הפרוטוקול התקשרות המדובר עובד כדלהלן:

**סביבת השרת:**

השרת מופעל ומצפה לקבל בקשה מהקליינט.

לכשיתקבל בקשה מהקליינט, יווצר חיבור בין השרת ללקוח, ה- recvBuff של בקשה מהקליינט יכיל בקשה שיכולה להיות PUT, GET, DELETE, TRACE, HEAD . השרת יחזיר תשובה\יבצע פעולה בהתאם לבקשת הלקוח (אם עברו יותר מ-2 דקות ייסגר הקשר עם הלקוח)

*קלטיים אפשריים ותוצאות:*

*HEAD -*  במקרה כזה תופעל הפונקציה SEND כאשר sendSubType יהיה HEAD ולכן

ישלח ללקוח פירוט על שדות ה-HEAD של הקובץ אותו ביקש, הכוללים

תאריך, אורך הטקסט, קוד שגיאה והודעת שגיאה.

*GET-* במקרה כזה תופעל הפונקציה SEND כאשר sendSubType יהיה GET ולכן

ישלח ללקוח פירוט על שדות ה-GET של הקובץ אותו ביקש, הכוללים

תאריך, אורך הטקסט, קוד שגיאה והודעת שגיאה ובנוסף ישלח התוכן

של הקובץ המבוקש.

*DELETE -* במקרה כזה תופעל הפונקציה SEND כאשר sendSubType יהיה DELETE

ולכן ישלח ללקוח פירוט על שדות ה-DELETE של הקובץ אותו ביקש,

הכוללים תאריך, אורך הטקסט, קוד שגיאה והודעת שגיאה ובנוסף אם

הקובץ קיים השרת ידאג למחוק אותו מהמאגר, אחרת יתריע שאין קובץ

כזה.

הערות עבור – & DELETE HEAD & GET - אם הבקשה לא הצליחה, זה נבע כתוצאה מבקשה של קובץ שאינו קיים. לכן הוחזר במקרים אלו קוד שגיאה 404. אם הקובץ קיים הוחזר הקוד 200.

*PUT –*  במקרה כזה תופעל הפונקציה SEND כאשר sendSubType יהיה PUT

ולכן ישלח ללקוח פירוט על שדות ה-PUT של הקובץ אותו ביקש,

הכוללים תאריך, אורך הטקסט, קוד שגיאה והודעת שגיאה, אם הקובץ

הושם בשרת יוחזר קוד 200, אחרת יוחזר קוד 400 ש אומר שהקובץ לא

הושם במערכת.

הערות עבור – PUT - אם הבקשה לא הצליחה, אזי הקובץ לא הוכנס לשרת. לכן הוחזר במקרים אלו

קוד שגיאה 400. אם הקובץ קיים הוחזר הקוד 200.

*TRACE-* במקרה כזה תופעל הפונקציה SEND כאשר sendSubType יהיה TRACE

ולכן ישלח ללקוח פירוט על שדות ה-TRACE של הקובץ אותו ביקש,

הכוללים תאריך, אורך הטקסט, קוד שגיאה והודעת שגיאה. נשים לב ש-

TRACE למעשה עושה ECHO לבקשת הקליינט ומחזירה את השדות לפי

הבקשה.

**הערות לגבי השרת:**

השרת מבצע בדיקה אם מחכה בקשת קליינט כלשהי, כמו כן השרת תומך עד 60 קליינטים במקביל. השרת יוצר סביבת התקשרות על TCP אמינה ובטוחה.

**סביבת הקליינט:**

הקליינט שולח בקשות לשרת. אם לאחר שתי דקות הקלינט לא מקבל תשובה הקשר יסגר. הקשר שנוצר הינו כאשר בטוח ואמין ע"ג TCP פרוטוקול. הקליינט בוחר איזה מידע/פעולה ברצונו לבקש מהשרת, חשוב להבחין כי בסוף כל מימוש פעולה מצד השרת הסוקט האחראי לקליינט נסגר וכך גם סביבת העבודה של הסוקט בווינדוס.

לכשתתקבל קלט מהמשתמש הקליינט יפנה לשרת ויקבל מידע בהתאם לבקשת המשתמש.

קלטיים אפשריים ותוצאות:

*HEAD* - במקרה זה הלקוח שולח בקשה לקבלת HEAD של קובץ מסוים מהשרת, ע"י

שימוש ב- SendBuff. את המידע הוא יחלץ ע"י הפונקציה Recv. הקליינט

יקבל את שדות המידע של קובץ זה כגון: האם הקובץ קיים, קוד ומשפט

שגיאה (מאפשר ניתוב האם הבקשה צלחה או לא), אורך התוכן של הקובץ

שהתבקש. נבחין כי HEAD לא מחזיר את התוכן עצמו! .

*GET* - במקרה זה הלקוח שולח בקשה לקבלת GET של קובץ מסוים מהשרת, ע"י

שימוש ב- SendBuff. את המידע הוא יחלץ ע"י הפונקציה Recv. הקליינט

יקבל את שדות המידע של קובץ זה כגון: האם הקובץ קיים, קוד ומשפט

שגיאה (מאפשר ניתוב האם הבקשה צלחה או לא), אורך התוכן של הקובץ

שהתבקש. נבחין כי במידה והקובץ קיים, ישלח בנוסף גם תוכן הקובץ

(להבדיל מ – HEAD)

*DELETE* - במקרה זה הלקוח שולח בקשה לקבלת Delete של קובץ מסוים מהשרת, ע"י

שימוש ב- SendBuff . לאחר פעולות השרת, יוחזר תגובה לקלינט וע"י

שימוש ב – Recv הקליינט יוכל לדעת אם הקובץ באמת נמחק (לפי הודעת

השגיאה).

PUT – במקרה כזה הלקוח שולח בקשה לשרת להוסיף קובץ לשרת. הלקוח מספק את

שם הקובץ ואת המיקום היחסי להוסיף את הקובץ בשרת, כמו כן הלקוח

מחליט מה יהיה תוכן השרת. שליחת הבקשה תתבצע ע"י SendBuff . לאחר

הטיפול בבקשה מצד השרת יבוצע Recv של הקליינט וניתן יהיה, לפי

הודעת השגיאה, לדעת האם הקובץ התווסף או לא.

TRACE – במקרה כזה הלקוח שולח בקשת TRACE לשרת, ע"י שימוש ב SendBuff שיכיל

את הבקשה. לאחר מכן, לאחר טיפול השרת, הלקוח יבצע Recv ויקבלאת

המידע מהשרת.

**הערות לגבי קליינט:**

בכל שלב ושלב מתבצעת בדיקה האם send ו recv הצליחו. אם מסיבה כלשהי נכשל הקלינט לבצע פקודה זו, תוחזר התראה על כך עם שם השגיאה, הסוקט של הקליינט ייסגר וכמו כן תיסגר סביבת העבודה של ווינדוס לסוקטים.

היות ווידאנו שלאחר 2 דקות יסגר הקשר, מובטח כי הקליינט לא יתקע