

NEGOTIATOR - PEER PROTOCOL :

- * Her es büyük sisteme dahil olabilmek için öncelikle NEGOTIATOR'a bağlanacaktır.
 - * NEGOTIATOR bağlanan ese diğer eslerin bağlantı bilgilerini iletecektir --> CONNECT_POINT
(Yani her bağlanan es, sunucuya kaydolup kendi bağlantı adresini ve portunu sağlamalıdır)
 - * Belirli zaman aralıklarıyla (UPDATE_INTERVAL) eslere kendisine bağlı olan eslerin bağlantı adresi listesini
iletmelidir veya her es belirli zamanlarda bu bilgiyi istemelidir.
 - * Sunucu üzerinden görüntü verisi alıverisi yapılmayacaktır.
 - * Yeni bir es bağlandığında ve kendisini kaydettiğinde (REGISTER), sunucunun bir es gibi o ese bağlanmaya çalışıp
[PEER IP, PEER PORT] ikisilinde bir problem olup olmadığını kontrol etmesi gerekir.
 - * Merkez sunucu belirli zaman aralıklarında kendi üzerindeki bağlantı adres listesini
(CONNECT POINT LIST) güncelleme çalışması yapacaktır.
 - * sunucu-es arasındaki protokol soru-cevap (senkron) şeklinde olacaktır.
 - ***** sunucu eslere sorgu yapmadığı sürece bir şey söylemeyecektir.
 - * dolayısıyla bu bağlantının eslerde asenkron veri beklemesi gerekmemektedir.
 - * NEGOTIATOR-PEER bağlantısının amacı ise PEER bulusturma olacaktır.
- negociator --> amacı tamamen bağlantı bilgisinin paylaşımı

1. NEGOTIATOR - PEER PROTOCOL:

NEGOTIATOR-PEER protokolü sayesinde eşler kendilerini sunucuya kayıt ederler. Yine bu protokol sayesinde eşler kendi kimlik doğrulamalarını yaparlar (authentication). Bunun dışında bu protokol sayesinde eşler kendilerine bağlı olan eşlerin bağlantı adresleri bilgisini elde ederler. NEGOTIATOR veritabanının üzerinden bilgileri ancak bu protokol sayesinde edinebilirler. Bir eş başka bir eş ile bağlantı kurmak istediği zaman yine bu işlem NEGOTIATOR üzerinden gerçekleşmektedir. Eşler ayrıca herhangi bir bilgi değişikliği için yine NEGOTIATOR'a başvururlar. NEGOTIATOR-PEER protokol kesinlikle soru-cevap şeklindedir (senkron). Burada sadece esler NEGOTIATOR a istekte bulunabilir ve

NEGOTIATOR bu istekler doğrultusunda cevap verir. Eşlerin NEGOTIATOR'dan asenkron cevaplar beklemesine gerek yoktur.

SUNUCU-EŞ SORGULARI: (Negotiator-Peer Requests)

PRPULSE: (*Peer Pulse*)

Peer Pulse dediğimiz bu mesaj PEER_CLIENT 'tan NEGOCIATOR a yollanan bir bildirimdir. Bu bildirimin amacı bu mesajı yollayan eşin hâlihazırda çevrimiçi olduğunu ve işlem görmek istediğini iletir. Belirli bir süre zarfı içerisinde eşlerimizin her birinden bu mesajın sunucuya gelmesi beklenmektedir. (bu süre 120 saniyedir). Eğer 5 dakika (300 saniye) boyunca bu bildirim sunucuya gelmezse sunucu bu eş çevrimdışı kabul eder ve listesinden bu eş çıkarır. Her PRPULSE mesajı için sunucu NEGOK (ok mesajı) veya NEGERR (hata mesajı) cevabı vermektedir.

Girilen komut: PRPULSE

Format (PEER_CLIENT tarafı): PRPULSE

Sunucu tarafından gönderilen cevap: NEGOK veya NEGERR. NEGERR bağlantının başarısız olması durumunda yollanacaktır. NEGOK mesajında ise NEGOK 'un yanında request yapan eşin ismi (ID), IP si ve durumu (status) verilecektir. (<komut> , <user_name> , <user_IP>)

Örnek cevap formatı:

- (NEGOK, Şakir, 212.36.37.38) Bu bir tuple şeklinde gönderilebilir.
- (NEGERR , <hata_sebebi>)

PRREGIS : (*Peer Register*)

PRREGIS eşin büyük sisteme dâhil olabilmek için NEGOTIATOR'a istekte bulunduğu bir komuttur. Bu request sayesinde aday olan eş sisteme girebilir veya giremeyebilir. Sunucu eş kabul etmeden önce mesajın formatına bakar. Mesaj formatı şöyle olmalıdır:

(<PRREGIS>, <user_name>, <user_IP>). Sunucu kendi CONNECT_POINT_LIST'ine bakacaktır ve eğer kayıt olmak isteyen eş bu listede mevcut değilse ve yollanan mesaj doğru formatta atılmışsa sunucu bu yeni eş sisteme dâhil edecektir.

Karşılığında yine NEGOK veya NEGERR mesajı yollayacaktır. NEGOK mesajı döndürdüğünde, NEGOK un yanına Merhaba ve kullanıcı ismini de yazacaktır. NEGERR durumunda ise neden eşin kabul edilmediğini sisteme yazacaktır NEGERR in yanında.

Girilen komut: PRREGIS

Format (PEER_CLIENT tarafı): (PRREGIS, <user_name>, <ip_no>)

Sunucu tarafından gönderilen cevap: (<NEGOK> , <Merhaba> , <user_name>) veya (<NEGERR> , <reason>)

Örnek cevap formatları:

- (NEGOK, Merhaba, veli_34)
- (NEGERR, böyle bir kullanıcı zaten sistemde mevcut)
- (NEGERR, Yanlış format)

PRCLOSE: *(Client End Connection Message)*

PRCLOSE ile eş sistemden çıkmak istediğini belirtir. Bu mesajı alan sunucu yine diğer durumlarda olduğu gibi ya NEGOK ya da NEGERR ile cevap verecektir. Mesaj yanlış formatta atıldıysa, sunucu buna NEGERR ile cevap verecektir aksi halde NEGOK ile cevap verecektir.

Girilen komut: PRCLOSE

Format (PEER_CLIENT tarafı): PRCLOSE

Sunucu tarafından gönderilen cevap: (<NEGOK> , <Hoşçakal> , <user_name>) veya (<NEGERR> , <reason>)

Örnek cevap formatları:

- (NEGOK, Hoşçakal, ahmet_dursun)
- (NEGERR, yanlış format)

PREDIT : *(Client Profile Edit)*

PREDIT sayesinde ilgili eş sunucuya profil bilgilerini değiştirmek istediğini iletir. PREDIT ile istekte bulunan eş içinde değişiklik yapılmış mesajı sunucuya yollar. Sunucu NEGOK veya NEGERR ile yine cevap verecektir bu durumda. Mesela yukarıda örneğini verdiğimiz 212.36.37.38 IP'li Şakir adlı kullanıcı örneğini verelim. Mesaj aynen şöyle gelecek:

---- (PREDIT, hakki_92, 212.36.37.38) ---- gördüğünüz gibi burada Şakir ismi geçmiyor, eş direkt olarak değişikliği yapıyor ve öyle sunucuya yolluyor. Sunucu IP'yi kontrol eder kendi listesi üzerinde ve eğer böyle bir IP varsa değişikliği onaylar ve NEGOK mesajını yollar.

Girilen komut: PREDIT

Format (PEER CLIENT tarafı): (PREDIT , <new_user_name> , <IP_no>)

Sunucu tarafından gönderilen cevap: (<NEGOK> , <user_name_modified>) veya (<NEGERR> , <reason>)

Örnek cevap formatları:

- (NEGOK, hakki_92)
- (NEGERR, 212.26.37.38 ipli bir kullanıcı mevcut değildir).

PRFIND : (*Client Find User*)

Client bu sorguyu sunucuya gönderdiği zaman, sunucu client a kayıtlı kullanıcıların listesini döndürür. Client burada PRFIND komutunun yanında anahtar kelime ekler. Sunucu yine NEGOK komutunu döndürür, NEGOK komutunun yanında o keyword ü içeren bütün kullanıcıların listesini de gönderir. Listede o kullanıcıların bilgilerini de gönderir.

Girilen komut: PRFIND

Format (PEER CLIENT tarafı): (PRFIND , <keyword>)

Sunucu tarafından gönderilen cevap: (<NEGOK> , <user_name_list_with_information>) veya (<NEGERR> , <reason>)

Örnek request - cevap formatları:

- (PRFIND , "met")
- (NEGOK ; Ahmet, information about Ahmet ; Samet, information about Samet ; Mehmet, information about Mehmet)

PRCONN: (*Client Contact Request*)

Bu komut ile bir eş başka bir eşe bağlanmak istediğini sunucuya iletir. Server bu isteği görünce kendi veritabanında (burada contact_list oluyor)bir tarama yapar ve bağlanılmak istenilen kullanıcıyı bulur. Hedef kullanıcıyı bulduğu zaman NEGOK cevabını bu mesajı gönderen verici kullanıcıya gönderir (yani ilk kullanıcı). Hedef kullanıcı eğer PRPULSE sorgusu atarsa sunucuya, o zaman sunucu NEGOK komutunun yanında Contact requesti de atar. Bunun karşılığında hedef kullanıcı bu requesti kabul eder veya etmez. Hedef kullanıcının bu kabul veya ret cevabı PRRSP dediğimiz dediğimiz komutla olacaktır. (altta)

PRRESP: Client Contact Request – Response:

Bir kullanıcı sunucu üzerinden bir contact request mesajı alırsa bunun karşılığında PRRESP mesajı yayınlar. PRRESP mesajının içinde PRRESP komutu ve bu komutun yanında ACCEPT veya DENY durumlarını ekler. Sunucu bunun karşılığında yine NEGOK veya NEGERR mesajını döner kullanıcıya.