i=0; i<128; i++) {
    if ((alloc_table[i]>0) && (additional_bw[i%4]==1.0)) {
        n++;
    }
}
/*if (n>0) fprintf(file, "%lf best effort to n einai arxika %d\n", op_sim_time(), n); */

```

```

while((n>0) && (up_bw_left>0)) {
    for (i=0; i<128; i++) {
        if ((alloc_table[i]>0) && (additional_bw[i%4]==1.0)) {
            temp =
min(floor(na_bw_mul[i%4]*(fixed_bw[i%4]*125+assured_bw[i%4]*125)),alloc_table[i]);
            temp = min((min(temp,up_bw_left)),max_bw[i%4]*125-bw_grants_table[i]);
            bw_grants_table[i]+=temp;
            alloc_table[i]-=temp;
            /*fprintf(file, "dinw %lf, twra sto pinaka alloc exei %d\n", temp,
alloc_table[i]);*/
            if ((alloc_table[i]==0) || (bw_grants_table[i]==max_bw[i%4]*125)) {
                n--;
                /*fprintf(file, "%lf to n meiwnetai se %d\n", op_sim_time(), n);*/
            }
            up_bw_left -= temp;
        }
    }

    /*fprintf(file, "%lf telos kyklou. up_bw_left= %d\n\n", op_sim_time(), up_bw_left);*/
}
/*fclose(file); */

/*Best Effort BW Assignment*/
n=0;
/*file = fopen("test.txt","a+"); */
for(i=0; i<128; i++) {
    if ((alloc_table[i]>0) && (additional_bw[i%4]==2.0)) {
        n++;
    }
}
/*if (n>0) fprintf(file, "%lf best effort to n einai arxika %d\n", op_sim_time(), n); */
while((n>0) && (up_bw_left>0)) {
    for (i=0; i<128; i++) {
        if ((alloc_table[i]>0) && (additional_bw[i%4]==2.0)) {
            temp = min(floor(be_bw_mul[i%4]*(max_bw[i%4]*125-fixed_bw[i%4]*125-
assured_bw[i%4]*125)),alloc_table[i]);
            temp = min((min(temp,up_bw_left)),max_bw[i%4]*125-bw_grants_table[i]);
            bw_grants_table[i]+=temp;
            alloc_table[i]-=temp;
            /*fprintf(file, "dinw %lf, twra sto pinaka alloc exei %d\n", temp,
alloc_table[i]);*/
            if ((alloc_table[i]==0) || (bw_grants_table[i]==max_bw[i%4]*125)) {
                n--;
                /*fprintf(file, "%lf to n meiwnetai se %d\n", op_sim_time(), n);*/
            }
            up_bw_left -= temp;
        }
    }

    /*fprintf(file, "%lf telos kyklou. up_bw_left= %d\n\n", op_sim_time(), up_bw_left); */
}

```

```

}

/*fclose(file);*/

/* add 40 bits due to an additional gem header required */
for (i=0; i<128; i++) {
    if ((dbru_rcvd==0) && (alloc_table[i]>0)) alloc_table[i]+=40;
}

/* Print bwmap to file if requested */
if (fprint_bwmap==1) {
    for(i=0; i<1; i++) {
        file = fopen("bw_grants.txt","a+");
        fprintf(file, "%lf %d %s %d", op_sim_time(), i+1, " tcont1 granted ",
bw_grants_table[4*i]);
        fprintf(file, "%s %d", " tcont2 granted ", bw_grants_table[4*i+1]);
        fprintf(file, "%s %d", " tcont3 granted ", bw_grants_table[4*i+2]);
        fprintf(file, "%s %d\n\n", " tcont4 granted ", bw_grants_table[4*i+3]);
        fclose(file);
    }
}

/* create bwmap */

/*file = fopen("1.bwmap.txt","a+"); */

n_alloc=128;
bwmap = op_pk_create (0);
start_bit_pointer = 0;

for(i=0; i < n_alloc; i++) {
    bwmap_alloc = op_pk_create (0);
    op_pk_fd_set_int32 (bwmap_alloc, 0, i, 12);
    if (dbru_req==1) op_pk_fd_set_int32 (bwmap_alloc, 1, 1, 12);
    else op_pk_fd_set_int32 (bwmap_alloc, 1, 0, 12);

    op_pk_fd_set_int32 (bwmap_alloc, 2, start_bit_pointer, 16);
    op_pk_fd_set_int32 (bwmap_alloc, 3, start_bit_pointer + bw_grants_table[i]
- 1, 16);

    /*fprintf(file, "%lf %s %d %s %d %s %d %d\n", op_sim_time(), "onu", i/4+1, "
tcont", i%4+1, "bwmap", start_bit_pointer, start_bit_pointer + bw_grants_table[i] - 1); */
    op_pk_fd_set_int32 (bwmap_alloc, 4, 0, 8);
    op_pk_fd_set_pkt (bwmap, i, bwmap_alloc, -1);
    start_bit_pointer = start_bit_pointer + bw_grants_table[i];
}

/*fclose(file);*/

/* Create GTCd frame including all fields */
gtc_d = op_pk_create (0);

```

```

pcbd = op_pk_create (0);

/* Psync field */
op_pk_fd_set_int32 (pcbd, 0, 3064672736, 32);

/* Ident field */
op_pk_fd_set_int32 (pcbd, 1, 0, 32);

/* Ploamd field */
ploamd = op_pk_create (104);
op_pk_fd_set_pkt (pcbd, 2, ploamd, -1);

/* BIP field */
op_pk_fd_set_int32 (pcbd, 3, 0, 8);

/* Plend field */
op_pk_fd_set_int32 (pcbd, 4, n_alloc, 32);

/* Plend field */
op_pk_fd_set_int32 (pcbd, 5, n_alloc, 32);

/* BWmap field */
op_pk_fd_set_pkt (pcbd, 6, bwmap, -1);

pcbd_size = op_pk_total_size_get(pcbd);

/* Encapsulate pcbd to gtc_d frame */
op_pk_fd_set_pkt (gtc_d, 0, pcbd, -1);

/* get data and fill payload, then encapsulate */
stream_empty = 0;
payload_space_left = (311040 - pcbd_size) ;
head_pk_size_ptr = (int *) op_ima_obj_svar_get (incoming_queue, "head_pk_size");

if ( (payload_space_left>0) && ((*head_pk_size_ptr)!=0) ){
    ds_payload = op_pk_create(0);
    ds_payload_exists = 1;
    op_intrpt_force_remote (payload_space_left, incoming_queue);
}

i = 0;

while (stream_empty == 0) {
    if (op_strm_empty (0) == OPC_FALSE ) {
        next_data_pk = op_pk_get (0);

        op_pk_nfd_get_int32 (next_data_pk, "onu", &dest_onu);
        data_pk_size = op_pk_total_size_get(next_data_pk);
        gem_frame = op_pk_create(0);
        op_pk_fd_set_int32 (gem_frame, 0, data_pk_size, 12);
        op_pk_fd_set_int32 (gem_frame, 1, dest_onu, 12);
        op_pk_fd_set_int32 (gem_frame, 2, 0, 3);
    }
}

```

```

        op_pk_fd_set_int32 (gem_frame, 3, 0, 13);
        op_pk_fd_set_pkt (gem_frame, 4, next_data_pk, -1);
        op_pk_fd_set_pkt (ds_payload, i, gem_frame, -1);
        i++;
    }
    else stream_empty = 1;
}

if (ds_payload_exists == 1) op_pk_fd_set_pkt (gtc_d, 1, ds_payload, -1);
op_pk_total_size_set (gtc_d, 311040);

/*broadcast gtc_d frame to all 32 ONUs*/
for (i = 1; i < 33; i++) {
    cp_pkptr = op_pk_copy (gtc_d);
    op_pk_send (cp_pkptr, i);
}

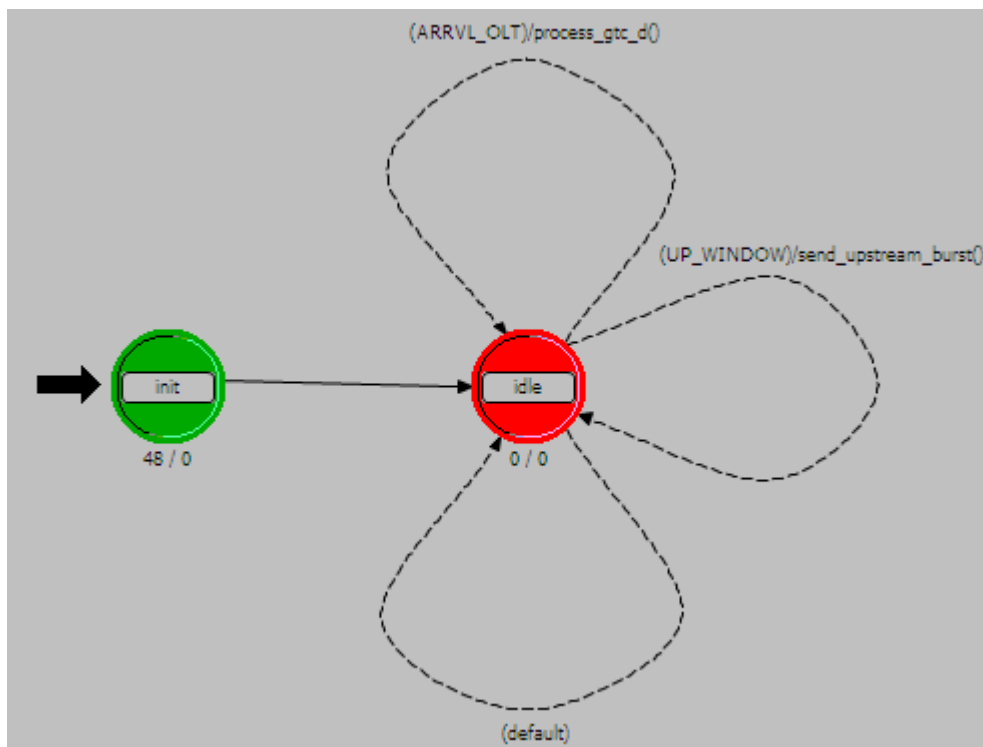
op_pk_destroy(gtc_d);

/*schedule next interrupt*/
op_intrpt_schedule_self (op_sim_time () + self_interrupt_freq, 0);

FOUT;
}

```

A.2 Ο επεξεργαστής της μονάδας ONU



A.2.1 Εκτελέσεις εισόδου κατάστασης “init”

```

Objid recstat_comp_objid;
Objid recstat_objid;
Objid src_1;
Objid src_2;
Objid src_3;
Objid src_4;

/* Obtain the object id of the ONU node, proc / queues / sources, get onu_id attribute*/

own_id = op_id_self ();
parent_id = op_topo_parent (own_id);
op_ima_obj_attr_get (parent_id, "onu_id", &onu_id);

incoming_queue_1 = op_id_from_name (parent_id, OPC_OBJTYPE_QUEUE, "tcont1_q");
incoming_queue_2 = op_id_from_name (parent_id, OPC_OBJTYPE_QUEUE, "tcont2_q");
incoming_queue_3 = op_id_from_name (parent_id, OPC_OBJTYPE_QUEUE, "tcont3_q");
incoming_queue_4 = op_id_from_name (parent_id, OPC_OBJTYPE_QUEUE, "tcont4_q");
src_1 = op_id_from_name (parent_id, OPC_OBJTYPE_PROC, "tcont1_src");
src_2 = op_id_from_name (parent_id, OPC_OBJTYPE_PROC, "tcont2_src");
src_3 = op_id_from_name (parent_id, OPC_OBJTYPE_PROC, "tcont3_src");
src_4 = op_id_from_name (parent_id, OPC_OBJTYPE_PROC, "tcont4_src");

/*Interrupt frequency value*/
interrupt_freq = 0.000125;

/* Obtain the OLT's object id and get statistic recording related attributes*/
olt_id = op_id_from_name (op_topo_parent(parent_id), OPC_OBJTYPE_NODE_FIX, "gpon_olt");
op_ima_obj_attr_get (olt_id, "Record Statistics to file", &recstat_comp_objid);
recstat_objid = op_topo_child (recstat_comp_objid, OPC_OBJTYPE_GENERIC, 0);

/*Obtain DBRu field size Attribute from OLT*/
op_ima_obj_attr_get (olt_id, "DBRu field size", &dbru_fd_size);
dbru_fd_size*=8;

```

A.2.1 Συνάρτηση “process_gtc_d()”

```

static void process_gtc_d(void) {

    Packet * gtc_d;
    Packet * pcbd;
    Packet * bwmap;
    Packet * bwmap_alloc;

    int i;
    int tcont;

```

```

double interrupt_time;

FIN (process_gtc_d());

/*Obtain incoming GTCd frame*/
gtc_d = op_pk_get (0);

/* Extract and Process BWmap */

op_pk_fd_get_pkt (gtc_d, 0, &pcbd);
op_pk_fd_get_pkt (pcbd, 6, &bwmap);
for (i=0; i<128; i++) {
    op_pk_fd_get_pkt (bwmap, i, &bwmap_alloc);
    op_pk_fd_get_int32 (bwmap_alloc, 0, &tcont);
    op_pk_fd_get_int32 (bwmap_alloc, 1, &dbru_req);
    if (tcont == 4*(onu_id-1)) {
        op_pk_fd_get_int32(bwmap_alloc, 2, &tcont_grants[0][0]);
        op_pk_fd_get_int32(bwmap_alloc, 3, &tcont_grants[0][1]);
    }
    else if (tcont == 4*(onu_id-1)+1) {
        op_pk_fd_get_int32(bwmap_alloc, 2, &tcont_grants[1][0]);
        op_pk_fd_get_int32(bwmap_alloc, 3, &tcont_grants[1][1]);
    }
    else if (tcont == (4*(onu_id-1)+2)) {
        op_pk_fd_get_int32(bwmap_alloc, 2, &tcont_grants[2][0]);
        op_pk_fd_get_int32(bwmap_alloc, 3, &tcont_grants[2][1]);
    }
    else if (tcont == (4*(onu_id-1)+3)) {
        op_pk_fd_get_int32(bwmap_alloc, 2, &tcont_grants[3][0]);
        op_pk_fd_get_int32(bwmap_alloc, 3, &tcont_grants[3][1]);
    }
    op_pk_destroy (bwmap_alloc);
}

op_pk_destroy (bwmap);
op_pk_destroy (pcbd);
op_pk_destroy (gtc_d);

/*schedule self-interrupt for upstream transmission*/
interrupt_time = tcont_grants[0][0];
op_intrpt_schedule_self (op_sim_time () + (interrupt_time/155520)*interrupt_freq,
0);

FOUT;
}

```

A.2.3 Συνάρτηση “send upstream burst()”


```

static void send_upstream_burst(void) {

    Packet * up_burst;
    Packet * up_alloc_tcont;
    Packet * gem_frame;
    Packet * next_data_pk;

    int * head_pk_size_ptr;
    int * ttl_queue_data_ptr;
    int * ttl_queue_pkts_ptr;
    int i;
    int stream_empty;
    int grant;
    int up_alloc_tcont_exists;
    int data_pk_size;
    int tcont_alloc_space;

    FIN (send_upstream_burst());

    /*create Upstream frame and set guard, PLOu, etc fields*/
    up_burst = op_pk_create (0);

    /* guard time field */
    op_pk_fd_set_int32 (up_burst, 0, 0, 32);

    /* preamble field */
    op_pk_fd_set_int32 (up_burst, 1, 0, 44);

    /* delimiter field */
    op_pk_fd_set_int32 (up_burst, 2, 0, 20);

    /* BIP field */
    op_pk_fd_set_int32 (up_burst, 3, 0, 8);

    /* ONU-ID field */
    op_pk_fd_set_int32 (up_burst, 4, onu_id, 8);

    /* Ind field */
    op_pk_fd_set_int32 (up_burst, 5, 0, 8);

    /* DBRu & T-Cont 1 allocation */

        /* T-Cont 1 payload set */

    up_alloc_tcont_exists = 0;
    stream_empty = 0;
    tcont_alloc_space = (tcont_grants[0][1] - tcont_grants[0][0] + 1 - 120) ;
    if (dbru_req==1) tcont_alloc_space-=dbru_fd_size;
    head_pk_size_ptr = (int *) op_ima_obj_svar_get (incoming_queue_1,
"head_pk_size");
    if ((tcont_alloc_space > 0) && (*head_pk_size_ptr != 0) ) {

```

```

        up_alloc_tcont = op_pk_create(0);
        up_alloc_tcont_exists = 1;
        op_intrpt_force_remote (tcont_alloc_space, incoming_queue_1);
    }
    i = 0;

    while (stream_empty == 0) {
        if (op_strm_empty (1) == OPC_FALSE ) {
            next_data_pk = op_pk_get (1);
            data_pk_size = op_pk_total_size_get(next_data_pk);
            gem_frame = op_pk_create(0);
            op_pk_fd_set_int32 (gem_frame, 0, data_pk_size, 12);
            op_pk_fd_set_int32 (gem_frame, 1, 4*(onu_id-1), 12);
            op_pk_fd_set_int32 (gem_frame, 2, 0, 3);
            op_pk_fd_set_int32 (gem_frame, 3, 0, 13);
            op_pk_fd_set_pkt (gem_frame, 4, next_data_pk, -1);
            op_pk_fd_set_pkt (up_alloc_tcont, i, gem_frame, -1);
            i++;
        }
        else stream_empty = 1;
    }

    /* DBRu field set, if requested */
    if (dbru_req==1) {
        ttl_queue_data_ptr = (int *) op_ima_obj_svar_get (incoming_queue_1,
"ttl_queue_data");
        ttl_queue_pkts_ptr = (int *) op_ima_obj_svar_get (incoming_queue_1,
"ttl_queue_pkts");
        grant = ceil(((double)(*ttl_queue_data_ptr))/8.0)+5*(*ttl_queue_pkts_ptr);
        op_pk_fd_set_int32 (up_burst, 6, grant, dbru_fd_size);
    }

    if (up_alloc_tcont_exists == 1) op_pk_fd_set_pkt (up_burst, 7, up_alloc_tcont, -1);

    /* DBRu & T-Cont 2 allocation */

    /* T-Cont 2 payload set */

    up_alloc_tcont_exists = 0;
    stream_empty = 0;
    tcont_alloc_space = (tcont_grants[1][1] - tcont_grants[1][0] + 1) ;
    if (dbru_req==1) tcont_alloc_space-=dbru_fd_size;
    head_pk_size_ptr = (int *) op_ima_obj_svar_get (incoming_queue_2,
"head_pk_size");
    if ((tcont_alloc_space > 0) && (*head_pk_size_ptr != 0) ) {
        up_alloc_tcont = op_pk_create(0);
        up_alloc_tcont_exists = 1;
        op_intrpt_force_remote (tcont_alloc_space, incoming_queue_2);
    }

    i = 0;

```

```

while (stream_empty == 0) {
    if (op_strm_empty (2) == OPC_FALSE ) {
        next_data_pk = op_pk_get (2);
        data_pk_size = op_pk_total_size_get(next_data_pk);
        gem_frame = op_pk_create(0);
        op_pk_fd_set_int32 (gem_frame, 0, data_pk_size, 12);
        op_pk_fd_set_int32 (gem_frame, 1, 4*(onu_id-1)+1, 12);
        op_pk_fd_set_int32 (gem_frame, 2, 0, 3);
        op_pk_fd_set_int32 (gem_frame, 3, 0, 13);
        op_pk_fd_set_pkt (gem_frame, 4, next_data_pk, -1);
        op_pk_fd_set_pkt (up_alloc_tcont, i, gem_frame, -1);
        i++;
    }
    else stream_empty = 1;
}

/* DBRu field set, if requested */

if (dbru_req==1) {
    ttl_queue_data_ptr = (int *) op_ima_obj_svar_get (incoming_queue_2,
"ttl_queue_data");
    ttl_queue_pkts_ptr = (int *) op_ima_obj_svar_get (incoming_queue_2,
"ttl_queue_pkts");
    grant = ceil(((double)(*ttl_queue_data_ptr))/8.0)+5*(*ttl_queue_pkts_ptr);
    op_pk_fd_set_int32 (up_burst, 8, grant, dbru_fd_size);
}

if (up_alloc_tcont_exists == 1) op_pk_fd_set_pkt (up_burst, 9, up_alloc_tcont, -1);

/* DBRu & T-Cont 3 allocation */

/* T-Cont 3 payload set */

up_alloc_tcont_exists = 0;
stream_empty = 0;
tcont_alloc_space = (tcont_grants[2][1] - tcont_grants[2][0] + 1) ;
if (dbru_req==1) tcont_alloc_space-=dbru_fd_size;
head_pk_size_ptr = (int *) op_ima_obj_svar_get (incoming_queue_3,
"head_pk_size");

if ((tcont_alloc_space > 0) && (*head_pk_size_ptr != 0) ) {
    up_alloc_tcont = op_pk_create(0);
    up_alloc_tcont_exists = 1;
    op_intrpt_force_remote (tcont_alloc_space, incoming_queue_3);
}

i = 0;

while (stream_empty == 0) {
    if (op_strm_empty (3) == OPC_FALSE ) {
        next_data_pk = op_pk_get (3);

```

```

        data_pk_size = op_pk_total_size_get(next_data_pk);
        gem_frame = op_pk_create(0);
        op_pk_fd_set_int32 (gem_frame, 0, data_pk_size, 12);
        op_pk_fd_set_int32 (gem_frame, 1, 4*(onu_id-1)+2, 12);
        op_pk_fd_set_int32 (gem_frame, 2, 0, 3);
        op_pk_fd_set_int32 (gem_frame, 3, 0, 13);
        op_pk_fd_set_pkt (gem_frame, 4, next_data_pk, -1);
        op_pk_fd_set_pkt (up_alloc_tcont, i, gem_frame, -1);
        i++;
    }
    else stream_empty = 1;
}

/* DBRu field set, if requested */

if (dbru_req==1) {
    ttl_queue_data_ptr = (int *) op_ima_obj_svar_get (incoming_queue_3,
"ttl_queue_data");
    ttl_queue_pkts_ptr = (int *) op_ima_obj_svar_get (incoming_queue_3,
"ttl_queue_pkts");
    grant = ceil(((double)(*ttl_queue_data_ptr))/8.0)+5*(*ttl_queue_pkts_ptr);
    op_pk_fd_set_int32 (up_burst, 10, grant, dbru_fd_size);
}

if (up_alloc_tcont_exists == 1) op_pk_fd_set_pkt (up_burst, 11, up_alloc_tcont, -1);

/* DBRu & T-Cont 4 allocation */

/* T-Cont 4 payload set */

up_alloc_tcont_exists = 0;
stream_empty = 0;
tcont_alloc_space = (tcont_grants[3][1] - tcont_grants[3][0] + 1) ;
if (dbru_req==1) tcont_alloc_space-=dbru_fd_size;
head_pk_size_ptr = (int *) op_ima_obj_svar_get (incoming_queue_4,
"head_pk_size");
if ((tcont_alloc_space > 0) && (*head_pk_size_ptr != 0) ) {
    up_alloc_tcont = op_pk_create(0);
    up_alloc_tcont_exists = 1;
    op_intrpt_force_remote (tcont_alloc_space, incoming_queue_4);
}

i = 0;

while (stream_empty == 0) {
    if (op_strm_empty (4) == OPC_FALSE ) {
        next_data_pk = op_pk_get (4);
        data_pk_size = op_pk_total_size_get(next_data_pk);
        gem_frame = op_pk_create(0);
        op_pk_fd_set_int32 (gem_frame, 0, data_pk_size, 12);

```

```

        op_pk_fd_set_int32 (gem_frame, 1, 4*(onu_id-1)+3, 12);
        op_pk_fd_set_int32 (gem_frame, 2, 0, 3);
        op_pk_fd_set_int32 (gem_frame, 3, 0, 13);
        op_pk_fd_set_pkt (gem_frame, 4, next_data_pkt, -1);
        op_pk_fd_set_pkt (up_alloc_tcont, i, gem_frame, -1);
        i++;
    }
    else stream_empty = 1;
}

/* DBRu field set, if requested */

if (dbru_req==1) {
    ttl_queue_data_ptr = (int *) op_ima_obj_svar_get (incoming_queue_4,
"ttl_queue_data");
    ttl_queue_pkts_ptr = (int *) op_ima_obj_svar_get (incoming_queue_4,
"ttl_queue_pkts");
    grant = ceil(((double)(*ttl_queue_data_ptr))/8.0)+5*(*ttl_queue_pkts_ptr);
    op_pk_fd_set_int32 (up_burst, 12, grant, dbru_fd_size);
}

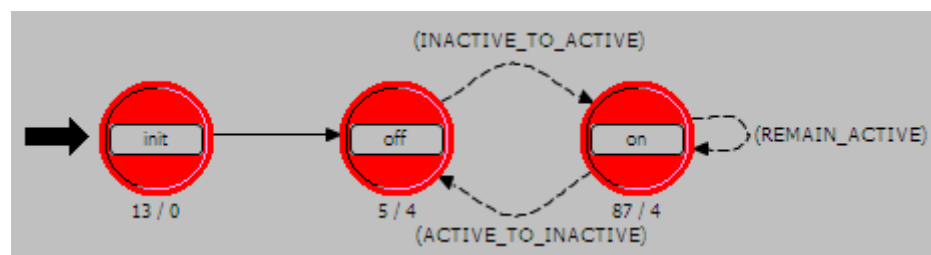
if (up_alloc_tcont_exists == 1) op_pk_fd_set_pkt (up_burst, 13, up_alloc_tcont, -1);

op_pk_send (up_burst, 0);

FOUT;
}

```

A.3 Η πηγή κίνησης “gpon bursty source”



A.3.1 Εκτελέσεις εισόδου κατάστασης “init”

```

/* Initialize the traffic generation parameters */
bursty_source_sv_init ();

/* Schedule the first OFF-period scheduling by setting */
/* a self-interrupt for the start time. If the start */

```

```

/* time is set to "Infinity", then there is no need to */
/* to schedule an interrupt as this node has been set */
/* will not generate any traffic */
if (start_time >= 0.0)
{
    op_intrpt_schedule_self (op_sim_time () + start_time, 0);
}

```

A.3.2 Εκτελέσεις εισόδου κατάστασης "off"

```

/* Schedule a self-interrupt to transit to the "ON" */
/* when the "OFF" state duration expires */
if (op_sim_time () + off_period < stop_time) {
    op_intrpt_schedule_self (op_sim_time () + off_period, OFF_TO_ON);
}

```

A.3.3 Εκτελέσεις εξόδου κατάστασης "off"

```

/* Determine the type of interrupt */
intrpt_type = op_intrpt_type ();
intrpt_code = op_intrpt_code ();

```

A.3.4. Εκτελέσεις εισόδου κατάστασης "on"

```

if (op_intrpt_code () == OFF_TO_ON) {
    off_state_start_time = op_sim_time () + on_period;
    /* Schedule a self-interrupt to transit to "OFF" */
    /* state when the "ON" state duration expires. */
    op_intrpt_schedule_self (off_state_start_time, ON_TO_OFF);
}

/* Generate the packets based on the loaded parameters */
/* for traffic generation */
next_packet_arrival_time = op_sim_time () + pk_interarrival_time;

/* Create a packet using the outcome of the loaded */
/* distribution */
pksize = floor ((double) oms_dist_positive_outcome_with_error_msg (packet_size_dist_handle, "This
occurs for packet size distribution in bursty_source process model.));
pksize*=8;
pkptr = op_pk_create_fmt ("data_pk");
op_pk_nfd_set_int32 (pkptr, "tcont", tcont);
op_pk_total_size_set (pkptr, pksize);

/* Record statistics to indicate that a packet */
/* was generated at the current simulation time */
op_stat_write (pksize_stathandle, (double) OPC_TRUE);

/* Update local statistics */

```

```

op_stat_write (bits_sent_stathandle,      pksize);
op_stat_write (pkts_sent_stathandle,     1.0);
op_stat_write (bitssec_sent_stathandle,  pksize);
op_stat_write (bitssec_sent_stathandle,  0.0);
op_stat_write (pktssec_sent_stathandle,  1.0);
op_stat_write (pktssec_sent_stathandle,  0.0);

/* Update global statistics */
op_stat_write (bits_sent_gstathandle,    pksize);
op_stat_write (pkts_sent_gstathandle,    1.0);
op_stat_write (bitssec_sent_gstathandle,  pksize);
op_stat_write (bitssec_sent_gstathandle,  0.0);
op_stat_write (pktssec_sent_gstathandle,  1.0);
op_stat_write (pktssec_sent_gstathandle,  0.0);

/* Loop through to remove segments of the original application packet and */
/* send them out to the lower layer */
if (segmentation_size > 0 && pksize > segmentation_size)
{
    /* Insert the packet into the segmentation buffer and pull out segments */
    /* of size specified by the attribute segment size. */
    op_sar_segbuf_pk_insert (segmentation_buf_handle, pkptr, 0);
    while (pksize > 0) {
        /* Remove segments of size equal to the segment size and send them to */
        /* transport layer. If the number of available bits in the buffer is */
        /* is lesser than the segment size a packet will be created only with */
        /* remaining bits */
        pkptr = op_sar_srcbuf_seg_remove (segmentation_buf_handle, segmentation_size);

        /* Update the pk_size remaining to reflect the reduced size. */
        pksize -= segmentation_size;

        /* Send the packet to the lower layer. */
        op_pk_send (pkptr, 0);
    }
}
else
{
    /* Send the packet to the lower layer */
    op_pk_send (pkptr, 0);
}

/* Check if the next packet arrival time is within the */
/* time in which the process remains in "ON" (active) */
/* state or not. Schedule the next packet arrival. */
if ((next_packet_arrival_time + PRECISION_RECOVERY < off_state_start_time) &&
    (next_packet_arrival_time + PRECISION_RECOVERY < stop_time))
{
    op_intrpt_schedule_self (next_packet_arrival_time, ON_TO_ON);
}

```

A.3.5 Εκτελέσεις εξόδου κατάστασης “on”

```
/* Determine the type of interrupt */  
intrpt_type = op_intrpt_type ();  
intrpt_code = op_intrpt_code ();
```

A.3.6. Ησυνάρτηση “bursty_source_sv_init()”

```
static void bursty_source_sv_init ()    {  
    Prohandle    my_prohandle;  
    int          my_pro_id;  
  
    Objid       my_id;  
    Objid       proc_id;  
    Objid       traf_gen_comp_attr_objid, traf_conf_objid;  
    Objid       pkt_gen_comp_attr_objid, pkt_gen_args_objid;  
  
    char        packet_size_string [128];  
    char        start_time_string [128];  
  
    double      bw_percentage;  
    double      pksize;  
    double      onu_bw;  
  
    OmsT_Dist_Handle    start_time_dist_handle;  
  
    /* Initializes state variables associated with this process model */  
  
    FIN (bursty_source_sv_init ());  
  
    /* Determine the prohandle of this process as well as */  
    /* the object IDs of the containing module, node and ONU proc */  
    my_prohandle = op_pro_self ();  
    my_pro_id = op_pro_id (my_prohandle);  
    my_id = op_id_self ();  
    proc_id = op_id_from_name (op_topo_parent(my_id), OPC_OBJTYPE_PROC, "proc");  
  
    /* Determine the process ID display string */  
    sprintf (pid_string, "bursty_source PID (%d)", my_pro_id);  
  
    /* Determine whether or not the simulation is in debug mode. Trace statement are only  
enabled when the simulation is in debug mode */  
    debug_mode = op_sim_debug ();  
  
    /* Read the T-Cont class of packet that will be generated */  
    op_ima_obj_attr_get (my_id, "T-Cont", &tcont);  
  
    /* Read the total Bandwidth requested by the ONU */
```



```

op_ima_obj_attr_get (proc_id, "Bandwidth Request", &onu_bw);

/* Read the traffic generation parameters */
op_ima_obj_attr_get (my_id, "Traffic Generation Parameters", &traf_gen_comp_attr_objid);
traf_conf_objid = op_topo_child (traf_gen_comp_attr_objid, OPC_OBJTYPE_GENERIC, 0);

/* Determine the start time for traffic generation */
op_ima_obj_attr_get (traf_conf_objid, "Start Time", start_time_string);
if (strcmp (start_time_string, "Never") != 0) {
    start_time_dist_handle = oms_dist_load_from_string (start_time_string);
    start_time = oms_dist_outcome (start_time_dist_handle);
}
else {
    start_time = -1.0;
}

/* Determine the stop time for traffic generation */
op_ima_obj_attr_get (traf_conf_objid, "Stop Time", &stop_time);
if (stop_time == -1.0) stop_time = OPC_DBL_INFINITY;

/* Determine the Burst Period for traffic generation */
op_ima_obj_attr_get (traf_conf_objid, "Burst Period", &burst_period);

/* Determine the burstiness factor for traffic generation */
op_ima_obj_attr_get (traf_conf_objid, "Burstiness", &burstiness);

/* If the start time is set to "Infinity", then there is no need to schedule an interrupt as this
node has been set will not generate any traffic */
if ((start_time >= 0.0) && (stop_time > start_time)) {
    /* Calculate ON state time */
    if (burstiness == -1.0) on_period = OPC_DBL_INFINITY ;
    else if (burstiness == 0.0) on_period = 0;
    else on_period = (burst_period)/(1+1/burstiness);

    /* Calculate OFF state time */
    if (burstiness == -1.0) off_period = 0;
    else if (burstiness == 0.0) off_period = OPC_DBL_INFINITY ;
    else off_period = (burst_period)/(1+burstiness);

    /* Calculate % of onu bw requested */
    op_ima_obj_attr_get (traf_conf_objid, "% of total ONU bandwidth",
&bw_percentage);
    bw_percentage*=0.01;

    /* packet generation arguments */
    op_ima_obj_attr_get (traf_conf_objid, "Packet Generation Arguments",
&pkt_gen_comp_attr_objid);
    pkt_gen_args_objid = op_topo_child (pkt_gen_comp_attr_objid,
OPC_OBJTYPE_GENERIC, 0);

    /* Load the distribution used to determine the size of the packets being generated */

```

```

op_ima_obj_attr_get (pkt_gen_args_objid, "Packet Size", packet_size_string);
packet_size_dist_handle = oms_dist_load_from_string (packet_size_string);

/*Calculate packet interarrival time for CBR and trimodal PDF traffic*/
if (tcont == 1) pksize = floor ((double) oms_dist_positive_outcome_with_error_msg
(packet_size_dist_handle,
"This occurs for packet size distribution in bursty_source process model."));
else pksize = (0.44*46 + 0.37*1488 + 0.09*278 + 0.07*980 + 0.0176*628 +
0.011*1300 + 0.0014*4470);

pk_interarrival_time = pksize*8/(bw_percentage*onu_bw*1000000);
if (burstiness != -1.0) {
    pk_interarrival_time *= on_period/(on_period+off_period);
}

/* Initilaize the packet generation status statistic to indicate that currently there are
no packets being generated */
pksize_stathandle = op_stat_reg ("Traffic Source.Packet Generation Status",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
op_stat_write (pksize_stathandle, (double) OPC_FALSE);

/* Initialize the statistic handles to keep track of traffic by this process */
bits_sent_stathandle = op_stat_reg ("Traffic Source.Traffic Sent (bits)",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
bitssec_sent_stathandle = op_stat_reg ("Traffic Source.Traffic Sent (bits/sec)",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
pkts_sent_stathandle = op_stat_reg ("Traffic Source.Traffic Sent (packets)",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
pktssec_sent_stathandle = op_stat_reg ("Traffic Source.Traffic Sent (packets/sec)",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
bits_sent_gstathandle = op_stat_reg ("Traffic Source.Traffic Sent (bits)",
OPC_STAT_INDEX_NONE, OPC_STAT_GLOBAL);
bitssec_sent_gstathandle = op_stat_reg ("Traffic Source.Traffic Sent (bits/sec)",
OPC_STAT_INDEX_NONE, OPC_STAT_GLOBAL);
pkts_sent_gstathandle = op_stat_reg ("Traffic Source.Traffic Sent (packets)",
OPC_STAT_INDEX_NONE, OPC_STAT_GLOBAL);
pktssec_sent_gstathandle = op_stat_reg ("Traffic Source.Traffic Sent (packets/sec)",
OPC_STAT_INDEX_NONE, OPC_STAT_GLOBAL);

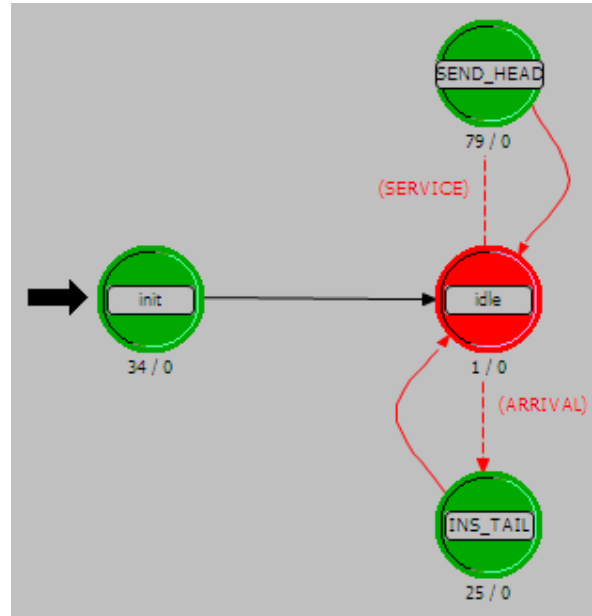
/* Check if packet segmentation is modeled */
op_ima_obj_attr_get (pkt_gen_args_objid, "Segmentation Size",
&segmentation_size);
if (segmentation_size != -1)
{
    segmentation_size *= 8;
    segmentation_buf_handle = op_sar_buf_create
(OPC_SAR_BUF_TYPE_SEGMENT, OPC_SAR_BUF_OPT_PK_BNDRY);
}
}

FOUT;

```

}

A.4 Ο ενταμιευτής “pc fifo enhanced”



A.4.1 Εκτελέσεις εισόδου κατάστασης “init”

```

Objid fprint_comp_objid;
Objid fprint_objid;
Objid olt_id;

```

```

if (!op_subq_empty (0)) {
    pkptr = op_subq_pk_access (0, OPC_QPOS_HEAD);
    head_pk_size = op_pk_total_size_get(pkptr);
    ttl_queue_data = op_subq_stat (0, OPC_QSTAT_BITSIZE);
    ttl_queue_pkts = op_subq_stat (0, OPC_QSTAT_PKSIZE);
}
else {
    head_pk_size = 0;
    ttl_queue_data = 0;
    ttl_queue_pkts = 0;
}

```

```

op_ima_obj_attr_get_str (op_id_self(), "name", 10, q_name);

```

```

/* Obtain print-to-file related attributes*/
olt_id = op_id_from_name (op_topo_parent(op_topo_parent (op_id_self()))),
OPC_OBJTYPE_NODE_FIX, "gpon_olt");
op_ima_obj_attr_get (olt_id, "Print to File", &fprint_comp_objid);
fprint_objid = op_topo_child (fprint_comp_objid, OPC_OBJTYPE_GENERIC, 0);

```

```
op_ima_obj_attr_get (fprint_objid, "Queue Report", &fprint_qreport);
```

A.4.2 Εκτελέσεις εισόδου κατάστασης "SEND HEAD"

```
int data_req;
int size;

Sbhandle sbh;

Packet * seg1_pkptr;
Packet * seg2_pkptr;

FILE * file;

/* a request has been made to access the queue, check if its empty */
if (!op_subq_empty (0)) {
    data_req = op_intrpt_code();
    if (fprint_qreport==1){
        file = fopen("queue_report.txt","a+");
        fprintf(file,"%lf %s %d %s", op_sim_time(), q_name, data_req ,"bits were requested,
");
    }
}

while ((data_req > 40) && (head_pk_size != 0)){
    if (head_pk_size <= (data_req - 40)) {

        pkptr = op_subq_pk_remove (0, OPC_QPOS_HEAD);
        op_pk_send_quiet (pkptr, 0);
        if (fprint_qreport==1) fprintf(file,"%d %s", head_pk_size ,"bits sent,");
        data_req = data_req - head_pk_size - 40;
        if (!op_subq_empty (0)) {
            pkptr = op_subq_pk_access (0, OPC_QPOS_HEAD);
            head_pk_size = op_pk_total_size_get(pkptr);
        }
        else head_pk_size = 0;
    }
    else {
        sbh = op_sar_buf_create (OPC_SAR_BUF_TYPE_SEGMENT,
        OPC_SAR_BUF_OPT_DEFAULT );
        pkptr = op_subq_pk_remove (0, OPC_QPOS_HEAD);
        op_sar_segbuf_pk_insert (sbh, pkptr, OPC_NIL);
        seg1_pkptr = op_sar_srcbuf_seg_remove (sbh, data_req - 40);
        size=op_pk_total_size_get(seg1_pkptr);
        if (fprint_qreport==1) fprintf(file," %s %d", "next packet too big, seg1 sent=", size);

        op_pk_send_quiet (seg1_pkptr, 0);
        size = head_pk_size - size;
        seg2_pkptr = op_sar_srcbuf_seg_remove (sbh, size);
        head_pk_size = op_pk_total_size_get(seg2_pkptr);
        op_subq_pk_insert (0, seg2_pkptr, OPC_QPOS_HEAD);
    }
}
```

```

    op_sar_buf_destroy (sbh);
    data_req=0;
}
}

/*update total data in queue state variable*/
ttl_queue_data = op_subq_stat (0, OPC_QSTAT_BITSIZE);
ttl_queue_pkts = op_subq_stat (0, OPC_QSTAT_PKSIZE);
if (fprint_qreport==1) {
    fprintf(file," %s %d %s %d\n", "new head pk(seg2)=", head_pk_size, "new ttl=",
    ttl_queue_data);
    fclose (file);
}
}
}

```

A.4.3 Εκτελέσεις εισόδου κατάστασης "INS TAIL"

```

FILE * file;

int  pk_size;

/* a new packet has arrived; acquire it */
pkptr = op_pk_get (op_intrpt_strm ());
pk_size = op_pk_total_size_get (pkptr);

if (op_subq_empty (0)) head_pk_size= pk_size;
op_subq_pk_insert (0, pkptr, OPC_QPOS_TAIL);

/*update total data in queue state variable*/
ttl_queue_data = op_subq_stat (0, OPC_QSTAT_BITSIZE);
ttl_queue_pkts = op_subq_stat (0, OPC_QSTAT_PKSIZE);

if (fprint_qreport==1){
    file = fopen("queue_report.txt","a+");
    fprintf(file,"%lf %s %s %d %s %d\n", op_sim_time(), q_name ,"a pk arrived, new header=",
    head_pk_size, "new ttl=", ttl_queue_data);
    fclose (file);
}
}

```

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Βενιέρης Ι.Σ. *Δίκτυα Ευρείας Ζώνης, 2^η Έκδοση*. Θεσσαλονίκη: Εκδόσεις Τζιόλα, 2007.
- [2] Ορφανουδάκης Θ.Γ. *Ενσύρματα Δίκτυα Πρόσβασης Ευρείας Ζώνης*. Τρίπολη: 2006.
- [3] Wikipedia, the free encyclopedia. <<http://en.wikipedia.org>>.
- [4] Recommendation ITU-T G.984.1 (2008), *Gigabit-capable passive optical networks (G-PON): General characteristics*.
- [5] Recommendation ITU-T G.984.2 (2003), *Gigabit-capable passive optical networks (G-PON): Physical Media Dependent (PMD) layer specification*.
- [6] ITU-T Recommendation G.984.2/Amd.1 (2006), *Gigabit-capable Passive Optical Networks (G-PON): Physical Media Dependent (PMD) layer specification, Amendment 1: New Appendix III – Industry best practice for 2.488 Gbit/s downstream, 1.244 Gbit/s upstream G-PON*.
- [7] Recommendation ITU-T G.984.3 (2008), *Gigabit-capable passive optical networks (G-PON): Transmission convergence layer specification*
- [8] Recommendation ITU-T G.984.4 (2008), *Gigabit-capable passive optical networks (G-PON): ONT management and control interface specification*.
- [9] Tom Van Caenegem. *B-PON, G-PON and EPON: A guided tour*.
- [10] T. Orphanoudakis, H.-C. Leligou, E. Kosmatos, J. D. Angelopoulos. "Performance Evaluation of GPON vs. EPON for Multi-Service Access." *International Journal of Communication Systems*, Dec 2007.
- [11] B. Skubic, J. Chen, J. Ahmed, L. Wosinska, B. Mukherjee. "A Comparison of Dynamic Bandwidth Allocation for EPON, GPON, and Next-Generation TDM PON." *IEEE Communications Magazine*, March 2009.
- [12] W. John, S. Tafvelin. "Analysis of Internet Backbone Traffic and Header Anomalies observed". IMC'07, San Diego, CA, USA, August 2007.