ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

Εργασία που υποβλήθηκε
στα πλαίσια του μαθήματος

"Quality of Services - QoS"

στον Επίκουρο Καθηγητή

Τσαουσίδη Βασίλειο

από τους Υποψήφιους Διδάκτορες

Βαλσαμίδη Σταύρο
Κοντογιάννη Σωτήριο

Ξάνθη 2004

Προσδιορισμός της TCP συμπεριφοράς των web εξυπηρετητών

Περίληψη

Το μεγαλύτερο μέρος της κυκλοφορίας σήμερα στο Διαδίκτυο ελέγχεται από το πρωτόκολλο ελέγχου μετάδοσης (Transmission Control Protocol - TCP). Ως εκ τούτου, η απόδοση του TCP ασκεί σημαντική επίδραση στην συνολική απόδοση του Διαδικτύου. Το TCP είναι ένα σύνθετο πρωτόκολλο με πολλές διαμορφώσιμες από τον χρήστη παραμέτρους και μια σειρά από διαφορετικές υλοποιήσεις. Επιπλέον η σχετική έρευνα συνεχίζει να παράγει τις νέες εξελίξεις στους μηχανισμούς ελέγχου συμφόρησης και τις επιλογές του TCP και είναι χρήσιμο να επισημανθεί η ανάπτυξη αυτών των νέων μηχανισμών στο Διαδίκτυο. Σαν τελική θεώρηση, η σταθερότητα και η δικαιοσύνη του τρέχοντος Διαδικτύου στηρίζονται στην εθελοντική χρήση των μηχανισμών ελέγχου συμφόρησης. Επομένως είναι σημαντικό να εξεταστούν οι εφα ρμογές TCP για την συμμόρφωση τους, στον έλεγχο συμφόρησης από το ένα άκρο στο άλλο. Έχει αναπτυχθεί ένα εργαλείο αποκαλούμενο **TCP Εργαλείο Προσδιορισμού Συμπεριφοράς – ΤΕΠΣ (TCP Behaviour Identification Tool - TBIT)** για τον χαρακτηρισμό της TCP συμπεριφοράς ενός απομακρυσμένου web εξυπηρετητή. Εδώ, περιγράφεται το TBIT και παρουσιάζονται τα αποτελέσματα για τις TCP συμπεριφορές σημαντικών web εξυπηρετητών που λήφθηκαν χρησιμοποιώντας αυτό το εργαλείο. Περιγράφεται επίσης η χρήση του TBIT για την ανίχνευση προβλημάτων και τη μη συμμόρφωση στις TCP υλοποιήσεις που αναπτύσσονται στους δημόσιους web εξυπηρετητές.

1. Εισαγωγή

Το μεγαλύτερο μέρος της κυκλοφορίας που υπάρχει σήμερα στο διαδίκτυο ελέγχεται από το **Πρωτόκολλο Ελέγχου Μετάδοσης (Transmission Control Protocol – TCP)** [6,27]. Κατά συνέπεια, η απόδοση του TCP ασκεί σημαντική επίδραση στην συνολική απόδοση του Διαδικτύου. Η κατανόηση της συμπεριφοράς του TCP πρωτοκόλλου μπορεί να είναι σημαντική για το Διαδίκτυο την σχετική έρευνα, τους ISPs, τους προμηθευτές λειτουργικών συστημάτων και τους δημιουργούς εφαρμογών. Η TCP συμπεριφορά σημαντικών web εξυπηρετητών είναι ιδιαίτερου ενδιαφέροντος. Έχει σχεδιαστεί ένα εργαλείο αποκαλούμενο **TCP Εργαλείο Προσδιορισμού Συμπεριφοράς – ΤΕΠΣ (TCP Behaviour Identification Tool - TBIT)**, για να χαρακτηρίζει τη TCP συμπεριφορά των απομακρυσμένων web εξυπηρετητών, χωρίς να απαιτείται η ύπαρξη κάποιων ειδικών προνομίων σ' αυτούς τους web εξυπηρετητές.

Το TCP είναι ένα σύνθετο πρωτόκολλο με ένα φάσμα διαμορφώσιμων παραμέτρων από τον χρήστη. Στο βασικό TCP πρωτόκολλο [22] έχουν προταθεί και αναπτυχθεί πολλές παραλλαγές. Οι παραλλαγές στο μηχανισμό ελέγχου συμφόρησης συνεχίζουν να αναπτύσσονται μαζί με νέες επιλογές του πρωτοκόλλου, όπως η **Επιλεκτική Αναγνώριση (Selective Acknow ledgment - SACK)** και η **Άμεση Γνωστοποίηση Συμφόρησης (Explicit Congestion Notification - ECN)**.

Όπως εξηγεί η παράγραφος 4.3, το TCP Reno είναι μια παλαιότερη παραλλαγή του ελέγχου συμφόρησης του TCP πρωτοκόλλου από το 1990, που αποδίδει ιδιαίτερα άσχημα όταν τα πολλαπλά πακέτα απορρίπτονται σε ένα παράθυρο δεδομένων. Το TBIT δείχνει ότι οι νεώτερες παραλλαγές του TCP όπως το NewReno και το SACK αναπτύσσονται ευρέως στο Διαδίκτυο και αυτό το γεγονός πρέπει να ληφθεί υπόψη για τις μελέτες προσομοίωσης και ανάλυσης του TCP. Θεωρείται ότι είναι η πρώτη φορά που αναφέρεται το πώς ποσοτικά δεδομένα μπορούν για να απαντήσουν σε ερωτήσεις σχετικά με είδος του TCP πρωτοκόλλου (μηχανισμός συμφόρησης, διόρθωσης λαθών). Με άλλα λόγια, το TBIT βοηθά στην τεκμηρίωση της ενσωμάτωσης των νέων TCP μηχανισμών στο δημόσιο Διαδίκτυο.

Το TBIT μπορεί ακόμα να ελένξει και να απαντήσει σε ερωτήσεις όπως: *«ποια είναι τα αρχικά παράθυρα που χρησιμοποιούνται στις TCP συνδέσεις στο Διαδίκτυο»*. Όπως εξηγείται στην παράγραφο 4.2, το αρχικό παράθυρο του TCP καθορίζει το ποσό των δεδομένων που μπορεί να διαβιβαστεί στον πρώτο «**round-trip**» **χρόνο**, μετά την εγκατάσταση μιας TCP σύνδεσης. Το αρχικό παράθυρο είναι μια παράμετρος διαμορφώσιμη από την εφαρμογή. Το αρχικό TCP παράθυρο που χρησιμοποιείται σε έναν web εξυπηρετητή δεν μπορεί να προκύψει απλά με τη γνώση του λειτουργικού συστήματος που χρησιμοποιείται σε εκείνο τον εξυπηρετητή. Η γνώση της κατανομής των ρυθμιζόμενων τιμών στα αρχικά παράθυρα, μπορεί να είναι χρήσιμη όχι μόνο στις προσομοιώσεις και την μοντελοποίηση, αλλά και στις αποφάσεις του σώματος προτύπων για την προώθηση μεγαλύτερων τιμών στα αρχικά παράθυρα [2].

Ένα ακόμα κίνητρο για το TBIT είναι να έχει την δυνατότητα εύκολα να επαληθευθεί ο έλεγχος συμφόρησης από το ένα άκρο στο άλλο, που χρησιμοποιείται στους hosts του Διαδικτύου (παράγραφος 4.4). Η ευστάθεια και η δικαιοσύνη του συνολικού Διαδικτύου εξαρτώνται προς το παρόν από την εθελοντική χρήση των μηχανισμών ελέγχου συμφόρησης, στην υλοποίηση των σωρών του TCP πρωτοκόλλου στους τελικούς hosts. Πιστεύεται, ότι η δυνατότητα προσδιορισμού δημόσια των end hosts που δεν συμμορφώνονται στον έλεγχο συμφόρησης από το ένα άκρο στο άλλο, μπορεί να βοηθήσει σημαντικά στην ενίσχυση της σωστής χρήσης αυτών των μηχανισμών από το ένα άκρο στο άλλο στο Διαδίκτυο.

Το TBIT μπορεί να βοηθήσει στον προσδιορισμό και τη διόρθωση των λαθών που ανιχνεύονται στις υλοποιήσεις του TCP πρωτοκόλλου. Χρησιμοποιώντας το TBIT, ανιχνεύονται λάθη στα προϊόντα των Microsoft, Cisco, SUN και της IBM και έχουν βοηθηθεί οι προμηθευτές στο να διορθώσουν αυτά τα λάθη. Για παράδειγμα, καθώς αρχίζει να αναπτύσσεται στο Διαδίκτυο η **Άμεση Γνωστοποίηση συμφόρησης –ΑΓΣ (Explicit Congestion Notification - ECN)** (παράγραφος 4.6), εμφανίζονται αναφορές για web εξυπηρετητές που αδυνατούν να επικοινωνήσουν με νεώτερα αναπτυχθέντες πελάτες. Το TBIT έχει χρησιμοποιηθεί για να βοηθήσει στον προσδιορισμό αυτών των τρόπων αποτυχίας και την ανίχνευση της προόδου (ή της έλλειψης προόδου) στην ανάπτυξη αυτών διορθώσεων. Οι πληροφορίες όπως αυτές, είναι κρίσιμες όταν τυποποιούνται οι νέοι μηχανισμοί πρωτοκόλλου όπως το ECN και επεκτείνονται πραγματικά στο Διαδίκτυο. Επιπλέον, όπως θα φανεί στις παραγράφους 4.3 και 4.5, λεπτά λάθη, μπορούν να προκαλέσουν μια υλοποίηση TCP, να συμπεριφερθεί αρκετά διαφορετικά από αυτά που ισχυρίζονται οι προμηθευτές. Από την σκοπιά του χρήστη, ένα εργαλείο όπως το TBIT είναι ουσιαστικό για την ανίχνευση τέτοιων λαθών.

Το υπόλοιπο του εγγράφου οργανώνεται ως εξής. Στην παράγραφο 2, περιγράφεται η σχεδίαση TBIT. Στην παράγραφο 3, συγκρίνεται και αντιπαραβάλλεται το TBIT με άλλα εργαλεία. Στην παράγραφο 4, παρουσιάζονται τα αποτελέσματα που επιτεύχθηκαν με τη χρησιμοποίηση του εργαλείου TBIT για την ανάπτυξη του TCP σε μερικούς δημοφιλείς web εξυπηρετητές.

2. Αρχιτεκτονική TBIT

Στόχος του TBIT είναι να αναπτυχθεί ένα εργαλείο για να χαρακτηρίσει την TCP συμπεριφορά των σημαντικών web εξυπηρετητών. Η πρώτη απαίτηση για την σχεδίαση του TBIT είναι ότι το TBIT θα πρέπει να έχει τη δυνατότητα να εξετάζει οποιοδήποτε web εξυπηρετητή, οποιαδήποτε στιγμή. Μια δεύτερη απαίτηση είναι ότι η κυκλοφορία που παράγεται από το TBIT δεν πρέπει να είναι εχθρική, ή ακόμα και να εμφανίζεται εχθρική ή έξω από την συνηθισμένη, στον απομακρυσμένο web εξυπηρετητή που εξετάζεται. Η ικανοποίηση της πρώτης απαίτησης, δεν μπορεί να απαιτεί από οποιεσδήποτε υπηρεσίες, δικαιώματα ή ειδικά προνόμια στο web εξυπηρετητή, που δεν είναι διαθέσιμες στο ευρύ κοινό. Επιπλέον, καμία υπόθεση δεν μπορεί να γίνει για το υλικό ή το λογισμικό που τρέχει στον απομακρυσμένο web εξυπηρετητή. Η δεύτερη απαίτηση της συνηθισμένης και μη εχθρικής κυκλοφορίας είναι αντίθετη με προγράμματα όπως το **NMAP** [11], που εκμεταλλεύονται την απόκριση των απομακρυσμένων TCP εξυπηρετητών στις ασυνήθεις σειρές πακέτων, όπως την αποστολή FIN σε μια θύρα, χωρίς άνοιγμα μιας σύνδεσης TCP (με τη διαδικασία της τριπλής χειραψίας). Τέτοιες τακτικές είναι συνήθως εύκολο να αναγνωριστούν και πολλοί web εξυπηρετητές χρησιμοποιούν **firewalls** για να ανιχνεύουν και να εμποδίζουν τις ασυνήθιστες ακολουθίες πακέτων. Προκειμένου να εξασφαλιστεί η δυνατότητα να εξεταστεί οποιοσδήποτε web εξυπηρετητής οποιαδήποτε στιγμή, το TBIT παράγει μόνο συμμορφούμενη με το TCP κυκλοφορία, που δεν θα σημειωθεί ως εχθρική από τα firewalls. Το TBIT παρέχει διάφορες δοκιμές, κάθε μια με σκοπό να εξετάσει μια συγκεκριμένη πτυχή της TCP συμπεριφοράς του απομακρυσμένου web εξυπηρετητή. Παρακάτω, περιγράφεται η δοκιμή του αρχικού παραθύρου, που επεξηγεί διάφορα εντυπωσιακά χαρακτηριστικά γνωρίσματα της αρχιτεκτονικής TBIT. Διάφορες άλλες δοκιμές που υλοποιήθηκαν στο TBIT περιγράφονται στην παράγραφο 4.

Η διαδικασία TBIT εγκαθιστά και διατηρεί μια TCP σύνδεση με τον απομακρυσμένο host. Η διαδικασία TBIT κατασκευάζει τα TCP πακέτα για να τα στείλει σε έναν απομακρυσμένο host. Εγκαθιστά επίσης ένα firewall για να αποτρέψει πακέτα από το απομακρυσμένο host να προσεγγίσουν τον πυρήνα της τοπικής μηχανής. Συγχρόνως, μια **BSD Packet Filter (BPF)** συσκευή [17] φίλτρων πακέτων, χρησιμοποιείται για να παραδώσει αυτά τα πακέτα στη διαδικασία TBIT. Αυτή η σύνδεση μπορεί να εξαγάγει πληροφορίες για την TCP υλοποίηση του απομακρυσμένου εξυπηρετητή.

Για επεξήγηση, ας θεωρηθεί το πρόβλημα της μέτρησης της **αρχικής τιμής του παραθύρου συμφόρησης (Intitial Congestion Window - ICW)** που χρησιμοποιείται από τους web εξυπηρετητές. Αυτή η τιμή είναι ο αριθμός των bytes που ένας αποστολέας μπορεί να στείλει σε έναν παραλήπτη, αμέσως μετά από την εγκατάσταση της σύνδεσης, πριν λάβει οποιαδήποτε ACKs από τον παραλήπτη. Το πρότυπο του TCP [3] καθορίζει ότι για ένα δεδομένο **Μέγιστο Μέγεθος Τμήματος (Maximum Segment Size - MSS)** το ICW μπορεί να είναι το πολύ **2*MSS** bytes και ένα εμπειρικό πρότυπο [2] επιτρέπει στο ICW να τεθεί ως:

Min (4*MSS, max(2*MSS, 4380)) bytes

Δεδομένου ότι η πλειοψηφία των ιστοσελίδων είναι μεγέθους κάτω από 10KB [4, 6, 19], η τιμή του ICW μπορεί να έχει σημαντική επίδραση στην απόδοση του web εξυπηρετητή [15]. Η δοκιμή TBIT που μετράει την τιμή ICW που χρησιμοποιείται από έναν web εξυπηρετητή λειτουργεί ως εξής. Υποθέτει ότι το TBIT τρέχει στον host A και ο απομακρυσμένος web εξυπηρετητής τρέχει στον host B.

- Το TBIT ανοίγει μια απλή υποδοχή IP.
- Το TBIT δημιουργεί μια συσκευή BPF και θέτει το φίλτρο έτσι ώστε να συλλάβει όλα τα πακέτα που πηγαίνουν στον A και που προέρχονται από τον οικοδεσπότη B.
- Το TBIT εγκαθιστά ένα firewall στο A για να αποτρέψει οποιαδήποτε πακέτα του host B από το να προσεγγίσουν τον πυρήνα του host A.
- Το TBIT στέλνει ένα πακέτο TCP SYN, με τη διεύθυνση προορισμού του host B και μια θύρα προορισμού 80. Το πακέτο διαφημίζει ένα πολύ μεγάλο παράθυρο αποστολέα και επιθυμητό MSS.
- Ο σωρός TCP που τρέχει στον host B θα δει αυτό το πακέτο και θα αποκριθεί με ένα πακέτο SYN - ACK.
- Το SYN - ACK φθάνει στον host A. Το firewall εμποδίζει τον πυρήνα να δει αυτό το πακέτο, ενώ η συσκευή BPF παραδίδει αυτό το πακέτο στη διαδικασία TBIT.
- Το TBIT δημιουργεί ένα πακέτο που περιέχει το αίτημα HTTP GET για τη σελίδα βάσης ("/"), μαζί με τον κατάλληλο πεδίο ACK, αναγνωρίζοντας το SYN - ACK. Αυτό το πακέτο στέλνεται στον οικοδεσπότη B.
- Μετά την λήψη του αιτήματος GET, ο host B θα αρχίσει να στέλνει τα πακέτα δεδομένων για την ιστοσελίδα βάσης στον host A.
- Το TBIT δεν αναγνωρίζει περαιτέρω πακέτα που στέλνονται από τον οικοδεσπότη B. Ο σωρός TCP που τρέχει στον host B θα είναι σε θέση μόνο να στείλει τα πακέτα που είναι μέσα στο ICW του και έπειτα κάνει time out, αναμεταδίδοντας τελικά το πρώτο πακέτο.
- Μόλις το TBIT δει αυτό το επανασταλμένο πακέτο, στέλνει ένα πακέτο με την ένδειξη RST στον host B. Αυτό κλείνει τη TCP σύνδεση.

Η τιμή του ICW που χρησιμοποιείται από το σωρό TCP που τρέχει στον host B, δίνεται από τον αριθμό μοναδικών Bytes δεδομένων που στέλνονται από τον host B μέ το τέλος της δοκιμής.

Τρία εμφανή χαρακτηριστικά γνωρίσματα της αρχιτεκτονικής TBIT εμφανίζονται σε αυτήν την δοκιμή. Κατ' αρχήν, αυτή η δοκιμή μπορεί να τρέξει σε οποιοδήποτε web εξυπηρετητή και δεν απαιτεί κάποια ειδικά προνόμια στον web εξυπηρετητή που εξετάζεται. Δεύτερον, σημειώνεται η δυνατότητα του TBIT να συνθέτει TCP πακέτα. Αυτό επιτρέπει να συμπεραίνεται η τιμή ICW για οποιοδήποτε MSS, με τον καθορισμό των κατάλληλων επιλογών στο πακέτο SYN. Αυτή η δυνατότητα είναι σημαντική για διάφορες άλλες δοκιμές που υλοποιούνται στο TBIT. Τέλος, η κυκλοφορία που παράγεται κατά τη διάρκεια της δοκιμής ICW θα εμφανιστεί να συμμορφώνεται με την TCP κυκλοφορία σε οποιαδήποτε οντότητα ελέγχου.

Η δοκιμή ενσωματώνει διάφορα μέτρα για να αυξηθεί η ευρωστία και να εξασφαλιστεί η ακρίβεια των αποτελεσμάτων. Η ευρωστία σε σχέση με τα λάθη που προκαλούνται από τις απώλειες πακέτων είναι μια σημαντική απαίτηση. Η απώλεια του SYN, του SYN - ACK ή του πακέτου που φέρνει το αίτημα «*HTTP GET*» εξετάζεται κατά τρόπο παρόμοιο με το TCP, δηλαδή χρησιμοποιώντας τις επαναμεταδόσεις που προκαλούνται από τα timeouts. Είναι πιο δύσκολο να αντιμετωπιστεί η απώλεια πακέτων δεδομένων που στέλνονται από τον host B. Μερικές απώλειες είναι ανιχνεύσιμες με την παρατήρηση ενός χάσματος στους ακολουθιακούς αριθμούς σειράς των Bytes δεδομένων που φθάνουν. Εάν το

TBIT ανιχνεύσει ένα τέτοιο χάσμα στους ακολουθιακούς αριθμούς σειράς, τερματίζει τη δοκιμή, χωρίς επιστροφή ενός αποτελέσματος. Πάντως, το TBIT δεν μπορεί πάντα να ανιχνεύει τα χαμένα πακέτα εάν τα διαδοχικά πακέτα στο τέλος του παραθύρου συμφόρησης χαθούν. Σε τέτοιες περιπτώσεις, το αποτέλεσμα του TBIT μπορεί να μην είναι σωστό. Κάποια ευρωστία σε σχέση με αυτό το λάθος μπορεί να επιτευχθεί με το τρέξιμο της δοκιμής πολλαπλές φορές. Μια άλλη περίπτωση είναι η ιστοσελίδα βάσης να μην είναι αρκετά μεγάλη να γεμίσει το αρχικό παράθυρο για ένα δεδομένο MSS. Εάν συμβεί αυτό, τότε ο απομακρυσμένος web εξυπηρετητής συνήθως θα στείλει ένα FIN είτε στο τελευταίο πακέτο δεδομένων, είτε αμέσως μετά από το τελευταίο πακέτο δεδομένων. Το TBIT μπορεί να το ανιχνεύσει αυτό. Για πρόσθετη ευρωστία, ο χρήστης μπορεί να διεξάγει τη δοκιμή με ένα διαφορετικό MSS, ή να καθορίσει το URL ενός μεγαλύτερου αντικειμένου στον web εξυπηρετητή, εάν ένα τέτοιο URL είναι γνωστό.

Το TBIT σχεδιάστηκε σε αρθρωτή μορφή. Ένας πυρήνας συνόλου λειτουργιών χρησιμοποιείται για την σύνθεση, την αποστολή και τη λήψη των TCP πακέτων καθώς επίσης και για την καταγραφή δεδομένων, λειτουργικότητας στην διασύνδεση με τον χρήστη. Διάφορες δοκιμές, όπως το αρχικό παράθυρο ελέγχου που περιγράφεται παραπάνω, χρησιμοποιούν αυτές τις λειτουργίες. Η αρθρωτή σχεδίαση καθιστά το εργαλείο επεκτάσιμο και οι νέες δοκιμές μπορούν να προστεθούν εύκολα. Έχουν υλοποιηθεί διάφορες τέτοιες δοκιμές στο TBIT, για να ελεγχθούν οι πτυχές της TCP συμπεριφοράς του απομακρυσμένου web εξυπηρετητή. Έχει περιγραφεί παραπάνω η δοκιμή του ICW. Στην συνέχεια εξετάζονται πέντε άλλες: μια δοκιμή για να καθορίσει την έκδοση του αλγόριθμου συμφόρησης (Tahoe, Reno, NewReno κ.λ.π.), που τρέχουν στον απομακρυσμένο web εξυπηρετητή, μια δοκιμή για να αποφασίσει εάν ο απομακρυσμένος web εξυπηρετητής μειώνει το παράθυρο συμφόρησής του στο μισό σε απόκριση της απόρριψης πακέτου, μια δοκιμή για να καθορίσει εάν ο απομακρυσμένος web εξυπηρετητής υποστηρίζει το SACK και χρησιμοποιεί σωστά τις πληροφορίες SACK, μια δοκιμή για να καθορίσει εάν ο απομακρυσμένος ιστός υποστηρίζει την ECN και τελικά μια δοκιμή για να μετρήσει τη διάρκεια του χρόνου αναμονής στον απομακρυσμένο web εξυπηρετητή. Επελέγησαν αυτές οι δοκιμές για να επεξηγηθούν καλύτερα η προσαρμοστικότητα του TBIT, καθώς επίσης και για να αναφερθούν οι ενδιαφέρουσες TCP συμπεριφορές που έχουν παρατηρηθεί.

3. Σχετική εργασία

Υπάρχουν διάφοροι τρόποι να αποσπαστούν πληροφορίες για την TCP συμπεριφορά ενός απομακρυσμένου web εξυπηρετητή. Στην προηγούμενη παράγραφο, περιγράφηκε η αρχιτεκτονική του TBIT με λεπτομέρεια.

Μια πιθανή προσέγγιση για να αποσπαστεί και να προσδιοριστεί ενεργά η TCP συμπεριφορά θα ήταν να χρησιμοποιηθεί ένα πρότυπο TCP στον web πελάτη τέτοιο ώστε να ζητήσει την ιστοσελίδα από τον εξυπηρετητή και να απορρίψει τα συγκεκριμένα πακέτα στη TCP σύνδεση (π.χ. όπως απορρίφθηκαν ACKs για τη δοκιμή ICW). Μια πιο σύνθετη εναλλακτική λύση θα ήταν να χρησιμοποιηθεί ένας προσομοιωτής όπως το NS [8] για να απορρίψει τα συγκεκριμένα πακέτα από τη TCP σύνδεση, σύμφωνα με τις μεθόδους του Dummynet [24]. Πάντως και οι δύο αυτές προσεγγίσεις στερούνται ευελιξιών που θα ήταν επιθυμητές. Δεδομένου ότι θα περιγραφούν στην παράγραφο 4.3 μερικές από τις δοκιμές, πρέπει να εξασφαλιστεί ότι θα ληφθεί ένας σημαντικός αριθμός πακέτων (20 έως 25) σε μια και μόνο μεταφορά. Αντί ο αποστολέας να ψάξει για μεγάλα αντικείμενα σε κάθε διαδικτυακό τόπο, ο ευκολότερος τρόπος να γίνει αυτό είναι να ελεγχθεί το μέγεθος πακέτων του αποστολέα σε Bytes, καθορίζοντας ένα μικρό μέγιστο μέγεθος τμήματος (Maximum Segment Size - MSS) στον παραλήπτη TCP. Αυτό δεν θα ήταν εύκολο να διεκπεραιωθεί είτε με το Dummynet είτε τον προσομοιωτή NS. Χωρίς τη δυνατότητα καθορισμού ενός μικρού MSS, δεν μπορεί να είμαστε σε θέση να εξετάσουμε πολλούς web εξυπηρετητές της επιλογής μας.

Μια εκτενής μελέτη της TCP συμπεριφοράς των διαδικτυακών hosts παρουσιάζεται στο [20]. Η μελέτη πραγματοποιήθηκε χρησιμοποιώντας ένα σταθερό σύνολο hosts Διαδικτύου στο οποίο ο συντάκτης είχε λάβει ειδικά δικαιώματα (χρήση «tcpdump», Raw packet construction). Μεγάλες μεταφορές αρχείων πραγματοποιήθηκαν μεταξύ των ζευγαριών των hosts που ανήκουν σε αυτό το σύνολο και τα ίχνη πακέτων που μεταφέρθηκαν, καταγράφηκαν με τη χρησιμοποίηση του «tcpdump» και στους δύο hosts. Τα ίχνη αναλύθηκαν, για να καθοριστεί η TCP συμπεριφορά των hosts που ενεπλάκησαν. Το έγγραφο ανέφερε σχετικά με την TCP απόδοση οκτώ σημαντικών TCP υλοποιήσεων. Στο έγγραφο αναφέρεται η αποτυχία ανάπτυξης ενός πλήρους και γενικού εργαλείου, για αυτόματη ανάλυση της συ-

μπεριφοράς μιας TCP υλοποίησης από ίχνη TCP πακέτων.

Σημειώνεται η μεθοδολογία που χρησιμοποιείται στο [20] δεν θα ταίριαζε με τους σκοπούς που τέθηκαν όσον αφορά τις συγκεκριμένες συμπεριφορές TCP των web εξυπηρετητών. Κατ' αρχάς, ο περιορισμός στους διαδικτυακούς hosts όπου θα μπορούσαν να ληφθούν τα απαραίτητα δικαιώματα, δεν θα επέτρεπε τις δοκιμές των web εξυπηρετητών σε ευρεία διάδοση. Δεύτερον, ορισμένες συμπεριφορές TCP των κόμβων τέλους, μπορούν να αναγνωριστούν μόνο εάν το σωστό μοντέλο απώλειας και καθυστέρησης εμφανιστεί κατά τη διάρκεια της μεταφοράς δεδομένων.

Οι συντάκτες του [20] εξετάζουν τις υλοποιήσεις του TCP/IP σε τρία βασικά λειτουργικά συστήματα δηλαδή, το FreeBSD 4.0, τα Windows 2000 και το Linux (Slackware 7.0), με τη χρήση προσομοιωμένων μεταφορών αρχείων σε ένα ελεγχόμενο εργαστήριο δοκιμών. Εισάγονται τα συγκεκριμένα μοντέλα απώλειας - καθυστέρησης χρησιμοποιώντας το «Dummynet» [24]. Οι συντάκτες εκθέτουν διάφορες ατέλειες, στις υλοποιήσεις του TCP/IP των λειτουργικών συστημάτων που εξέτασαν. Δεδομένου ότι η μεθοδολογία απαιτεί τον πλήρη έλεγχο και των δύο hosts, καθώς επίσης και των δρομολογητών μεταξύ των (για να εισαγάγουν την απώλεια και την καθυστέρηση), μια τέτοια δοκιμή δεν μπορεί να χρησιμοποιηθεί για να απαντήσει σε ερωτήσεις που αφορούν την ανάπτυξη TCP στο συνολικό Διαδίκτυο.

Το «NMAP» [11] είναι ένα εργαλείο για την αναγνώριση των λειτουργικών συστημάτων (ΛΣ) που τρέχουν στους απομακρυσμένους host στο Διαδίκτυο. Το «NMAP» ερευνά τις απομακρυσμένες μηχανές με ποικιλία, από συνηθισμένες και έξω από τις συνηθισμένες ακολουθίες TCP/IP πακέτων. Η απόκριση της απομακρυσμένης μηχανής σε αυτούς τους ελέγχους αποτελεί το αποτύπωμα του TCP/IP σωρού του απομακρυσμένου ΛΣ. Με τη σύγκριση του αποτυπώματος με μια βάση δεδομένων γνωστών αποτυπωμάτων, το NMAP είναι σε θέση να κάνει μια εικασία για το ΛΣ που τρέχει στον απομακρυσμένο host. Το TBIT διαφέρει από το «NMAP» από πολλές απόψεις. Ο στόχος του «NMAP» είναι να ανιχνευθεί το λειτουργικό σύστημα που τρέχει στον απομακρυσμένο host και όχι να χαρακτηριστεί η TCP συμπεριφορά. Κατά συνέπεια, η εξέταση «NMAP» δεν περιορίζεται στα TCP πακέτα μόνο. Πέρα από την αποτύπωση, το «NMAP» δεν συλλέγει καμία πληροφορία για τη TCP συμπεριφορά των απομακρυσμένων host. Έτσι πληροφορίες όπως η περιοχή τιμών ICW που παρατηρούνται στο Διαδίκτυο δεν μπορούν να ληφθούν χρησιμοποιώντας το «NMAP». Επίσης, όπως αναφέρεται στην παράγραφο 2, το «NMAP» χρησιμοποιεί έξω από τις συνηθισμένες σειρές πακέτων TCP/IP για αρκετούς από τους ελέγχους αποτυπωμάτων του, ενώ το TBIT χρησιμοποιεί μόνο τους κανονικούς χειρισμούς TCP μεταφοράς δεδομένων για να αποσπάσει τις πληροφορίες αυτές.

Θα μπορούσε κάποιος να υποστηρίξει ότι για να χαρακτηριστεί η TCP συμπεριφορά ενός απομακρυσμένου host, είναι αρκετό να ανιχνευθεί το ΛΣ που τρέχει στον host με τη χρήση ενός εργαλείου όπως το «NMAP». Η TCP συμπεριφορά μπορεί να αναλυθεί με τη μελέτη του ιδίου του ΛΣ, χρησιμοποιώντας είτε τον πηγαίο κώδικα (όταν διατίθεται), είτε τις πληροφορίες που παρέχονται από τον προμηθευτή (π.χ. ο διαδικτυακός τόπος της Microsoft προσφέρει τις πληροφορίες για το σωρό TCP/IP στο λειτουργικό σύστημα Windows), είτε με εργαστηριακά πειράματα [12]. Αρχικά, υποστηρίζεται ότι ο προσδιορισμός του ΛΣ του απομακρυσμένου host δεν είναι αρκετός, επειδή το πρότυπο του TCP πρωτοκόλλου καθορίζει διάφορες παραμέτρους, διαμορφούμενες από το επίπεδο εφαρμογής. Αυτές τίθενται διαφορετικά από εφαρμογή σε εφαρμογή και τα δεδομένα για αυτές τις παραμέτρους δεν μπορούν να ληφθούν μόνο με την αναγνώριση του ΛΣ ή με την ανάλυση του πηγαίου κώδικα. Δεύτερον, ανεξάρτητα από τους ισχυρισμούς του προμηθευτή, ο κώδικας του TCP μπορεί να περιέχει μικρολάθη [21] και ως εκ τούτου, η παρατηρηθείς συμπεριφορά μπορεί να είναι σημαντικά διαφορετική από τους ισχυρισμούς των προμηθευτών που φαίνεται στα σχετικά εγχειρίδια. Συνεπώς απαιτείται, άμεσος πειραματισμός είτε σε εργαστηριακά πειράματα είτε σε ολόκληρο το Διαδίκτυο των δημόσιων web εξυπηρετητών. Ενώ τα εργαστηριακά πειράματα είναι καλά για μια εξονυχιστική εξερεύνηση της συμπεριφοράς των σημαντικών, ευρέως διανεμημένων TCP υλοποιήσεων, δεν είναι πρακτικά για το χαρακτηρισμό ολόκληρου του φάσματος των TCP υλοποιήσεων στο δημόσιο Διαδίκτυο. Κατά συνέπεια, θεωρείται ότι το TBIT έχει συμπληρωματικό χαρακτήρα. Να βοηθήσει στην περαιτέρω ανάλυση, των εργαστηριακών πειραμάτων και των εργαλείων TCP αποτυπώσεων ΛΣ («*NMAP*»).

4. Η TCP συμπεριφορά των web εξυπηρετητών

Σε αυτή την παράγραφο, περιγράφονται μερικές από τις δοκιμές που υλοποιήθηκαν στο TBIT. Έχουν εξεταστεί οι TCP συμπεριφορές διάφορων web εξυπηρετητών χρησιμοποιώντας αυτές τις δοκιμές. Τα αποτελέσματα συμπεριλαμβάνονται επίσης μαζί με την περιγραφή κάθε δοκιμής. Η παράγραφος οργανώνεται ως εξής. Στην ενότητα 4.1, περιγράφεται σε συντομία το σύνολο web εξυπηρετητών που χρησιμοποιήθηκαν για τη δοκιμή. Στις ενότητες 4.2-4.7, περιγράφονται οι δοκιμές και παρέχονται τα αποτελέσματα των ερευνών. Τέλος, στην ενότητα 4.8, παρατίθεται επιχειρηματολογία των διάφορων παραγόντων που έχουν επιπτώσεις στα αποτελέσματα της δοκιμής.

4.1 Κεντρικοί υπολογιστές δικτύου που χρησιμοποιούνται για τη δοκιμή

Χρησιμοποιήθηκε έναν κατάλογος από 10.000 web εξυπηρετητές με στατικές IP διευθύνσεις. Δεν προβάλλεται κανένας ισχυρισμός για την αντιπροσωπευτικότητα αυτού του καταλόγου, εκτός από το ότι ο κατάλογος αυτός περιέχει κάποια επιλογή web εξυπηρετητών με υψηλή κυκλοφορία στο Διαδίκτυο. Χρησιμοποιήθηκε το «NMAP» [11] για να προσδιοριστούν τα λειτουργικά συστήματα που τρέχουν σε απομακρυσμένους web hosts.

4.2 Αρχική τιμή του παραθύρου συμφόρησης

Περιγράφεται η δοκιμή ICW στην παράγραφο 2. Η δοκιμή αυτή έτρεξε στον κατάλογο των web εξυπηρετητών που περιγράφηκε παραπάνω. Το MSS τέθηκε σε 100 bytes. Εξετάστηκε κάθε εξυπηρετητής τρεις φορές. Για να εξασφαλιστεί η ακρίβεια, θεωρήθηκαν τα αποτελέσματα μόνο από εκείνους τους εξυπηρετητές που εξετάστηκαν επιτυχώς τουλάχιστον δύο φορές και όλες οι επιτυχείς περιπτώσεις δοκιμής επέστρεψαν την ίδια απάντηση.

Τύπος	Αριθμός των web εξυπηρετητών
NewReno	1441
Reno	1154
TahoeNoFR	1024
Tahoe	251
Unidentified	47
Total	3917

Πίνακας 1: Τύπος αλγόριθμων ελέγχου συμφόρησης

Διαπιστώθηκε ότι 81% των web εξυπηρετητών θέτουν το ICW σε δύο τμήματα, ενώ 13,8% του συνόλου σε ένα και μόνο τμήμα. Μόνο 0,5% των web εξυπηρετητών θέτουν το ICW σε τέσσερα τμήματα, σύμφωνα με το [2]. Ένας μικρός αριθμός web εξυπηρετητών βρέθηκε να θέτει το ICW σε περισσότερα από 8000 bytes. Επαναλήφθηκε το πείραμα με MSS 512 bytes, οι οποίες επιβεβαίωσαν αυτές τις τάσεις. Τα αποτελέσματα του NMAP δείχνουν ότι πολλοί από τους web εξυπηρετητές που θέτουν ICW τους σε τέσσερα τμήματα τρέχουν μια beta έκδοση του λειτουργικού συστήματος Solaris 8. Οι web εξυπηρετητές που θέτουν το ICW σε 8000 bytes φαίνονται να τρέχουν διάφορες εκδόσεις του λειτουργικού συστήματος Digital (Compaq) Unix.

4.3 Αλγόριθμος ελέγχου συμφόρησης

Υπάρχει μια σειρά συμπεριφορών σε ότι αφορά τον έλεγχο συμφόρησης στις αναπτυχθείσες TCP υλοποιήσεις, συμπεριλαμβανομένων των Tahoe [13], Reno [3], NewReno [10] και το SACK [16], οι οποίοι χρονολογούνται από τα έτη 1988, 1990, 1996 και 1996 αντίστοιχα. Αυτές οι διαφορετικές παραλλαγές του ελέγχου συμφόρησης στο TCP περιγράφονται και αναλύονται με λεπτομέρεια στο [7]. Μια TCP σύνδεση δεν μπορεί να χρησιμοποιήσει την επιλογή SACK, εκτός αν και οι δύο κόμβοι είναι SACK enabled. Ελλείψη του SACK, οι μηχανισμοί ελέγχου συμφόρησης που χρησιμοποιούνται από έναν απομακρυσμένο host είναι πιθανό να είναι είτε Tahoe, είτε Reno, είτε NewReno. Οι διαφορετικές εκδόσεις του TCP μπορούν να έχουν σημαντικά διαφορετική απόδοση κάτω από ορισμένες συνθήκες απώλειας πακέτων. Αυτές οι διαφορετικές παραλλαγές TCP, δεν σημειώνονται στις επιγραφές των πακέτων. Ο μόνος τρόπος να καθοριστεί ποια χρησιμοποιείται από έναν συγκεκριμένο host, είναι να παρατηρηθεί μία TCP σύνδεσης που περιέχει απορρίψεις πακέτων, προκαλώντας την επιθυμητή συμπεριφορά. Η χρησιμοποίηση της δυνατότητας του TBIT να δημιουργεί τεχνητές απορρίψεις πακέτων, έχει βοηθήσει στην σχεδίαση μιας δοκιμής που διακρίνει τους μηχανισμούς ελέγχου συμφόρησης TCP μεταξύ των Tahoe, Reno και NewReno.

Η δοκιμή βασίζεται σε προσομοιώσεις που περιγράφονται στο [7].

Το TBIT εγκαθιστά μια σύνδεση με τον απομακρυσμένο web εξυπηρετητή, κατά τρόπο παρόμοιο με τη δοκιμή ICW που περιγράφεται στην παράγραφο 2. Στο MSS δίνεται μια μικρή τιμή (π.χ. 100 bytes) για να αναγκάσει τον απομακρυσμένο εξυπηρετητή να στείλει διάφορα πακέτα δεδομένων για τη δοκιμή, ακόμα κι αν η ζητούμενη ιστοσελίδα είναι μικρή σε μέγεθος. Το TBIT δηλώνει ένα παράθυρο παραλήπτη 5·MSS.

- Το TBIT ζητά την ιστοσελίδα βάσης.
- Ο απομακρυσμένος εξυπηρετητής αρχίζει να στέλνει την ιστοσελίδα βάσης στον πελάτη TBIT σε πακέτα των 100-bytes.
- Το TBIT αναγνωρίζει κάθε πακέτο σύμφωνα με το πρωτόκολλο TCP [22], έως ότου παραληφθεί το 13ο πακέτο.
- Το TBIT απορρίπτει αυτό το πακέτο, όπως φαίνεται στις δοκιμές και στα σχήματα 1 (α)-1 (γ).
- Το TBIT λαμβάνει και αναγνωρίζει τα πακέτα 14 και 15. Τα ACKs για αυτά τα πακέτα θα είναι διπλά ACKs για το πακέτο 12.
- Το πακέτο 16 απορρίπτεται. Όλα τα περαιτέρω πακέτα αναγνωρίζονται κατάλληλα.
- Το TBIT κλείνει τη σύνδεση μόλις παραλαμβάνονται 25 πακέτα δεδομένων, συμπεριλαμβανομένων των επαναμεταδόσεων.

Με βάση αυτό το ρεύμα 25 πακέτων, το TBIT μπορεί να καθορίσει τη συμπεριφορά ελέγχου συμφόρησης του απομακρυσμένου TCP. Το TCP NewReno χαρακτηρίζεται από γρήγορη επαναμετάδοση του πακέτου 13, καμία πρόσθετη γρήγορη επαναμετάδοση ή Timeout και καμία περιττή επαναμετάδοση του πακέτου 17, όπως φαίνεται στο σχήμα 1 (α). Το TCP Reno χαρακτηρίζεται από γρήγορη επαναμετάδοση του πακέτου 13, Timeout για το πακέτο 16 και καμία περιττή επαναμετάδοση του πακέτου 17 όπως φαίνεται στο σχήμα 1 (β). Το TCP Tahoe με γρήγορη επαναμετάδοση, σε αυτό το σενάριο χαρακτηρίζεται από κανένα Retransmit Timeout πριν από την επαναμετάδοση του πακέτου 13, αλλά από μια περιττή επαναναμετάδοση του πακέτου 17, όπως φαίνεται στο σχήμα 1 (γ). Για μια πιο λεπτομερή εξήγηση αυτής της συμπεριφοράς, παραπέμπεται ο αναγνώστης στο [7].

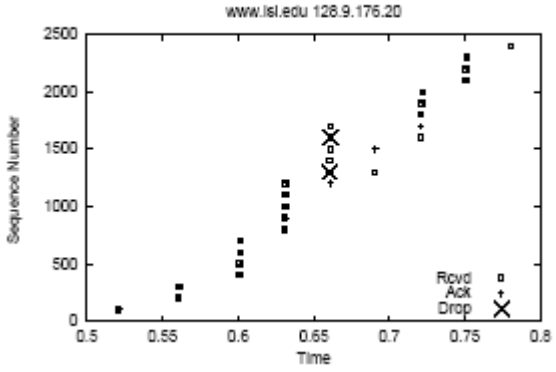
Για να εξασφαλιστεί η ακρίβεια των αποτελεσμάτων της δοκιμής, η δοκιμή τερματίζεται χωρίς επιστροφή οποιονδήποτε αποτελεσμάτων εάν κάποια πακέτα χαθούν πέρα από τα πακέτα που απορρίφθηκαν από το ίδιο το TBIT. Η ανεπιθύμητη απώλεια πακέτων μπορεί συνήθως να συναχθεί από απροσδόκητα χάσματα στην σειρά των ακολουθιακών αριθμών. Η δοκιμή τερματίζεται επίσης, εάν ο εξυπηρετητής δεν στείλει έναν ικανοποιητικό αριθμό πακέτων ακόμη και με μικρό MSS. Η δοκιμή μπορεί να επιστρέψει ανακριβή αποτελέσματα, εάν προκληθεί ένα timeout ή μια επαναμετάδοση λόγω σοβαρής απώλειας των ACKs που στέλνονται από το TBIT. Η ευρωστία σε σχέση με αυτά τα λάθη μπορεί να επιτευχθεί με το τρέξιμο της δοκιμής πολλαπλές φορές.

Εξετάζεται ο κατάλογος των εξυπηρετητών χρησιμοποιώντας αυτήν την δοκιμή. Το MSS τέθηκε για 100 bytes, για να εξασφαλίσει επαρκή αριθμό πακέτων για τη δοκιμή. Κάθε εξυπηρετητής εξετάστηκε τουλάχιστον τέσσερις φορές σε διαφορετικές χρονικές στιγμές. Για να εξασφαλιστεί η ακρίβεια των μετρήσεων, αναφέρονται τα αποτελέσματα για έναν web εξυπηρετητή μόνο εάν η δοκιμή ήταν επιτυχής τουλάχιστον τρεις φορές και η απάντηση που ήρθε σε όλες τις επιτυχείς περιπτώσεις ήταν η ίδια. Τα συσσωρευτικά αποτελέσματα παρουσιάζονται στον πίνακα 1.

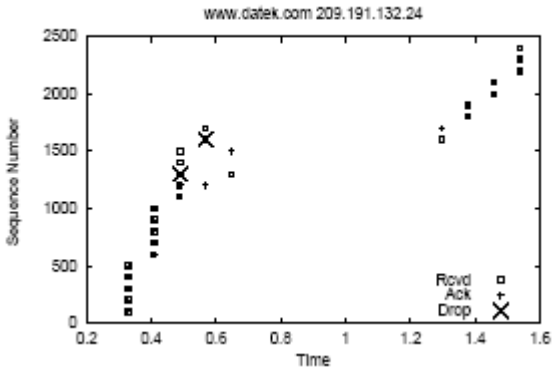
Η κύρια έκπληξη σε αυτά τα αποτελέσματα ήταν ο αριθμός των web εξυπηρετητών που ήταν κατηγοριοποιημένοι ως "Tahoe χωρίς Fast Retransmit", χαρακτηρισμένος από ένα Timeout επαναμετάδοσης για το πακέτο 13 και μια περιττή επαναμετάδοση του πακέτου 17, όπως φαίνεται στο σχήμα 1 (δ). Δεν αναμενότανε να βρεθούν υλοποιήσεις TCP που δεν χρησιμοποίησαν τη διαδικασία της γρήγορης επαναμετάδοσης, η οποία υπήρχε στις υλοποιήσεις του TCP από το 1988. Για το TCP χωρίς γρήγορη επαναμετάδοση, ο αποστολέας δεν συμπεραίνει μια απώλεια πακέτων από την παραλαβή τριών διπλών ACKs, αλλά πρέπει να περιμένει για να λήξει το χρονόμετρο επαναμετάδοσης πριν συμπεράνει την απώλεια και επαναμεταδόσει ένα πακέτο. Το σχήμα 1 (δ) παρουσιάζει ότι την τιμωρία απόδοσης στο χρήστη με την απουσία της γρήγορης επαναμετάδοσης.

Περισσότερο από το 70% των web εξυπηρετητών που ταξινομήθηκαν από τη δοκιμή σαν Tahoe χωρίς γρήγορη επαναμετάδοση, αναγνωρίστηκαν από το «NMAP» στις εκδόσεις των λειτουργικών τους συστημάτων ως Windows της Microsoft. Για να ερευνηθεί αυτή η συμπεριφορά περαιτέρω, αναπτύχθηκε η δοκιμή TBIT που επι-

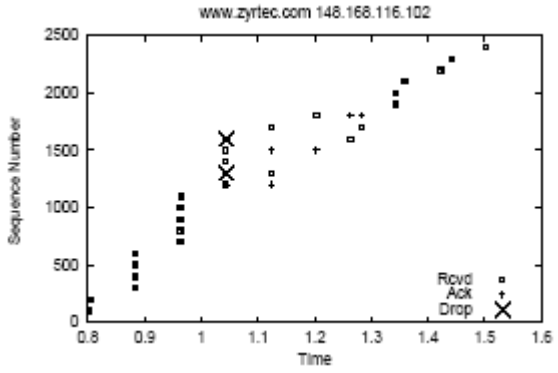
βεβαιώνει την απόκριση του web εξυπηρετητή σε ένα και μόνο πακέτο που απορρίφθηκε από ένα παράθυρο πέντε πακέτων και επαληθεύθηκε ότι οι περισσότεροι από αυτούς τους εξυπηρετητές δεν χρησιμοποιούν γρήγορη επαναμετάδοση, ακόμη και σε ένα σενάριο με ένα και μόνο πακέτο να απορρίπτεται. Η εταιρεία αναφέρει ότι θα διορθώσει το λάθος, το λειτουργικό σύστημα της επόμενης γενιάς έχει υποσχεθεί ένα patch για να διορθώσει το πρόβλημα στα Windows 2000. Πάντως, κατά την διάρκεια του γραψίματος του σχετικού εγγράφου, το patch δεν ήταν διαθέσιμο.



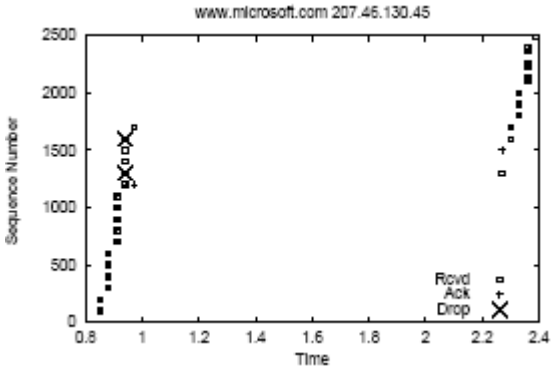
Σχήμα 1 (α): Παράδειγμα συμπεριφοράς ελέγχου συμφόρησης στο New Reno



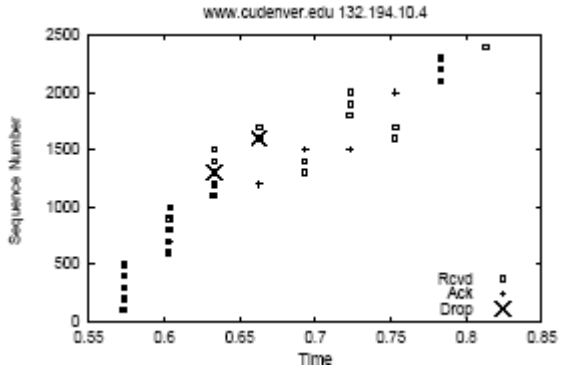
Σχήμα 1 (β): Παράδειγμα συμπεριφοράς ελέγχου συμφόρησης στο Reno



Σχήμα 1 (γ): Παράδειγμα συμπεριφοράς ελέγχου συμφόρησης στο Tahoe με Fast Retransmit



Σχήμα 1 (δ): Παράδειγμα συμπεριφοράς ελέγχου συμφόρησης στο Tahoe χωρίς Fast Fast Retransmit

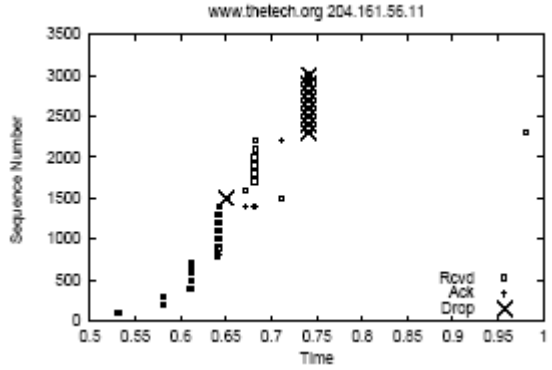


Σχήμα 2: Δύο περιπτές επαναμεταδόσεις

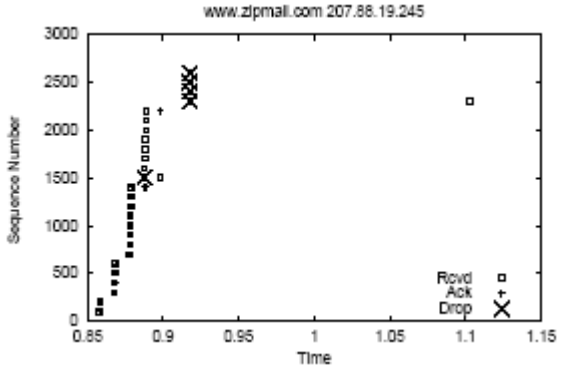
Χρησιμοποιώντας το «NMAP», διαπιστώνεται ότι πάνω από 78% των συστημάτων που αναγνωρίστηκαν από το TBIT να χρησιμοποιούν το NewReno τρέχουν τις νεώτερες εκδόσεις των λειτουργικών συστημάτων Linux (2.1 ή νεότερων) και Solaris (2.6 ή νεότερων). Σχεδόν 60% των συστημάτων που αναφέρουν την παλαιότερη συμπεριφορά Reno φαίνονται να τρέχουν τις

διάφορες εκδόσεις FreeBSD και BSDI. Πολλά από τα άλλα φαίνονται να τρέχουν τις διάφορες εκδόσεις των λειτουργικών συστημάτων Windows, αλλά με μεγάλες ιστοσελίδες βάσης. Πάνω από 67% των συστημάτων που αναφέρουν τη συμπεριφορά Tahoe, φαίνονται να τρέχουν τη έκδοση Linux (2.2 και προγενέστερη).

Οι περισσότεροι (66%) των web εξυπηρετητών που ανήκουν στη *"μη αναγνωρισμένη"* κατηγορία χρησιμοποιούν την γρήγορη επαναμετάδοση για το πακέτο 13 και επαναμεταδίδουν αχρείαστα το πακέτο 14, καθώς επίσης και μια περιττή επαναμετάδοση του πακέτου 17, αλλά κανένα Timeout επαναμετάδοσης. Ένα παράδειγμα αυτής της συμπεριφοράς παρουσιάζεται στο σχήμα 2.



Σχήμα 3 (α): Χωρίς ελάττωση παραθύρου



Σχήμα 3 (β): Με ελάττωση παραθύρου σε τέσσερα τμήματα

Πίνακας 2. Αναγωγή παράθυρου μετά από μια απώλεια πακέτου, από ένα παράθυρο οκτώ τμημάτων.

4.4 Συμμόρφωση στον Έλεγχο συμφόρησης

Ένας αποστολέας TCP αναμένεται να διχοτομήσει το παράθυρο συμφόρησής του μετά από μια απώλεια πακέτων. Αυτή η πτυχή της συμπεριφοράς TCP είναι το κλειδί στη σταθερότητα του Διαδικτύου [9]. Έτσι, αναπτύχθηκε μια δοκιμή του TBIT που ελέγχει αυτήν την συμπεριφορά. Η δοκιμή πραγματοποιείται ως εξής.

- Το TBIT εγκαθιστά μια σύνδεση με τον απομακρυσμένο εξυπηρετητή, χρησιμοποιώντας ένα μικρό MSS και ζητά την ιστοσελίδα βάσης.
- Το TBIT αναγνωρίζει όλα τα πακέτα έως ότου παραληφθεί το πακέτο 15. Εάν το απομακρυσμένο TCP έχει επιδείξει τη σωστή συμπεριφορά αργής εκκίνησης (slowstart), το παράθυρο συμφόρησης πρέπει να είναι τουλάχιστον οκτώ τμήματα αυτή τη στιγμή. Το TBIT απορρίπτει το Πακέτο 15.
- Το TBIT αναγνωρίζει όλα τα πακέτα κατάλληλα, στέλνοντας διπλά ACKs, αναγνωρίζοντας το πακέτο 14, μέχρι το πακέτο 15 να επαναμεταδοθεί. Η επαναμετάδοση αναγνωρίζεται κατάλληλα. Μετά από αυτό, το TBIT δεν αναγνωρίζει άλλα πακέτα. Αυτό θα αναγκάσει τελικά τον απομακρυσμένο εξυπηρετητή σε time out και θα επαναμεταδώσει το πρώτο μη αναγνωρισμένο πακέτο.
- Μόλις το TBIT ανιχνεύσει αυτήν την επαναμετάδοση, κλείνει τη σύνδεση και τερματίζει τη δοκιμή.

Το μέγεθος του μειωμένου παραθύρου συμφόρησης, σε bytes, είναι η διαφορά μεταξύ του μέγιστου αριθμού σειράς που παραλαμβάνεται από το TBIT και του υψηλότερου αριθμού σειράς που αναγνωρίστηκε από το TBIT. Συγκρίνοντάς το με το μέγεθος του παραθύρου συμφόρησης πριν από τη μείωση (8 τμήματα), μπορεί να αποφασιστεί εάν το απομακρυσμένο TCP χρησιμοποιεί τον σύμφωνο έλεγχο συμφόρησης.

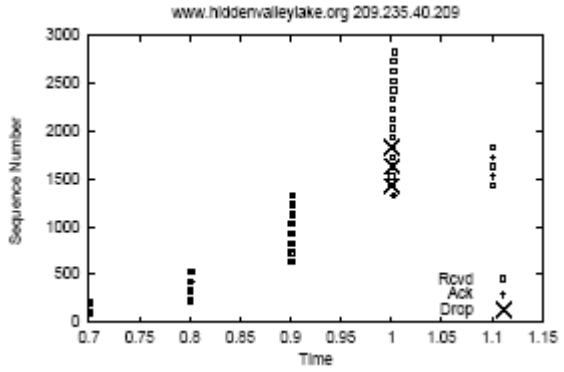
Τα ζητήματα ευρωστίας που περιλαμβάνονται σε αυτήν την δοκιμή είναι παρόμοια με εκείνα που αναφέρθηκαν στην παράγραφο 4.3 και όταν αντιτέθηκε στη δοκιμή το σύνολο των στόχων, πάρθηκαν παρόμοιες προφυλάξεις (δηλαδή εξέταση κάθε host τέσσερις φορές κ.λ.π.). Τα συσσωρευτικά αποτελέσματα που παρουσιάζο-

Παράθυρο μετά την απώλεια	**Αριθμός των web εξυπηρετητών**
5 τμήματα ή λιγότερα	3757
Περισσότερα από 5 τμήματα	213
Σύνολο	3970

νται στον πίνακα 2 και ένα αντιπροσωπευτικό παράδειγμα κάθε κατηγορίας εμφανίζεται στο σχήμα 3. Από τους 44 κεντρικούς υπολογιστές που δεν μείωσαν το παράθυρο συμφόρησής τους σε πέντε τμήματα ή λιγότερα, οι πιο πολλοί αναγνωρίστηκαν από το «NMAP» να τρέχουν ένα παλαιότερο σύστημα Solaris 2.5 ή 2.51. Μετά από επαφή με ανθρώπους της Sun, οι οποίοι εξέτασαν τον κώδικα και ανέφεραν ότι η συμπεριφορά οφειλόταν σε ένα λάθος στο σωρό του TCP, όπου προστίθενται τρία τμήματα στο παράθυρο συμφόρησης μετά από την διχοτόμηση του από μια γρήγορη επαναμετάδοση. Δεν παρατηρήθηκε αυτό το πρόβλημα σε καμιά από τις πιο πρόσφατες εκδόσεις αυτού του λειτουργικού συστήματος.

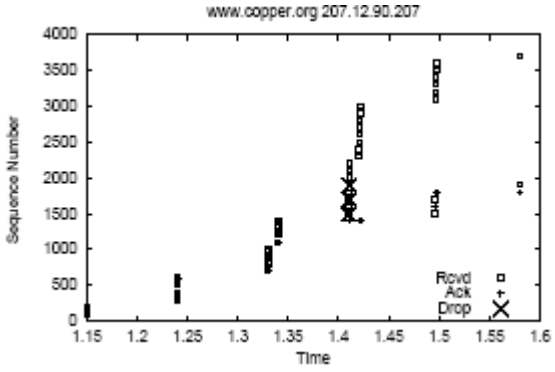
4.5 Απόκριση στις επιλεκτικές αναγνωρίσεις

Διάφοροι σωροί TCP έχουν υλοποιήσει την επιλογή **Επιλεκτικής Αναγνώρισης (Selective Acknowledgment)** TCP (SACK) [16]. Είναι δυνατό να καθοριστεί από τα παθητικά ίχνη εάν ένα απομακρυσμένο TCP υποστηρίζει την επιλογή TCP SACK απλά με την παρατήρηση αν το πακέτο TCP SYN περιλαμβάνει την επιλογή SACK_PERMITTED [1]. Πάντως, με την χρησιμοποίηση του παθητικού ελέγχου είναι δύσκολο να καθοριστεί εάν το απομακρυσμένο TCP χρησιμοποιεί πραγματικά τις πληροφορίες που περιλαμβάνονται στα SACKs που στέλνονται από τον αποστολέα. Έχει σχεδιαστεί η ακόλουθη δοκιμή TBIT για να ελεγχθεί αυτό.

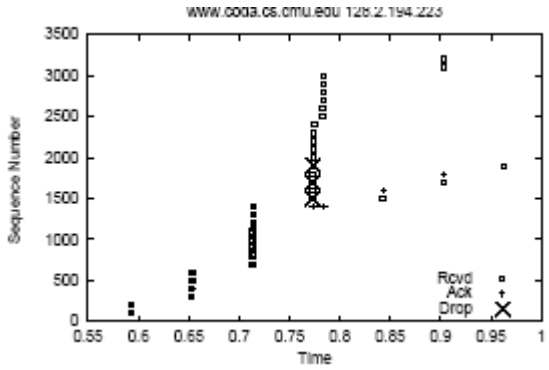


Αποτέλεσμα	Αριθμός των web εξυπηρετητών
Επαληθευθείσα χρήση SACK	440
Μη επαληθευθείσα χρήση SACK	1037
Σύνολο	1477

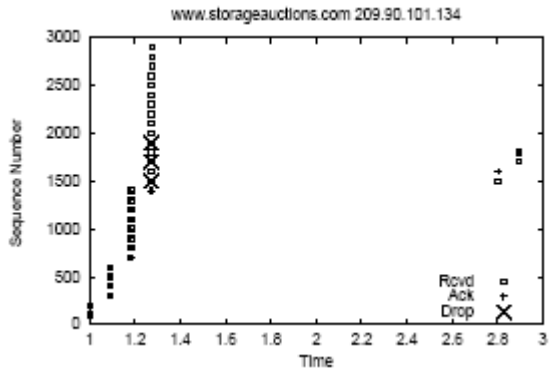
Σχήμα 4 (α): Επαναναποστολές σε ένα RTT: Προαιρετική χρήση SACK



Σχήμα 4 (β): Επαναναποστολές σε δύο RTTs: Χρήση SACK



Σχήμα 4 (γ): Συμπεριφορά σαν New Reno: Χωρίς χρήση SACK



Σχήμα 4 (δ): Tahoe χωρίς γρήγορη επαναποστολή

Πίνακας 3. Δοκιμή SACK

Η ιδανική συμπεριφορά ενός αποστολέα SACK_enabled θα ήταν να ξαναστείλει τα πακέτα 15, 17 και 19 σε ένα και μόνο RTT και να μην στείλει οποιεσδήποτε περιττές επαναμεταδόσεις. Αυτή η συμπεριφορά είναι αρκετά διαφορετική από αυτήν ενός παραλήπτη NewReno, ο οποίος θα πάρει τουλάχιστον τρεις χρόνους round trip για να στείλει όλες τις επαναμεταδόσεις.

Τα ζητήματα ευρωστίας που περιλαμβάνονται σε αυτήν την δοκιμή είναι παρόμοια με εκείνα που αναφέρθηκαν στην παράγραφο 4.3. Πριν πραγματοποιηθεί αυτή η δοκιμή στον κατάλογό των web εξυπηρετητών πρώτα ελέγχθηκαν, χρησιμοποιώντας μια άλλη απλή δοκιμή TBIT, για να φανεί ποιοι web εξυπηρετητές ήταν SACK enabled. Αυτή η δοκιμή συνίσταται στην αποστολή ενός SYN με την επιλογή SACK PERMITTED και την εξέταση του επιστρεφόμενου SYN/ACK. Με βάση αυτήν την δοκιμή, προσδιορίστηκε ότι από το αρχικό σύνολο, 2953 web εξυπηρετητές ήταν SACK_enabled. Τα αποτελέσματα που παρουσιάζονται στον πίνακα 3, βασίζονται σε αυτό το υποσύνολο. Η τιμή MSS τέθηκε στα 100 bytes. Κάθε host εξετάστηκε τρεις φορές και τα αποτελέσματα συμπεριλαμβάνονται στην καταγραφή μόνο εάν η δοκιμή ήταν επιτυχής τουλάχιστον δύο φορές και όλες οι επιτυχείς περιπτώσεις επέστρεψαν την ίδια απάντηση.

Η συμπεριφορά που φαίνεται στο σχήμα 4 (α) αντιπροσωπεύει τη βέλτιστη χρήση των πληροφοριών SACK. Ο αποστολέας TCP επαναμεταδίδει και τα τρία πακέτα σε έναν και μόνο χρόνο round-trip και δεν επαναμεταδίδει κάποια πακέτα άσκοπα. Τα αποτελέσματα NMAP δείχνουν ότι οι περισσότεροι από τους hosts που εμφανίζουν αυτόν τον τύπο συμπεριφοράς τρέχουν νεώτερες εκδόσεις των λειτουργικών συστημάτων Linux (2.2.13) ή Solaris (2.6 ή νεότερο).

Η συμπεριφορά που φαίνεται στο σχήμα 4 (β) είναι ελαφρά λιγότερο βέλτιστη, καθώς ο αποστολέας παίρνει δύο round trip χρόνους για να επαναμεταδώσει τα χαμένα πακέτα, αλλά ο αποστολέας TCP κάνει ακόμα σαφή χρήση των πληροφοριών SACK. Ο αποστολέας δεν επαναμεταδίδει κάποια πακέτα άσκοπα. Αυτός ο τύπος συμπεριφοράς εμφανίζεται συνήθως από τους hosts που τρέχουν τις διάφορες εκδόσεις του λειτουργικού συστήματος Windows 2000 και έχουν ελαφρά μεγάλες σελίδες βάσης. Οι αποστολείς που φαίνονται στην πρώτη σειρά του πίνακα 3, εμφανίζουν μια από αυτές τις δύο συμπεριφορές.

Στο σχήμα 4 (γ), ο αποστολέας φαίνεται να παίρνει τρεις χρόνους round trip για να τελειώσει τις επαναμεταδόσεις. Αυτή είναι η αναμενόμενη συμπεριφορά από έναν αποστολέα NewReno. Δεν υπάρχει καμία ένδειξη ότι ο αποστολέας TCP κάνει κάποια χρήση των πληροφοριών στα πακέτα SACK. Τα αποτελέσματα NMAP δείχνουν ότι οι περισσότεροι από τους hosts που εμφανίζουν αυτόν τον τύπο συμπεριφοράς, τρέχουν διάφορες εκδόσεις του λειτουργικού συστήματος Linux.

Τέλος, στο σχήμα 4 (δ), φαίνεται ένας αποστολέας που φαίνεται να αγνοεί όλες τις πληροφορίες SACK. Αυτό γίνεται επειδή ο αποστολέας χρησιμοποιεί ένα Timeout επαναμετάδοσης, αντί για γρήγορη επαναμετάδοση, για να επαναμεταδώσει το πακέτο 15. Ένας αποστολέας TCP πρέπει για να απορρίψει τις πληροφορίες που λαμβάνονται από τα blocks SACK μετά από ένα Timeout επαναμετάδοσης [16]. Οι hosts που εμφανίζουν αυτήν την συμπεριφορά φαίνονται να τρέχουν διάφορες εκδόσεις των λειτουργικών συστημάτων Windows της Microsoft και φαίνονται να έχουν μικρές σελίδες βάσης. Αυτή η αποτυχία να χρησιμοποιηθεί η γρήγορη επαναμετάδοση αναφέρθηκε στην παράγραφο 4.3.

4.6 Απόκριση σε ECN

Η **Ρητή Γνωστοποίηση Συμφόρησης - ΡΓΣ (Explicit Congestion Notication -ECN)** [23] είναι ένας μηχανισμός για να επιτρέψει στους δρομολογητές να χαρακτηρίσουν τα πακέτα TCP ως προς τη συμφόρηση, αντί της απόρριψης των, όταν είναι δυνατό. Ενώ οι δρομολογητές που έχουν την δυνατότητα ECN δεν έχουν ακόμα αναπτυχθεί ευρέως, οι πιο πρόσφατες εκδόσεις του λειτουργικού συστήματος Linux περιλαμβάνουν πλήρη υποστήριξη ECN. Μετά από αυτήν την ανάπτυξη των ECN - enabled τελικών κόμβων, υπήρξαν αρκετά παράπονα ότι οι hosts που έχουν την δυνατότητα ECN, δεν μπορούσαν να έχουν πρόσβαση σε διάφορους διαδικτυακούς τόπους [14]. Γράφτηκε μια δοκιμή TBIT που ερευνά εάν πακέτα που έχουν την δυνατότητα ECN απορρίπτονται από δημοφιλείς web εξυπηρετητές ή δρομολογητές κατά μήκος της πορείας τους.

Η δημιουργία μιας TCP σύνδεσης που έχει την δυνατότητα ECN αποτελείται από μια χειραψία μεταξύ του αποστολέα και του παραλήπτη. Αυτή η διαδικασία περιγράφεται λεπτομερώς στο [23]. Εδώ παρέχεται μόνο μια συνοπτική περιγραφή των πτυχών ECN για τις οποίες υπάρχει ενδιαφέρον. Ένας πελάτης που έχει την δυνατότητα ECN θέτει τα flags ECN_ECHO και CWR στην κεφαλίδα του πακέτου SYN. Αυτό καλείται **ECN-setup SYN**. Εάν ο εξυπηρετητής έχει την δυνα-

τότητα ECN, θα αποκριθεί με τη ρύθμιση ECN_ ECHO flag στο SYN - ACK. Από εκείνο το σημείο και μετά, όλα τα πακέτα που ανταλλάσσονται μεταξύ των δύο hosts, μπορούν να έχουν την δυνατότητα ECN θέτοντας το (ECT) bit στην κεφαλίδα IP. Εάν ένας δρομολογητής κατά μήκος της πορείας επιθυμεί να χαρακτηρίσει ένα τέτοιο πακέτο ως ένδειξη της συμφόρησης, το κάνει με τη ρύθμιση του **Congestion Experienced (CE)** bit στην κεφαλίδα IP του πακέτου.

Ο στόχος της δοκιμής είναι να ανιχνευθεί ο ελαττωματικός εξοπλισμός που οδηγεί στην άρνηση της πρόσβασης σε ορισμένους web εξυπηρετητές από κόμβους (end nodes) που έχουν την δυνατότητα ECN. Η δοκιμή δεν προορίζεται να ελέγξει την πλήρη συμμόρφωση στα πρότυπα ECN [23].

1. Το TBIT κατασκευάζει ένα ECN-setup πακέτο SYN και το στέλνει στον απομακρυσμένο web εξυπηρετητή.
2. Εάν το TBIT λάβει ένα SYN - ACK από τον απομακρυσμένο host, το TBIT προχωρά στο βήμα 4.
3. Εάν κανένα SYN - ACK δεν παραλαμβάνεται μετά από τρεις επαναπροσπάθειες (τρόπος αποτυχίας 1), ή εάν ένα πακέτο με RST παραληφθεί (τρόπος αποτυχίας 2), το TBIT καταλήγει στο συμπέρασμα ότι ο απομακρυσμένος εξυπηρετητής αναφέρει μια αποτυχία. Η δοκιμή ολοκληρώνεται.
4. Το TBIT ελέγχει για να δει εάν το SYN - ACK ήταν ένα ECN-setup SYN - ACK, με το ECN_ ECHO ενεργοποιημένο και το CWR flag απενεργοποιημένο (unset). Εάν αυτό συμβεί, κατόπιν ο απομακρυσμένος web εξυπηρετητής διαπραγματεύεται τη χρήση ECN. Διαφορετικά, ο απομακρυσμένος εξυπηρετητής δεν έχει την δυνατότητα ECN.
5. Αγνοώντας εάν ο απομακρυσμένος web εξυπηρετητής, διαπραγματεύθηκε την χρήση ECN, το TBIT στέλνει ένα πακέτο δεδομένων που περιέχει ένα έγκυρο αίτημα HTTP, με τα bits ECT και CE ενεργοποιημένα στην κεφαλίδα IP.
6. Εάν ένα ACK παραληφθεί, ελέγχει για να δει εάν είναι ενεργοποιημένο το ECN_ ECHO flag. Εάν κανένα ACK δεν παραληφθεί μετά από τρεις προσπάθειες, ή εάν το προκύπτον ACK δεν έχει το ECN_ ECHO flag ενεργοποιημένο (τρόπος αποτυχίας 3), το TBIT καταλήγει στο συμπέρασμα ότι ο απομακρυσμένος web εξυπηρετητής δεν υποστηρίζει το ECN σωστά.

Διάρκεια	**Αριθμός των web εξυπηρετητών**
No wait	2120
0 < 2*MSL < 64	3714
64 < 2*MSL < 128	150
128 < 2*MSL < 192	121
192 < 2*MSL < 256	1020
2*MSL > 320	22
Σύνολο	24030

Πίνακας 4.Διάρκεια του χρόνου αναμονής

Για να εξασφαλιστεί η ευρωστία, πριν τρέξει η δοκιμή ελέγχεται ότι ο απομακρυσμένος web εξυπηρετητής είναι εφικτός από τον διαδικτυακό τόπο που τρέχει η δοκιμή και θα αναγνωρίσει ένα πακέτο SYN που στέλνεται χωρίς τα flags ECN_ ECHO και CWR να είναι ενεργοποιημένα. Η ευρωστία σε σχέση με την απώλεια πακέτων εξασφαλίζεται από την επαναμετάδοση ενός SYN ή ενός πακέτου δεδομένων όπως αναφέρθηκε στα βήματα 4 και 6.

Χρησιμοποιήθηκε ένα μεγαλύτερο σύνολο hosts (περίπου 27.000) για να διεκπεραιωθεί αυτή η δοκιμή. Αυτό το σύνολο περιέχει περισσότερους από 10.000 hosts που χρησιμοποιούνται για άλλες δοκιμές που περιγράφηκαν σε αυτή την παράγραφο. Ο λόγος της χρησιμοποίησης ενός διαφορετικού συνόλου για αυτήν την δοκιμή είναι ότι γίνεται προσπάθεια να ερευνηθεί το πρόβλημα που αναφέρθηκε στο [14]. Τα συσσωρευτικά συμπεράσματα αναφέρονται στον πίνακα 4. Η πρώτη σειρά εμφανίζει τους hosts που δεν υποστηρίζουν ECN, αλλά αλληλεπιδρούν σωστά με τους πελάτες που υποστηρίζουν ECN. Η δεύτερη και τρίτη σειρά παριστάνουν τους hosts που αρνούνται την πρόσβαση σε πελάτες που έχουν την δυνατότητα ECN. Η τέταρτη σειρά παριστάνει τους hosts που διαπραγματεύονται την υποστήριξη ECN, αλλά αποτυγχάνουν να αποκριθούν σε bits CE που τίθενται στα πακέτα δεδομένων. Αυτές οι τρεις περιπτώσεις, τρόποι αποτυχίας 1 μέχρι 3, είναι υλοποιήσεις που *«χτύπησαν»* και πρέπει να διορθωθούν. Η τέταρτη σειρά παριστάνει τους hosts που φαίνονται να υποστηρίζουν το ECN σωστά.

Τα αποτελέσματα «NMAP» έδειξαν ότι οι περισσότεροι hosts με τον τρόπο αποτυχίας 2 ήταν

πίσω από Cisco Localdirector 430 [5], το οποίο είναι ένας proxy ισορροπίας φορτίου. Αυτό το πρόβλημα υπέπεσε στην αντίληψη της Cisco και από τότε έχει παρασχεθεί μια διόρθωση. Οι περισσότεροι hosts με τον τρόπο αποτυχίας 1 φαίνονται να τρέχουν μια έκδοση του λειτουργικού συστήματος AIX. Οι άνθρωποι της IBM λειτουργούν στο πρόβλημα. Επίσης θεωρείται ότι μερικές από αυτές τις αποτυχίες οφείλονται στα firewalls που μπερδεύουν τη χρήση των flags που σχετίζονται με το ECN στο TCP, με μια υπογραφή του εργαλείου ανίχνευσης θυρών [18]. Οι περισσότεροι από τους hosts με τον τρόπο αποτυχίας 3 φαίνονται να τρέχουν παλαιότερες εκδόσεις Linux (Linux 2.0.27-34). Από τους 22 hosts στη τέταρτη σειρά, για τη διαπραγμάτευση ECN και τη σωστή χρησιμοποίηση ECN, οι 18 ανήκουν σε ένα ενιαίο υποδίκτυο (subnet). Το NMAP δεν θα μπορούσε να προσδιορίσει τα λειτουργικά συστήματα που τρέχουν σε αυτούς τους 18 hosts. Από τους υπόλοιπους τέσσερις, τρεις φαίνονται να τρέχουν τις νεώτερες εκδόσεις του Linux (2.1-2.2.13).

Αναφορές

[1] M. Allman. A web server's view of the transport layer, June 2000. http://roland.grc.nasa.gov/ mallman/tcp-opt-deployment/.

[2] M. Allman, S. Floyd, and C. Partridge. Increasing TCP's initial window, September 1998. RFC2414.

[3] M. Allman, V. Paxson, and W. Stevens. TCP congestion control, April 1999. RFC2581.

[4] N. Cardwell, S. Savage, and T. Anderson. Modeling TCP latency. In Proc. IEEE INFOCOM, 2000.

[5] Cisco Systems. How to cost-effectively scale web servers. Packet Magazine, Third Quarter 1996. http://www.cisco.com/warp/public/784/5.html .

[6] K. Claffy, G. Miller, and K. Thompson. The nature of the beast: recent traffic measurements from an Internet back-bone. In Proceedings of INET'98, 1998. http://www.caida.org/outreach/papers/Inet98/ .

[7] K. Fall and S. Floyd. Simulation-based comparisons of Tahoe, Reno, and SACK TCP Computer Communication Review, 26(3), July 1996.

[8] K. Fall and K. Varadhan. ns: Manual, February 2000.

[9] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the Internet. IEEE/ACM Trans. Networking, August 1999.

[10] S. Floyd and T. Henderson. The NewReno modification to TCP's fast recovery algorithm, April 1999. RFC2582.

[11] Fyodor. Remote os detection via tcp/ip stack fingerprinting. Phrack 54, 8, Dec. 1998. URL http://www.insecure.org/nmap/nmap fingerprinting-aticle.hml.

[12] T.Gao and J.Mahdavi. On current TCP/IP implementations and performance testing, August 2000. Unpublished manuscript.

[13] V. Jacobson. Congestion avoidance and control. Computer Communication Review, 18(4), August 1988.

[14] D. Kelson, September 2000. note sent to Linux kernel mailing list.

[15] B. Krishnamurthy and J. Rexford. Web Protocols and Practice: http/1.1, Networking Protocols, Caching, and Traffic Measurement. Addison Wesley, 2001.

[16] M. Mathis, J. Mahdavi, S. Floyd, and A. Romonow. TCP selective acknowledgement options, October 1996. RFC2018.

[17] S. McCanne and V. Jacobson. The BSD packet filter: A new architecture for user-level packet capture. In Proceedings of the winter USENIX technical conference, January 1993.

[18] T. Miller. Intrusion detection level analysis of nmap and queso, 2000.

[19] K. Park, G. Kim, and M. Crovella. On the relationship between le sizes, transport protocols and self-similar network traffic. In Proc. International Conference on Network Protocols, 1996.

[20] V. Paxson. End-to-end Internet packet dynamics. In Proc. ACM SIGCOMM, 1997.

[21] V. Paxson, M. Allman, S. Dawson, W. Fenner, J. Griner, I. Heavens, K. Lahey, J. Semke, and B. Volz. Known TCP implementation problems, March 1999. RFC2525.

[22] J. Postel. Transmission control protocol, September 1981. RFC793.

[23] K. K. Ramakrishnan and S. Floyd. A proposal to add explicit congestion notification (ECN) to IP, January 1999. RFC2481.

[24] L. Rizzo. Dummynet and forward error correction. In Proc. Freenix, 1998.

[25] S. Savage. Sting: a TCP-based network measurement tool. Proceedings of the USENIX Symposium on Internet Technologies and Systems, pages 71-79, Oct. 1999.

[26] W. Stevens. TCP/IP Illustrated, Vol.1 The Protocols. Addison-Wesley, 1997. 10th printing.

[27] K.Thompson, G. Miller, and M. Wilder. Wide-area internet traffic patterns and characteristics. IEEE Network Magazine, 11(6), November 1997.